Sequential Procedures for Nonparametric Kernel Regression

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

T.D.K. Lasitha Sandamali Dharmasena

B. Sci. (Hons.), M. App. Sci.

School of Mathematical and Geospatial Sciences

RMIT University

Science, Engineering and Technology Portfolio

December 2008

Dedicated with the greatest affection and deepest respect to my parents

Declaration

The candidate hereby declares that:

- except where due acknowledgement has been made, the work is that of the candidate alone;
- the work has not been submitted previously, in whole or in part, to qualify for any other academic award;
- the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program;
- any editorial work, paid or unpaid, carried out by a third party is acknowledged;
- ethics procedures and guidelines have been followed.

. . . .

T.D.K. Lasitha Sandamali Dharmasena

December 2008

Acknowledgements

There are many people who have facilitated me along my journey of accomplishing this thesis.

Firstly, I am indebted to my father, Mr. Patin Dharmasena who offered me an opportunity to come Australia to carry out my postgraduate studies, for his munificent continual assistance with everything and encouraging me to further excel. Without his assistance I could never have made it.

I wish to express gratitude to my revered first supervisor Associate Professor Basil M de Silva in many ways. I have received his invaluable attachment since I commenced my postgraduate studies at RMIT University. Since then his continuous inspiration, proper guidance and dynamic foresight enhance my passion to the subject.

Very special thanks to my second supervisor Professor Panlop Zeephongsekul who has provided continuous intellectual and guidance which encouraged me to sustain my interest in this research. Your inspirational supervision not only has enhanced my research extensively but also make me very enthusiastic in my research.

To RMIT University and the APA for providing me financial support and other essential facilities.

To all the staff in the department of Mathematics and Statistics for providing friendship and tutoring assistance.

To my colleagues, friends and all others for their support during this study. Among them, Arnold, Terry, Raj and Paul deserve special mention.

To the reviewers of papers along the way my thanks.

Last but not the least in importance, to my two brothers, Aruna and Saliya for their abundant affection.

List of Tables

List of Figures

Contents

Summary

In a nonparametric setting, the functional form of the relationship between the response variable and the associated predictor variables is unspecified; however it is assumed to be a smooth function. The main aim of nonparametric regression is to highlight an important structure in data without any assumptions about the shape of an underlying regression function. In regression, the random and fixed design models should be distinguished. Among the variety of nonparametric regression estimators currently in use, kernel type estimators are most popular. Kernel type estimators provide a flexible class of nonparametric procedures by estimating unknown function as a weighted average using a kernel function. The bandwidth which determines the spread of the kernel has to be adapted to any kernel type estimator. Our focus is on Nadaraya–Watson estimator and local linear estimator which belong to a class of kernel type regression estimators called local polynomial kernel estimators.

A closely related problem is the determination of an appropriate sample size that would be required to achieve a desired confidence level of accuracy for the nonparametric regression estimators. Since sequential procedures allow an experimenter to make decisions based on the smallest number of observations without compromising accuracy, application of sequential procedures to a nonparametric regression model at a given point or series of points is considered. The motivation for using such procedures is: in many applications the quality of estimating an underlying regression function in a controlled experiment is paramount; thus, it is reasonable to invoke a sequential procedure of estimation that chooses a sample size based on recorded observations that guarantees a preassigned accuracy.

We have employed sequential techniques to develop a procedure for con-

structing a fixed-width confidence interval for the predicted value at a specific point of the independent variable. These fixed-width confidence intervals are developed using asymptotic properties of both Nadaraya–Watson and local linear kernel estimators of nonparametric kernel regression with data-driven bandwidths and studied for both fixed and random design contexts. The sample sizes for a preset confidence coefficient are optimized using sequential procedures, namely two-stage procedure, modified two-stage procedure and purely sequential procedure. The proposed methodology is first tested by employing a large-scale simulation study. The performance of each kernel estimation method is assessed by comparing their coverage accuracy with corresponding preset confidence coefficients, proximity of computed sample sizes match up to optimal sample sizes and contrasting the estimated values obtained from the two nonparametric methods with actual value or values of at a given design point or at given series of design points of interest etc.

We also employed the symmetric bootstrap method which is considered as an alternative method of estimating properties of unknown distributions. Resampling is done from a suitably estimated residual distribution and utilizes the percentiles of the approximate distribution to construct confidence intervals for the curve at a set of given design points. A methodology is developed for determining whether it is advantageous to use the symmetric bootstrap method to reduce the extent of oversampling that is normally known to plague Stein's two-stage sequential procedure. The procedure developed is validated using an extensive simulation study and we also explore the asymptotic properties of the relevant estimators.

Finally, we apply our proposed sequential nonparametric kernel regression methods to some problems in software reliability (estimating software reliability growth models) and finance (estimating capital asset pricing model).

Chapter 1

Introduction

1.1 Background

When fitting a regression model to data, the choice of parametric model depends very much on the situation being modelled. Sometimes, there are scientific reasons for modeling response variable as a particular function of explanatory variable, while at other times the model is based on experience gained through analysis of previous data sets of the same type. The restriction that a regression function belongs to a parametric family is often too rigid as this often requires that the function be linear, parabolic, periodic or monotone, each of which might be too restrictive for adequate estimation of the true regression function. If the selected function is not appropriate, then this will result in the likelihood of reaching incorrect conclusions during the regression analysis. The removal of the restriction that regression function has to belong to a parametric family will overcome the rigidity inherent in parametric regression. The approach whereby no parametric function is prescribed is referred to as nonparametric regression. The nonparametric approach to regression is desirable when a scatter plot shows no discernible simple functional form because one would want to let the data decide which function fits them best without the restrictions imposed by a parametric model. In some cases a nonparametric regression estimate will suggest a simple parametric model, while in other cases it will be clear that the underlying regression function is sufficiently complicated that no reasonable parametric model would be adequate.

In a nonparametric setting, the functional form of the relationship between the response variable and the associated predictor variables are assumed to be unknown when data is fitted to the model. The main aim of nonparametric regression is to provide a simple way of highlighting important features of the data sets without imposing any assumptions on the shape of the underlying regression function. Hence, a nonparametric approach allows the data to speak for itself. Nonparametric regression models can be used for the same types of applications such as estimation, prediction, calibration and optimization that parametric regression models are used for.

There are now several useful techniques for obtaining nonparametric regression estimates. Some of these are based on fairly simple ideas while others are more sophisticated. Among these, kernel methods (Rosenblatt, 1956; Nadaraya, 1964; Watson, 1964; Gasser and Müller, 1979; Müller, 1988; Wand and Jones, 1995), local polynomial methods (Stone, 1977; Cleveland, 1979; Fan, 1993; Fan and Gijbels, 1996), spline methods (Wahba, 1977; Eubank, 1988; Nychka, 1988; Wahba, 1990; Green and Silverman, 1994), fourier methods (Efromovich and Pinsker, 1982; Efromovich, 1999) and wavelet methods (Donoho and Johnstone, 1994; Vidakovic, 1999) are the most popular. There are several different approaches within each of these broad classes of nonparametric regression estimators. For example, the local linear estimator (Fan, 1992), is a special case of local polynomial kernel estimators discussed in

Cleveland (1979) since, it can be shown to correspond to fitting a first degree polynomial to the data with kernel function as weight. Each of these approaches has its own particular merits and weaknesses (Fan, 1996).

However, kernel type regression estimators have an advantage of mathematical and intuitive simplicity. Also, a class of kernel type regression estimates has an advantage over the other classes of regression estimators, since it depends on only one positive parameter, namely, bandwidth which controls the smoothness of the estimate. Besides specific weight sequences which have been introduced for kernel regression, splines smoothing and orthogonal series smoothing are related to each other, it is argued that one of the simplest ways of computing a weight sequence is kernel regression though. Of particular importance and simplicity, our focus is on two kernel type regression estimators namely Nadaraya–Watson and local linear which belong to a family of local polynomial kernel kernel estimators. Nadaraya (1964) and Watson (1964) introduced this family of estimates by estimating the regression function at a particular point by locally fitting a degree zero polynomials, that is, local constants. Fan (1992) pioneered local linear smoother which fits a first degree polynomial that is, local linear regression to the data via weighted least squares.

In general, local polynomial estimators (Fan and Gijbels, 1996) are superior to Nadaraya–Watson estimator in some respects (Fan, 1993), but recent contributions by Boularan et. al. (1995), Einmahl and Mason (2000) as well as Quian and Mammitzsch (2000), among others, have given evidence of continuing interest in the Nadaraya–Watson estimator. One of the strengths of this estimator certainly consists in its automatic adaptation to designs where the local polynomial estimator may not be performing reliably over all. Also, the Nadaraya-Watson estimator retains some optimality properties as demonstrated in Hardle and Marron (1985).

Let (X, Y) be a bivariate random variable with joint probability density function $f(x, y)$ where we will assume $0 < x < 1$ for simplicity. Consider a sequence of observations (x_i, y_i); $i = 1, \ldots, n$ generated by the distribution and described by the regression model

$$y_i \quad = \quad m(x_i) + \varepsilon_i; \quad i = 1, \ldots, n \tag{1.1}$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed random error terms with $\mathbf{E}(\varepsilon_i) = 0$ and $\mathbf{Var}(\varepsilon_i) = \sigma^2$. Note that the conditional expectation $\mathbf{E}(Y|X = x) = m(x)$ is commonly known as the regression function; also, the conditional variance $\mathbf{Var}(Y|X = x) = \sigma^2$. Both $m(x)$ and σ^2 are unknown and must be estimated from data. Two cases arise in practice:

- when the parametric form of $m(x)$ is known;
- when the parametric form of $m(x)$ is unknown.

In the first case, standard least squared method is used to estimate the parameters of the models and σ^2 is estimated from the residual terms. In the second case, a nonparametric approach, which involves estimating the regression function directly using the data without any parametric assumptions placed on the form of the function, is generally employed. A novel approach must also be used to estimate the conditional variance σ^2.

When using nonparametric regression methods, one question often arises: what sample size do we need to achieve a level of accuracy within some prespecified error bound. This question naturally falls into the domain of sequential procedures which in general comes in handy if we want to control the error of estimation at some preassigned level i.e. one of the key objectives is to ensure that the fitted curve $\hat{m}(x)$, based on a sample of size n, achieves a reasonably good fit to the true but unknown regression function $m(x)$. Indeed for any $d > 0, 0 < \alpha < 1$ and x_0, we wish to claim

$$Pr(\hat{m}(x_0) - d < m(x_0) < \hat{m}(x_0) + d) \geq 1 - \alpha. \tag{1.2}$$

Sequential analysis refers to the area of statistical theory and methods where the sample size is random by nature and depends on the observed data. The theoretical development of sequential procedure began with the Wald's discovery of *Sequential Probability Ratio Test* (SPRT) in the 1940's. The sequential probability ratio test is well documented in Wald (1947). Sequential analysis has made rapid advances and has undergone extensive development and has enriched statistics in general with sophisticated probability and inferential techniques. Its successes can be attributed to its various applications in applied statistics where it is used in routine statistical investigation, clinical trials, industrial process control, system reliability and life testing, time sequential application and others.

The primary goal of sequential analysis is to achieve a given accuracy specified in (1.2) above by using the smallest possible sample sizes and allowing an experimenter to make decisions based on the smallest number of observations without compromising this accuracy. The procedure is convenient and inexpensive when there is a cost involved in each stages of sampling. Decision to terminate the sampling procedure depends entirely, at each stage, on the results of the observations previously made. Typically, sequential estimation is used when there is a price attached to each observation. For an example, if we have to destroy or malfunction a product in order to get observations for sampling procedures. Although all the observations which are needed for conducting a particular experiment are freely available to the statistician, there is a often price to be paid when using outdated information.

Sequential sampling procedure is a method of statistical inference in which the number of observations are not predetermined and is obtained by the specific criterion of achieving the goal of an experiment. The procedure deals with observations which are random but not necessarily independent. One of the key elements of a sequential procedure is a stopping rule that dictates whether to stop or continue the experiment and a decision rule that tells what terminal action is to be taken about the given problem after the experimentation has stopped. When both fixed sampling procedures and sequential procedures are applicable to a given problem, the most economical one in terms of reduction of sample size, cost or duration of the experimentation is often chosen.

1.2 General View of the Problem

A closely related problem is the determination of an appropriate sample size that would be required to achieve a desired confidence level of accuracy for the nonparametric regression estimators. Although much research has been done in sequential analysis, sequential procedures are not commonly employed in practice. But they are of great importance as we can find many situations where we do not know in advance how many observations or sample size will be required to reach a decision. There has been a great deal of interest in applying *sequential procedures* to obtain optimal sample size, (refer to Ghosh et. al. 1997 for a review). In estimation, a sequential approach would involve repeated sampling with successive sample added to the samples already selected, terminating when a desired level of error of estimation is reached.

Whether one wants to estimate $m(x)$ at one single point $x = x_0$ or for all $x \in \mathbb{R}$, depending on the specific goal and error criterion, one would like to determine the sample size n in an optimal fashion. That is, in order to have the error controlled at a preassigned level, the sample size has to be adaptively estimated in the process by a positive integer valued random variable N where the event $[N = n]$ will depend only on $(X_1, Y_1), \ldots, (X_n, Y_n)$ for all $n \geq 1$. Finally, regression function $m(x)$ is estimated by nonparametric regression estimator $\hat{m}_N(x)$ i.e. nonparametric regression estimate $\hat{m}(x)$ based on the sample $(X_1, Y_1), \ldots, (X_N, Y_N)$.

We employ sequential procedures to estimate the sample size, N required to obtain fixed-width $2d\,(d > 0)$ confidence interval for an unknown regression function, $m(x)$ at a point $x = x_0$. A natural way of constructing a fixed-width confidence band for $m(x_0)$ is as follows. Suppose that $\hat{m}(x_0)$ is an estimator of $m(x_0)$, then a $100(1-\alpha)\%$ confidence band for $m(x_0)$ where $x_0 \in [0, 1]$ is a stopping random variable N such that

$$Pr\{|\hat{m}_N(x_0) - m(x_0)| \leq d\} \geq 1 - \alpha \tag{1.3}$$

for a given d.

Essentially, the problem of constructing an interval $I_{N,d} = (\hat{m}_N(x_0) - d, \hat{m}_N(x_0) + d)$ is translated into a problem of determining the sample size. Indeed if the sample size N is too small then the interval $I_{N,d}$ will not achieve the preset coverage probability $1-\alpha$. The key difficulty in determining an N or the *stopping rule* that would achieve (1.3), is in deriving the distribution of $|\hat{m}(x_0) - m(x_0)|$.

Usually, in principle, confidence intervals can be obtained using asymptotic or approximate normality results for $\hat{m}(x_0)$. However, if the limiting bias and variance are unknown then they have to be estimated consistently in order to construct an asymptotic confidence interval. Hence, the construction of asymptotic confidence intervals for a value of the regression function $m(x)$ begins with obtaining asymptotic properties of nonparametric kernel regression estimators. We use kernel type regression estimators not because it is necessarily the best method to use in all circumstances, but because of its wide applicability, particularly in univariate case, and also its properties are best understood. Note that, standard sequential procedures, namely Stein's two-stage sequential procedure, modified two-stage sequential procedure and purely sequential procedure etc. rely on normal approximation. Thus, we can not overlook the fact that accuracy of these procedures heavily depends on how good this normal approximation is.

Confidence bands for the unknown regression function can also be derived using bootstrapping by obtaining the percentiles of the approximate distribution. It is a well known fact that for a wide class of statistical analysis, a bootstrap approximation has a higher degree of accuracy and is therefore a popular tool for approximating sampling distribution of an estimate of interest. Hence, we suggest an approach which combines bootstrap ideas with the sequential procedures where the distribution of $\hat{m}(x_0) - m(x_0)$ is not approximated by the estimated asymptotic distribution but by a distribution obtained from resampling and whose quantiles can therefore be computed. Approximate fixed-width confidence intervals are then be constructed by employing these quantiles.

This suggest a way in which the distribution of the nonparametric estimate about the true curve at some point of interest may be approximated by the distribution of suitable nonparametric estimates based on bootstrap samples. The proposed bootstrap sequential procedures estimate unknown regression function $m(x)$ at a given point $x = x_0$ using a smallest possible sample size with pre-assigned level of accuracy. Our endeavor is to highlight the advantage of using the bootstrap approximation especially when it is difficult to obtain the theoretical distribution of estimates due to unknown terms.

The procedures developed in this study should find wide applicability since many practical problems which arise in practice involve estimating an unknown function.

1.3 Literature Review

Recall that we review the proposed study as an extension of ideas used in sequential kernel density estimation to nonparametric kernel regression estimation. Research into sequential density estimation was first conducted by Yamato (1971). Wagman and Davis (1975) presented a naive sequential nonparametric density estimation procedure using kernel estimates proposed by Parzen (1962). The asymptotic distribution of the stopping variable was also examined. Stute (1983) constructed sequential fixed-width confidence intervals for an unknown density function $f(x)$ at a point $x = x_0$. The efficiency of that procedure was measured in terms of the expected stopping time.

Isogai (1987) considered procedure for construction of confidence interval for a nonparametric density function at a given point based on recursive estimation of the kernel function. He also investigated the asymptotic consistency of the estimated density function. Kundu and Martinsek (1994) and Kundu (1994) looked at the problem of estimating stopping variables N via two-stage and purely sequential procedures and obtained results for $\mathbf{E}[N]$, J_N/w and $\mathbf{E}[J_N/w]$ as $w \to 0$ where $J_N = \int_x |\hat{f}_N(x) - f(x)|dx$ and they wished to have $\mathbf{E}[J_N] \leq w$ for preassigned $w > 0$. Xu and Martinsk (1995) presented sequential procedures for estimating $f(x)$ on a bounded interval and obtained the relevant asymptotic results.

Stein (1945) proposed a sequential procedure which uses two steps to obtain final sample size. However, Stein's (1945) two-stage procedure oversamples and fails to attain the asymptotic first order efficiency (Ghosh et. al., 1997) even though it meets the property of consistency. Over the years a number of authors applied various modifications to two-stage procedure to overcome the oversampling problem. Stein's procedure is initially modified by Cox (1952) and further improvement was made by Mukhopadhyay (1980) who introduced the modified two-stage sequential procedure. Consistency and asymptotic first order efficiency (i.e. $\lim_{d\to 0} \mathbf{E}\left[\frac{N}{n_{opt}}\right] = 1$; n_{opt} is referred to as the optimal fixed sample size) properties of the modified two-stage procedure have been established by Ghosh and Mukhopadhyay (1981). Recent research of Mukhopadhyay and Duggan (1997) proposed a revised version of Stein's two-stage procedure with a second order asymptotic efficient (i.e. $\liminf_{d\to 0} \mathbf{E}[N - n_{opt}] = 0$) and consistency properties.

Adding one extra stage to Stein's sampling procedure gives a method which combines simplicity with the efficiency of the fully sequential procedure. The idea of three operations instead of two was introduced by Hall (1981) for the estimation of the normal mean. His triple sampling procedure was designed to combine the operational savings made possible by sampling in three batches and the efficiency of purely sequential procedures. The basic idea of the three-stage sampling was put forth in Mukhopahyay (1976) for constructing a fixed-width confidence interval for the mean of a normal distribution when the variance was unknown, and thereby obtaining first-order asymptotic results.

de Silva (2000) developed a fixed-width confidence interval procedure and critically examined its consistency property for estimating unknown density function at a point $x = x_0$, $f(x_0)$ by extending ideas from Carrol (1976) and

Isogai (1987). Subsequently, de Silva and Mukhopadhyay (2001) employed sequential procedures namely purely sequential, accelerated sequential, two-stage and three-stage to estimate the sample size N required to obtain a fixed-width confidence intervals for $f(x)$ at a point $x = x_0$ and a comparison was performed between them. One may refer to Mukhopadhyay and Solanky (1994) or Ghosh et. al. (1997) for more details on sequential procedures.

1.4 Contributions

The major contributions met by this thesis are

- The introduction of a quick and simple method of bandwidth selector which does not require estimating quantities such as derivatives of unknown regression function, pilot bandwidths etc. as in other proposed bandwidth selectors in nonparametric regression estimation. The proposed bandwidth selector is also more attractive as we need to have fast automatically generated kernel estimates for computer algorithms that require many regression estimation steps.
- The investigation into whether a residual variance estimate use in a particular situation is accurate because this is a crucial issue in assessing performance of various sequential sampling stopping rules.
- The construction of a fixed-width confidence interval for the predicted value at a specified point of the independent variable with preassigned accuracy for fixed equidistant regression model using two-stage and modified two-stage sequential procedures. This is achieved by employing some asymptotic properties of the Nadaraya–Watson and local linear estimators.

- The determination of an appropriate sample size that would be required to achieve a desired confidence level of accuracy for the nonparametric regression estimators using two-stage, modified two-stage and purely sequential procedures for random designed regression models.

- The construction of bootstrap symmetric fixed-width confidence interval for the predicted value at a specified point of the independent variable with preassigned accuracy for both fixed equidistant and random design regression models.

- The development of a methodology for determining whether it is advantageous to use the symmetric bootstrap method to reduce the extent of oversampling that is normally known to plague Stein's two-stage sequential procedure for both fixed equidistant and random design regression models.

- The investigation of potential benefits of using sequential nonparametric kernel regression curve estimation to fit software reliability growth models. This approach is novel and is of great potential benefit since it does with the need to estimate parameters of such models, a process which is often impossible to accomplish.

- The development of a data-driven sequential nonparametric procedure which allows the investor to analyse the relationship between the excess rate of returns of an asset and the excess rate of returns by the market using the shortest period of historical data. We develop a robust sequential nonparametric version of the Capital asset pricing model (CAPM) that can be used when the underlying parametric assumptions fail.

1.5 Structure of the Thesis

Chapter 2: is devoted to brief introduction to nonparametric kernel regression estimators and their asymptotic properties. We study elementary properties of the nonparametric kernel regression estimators, namely Nadaraya–Watson estimator and local linear estimator for univariate data in detail. We identify two distinct types of design, i.e. fixed (equidistant or non-equidistant) design and conditioned random design. Asymptotic expressions for the bias and variance for fixed design and conditional bias and variance for random design are derived and we use these to investigate how the mean squared error and integrated mean squared error behave. The choice of the kernel function is one vital concern as it is desirable to base the choice of kernel function based on computational effort involved. We restrict our attention to a reasonable and more simpler data driven bandwidth selector which does not require the estimation of quantities such as derivatives of unknown regression function, pilot bandwidths etc. We also conduct a simulation study to examine relative merits of two selected regression estimators i.e. Nadaraya–Watson and local linear estimators for different regressions functions, kernel functions, sample sizes, data designs etc. Simulation result has become an important tool to assess the performance of proposed bandwidth selector.

Chapter 3: The properties of the variance estimators in nonparametric regression based on quadratic form are investigated. In particular, two classes of estimators are compared: difference-based estimators and curve fitting estimators. Our discussions are presented for residuals based on kernel type regression estimators. An elementary account of bias and variance properties and approximate and/or asymptotic distribution of each residual variance estimator are examined. We address the problem of which of these is the better residual variance estimator. Besides we are more interested in being confident that the residual variance estimate use in a particular situation is accurate. Even though a considerable progress has been made in the development of highly performing residual variance estimators, no estimate comes with a guarantee that it will work well in all cases. In fact, more precise residual variance estimate is necessary in constructing confidence interval for unknown regression function. Hence, selected residual variance estimators are compared using an extensive simulation study for different cases depending on the data design, distribution of residual variance, sample size and underlying regression function.

Chapter 4: We develop a procedure for constructing a fixed-width confidence interval for the predicted value at a specified point of the independent variable. The optimal sample size for constructing this interval is obtained using the purely, two-stage and modified two-stage sequential procedures together with asymptotic properties of the Nadaraya–Watson and local linear estimators. The methodology for constructing fixed-width confidence intervals with a given coverage probabilities is studied in both fixed equidistant and random design contexts. Finally, a large-scale simulation study is performed to compare the performance of proposed confidence bands based on the local linear estimator with those constructed by using Nadaraya–Watson estimator. We also assess whether both estimators are shown to have asymptotically correct coverage properties or not.

Chapter 5: We employ symmetric bootstrap method, which is considered as an alternative method of estimating properties of unknown distributions, to construct a fixed-width confidence interval in nonparametric kernel regression estimation. The sample size for a preset confidence interval is optimised using a two-stage sampling procedure. Resampling is done from a suitably estimated residual distribution and utilizes the percentiles of the approximate distribution to construct confidence intervals for the curve at a set of given design points. Both fixed equidistant and random designed nonparametric regression models with one independent variable are considered and the decision to terminate the sampling procedure depends, at each stage, on the results of the observations previously made. A methodology is developed for determining whether it is advantageous to use the symmetric bootstrap method to reduce the extent of oversampling that is normally known to plague Stein's two stage sequential procedure. The procedure developed is validated using an extensive simulation study.

Chapter 6: The problem of sequentially selecting bivariate data points for a nonparametric regression curve estimation is considered. First two applications refer to data points of explanatory variable of interest are in the form of fixed equally spaced design whereas the third application corresponds to random design data. In all three applications, sample size consideration based on using Nadaraya–Watson and local linear methods is also considered.

We use nonparametric kernel regression methods to predict the growth of software reliability. The main advantage of using these methods is that they place minimum requirement on the distributional form of the stochastic process which gave rise to software failure data and hence dispense with the need to estimate parameters from complex models. Numerical examples involving four sets of real software data are presented.

Two-stage sequential kernel regression procedure to estimate row average intensity of a digital photo of Leonardo da Vinci's painting, “Mona Lisa” in each row of the image is employed.

We also develop a sequential nonparametric kernel regression approach to estimate capital price asset pricing model (CAPM) when the underlying assumption of existence of linear relationship fails, using the smallest possible sample with a given accuracy.

Chapter 7: Conclusions and suggestions for further work are presented.

1.6 Publications

- Materials from Chapter 4 have been accepted for publication in the refereed journals: *Australia and New Zealand Industrial Applied Mathematics ANZIAM Journal* and *International Association of Engineers IAENG Journal*. Further, material from this chapter has been published in the refereed conference proceedings of: *International Multi-Conference of Engineers and Computer Scientists 2008 (IMECS 2008).*
- Materials from Chapter 5 have been published in the refereed conference proceedings of: *International Conference on Data Management ICDM 2008.*
- Materials from Chapter 6 have been published in the refereed conference proceedings of: 36th *Australian Conference of Economists* and 14th *ISSAT International Conference on Reliability and Quality in Design.*

Detailed references of the above papers are given in the bibliography.

Chapter 2

Nonparametric Kernel Regression Estimation

2.1 Introduction

To estimate regression curves, their derivatives and other curves of relevance without the restrictive assumptions of parametric model a number of different methods have been devised. Nonparametric simple regression is often called scatterplot smoothing because an important application is to tracing a smooth curve through a scatterplot of y against x. Many useful techniques have been proposed for univariate smoothing. Kernel regression estimators are more popular as they have an advantage of mathematical and intuitive simplicity. In the context of kernel regression traditional approaches have involved the Nadaraya (1964) and Watson (1964) estimator, local polynomial kernel estimators (Stone, 1977) and some alternative kernel estimators (Gasser and Müller, 1979; Priestley and Chao, 1972).

The basic assumption in nonparametric regression is the existence of a smooth function $m(\cdot)$ relating the response variable Y and explanatory variable or predictor X:

$$Y = m(X) + \varepsilon \tag{2.1}$$

where ε is an error component.

Among several proposed kernel methods for estimating $m(\cdot)$ Nadaraya–Watson and local linear estimators are more popular. Both estimators are linear smoothers that is linear combination of the observed responses.

In regression, random and fixed design models should be distinguished. Depending upon the probabilistic structure in the data $(X_i, Y_i) : 1 \leq i \leq n$, the regression is referred as random design or fixed design regression. The first case occurs when the predictors, x_i's are ordered non-random numbers either equidistant i.e. $|x_{i+1} - x_i|$ is a constant for all i or non-equidistant. For the univariate fixed design case the response variable Y is assumed to satisfy

$$Y_i = m(x_i) + \varepsilon_i, \quad i = 1, \ldots, n \tag{2.2}$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are independent random variables for which $\mathbf{E}[\varepsilon_i] = 0$, $\mathbf{Var}[\varepsilon_i] = \sigma^2$ where σ^2 is a constant. Note that throughout this thesis it is assumed that $\varepsilon_1, \ldots, \varepsilon_n$ have the same probability distribution. Since $\mathbf{E}[Y_i|X = x] = m(x)$, $m(\cdot)$ is called the mean regression function or simply regression function.

Random design occurs when the data come from a joint probability density function $f(x, y)$ that is the point x itself is the observed value of a random variable X. The discussion above leads to a random variable X is observed, and if $X = x_0$, then an observation is taken at the point x_0. One of the most important question might be posed in this framework is, given that $X = x_0$, construct a predictor of Y corresponding to x_0. The proposed predictor is the conditional expectation of Y, given $X = x_0$. Hence, the value of the function $m(x)$ corresponding to a predictor value x_0 is equivalent to the expectation of response variable Y under the condition that the value of the predictor is fixed at x_0. For random design data the regression model of interest is defined to be:

$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, \ldots, n \tag{2.3}$$

where conditional on $X_1, \ldots, X_n$ and $\{\varepsilon_i\}_{i=1}^n$ are independent random variables with zero mean and constant variance σ^2. Also in the random design context,

$$m(x) = \mathbf{E}[Y|X = x] \quad \text{and} \quad \mathbf{Var}[Y|X = x] = \sigma^2. \tag{2.4}$$

From the definition of $\mathbf{E}[Y|X = x]$:

$$m(x) = \mathbf{E}[Y|X = x] = \int_y y f(y|x) dy. \tag{2.5}$$

This conditional distribution can be expressed in several ways. In particular:

$$m(x) = \mathbf{E}[Y|X = x] = \frac{\int_y y f(x, y) dy}{f(x)}. \tag{2.6}$$

The following multiplication rule

$$f(x, y) = f(y|x) f(x)$$

is used to derive (2.6).

There are many versions of kernel type regression estimators. Some of them perform well for random design data such as in observational studies and others act upon either fixed equidistant or fixed non-equidistant designs. Besides, most nonparametric kernel regression estimators have boundary effects and modifications are necessary near boundary points.

For kernel regression curve fitting, we are interested in weighting the response variable Y in a certain neighbourhood of x_0 and weight the observations Y_i depending on the distance of x_0 to X_i scaled by a bandwidth h_n.

Thus kernel regression estimator takes the general form of

$$\hat{m}(x_0) = \hat{m}_{h_n}(x_0) = n^{-1} \sum_{i=1}^{n} w_{h_n,i}(x_0; X_1, \ldots, X_n) Y_i \tag{2.7}$$

where weight function w_{h_n} depends on the bandwidth h_n also called the smoothing parameter or window width by some authors and the observations $x_1, \ldots, x_n$ of explanatory variable X.

The kernel regression estimator $\hat{m}_{h_n}(\cdot)$ of course depends on the bivariate data $(X_1, Y_1), \ldots, (X_n, Y_n)$ as well as on the kernel $K(\cdot)$ and bandwidth h_n will not be generally be expressed explicitly. The practical implementation of the kernel type regression estimator requires the specification of the bandwidth h_n. Bandwidth h_n controls the amount by which the data are smoothed to produce the estimate. Hence bandwidth h_n plays the role of a scaling factor which determines the spread of the kernel. There is as yet no universally accepted approach to this problem. We can choose bandwidth h_n either subjectively or objectively using data on hand. In Section 2.7 various methods for choosing the bandwidth are explained.

A kernel $K(\cdot)$ is a continuous, bounded and symmetric real function. For simplicity we shall assume throughout this discussion that the kernel $K(\cdot)$ satisfies

$$\int_t K(t)dt = 1, \qquad \int_t tK(t)dt = 0 \quad \text{and} \quad \int_t t^2 K(t)dt = k_1 \neq 0.$$

Usually the kernel $K(\cdot)$ is chosen to be a probability density function which is unimodal and symmetric about zero and the constant k_1 will then be the variance of the distribution. It should be stressed that the kernel $K(\cdot)$ is under the user's control and therefore it is only necessary for practical purposes to consider results which hold for the particular kernel being used. Hence, a discussion of using different kernel functions in nonparametric kernel regression is done in Section 2.2.

Order (O and o) and asymptotic notations given throughout this chapter are defined as follows:

Let each a_n and b_n be sequences of real numbers. Then we will say that a_n is of order b_n as $n \to \infty$ and write

$$a_n = O(b_n) \quad \text{as} \quad n \to \infty \quad \text{if and only if} \quad \limsup_{n \to \infty} |a_n/b_n| < \infty$$

which is equivalent to saying $a_n = O(b_n)$ if $|a_n/b_n|$ remains bounded as $n \to \infty$.

We say that a_n is of small order b_n and write

$$a_n = o(b_n) \quad \text{as} \quad n \to \infty \quad \text{if and only if} \quad \lim_{n \to \infty} |a_n/b_n| = 0.$$

In order to obtain asymptotic approximations in kernel regression estimators, we incorporate the above with a Taylor's series expansion.

Theorem 1 *Taylor's Theorem: Suppose that $f(x)$ is a real valued function defined on $\mathbb{R}$ and let $x \in \mathbb{R}$. Assume that $f(x)$ and p continuous derivatives in an intervals $(x - \delta, x + \delta)$ for some $\delta > 0$. Then for any sequence a_n converging to zero,*

$$f(x + a_n) = \sum_{j=0}^{p} \frac{a_n^j}{j!} f^{(j)}(x) + o(a_n^p).$$

We also consider bandwidth h_n is a non-random sequence of positive numbers and assume that h_n satisfies:

$$\lim_{n \to \infty} h_n = 0 \quad \text{and} \quad \lim_{n \to \infty} nh_n = \infty.$$

In other words h_n approaches to zero but at a rate slower than n^{-1}.

As done in classical parametric statistics here we use mean squared error (MSE) and mean integrated squared error (MISE) to measure the closeness of a nonparametric kernel regression estimator $\hat{m}_{h_n}(\cdot)$ to its true value $m(\cdot)$ which are defined by

$$\begin{aligned} MSE(\hat{m}(x_0)) &= \mathbf{E}\left[\hat{m}(x_0) - m(x_0)\right]^2 \\ &= \mathbf{Var}[\hat{m}(x_0)] + (\mathbf{E}[\hat{m}(x_0)] - m(x_0))^2 \end{aligned}$$

and

$$\begin{aligned} MISE &= \int_{x_0} MSE(\hat{m}(x_0))dx_0 \\ &= \int_{x_0} \mathbf{E}\left[\hat{m}(x_0) - m(x_0)\right]^2 dx_0. \end{aligned}$$

In this chapter, elementary properties of the nonparametric kernel regression estimators in the univariate case will be discussed in more detail. The concentration on the kernel regression estimators is not intended to imply that the method is the best to use in all circumstances but there are several reasons for considering the regression estimators based on kernel method first of all. The method is of wide applicability and it is certainly worth understanding its behaviour before going on to consider other methods and discussion of these properties raises issues which relate to other methods of nonparametric regression estimation. Throughout this chapter except where otherwise stated $\hat{m}_{h_n}(\cdot)$ will be the nonparametric kernel estimator with kernel $K(\cdot)$ and window width h_n as explained in Section 2.7. In the later section of this chapter we shall derive asymptotic expressions for the bias and variance for fixed design and conditional bias and conditional variance for random design and use these to investigate how the mean square error and integrated mean square error will behave. An extensive simulation study has been carried over in Section 2.8 to discuss the closeness of the estimator $\hat{m}_{h_n}(\cdot)$ to the true value $m(\cdot)$ in various senses.

2.2 Kernel Function

Almost all kernel regression curve estimators are weighted averages of the response variable Y. However, the methodology of defining weights w_i; $i = 1, \ldots, n$ are different from technique to technique but all the weights are calculated using a kernel function. A conceptually simple approach to represent the sequence of weights $w_1, \ldots, w_n$ is to describe the shape of the weight function by a density function with a scale parameter adjusting the size and the form of the weights near x_0. It is a reasonably frequent practice to refer to this shape function as kernel $K(\cdot)$. In nonparametric regression, the size of the weights is parameterized by a scale parameter h_n which is known as bandwidth.

The kernel weights $K(\cdot)$ are calculated under two distinct approaches: (i) a fixed window width and (ii) a fixed fraction of the data. When the generic kernel has compact support such as uniform on $[-1, 1]$, triangular, quadratic or biweight, the estimator depends only on those pairs whose x_i are in the interval $(x_0 - h_n, x_0 + h_n)$. The size of the neighbourhood is called the bandwidth or window width, denoted by h_n. In this formulation smoothing parameter is a scale parameter. The second approach uses the n nearest neighbors to x_0. In both methods the pairs with x_i close to x_0 influence the estimate purely based on how distant x_i be from x_0. These two distinct methods yield either a random number of observations x_i's within a fixed interval h_n or a fixed number of observations n_0 within an interval of random width. For uniformly distributed x's these two are equivalent. Whereas for the case of random design the estimates and their properties differ. Here we employ the first approach which gives higher weight to observations that are close to the focal point x_0 and lower weight to those which are remote.

The choice of the kernel function $K(\cdot)$ is one vital concern in kernel regression estimation. In general a variety of kernel functions are possible. However, the choice is limited by both theoretical and practical considerations. For example, we might restrict attention to kernel functions that are zero outside some fixed interval because kernel functions that take on very small values can result in numerical underflow on a computing programmes. Apart from that some kernels are not differentiable at particular points and cause '0/0' cases which define $\hat{m}_{h_n}(x)$ as being 0. Further more some kernels are not defined at all possible values $(-\infty, \infty)$ for instance a commonly used kernel function, Epanechnikov kernel developed by V.A. Epanechnikov (1969) which is of parabolic shape define only in the range of $(-\sqrt{5}, \sqrt{5})$. Since the regression estimates are based on the local regression no negative weight $K(\cdot)$ should be used.

The frequently used kernel functions or weight functions are given in Table 2.1. Figure 2.1 shows the shape of the different kernels given in the Table 2.1 in a range of $t \in (-3, 3)$.

Table 2.1: Functions of Various Kernels.

Kernel	$K(t)$	
Epanechnikov	$\frac{3}{4}(1-\frac{1}{5}t^2)/\sqrt{5}$	for $\|t\| < \sqrt{5}$
	0	otherwise
Biweight	$\frac{15}{16}(1-t^2)^2$	for $\mid t \mid < 1$
	0	otherwise
Double Exponential	$\frac{1}{2}\exp(-\|t\|)$	for $\mid t \mid < 1$
	0	otherwise
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-(1/2)t^2)$	$-\infty < t < \infty$
Uniform	$\frac{1}{2}$	for $\mid t \mid < 1$
	0	otherwise

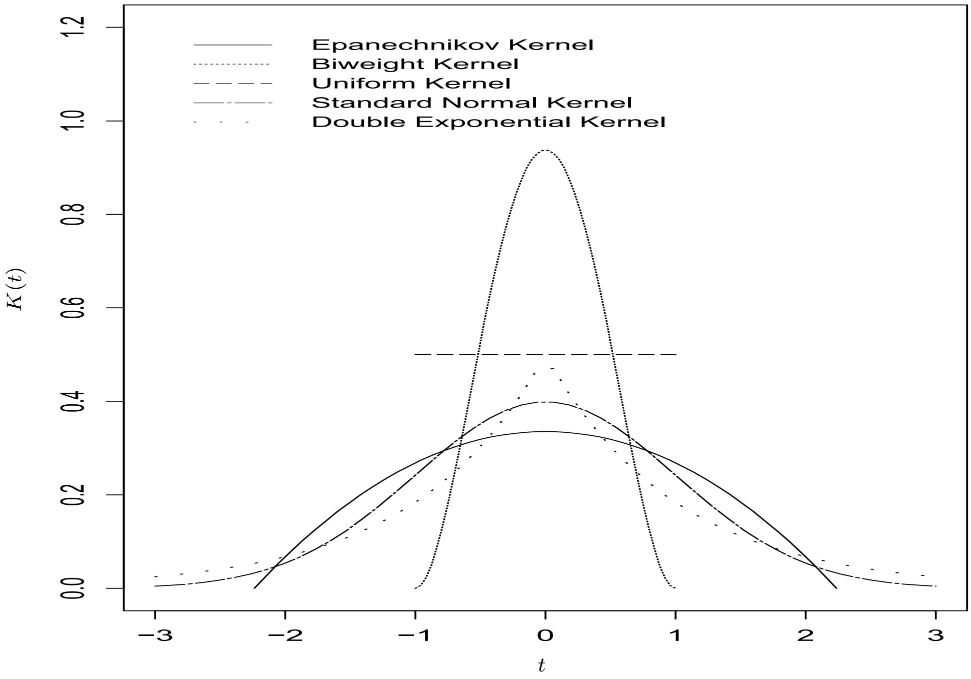


Figure 2.1: Shapes of Different Kernel Functions.

2.3 Local Polynomial Kernel Regression

A class of kernel type regression estimators called local polynomial estimators was systematically studied by Stone (1977), Cleveland (1979) and Müller (1987). In this approach, regression function is estimated at a particular point by locally fitting p^{th} degree polynomial to the data via weighted least squares. From a computational point of view local polynomial estimators are attractive due to their simplicity.

In parametric regression we use higher-order polynomials to approximate a large class of possible regression curves. The degree of the polynomial in parametric regression plays a role analogous to the degree of smoothing in nonparametric regression. However, the use of higher order polynomials for nonparametric regression can not be recommended as the order of polynomial increases even parametric polynomial fits can exhibit rapid oscillations. Stone (1977) has shown that a low-order polynomial fit should be adequate locally if the true regression curve is smooth. Recent work on local polynomial fitting includes Fan (1992, 1993), Fan and Gijbels (1996) and Ruppert and Wand (1994) who gave a detailed discussion of the advantages in using this method.

The weighted least square regression fits the equation

$$Y_i = \beta_0 + \beta_1(X_i - x_0) + \beta_2(X_i - x_0)^2 + \ldots + \beta_p(X_i - x_0)^p + \varepsilon_i; \quad i = 1, \ldots, n. \tag{2.8}$$

Local polynomial regression extends kernel estimation to a polynomial fit at the focal point x_0, using local weights, $w_i = h_n^{-1}K[(X_i - x_0)/h_n]$. The local polynomial kernel estimator is a weighted regression on the data, centred about x_0. The goal is to estimate $m(x_0)$. Note that If we let p be the degree of the polynomial being fit then at a point x_0 the estimator $\hat{m}(x_0; p, h_n)$ is obtained by fitting the polynomial

$$\beta_0 + \beta_1(X_i - x_0) + \ldots + \beta_p(X_i - x_0)^p$$

to the (X_i, Y_i) using weighted least squares with kernel weights.

The constant regression fit which is the polynomial of degree 0 to a scatter diagram is $\hat{m}_{h_n}(\cdot) = \bar{y}$ which is the estimate that minimizes the least-squares criterion:

$$\bar{y} = arg\ min_{\beta_0} \sum_{i=1}^{n} (Y_i - \beta_0)^2 \tag{2.9}$$

where $arg\ min_{\beta_0}$ indicates that the constant $\beta_0 = \bar{Y}$ is the argument that minimizes the criterion. Now consider a local constant fit at x_0 to the data. Here local means including only those data (X_i, Y_i) for which $X_i \in (x_0 - h_n, x_0 + h_n)$ in the sum in (2.9) or it may mean including only the q design points of explanatory variable nearest to x_0. Now we can introduce kernel function $K(\frac{X_i - x_0}{h_n})$ to facilitate as weighting function to indicate precisely which terms are included and the weights of the selected design points. The local constant fit $\hat{m}(x)$ is the estimate that minimizes

$$arg\ min_{\beta_0} \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(Y_i - \beta_0)^2 \tag{2.10}$$

with respect to β_0. That is

$$\frac{d}{d\beta_0}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(Y_i - \beta_0)^2 = 0,$$

which leads to

$$\begin{aligned} -2\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(Y_i - \hat{\beta}_0) &= 0 \\ \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)Y_i &= \hat{\beta}_0 \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) \\ \hat{\beta}_0 &= \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)Y_i}{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)}. \end{aligned}$$

Hence, $\hat{m}(x_0; 0, h_n) = \hat{\beta}_0$ where $\hat{m}(x_0; 0, h_n)$ is the intercept coefficient of $p = 0$ polynomial and

$$\hat{m}(x_0; 0, h_n) = \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)Y_i}{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)} \tag{2.11}$$

which is precisely the Nadaraya–Watson kernel estimator denoted by $\hat{m}_{h_n,NW}\ (x_0) = \hat{m}(x_0; 0, h_n)$ in the later stage of this chapter as well as other chapters. Note that $\hat{m}(x_0; p, h_n)$ is the kernel regression estimator of $m(x_0)$ based on p^{th} degree polynomial. This point-wise result can be extended to the entire regression function by defining as

$$\hat{m}(x) = arg\ min_{\beta(x)} \int_x \sum_{i=1}^{n} K\left(\frac{X_i - x}{h_n}\right)(Y_i - \beta(x))^2 dx \tag{2.12}$$

where $\beta(x) = \beta_0 + \beta_1(X_i - x) + \ldots + \beta_p(X_i - x)^p$.

The above integrand is minimized by $\hat{\beta}(x) = \hat{m}(x)$ as in (2.11) for each x_0.

Now define $\beta(x_0) = \beta_0 + \beta_1(X_i - x_0)$ and then minimizing

$$\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)[\beta_0 + \beta_1(X_i - x_0) - Y_i]^2$$

with respect to β_0

$$2\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)\left[\hat{\beta}_0 + \hat{\beta}_1(X_i - x_0) - Y_i\right] = 0 \tag{2.13}$$

and with respect to β_1

$$2\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)\left[\hat{\beta}_0 + \hat{\beta}_1(X_i - x_0) - Y_i\right](X_i - x_0) = 0, \tag{2.14}$$

our aim is to find $\hat{m}(x_0; 1, h_n)$. From (2.13)

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) Y_i - \hat{\beta}_0 \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0)}. \tag{2.15}$$

Now substituting (2.15) to (2.14)

$$\hat{\beta}_0 \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0) + \left[\frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) Y_i - \hat{\beta}_0 \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0)}\right]$$

$$\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0)^2 - \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) Y_i(X_i - x_0) = 0$$

$$\hat{\beta}_0 \left[\left(\sum_{i=1}^{n} K(\cdot)(X_i - x_0)\right)^2 - \sum_{i=1}^{n} K(\cdot) \sum_{i=1}^{n} K(\cdot)(X_i - x_0)^2\right]$$

$$= \sum_{i=1}^{n} K(\cdot) Y_i(X_i - x_0) \sum_{i=1}^{n} K(\cdot)(X_i - x_0) - \sum_{i=1}^{n} K(\cdot) Y_i \sum_{i=1}^{n} K(\cdot)(X_i - x_0)^2$$

$$\begin{aligned}\hat{\beta}_0 &= n^{-1} \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) Y_i\left[(X_i - x_0)s_1 - s_2\right]}{s_1^2 - s_0 s_2} \\ &= \hat{m}(x_0, 1, h_n) \quad \text{(say)}\end{aligned} \tag{2.16}$$

where $s_j = n^{-1}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0)^j$ and $j = 0, 1, 2$.
Note that $\hat{m}(x_0, 1, h_n)$ is called local linear estimator $\hat{m}_{h_n,LL}(x_0) = \hat{m}(x_0; 1, h_n)$.

Fan (1992) has showed that the local linear regression estimator is superior to other kernel type regression estimators in terms of ability to adapt to both random and fixed designs, to both highly clustered and nearly uniform designs and also both interior and boundary points. He also showed that the local linear kernel regression estimators have high efficiency among all possible nonparametric regression estimators based on kernel, orthogonal series and spine methods.

Let p be the degree of the polynomial being fit. At a point x_0 the estimator $\hat{m}(x_0; p, h_n)$ is obtained by fitting the polynomial $\sum_{j=0}^{p} \beta_j (X - x_0)^j$ to the (X_i, Y_i) using weighted least squares with kernel weights $h_n^{-1} K\left(\frac{X_i - x_0}{h_n}\right)$. The estimator of $\hat{m}(x_0; p, h_n)$ is the intercept of the fit $\hat{\beta}_0$ where $\hat{\mathbf{b}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^T$ is the minimizer of

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_{j=0}^{p} \beta_j (X_i - x_0)^j \right\}^2 h_n^{-1} K\left(\frac{X_i - x_0}{h_n}\right).$$

Standard weighted least squares theory leads to the solution $\hat{\mathbf{b}} = (\mathbf{X^T W X})^{-1}\mathbf{X^T W Y}$ where $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ is the vector of responses,

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 - x_0 & \cdots & (X_1 - x_0)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n - x_0 & \cdots & (X_n - x_0)^n \end{bmatrix}$$

is an $n \times (p+1)$ design matrix and

$$W = diag\left(h_n^{-1}\left\{K\left(\frac{X_1 - x_0}{h_n}\right), \ldots, K\left(\frac{X_n - x_0}{h_n}\right)\right\}\right)$$

is an $n \times n$ diagonal matrix of weights.

The estimator $\hat{\mathbf{b}}$ of $\mathbf{b}$ is obtained as solution of the linear system $\mathbf{X^T W X \hat{b}} = \mathbf{X^T W Y}$. Given that $\hat{m}(x_0; p, h_n)$ is the estimator of $m(x_0)$

and is the intercept coefficient we obtain

$$\hat{m}(x_0; p, h_n) = \mathbf{e}^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{Y} \tag{2.17}$$

where $\mathbf{e}$ is the $(p+1)\times 1$ vector having 1 in the first entry and zero elsewhere. Denote $\mathbf{S} = \mathbf{X^T W X} = \begin{bmatrix} s_0 & \cdots & s_p \\ \vdots & & \vdots \\ s_p & \cdots & s_{2p} \end{bmatrix}_{(p+1)\times(p+1)}$ with elements

$$s_j = h_n^{-1}\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right)(X_i - x_0)^j$$

for $j = 0, 1, \ldots, 2p$.

Then the so called Nadaraya–Watson estimator $\hat{m}_{h_n,NW}(\cdot)$ can be considered as a special case within this class because it can be shown as fitting a zero degree polynomials ($p = 0$), that is, local constants;

$$\hat{m}_{h_n,NW}(x_0) = \hat{m}(x_0; 0, h_n) = \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h_n}\right) Y_i}{\sum_{j=1}^{n} K\left(\frac{X_j - x_0}{h_n}\right)}. \tag{2.18}$$

Whereas local linear kernel estimator $\hat{m}_{h_n,LL}(\cdot)$ corresponds to fitting first degree polynomial ($p = 1$)

$$\begin{aligned} \hat{m}(x_0; 1, h_n) &= \frac{n^{-1}\sum_{i=1}^{n} \{\hat{s}_2(X; h_n) - \hat{s}_1(X; h_n)(X_i - x_0)\} K\left(\frac{X_i - x_0}{h_n}\right) Y_i}{\hat{s}_2(X; h_n)\hat{s}_0(X; h_n) - \hat{s}_1(X; h_n)^2} \\ &= \hat{m}_{h_n,LL}(x_0) \end{aligned} \tag{2.19}$$

where $\hat{s}_j = n^{-1}\sum_{i=1}^{n}(X_i - x_0)^j K\left(\frac{X_i - x_0}{h_n}\right); \quad j = 0, 1, 2.$

Although this class of estimators have favourable asymptotic properties and boundary behaviour compared to traditional kernel regression estimators, both estimators also share some similarities.

When this local fitting process is performed at each point of X varying in an appropriate estimation domain, a solid curve results. If K_{h_n} is a kernel function scaled by a bandwidth h_n then, for estimation at a particular point X_i, the weight assigned to a particular point Y_i is $K_{h_n}(X_i - X)$.

An issue in local polynomial fitting is the choice an order of the local polynomial. Since the modelling bias is primarily controlled by the bandwidth the issue is less crucial however. For a given bandwidth h_n a large value of p would expectedly reduce the modelling bias but on the other hand also cause higher variance by introducing more parameters at a considerable computational cost.

2.4 Nadaraya–Watson Kernel Regression Estimator

One of the most popular nonparametric regression smoothers, Nadaraya–Watson kernel regression estimator, was developed based on (2.6) independently by Nadaraya (1964) and Watson (1964). As we mentioned in Section 2.3 Nadaraya–Watson estimator can be obtained from local constant fitting that is $\hat{m}_{h_n,NW}(\cdot) = \hat{m}(\cdot, 0, h_n)$ although this was not the original motivation for the method. Note that throughout this section $\hat{m}_{h_n}(\cdot) = \hat{m}_{h_n,NW}(\cdot)$.

Let us adopt the kernel density estimators as one of the simplest methods of estimating $f(x, y)$ and $f(x)$ in (2.6). This is the basic idea of Nadaraya–Watson estimator. Corresponding kernel densities are $\hat{f}(x, y)$ and $\hat{f}(x)$,

$$\hat{f}(x,y) = \frac{1}{nh_{x,n}h_{y,n}} \sum_{i=1}^{n} K_x\left(\frac{x - X_i}{h_{x,n}}\right) K_y\left(\frac{y - Y_i}{h_{y,n}}\right) \tag{2.20}$$

$$\hat{f}(x) = \frac{1}{nh_{x,n}} \sum_{i=1}^{n} K_x\left(\frac{x - X_i}{h_{x,n}}\right) \tag{2.21}$$

where $K_x(\cdot)$ and $K_y(\cdot)$ are kernel functions.

The substitution of (2.20) and (2.21) into (2.6) results in

$$\begin{aligned}
\hat{m}(x) = \hat{m}_{h_n}(x) &= \frac{\int_y y\hat{f}(y,x)dy}{\hat{f}(x)} \\
&= \frac{\int_y \frac{1}{nh_{x,n}h_{y,n}} \sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right) K_y\left(\frac{y-Y_i}{h_{y,n}}\right)}{\frac{1}{nh_{x,n}} \sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right)} ydy \\
&= \frac{(h_{y,n})^{-1} \sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right) \int_y yK_y\left(\frac{y-Y_i}{h_{y,n}}\right) dy}{\sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right)}. \qquad (2.22)
\end{aligned}$$

If we set $s = (y - Y_i)/h_{y,n}$ then $K_y = K_s$ (previously we use y values now we use s values instead) and

$$\begin{aligned}
\frac{1}{h_{n,y}} \int_y yK_y\left(\frac{y-Y_i}{h_{y,n}}\right) dy &= \int_s (sh_{n,y} + Y_i)\, K_s(s)ds \\
&= Y_i \qquad (2.23)
\end{aligned}$$

because $\int_s K_s ds = 1$ and $\int_s sK_s ds = 0$ from the properties of kernel function as mentioned in Section 2.1.

Substitution of (2.23) in to (2.22) results in

$$\hat{m}_{h_n}(x) = \frac{\sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right) Y_i}{\sum_{i=1}^n K_x\left(\frac{x-X_i}{h_{x,n}}\right)}. \qquad (2.24)$$

Note that we used the notations $K_x(\cdot)$, $K_y(\cdot)$, $h_{n,x}$ and $h_{n,y}$ in the earlier part of this section simply to show that the kernel function and bandwidth for X and Y variables are not necessary to be the same. As we do not have both of them anymore we simply use the notation $K(\cdot)$ instead of $K_x(\cdot)$ and similarly h_n to $h_{x,n}$. Hence from (2.24), we have

$$\begin{aligned}
\hat{m}_{h_n}(x) &= \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{h_n}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right)} \\
&= \sum_{i=1}^n w_i(x) Y_i \qquad (2.25)
\end{aligned}$$

where

$$w_i(x) = \frac{K\left(\frac{x-X_i}{h_n}\right)}{\sum_{j=1}^n K\left(\frac{x-X_j}{h_n}\right)}. \tag{2.26}$$

Thus, at a given point x_0 the estimate of $\mathbf{E}[Y|X = x_0]$ is a weighted average of the Y_i values near x_0. $\hat{m}_{h_n}(x)$ defined by (2.24) is called the Nadaraya–Watson estimator. Since the estimate is written as (2.25), the Nadaraya–Watson estimator is a linear estimator and a weighted average of $\{Y_i\}_{i=1}^n$. As for kernel density estimation, the bandwidth h_n determines the level of smoothness of the estimate. Decreasing the bandwidth leads to a less smooth estimate. By re-writing (2.24) as follows:

$$\hat{m}_{h_n}(x) = \frac{(nh_n)^{-1}\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right) Y_i}{\hat{f}(x)} \tag{2.27}$$

and

$$w_i(x) = \frac{K\left(\frac{x-X_i}{h_n}\right)}{\hat{f}(x)} \tag{2.28}$$

where

$$\hat{f}(x) = \frac{1}{nh_n}\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right),$$

we can notice that

- Weights (2.28) depend on the observations from predictor variable $\{X_i\}_{i=1}^n$ through $\hat{f}(x)$ estimated by the kernel density.
- At an observation X_i, $m(X_i) \to Y_i$ as $h_n \to 0$.
- From (2.26) the weighting function $w_i(x) \to \frac{1}{n}$ for all x as $h_n \to \infty$ because $w_i(x) \to \frac{K(0)}{nK(0)}$ as $h_n \to \infty$. Hence, $m(x) \to \bar{Y}$.

2.4.1 Properties of Nadaraya–Watson Estimator

Since the numerator and denominator of Nadaraya–Watson estimator are both random variables the analysis for the numerator and denominator is done separately as in Hardle (1991). Define

$$r(x) = \int_y y f(x,y) dy$$

and from (2.6)

$$r(x) = m(x) f(x). \tag{2.29}$$

Now from (2.27)

$$\begin{aligned}\hat{r}_{h_n}(x) &= (nh_n)^{-1} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right) Y_i \\ &= n^{-1} \sum_{i=1}^{n} K_{h_n}(x - X_i) Y_i \end{aligned} \tag{2.30}$$

where $K_{h_n}(x - X_i) = (h_n)^{-1} K\left(\frac{x - X_i}{h_n}\right)$.

The regression curve estimate is thus given by

$$\hat{m}_{h_n}(x) = \frac{\hat{r}_{h_n}(x)}{\hat{f}_{h_n}(x)}. \tag{2.31}$$

Let us start with checking the asymptotic unbiasedness of $\hat{f}_{h_n}(x)$. Since X_is are identically distributed

$$\begin{aligned}\mathbf{E}[\hat{f}_{h_n}(x_0)] &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{E}[K_{h_n}(x_0 - X_i)] \\ &= \mathbf{E}[K_{h_n}(x_0 - X)] \\ &= \int_x K_{h_n}(x_0 - x) f(x) dx \\ &= \int_s K(s) f(x_0 - s h_n) ds \end{aligned} \tag{2.32}$$

where $s = (x_0 - x)/h_n$.

If we let $h_n \to 0$, we can see that

$$\mathbf{E}[\hat{f}_{h_n}(x_0)] \to f(x_0) \int_s K(s) ds = f(x_0). \tag{2.33}$$

As bandwidth h_n converges to 0, $\hat{f}_{h_n}(x)$ is asymptotically unbiased for $f(x)$. Unlike in parametric models there do not exist unbiased estimates for nonparametric models. This property is summarized in the following Theorem 2 (Collomb, 1976):

Theorem 2 *Let $\mathcal{D}$ be the set of distributions that are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^2$ with continuous density $f_{X,Y}(x,y)$ and such that $\forall\, x \in \mathbb{R}$, $\int_y f_{X,Y}(x,y)dy \neq 0$, then there is no unbiased estimate of the regression function $m(x)$. (In the sense that there is no estimate $\hat{m}_{h_n}(\cdot)$ of $m(\cdot)$ such that, for almost all x in $\mathbb{R}$ and all $f_{X,Y}(x,y)$ in $\mathcal{D}$, $\boldsymbol{E}[\hat{m}_{h_n}(x)] = m(x)$)*

Later we show that the kernel estimate is asymptotically unbiased at some fixed point x_0. Taylor series expansion of $f(x+sh_n)$ can be used to analyse the bias of Nadaraya–Watson estimate at a fixed point x_0 and might be able to see how the bias depends on the regularity of the regression function $m(x)$. For the denominator $\hat{f}_{h_n}(x)$ we compute expectation and variance as follows (complete proof is given in Prazen, 1962):

$$
\begin{aligned}
Bias[\hat{f}_{h_n}(x)] &= \int_s K(s) f(x - sh_n) ds - f(x) \\
&= \int_s K(s) \left[f(x) + sh_n f'(x) + \frac{h_n^2 s^2}{2} f''(x) + o(h_n^2) \right] ds - f(x) \\
&= f(x) \int_s K(s) ds - h_n f'(x) \int_s sK(s) ds + \frac{h_n^2}{2} f''(x) \int_s s^2 K(s) ds \\
&\qquad + o(h_n^2) - f(x).
\end{aligned} \tag{2.34}
$$

According to standard assumptions regarding the kernel density $K(\cdot)$ mentioned in Section 2.1, $\int_s K(s)ds = 1$ and $\int_s sK(s)ds = 0$. Hence, the bias given in (2.34) reduces to

$$
Bias[\hat{f}_{h_n}(x)] = \frac{h_n^2}{2} f''(x) \mu_2(K) + o(h_n^2) \tag{2.35}
$$

where $\mu_l(K) = \int_u u^l K(u)du$. Note that $\hat{f}_{h_n}(x)$ is asymptotically unbiased because the bias of $\hat{f}_{h_n}(x)$ is of order h_n^2.

Since X_i's are independent and identically distributed (i.i.d.),

$$\begin{aligned}
\mathbf{Var}[\hat{f}_{h_n}(x)] &= n^{-2}\mathbf{Var}\left[\sum_{i=1}^{n} K_{h_n}(x - X_i)\right] \\
&= n^{-2}\sum_{i=1}^{n}\mathbf{Var}[K_{h_n}(x - X_i)] \\
&= n^{-1}\mathbf{Var}[K_{h_n}(x - X)] \\
&= n^{-1}\{\mathbf{E}[K_{h_n}^2(x - X)] - (\mathbf{E}[K_{h_n}(x - X)])^2\} \\
&= n^{-1}\left\{h_n^{-2}\int_u K^2\left(\frac{x-u}{h_n}\right) f(u)du - (f(x) + o(h_n))^2\right\} \\
&= n^{-1}\left\{h_n^{-1}\int_s K^2(s) f(x - sh_n)ds - (f(x) + o(h_n))^2\right\} \\
&= (nh_n)^{-1} \parallel K \parallel_2^2 \{f(x) + o(1)\} - n^{-1}\left\{(f(x) + o(h_n))^2\right\},
\end{aligned} \tag{2.36}$$

where $\parallel K \parallel_2^2 = \int_s K^2(s)ds$. Here we use $\mathbf{E}[K_{h_n}(x - X)] = f(x) + o(h_n)$ from (2.35) and Taylor series expansion which leads to

$$\int_s K^2(s)f(x + sh_n)ds = \int_s K^2(s)ds(f(x) + o(1)) = \parallel K \parallel_2^2 (f(x) + o(1)).$$

The variance of $\hat{f}_{h_n}(x)$ is thus given by

$$\mathbf{Var}[\hat{f}_{h_n}(x)] = (nh_n)^{-1} \parallel K \parallel_2^2 f(x) + o((nh_n)^{-1}), \quad n \to \infty. \tag{2.37}$$

Now we analyse the same properties of the numerator $\hat{r}_{h_n}(x)$.

$$\begin{aligned}
\mathbf{E}[\hat{r}_{h_n}(x)] &= \mathbf{E}[n^{-1}\sum_{i=1}^{n} K_{h_n}(x - X_i)Y_i] \\
&= \mathbf{E}[K_{h_n}(x - X)Y] \\
&= \int_u \int_y y K_{h_n}(x - u) f(y|u) f(u) dy du \\
&= \int_u K_{h_n}(x - u) f(u) \left(\int_y y f(y|u) dy \right) du \\
&= \int_u K_{h_n}(x - u) f(u)(\mathbf{E}[Y|X = u]) du \\
&= \int_u K_{h_n}(x - u) f(u) m(u) du \\
&= \int_u K_{h_n}(x - u) r(u) du. \qquad (2.38)
\end{aligned}$$

Similar to $\hat{f}_{h_n}(x)$, by expanding the kernel, we note that

$$\mathbf{E}[\hat{r}_{h_n}(x)] = r(x) + \frac{h_n^2}{2} r''(x)\mu_2(K) + o(h_n^2), \quad h_n \to 0. \qquad (2.39)$$

Hence, $\hat{r}_{h_n}(x)$ is an asymptotically unbiased for $r(x)$ as $h_n \to 0$. To compute the variance of $\hat{r}_{h_n}(x)$, we let $S^2(x) = \mathbf{E}[Y^2|X = x]$. Thus,

$$\begin{aligned}
\mathbf{Var}[\hat{r}_{h_n}(x)] &= \mathbf{Var}\left[n^{-1} \sum_{i=1}^{n} K_{h_n}(x - X_i)Y_i \right] \\
&= n^{-1}\mathbf{Var}[K_{h_n}(x - X)Y] \\
&= n^{-1} \left\{ \int_u K_{h_n}^2(x - u) S^2(u) f(u) du - \left(\int_u K_{h_n}(x - u) r(u) du \right)^2 \right\} \\
&= (nh_n)^{-1} \int_u K^2(u) S^2(x + uh_n) f(x + uh_n) du + o((nh_n)^{-1}) \\
&= n^{-1} h_n^{-1} f(x) S^2(x) \parallel K \parallel_2^2 + o((nh_n)^{-1}); \quad nh_n \to \infty. \qquad (2.40)
\end{aligned}$$

Details on these computations are rather long and tedious and can be found in Collomb (1976). By combining (2.39) and (2.40) the formula for mean squared error (MSE) of $\hat{r}_{h_n}(x)$ is

$$\begin{aligned} MSE[\hat{r}_{h_n}(x)] &= \mathbf{Var}[\hat{r}_{h_n}(x)] + [\mathbf{E}(\hat{r}_{h_n}(x) - r(x))]^2 \\ &= \frac{1}{nh_n} f(x)s^2(x)\|K\|_2^2 + \frac{h_n^4}{4}\left(r''(x)\mu_2(K)\right)^2 + o(h_n^4) + o((nh_n)^{-1}). \end{aligned} \tag{2.41}$$

Theorem 3 *Let $K(\cdot)$ satisfy $\int_t |K(t)|dt \leq \infty$ and $\lim_{|t|\to\infty} tK(t) = 0$. Suppose also that $m(x)$ and $f(x)$ are continuous at $x = x_0$ and $f(x) > 0$. Then provided $h_n \to 0$ and $nh_n \to \infty$ as $n \to \infty$ we have $\hat{m}_{h_n}(x) \xrightarrow{p} m(x)$.*

If we let $h_n \to 0$ such that $nh_n \to \infty$ then $MSE[\hat{r}_{h_n}(x)] \to 0$ and $\hat{r}_{h_n}(x) \xrightarrow{p} r(x)$ by Chebychev's inequality which is

$$\begin{aligned} \forall \quad \varepsilon > 0, \quad \Pr\left[|\hat{r}_{h_n}(x) - r(x)| \geq \varepsilon\right] &\leq \frac{\mathbf{E}\left[(\hat{r}_{h_n}(x) - r(x))\right]^2}{\varepsilon^2} \\ &= \frac{MSE\left[\hat{r}_{h_n}(x)\right]}{\varepsilon^2}. \end{aligned}$$

Thus, if $MSE[\hat{r}_{h_n}(x)] \to 0$ then $\hat{r}_{h_n}(x) \xrightarrow{p} r(x)$. Hence, $\hat{r}_{h_n}(x)$ is consistent. As the denominator of $\hat{m}_{h_n}(x)$, the kernel density estimate $\hat{f}_{h_n}(x)$, is consistent, using Slutzky's theorem,

$$\hat{m}_{h_n}(x) = \frac{\hat{r}_{h_n}(x)}{\hat{f}_{h_n}(x)} \xrightarrow{p} \frac{r(x)}{f(x)} = m(x) \quad \text{as } h_n \to 0,\ nh_n \to \infty. \tag{2.42}$$

As a consequence $\hat{m}_{h_n}(x)$ is a consistent estimate of the regression curve $m(x)$ if $h_n \to 0$, $nh_n \to \infty$. Conditions for consistency of $\hat{m}(x)$ have been proved in Schuster (1972).

The bias of the Nadaraya–Watson estimator $\hat{m}_{h_n,NW}(x)$ depends not only on $m''(x)$ but also on $m'(x)\frac{f'(x)}{f(x)}$ due to the local constant fit. Hence, keeping $m''(x)$ fixed it is observed that either when $|m'(x)|$ is large or when the highly clustered designed data where $\frac{f'(x)}{f(x)}$ is large, so is the bias of $\hat{m}_{h_n,NW}(\cdot)$. As a result, it is worthwhile to remark that $\hat{m}_{h_n,NW}(\cdot)$ can not adapt to highly clustered designs. Besides it is not good at testing linearity as the bias of the estimator is large in the case of linear regression $m(x) = a + bx$ with a large coefficient b. It is not surprising therefore to note that the performance of $\hat{m}_{h_n,NW}(\cdot)$ estimator worsens for larger values of explanatory variable X because $|m'(x)|$ turn out to be larger. Also the situation becomes worst for large $|x|$ as then $\frac{f'(x)}{f(x)}$ increases. However, $\hat{m}_{h_n,NW}(\cdot)$ has the same asymptotic properties as the local linear estimator $\hat{m}_{h_n,LL}(\cdot)$ in the case of fixed equidistant design. Because in the fixed design $f(x) = 1$ hence $f'(x) = 0$ and the bias term $m'(x)\frac{f'(x)}{f(x)}$ equal to zero. If more design points are in the interval $(x, x+h_n)$ than in $(x-h_n, x)$ then the local average will be biased as the average will include more responses over $(x, x+h_n)$ than over $(x-h_n, x)$. The bias will be positive if $m'(x) > 0$ and negative otherwise.

2.5 Local Linear Kernel Regression

Local linear regression estimator $\hat{m}_{h_n,LL}(x) = \hat{m}(x; 1, h_n)$ has a very good reputation as an estimator which overcomes the disadvantages of other kernel type regression estimators. It is introduced as a design adaptive regression estimator based on a weighted local linear regression. Suppose that $m''(x)$ exists. In a small neighbourhood of a point x, $m(y) \approx m(x) + m'(x)(y-x) \equiv \beta_0 + \beta_1(y-x)$. Now the problem of estimating $m(x)$ corresponds to a local linear regression problem of estimating the intercept β_0 which is equivalent minimizing

$$\sum_{i=1}^{n} \left[(Y_i - \beta_0 - \beta_1(X_i - x))^2 \, h_n^{-1} K\left(\frac{X_i - x}{h_n}\right) \right]. \tag{2.43}$$

The solution of $\hat{\beta}_0$ to the weighted least squares problem (2.43) is defined as the local linear regression smoother $\hat{m}_{h_n,LL}(x)$ as in (2.19). It is obvious that

Taylor's theorem is the foundation of the well known kernel type regression estimators $\hat{m}_{h_n,NW}(\cdot)$ and $\hat{m}_{h_n,LL}(\cdot)$.

The following assumptions are made in the rest of analysis:
(i) The function $m''(x)$ is continuous on $x \in [0, 1]$.
(ii) The kernel $K(x)$ is symmetric about zero and is supported on $x \in [-1, 1]$.
(iii) The bandwidth h_n is a sequence satisfying $h_n \to 0$ and $nh_n \to \infty$ as $n \to \infty$.
(iv) The point x_0 at which the estimation is taking place satisfies $h_n < x_0 < 1 - h_n \Rightarrow h_n < \frac{1}{2}$ for $\forall\ x_0$. In other words, the given point x_0 is a point in the interior of the design which is more than a bandwidth h_n away from the boundary for all sufficiently large n.

2.5.1 Asymptotic MSE Approximations for Fixed Equidistant Design Regression Model

Here we consider equally spaced design points of the form $x_i = \frac{i}{n}; \quad i = 1, \ldots, n$ hence $f(x) = 1$. From (2.17):

$$\mathbf{E}\left[\hat{m}(x; 1, h_n)\right] = \mathbf{E}\left[\mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{\mathbf{-1}}\mathbf{X^TWY}\right] \tag{2.44}$$

where $\mathbf{X} = \begin{bmatrix} 1 & X_1 - x \\ \vdots & \vdots \\ 1 & X_n - x \end{bmatrix}_{n\times 2}$ as we consider local linear fitting which equivalent to $p = 1$.

Since $\mathbf{E}[\mathbf{Y}] = \mathbf{E}[\mathbf{M}(\mathbf{x}) + \varepsilon] = \mathbf{M}(\mathbf{x})$, (2.44) can be rewritten as

$$\mathbf{E}\left[\hat{m}(x; 1, h_n)\right] = \mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{\mathbf{-1}}\mathbf{X^TWM}(\mathbf{x}) \tag{2.45}$$

where $\mathbf{M}(\mathbf{x}) = \left[m(x_1), \ldots, m(x_n)\right]^T$.

From Taylor's theorem for any $x \in [0,1]$,

$$m(x_i) = m(x) + (x_i - x)m'(x) + \frac{1}{2}(x_i - x)^2 m''(x) + \dots$$

which entail that $\mathbf{M(x)} = \mathbf{X}\begin{bmatrix} m(x) \\ m'(x) \end{bmatrix} + \frac{1}{2}m''(x)\begin{bmatrix} (X_1 - x)^2 \\ \vdots \\ (X_n - x)^2 \end{bmatrix} + \dots.$

Hence, $\mathbf{E}\left[\hat{m}(x;1,h_n)\right]$ given by (2.45) can be expanded as

$$\mathbf{E}\left[\hat{m}(x;1,h_n)\right] = \mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{-1}\mathbf{X^TWX}\begin{bmatrix} m(x) \\ m'(x) \end{bmatrix} +$$

$$\frac{1}{2}m''(x)\mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{-1}\mathbf{X^TW}\begin{bmatrix} (x_1 - x)^2 \\ \vdots \\ (x_n - x)^2 \end{bmatrix} + \dots$$

which leads to

$$\mathbf{E}\left[\hat{m}(x;1,h_n)\right] = m(x) + \tfrac{1}{2}m''(x)\mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{-1}\mathbf{X^TW}\begin{bmatrix} (x_1 - x)^2 \\ \vdots \\ (x_n - x)^2 \end{bmatrix} + \dots.$$

Note that if the unknown function $m(x)$ is in linear form then r^{th} derivative of $m(x)$, is 0 that is $m^{(r)}(x) = 0 \quad \forall \quad r \geq 2$. Therefore local linear smoother has one of the desirable property, unbiasedness only for linear regression functions.

The bias of $\hat{m}(x;1,h_n)$ is

$$\mathbf{E}\left[\hat{m}(x;1,h_n)\right] - m(x) = \frac{1}{2}m''(x)\mathbf{e_1^T}\left(\mathbf{X^TWX}\right)^{-1}\mathbf{X^TW}\begin{bmatrix} (x_1 - x)^2 \\ \vdots \\ (x_n - x)^2 \end{bmatrix} + \dots. \tag{2.46}$$

Observe that

$$n^{-1}\mathbf{X^TWX} = n^{-1}\begin{bmatrix} \sum_{i=1}^n w_i & \sum_{i=1}^n w_i(x_i-x) \\ \sum_{i=1}^n w_i(x_i-x) & \sum_{i=1}^n w_i(x_i-x)^2 \end{bmatrix} \tag{2.47}$$

and

$$n^{-1}\mathbf{X^TW}\begin{bmatrix} (x_1-x)^2 \\ \vdots \\ (x_n-x)^2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n w_i(x_i-x)^2 \\ \sum_{i=1}^n w_i(x_i-x)^3 \end{bmatrix} \tag{2.48}$$

where $w_i = \frac{1}{h_n}K\left(\frac{x_i-x}{h_n}\right)$.

To compute leading bias term in (2.46) for nonlinear $m(x)$, we let $s_j(x;h_n) = n^{-1}\sum_{i=1}^n w_i(x_i-x)^j = (nh_n)^{-1}\sum_{i=1}^n K\left(\frac{x_i-x}{h_n}\right)(x_i-x)^j$; $j=0,1,2,\ldots$. Hence,

$$n^{-1}\mathbf{X^TWX} = \begin{bmatrix} s_0(x;h_n) & s_1(x;h_n) \\ s_1(x;h_n) & s_2(x;h_n) \end{bmatrix} \tag{2.49}$$

and

$$n^{-1}\mathbf{X^TW}\begin{bmatrix} (x_1-x)^2 \\ \vdots \\ (x_n-x)^2 \end{bmatrix} = \begin{bmatrix} s_2(x;h_n) \\ s_3(x;h_n) \end{bmatrix}. \tag{2.50}$$

Now using the property of bounded $K'(\cdot)$ and the assumptions (ii)-(iv) for n sufficiently large, $s_j(x;h_n)$ can be approximated by integrals as given below.

$$\begin{aligned} s_j(x;h_n) &= (nh_n)^{-1}\sum_{i=1}^n K\left(\frac{x_i-x}{h_n}\right)(x_i-x)^j \\ &= \int_0^1 (y-x)^j \frac{1}{h_n}K\left(\frac{y-x}{h_n}\right)dy + O(n^{-1}) \\ &= h_n^j\int_{\frac{-x}{h_n}}^{\frac{(1-x)}{h_n}} t^jK(t)dt + O(n^{-1}) \\ &= h_n^j\int_{-1}^{1} t^jK(t)dt + O(n^{-1}). \end{aligned} \tag{2.51}$$

Obviously when j is an odd number, $s_j(x;h_n) = 0$ due to the symmetry and compact support of the kernel $K(\cdot)$. The bias terms in (2.49) and (2.50) therefore turn out to be

$$n^{-1}\mathbf{X^T W X} = \begin{bmatrix} 1 + O(n^{-1}) & O(n^{-1}) \\ O(n^{-1}) & h_n^2 \mu_2(K) + O(n^{-1}) \end{bmatrix} \tag{2.52}$$

and

$$n^{-1}\mathbf{X^T W} \begin{bmatrix} (x_1 - x)^2 \\ \vdots \\ (x_n - x)^2 \end{bmatrix} = \begin{bmatrix} h_n^2 \mu_2(K) + O(n^{-1}) \\ O(n^{-1}) \end{bmatrix} \tag{2.53}$$

where $\mu_j(K) = \int_t t^j K(t)dt$.

Substituting (2.52) and (2.53) to (2.46) and performing some simple matrix algebra, the bias approximation of local linear smoother leads to

$$\mathbf{E}\left[\hat{m}_{h_n,LL}(x)\right] = \mathbf{E}\left[\hat{m}(x;1,h_n)\right] = m(x) + \frac{1}{2}h_n^2 m''(x)\mu_2(K) + o(h_n^2) + O(n^{-1}). \tag{2.54}$$

The variance approximation of local linear smoother takes the form

$$\mathbf{Var}\left[\hat{m}(x;1,h_n)\right] = \mathbf{e_1^T}\left(\mathbf{X^T W X}\right)^{\mathbf{-1}} \mathbf{X^T W V W X} \left(\mathbf{X^T W X}\right)^{\mathbf{-1}} \mathbf{e_1} \tag{2.55}$$

where $\mathbf{V} = diag\{\sigma^2, \ldots, \sigma^2\}$.

Following the similar steps those used in bias calculation, Wand and Jones (1995) have shown that,

$$\begin{aligned} n^{-1}\mathbf{X^T W V W X} &= n^{-1}\sum_{i=1}^{n} K_{h_n}(x_i - x)^2\sigma^2 \begin{bmatrix} 1 & (x_i - x) \\ (x_i - x) & (x_i - x)^2 \end{bmatrix} \\ &= \begin{bmatrix} (h_n)^{-1}R(K)\sigma^2 + (h_n^{-1}) & O(n^{-1}) \\ O(n^{-1}) & h_n \mu_2(K^2)\sigma^2 + O(n^{-1}) \end{bmatrix} \end{aligned} \tag{2.56}$$

where $R(K) = \int_t K(t)^2 dt$.

Substituting the results given in (2.56) and (2.52) to (2.55):

$$\mathbf{Var}\left[\hat{m}_{h_n,LL}(x)\right] = \mathbf{Var}\left[\hat{m}(x;1,h_n)\right] = (nh_n)^{-1}R(K)\sigma^2 + o\{(nh_n)^{-1}\}. \quad (2.57)$$

(Wand and Jones, 1995).

2.5.2 Asymptotic MSE Approximations for Random Design Regression Model

Consider the random design regression model which is a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of random pairs and $m(x) = \mathbf{E}(Y|X = x)$. Suppose that the design is an independent sample, denoted by $X_1, \ldots, X_n$, having a density function $f(x)$. In addition to the previously mentioned assumptions (i)-(iv) for simplicity, here we assume $f(x)$ has support on $[0, 1]$ and $f'(x)$ is continuous. Provided we condition on the predictor variables, the bias and variance calculations in the random design model can be done similar to the fixed equidistant design case.

It follows directly from (2.45) that

$$\mathbf{E}\left[\hat{m}(x;1,h_n)|X_1,\ldots,X_n\right] = \mathbf{e}^T(\mathbf{X}^\mathbf{T}\mathbf{W}\mathbf{X})^{-1}\mathbf{X}\mathbf{W}\mathbf{M}.$$

As we refer to the case of local linear fitting, $\mathbf{X} = \begin{bmatrix} 1 & X_1 - x \\ \vdots & \vdots \\ 1 & X_n - x \end{bmatrix}_{n\times 2}$

and analogous to the fixed design setting

$$m(X_i) = m(x) + (X_i - x)m'(x) + \frac{1}{2}(X_i - x)^2 m''(x) + \ldots .$$

Using the same arguments as in Section 2.5.1 the conditional bias of

$\hat{m}(x;1,h_n)$ is defined as

$$\mathbf{E}\left[\hat{m}(x;1,h_n)-m(x)|X_1,\ldots,X_n\right]=\frac{1}{2}m''(x)\mathbf{e}_1^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\begin{bmatrix}(X_1-x)^2\\ \vdots\\ (X_n-x)^2\end{bmatrix}+\ldots. \quad (2.58)$$

Yet again if $m(x)$ is a linear function then $\hat{m}(x;1,h_n)$ is conditionally unbiased given $X_1,\ldots,X_n$ because $m^r(x)=0; \quad \forall \quad r\geq 2$. From (2.46), (2.47) and (2.48) the conditional bias of $\hat{m}(x;1,h_n)$ is

$$\mathbf{E}\left[\hat{m}(x;1,h_n)-m(x)|X_1,\ldots,X_n\right]=\frac{1}{2}m''(x)\mathbf{e}_1^T\begin{bmatrix}s_0(x;h_n) & s_1(x;h_n)\\ s_1(x;h_n) & s_2(x;h_n)\end{bmatrix}^{-1}\begin{bmatrix}s_2(x;h_n)\\ s_3(x;h_n)\end{bmatrix}+\ldots \quad (2.59)$$

Approximation of leading bias terms in (2.59) does not directly follow from (2.51) as they were calculate for fixed equally spaced design data where a strong condition was imposed that the probability density function of design points is $f(x)=1$.

Here we use expansion of $f(x-ht)$ for $t\in(0,1)$ in a Taylor series about x;

$$f(x-h_nt)=f(x)-h_ntf'(x)+\frac{1}{2}h_n^2t^2f''(x)+o(h_n^2) \quad (2.60)$$

and for random variable X having density $f(x)$

$$\mathbf{E}[\hat{f}(x)]=\mathbf{E}\left[\frac{1}{h_n}K\left(\frac{x-X}{h_n}\right)\right]=\frac{1}{h_n}\int_y K\left(\frac{x-y}{h_n}\right)f(y)dy \quad (2.61)$$

to approximate leading bias terms in (2.59).

Now using (2.60), (2.61) plus the property of bounded K' and the assumptions (ii)-(iv) for n sufficiently large, $s_j(x;h_n)$ can be expanded by integral as given below.

$$\begin{aligned} s_j(x;h_n) &= \int_0^1 (y-x)^j \frac{1}{h_n} K\left(\frac{y-x}{h_n}\right) f(y)dy \\ &= h_n^j \int_{\frac{-x}{h_n}}^{\frac{(1-x)}{h_n}} t^j K(t) f(x+th_n)dt \\ &= h_n^j \int_{\frac{-x}{h_n}}^{\frac{(1-x)}{h_n}} \left(f(x) + h_n t f'(x) + \frac{1}{2} h_n^2 t^2 f''(x) + o(h_n^2)\right) t^j K(t)dt. \end{aligned} \tag{2.62}$$

From (2.62) follows that $s_j(x;h_n)$ equivalent to

$$s_j(x;h_n) = \begin{cases} h_n^{j+1}\mu_{j+1}(K)f'(x) + o_p(h_n^{j+1}) & j \quad \text{is} \quad \text{odd} \\ h_n^j \mu_j(K) f(x) + o_p(h_n^j) & j \quad \text{is} \quad \text{even.} \end{cases} \tag{2.63}$$

Note that for two real-valued random sequences A_n and B_n, $A_n = o_P(B_n)$ if for all $\varepsilon > 0$, $\lim_{n\to\infty} \; Pr(|A_n/B_n| > \varepsilon) = 0$.

When j is an odd number, $j+2$ is also an odd number. Hence, from the standard properties of kernel as mentioned in Section 2.1, $\int_t t^j K(t)dt$ vanishes as $\int_t t^j K(t)dt = 0$ and $\int_t t^{j+2} K(t)dt = 0$. Similarly, even value of j, $\int_t t^{j+1} K(t)dt = 0$ as $j+1$ becomes an odd number so the corresponding term vanishes.

Above (2.63) leads to

$$n^{-1}\mathbf{X}^T\mathbf{W}\mathbf{X} = \begin{bmatrix} f(x) + o_p(1) & h_n^2 f'(x)\mu_2(K) + o_p(h_n^2) \\ h_n^2 f'(x)\mu_2(K) + o_p(h_n^2) & h_n^2 f(x)\mu_2(K) + o_p(h_n^2) \end{bmatrix} \tag{2.64}$$

and

$$n^{-1}\mathbf{X}^T\mathbf{W} \begin{bmatrix} (X_1 - x)^2 \\ \vdots \\ (X_n - x)^2 \end{bmatrix} = \begin{bmatrix} h_n^2 f(x)\mu_2(K) + o_p(h_n^2) \\ h_n^4 f'(x)\mu_4(K) + o_p(h_n^4) \end{bmatrix}. \tag{2.65}$$

Notice that

$$\left(n^{-1}\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1} = \begin{bmatrix} f(x)^{-1} + o_p(1) & -f'(x)/f(x)^2 + o_p(1) \\ -f'(x)/f(x)^2 + o_p(1) & \{h_n^2 f(x)\mu_2(K)\}^{-1} + o_p(h_n^{-2}) \end{bmatrix}.$$

Some simple matrix algebra then leads to the expression of the conditional bias

$$\mathbf{E}\left[\hat{m}(x;1,h_n) - m(x)|X_1,\ldots,X_n\right] = \frac{1}{2}h_n^2 m''(x)\mu_2(K) + o_p(h_n^2). \qquad (2.66)$$

(Wand and Jones, 1995.)

Using approximations similar to those used above, the conditional variance is obtained as

$$\mathbf{Var}\left[\hat{m}(x;1,h_n) - m(x)|X_1,\ldots,X_n\right] = \frac{R(K)}{nh_n f(x)}\sigma^2 + o_p\{(nh_n)\}^{-1} \qquad (2.67)$$

where $R(K) = \int_u K(u)^2 du$.

(Fan, 1992a.)

The conditional MSE and weighted MISE of the local linear smoother as given in Fan (1992) are as follows:

$$\mathbf{E}[(\hat{m}(x;1,h_n) - m(x))^2|X_1,\ldots,X_n] = \frac{1}{4}(m''(x)\mu_2(K))^2 h_n^4 + \frac{R(K)\sigma^2}{nh_n f(x)} + o_p\left(h_n^4 + \frac{1}{nh_n}\right) \qquad (2.68)$$

and

$$\mathbf{E}[(\hat{m}(x;1,h_n) - m(x))^2 w(x)|X_1,\ldots,X_n] = \frac{(\mu_2(K))^2}{4}\int_x m''(x)^2 w(x)dx h_n^4 + \frac{R(K)}{nh_n}\int_x \frac{\sigma^2}{f(x)} w(x)dx + o_p\left(h_n^4 + \frac{1}{nh_n}\right). \qquad (2.69)$$

The mean squared error splits up into two parts, squared of the bias and the variance. According to (2.68) and (2.69) the bias is a increasing function of h_n conversely the variance is decreasing function of h_n. Thus, smoothing problem is about balancing the variance versus the squared bias.

Remarks

The leading bias term depends on x only through $m''(x)$ which reflects the error of linear approximation. If $m(\cdot)$ is close to being linear at x then $m''(x)$ is relatively small, results in less bias in this case. Whereas if $m(\cdot)$ has a high amount of curvature at x then $m''(x)$ is higher and more biased estimates are produced. The bias is increased with more smoothing since it also depends on h_n.

2.6 Other Kernel-Based Regression Estimates

Other versions of kernel type regression estimates have been introduced in the literature. Here we just give definitions for those estimates that are of primary importance. Priestley and Chao (1972) considered the problem of estimating an unknown regression function $m(x)$ given for observations at a fixed set of points by

$$\hat{m}_{PC}(x) = \sum_{i=2}^{n} (x_i - x_{i-1}) h_n^{-1} K\left(\frac{x - x_i}{h_n}\right) Y_i. \tag{2.70}$$

Gasser and Müller (1979) estimator is given for x restricted to some interval $[a, b]$, by

$$\hat{m}_{GM}(x) = \sum_{i=1}^{n} \int_{s_i - 1}^{s_i} h_n^{-1} K\left(\frac{x - x_i}{h_n}\right) dx\, Y_i \tag{2.71}$$

where $s_i = (x_i + x_{i+1})/2;\ i = 1, \ldots, n-1;\ s_0 = a;\ s_n = b.$

There are several important issues which have to be discussed. First of all there is a choice of the bandwidth parameter h_n which plays a rather crucial role. A too large bandwidth over smooth nonparametric fit of unknown regression function while too small bandwidth under smooth the nonparametric fit of unknown function. Ideal theoretical choice is not always practically usable since it depends on unknown quantities. Finding a practical procedure for selecting bandwidth parameter is one of the most important tasks.

2.7 Bandwidth Selection

As we have already pointed out in the earlier part of this chapter, the bias in the estimation of $\hat{m}_{h_n}(\cdot)$ does not depend directly on the sample size n but does depend on the bandwidth h_n. Of course if h_n chosen as a function of n then the bias will depend directly on n through its dependance on h_n. In the following sections some popular methods for choosing the bandwidth are discussed. There is as yet no universally accepted approach to this problem.

The selection of appropriate values for bandwidth h_n is the most challenging aspect of nonparametric regression. Each nonparametric kernel technique involves selection of smoothing parameters. The accuracy of the estimator is far more sensitive to the value of h_n than it is to choice of kernel function $K(\cdot)$. Given the usual kernel function means observations close to x_i have more influence on the regression estimate at x_i than those farther away. Bandwidth controls the amount of relative influence. Small h_n results in local linear fitting process depending heavily on those observations that are closet to x_i and tends to yield a more wiggly estimate.

An effective approach is guided trial and error. If the fitted regression looks too rough, then try increasing the bandwidth; if it looks too smooth, then see if the bandwidth can be decreased without making the fit too rough. The smallest value of h_n that provides a smooth fit is required. A complementary visual approach is to find the residuals from the fit, $e_i = y_i - \hat{y}_i$, and to smooth the residuals against the predictor x. If the data have been over smoothed, then there will be a systematic relationship between the average residual and the predictor x. If the fit does not over smooth the data, then the average residual will be approximately 0 regardless the value of x. The largest value of h_n that yields residuals that are unrelated to the value of x which is the optimal bandwidth.

We can find many situations where the bandwidth is subjectively chosen by eye produce satisfactory results. In this case, we would look at several regression estimates for a given point over range of bandwidths and the estimate that is the most agreeable in some sense is being selected. One feasible approach is to begin with a large bandwidth and to shrink the amount of smoothing until fluctuations that are more random than structural starts to emerge. This approach is more viable when the estimates from the interpolation are convincingly close enough to the actual vales.

On the other hand, it is very beneficial to have the bandwidth automatically selected from the data. Mainly it can be very time consuming to select the bandwidth by eye if there are many estimates required for a given regression function. Besides, in extrapolation the user would not have any knowledge of which bandwidth gives an estimate closest to the true value. A method that uses the bivariate data (X_i, Y_i) to produce bandwidth $\hat{h}_n$ is called a bandwidth selector. The problem of bandwidth selection exists in all types of kernel regression estimation including the scatterplot smoothing.

Several data-driven methods have been developed. Cross-validation (Stone, 1974; Rudemo, 1982) and generalized cross-validation (Wahba, 1977) are generally applicable methods. Yet, their resulting bandwidths can vary substantially (Hall and Johnstone, 1991). An alternative method is plug-in method (Hall, Sheather, Jones and Marron 1992). For more details of these bandwidth selectors refer Wand and Jones (1995).

2.7.1 Plug-in Method

This approach addresses efficiency through the asymptotic mean squared error (AMSE) but attempts a direct estimate of the optimal bandwidth. Fan and Gijbles (1996) elaborated on the basic concept and elucidated the more sophisticated applications of the plug-in principle.

Here we are looking at a version of the simple direct plug-in idea for local linear regression $\hat{m}_{h_n,LL}(\cdot)$ that has been shown to possess attractive theoretical and practical properties (Ruppert, Sheather and Wand, 1995). For simplicity assume that the errors ε_i are homoscedastic with common variance σ^2 and that the X_i's are from a compactly supported density on $[0,1]$. An appropriate global error criterion is the weighted conditional mean integrated square error (MISE):

$$MISE\left\{\hat{m}(x;1,h_n)|X_1,\ldots,X_n\right\} = \\ \mathbf{E}\left[\int_x \{\hat{m}(x;1,h_n)-m(x)\}^2 f(x)\,dx|X_1,\ldots,X_n\right]. \qquad (2.72)$$

This weighting by $f(x)$ puts more emphasis on those regions where there are more data as well as simplifying the plug-in methodology. With respect to this criterion the asymptotically optimal bandwidth with respect to mean integrated squared error criteria is

$$h_{AMISE} = C_1(K)\left[\frac{\sigma^2}{\theta_{22}n}\right]^{1/5} \qquad (2.73)$$

where AMISE: asymptotic mean integrated squared error, $C_1(K) = \{R(K)/\mu_2(K)^2\}^{1/5}$ and θ_{22} is a special case of the notation $\theta_{rs} = \int_x m^{(r)}(x)m^{(s)}(x)f(x)\,dx$ where $m^{(j)}$ is j^{th} derivative of $m(x)$.

Plug-in bandwidth selectors are based on the simple idea of plug in estimates of unknown quantities that appear in formulae for the asymptotically optimal bandwidth. A natural estimator for θ_{22} is $\hat{\theta}_{22}(g_n) = n^{-1}\sum_{i=1}^{n}\hat{m}^{(2)}(X_i; 3, g_n)^2$ where $\hat{m}^{(2)}(X_i; 3, g_n)$ is the second derivative of the regression estimator $\hat{m}(X_i; 3, g_n)$ using third degree polynomial ($p = 3$) with bandwidth g_n. While a natural estimator for σ^2 is

$$\hat{\sigma}^2 = \nu^{-1}\sum_{i=1}^{n}\{Y_i - \hat{m}(X_i; 1, \lambda_n)\}^2 \tag{2.74}$$

where

$\nu = n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2,\ w_{ij} = \mathbf{e}_1^T\left(\mathbf{XWX}\right)^{-1}\mathbf{X}^T\mathbf{WX}$ and $\mathbf{W}$ is based on bandwidth λ_n.

Hence, direct plug-in rule (DPI) for selection h_n is of the form

$$\hat{h}_{n,DPI} = C_1(K)\left[\frac{\hat{\sigma}^2(\lambda_n)}{\hat{\theta}_{22}(g_n)n}\right]^{1/5}. \tag{2.75}$$

Substituting estimates of unknown quantities in (2.75) produces a variety of plug-in estimates which have worked well in local linear regression in some settings.

However, $\hat{h}_{n,DPI}$ depends on the choice of the pilot bandwidths g_n and λ_n as a result, this rule (2.75) is not fully automatic. Need to formulate rules for selection of the auxiliary bandwidths g_n and λ_n. In practice, these quantities have to be estimated on the basis of some preliminary smoothing process which raises a second-order bandwidth selection problem. There is a considerable uncertainty about how to choose bandwidths g_n and λ_n in that first step. On the other hand from a theoretical point of view it is always restricted to a certain smoothness class which is to twice differentiable regression function.

2.7.2 Cross-Validation Method

Cross-validation method (Stone, 1977) is a most commonly used bandwidth selection technique. Under this method h_n is chosen by minimizing

$$\sum_{i=1}^{n} \left(Y_i - \hat{m}_{-i,h_n}(X_i)\right)^2$$

where $\hat{m}_{-i,h_n}(X_i)$ is the regression estimator without using the i^{th} observation (X_i, Y_i). The key idea is to omit the i^{th} observation from the local regression at the focal value x_i. Omitting the i^{th} observation makes the fitted value $\hat{m}_{-i,h_n}(X_i)$ independent of the observed value Y_i which is denoted by $\hat{m}_{-i,h_n}(X_i)$ given $X_i = x_i$.

Define

$$CVal(h_n) = \frac{\sum_{i=1}^{n} \left(Y_i - \hat{m}_{-i,h_n}(X_i)\right)^2}{n} \tag{2.76}$$

where $\hat{m}_{-i,h_n}(X_i)$ is $\hat{m}_{-i,h_n}(X_i) \mid x_i$ for bandwidth h_n. The object is to find the value of h_n that minimizes $CVal(h_n)$.

The cross-validation function is a kind of estimate of the mean average squared error (MASE) at the observed x,

$$MASE(h_n) = \mathbf{E}\left[\frac{\sum_{i=1}^{n} \left(\hat{m}_{-i,h_n}(X_i) - m(X_i)\right)^2}{n}\right]. \tag{2.77}$$

Because of the independence of $\hat{m}_{-i,h_n}(X_i)$ and Y_i, the expectation of $CVal(h_n)$ is

$$\begin{aligned}\mathbf{E}[CVal(h_n)] &= \frac{\sum_{i=1}^{n} \mathbf{E}\left[Y_i - \hat{m}_{-i,h_n}(X_i)\right]^2}{n} \\ &\simeq MASE(h_n) + \sigma^2. \end{aligned} \tag{2.78}$$

The function $CVal(h_n)$ is commonly called a cross-validation function since it validates the ability to predict $\{Y_k\}_{k=1}^{n}$ across the subsamples $\{(X_i, Y_i)\}_{i \neq k}$

(Stone, 1974). This $CVal(h_n)$ function for finding bandwidth h_n in the context of kernel regression was proposed by Clark (1975). Although cross-validation is a useful method for selecting the bandwidth in kernel regression, it is only an estimate and is therefore subject to sampling variation. Particularly in small samples, this variability can be substantial.

2.7.3 Our Bandwidth Selection Method

Available bandwidth selectors can be generally divided into two groups. The first type of bandwidth selector consists of plain and straightforwardly assessable formulae which aim to find a bandwidth that is sensible for a wide range of situations but without any mathematical assurances of being close to optimal bandwidth. This type of bandwidth selectors are known as quick and simple. The quick and simple bandwidth selectors are more attractive as we need to have fast automatically generated kernel estimates for computer algorithms those require many regression estimation steps as well as providing a practical preliminary point for subjective choice of the smoothing parameter.

The second type of bandwidth selector is based on more mathematical arguments and require considerably more computational effort to give good answers for more general classes of underlying unknown regression functions. Due to their sophisticated nature, they will be labeled as hi-tech bandwidth selectors. Each of the hi-tech bandwidth selectors those we discussed previously are being motivated through aiming to minimize mean integrated squared error MISE of $\hat{m}_{h_n}(\cdot)$ or can be shown to attain this goal asymptotically. However these hi-tech bandwidth selectors are more complicated in practical use as they require to estimate quantities such as derivatives of unknown regression function, pilot bandwidths etc. Thus we will restrict our attention to a more simpler data driven bandwidth selector.

As used in sequential procedures (Isogai, 1987), a quick way of choosing the bandwidth therefore would be to take h_n as a function of sample size n and r, n^{-r} where r is a constant. Besides, Fan (1993) has chosen bandwidth h_n as function of n, $d \in \mathbb{R}$ and $\beta \in (0,1)$ i.e. $h_n = dn^{-\beta}$. According to the assumptions listed in Section 2.5, bandwidth h_n is a sequence satisfying $h_n \to 0$ and $nh_n \to \infty$ as $n \to \infty$. Hence $n^{-r} \to 0$ and $n^{1-r} \to \infty$ as $n \to \infty$ which result in $0 < r < 1$. In addition, it is required that the assumption (iv) is satisfied, that is for any point x_0, $h_n < x_0$ and $x_0 < 1 - h_n$. Based on these assumptions a range of values which r takes will now be decided.

Since $h_n = n^{-r}$,

$h_n < x_0 \;\Rightarrow\; n^{-r} < x_0$ and $x_0 < 1 - h_n \;\Rightarrow\; n^{-r} < 1 - x_0$ which implies

$$\begin{aligned} n^{-r} &< \min\left(x_0, 1 - x_0\right) \\ -r \log n &< \log\left[\min\left(x_0, 1 - x_0\right)\right] \\ r &> \left\{\frac{-\log\left[\min\left(x_0, 1 - x_0\right)\right]}{\log n}\right\}. \end{aligned} \tag{2.79}$$

Now we let $r_0 = \left\{\frac{-\log[\min(x_0,1-x_0)]}{\log n}\right\}$. From (2.79) and because $r \in (0,1)$, minimum value for r is $r_{\min} = \max(0, r_0)$ which revises the range of values that r can take to $r \in (r_{\min}, 1)$. Furthermore,

$$h_n < x < 1 - h_n \Longrightarrow h_n < 1 - h_n \Longrightarrow 0 < h_n < 1/2 \equiv n^r > 2$$

which leads to $r > \frac{\ln 2}{\ln n}$. By combining $r < 1$ and $r > \frac{\ln 2}{\ln n}$ it follows that $n > 2$.

Let $\triangle_0 = \min(x_0, 1 - x_0)$ then from (2.79) $r > \frac{-\ln \triangle_0}{\ln n}$. Since $r < 1$

$$\frac{-\ln \triangle_0}{\ln n} < 1 \quad \Rightarrow \quad n > \frac{1}{\triangle_0}$$

result in

$$n > \frac{1}{\min(x_0, 1 - x_0)} = \max\left(\frac{1}{x_0}, \frac{1}{1 - x_0}\right).$$

These conditions will be very useful in Chapter 3 because sequential procedures determines final sample size with an aid of a pilot sample size. Therefore when we are selecting a value for pilot sample size these conditions will be in much use.

2.8 Simulation Results

Classical kernel based approaches still have much to offer for practitioners in terms of familiarity, simplicity and accuracy. A simulation study was undertaken to illustrate performance of different kernel functions in estimating unknown regression function at a given point purely based on the bias and standard error. The computational methods involved, firstly, selecting a data design. Here we looked at both fixed design data which divided into two cases equidistant and non-equidistant and random design data.

A sample of size 25 was chosen. Thus, for equidistant fixed design data x's were in the form of $x_i = i/25;\ i = 1, \ldots, 25$. For both non-equidistant fixed design observations and random design data x's were generated from uniform distribution on $[0, 1]$ i.e. $X \sim U(0, 1)$. Since we have to use paired data (x, Y), corresponding Y values were calculated first estimating y values using a known function $m(x)$ for selected sample of x values i.e. $y = m(x)$ and then error term ε was added to that y value in order to get observed Y values i.e. $Y = y + \varepsilon = m(x) + \varepsilon$. Errors ε were generated from Normal distribution $\varepsilon \sim N(0, 0.5^2)$.

Three different models were put to test to gain an understanding of the performance of each kernel estimator. The models of choice were not only restricted to nonlinear functions but also a linear function.

- Model I: $Y = m(x) + \varepsilon = 4x + 3 + \varepsilon$
- Model II: $Y = m(x) + \varepsilon = 2\exp(-x^2/0.72) + 3\exp(-(x-1)^2/0.98) + \varepsilon$
- Model III: $Y = m(x) + \varepsilon = \sin^2(0.75x) + 3 + \varepsilon$

Once the bivariate data sets had been generated the relationships between x and Y were assumed to be unknown functions. Nonparametric kernel estimation was then applied to these samples to estimate Y for a given point of x, x_0. The local linear and Nadaraya–Watson estimation procedures involved using each data point in the sample of size 25 at a given point x_0. Next question arises of which kernel function to use for above mentioned kernel regression estimators (2.18, 2.19). Standard normal, Epanechinkov and double exponential kernels were employed. Each kernel function has its own distinct formula $K(u)$ as given in Table 2.1 where $u = \frac{x_0 - x_i}{h_n}$. Bandwidth h_n was computed as in Section 2.7.3. The estimation has been done at the points $x_0 = 0.2, 0.4, 0.5, 0.7, 0.9$ and 15000 simulated samples were used for each estimation. The actual value of y for given points x_0 are given in tables as $m(x_0)$ so that comparisons of kernel estimates can be made for different given values of x.

The given point x_0, theoretical value at x_0 $m(x_0)$, averages of local linear estimator $\overline{\hat{m}_{h_n,LL}(x_0)}$ and Nadaraya–Watson estimator $\overline{\hat{m}_{h_n,NW}(x_0)}$ along with their standard errors which are given underneath each average reported for standard normal kernel (K_{SN}), Epanechinkov kernel (K_{EP}) and double exponential kernel (K_{DE}) in Table 2.2, Table 2.3 and Table 2.4 for fixed equidistant design data, fixed non-equidistant design data and random design data respectively.

The formulae for the estimated terms displayed on the tables are given below:

- $\overline{\hat{m}_{h_n,LL}(x_0)} = \frac{1}{n_{sim}} \sum_{j=1}^{n_{sim}} (\hat{m}_{h_n,LL}(x_0))_j$
- $SE\left(\overline{\hat{m}_{h_n,LL}(x_0)}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}} \sum_{j=1}^{n_{sim}} \left((\hat{m}_{h_n,LL}(x_0))_j - \overline{\hat{m}_{h_n,LL}(x_0)}\right)^2\right\}^{1/2}$

where $(\hat{m}_{h_n,LL}(x_0))_j$ is the local linear estimate for j^{th} replication, n_{sim} (= 15000) is the number of simulation replications and $SE(\bar{Y})$ is the standard error of $\bar{Y}$. Note that standard error of a sample mean $\bar{Y}$, is calculated by $SE(\bar{Y}) = \mathbf{Var}(Y)/\sqrt{n}$ where $\mathbf{Var}(Y) = \sqrt{\sum_{j=1}^{n}(Y_i - \bar{Y})/n-1}$, $\mathbf{Var}(Y)$ is variance of Y and n is sample size.

- $\overline{\hat{m}_{h_n,NW}(x_0)} = \frac{1}{n_{sim}} \sum_{j=1}^{n_{sim}} (\hat{m}_{h_n,NW}(x_0))_j$
- $SE\left(\overline{\hat{m}_{h_n,NW}(x_0)}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}} \sum_{j=1}^{n_{sim}} \left((\hat{m}_{h_n,NW}(x_0))_j - \overline{\hat{m}_{h_n,NW}(x_0)}\right)^2\right\}^{1/2}$

where $(\hat{m}_{h_n,NW}(x_0))_j$ is the Nadaraya–Watson estimate for j^{th} replication.

These tables contain simulation results for different combinations of kernel functions and regression functions. In all these cases, Nadaraya–Watson method performs marginally worse than the local linear method although differences were not significant in the case of fixed design data. They also give you an idea about the relative closeness of average values of local linear regression smoother $\overline{\hat{m}_{h_n,LL}(x_0)}$ to the theoretical value $m(x_0)$ in comparison with the average values of Nadaraya–Watson regression smoother $\overline{\hat{m}_{h_n,NW}(x_0)}$.

Table 2.2 shows the simulation results of three different regression models based on 15000 simulations for fixed equidistant design data. The estimated averages of local linear regression estimator $\overline{\hat{m}_{h_n,LL}(x_0)}$ are very much close to the given theoretical $m(x_0)$ values for all given different values of x_0 for Model I. In this case, the regression function is a linear and the Nadaraya–Watson estimate does not perform well in detecting linearity. Hence, local linear estimator $\hat{m}_{h_n,LL}(\cdot)$ is suitable for detecting linearity. This is confirmed by the discussion given in Section 2.5 that is when $m(\cdot)$ is in linear form then local linear method has the appealing property that it is exactly unbiased for linear

$m(\cdot)$. Furthermore Nadaraya–Watson estimator $\hat{m}_{h_n,NW}(\cdot)$ has larger relative bias in the case of underlying regression function is nonlinear as well i.e. in Models II and III. Use of different kernel functions make neither significant effect on estimation of local linear regression estimator nor on Nadaraya–Watson estimator. However, averages of Nadraraya–Watson estimator based on standard normal and double exponential kernels tend to produce values more closer to corresponding theoretical values. This can be observed in all three models. As for Model I, $\overline{\hat{m}_{h_n,NW}(x_0)}$ are overestimated at the points $x_0 = 0.2, 0.4, 0.5$ and underestimated at the points $x_0 = 0.7, 0.9$. Whereas in Model II, both averages $\overline{\hat{m}_{h_n,LL}(x_0)}$ and $\overline{\hat{m}_{h_n,NW}(x_0)}$ overestimated at the interior points $x_0 = 0.4, 0.5, 0.7$ and underestimated at the boundary points $x_0 = 0.2, 0.9$. Besides local linear method overestimated all the estimations for Model III whereas Nadaraya–Watson method overestimated at the points $x_0 = 0.2, 0.4, 0.5$ and underestimated at the points $x_0 = 0.7, 0.9$. In Models II and III, local linear estimator overestimates the regression function a little but is very close to the actual values everywhere else.

Table 2.3 result is very similar to those in Table 2.2. In these designs of data, the local linear method performed well in estimation especially when the underlying regression function is linear compared to Nadaraya–Watson method. There is no significant difference between the estimates those based on different kernel functions. For fixed design data, differences between local linear and Nadaraya–Watson estimator turned out to be small with the latter being somewhat better in closing the gap between theoretical value and estimated value.

Finally, Table 2.4 shows the Nadaraya–Watson method struggled in random design data to produce more accurate estimate close to theoretical values. Whereas the local linear regression method performed well in estimating the regression function for all the given points of explanatory variable regardless of the form of the model. This seems compatible with the claim that the Nadaraya–Watson estimator can not be adaptive to different design of data as local linear method does. Nadaraya–Watson estimator fits with either standard normal or double exponential weights behave best but are more sensitive to undersmoothing or oversmoothing as compared to local linear kernel estimators.

Table 2.2: Fixed Equidistant Design Data Using Different Kernels.

x_0	$m(x_0)$	$\overline{\hat{m}_{h_n,LL}(x_0)}$			$\overline{\hat{m}_{h_n,NW}(x_0)}$		
		K_{SN}	K_{EP}	K_{DE}	K_{SN}	K_{EP}	K_{DE}
Model I : $\mathbf{y = 3x_0 + 4}$							
0.20	3.8000	3.8019	3.8019	3.8020	4.0626	4.1017	4.1492
		0.0013	0.0013	0.0013	0.0011	0.0011	0.0011
0.40	4.6000	4.5991	4.5990	4.5991	4.8733	4.9612	4.8392
		0.0008	0.0008	0.0009	0.0008	0.0008	0.0009
0.50	5.0000	5.0005	5.0005	5.0004	5.0566	5.0687	5.0466
		0.0008	0.0008	0.0010	0.0008	0.0008	0.0010
0.70	5.8000	5.7987	5.7988	5.7985	5.5316	5.4557	5.5159
		0.0010	0.0010	0.0010	0.0009	0.0009	0.0009
0.90	6.6000	6.5999	6.6000	6.5999	6.5164	6.5099	6.4478
		0.0016	0.0016	0.0017	0.0015	0.0015	0.0015
Model II : $\mathbf{y = 2exp(-x_0^2/.18) + 3exp(-(x_0-1)^2/.98)}$							
0.20	3.1628	3.1153	3.1127	3.1165	3.0628	3.0469	3.0636
		0.0013	0.0013	0.0013	0.0011	0.0111	0.0011
0.40	2.8999	2.9936	3.0008	2.9841	2.9712	2.9749	2.9645
		0.0009	0.0009	0.0009	0.0008	0.0009	0.0008
0.50	2.8232	2.9700	2.9778	2.9161	2.9663	2.9735	2.9161
		0.0008	0.0008	0.0009	0.0008	0.0008	0.0009
0.70	2.8682	2.9179	2.9243	2.9134	2.9216	2.9320	2.9211
		0.0010	0.0010	0.0010	0.0009	0.0009	0.0009
0.90	2.9918	2.9821	2.9811	2.9816	2.9728	2.9715	2.9661
		0.0016	0.0116	0.0017	0.0015	0.0015	0.0015
Model III : $\mathbf{y = sin(0.75x_0)^2 + 3}$							
0.20	3.0223	3.0314	3.0322	3.0292	3.0529	3.0556	3.0638
		0.0013	0.0013	0.0013	0.0011	0.0111	0.0011
0.40	3.0873	3.1115	3.1139	3.1085	3.1447	3.1585	3.1378
		0.0009	0.0009	0.0009	0.0008	0.0009	0.0008
0.50	3.1342	3.1632	3.1655	3.1596	3.1703	3.1741	3.1654
		0.0008	0.0009	0.0008	0.0008	0.0009	0.0008
0.70	3.2512	3.2623	3.2636	3.2610	3.2253	3.2162	3.2231
		0.0010	0.0010	0.0010	0.0009	0.0009	0.0009
0.90	3.3905	3.3911	3.3913	3.3907	3.3765	3.3754	3.3652
		0.0016	0.0117	0.0016	0.0015	0.0015	0.0015

Table 2.3: Fixed Non-equidistant Design Data Using Different Kernels.

x_0	$m(x_0)$	$\overline{\hat{m}_{h_n,LL}(x_0)}$			$\overline{\hat{m}_{h_n,NW}(x_0)}$		
		K_{SN}	K_{EP}	K_{DE}	K_{SN}	K_{EP}	K_{DE}
Model I : $\mathbf{y = 3x_0 + 4}$							
0.20	3.8000	3.8004	3.8002	3.8004	4.0581	4.0792	4.2639
		0.0014	0.0014	0.0074	0.0012	0.0052	0.0013
0.40	4.6000	4.6007	4.6008	4.6006	5.2612	5.3960	5.1625
		0.0011	0.0011	0.0012	0.0009	0.0008	0.0010
0.50	5.0000	5.0003	5.0005	5.0001	5.0976	5.1117	5.0854
		0.0008	0.0008	0.0010	0.0008	0.0008	0.0010
0.70	5.8000	5.7995	5.7997	5.7993	5.6762	5.5999	5.6209
		0.0010	0.0010	0.0015	0.0010	0.0009	0.0015
0.90	6.6000	6.6034	6.6032	6.6035	6.4200	6.4331	6.3448
		0.0021	0.0128	0.0090	0.0017	0.0012	0.0013
Model II : $\mathbf{y = 2exp(-x_0^2/.18) + 3exp(-(x_0 - 1)^2/.98)}$							
0.20	3.1628	3.0557	3.0478	3.0629	3.0199	3.0060	3.0163
		0.0014	0.0013	0.0014	0.0012	0.0012	0.0012
0.40	2.8999	2.9710	2.9738	2.9656	2.9470	2.9499	2.9457
		0.0011	0.0011	0.0011	0.0009	0.0008	0.0009
0.50	2.8232	2.9545	2.9630	2.9377	2.9485	2.9564	2.9324
		0.0008	0.0008	0.0009	0.0008	0.0008	0.0009
0.70	2.8682	2.9454	2.9510	2.9398	2.9495	2.9582	2.9481
		0.0010	0.0010	0.0010	0.0010	0.0009	0.0010
0.90	2.9918	2.9930	2.9916	2.9885	2.9682	2.9706	2.9585
		0.0021	0.0022	0.0020	0.0017	0.0017	0.0016
Model III : $\mathbf{y = sin(0.75x_0)^2 + 3}$							
0.20	3.0223	3.0393	3.0402	3.0379	3.0591	3.0603	3.0830
		0.0014	0.0013	0.0014	0.0012	0.0012	0.0012
0.40	3.0873	3.1035	3.1046	3.1016	3.1914	3.2113	3.1769
		0.0011	0.0011	0.0011	0.0009	0.0008	0.0009
0.50	3.1342	3.1602	3.1626	3.1563	3.1724	3.1765	3.1670
		0.0008	0.0008	0.0009	0.0008	0.0008	0.0009
0.70	3.2512	3.2677	3.2684	3.2667	3.2513	3.2415	3.2441
		0.0010	0.0010	0.0010	0.0010	0.0009	0.0010
0.90	3.3905	3.3936	3.3938	3.3922	3.3611	3.3633	3.3484
		0.0021	0.0022	0.0020	0.0017	0.0017	0.0016

Table 2.4: Random Design Data Using Different Kernels.

x_0	$m(x_0)$	$\overline{\hat{m}_{h_n,LL}(x_0)}$			$\overline{\hat{m}_{h_n,NW}(x_0)}$		
		K_{SN}	K_{EP}	K_{DE}	K_{SN}	K_{EP}	K_{DE}
Model I : $\mathbf{y = 3.0x_0 + 4}$							
0.20	3.8000	3.8012	3.8014	3.8013	4.0410	4.0710	4.1375
		0.0013	0.0013	0.0013	0.0016	0.0017	0.0016
0.40	4.6000	4.5987	4.5986	4.5987	4.8308	4.9039	4.8032
		0.0009	0.0009	0.0011	0.0018	0.0019	0.0017
0.50	5.0000	4.9985	4.9985	4.9985	4.9950	4.9948	4.9953
		0.0008	0.0008	0.0009	0.0019	0.0020	0.0017
0.70	5.8000	5.8000	5.8000	5.8000	5.4905	5.4147	5.4720
		0.0010	0.0010	0.0011	0.0017	0.0018	0.0016
0.90	6.6000	6.5990	6.6153	6.5996	6.4757	6.4639	6.4016
		0.0021	0.0158	0.0019	0.0019	0.0019	0.0018
Model II : $\mathbf{y = 2exp(-x_0^2/.18) + 3exp(-(x_0 - 1)^2/.98)}$							
0.20	3.1628	3.1075	3.1043	3.1090	3.0603	3.0461	3.0599
		0.0013	0.0013	0.0013	0.0011	0.0011	0.0011
0.40	2.8999	2.9900	2.9966	2.9815	2.9727	2.9762	2.9662
		0.0009	0.0009	0.0009	0.0009	0.0009	0.0009
0.50	2.8232	2.9628	2.9698	2.9487	2.9667	2.9739	2.9522
		0.0009	0.0009	0.0009	0.0008	0.0008	0.0009
0.70	2.8682	2.9139	2.9185	2.9100	2.9225	2.9321	2.9231
		0.0011	0.0010	0.0011	0.0009	0.0009	0.0009
0.90	2.9918	2.9844	3.0020	2.9815	2.9689	2.9673	2.9615
		0.0021	0.0158	0.0019	0.0016	0.0016	0.0015
Model III : $\mathbf{y = sin(0.75x_0)^2 + 3}$							
0.20	3.0223	3.0316	3.0326	3.0299	3.0515	3.0536	3.0633
		0.0013	0.0013	0.0013	0.0011	0.0011	0.0011
0.40	3.0873	3.1119	3.1144	3.1090	3.1399	3.1517	3.1338
		0.0009	0.0009	0.0009	0.0009	0.0009	0.0009
0.50	3.1342	3.1609	3.1631	3.1575	3.1618	3.1641	3.1583
		0.0008	0.0008	0.0009	0.0008	0.0008	0.0009
0.70	3.2512	3.2620	3.2630	3.2610	3.2203	3.2106	3.2182
		0.0010	0.0010	0.0011	0.0009	0.0009	0.0010
0.90	3.3905	3.3898	3.4102	3.3894	3.3690	3.3670	3.3570
		0.0021	0.0158	0.0019	0.0016	0.0016	0.0016

Figures 2.1, 2.2 and 2.3 allow a closer assessment of the sensitivity to choice of kernel weights in estimators $\hat{m}_{h_n,LL}(\cdot)$ and $\hat{m}_{h_n,LL}(\cdot)$ for the three data designs, fixed equidistant design, fixed non-equidistant design and random design. Also these figures show how fast averages of both estimators $\overline{\hat{m}_{h_n,NW}(x_0)}$

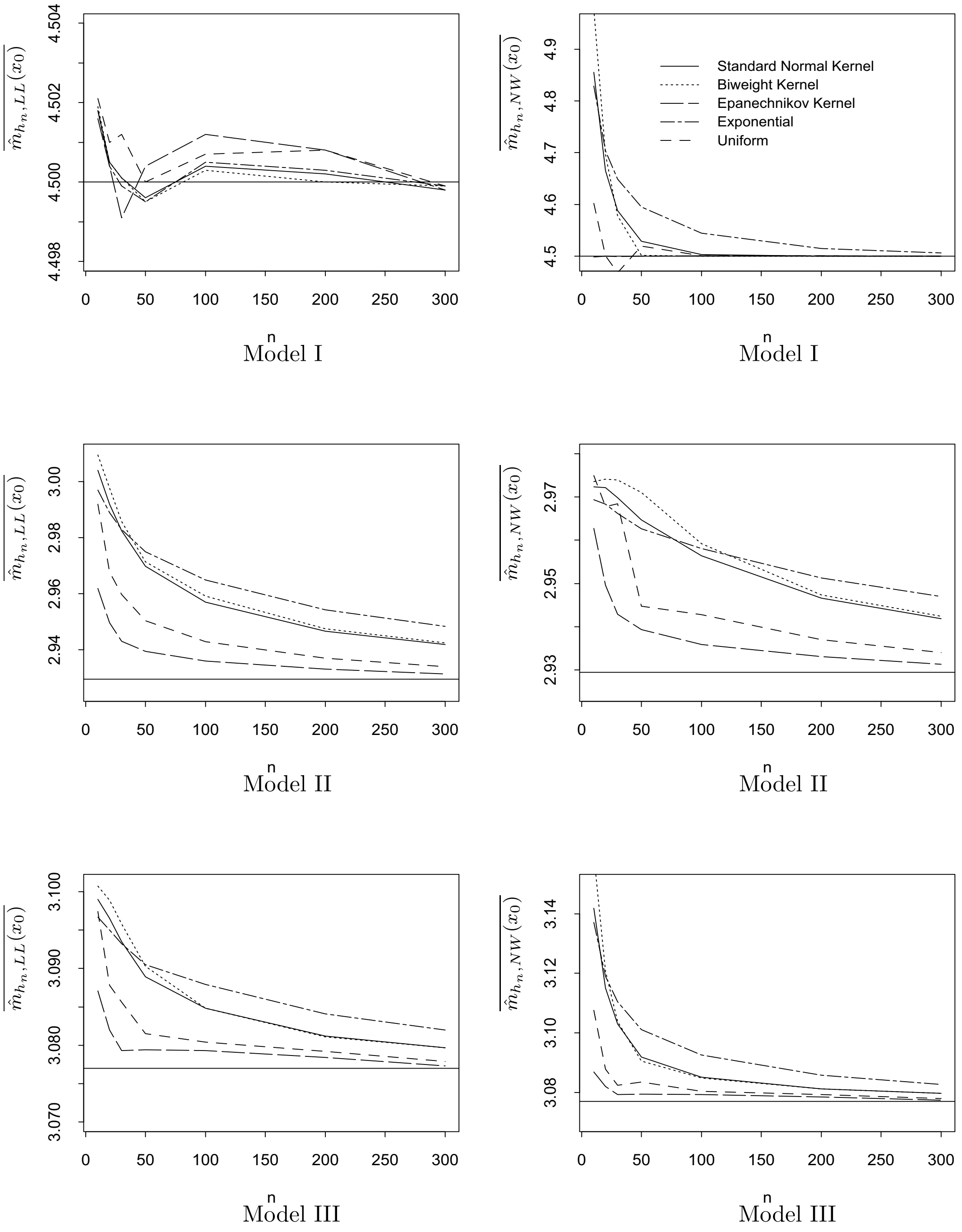
and $\overline{\hat{m}_{h_n,LL}(x_0)}$ are approached for theoretical value estimated at a given point $x_0 = 0.375$. To visualize the performance of the regression estimators here we take into consideration five different kernel functions which are standard normal, epanechinkov, biweight, uniform and double exponential functions. The most striking result is that average estimates of local linear method $\overline{\hat{m}_{h_n,LL}(x_0)}$ approaches to the theoretical value faster than those from Nadaraya–Watson $\overline{\hat{m}_{h_n,NW}(x_0)}$ estimation for each different kernel.

Figures 2.1 and 2.2 correspond to the case where the underlying data designs are fixed equidistant and fixed non-equidistant respectively. Figure 2.2 corresponds to the case of random design data. All three figures first showing the relative performance of the two estimators based on five different kernels for three different models and also showing how the accuracy of each estimator changes for sample sizes ranging from small to large. In all these cases local linear method performs marginally better than the Nadaraya–Watson method although differences were not significant for fixed equidistant or non-equidistant design data. All estimates behave well for large sample sizes as expected from asymptotic theory. The behaviour of local linear method is the same for fixed equidistant design or non-equidistant as for random design for three different models. By intuition and from the discussion given in Section 2.4 one would expect regression estimates by Nadaraya–Watson method to be equivalent to local linear regression estimates when the data design is fixed for large sample sizes.

Both estimators start relatively high values for small sample sizes and reach corresponding theoretical values as sample size increases. In general, estimators based on biweight kernel starts with highly overestimated values and those based on uniform kernel start with less overestimated values. The local linear method with biweight kernel is best among others for small to large sample sizes for random design data. Although the effects of using between different kernels are not serious. The simulations have shown that the local linear estimator beats its competitor from small sample sizes onward for random design data. Similar conclusions can be obtained for other situations. The local linear smother thus leads to a much more valid in overall cases.

Nevertheless the chosen different weighting schemes guarantee the correct adaptation to the different design of data. This means the use of different kernels is not such a big problem as we have least expected. When the sample size is not too small, local linear estimator gives results that are good in the sense of closeness to the theoretical value in all different cases. The accuracy of both methods dropped off significantly for samples as small as 10 in all the different combinations of data designs and kernel functions. Of course this is simply of the fact that one can hardly expect to estimate an estimate for unknown regression function accurately without sufficient data.

In the case of fixed design, the Nadaraya–Watson method performed well in estimating the underlying regression function but performed poorly in random design. Moreover, the accuracy of Nadaraya–Watson estimator improved as sample size increased from moderate to large. This is the same for fixed design as for random design. Another aspect that has been stressed out is Nadaraya–Watson method fairly well for Model III compared to Model II even both are nonlinear functions. This is due to the structural difference between two models.



Model I Model I

Model II Model II

Model III Model III

Figure 2.2: Fixed Equidistant Design Data Using Different Sample Sizes and Kernels.

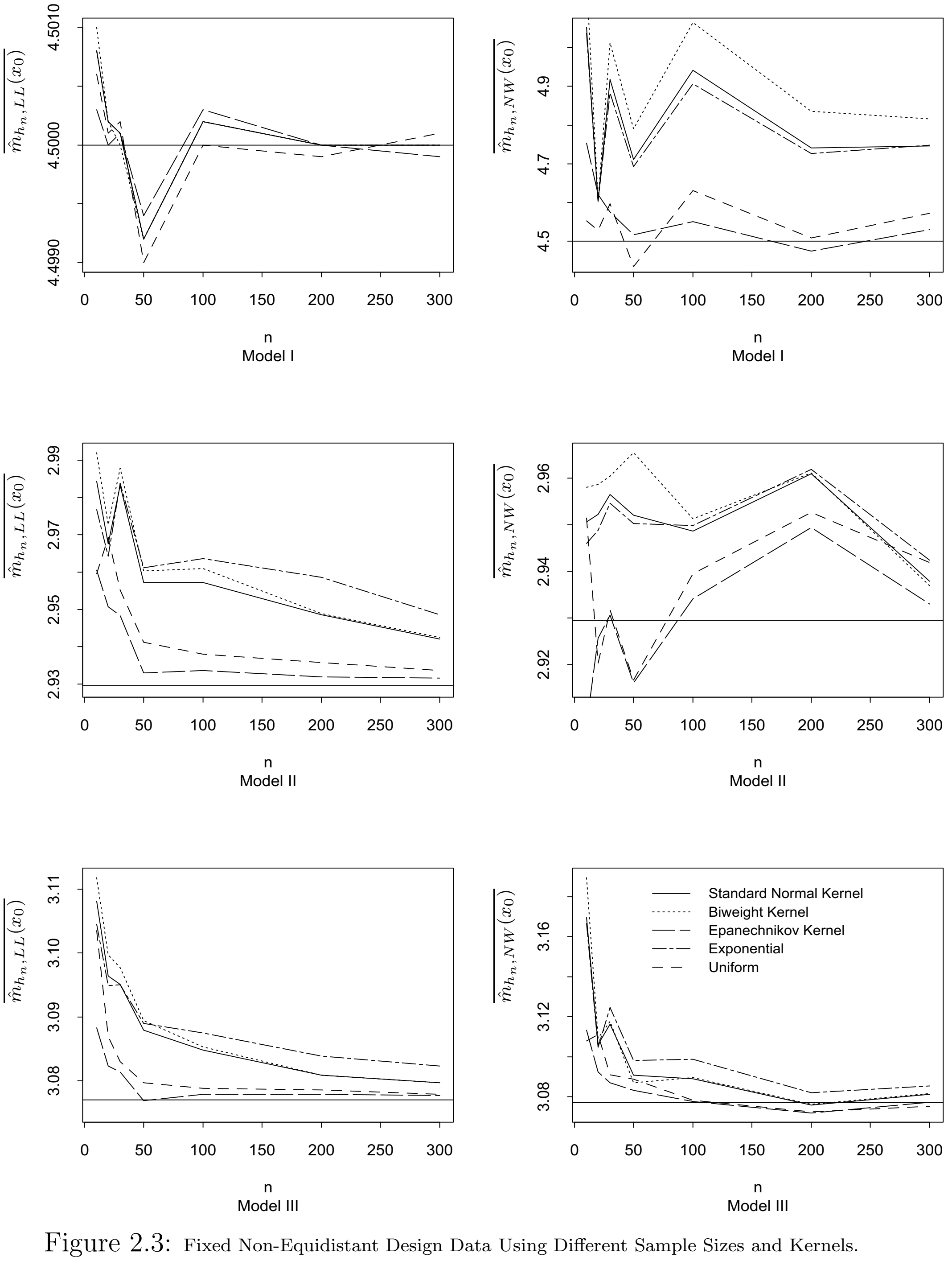
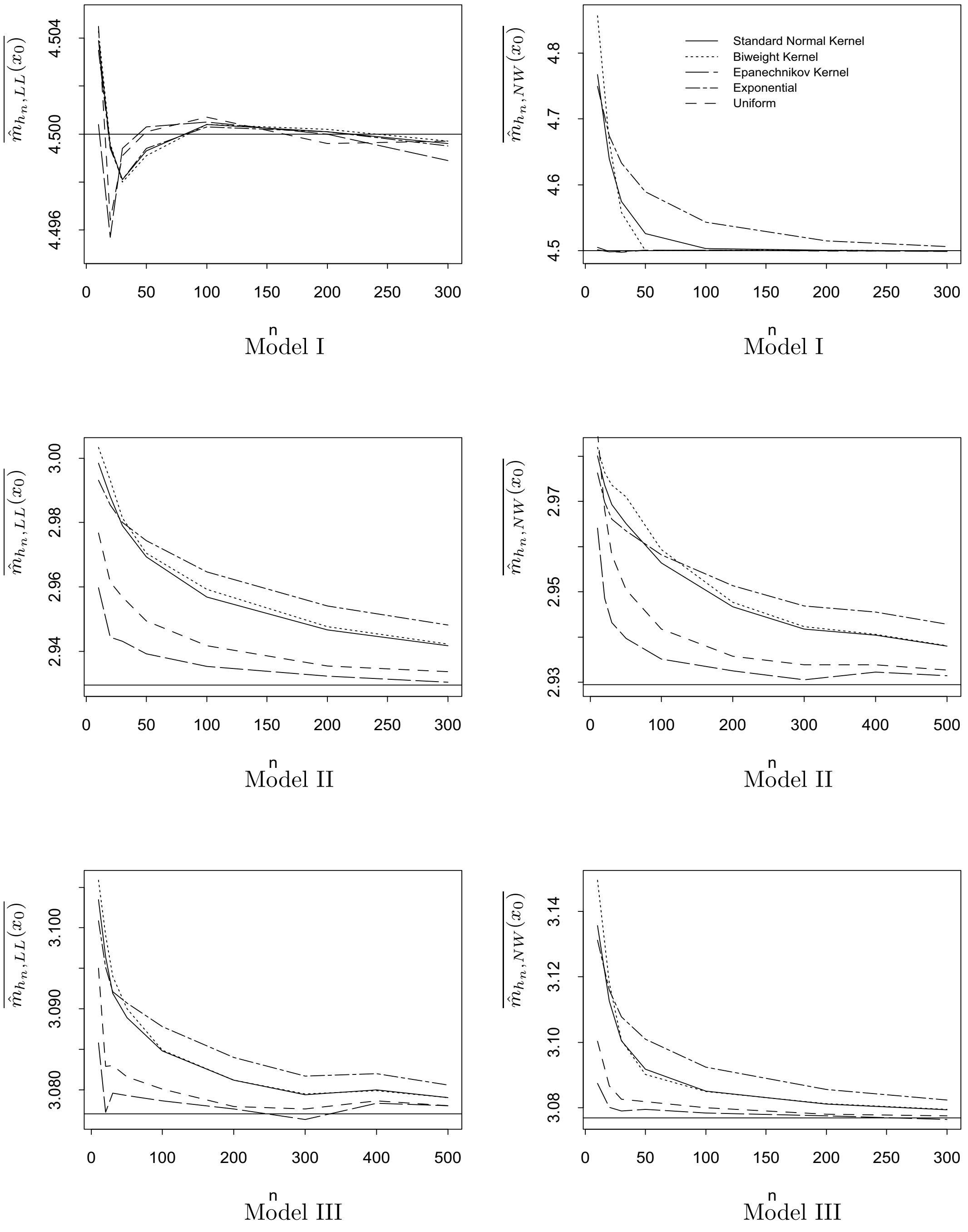


Figure 2.3: Fixed Non-Equidistant Design Data Using Different Sample Sizes and Kernels.



Model I

Model I

Model II

Model II

Model III

Model III

Figure 2.4: Random Design Data Using Different Sample Sizes and Kernels.

Remarks:

Asymptotically the performance of the local linear regression smoother is the same as the Nadaraya–Watson estimator. However, figures indicate that the former smoother performs better than the latter at even small to moderate sample sizes. This suggests that the asymptotic theory takes in effect at a larger sample size for the Nadaraya–Watson estimator. The closeness between theoretical value and the corresponding estimates from each method decreases when sample size increases regardless of the design of data on the hand. An advantage of local linear method to Nadaraya–Watson method is that it can be adapted very easily to a variety of different data design contexts. According to theory the Nadaraya–Watson estimator is expected to perform as good as local linear method for fixed design but this is not very borne out by simulation results.

The simulation results suggest that at least moderate size of samples, the choice of kernel is not critical. However, the properties of both estimators tend to depend least critically on the choice of kernel function but on bandwidth selector. The bandwidth selector which is discussed in detail in Section 2.7.3 seems to be more appropriate as well as quick and simple. Also it can be employed in practical situations as it provides estimate very close to the theoretical values regardless of the value of the point estimation and choice of various kernel functions. The results from this simulation study is consistent with the properties of both estimators discussed in the previous sections of this chapter.

Chapter 3

Residual Variance Estimation in Nonparametric Kernel Regression

3.1 Introduction

Methods for nonparametric residual estimation is an important subject of statistical research with practical relevance. This chapter gives an overview of a variety of methods for nonparametric variance estimation. Therefore we concentrate on the main ideas of, differences as well as similarities between such methods. Consider the nonparametric regression model as defined in Section 2.1

$$Y_i = m(x_i) + \varepsilon_i, \quad i = 1, \ldots, n \tag{3.1}$$

where $Y_1, \ldots, Y_n$ are observable data variables with respect to the design variables $x_1, \ldots, x_n$ and $\mathbf{Var}(\varepsilon_i) = \sigma^2$ and the regression function $m(\cdot)$ are unknown. A great deal of research has been done in the estimation of the unknown regression function $m(\cdot)$. Nevertheless estimation of σ^2 is equally as important as the estimation of $m(\cdot)$ itself because knowledge of σ^2 is necessary in constructing confidence interval for $m(\cdot)$ as well as many other applications such as prediction, calculation of smallest possible sample sizes with an aid of sequential procedures etc. Most of the objectives and data-dependent stopping rules for sample size selection need an explicit estimator of the residual variance. Besides, inference about the regression function also requires knowledge about the residual variance.

The class of estimators of σ^2 which covers all the proposed estimators have quadratic forms. If we use matrix and vector notation the variance estimator can be written as

$$\hat{\sigma}^2 = \frac{\mathbf{Y^T Q Y}}{tr(\mathbf{Q})} \tag{3.2}$$

where $\mathbf{Q}$ is a suitable symmetric and positive semi-definite matrix, $\mathbf{Y}^T = (Y_1, Y_2, \ldots, Y_n)$ and $tr(\mathbf{Q})$ is a trace of the matrix $\mathbf{Q}$. When $\mathbf{m(x)} = \mathbf{0}$, the divisor $tr(\mathbf{Q})$ ensures $\hat{\sigma}^2$ is an unbiased estimator for σ^2 that is $\mathbf{E}[\hat{\sigma}^2] = \sigma^2$ where $\mathbf{m(x)}^T = [m(x_1), m(x_2), \ldots, m(x_n)]$. Since $\mathbf{Y} = \mathbf{m(x)} + \varepsilon$ where $\varepsilon^T = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n]$ we can rewrite (3.2) as

$$\begin{aligned}
\hat{\sigma}^2 &= \frac{(\mathbf{m(x)} + \varepsilon)^{\mathbf{T}}\, \mathbf{Q}\, (\mathbf{m(x)} + \varepsilon)}{tr(\mathbf{Q})} \\
&= \frac{\left(\mathbf{m(x)^T} + \varepsilon^{\mathbf{T}}\right) \mathbf{Q}\, (\mathbf{m(x)} + \varepsilon)}{tr(\mathbf{Q})} \\
&= \frac{\mathbf{m(x)^T Q m(x)} + 2\mathbf{m(x)^T Q}\varepsilon + \varepsilon^{\mathbf{T}}\mathbf{Q}\varepsilon}{tr(\mathbf{Q})}.
\end{aligned} \tag{3.3}$$

From (3.3) we can see that quadratic estimate of the residual variance consists of three terms: a natural estimator of σ^2, $\frac{\varepsilon^{\mathbf{T}}\mathbf{Q}\varepsilon}{tr(\mathbf{Q})}$, a positive bias $\frac{\mathbf{m(x)^T Q m(x)}}{tr(\mathbf{Q})}$ and $\frac{2\mathbf{m(x)^T Q}\varepsilon}{tr(\mathbf{Q})}$. The unbiasedness property of estimated residual variance estimator $\hat{\sigma}^2$ is only valid for $\mathbf{m(x)} = \mathbf{0}$ which seems very strong condition. However, from (3.3) we can see all that is required is that $\mathbf{m(x)^T Q m(x)} = \mathbf{0}$ which may hold or nearly hold for $\mathbf{m(x)} \neq \mathbf{0}$.

If $\mathbf{m}(\mathbf{x}) = \mathbf{0}$ then

$$\hat{\sigma}^2 = \frac{\varepsilon^{\mathbf{T}}\mathbf{Q}\varepsilon}{tr(\mathbf{Q})}.$$

Hence

$$\begin{aligned}\mathbf{E}[\hat{\sigma}^2] &= \frac{\mathbf{E}\left[\varepsilon^{\mathbf{T}}\mathbf{Q}\varepsilon\right]}{tr(\mathbf{Q})} \\ &= \frac{tr(\mathbf{Q}\boldsymbol{\sigma}^{\mathbf{2}}\mathbf{I}) + \mathbf{E}[\varepsilon]^{\mathbf{T}}\mathbf{Q}\mathbf{E}[\varepsilon]}{tr(\mathbf{Q})} \\ &= \frac{tr(\mathbf{Q})\sigma^2}{tr(\mathbf{Q})} \\ &= \sigma^2\end{aligned}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{E}[\varepsilon] = \mathbf{0}$.

During last three decades an assortment of proposals have been made explaining as to how σ^2 can be estimated nonparametrically. The available estimators are broadly divided into two subclasses of estimators which are variance estimators based on curve fitting and variance estimators based on differencing. Silverman (1985) and Wahba (1983) proposed estimators for σ^2 using naive nonparametric residuals obtained by subtracting an appropriate smoothed curve from the observations. Rice (1984) introduced a simple difference-based estimator of σ^2 for fixed design data. Several authors discussed improvements (Gasser et. al., 1986; Buckley and Eagleson, 1989; Hall and Marron, 1990; Hall et. al., 1990).

Variance estimators based on curve fitting estimate the residual variance σ^2 with a sum of squared residuals $\hat{\varepsilon}^T\hat{\varepsilon}$ from a nonparametric fit $\hat{\mathbf{m}}(\mathbf{x})$. In general, the residuals are estimated mostly by a linear function $\hat{\varepsilon} = \mathbf{Y} - \hat{\mathbf{m}}(\mathbf{x})$. Curve fitting residual variance estimators proposed by Wahba (1978) and Carter and Eagleson (1992) have employed spline smoothing methods to estimate $\hat{\mathbf{m}}(\mathbf{x})$ whereas Muller and Stadtmuller (1987), Hall and Carroll (1989) and Hall and Marron (1990) used kernel-based smoothing estimators.

From Section 2.2 it is obvious that the nonparametric curve fitting is a linear fit of the form $\hat{\mathbf{m}}(\mathbf{x}) = \mathbf{WY}$ where $\mathbf{W} = \{w_{ij}\}$; $i, j = 1, \ldots, n$. This lead to the corresponding residual variance based on curve fitting estimator of the form (3.2) with $\mathbf{Q} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$ where $\mathbf{I}$ is an identity matrix. Note that every residual variance estimator based on curve fitting depends explicitly on the choice of bandwidth h_n. In Section 3.2 we describe in detail the residual variance estimator based on curve fitting suggested by Hall and Marron (1990).

The second subclass of residual variance is difference-based estimators. In this setting $\mathbf{Q} = [\mathbf{D}^T\mathbf{D}]_{n \times n}$ is a symmetric non-negative definite matrix and the elements of $\mathbf{D}_{(n-g)\times n}$ depend only on the values of predictor x_i but not on the observations $\{Y_i\}_{i=1}^n$ and the order of a difference based estimator g which is the number of related observations $(\mathbf{X}, \mathbf{Y})$ involved to estimate a local residual $\hat{\varepsilon}$. The main advantage of this type of residual variance estimator is, it does not require an estimator for unknown regression function $\hat{\mathbf{m}}(\mathbf{x})$. However, the order g of the difference based estimator has an impact which may be comparable with the estimating unknown regression function in the curve fitting based residual variance estimators. Rice (1984) and Gasser et al. (1986) suggested difference-based residual variance estimators for the fixed designed data based on first-order $g = 1$ and second-order $g = 2$ respectively. Holger et al. (1998) showed that in practice difference-based estimators are more attractive because they have a small bias for small sample size besides large asymptotic variance. Hence, he explained that the difference-based residual variance estimators proposed by Gasser et. al. (1986) and Rice (1984) do not achieve the asymptotic optimal rate which is defined by

$$MSE[\hat{\sigma}^2] = n^{-1}\mathbf{Var}[\varepsilon^2] + o(n^{-1}) \tag{3.4}$$

as for curve fitting based residual variance estimator suggested by Hall and

Marron (1990). However, Müller et. al. (2003) proposed a difference-based residual variance for random designed data using covariate-matched U-statistic which achieves the asymptotic optimal rate. In Section 3.3, we discuss the difference-based residual variance estimators which are proposed by Rice (1984), Gasser et. al. (1986) and Müller et. al. (2003) in detail.

In the remaining part of this chapter, we follow the same standard assumptions for nonparametric regression model and the kernel regression estimators which are introduced in Chapter 2. Here we consider the design points $x_1, \ldots, x_n$ are either from a fixed design setting (equidistant $x_i = i/n$; $i = 1, \ldots, n$ or non-equidistant $x_i \neq i/n$) or a sample of random variables with a common design density $f(x)$. For random design, we assume that $(X_i, Y_i)_{i=1}^n$ is a bivariate sample of independent and identically distributed random variables with the same distribution as a random variable (X, Y) that satisfies $m(x) = \mathbf{E}(Y|X = x)$ and $\sigma^2 = \mathbf{Var}(Y|X = x)$. In addition, we assume that the design density $f(x)$ has support $[a, b]$, where $a, b \in \mathbb{R}$ and $a < b$. Variance and regression estimations are restricted to this compact interval. The error variables ε_i are assumed to be independent and identically distributed random variables such that $\mathbf{E}[\varepsilon_i] = 0$ and variance is constant $\mathbf{Var}[\varepsilon_i] = \sigma^2$. Unless otherwise stated, it is assumed that the design density $f(x)$ and the regression function $m(\cdot)$ have at least two continuous derivatives on the interval $[a, b]$.

In Section 3.4, we compare the estimation procedures discussed in Sections 3.2 and 3.3 using a simulation study for different cases depending on the data design, distribution of residual variance and underlying regression function. It is important to note that we do not address the problem of which estimator might be the best estimator; rather we are more interested in being confident that the estimate use in a particular situation is an accurate.

This is very vital as the accuracy of the residual variance is a very sensitive issue especially in the performance of stopping rules which we will propose in Chapter 4 and Chapter 5.

3.2 Variance Estimators Based on Curve Fitting

Estimated residuals based on nonparametric regression model defined by (3.1) are

$$\hat{\varepsilon}_i = Y_i - \hat{m}(x_i); \quad i = 1, \ldots, n. \tag{3.5}$$

The design points x_i's are confined to the interval $[0, 1]$. The analysis applies equally to both types of design. In this approach a variance estimator $\hat{\sigma}^2$ is obtained as a weighted average of a sequence of squared residuals and is defined by

$$\hat{\sigma}^2 = \frac{1}{c\,n} \sum_{i=1}^{n} (Y_i - \hat{m}(x_i))^2 \tag{3.6}$$

where the normalizing factor c may be defined such that the variance estimator is unbiased when unknown regression function is zero; that is $m(X) = 0$.

Early references for such approaches are Breiman and Meisel (1976) who used piecewise linear fits with an adaptive number of pieces for curve fitting and Cleveland (1979) used a robust variant of local polynomial estimation for regression estimation.

In this section it is assumed that kernel type regression estimator $\hat{m}(x_i) = \sum_{j=1}^{n} w_{ij} Y_j;\ i = 1, \ldots, n$ is used to estimate the residuals. Hence

$$\hat{\varepsilon}_i = Y_i - \sum_{j=1}^{n} w_{ij} Y_j; \quad i = 1, \ldots, n \tag{3.7}$$

$$w_{ij} = \frac{K\left\{\frac{(x_i - x_j)}{h_n}\right\}}{\sum_{k=1}^{n} K\left\{\frac{(x_i - x_k)}{h_n}\right\}}$$

where $K(\cdot)$ is a kernel function, h_n is a bandwidth and w_{ij} be constants satisfying $\sum_j w_{ij} = 1$ for each i. The obvious problem that occurs when using such estimators is the choice of the bandwidth h_n. Here we shall stick to the method discussed in Section 2.7.3 as results were shown to be promising.

The assumption (3.8) lead to variance estimator, defined as

$$\hat{\sigma}^2 = \frac{1}{cn} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{n} w_{ij} Y_j \right)^2 . \tag{3.8}$$

By letting w_{ij}; $i, j = 1, \ldots, n$ be the entries of weight matrix $\mathbf{W}$, $\mathbf{Y} = [Y_i]_{n\times 1}$ and $\mathbf{I}$ be the identity matrix we can rewrite (3.7) and (3.8) in matrix form as

$$\begin{aligned} \hat{\varepsilon} &= \mathbf{Y} - \mathbf{WY} \\ &= (\mathbf{I} - \mathbf{W})\mathbf{Y} \end{aligned} \tag{3.9}$$

and

$$\begin{aligned} \hat{\sigma}^2 &= \hat{\varepsilon}^{\mathbf{T}} \hat{\varepsilon} \\ &= \{(\mathbf{I} - \mathbf{W})\mathbf{Y}\}^T (\mathbf{I} - \mathbf{W})\mathbf{Y} \\ &= \mathbf{Y}^{\mathbf{T}} (\mathbf{I} - \mathbf{W})^{\mathbf{T}} (\mathbf{I} - \mathbf{W})\mathbf{Y}. \end{aligned} \tag{3.10}$$

3.2.1 Hall and Marron Estimator $\hat{\sigma}^2_{HM}$

Residual variance estimator proposed by Hall and Marron (1990) $\hat{\sigma}^2_{HM}$ for random design based on rth-order differences have the property that

$$\mathbf{E}[\hat{\sigma}_{HM} - \sigma^2]^2 \sim n^{-1} c_r \mathbf{Var}(\varepsilon^2) \tag{3.11}$$

where $1 \leq c_r \leq \infty$ and c_r is a constant depends only on the order r.
The aim is to show that there exist simple estimators which are applicable whenever the error distribution has finite fourth moment and which achieve $c_r = 1$ in the above formula thus achieving the optimal rate (3.4). $\hat{\sigma}^2_{HM}$ is simply defined as the mean square of a sequence of residuals.

Let $\mathbf{S} = \sum_{i=1}^{n} \hat{\varepsilon_i}^2$ then from (3.10)

$$\mathbf{S} = \mathbf{Y^T}(\mathbf{I} - \mathbf{W})^\mathbf{T}(\mathbf{I} - \mathbf{W})\mathbf{Y}. \tag{3.12}$$

When the mean function $m(\cdot)$ is zero then from (3.1), $Y_i = \varepsilon_i$ which leads to $\mathbf{E}[Y_i] = 0$ and $\mathbf{E}[Y_i^2] = \sigma^2$. Thus $\mathbf{E}[\mathbf{S}] = \nu\sigma^2$ where $\nu = n - 2\sum_i w_{ii} + \sum\sum_{ij} w_{ij}^2$. This is shown below:

$$\begin{aligned}
\mathbf{E}[\mathbf{S}] &= \mathbf{E}\left[\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{n} w_{ij}Y_j\right)^2\right] \\
&= \mathbf{E}\left[\sum_{i=1}^{n} Y_i^2 - 2\sum_{i=1}^{n} Y_i \sum_{j=1}^{n} w_{ij}Y_j + \sum_{i=1}^{n}\left(\sum_{i=1}^{n} w_{ij}Y_j\right)^2\right] \\
&= n\sigma^2 - 2\mathbf{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}Y_iY_j\right] + \sum_{i=1}^{n}\mathbf{E}\left[\left(\sum_{j=1}^{n} w_{ij}Y_j\right)^2\right] \\
&= n\sigma^2 - 2\sum_{i=1}^{n} w_{ii}\mathbf{E}\left[Y_i^2\right] + \sum_{i=1}^{n}\mathbf{E}\left[\sum_{j=1}^{n} w_{ij}^2Y_j^2\right] \\
&= n\sigma^2 - 2\sigma^2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2\mathbf{E}\left[Y_j^2\right] \\
&= n\sigma^2 - 2\sigma^2\sum_{i=1}^{n} w_{ii} + \sigma^2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2
\end{aligned}$$

$$
\begin{aligned}
\mathbf{E}[\mathbf{S}] &= \sigma^2 \left[n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2 \right] \\
&= \sigma^2 \nu \qquad (3.13)
\end{aligned}
$$

where $\nu = n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2$.

Note that ν is the trace of $(\mathbf{I}-\mathbf{W})^{\mathbf{T}}(\mathbf{I}-\mathbf{W})$. Motivated from the fact

$$\frac{\mathbf{E}[\mathbf{S}]}{\nu} = \sigma^2,$$

Hall and Marron (1990) proposed a residual estimator $\hat{\sigma}_{HM}^2$ which is consistent with (3.2)

$$\hat{\sigma}_{HM}^2 = \frac{1}{\nu}\mathbf{Y}^T \left(\mathbf{I}-\mathbf{W}\right)^T \left(\mathbf{I}-\mathbf{W}\right)\mathbf{Y} \qquad (3.14)$$

where

$$
\begin{aligned}
\nu &= tr\left\{(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\right\} \\
&= n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2 \\
&= n\left(1 - \frac{2}{n}\sum_{i=1}^{n} w_{ii} + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2\right) \\
&= c\,n \qquad (3.15)
\end{aligned}
$$

and the normalizing factor c is defined as follows:

$$c = 1 - \frac{2}{n}\sum_{i=1}^{n} w_{ii} + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2.$$

Therefore when mean function $m(\cdot)$ is zero

$$\frac{1}{c\,n}\mathbf{E}[\mathbf{S}] = \frac{\sigma^2\nu}{c\,n} = \sigma^2 \qquad (3.16)$$

and (3.16) says $\hat{\sigma}^2_{HM}$ is an unbiased for σ^2. Sarda and Vieu (2000) showed that the normalizing factor c can be neglected asymptotically as $c = 1 + O_p\left(\{nh_n\}^{-1}\right)$ for kernel regression estimators.

The above (3.16) is the motive behind the proposed residual estimator $\hat{\sigma}^2_{HM}$ (3.14) which can be written as

$$\hat{\sigma}^2_{HM} = \frac{\sum_{i=1}^n (Y_i - \sum_{j=1}^n w_{ij} Y_j)^2}{n - 2\sum_{i=1}^n w_{ii} + \sum_{i=1}^n \sum_{j=1}^n w_{ij}^2}. \tag{3.17}$$

However, proposed estimator $\hat{\sigma}^2_{HM}$ is not an unbiased estimator of σ^2 if mean function $m(\cdot)$ not equal to zero. In this case, we will have

$$\mathbf{E}[\hat{\sigma}^2_{HM}] = \sigma^2 + \nu^{-1} \sum_i \delta_i^2 \tag{3.18}$$

where

$$\delta_i = m(x_i) - \sum_j w_{ij} m(x_j) = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{m}(\mathbf{x}).$$

This is shown below:

$$\begin{aligned}
\mathbf{E}[\hat{\sigma}^2_{HM}] &= \nu^{-1} \mathbf{E}[\mathbf{S}] \\
&= \nu^{-1} \mathbf{E}\left[\sum_{i=1}^n \left(Y_i - \sum_{j=1}^n w_{ij} Y_j\right)^2\right] \\
&= \nu^{-1} \mathbf{E}\left\{\sum_{i=1}^n \left[(m(X_i) + \varepsilon_i) - \sum_{j=1}^n w_{ij}\left(m(X_j) - \varepsilon_j\right)\right]^2\right\} \\
&= \nu^{-1} \mathbf{E}\left\{\sum_{i=1}^n \left[m(X_i) - \sum_{j=1}^n w_{ij} m(X_j) + \varepsilon_i - \sum_{j=1}^n w_{ij} \varepsilon_j\right]^2\right\} \\
&= \nu^{-1}\left\{n\sigma^2 + \sum_{i=1}^n \delta_i^2 + \mathbf{E}\left[\sum_{i=1}^n \left(\sum_{j=1}^n w_{ij}\varepsilon_j\right)^2\right]\right\} \\
&\quad - \nu^{-1}\left\{2\mathbf{E}\left[\sum_{i=1}^n \varepsilon_i \sum_{j=1}^n w_{ij}\varepsilon_j\right]\right\}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{E}[\hat{\sigma}^2_{HM}] &= \nu^{-1}\left\{n\sigma^2 + \sum_{i=1}^{n}\delta_i^2 + \sum_{i=1}^{n}\mathbf{E}\left[\sum_{j=1}^{n} w_{ij}^2\varepsilon_j^2\right] - 2\sum_{i=1}^{n} w_{ii}\mathbf{E}[\varepsilon_i^2]\right\} \\
&= \nu^{-1}\left\{n\sigma^2 - 2\sigma^2\sum_{i=1}^{n} w_{ii} + \sigma^2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2 + \sum_{i=1}^{n}\delta_i^2\right\} \\
&= \nu^{-1}\sigma^2\left[n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2\right] + \nu^{-1}\sum_{i=1}^{n}\delta_i^2 \\
&= \sigma^2 + \nu^{-1}\sum_{i=1}^{n}\delta_i^2. \qquad (3.19)
\end{aligned}
$$

Similarly it can be shown that

$$
\mathbf{Var}[\hat{\sigma}^2_{HM}] = \nu^{-2}\left\{\sum_{j}\mathbf{E}(\Delta_j^2) + 2\sigma^2\sum\sum_{j\neq k} t_{jk}^2\right\} \qquad (3.20)
$$

where

$$
t_{jk} = \sum_{i} w_{ij}w_{ik} - 2w_{jk}
$$

and

$$
\Delta_j = \left(\delta_j - \sum_{i}\delta_i w_{ij}\right)\varepsilon_j + \left(1 - 2w_{ij}\sum_{i} w_{ij}^2\right)(\varepsilon_j^2 - \sigma).
$$

It may be proved that if $h_n \to 0$ and $nh_n \to \infty$ then $\hat{\sigma}^2_{HM} - \mathbf{E}[\hat{\sigma}^2_{HM}]$ is asymptotically normally distributed with variance $n^{-1}\mathbf{Var}(\varepsilon_i^2)$. Hall and Marron (1990) showed that $\hat{\sigma}^2_{HM}$ has optimal first and second order properties (3.11). Müller et. al. (2004) showed that, more precisely, Hall and Marron's estimator $\hat{\sigma}^2_{HM}$ is stochastically equivalent to $\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2$.

3.3 Residual Variance Estimators Based on Differences

The motivation of the variance estimators based on differencing is slightly different from those based on curve fitting. Instead of aiming to estimate residuals, especially squared residuals, they first estimate a functional of the variance and then normalize this estimator. Suppose that a constant variance is to be estimated in our regression model, under the assumptions mentioned in Section 3.1. The basic idea is using coefficients of some kind of differencing scheme $d_{i,k}$ based on observed data to estimate the i^{th} residual $\hat{\varepsilon_i}$:

$$\hat{\varepsilon_i} = \sum_{k=0}^{g} d_{i,k} Y_{i+k}; \quad i = 1, \ldots, n-g \tag{3.21}$$

where $g \geq 1$. Note that g only indicates how many and which data points are to be used to form the residuals; it does not specify a further parameter of the estimation procedure. The order of differencing scheme g restricts which of the residuals ε_is are estimated by the differencing scheme as $i = 1, \ldots, n-g$.

Assumptions on the differencing weights are

$$\sum_{k=0}^{g} d_{i,k} \left(x_{i+k} - x_i\right)^j = 0; \quad j = 0, \ldots, g-1, \quad \sum_{k=0}^{g} d_{i,k}^2 = 1 \quad \text{and}$$
$$\sum_{i=1}^{n-g} \sum_{k=0}^{g} d_{i,k}^2 = n - g. \tag{3.22}$$

The assumptions lead to variance estimator defined as

$$\hat{\sigma}^2 = \sum_{i=1}^{n-g} \frac{1}{n-g} \left(\sum_{k=0}^{g} d_{i,k} Y_{i+k} \right)^2 . \tag{3.23}$$

Note that as mentioned in Section 3.1 the above defined residual variance estimator (3.23) is unbiased when the unknown regression function is zero. We show and discuss this fact for the residual variance estimators based on the order $g = 1$ and $g = 2$ separately later in this section.

Let us introduce a matrix $\mathbf{D_g}$, defined as

$$\mathbf{D_g} = \begin{pmatrix} d_{1,0} & & \ldots & d_{1,g} & & \mathbf{0} \\ \vdots & \ddots & & & \ddots & \vdots \\ 0 & \ldots & d_{n-g,0} & \ldots & & d_{n-g,g} \end{pmatrix}_{(n-g)\times n}$$

Then the residuals defined in (3.21) can also be written in the vector form $\hat{\varepsilon} = \mathbf{D_g Y}$. Thus difference-based estimator of the residual variance defined in (3.23) can also be written in quadratic form which is consistent with (3.2) as

$$\hat{\sigma}^2 = \frac{\varepsilon^T \varepsilon}{(n-g)} = \frac{\mathbf{Y^T Q Y}}{(n-g)}$$

where $\mathbf{Q} = \mathbf{D_g}^T \mathbf{D_g}$ and $(n-g)$ is the trace of $\mathbf{Q} = \mathbf{D_g}^T \mathbf{D_g}$.

3.3.1 Rice Estimator $\hat{\sigma}_R^2$

Rice (1984) proposed a residual variance estimator based on differencing scheme of order one that is $g = 1$ to be used in fixed design data. That is, it satisfies (3.23) for $g = 1$. The crucial fact is here we estimate only $n - 1$ residuals by the differencing scheme. Because here $i = 1, \ldots, n-g \quad \Rightarrow$ $i = 1, \ldots, n-1$. Under this circumstance, residuals are defined as follows:

$$\hat{\varepsilon}_i = \sum_{k=0}^{1} d_{i,k} Y_{i+k}; \quad i = 1, \ldots, n-1. \tag{3.24}$$

Also variance estimator is defined as

$$\hat{\sigma}^2 = \sum_{i=1}^{n-1} \frac{1}{n-1} \left(\sum_{k=0}^{1} d_{i,k} Y_{i+k} \right)^2. \tag{3.25}$$

The coefficients $d_{i,k}$ of differencing scheme follow the assumptions:

$$\sum_{k=0}^{1} d_{i,k} \left(x_{i+k} - x_i\right)^j = \sum_{k=0}^{1} d_{i,k} = 0 \quad \text{as} \quad j = g - 1 = 0, \quad \sum_{k=0}^{1} d_{i,k}^2 = 1 \quad \text{and}$$

$$\sum_{i=1}^{n-1} \sum_{k=0}^{1} d_{i,k}^2 = n - 1. \tag{3.26}$$

The corresponding differencing scheme in a matrix form $\mathbf{D_1}$ of order $(n -$

1) × n, defined as

$$\mathbf{D_1} = \begin{pmatrix} d_{1,0} & d_{1,1} & 0 & \dots & 0 \\ 0 & d_{2,0} & d_{2,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-1,0} & d_{n-1,1} \end{pmatrix}_{(n-1)\times n}$$

and also the estimated residuals

$$\hat{\varepsilon} = \mathbf{D_1}\mathbf{Y} = \begin{bmatrix} d_{1,0} & d_{1,1} & 0 & \dots & 0 \\ 0 & d_{2,0} & d_{2,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-1,0} & d_{n-1,1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

which leads to

$$\hat{\varepsilon} = \begin{pmatrix} d_{1,0}y_1 + d_{1,1}y_2 \\ d_{2,0}y_2 + d_{2,1}y_3 \\ \vdots \\ d_{n-1,0}y_{n-1} + d_{n-1,1}y_n \end{pmatrix}_{(n-1)\times 1}$$

and

$$\hat{\sigma}^2 = \hat{\varepsilon}^T\hat{\varepsilon} = \mathbf{Y}^T\mathbf{Q}\mathbf{Y}$$

where $\mathbf{Q}_{n\times n} = \mathbf{D_1}^T\mathbf{D_1}$.

We should not forget the fact that $\hat{\sigma}_R^2$ is proposed for fixed design data and hence coefficients of differencing scheme $\mathbf{D_1}_{(n-1)\times n}$ which also satisfied (3.26) take the values

$$d_{i,k} = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ \frac{-1}{\sqrt{2}} & k = 1 \end{cases} \tag{3.27}$$

where $i = 1, 2, \dots, n-1$.

Let $\mathbf{m}(\mathbf{x}) = [m(x_1), \ldots, m(x_n)]^T$ then

$$\mathbf{E}[\hat{\varepsilon}^T\hat{\varepsilon}] = \sigma^2 tr[\mathbf{Q}] + \mathbf{m}(\mathbf{x})^T\mathbf{Q}\mathbf{m}(\mathbf{x}). \tag{3.28}$$

From (3.26)

$$tr[\mathbf{Q}] = \sum_{i=1}^{n-1}\sum_{k=0}^{1} d_{ik}^2 = n-1.$$

This is shown below:

$$\begin{aligned}
\mathbf{E}[\hat{\varepsilon}^T\hat{\varepsilon}] &= \frac{1}{2}\sum_{i=1}^{n-1}\mathbf{E}\left[(y_{i+1}-y_i)^2\right] \\
&= \frac{1}{2}\sum_{i=1}^{n-1}\mathbf{E}\left[y_{i+1}^2 - 2y_iy_{i+1} + y_i^2\right] \\
&= \frac{1}{2}\sum_{i=1}^{n-1}\left(\mathbf{E}\left[y_{i+1}^2\right] + \mathbf{E}\left[y_i^2\right]\right) \\
&= \frac{1}{2}\sum_{i=1}^{n-1}\left\{\mathbf{E}\left[m(x_{i+1}) + \varepsilon_{i+1}\right]^2 + \mathbf{E}\left[m(x_i) + \varepsilon_i\right]^2\right\} \\
&= \frac{1}{2}\sum_{i=1}^{n-1}\left\{\mathbf{E}\left[m(x_{i+1})\right]^2 + \mathbf{E}\left[\varepsilon_{i+1}\right]^2 + \mathbf{E}\left[m(x_i)\right]^2 + \mathbf{E}\left[\varepsilon_i\right]^2\right\} \\
&= \frac{1}{2}\sum_{i=1}^{n-1}\left\{2\sigma^2 + \mathbf{E}\left[m(x_{i+1})\right]^2 + \mathbf{E}\left[m(x_i)\right]^2\right\} \\
&= \frac{1}{2}\left\{2(n-1)\sigma^2 + \sum_{i=1}^{n-1}\left(m(x_{i+1})^2 + m(x_i)^2\right)\right\} \\
&= (n-1)\sigma^2 + \left\{\frac{1}{2}\sum_{i=1}^{n-1}\left(m(x_{i+1})^2 + m(x_i)^2\right)\right\}. \qquad (3.29)
\end{aligned}$$

Note that $\hat{\sigma}^2$ is defined as an unbiased residual variance estimator when either $\mathbf{m}(\mathbf{x})^T\mathbf{Q}\mathbf{m}(\mathbf{x}) = \mathbf{0}$ or $\mathbf{m}(\mathbf{x}) = \mathbf{0}$. Using (3.29), it follows that

$$\hat{\sigma}^2 = \frac{\mathbf{E}[\sum_{i=1}^{n-1}\hat{\varepsilon}_i^2]}{(n-1)} = \sigma^2 + \frac{1}{2(n-1)}\left\{\sum_{i=1}^{n-1}\left(m(x_{i+1})^2 + m(x_i)^2\right)\right\}.$$

Rice (1984) in an unpublished report, proposed the uniquely defined estimator of residual variance as

$$\hat{\sigma}_R^2 = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} (y_{i+1} - y_i)^2 . \tag{3.30}$$

The Gasser et. al. (1986) have shown that

$$\mathbf{E}[\hat{\sigma}_R^2] = \sigma^2 + \mathrm{O}\left(n^{-2}\right), \tag{3.31}$$

$$\mathbf{Var}[\hat{\sigma}_R^2] = \frac{\sigma^4 m_4}{(n-1)} + \mathrm{O}\left(n^{-2}\right), \tag{3.32}$$

where $\mathbf{E}[\varepsilon_i^4] = m_4\sigma^4$.

3.3.2 Gasser Sroka and Jennen-Steinmetz Estimator $\hat{\sigma}_{GSJ}^2$

This approach is based on local fitting and has grown out of work on the analysis of growth curves (Gasser et. al., 1986). This estimator relies on a differencing scheme with $g = 2$. As in (3.21), $k = 0, \dots, g \quad \Rightarrow \quad k = 0, 1, 2$ which implies that three observations are being used to form each residual ε_i in this case. As a result, only $n-2$ residuals $\sum_{i=1}^{n-2} \hat{\varepsilon}_i$ are being estimated by this differencing scheme. Under this circumstance, residuals are defined as follows:

$$\hat{\varepsilon}_i = \sum_{k=0}^{2} d_{i,k} Y_{i+k}, \quad i = 1, \dots, n-2. \tag{3.33}$$

Since $g = 2$, assumptions on the the coefficients of differencing scheme as defined in (3.22) are now take the form of:

$$\sum_{k=0}^{2} d_{i,k} \left(x_{i+k} - x_i\right)^j = 0 \quad \text{where} \quad j = 0, 1 \;\Rightarrow\; \sum_{k=0}^{2} d_{i,k} = 0,$$

$$\sum_{k=0}^{2} d_{i,k} \left(x_{i+k} - x_i\right) = 0 \quad \text{and} \quad \sum_{k=0}^{2} d_{i,k}^2 = 1, \quad \text{thus} \quad \sum_{i=1}^{n-2} \sum_{k=0}^{2} d_{i,k}^2 = n-2. \tag{3.34}$$

The above mentioned assumptions lead to the residual variance estimator to be defined as

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^{n-2} \left(\sum_{k=0}^{2} d_{i,k} Y_{i+k} \right)^2 . \qquad (3.35)$$

Here we define the above discussed differencing scheme with an aid of a matrix $\mathbf{D_2}$ where

$$\mathbf{D_2} = \begin{pmatrix} d_{1,0} & d_{1,1} & d_{1,2} & 0 & \cdots & 0 \\ 0 & d_{2,0} & d_{2,1} & d_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-2,0} & d_{n-2,1} & d_{n-2,2} \end{pmatrix}_{(n-2)\times n} .$$

Now (3.33) is equivalent to

$$\hat{\varepsilon} = \mathbf{D_2 Y} = \begin{bmatrix} d_{1,0} & d_{1,1} & d_{1,2} & 0 & \cdots & 0 \\ 0 & d_{2,0} & d_{2,1} & d_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-2,0} & d_{n-2,1} & d_{n-2,2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} .$$

Thus

$$\hat{\varepsilon} = \begin{pmatrix} d_{1,0}y_1 + d_{1,1}y_2 + d_{1,2}y_3 \\ d_{2,0}y_2 + d_{2,1}y_3 + d_{2,2}y_4 \\ \vdots \\ d_{n-2,0}y_{n-2} + d_{n-2,1}y_{n-1} + d_{n-2,2}y_n \end{pmatrix}_{(n-2)\times 1} .$$

Hence, the estimated residuals can be interpreted up to normalization as differences between the data and the line that joins the two neighbouring points. This method is equivalent, up to normalization, to estimating residuals from fitting a least squares line to successive triple of points, an approach that has already been sketched by Rice (1984).

This similar to the pseudo-residuals $\hat{\varepsilon}$ are obtained by taking continuous triples of data points (X_{i-1}, Y_{i-1}), (X_i, Y_i), (X_{i+1}, Y_{i+1}); $i = 2, \ldots, n-1$, joining the two outer observations by a straight line and then computing the difference between this straight line and the middle observation. Let us assume that $Y = mX + c$ is the equation of the straight line joining two outer observations (X_{i-1}, Y_{i-1}) and (X_{i+1}, Y_{i+1}) where

$$m = \frac{Y_{i+1} - Y_{i-1}}{X_{i+1} - X_{i-1}} \quad \text{and} \quad c = \frac{Y_{i-1}X_{i+1} - Y_{i+1}X_{i-1}}{X_{i+1} - X_{i-1}}.$$

Next we estimate the value of Y when $X = X_i$ using the rule $Y = mX + c$. That is $\hat{Y_i} = mX_i + c$. Hence estimated i^{th} residual at the point (X_i, Y_i) is $\hat{\varepsilon}_i = \hat{Y_i} - Y_i$ which leads to

$$\begin{aligned}
\hat{\varepsilon}_i &= \frac{Y_{i+1} - Y_{i-1}}{X_{i+1} - X_{i-1}} X_i + \frac{Y_{i-1}X_{i+1} - Y_{i+1}X_{i-1}}{X_{i+1} - X_{i-1}} - Y_i \\
&= \frac{X_{i+1} - X_i}{X_{i+1} - X_{i-1}} Y_{i-1} + \frac{X_i - X_{i-1}}{X_{i+1} - X_{i-1}} Y_{i+1} - Y_i \\
&= a_i Y_{i-1} + b_i Y_{i+1} - Y_i, \qquad i = 2, \ldots, n-1 \qquad (3.36)
\end{aligned}$$

where $a_i = \frac{(X_{i+1} - X_i)}{(X_{i+1} - X_{i-1})}$ and $b_i = \frac{(X_i - X_{i-1})}{(X_{i+1} - X_{i-1})}$.

Analogous to previous estimator, next step is to find $\mathbf{E}[\hat{\varepsilon}^{\mathbf{T}} \hat{\varepsilon}]$.

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=1}^{n-2} \hat{\varepsilon_i}^2\right] &= \sum_{i=2}^{n-1} \mathbf{E}\left[(a_i Y_{i-1} + b_i Y_{i+1} - Y_i)^2\right] \\
&= \sum_{i=2}^{n-1} \mathbf{E}\left[a_i^2 Y_{i-1}^2 + Y_i^2 + b_i^2 Y_{i+1}^2 + 2a_i b_i Y_{i-1} Y_{i+1}\right] \\
&\quad - \mathbf{E}\left[2a_i Y_{i-1} Y_i\right] - \mathbf{E}\left[2b_i Y_{i+1} Y_i\right] \\
&= \sum_{i=2}^{n-1} \left(a_i^2 \mathbf{E}\left[Y_{i-1}^2\right] + \mathbf{E}\left[Y_i^2\right] + b_i^2 \mathbf{E}\left[Y_{i+1}^2\right]\right) \\
&= \sum_{i=2}^{n-1} a_i^2 \mathbf{E}\left[m(X_{i-1}) + \varepsilon_{i-1}\right]^2 + \mathbf{E}\left[m(X_i) + \varepsilon_i\right]^2 \\
&\qquad + b_i^2 \mathbf{E}\left[m(X_{i+1}) + \varepsilon_{i+1}\right]^2
\end{aligned}$$

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=1}^{n-2} \hat{\varepsilon}_i{}^2\right] &= \sum_{i=2}^{n-1} a_i^2 \mathbf{E}\left[m(X_{i-1})\right]^2 + a_i^2 \mathbf{E}\left[\varepsilon_{i-1}\right]^2 + \mathbf{E}\left[m(X_i)\right]^2 + \mathbf{E}\left[\varepsilon_i\right]^2 \\
&\quad + b_i^2 \mathbf{E}\left[m(X_{i+1})\right]^2 + b_i^2 \mathbf{E}\left[\varepsilon_{i+1}\right]^2 \\
&= \sum_{i=2}^{n-1} \left\{ \left(a_i^2 + b_i^2 + 1\right)\sigma^2 + a_i^2 m(X_{i-1})^2 + m(X_i)^2 + b_i^2 m(X_{i+1})^2 \right\} \\
&= (n-2)\left(a_i^2 + b_i^2 + 1\right)\sigma^2 + \left\{ \sum_{i=1}^{n-2} a_i^2 m(X_{i-1})^2 + m(X_i)^2 \right\} \\
&\quad + \left\{ \sum_{i=1}^{n-2} b_i^2 m(X_{i+1})^2 \right\} \\
&= (n-2)c_i^2\sigma^2 + \left\{ \sum_{i=1}^{n-2} a_i^2 m(X_{i-1})^2 + m(X_i)^2 + b_i^2 m(X_{i+1})^2 \right\},
\end{aligned}$$

where $c_i^2 = (a_i^2 + b_i^2 + 1) = \left(\frac{X_{i+1} - X_i}{X_{i+1} - X_{i-1}}\right)^2 + \left(\frac{X_i - X_{i-1}}{X_{i+1} - X_{i-1}}\right)^2 + 1.$

Gasser et. al. (1986) showed that

$$\frac{\mathbf{E}\left[\sum_{i=1}^{n-2} \frac{\hat{\varepsilon}_i^2}{c_i^2}\right]}{n-2} = \sigma^2 + O(n^{-2}), \tag{3.37}$$

for $m(\cdot)$ twice differentiable and proposed the following residual variance estimator $\hat{\sigma}^2_{GSJ}$ for fixed designed data

$$\begin{aligned}
\hat{\sigma}^2_{GSJ} &= \frac{1}{(n-2)} \sum_{i=1}^{n-2} \frac{\hat{\varepsilon}_i^2}{c_i^2} \\
&= \frac{1}{(n-2)} \sum_{i=1}^{n-2} \frac{\left[a_i Y_{i-1} + b_i Y_{i+1} - Y_i\right]^2}{c_i^2}.
\end{aligned} \tag{3.38}$$

The estimated residuals can also be rewritten as divided differences:

$$\frac{\hat{\varepsilon}_i}{c_i} = \frac{(x_{i+1} - x_i)(x_i - x_{i-1})}{x_{i+1} - x_{i-1}} \left(\frac{Y_{i+1} - Y_i}{x_{i+1} - x_i} - \frac{Y_i - Y_{i-1}}{x_i - x_{i-1}}\right).$$

The underlying idea can be extended in a straightforward manner to fit higher order polynomials and use more neighboring points. The resulting differencing scheme again depends on the design points of explanatory variable X.

For equidistant x_i design points, $a_i = b_i = 1/2$ and hence $c_i^2 = (\frac{1}{2})^2 + (\frac{1}{2})^2 + 1 = \frac{6}{4}$. Now (3.38) becomes to

$$\begin{aligned}\hat{\sigma}^2_{GSJ} &= \frac{1}{(n-2)} \sum_{i=2}^{n-1} \frac{\left(\frac{1}{2}Y_{i-1} + \frac{1}{2}Y_{i+1} - Y_i\right)^2}{\left[(\frac{1}{2})^2 + (\frac{1}{2})^2 + 1\right]} \\ &= \frac{1}{(n-2)} \frac{\sum_{i=2}^{n-1} \left(Y_{i-1} + Y_{i+1} - 2Y_i\right)^2}{4 \times \frac{6}{4}} \\ &= \frac{1}{6(n-2)} \sum_{i=2}^{n-1} \left(Y_{i-1} + Y_{i+1} - 2Y_i\right)^2 . \qquad (3.39)\end{aligned}$$

The estimator given in (3.39) is equivalent to

$$\begin{aligned}\hat{\sigma}^2_{GSJ} &= \frac{1}{(n-2)} \sum_{i=2}^{n-1} \left(\frac{1}{\sqrt{6}}Y_{i-1} + \frac{1}{\sqrt{6}}Y_{i+1} - \frac{2}{\sqrt{6}}Y_i\right)^2 \\ &= \frac{1}{(n-2)} \sum_{i=2}^{n-1} \left(\frac{1}{\sqrt{6}}Y_{i-1} + \frac{1}{\sqrt{6}}Y_{i+1} - \sqrt{\frac{2}{3}}Y_i\right)^2 .\end{aligned}$$

Hence second order difference sequence defined by $\mathbf{D_{2,0}}$ is of the form

$$d_{i,k} = \begin{cases} \frac{1}{\sqrt{6}} & k = 0, 2 \\ -\sqrt{\frac{2}{3}} & k = 1. \end{cases} \qquad (3.40)$$

Note that, in general, above $d_{i,k}$ values (3.40) can be obtained solving the conditions given in (3.34).

The following assumptions have been imposed to obtain asymptotic results of $\hat{\sigma}^2_{GSJ}$:

1. There are no multiple measurements at any design point i.e. $a = x_1 < x_2 < \ldots < x_n = b$; Without loss of generality we take $a = 0$, $b = 1$.

2. $\max|x_i - x_{i-1}| = O(\frac{1}{n})$.

3. ε_is are independent and identically distributed with $\mathbf{E}[\varepsilon_i] = 0$, $\mathbf{Var}[\varepsilon_i] = \sigma^2$ and $\mathbf{E}[\varepsilon_i^4] < \infty$.

4. The function $m(\cdot)$ is continuous.

Define $\mathbf{C}$ to be a $(n-2) \times (n-2)$ diagonal matrix with elements $C_{ii} = c_{i+1}$ and $\mathbf{A}$ to be $(n-2) \times n$ tri-diagonal with elements $A_{ii} = a_{i+1}$, $A_{i,i+1} = -1$, $A_{i,i+2} = b_{i+1}$ and $Q = A^T C^2 A$. Gasser et. al. (1986) have showed that the estimator of σ^2,

$$\hat{\sigma}^2_{GSJ} = (n-2)^{-1} \|C^{-1}\hat{\varepsilon}\|^2$$

and

$$\mathbf{Var}[\hat{\sigma}^2_{GSJ}] = \frac{2\sigma^4}{(n-2)^2} tr(\mathbf{Q}^2) + \frac{\sigma^4}{(n-2)^2}(m_4 - 3)\sum_{i=1}^{n} Q^2_{ii} + m_3 O(\frac{1}{n^3}) + O(\frac{1}{n^5}) \tag{3.41}$$

where $\mathbf{E}[\varepsilon_i^3] = m_3\sigma^3$ and also proved the following theorem.

Theorem 4 *Given Assumptions* $1-4$ *above,* $\hat{\sigma}^2_{GSJ}$ *is a strongly consistent estimate of* σ^2*. Assuming in addition that* $|m(t) - m(s)| \leq const|t-s|^\gamma$*, for* $t, s \in [0,1]$, $\gamma > \frac{1}{4}$, *we obtain*

$$U_n^{-1}(\hat{\sigma}^2_{GSJ} - \sigma^2) \sim N(0, \sigma^4),$$

where

$$U_n = \frac{1}{(n-2)}\left\{2tr(\mathbf{Q}^2) + (m_4 - 3)\sum_{i=1}^{n} Q^2_{ii}\right\}.$$

If the bias term of $\hat{\sigma}^2_{GSJ}$ is disregarded, the finite sample distribution of $\hat{\sigma}^2_{GSJ}$ for normally distributed residuals is shown to be,

$$\hat{\sigma}^2_{GSJ} \sim Y = \frac{\sigma^2}{(n-2)}\sum_{j=2}^{n-2} \lambda_j \chi^2_j \tag{3.42}$$

where χ_j^2 are independent χ^2 variables with one degree of freedom and the λ_j are eignevalues of $\mathbf{Q}$. As in Box (1954), by equating first two moments one can show

$$\hat{\sigma}^2_{GSJ} \sim q\chi^2_v \tag{3.43}$$

where $q = \sigma^2 tr(\mathbf{Q}^2)/(n-2)^2$ and $v = (n-2)^2/tr(\mathbf{Q}^2)$.

Alternatively, one might take $\log[\hat{\sigma}^2_{GSJ}]$ as normally distributed with expectation $\log[\hat{\sigma}^2]$ and variance $2tr(\mathbf{Q}^2)/(n-2)^2$.

3.3.3 Estimator Based on a Covariate-Matched U-Statistic $\hat{\sigma}^2_{MSW}$

For known regression function $m(\cdot)$ the errors $\varepsilon_i = Y_i - m(x_i)$ are observable and the residual variance σ^2 could be estimated by the second sample moment

$$\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2. \tag{3.44}$$

Alternatively sample variance based on errors could be used. The motive of proposed estimator is a sample variance S^2 based on U-statistic. Let $X_1, \ldots, X_n$ be a sample of independent and identically distributed random variables with a distribution function $F(x) \in \mathfrak{F}$ where $\mathfrak{F}$ be the set of all distributions with second moment finite:

$$\mathfrak{F} = \left\{ F : \int_x |x|^2 dF(x) < \infty \right\}.$$

Then we can define the variance functional on $\mathfrak{F}$ by

$$\mathbf{Var}[F] = \int_{x_1}\int_{x_2} \frac{1}{2}(x_1 - x_2)^2 dF(x_1)dF(x_2),$$

which is estimated by the sample variance S^2

$$S^2 = \frac{1}{n(n-1)} \sum_{1 \leq i < j \leq n} \sum_{i \neq j} \frac{1}{2}(X_i - X_j)^2. \tag{3.45}$$

Using (3.45) sample variance of $\varepsilon_1, \ldots, \varepsilon_n$ based on U-statistic can be written as:

$$\hat{\sigma}^2 = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} \left(\varepsilon_i - \varepsilon_j\right)^2 \tag{3.46}$$

which is asymptotically equivalent to (3.44). Motivated by the U-statistic representation in (3.46) Müller, Schick and Wefelmeyer (2003) have introduced the covariate-matched U-statistic type variance estimator $\hat{\sigma}^2_{MSW}$ based on differencing weight scheme D:

$$\hat{\sigma}^2_{MSW} = \frac{1}{n(n-1)} \sum \sum_{i \neq j} \frac{1}{2} [Y_i - Y_j]^2 D_{ij}, \tag{3.47}$$

where the random weights D_{ij} base on the predictor variable X only and will be small or zero if X_i and X_j are not close. Introduced residual variance estimator $\hat{\sigma}^2_{MSW}$ is related to difference-based estimators for both fixed design and random design data. However, there is no direct relationship to (3.21) and (3.23) as in other two residual variance estimators $\hat{\sigma}^2_R$ and $\hat{\sigma}^2_{GSJ}$ discussed in Sections 3.2.1 and 3.2.2 respectively. But the motive is same as the previous two estimators which is estimation of regression function $m(\cdot)$ is not required as for residual variance estimators based on curve fitting. The random weights D_{ij} are defined in a way to guarantee $\hat{\sigma}^2_{MSW}$ behaves asymptotically like the sample second moment based on errors i.e. it has i.i.d representation

$$\hat{\sigma}^2_{MSW} = \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i^2 + o_p(n^{-\frac{1}{2}}).$$

Thus, unlike other residual estimators based on differencing (3.30 and 3.35) $\hat{\sigma}^2_{MSW}$ has been shown efficient for σ^2 (Müller et. al., 2003). Kernel functions have been used for the construction of explicit differencing weights D_{ij}.

The Asymptotic Behaviour of the Covariate-Matched U-Statistic

All the materials presented here are taken from Müller et. al. (2003). The following assumptions are made on the error distribution and the weights.

1. ε is centered and possesses a finite fourth moment:

$$\mathbf{E}[\varepsilon] = 0 \quad \text{and} \quad \mathbf{E}[\varepsilon^4] < \infty.$$

2. The differencing weights D_{ij} depend on the covariates but not on the errors and they are non-negative, symmetric and average to one:

$$D_{ij} \geq 0,\ D_{ij} = D_{ji}; \quad i, j = 1, \ldots, n, \quad i \neq j$$

$$\frac{1}{n(n-1)} \sum\sum_{i \neq j} D_{ij} = 1.$$

The following theorem gives conditions under which $\hat{\sigma}^2_{MSW}$ behaves asymptotically like the average of the squared errors.

Theorem 5 *Suppose that assumptions* 1 *and* 2 *hold, and that*

$$\frac{1}{n(n-1)} \sum\sum_{i \neq j} D_{ij}^2 = o_p(n), \tag{3.48}$$

$$\frac{1}{n} \sum_{i=1}^{n} (\bar{D}_i - 1)^2 = o_p(1), \tag{3.49}$$

$$\frac{1}{n} \sum_{i=1}^{n} \Delta_i^2 = o_p(1), \tag{3.50}$$

$$\frac{1}{n(n-1)} \sum\sum_{i \neq j} (m(x_i) - m(x_j))^2 D_{ij} = o_p(n^{-1/2}). \tag{3.51}$$

Then

$$\hat{\sigma}^2_{MSW} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^2 + o_p(n^{-1/2}), \tag{3.52}$$

where

$$\bar{D}_i = \frac{1}{n-1}\sum_{j:j\neq i} D_{ij}, \quad i = 1, \ldots, n$$

and

$$\Delta_i = \frac{1}{n-1}\sum_{j:j\neq i}(m(x_i) - m(x_j))d_{ij}.$$

Also they have showed that $\sqrt{n}(\hat{\sigma}^2_{MSW} - \sigma^2)$ converges in distribution to a normal random variable with mean zero and variance $\int_x x^4 dF(x) - \sigma^4$. To satisfy above two properties (3.48) and (3.49), the bandwidth h_n will need to satisfy $nh_n \to \infty$ as $n \longrightarrow \infty$.

The following additional assumptions have been imposed in order to construct differencing weights.

3. The covariate X takes values in the interval $[0,1]$ and possesses a density $f(x)$ whose restriction to $[0,1]$ is continuous and positive.

4. The regression function $m(\cdot)$ satisfies the Hölder condition (Müller et. al., 2003):

$$|m(s) - m(t)| \leq H|s-t|^{\beta}, \quad s, t \in [0,1]$$

for some positive constant H and some positive β with $\beta \leq 1$.

Let $K(\cdot)$ be a bounded symmetric density with compact support $[-1,1]$. Then

$$d_{ij} = \frac{1}{2}\left[\frac{1}{\hat{g}_i} + \frac{1}{\hat{g}_j}\right] K_{h_n}(X_i - X_j), \tag{3.53}$$

where

$$K_{h_n}(X) = \frac{1}{h_n}K\left(\frac{x}{h_n}\right) \quad \text{and} \quad \hat{g}_i = \frac{1}{n-1}\sum_{j:j\neq i} K_{h_n}(x_i - x_j).$$

The residual estimator $\hat{\sigma}^2_{MSW}$ corresponding to the above weight scheme is:

$$\hat{\sigma}^2_{MSW} = \frac{1}{n(n-1)} \sum\sum_{i \neq j} \frac{1}{2}[Y_i - Y_j]^2 \frac{1}{2}\left(\frac{1}{\hat{g}_i} + \frac{1}{\hat{g}_j}\right) K_{h_n}(X_i - X_j) \qquad (3.54)$$

and asymptotic distribution of $\hat{\sigma}^2_{MSW}$ can be formed by the following theorem

Theorem 6 *Suppose assumptions 1, 3 and 4 hold and bandwidth h_n satisfy $n^{\frac{1}{2}} h_n^{2\beta} \to 0$ and $nh_n / \log n \to \infty$ then*

$$\hat{\sigma}^2_{MSW} = \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i^2 + o_p(n^{\frac{-1}{2}}). \qquad (3.55)$$

3.4 Comparison of the Different Approaches

We carried out a simulation study to assess bias and standard error (SE) of residual variance estimators based on differences and those based on curve fitting. The following three regression models were considered:

Model A: $Y = m(x) + \varepsilon = 4x + 3 + \varepsilon$,

Model B: $Y = m(x) + \varepsilon = 2\exp(-x^2/0.72) + 3\exp(-(x-1)^2/0.98) + \varepsilon$,

Model C: $Y = m(x) + \varepsilon = \sin^2(0.75x) + 3 + \varepsilon$.

Results were compared for fairly small value of residual variance, $\sigma^2 = 0.05$. If these estimators perform better for small value, they will most likely do for moderate to large values as well. The sample size was varied from very small that is $n = 10$ to large, that is 500. Three types of data design on $[0, 1]$ were studied:

Design 1: Fixed equidistant design: $x_i = i/n$,

Design 2: Fixed non-equidistant design: $x_i \neq i/n$,

Design 3: Random design: x_is are uniformly distributed on $[0, 1]$.

Errors were generated from

(i) symmetric: Normal distribution $\varepsilon \sim N(0, 0.05)$,

(ii) skewed: Exponential distribution $\varepsilon \sim Expo(\lambda)$ and

(iii) robust symmetric: Laplace distribution $\varepsilon \sim Laplace(0, \beta)$.

Values for scale parameters $\lambda\,(= \sigma = \sqrt{0.05})$ and $\beta\,(= \sigma/\sqrt{2} = \sqrt{0.05/2})$ were selected to be consistent with $\mathbf{Var}[\varepsilon] = 0.05$ regardless of distribution function for comparison purpose. To obtain reasonably accurate estimates 15000 replications were considered to be sufficient.

3.4.1 Fixed Design

The data design, distribution used to generate errors, sample size n, average of estimated values of Rice estimator $(\hat{\sigma}^2_R)$, $\overline{\hat{\sigma}^2_R}$ and Gasser et. al. estimator $(\hat{\sigma}^2_{GSJ})$, $\overline{\hat{\sigma}^2_{GSJ}}$ along with their standard errors are reported in Table 3.1, Table 3.2 and Table 3.3 for Models A, B and C respectively. The statistics that appear in the tables are defined below:

- $\overline{\hat{\sigma}^2_R} = \frac{1}{n_{sim}} \sum_{j=1}^{n_{sim}} (\hat{\sigma}^2_R)_j$
- $SE\left(\overline{\hat{\sigma}^2_R}\right) = \left\{ \frac{1}{(n_{sim}-1)n_{sim}} \sum_{j=1}^{n_{sim}} \left((\hat{\sigma}^2_R)_j - \overline{\hat{\sigma}^2_R} \right)^2 \right\}^{1/2}$

where $(\hat{\sigma}^2_R)_j$ is the Rice estimator for j^{th} simulated sample.

- $\overline{\hat{\sigma}^2_{GSJ}} = \frac{1}{n_{sim}} \sum_{j=1}^{n_{sim}} (\hat{\sigma}^2_{GSJ})_j$
- $SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right) = \left\{ \frac{1}{(n_{sim}-1)n_{sim}} \sum_{j=1}^{n_{sim}} \left((\hat{\sigma}^2_{GSJ})_j - \overline{\hat{\sigma}^2_{GSJ}} \right)^2 \right\}^{1/2}$

where $(\hat{\sigma}^2_{GSJ})_j$ is the Gasser et. al. estimator for j^{th} simulated sample.

In each table the first column lists the three different error distributions, the second column lists the sample sizes n and average values along with their standard errors of Rice estimator $\overline{\hat{\sigma}^2_R}$ and Gasser et. al. estimator $\overline{\hat{\sigma}^2_{GSJ}}$ are given in the columns 3-6 and columns 7-10 for both equidistant and non-equidistant designed data respectively.

Table 3.1 shows the performance of each residual estimator when underline regression function is in the linear form. All three error distributions shows similar results in overall. Not only $\overline{\hat{\sigma}^2_R}$ shows significantly higher values for non-equidistant designed data compared to equidistant data but also fails to boost the accuracy of the estimate even for large sample sizes. Conversely, $\overline{\hat{\sigma}^2_R}$ produces more precise estimates for large sample sizes n for equidistant data. On the contrary Gasser et. al. estimator $\overline{\hat{\sigma}^2_{GSJ}}$ performed equally well in both equidistant and non-equidistant designed data even for small sample sizes. As expected, standard errors decrease with increasing sample sizes.

Both Tables 3.2 and 3.3 illustrate the performance of each residual estimator when the underline regression function is in the nonlinear form. The two models A and B are selected in such a way that the shape of the underlying regression functions are completely different from each other to investigate whether the performance of each residual variance estimator heavily depends on the model structure or not. Unlike in *Model A* both residual estimates $\overline{\hat{\sigma}^2_R}$ and $\overline{\hat{\sigma}^2_{GSJ}}$ shows similar behaviour despite the consequences of differences in the data structures as well as shape of the underlying regression function. The $\hat{\sigma}^2_R$ estimator overestimates the residual variance in the case of non-equidistant designed data. But overestimation effect reduces with increasing sample size and hence produces more accurate results for large sample sizes only. However, in the case of equidistant data $\overline{\hat{\sigma}^2_R}$ tends to produce better results even for small sample size and excellent results for medium sample sizes.

Observe that the performance of $\hat{\sigma}^2_{GSJ}$ is slightly different from *Model B* to *Model C*. The values of $\overline{\hat{\sigma}^2_{GSJ}}$ for equidistant data are just about those for non-equidistant in the *Model C*. Nonetheless $\overline{\hat{\sigma}^2_{GSJ}}$ values corresponding to *Model B* are higher for non-equidistant data compared to equidistant data as in $\hat{\sigma}^2_R$ estimation method. Besides the discrepancy becomes worse in particular for small sample size as the $\overline{\hat{\sigma}^2_{GSJ}}$ values are even higher than those of $\overline{\hat{\sigma}^2_R}$ values. However $\hat{\sigma}^2_{GSJ}$ method produces more accurate estimates for moderate sample sizes unlike in $\hat{\sigma}^2_R$ estimation method. As typically expected standard errors decrease with increasing sample sizes.

Table 3.1: Fixed Design Data Using Model A for $\sigma^2 = 0.05$.

		Equidistant				Non-equidistant			
ε	n	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$
Normal	10	.1300	.0003	.0498	.0003	1.8733	.0019	.0499	.0003
	30	.0588	.0001	.0499	.0001	1.8214	.0011	.0499	.0002
	50	.0532	.0001	.0499	.0001	1.8986	.0009	.0501	.0001
	100	.0508	.0001	.0499	.0001	1.4234	.0005	.0500	.0001
	500	.0500	.0000	.0500	.0000	1.3959	.0002	.0500	.0000
Expo	10	.1294	.0004	.0498	.0004	1.8714	.0019	.0494	.0004
	30	.0591	.0002	.0502	.0002	0.9507	.0008	.0504	.0003
	50	.0532	.0002	.0501	.0002	1.5214	.0007	.0499	.0002
	100	.0506	.0001	.0498	.0001	1.2826	.0005	.0498	.0001
	500	.0501	.0001	.0500	.0001	1.4372	.0002	.0500	.0001
Laplace	10	.1300	.0003	.0502	.0004	1.8736	.0019	.0498	.0004
	30	.0590	.0002	.0501	.0002	1.0392	.0008	.0503	.0002
	50	.0529	.0001	.0497	.0002	1.2988	.0007	.0502	.0002
	100	.0508	.0001	.0500	.0001	1.2725	.0005	.0499	.0001
	500	.0500	.0000	.0500	.0000	1.3998	.0002	.0499	.0001

Table 3.2: Fixed Design Data Using Model B for $\sigma^2 = 0.05$.

		Equidistant				Non-equidistant			
ε	n	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$
Normal	10	.0528	.0002	.0503	.0003	.0723	.0003	.0767	.0004
	30	.0503	.0001	.0499	.0001	.0640	.0002	.0552	.0002
	50	.0501	.0001	.0499	.0001	.0746	.0001	.0559	.0001
	100	.0500	.0001	.0499	.0001	.0681	.0001	.0580	.0001
	500	.0500	.0000	.0500	.0000	.0666	.0000	.0568	.0000
Expo	10	.0526	.0004	.0502	.0004	.0716	.0004	.0761	.0005
	30	.0506	.0002	.0503	.0002	.0681	.0003	.0534	.0003
	50	.0502	.0002	.0501	.0002	.0659	.0002	.0580	.0002
	100	.0499	.0001	.0498	.0001	.0704	.0001	.0557	.0001
	500	.0500	.0001	.0500	.0001	.0661	.0001	.0574	.0001
Laplace	10	.0531	.0003	.0507	.0004	.0720	.0004	.0764	.0005
	30	.0504	.0002	.0501	.0002	.0679	.0002	.0533	.0002
	50	.0499	.0001	.0497	.0002	.0657	.0002	.0579	.0002
	100	.0501	.0001	.0500	.0001	.0702	.0001	.0555	.0001
	500	.0500	.0000	.0500	.0000	.0660	.0001	.0570	.0001

Table 3.3: Fixed Design Data Using Model C for $\sigma^2 = 0.05$.

		Equidistant				Non-equidistant			
ε	n	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$	$\overline{\hat{\sigma}^2_R}$	$SE\left(\overline{\hat{\sigma}^2_R}\right)$	$\overline{\hat{\sigma}^2_{GSJ}}$	$SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right)$
Normal	10	.0514	.0002	.0498	.0003	.0803	.0003	.0517	.0003
	30	.0501	.0001	.0499	.0003	.0761	.0002	.0505	.0002
	50	.0500	.0001	.0499	.0001	.0740	.0001	.0506	.0001
	100	.0500	.0001	.0499	.0001	.0698	.0001	.0505	.0001
	500	.0500	.0000	.0500	.0001	.0703	.0000	.0505	.0000
Expo	10	.0511	.0004	.0498	.0004	.0798	.0005	.0513	.0004
	30	.0504	.0002	.0502	.0002	.0646	.0002	.0507	.0003
	50	.0501	.0002	.0501	.0002	.0729	.0002	.0505	.0002
	100	.0499	.0001	.0501	.0001	.0683	.0001	.0502	.0001
	500	.0500	.0001	.0500	.0001	.0709	.0001	.0505	.0001
Laplace	10	.0516	.0003	.0503	.0004	.0802	.0004	.0516	.0004
	30	.0502	.0002	.0501	.0002	.0654	.0002	.0505	.0002
	50	.0498	.0001	.0497	.0002	.0684	.0002	.0508	.0002
	100	.0501	.0001	.0500	.0001	.0660	.0001	.0503	.0001
	500	.0500	.0000	.0500	.0000	.0706	.0001	.0504	.0001

3.4.2 Random Design

For random design data, the distribution used to generate errors, sample size n, value of r, averages of Hall and Marron estimator $(\hat{\sigma}^2_{HM})$, $\overline{\hat{\sigma}^2_{HM}}$ and estimator based on a covariate-matched U-statistic $(\hat{\sigma}^2_{MSW})$, $\overline{\hat{\sigma}^2_{MSW}}$ along with their standard errors which are enclosed in brackets underneath each estimated average residual variance are shown in the tables given below. Table 3.4 gives results for *Model A* while results for *Model B* and *Model C* are given in Tables 3.5 and 3.6 respectively. Standard normal kernel $k(t) = (2\sqrt{\pi})^{-1}\exp(-0.5t^2)$ is used where necessary and several choices of bandwidth $h_n \,(= n^{-r})$ for different values of $r \,(= 0.21, 0.51, 0.76)$ were employed. The statistics that appear in the tables are defined below:

• $\overline{\hat{\sigma}^2_{MSW}} = \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}(\hat{\sigma}^2_{MSW})_j$

• $SE\left(\overline{\hat{\sigma}^2_{MSW}}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}}\sum_{j=1}^{n_{sim}}\left((\hat{\sigma}^2_{MSW})_j - \overline{\hat{\sigma}^2_{MSW}}\right)^2\right\}^{1/2}$

where $(\hat{\sigma}^2_{MSW})_j$ is the residual variance estimator based on covariate-matched U-statistic for j^{th} simulated sample.

• $\overline{\hat{\sigma}^2_{HM}} = \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}(\hat{\sigma}^2_{HM})_j$

• $SE\left(\overline{\hat{\sigma}^2_{HM}}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}}\sum_{j=1}^{n_{sim}}\left((\hat{\sigma}^2_{HM})_j - \overline{\hat{\sigma}^2_{HM}}\right)^2\right\}^{1/2}$

where $(\hat{\sigma}^2_{HM})_j$ is the Hall and Marron residual estimator for j^{th} simulated sample.

In each table, the first column gives the sample size n, the second, value of $r \in (0, 1)$ and columns 3-4, columns 5-6 and the last two columns 7-8 show average values of each estimator and its standard error for three different error distributions. Similar to what we observed in fixed design, the effect of using diverse error distributions on estimation of residual variance estimators is less significant. Because the differences between average residual variance estimates from the three different distributions are very close to each other.

Table 3.4: Random Design Data Using Model A for $\sigma^2 = 0.05$.

		Normal		Exponential		Laplace	
n	r	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$
10	.21	1.112 (.0025)	.962 (.0018)	1.116 (.0026)	.965 (.0019)	1.113 (.0026)	.962 (.0019)
	.51	.618 (.0009)	.370 (.0007)	.617 (.0009)	.371 (.0007)	.618 (.0009)	.372 (.0007)
	.76	.284 (.0005)	.132 (.0004)	.284 (.0006)	.133 (.0005)	.284 (.0006)	.133 (.0004)
30	.21	.965 (.0010)	.760 (.0006)	.965 (.0010)	.760 (.0006)	.965 (.0010)	.761 (.0006)
	.51	.265 (.0002)	.117 (.0002)	.264 (.0003)	.117 (.0003)	.264 (.0002)	.117 (.0002)
	.76	.096 (.0001)	.059 (.0001)	.097 (.0002)	.060 (.0002)	.097 (.0002)	.060 (.0002)
50	.21	.888 (.0006)	.663 (.0003)	.889 (.0007)	.663 (.0004)	.888 (.0007)	.663 3 (.0003)
	.51	.182 (.0001)	.081 (.0001)	.182 (.0002)	.081 (.0002)	.182 (.0002)	.081 (.0001)
	.76	.071 (.0001)	.054 (.0001)	.071 (.0002)	.053 (.0002)	.071 (.0001)	.053 (.0001)
100	.21	.777 (.0003)	.532 (.0002)	.778 (.0004)	.532 (.0002)	.778 (.0003)	.534 (.0002)
	.51	.118 (.0001)	.061 (.0001)	.117 (.0001)	.061 (.0001)	.117 (.0001)	.061 (.0001)
	.76	.058 (.0001)	.051 (.0001)	.057 (.0001)	.051 (.0001)	.057 (.0001)	.051 (.0001)
500	.21	.522 (.0001)	.280 (.0001)	.522 (.0001)	.280 (.0001)	.522 (.0001)	.235 (.0001)
	.51	.064 (.0000)	.051 (.0000)	.064 (.0001)	.051 (.0001)	.064 (.0000)	.034 (.0000)
	.76	.051 (.0000)	.050 (.0000)	.051 (.0001)	.050 (.0001)	.051 (.0000)	.032 (.0000)

Overall averages of both estimators $\overline{\hat{\sigma}^2_{HM}}$ and $\overline{\hat{\sigma}^2_{MSW}}$ turn out to be more precise for higher values of r and n. $\overline{\hat{\sigma}^2_{HM}}$ values are much higher compared to those of $\overline{\hat{\sigma}^2_{MSW}}$ for small to moderate sample sizes in *Model A* which is a linear model. Obviously it achieves high level of accuracy when sample sizes are large and/or higher values of r. However, both averages $\overline{\hat{\sigma}^2_{HM}}$ and $\overline{\hat{\sigma}^2_{MSW}}$ overestimate the residual variance especially for small sample sizes n and r values. Table 3.5 and Table 3.6 demonstrate that the difference between the results of $\overline{\hat{\sigma}^2_{HM}}$ and $\overline{\hat{\sigma}^2_{MSW}}$ for *Model B* and *Model C* is less significant.

Table 3.5: Random Design Data Using Model B for $\sigma^2 = 0.05$.

		Normal		Exponential		Laplace	
n	r	$\hat{\sigma}^2_{HM}$	$\hat{\sigma}^2_{MSW}$	$\hat{\sigma}^2_{HM}$	$\hat{\sigma}^2_{MSW}$	$\hat{\sigma}^2_{HM}$	$\hat{\sigma}^2_{MSW}$
10	.21	.0655 (.0002)	.0647 (.0002)	.0658 (.0004)	.0650 (.0004)	.0660 (.0003)	.0652 (.0003)
	.51	.0622 (.0002)	.0600 (.0002)	.0620 (.0004)	.0598 (.0004)	.0619 (.0003)	.0597 (.0003)
	.76	.0571 (.0002)	.0541 (.0002)	.0569 (.0004)	.0540 (.0004)	.0572 (.0003)	.0542 (.0003)
30	.21	.0647 (.0001)	.0634 (.0001)	.0645 (.0002)	.0632 (.0002)	.0646 (.0002)	.0634 (.0002)
	.51	.0566 (.0001)	.0536 (.0001)	.0563 (.0002)	.0534 (.0002)	.0563 (.0002)	.0534 (.0002)
	.76	.0517 (.0001)	.0505 (.0001)	.0523 (.0002)	.0510 (.0002)	.0521 (.0002)	.0509 (.0002)
50	.21	.0641 (.0001)	.0625 (.0001)	.0644 (.0002)	.0629 (.0002)	.0645 (.0001)	.0630 (.0001)
	.51	.0545 (.0001)	.0519 (.0001)	.0548 (.0002)	.0522 (.0002)	.0547 (.0001)	.0521 (.0001)
	.76	.0508 (.0001)	.0502 (.0001)	.0505 (.0002)	.0498 (.0002)	.0506 (.0001)	.0500 (.0001)
100	.21	.0636 (.0001)	.0617 (.0001)	.0635 (.0001)	.0616 (.0001)	.0635 (.0001)	.0616 (.0001)
	.51	.0525 (.0001)	.0507 (.0001)	.0524 (.0001)	.0507 (.0001)	.0524 (.0001)	.0507 (.0001)
	.76	.0503 (.0001)	.0501 (.0001)	.0502 (.0001)	.0499 (.0001)	.0502 (.0001)	.0499 (.0001)
500	.21	.0610 (.0000)	.0583 (.0000)	.0610 (.0001)	.0583 (.0001)	.0610 (.0000)	.0583 (.0000)
	.51	.0506 (.0000)	.0501 (.0000)	.0505 (.0001)	.0500 (.0001)	.0505 (.0000)	.0500 (.0000)
	.76	.0500 (.0000)	.0500 (.0000)	.0501 (.0001)	.0501 (.0001)	.0501 (.0000)	.0501 (.0000)

Note that both *Model B* and *Model C* are nonlinear and $\hat{\sigma}^2_{MSW}$ estimator produces reasonable estimates even for small sample sizes and r values.

Table 3.6: Random Design Data Using Model C for $\sigma^2 = 0.05$.

		Normal		Exponential		Laplace	
n	r	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$	$\overline{\hat{\sigma}^2_{HM}}$	$\overline{\hat{\sigma}^2_{MSW}}$
10	.21	.0663 (.0002)	.0641 (.0002)	.0671 (.0004)	.0648 (.0004)	.0666 (.0003)	.0644 (.0003)
	.51	.0589 (.0002)	.0552 (.0002)	.0587 (.0004)	.0550 (.0004)	.0587 (.0003)	.0550 (.0003)
	.76	.0537 (.0002)	.0513 (.0002)	.0536 (.0004)	.0512 (.0004)	.0537 (.0003)	.0513 (.0003)
30	.21	.0641 (.0001)	.0611 (.0001)	.0639 (.0002)	.0609 (.0002)	.0640 (.0002)	.0610 (.0002)
	.51	.0534 (.0001)	.0511 (.0001)	.0532 (.0002)	.0509 (.0002)	.0532 (.0002)	.0509 (.0002)
	.76	.0507 (.0001)	.0501 (.0001)	.0513 (.0002)	.0507 (.0002)	.0512 (.0002)	.0506 (.0002)
50	.21	.0629 (.0001)	.0596 (.0001)	.0632 (.0002)	.0598 (.0002)	.0632 (.0001)	.0599 (.0001)
	.51	.0522 (.0001)	.0506 (.0001)	.0525 (.0002)	.0509 (.0002)	.0524 (.0001)	.0508 (.0001)
	.76	.0503 (.0001)	.0500 (.0001)	.0500 (.0002)	.0497 (.0002)	.0501 (.0001)	.0498 (.0001)
100	.21	.0613 (.0001)	.0577 (.0001)	.0612 (.0001)	.0576 (.0001)	.0614 (.0001)	.0577 (.0001)
	.51	.0511 (.0001)	.0502 (.0001)	.0511 (.0001)	.0502 (.0001)	.0511 (.0001)	.0501 (.0001)
	.76	.0501 (.0001)	.0500 (.0001)	.0500 (.0001)	.0499 (.0001)	.0500 (.0001)	.0499 (.0001)
500	.21	.0575 (.0001)	.0538 (.0000)	.0575 (.0000)	.0538 (.0001)	.0575 (.0000)	.0538 (.0000)
	.51	.0502 (.0000)	.0500 (.0000)	.0501 (.0001)	.0499 (.0001)	.0502 (.0000)	.0500 (.0000)
	.76	.0500 (.0000)	.0500 (.0000)	.0501 (.0001)	.0501 (.0001)	.0501 (.0000)	.0501 (.0000)

Final Remarks:

When the fixed non-equidistant designed data is being used to estimate residual variance, Gasser et. al. $\hat{\sigma}^2_{GSJ}$ estimator behaves somewhat similar to the Rice estimator $\hat{\sigma}^2_R$. However we should not overlook the fact that $\hat{\sigma}^2_{GSJ}$ is less amount of overestimated compared to $\hat{\sigma}^2_R$. Besides the precision of $\hat{\sigma}^2_{GSJ}$ estimator was to a great extent higher compared to $\hat{\sigma}^2_R$ estimator as for the fixed equidistant design.

The Hall and Marron estimator $\hat{\sigma}^2_{HM}$ behaves well but may be somewhat less accurate for small to moderate sample sizes, r values and when the underline function is linear which is less important as we focus on nonlinear functions. Consequently, the $\hat{\sigma}^2_{MSW}$ estimator is a reasonable compromise.

Therefore the Gasser et. al. estimator $\hat{\sigma}^2_{GSJ}$ is more suitable for fixed designed data whereas $\hat{\sigma}^2_{MSW}$ performs well in the case of random designed data.

Chapter 4

Fixed-width Confidence Bands for Nonparametric Kernel Regression

4.1 Introduction

Methods for obtaining confidence bands for $m(x)$ at a given point $x = x_0$ can be found in Hall and Titterington (1988), Eubank and Speckman (1993) and Diebolt (1995). The most widely used confidence band for $m(x)$ is based on the theorem of Bickel and Rosenblatt (1973) for kernel estimation of a density function. Bias-corrected confidence bands for general nonparametric regression models are considered by Xia (1998). In principle, confidence intervals can be obtained from asymptotic normality results for a kernel estimator of $m(x)$. However, the limiting bias and variance depend on unknown quantities which have to be estimated consistently in order to construct asymptotic confidence intervals.

This chapter proposes new classes of sequential stopping rules for final sample size N in constructing fixed-width confidence intervals for the unknown regression function $m(x)$ at a point $x = x_0$ with a preassigned level of accuracy. We employ Stein's two-stage sequential procedure, modified two-stage sequential procedure and purely sequential procedure for fixed equidistant design and random design data. These sequential procedures are extensively applied to the problem of constructing fixed-width confidence interval for an unknown density estimation $f(x)$ at a known point x_0 using nonparametric kernel density estimation. Wagman and Davis (1975) presented some sequential procedures which satisfy certain error control. Stute (1983) considered similar type of problem and presented asymptotic results for the stopping time. Isogai (1987) considered procedure for construction of confidence interval for a nonparametric density function at a given point based on recursive estimation of the kernel function. He also investigated the asymptotic consistency of the estimated density function. Kundu and Martinsek (1994) and Kundu (1994) looked at the problem of estimating $f(x)$ via Stein's two-stage and purely sequential procedures.

Motivated by existing research on sequential nonparametric kernel density estimation we extend the use of sequential procedures to nonparametric kernel regression estimation. Fixed-width confidence intervals are developed using both Nadaraya–Watson and local linear kernel estimators of nonparametric regression with data-driven bandwidths. The sample size is optimized using the purely, two-stage and modified two-stage sequential procedures together with asymptotic properties of the Nadaraya–Watson and local linear estimators. In contrast to fixed sampling procedures, sequential procedures draw observations one at a time or in batches to allow data analyst to look at an appropriate stopping time along with an appropriate statistical decision or to continue sampling. Sequential analysis, in general, comes in handy when the experimenter's objective is to control the error of estimation at some preassigned level. Whether one wants to estimate $m(x)$ at one single point x_0 or for all $x \in \mathbb{R}$, depending on the specific goal and error criterion, one would like to determine the sample size n in an optimal fashion. That is, in order to have the error controlled at a preassigned level, sample size has to be adaptively estimated in the process by a positive integer valued random variable N where the event $[N = n]$ will depend only on $(X_1, Y_1), \ldots, (X_n, Y_n)$ for all $n \geq 1$. Finally $m(x_0)$ is estimated by $\hat{m}_{h_N}(x_0)$ constructed from $(X_1, Y_1), \ldots, (X_N, Y_N)$.

As given in Section 2.2, the kernel estimate of unknown regression function $m(x)$ at a given point $x = x_0$ can be written as

$$\hat{m}_{h_n,q}(x_0) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} y_i \tag{4.1}$$

where $q = NW$ for Nadaraya–Watson estimator as in (2.18), $q = LL$ for local linear estimator as in (2.19). We describe an estimation procedure of nonparametric regression model at a given point $m(x_0)$ by some appropriately constructed fixed-width $(2d)$ confidence interval I_N with the coverage probability at least $1 - \alpha$, such that

$$Pr\left(m(x_0) \in I_N = [\hat{m}_{h_n,q}(x_0) \pm d]\right) \geq 1 - \alpha. \tag{4.2}$$

Here, $d(> 0)$ and $\alpha \in (0, 1)$ are two preassigned values. There are many difficulties with finding a good solution for an optimal sample size n_{opt} from the inequality (4.2). Firstly, we must derive the distribution of $|\hat{m}_{h_n,q}(x_0) - m(x_0)|$; secondly, the practical implementation of the kernel regression estimator requires the specification of the bandwidth h_n.

In this chapter we follow the sequential procedures in light of a optimal sample size calculations given in Section 4.2. The construction of fixed-width confidence intervals for kernel regression estimators for different coverage probabilities using fixed equidistant design data and random design data are presented in Section 4.3 and Section 4.4 respectively. The asymptotic properties and performance of proposed sequential stopping rules are assessed and compared through their coverage accuracy using a large scale simulation study and results are given in Section 4.5.

4.2 Fixed-Width Confidence Interval

Given $d(>0)$, $\alpha \in (0,1)$ and $N = n$ from (4.2), we wish to claim

$$Pr\left(\hat{m}_{h_n,q}(x_0) - d < m(x_0) < \hat{m}_{h_n,q}(x_0) + d\right) \geq 1 - \alpha, \tag{4.3}$$

which can rewrite as

$$Pr\left(\frac{-d}{\sqrt{\mathbf{Var}\left[\hat{m}_{q,h_n}(x_0)\right]}} < \frac{\hat{m}_{h_n,q}(x_0) - m(x_0)}{\sqrt{\mathbf{Var}\left[\hat{m}_{h_n,q}(x_0)\right]}} < \frac{d}{\sqrt{\mathbf{Var}\left[\hat{m}_{h_n,q}(x_0)\right]}}\right) \geq 1 - \alpha.$$

From Sections 2.3 and 2.4,

$$Bias_q = \mathbf{E}[\hat{m}_{h_n,q}(x)] - m(x)$$

where

$$Bias_q = \begin{cases} \frac{h_n^2}{2} m''(x) \int_u u^2 K(u) du + \frac{h_n^2 \mu_2(K) m'(x) f'(x)}{f(x)} + o(h_n^2) & \text{if q=NW} \\ \frac{h_n^2}{2} m''(x) \int_u u^2 K(u) du + o(h_n^2) & \text{if q=LL} \end{cases}$$

and

$$\mathbf{Var}[\hat{m}_{h_n,q}(x)] = \frac{B\sigma^2}{nh_n f(x)} + o\left\{(nh_n)^{-1}\right\}$$

where x is held fixed and $B = \int_u K^2(u) du$.

However, as discussed in Sections 2.3 and 2.4, since $h_n \to 0$ and $nh_n \to \infty$ as $n \to \infty$,

$$\mathbf{E}\left[\hat{m}_{h_n,q}(x)\right] \to m(x)$$

and

$$\mathbf{Var}\left[\hat{m}_{h_n,q}(x)\right] \to \frac{B\sigma^2}{nh_n f(x)}$$

where x is held fixed.

In order to determine the optimal sample size n_{opt} or the stopping rule that would attain (4.3) we use the following theorem introduced by Schuster (1972)

Theorem 7 *Let kernel function $K(\cdot)$ satisfy $\int_u uK(u)du = 0$, $\int_u u^2K(u)du \leq \infty$, $K(u)$ and $|uK(u)|$ are bounded, h_n is such that $\lim nh_n^3 = \infty$ and $\lim nh_n^5 = 0$. Suppose $x_1, \ldots, x_k$ are distinct points and $g(x_i) > 0$ for $i = 1, 2, \ldots, k$. If $\boldsymbol{E}[Y^3]$ is finite and if g', w', v', g'' and w'' exist and bounded where $g(x) = \int_y f(x,y)dy$, $w(x) = \int_y yf(x,y)dy$ and $v(x) = \int_y y^2 f(x,y)dy$ respectively, then*

$$\sqrt{nh_n}\,(\hat{m}_{h_n}(x_1) - m(x_1), \ldots, \hat{m}_{h_n}(x_k) - m(x_k)) \xrightarrow{d} \mathbf{Z}^*$$

where $\mathbf{Z}^*$ is multivariate normal with mean vector $\mathbf{0}$ and diagonal covariance matrix $\mathbf{C} = [\mathbf{C}_{ii}]$ where $C_{ii} = \boldsymbol{Var}[Y|X = x_i]\int_u K^2(u)du/g(x_i)$ $(i = 1, \ldots, k)$.

The complete proof of the above theorem is given in Schuster (1972). In the univariate case we have

$$\sqrt{nh_n}\left[\hat{m}_{h_n,q}(x_0) - m(x_0)\right] \sim N\left(0, \frac{B\sigma^2}{f(x)}\right). \tag{4.4}$$

Therefore

$$\frac{\hat{m}_{h_n,q}(x_0) - m(x_0)}{\sigma\sqrt{\frac{B}{f(x)nh_n}}} \sim N(0,1) \qquad \text{as } n \to \infty. \tag{4.5}$$

Since half width of the interval $\hat{m}_{h_n,q}(x_0) - m(x_0)$ is fixed for a given preassigned value $d\,(> 0)$, we require

$$Pr\left(\left|\frac{\hat{m}_{h_n,q}(x_0) - m(x_0)}{\sqrt{\mathbf{Var}(\hat{m}_{h_n,q}(x_0))}}\right| < \frac{d}{\sqrt{\mathbf{Var}(\hat{m}_{h_n,q}(x_0))}}\right) \geq 1 - \alpha. \tag{4.6}$$

If Z is a standard normal random variable then

$$Pr\left(-z_{\alpha/2} < Z < z_{\alpha/2}\right) = 1 - \alpha$$

where $z_{\alpha/2}$ is given by $\Phi(z_{\alpha/2}) = 1 - \frac{1}{2}\alpha$ and $\Phi(\cdot)$ is the standard normal cumulative distribution function.

From (4.5), (4.6) can be achieved asymptotically by

$$\left|\frac{d}{\sqrt{\mathbf{Var}\left[\hat{m}_{h_n,q}(x_0)\right]}}\right| \geq z_{\alpha/2}$$

or

$$\frac{d^2}{\mathbf{Var}\left[\hat{m}_{h_n,q}(x_0)\right]} \geq z^2_{\frac{\alpha}{2}}.$$

Since

$$\mathbf{Var}\left[\hat{m}_{h_n,q}(x_0)\right] \approx \frac{B\sigma^2}{nh_n f(x)},$$

we have

$$\frac{d^2}{B\sigma^2/nh_n f(x)} \geq z^2_{\frac{\alpha}{2}}$$

that is

$$\frac{nh_n d^2}{B\sigma^2(f(x))^{-1}} \geq z^2_{\alpha/2}. \tag{4.7}$$

As explained in Section 2.6, we take bandwidth $h_n = n^{-r}$ where $0 < r < 1$. However, to apply above theorem, h_n is selected in such a way that $\lim nh_n^3 \to \infty$ and $\lim nh_n^5 = 0$ as $n \to \infty$. This leads to $n^{1-5r} \to 0$ and $n^{1-3r} \to \infty$ as $n \to \infty$. Hence, $1 - 5r < 0$ and $1 - 3r > 0$. Thus we take bandwidth $h_n = n^{-r}$ where $\frac{1}{5} < r < \frac{1}{3}$.

Therefore, from (4.7) we have

$$\frac{n^{1-r}}{B\sigma^2(d^2 f(x))^{-1}} \geq z^2_{\alpha/2}$$

which leads to

$$n \geq \left[\frac{z^2_{\alpha/2} B\sigma^2}{d^2 f(x)}\right]^{\frac{1}{1-r}} = n_{opt}(d) \equiv n_{opt}, \qquad \text{say.} \tag{4.8}$$

The optimal sample size n_{opt} for the required fixed-width confidence interval with confidence coefficient $1 - \alpha$ given in (4.8) can be computed only if σ^2 and $f(x)$ are known. However, the optimal value obtained from (4.8) is still an approximation since we are using asymptotic theory.

4.3 Fixed Equidistant Design Data

Let $x_1, \ldots, x_{n_0}$ be a sample of fixed equidistant design points of size n_0. As explained in Section 2.1 observations take the form of $\{x_i = \frac{i}{n_0}\}_{i=1}^{n_0}$. We know that most of the multistage sequential procedures start with an initial sample size and then continue sampling until a certain condition is satisfied. If we apply the concept of multi-stage sequential procedures to equidistant fixed design points then initial sample size is n_0 and final sample size is N. However, the final sample consists of all the observations from pilot sample and extra $N - n_0$ observations taken at subsequent stages. To comply with the structural conditions inherited in these equidistant design points, the final sample size N has to be a multiple of pilot sample size n_0, otherwise, we can not continue using observations from initial sample. That is $N = Tn_0$ where T is a positive integer.

For example, if initial sample size is $n_0 = 5$ then $x_i = \frac{1}{5}, \frac{2}{5}, \ldots, \frac{5}{5}$; $i = 1, \ldots, 5$ and if corresponding final sample size is $N = 8$ then observations of final sample $x_i = \frac{1}{8}, \frac{2}{8}, \ldots, \frac{8}{8};\ i = 1, \ldots, 8$. Except one x value which is $x = 1$ all other x values in the final sample is completely different from observations which are already selected in pilot sample. On the other hand, if final sample size is $N = 10$ which is a multiple of pilot sample size $n_0 = 5$ then all the x values from pilot sample are included in the final sample as $\frac{2}{10}, \frac{4}{10}, \ldots, \frac{10}{10}$ correspond to $\frac{1}{5}, \frac{2}{5}, \ldots, \frac{5}{5}$.

In general, observations from independent variable X in the final sample must be in the form of $x_i = \frac{i}{n_0 T};\ i = 1, \ldots, n_0 T$ where already selected x values $\{x_i\}_{i=1}^{n_0}$ result when $i = T, 2T, 3T, \ldots, nT$. This will limit the use of most of the available sequential procedures to two-stage and modified two-stage sequential procedures. This is because in these two sequential procedures, in the first stage, we take an initial sample of size n_0 and then use this pilot sample to derive final sample size N. Hence, additional sample of $N - n_0$ is required in the second stage of sampling procedure. As a result the only way we can exploit these two procedures is to ensure that the final sample size is a multiple of pilot sample size. This is not a problem because even final sample is not exactly divisible by pilot sample size we can rounded up to the nearest integer and then multiply that number by pilot sample size in order to obtain final sample size. However, this rounding up will result in oversampling. Besides we can not overlook the fact of oversampling which is inherent to two-stage sequential procedure. Consequently, even if we can apply two-stage sequential procedure for fixed equidistant data, that will worsen the oversampling problem due to the effects of proposed rounding up sample size calculation procedure.

Modified two-stage sequential procedure was proposed as an alternative to two-stage sequential procedure by Mukhopadhyay (1980) which has more attractive properties, most importantly it minimizes oversampling problem to a great extent. Nevertheless the use of modified two-stage procedure in this case will still result in oversampling to a certain extent due to the rounding up effect. This will be discussed in detail later in this chapter.

4.3.1 Two-Stage Sequential Procedure

Two-stage sampling procedure was first developed by Stein (1945). The goal of a two-stage sequential procedure is to determine an optimum sample size under a specified stopping rule and an optimum decision rule that would meet certain desirable criterion which is prompted by the nature of problem. Here we relate the concept of two-stage procedure given in Ghosh et. al. (1997) to develop stopping rules for sampling which is convenient when making an estimation of nonparametric kernel regression function using a fixed-width confidence intervals with preassigned coverage probability. Note that the following results are valid for both Nadaraya–Watson and local linear estimators and we take $\hat{m}(x_0) = \hat{m}_{h_n,q}(x_0)$ for simplicity.

Let $\{(x_1, Y_1), \ldots, (x_{n_0}, Y_{n_0})\}$ be the initial sample where Y_i is the observed value of $m(x_i)$ at $x_i = i/n_0$ for $i = 1, \ldots, n_0$. Note that for fixed equidistant design data $f(x) = 1.0$. Hence optimal sample size n_{opt} given in (4.8) reduces to

$$n_{opt} = \left[\frac{z^2_{\alpha/2} B \sigma^2}{d^2} \right]^{\frac{1}{1-r}}. \tag{4.9}$$

In general residual variance σ^2 is unknown and hence an estimator is required. Since we consider fixed equidistant design data points, we will use the Gasser et. al. estimator $\hat{\sigma}^2_{GSJ}$ as discussed in Section 3.3.2:

$$\hat{\sigma}^2 = \hat{\sigma}^2_{\mathrm{GSJ}} = \frac{1}{(n-2)} \sum_{i=2}^{n-1} \frac{[a_i Y_{i-1} + b_i Y_{i+1} - Y_i]^2}{(a_i^2 + b_i^2 + 1)}.$$

From Section 3.3.2, we know that

$$V = \frac{1}{\eta}\hat{\sigma}^2_{GSJ} \sim \chi^2_\nu$$

where

$$\nu = (n-2)^2/tr(\mathbf{Q}^2) \quad \text{and} \quad \eta = \sigma^2 tr(Q^2)/(n-2)^2$$

as in (3.43). Also from (4.5),

$$Z = \frac{\hat{m}(x_0) - m(x_0)}{\sqrt{\frac{B\sigma^2}{nh_n}}},$$

where Z is a standard normal random variable. Since V is a χ^2_ν random variable and also Z and V are independent,

$$T = \frac{Z}{\sqrt{\frac{V}{\nu}}} = \frac{\frac{\hat{m}(x_0)-m(x_0)}{\sqrt{\frac{B\sigma^2}{nh_n}}}}{\sqrt{\frac{\frac{1}{\eta}\hat{\sigma}^2_{GSJ}}{\nu}}} \sim t_\nu. \tag{4.10}$$

According to (3.43) given in Section 3.3.2

$$\frac{1}{\eta\nu} = \frac{1}{\sigma^2}.$$

This simplifies T in (4.10) to

$$T = \frac{\frac{\hat{m}(x_0)-m(x_0)}{\sqrt{\frac{B\sigma^2}{nh_n}}}}{\sqrt{\frac{\hat{\sigma}^2_{GSJ}}{\sigma^2}}} \sim t_\nu$$

which leads to

$$T = \frac{\hat{m}(x_0) - m(x_0)}{\sqrt{\frac{B\sigma^2}{nh_n}}} \times \sqrt{\frac{\sigma^2}{\hat{\sigma}^2_{GSJ}}} \sim t_\nu.$$

That is,

$$\frac{\hat{m}(x_0) - m(x_0)}{\sqrt{\frac{B\hat{\sigma}^2_{GSJ}}{nh_n}}} \sim t_\nu. \tag{4.11}$$

From (4.2),

$$Pr[|m(x_0) - \hat{m}(x_0)| < d] = 1 - \alpha.$$

Therefore using (4.11)

$$Pr\left(\left|\frac{\hat{m}(x_0) - m(x_0)}{\sqrt{\frac{B\hat{\sigma}^2_{GSJ}}{nh_n}}}\right| < t_{\nu,\alpha/2}\right) = 1 - \alpha.$$

Following the same steps as in Section 4.2

$$d \geq t_{\nu,\alpha/2}\sqrt{\frac{B\hat{\sigma}^2_{GSJ}}{nh_n}}$$

$$n \geq \left[\frac{t^2_{\nu,\alpha/2}B\hat{\sigma}^2_{GSJ}}{d^2}\right]^{\frac{1}{1-r}}. \tag{4.12}$$

From (4.12), we propose the following rule for N_1:

$$N_1 = \max\left\{n_0, \left\lfloor\left(\frac{Bt^2_{\alpha/2,\nu}\hat{\sigma}^2_{GSJ}}{d^2}\right)^{\frac{1}{1-r}}\right\rfloor + 1\right\} \tag{4.13}$$

where $t_{\alpha/2,\nu}$ is the upper $\alpha/2$ of the t-distribution with ν degrees of freedom from (3.43), $\lfloor n \rfloor$ refers to the floor function i.e. rounded up integer value of n and $r \in (\frac{1}{5}, \frac{1}{3})$. In order to comply with the fixed equidistant data design and to continually use the observed data in the initial sample, we take the final sample size as

$$N = n_0T \tag{4.14}$$

where T is a positive integer given by

$$T = \left\lfloor\frac{N_1}{n_0}\right\rfloor = \max\left\{1, \left\lfloor\frac{1}{n_0}\left\{Bt^2_{\alpha/2,\nu}\hat{\sigma}^2_{GSJ}d^{-2}\right\}^{\frac{1}{1-r}}\right\rfloor\right\} \tag{4.15}$$

and $N \geq N_1$. Clearly if $T = 1$, no additional observations are required in the second stage and $N = n_0$. However, if $T > 1$ we take extra sample of size $N - n_0 = n_0(T-1)$ in the second stage with

$$x_i = \frac{i}{n_0T} \quad \text{for} \quad i = 1, \ldots, (n_0T - 1) \quad \text{and} \quad i \neq T, 2T, \ldots, n_0T.$$

Note that the initial sample data corresponds to (x_i, Y_i) for $i = T, 2T, \ldots, n_0T$. In the application of above stopping rule (4.14), it is important to select the best available values for the design constants r and $n_0 \geq 2$ for fixed predesigned values of d and α. The value for r is selected as explained in Section 4.2 and pilot sample size is a subjective choice. We start experiment with a sensible sample size or possibly try different sample sizes and pick the one which gives best results. Finally we use the sample $\{(x_1, Y_1), \ldots, (x_N, Y_N)\}$ with $x_i = i/N$ to compute Nadaraya–Watson (2.18) and local linear (2.19) estimates for $m(x_0)$ and construct the confidence band given by (4.2).

4.3.2 Modified Two-Stage Sequential Procedure

It is well known fact that Stein's procedure oversamples. However, a significant reduction of this oversampling problem can be achieved by using the modified two-stage procedure introduced in Mukhopadhyay (1980). According to (4.8), for very small values of d, large sample sizes result at the second stage of the two-stage procedure regardless of the fixed sample size n_0. Furthermore, large sample sizes result at the second stage when n_0 is small since $\hat{\sigma}_{GSJ,n_0}$ may be large compared to σ^2. However, we can afford to start with a larger sample size when d is preassigned a smaller value. This procedure allows initial sample size n_0 to be large when d is small. Mukhopadhyay (1980) gave a specific choice for the initial sample size in the case of density estimation. Our modified two-stage procedure is proposed as follows:

Step 1: Compute an initial sample of size n_0 by

$$n_0 = \max\left\{2, \left\lfloor \left(\frac{Z_{\alpha/2}}{d}\right)^{2\eta} \right\rfloor + 1\right\} \tag{4.16}$$

where $0 < \eta < 1$ and simulations have been used to find suitable value for η. Next select $\{(x_1, y_1), \ldots, (x_{n_0}, y_{n_0})\}$ where y_i is the observed value of $m(x_i)$

at $x_i = i/n_0$ for $i = 1, \ldots, n_0$.

Step 2: Calculate

$$T = \left\lfloor \frac{N_1}{n_0} \right\rfloor = \max \left\{ 1, \left\lfloor \frac{1}{n_0} \left\{ B t^2_{\alpha/2,\nu} \hat{\sigma}^2_{GSJ} d^{-2} \right\}^{\frac{1}{1-r}} \right\rfloor \right\}. \tag{4.17}$$

Notice that (4.17) is merely (4.13) divided by n_0.

Step 3: If $T = 1$, no observations are required in the second stage and the process terminates. Final sample size N is equivalent to initial sample size n_0 i.e. $N = n_0$. Go to step 5.

Step 4: If $T > 1$, then final sample size N equals to $n_0 T$ i.e. $N = n_0 T$. Hence, take an extra sample of size $N - n_0 = n_0 T - n_0 = n_0(T - 1)$ in the second stage with design points

$$x_i = \frac{i}{n_0 T}$$

for $i = 1, \ldots, (n_0 T - 1)$ and $i \neq T, 2T, \ldots, n_0 T$. Go to step 5.

Step 5: Use the final sample of size N i.e. $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ with $x_i = \frac{i}{N}$; $i = 1, \ldots, N$ to compute Nadaraya–Watson estimate $\hat{m}_{h_N,NW}(x_0)$ and local linear estimate $\hat{m}_{h_N,LL}(x_0)$ for $m(x_0)$ and construct the confidence band given by (4.2).

4.4 Random Design Data

Suppose that $(X_1, Y_1), \ldots, (X_n, Y_n)$ is a sequence of independent and identically distributed (i.i.d.) bivariate random variables having an unknown continuous probability distribution function (pdf) $f_{XY}(x,y)$ and for simplicity we assume that $X_i \in (0,1)$ with an known/unknown pdf $f(x)$. In the case of unknown $f(x)$ Kernel density estimation is employed to estimate $f(x)$.

4.4.1 Purely Sequential Procedure

In general σ^2 in (4.8) is unknown and purely sequential procedure suggests to substitute the variance parameter σ^2 by a estimator $\hat{\sigma}_n^2$ based on a sample $(X_1,Y_1),\ldots,(X_n,Y_n)$ of size n. Since a regression model with random design data points is considered, as explained in Section 3.3 we use the $\hat{\sigma}^2_{MSW}$ residual variance estimator:

$$\hat{\sigma}^2_{MSW} = \frac{\sum\sum_{i\neq j} \frac{1}{2}\left(Y_i - Y_j\right)^2 \frac{1}{2}\left(\frac{1}{\hat{g}_i - \hat{g}_j}\right) K\left(\frac{X_i - X_j}{h_n}\right)}{n(n-1)}.$$

Hence from the optimal sample size n_{opt} given in (4.8) we continue sampling until

$$n \geq \left\{ \frac{z^2_{\alpha/2} B \hat{\sigma}^2_{MSW,n}}{d^2 f(x)} \right\}^{\frac{1}{1-r}} \tag{4.18}$$

where $\hat{\sigma}^2_{MSW,n}$ is a residual variance estimator $\hat{\sigma}^2_{MSW}$ based on sample size n. Thus by taking $n = n_0$ we propose the following stopping rule for purely sequential procedure which is given by

$$N = \max\left\{ n, \left\lfloor \left\{ \frac{z^2_{\alpha/2} B \hat{\sigma}^2_{MSW,n}}{d^2 f(x)} \right\}^{\frac{1}{1-r}} \right\rfloor + 1 \right\} \tag{4.19}$$

where $\lfloor n \rfloor$ refers to the floor function and $r \in (\frac{1}{5}, \frac{1}{3})$.

In purely sequential procedure we take one observation at a time until the condition given in (4.19) is satisfied and steps involved in this procedure are as follows:

Step 1: Take an initial sample of size n_0, that is select $\{(X_1,Y_1),\ldots,$

$(X_{n_0}, Y_{n_0})\}$; $n_0 \geq 2$ where Y_i is the observed value of $m(X_i)$ at X_i for $i = 1, \ldots, n_0$.

Step 2: Now let $n = n_0$ and calculate

$$\left\{ \frac{Z^2_{\alpha/2} B \hat{\sigma}^2_{MSW,n}}{d^2 f(x)} \right\}^{\frac{1}{1-r}}.$$

Step 3: Compare n with $\left\{ \frac{z^2_{\alpha/2} B \hat{\sigma}^2_{MSW,n}}{d^2 f(x)} \right\}^{\frac{1}{1-r}}$. If

$$n \geq \left\{ \frac{z^2_{\alpha/2} B \hat{\sigma}^2_{MSW,n}}{d^2 f(x)} \right\}^{\frac{1}{1-r}}$$

then final sample size N equals to n i.e. $N = n$ and no more observations are required furthermore and the process terminates. Go to step 5. Otherwise go to step 4.

Step 4: Increase sample size by one that is, new sample size is $n + 1$ and set $n = n + 1$. Go to step 3.

Step 5: Use the sample $\{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ to compute $\hat{m}_{h_N,NW}(x_0)$ and $\hat{m}_{h_N,LL}(x_0)$ estimates for $m(x_0)$ and hence construct the confidence band given by (4.2).

4.4.2 Two-stage Sequential Procedure

The above purely sequential procedure involves a lot of computational effort. Stein (1945) two-stage sequential sampling procedure explained in Section 4.3.1 requires only two sampling operations. However, it turned out that this two-stage procedure is less efficient than the purely sequential procedure. Note that it is not possible to use the stopping rule developed in Section 4.3 here as it was uniquely defined for fixed equidistant design points and more importantly it completely based on distribution of particular residual variance estimator $\hat{\sigma}^2$. Nevertheless, we can develop a new stopping rule using two-stage sequential procedure for random design data points by following the similar steps as in Section 4.3.1.

Using the asymptotic normality results in Theorem 7 for univariate random design case we can write

$$\frac{\sqrt{nh_n}\left\{\hat{m}_{h_n,q}(x_0) - m(x_0)\right\}}{\sigma\sqrt{B(f(x))^{-1}}} \rightarrow N(0,1). \tag{4.20}$$

From (3.55) for a random sample of normally distributed residuals $\{\varepsilon_i\}_{i=1}^{n}$ with mean 0 and variance σ^2

$$\frac{n\hat{\sigma}^2_{MSW}}{\sigma^2} \sim \chi^2_n \tag{4.21}$$

where χ^2_n is the chi-squared distribution with n degrees of freedom. Hence, we combine (4.20) and (4.21) to claim that

$$\frac{\frac{\sqrt{nh_n}\left\{\hat{m}_{h_n,q}(x_0)-m(x_0)\right\}}{\sigma\sqrt{B(f(x))^{-1}}}}{\sqrt{\frac{\hat{\sigma}^2_{MSW}}{\sigma^2}}} \sim t_n. \tag{4.22}$$

The following statement (4.23) is obviously equivalent to (4.2)

$$\begin{aligned} Pr\left\{m(x) \in I_n\right\} &\approx t\left(\frac{\sqrt{nh_n}d}{\sqrt{B(f(x))^{-1}}\hat{\sigma}_{MSW}}\right) - t\left(-\frac{\sqrt{nh_n}d}{\sqrt{B(f(x))^{-1}}\hat{\sigma}_{MSW}}\right) \\ &= 2t\left(\frac{\sqrt{nh_n}d}{\sqrt{B(f(x))}\hat{\sigma}_{MSW}}\right) - 1 \end{aligned} \tag{4.23}$$

where $t(\cdot)$ is the cumulative student-t distribution and an approximate solution to the problem is provided by taking the smallest integer $n \geq 1$ such that

$$2t\left(\frac{\sqrt{nh_n}d}{\sqrt{B(f(x))^{-1}}\hat{\sigma}_{MSW}}\right) - 1 \geq 1 - \alpha \tag{4.24}$$

and since $h_n = n^{-r}$

$$n \geq \left(\frac{t^2_{\alpha/2,n} B \hat{\sigma}^2_{MSW}}{d^2 f(x)} \right)^{\frac{1}{1-r}} \tag{4.25}$$

where $t_{\alpha/2,n} = t^{-1}(1 - \alpha/2)$ is the $(1 - \alpha/2)^{th}$ quantile of the student-t distribution.

Two-stage sampling procedure is started by taking a pilot bivariate sample $\{X_i, Y_i\}_{i=1}^{n_0}$ and then estimate the required final sample size by N. Now using (4.25), we propose the following stopping rule using the two-stage sequential procedure:

$$N \equiv N(d) = \max \left\{ n_0, \left\lfloor \left(\frac{t^2_{\alpha/2,n_0} B \hat{\sigma}^2_{MSW,n_0}}{d^2 f(x)} \right)^{\frac{1}{1-r}} \right\rfloor + 1 \right\}. \tag{4.26}$$

If $N = n_0$ then we need no more observations in the second stage. But if $N > n_0$ then we take additional bivariate sample $\{X_i, Y_i\}_{i=n_0+1}^{N}$ of size $N - n_0$ in the second stage. Finally, we use the sample $\{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ to compute Nadaraya–Watson (2.18) and local linear (2.19) estimates for $m(x_0)$ and construct the confidence band given in (4.2). In an application of the stopping rule (4.26), it is important to select the best available values for the design constants r and n_0 for fixed predesigned values of d and α. The value for r is selected as explained in Section 4.2 and pilot sample size is a subjective choice.

4.4.3 Modified Two-Stage Sequential Procedure

The two stages of the proposed modified two-stage sequential procedure are defined as follows:

Stage 1: The initial sample size n_0 is determined using

$$n_0 = \max\left\{2, \left\lfloor \left(\frac{Z_{\alpha/2}}{d}\right)^{2\eta} \right\rfloor + 1\right\} \tag{4.27}$$

where $0 < \eta < 1$ and note that an appropriate value for η is obtained by means of simulations.

Stage 2: Compute N given by

$$N = \max\left\{n_0, \left\lfloor \left(\frac{Bt^2_{n_0,\alpha/2}\hat{\sigma}^2_{MSW,n_0}}{f(x)d^2}\right)^{\frac{1}{1-r}} \right\rfloor + 1\right\}. \tag{4.28}$$

If $N > n_0$ then we take further $N - n_0$ observations; otherwise no more observations are required in the second stage. Finally, we use the sample $\{X_j, Y_j\}_{j=1}^N$ to construct the fixed-width confidence $2d$ interval for $m(x_0)$: $[\hat{m}_{h_N,q}(x_0) - d, \hat{m}_{h_N,q}(x_0) + d]$.

Mukhopadhyay and Solanky (1994) used minimum sample size 2 in (4.27) for the theoretical computations and asymptotic analysis of the modified two-stage procedure.

4.5 Simulation Results

4.5.1 Equidistant Fixed Design

We use the following two models to assess the performance of the confidence bands developed in Section 4.3:

Model I : $Y = \sin^2(0.75x) + 3 + \varepsilon$

Model II: $Y = 2\exp\{-x^2/0.18\} + 3\exp\{-(x-1)^2/0.98\} + \varepsilon$.

Errors ε_i were generated from

(i) normal distribution $\varepsilon \sim N(0, .5^2)$ and

(ii) double exponential (Laplace) distribution $\varepsilon \sim DoubleExpo(0, \beta)$.

Value for scale parameter β, which is $0.5/\sqrt{2}$, was calculated to make $\mathbf{Var}(\varepsilon) = \sigma^2 = 0.25$. Half widths of the interval $d = 0.05, 0.07, 0.09, 0.12, 0.14$ were used. The initial sample size n_0 was chosen to be 25. The confidence bands were investigated for $\alpha = 0.1$ and $\alpha = 0.05$. For all simulations, we used standard normal kernel $K(u) = (2\pi)^{-1/2}\exp(-u^2/2)$; $-\infty < u < \infty$. For standard normal kernel,

$$\begin{aligned} B &= \int_t K^2(t)dt = \int_t \frac{1}{2\pi}e^{-t^2}dt = \frac{1}{2\pi}\int_t \exp\left\{-\frac{1}{2}\left(\frac{t}{\frac{1}{\sqrt{2}}}\right)^2\right\}dt \\ &= \frac{1}{2\pi}\int_t \exp\left\{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right\}dt \quad \text{with} \quad (\mu = 0) \quad \text{and} \quad \left(\sigma = \frac{1}{\sqrt{2}}\right) \\ &= \frac{1}{2\pi}\sqrt{2\pi}\sigma \\ &= (2\sqrt{\pi})^{-1}. \end{aligned} \tag{4.29}$$

In both models 15000 replicate samples for each experimental setting were carried out to obtain the final sample sizes required to estimate $m(x)$ at $x_0 = 0.306$ for a given fixed-width, $2d$. The statistics appear in Table 4.1 and Table 4.2 are defined below:

- $\bar{n} = \frac{\sum_{j=1}^{n_{sim}}(N)_j}{n_{sim}}$

where $(N)_j$ is the final sample size given in (4.14) calculated from j^{th} simulated sample and $n_{sim}\,(= 15000)$ is number of simulated samples.

- $SE\left(\bar{n}\right) = \sqrt{\frac{\sum_{j=1}^{n_{sim}}((N)_j - \bar{n})^2}{(n_{sim}-1)n_{sim}}}$
- $\%Over = ((\bar{n} - n_{opt})/n_{opt})\,100\%$
- $\bar{n_1} = \frac{\sum_{j=1}^{n_{sim}}(N_1)_j}{n_{sim}}$

where $(N_1)_j$ is sample size given in (4.13) calculated from j^{th} simulated sample.

- $SE\left(\bar{N_1}\right) = \sqrt{\frac{\sum_{j=1}^{n_{sim}}((N_1)_j - \bar{n_1})^2}{(n_{sim}-1)n_{sim}}}$
- $\bar{T} = \frac{\sum_{j=1}^{n_{sim}}(T)_j}{n_{sim}}$

where $(T)_j$ as given in (4.15) calculated from j^{th} simulated sample.

- $SE\left(\bar{T}\right) = \sqrt{\frac{\sum_{j=1}^{n_{sim}}\left((T)_j - \bar{T}\right)^2}{(n_{sim}-1)n_{sim}}}$
- $\overline{\hat{m}_{h_N,LL}(x_0)} = \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}(\hat{m}_{h_N,LL}(x_0))_j$
- $SE\left(\overline{\hat{m}_{h_N,LL}(x_0)}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}}\sum_{j=1}^{n_{sim}}\left((\hat{m}_{h_N,LL}(x_0))_j - \overline{\hat{m}_{h_N,LL}(x_0)}\right)^2\right\}^{1/2}$

where $(\hat{m}_{h_N,LL}(x_0))_j$ is the local linear estimator for j^{th} simulated sample.

- $\overline{\hat{m}_{h_N,NW}(x_0)} = \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}(\hat{m}_{h_N,NW}(x_0))_j$
- $SE\left(\overline{\hat{m}_{h_N,NW}(x_0)}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}}\sum_{j=1}^{n_{sim}}\left((\hat{m}_{h_N,NW}(x_0))_j - \overline{\hat{m}_{h_N,NW}(x_0)}\right)^2\right\}^{1/2}$

where $(\hat{m}_{h_N,NW}(x_0))_j$ is the Nadaraya–Watson estimator for j^{th} simulated sample.

- $\overline{\hat{\sigma}^2_{GSJ}} = \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}(\hat{\sigma}^2_{GSJ})_j$
- $SE\left(\overline{\hat{\sigma}^2_{GSJ}}\right) = \left\{\frac{1}{(n_{sim}-1)n_{sim}}\sum_{j=1}^{n_{sim}}\left((\hat{\sigma}^2_{GSJ})_j - \overline{\hat{\sigma}^2_{GSJ}}\right)^2\right\}^{1/2}$

where $(\hat{\sigma}^2_{GSJ})_j$ is the residual variance estimator for j^{th} simulated sample.

- $\tilde{p}_{LL} = \frac{n_{\hat{m}_{h_N,LL}(x_0)}}{n_{sim}}$

where $n_{\hat{m}_{LL}(x_0)}$ is the number of local linear fixed-width confidence intervals that contain $m(x_0)$ among n_{sim} confidence intervals in other words number of confidence intervals which satisfied $|(\hat{m}_{h_N,LL}(x_0))_{N_j} - m(x_0)| < d$ where $j = 1, \ldots, n_{sim}$.

- $SE_{\tilde{p}_{LL}} = \sqrt{\frac{\tilde{p}_{LL}(1-\tilde{p}_{LL})}{n_{sim}}}$

where SE_p is the standard error of the proportion p. Note that SE_p is calculated as $SE_p = \sqrt{p(1-p)/n}$ where n is the number of trials.

- $\tilde{p}_{NW} = \frac{n_{\hat{m}_{NW}(x_0)}}{n_{sim}}$

where $n_{\hat{m}_{NW}(x_0)}$ is the number of Nadaraya–Watson fixed-width confidence intervals that contain $m(x_0)$ among n_{sim} confidence intervals in other words number of confidence intervals which satisfied $|(\hat{m}_{h_N,NW}(x_0))_{N_j} - m(x_0)| < d$ where $j = 1, \ldots, n_{sim}$.

- $SE_{\tilde{p}_{NW}} = \sqrt{\frac{\tilde{p}_{NW}(1-\tilde{p}_{NW})}{n_{sim}}}$

First we consider the two-stage sequential procedure with a fixed design data for $\alpha = 0.05, 0.10$. The average final sample size ($\bar{n}$), average sample size which is not rounding up to get a multiple of n_0 ($\bar{n_1}$), average residual variance estimate ($\overline{\hat{\sigma^2}}$) averages of local linear ($\overline{\hat{m}_{h_N,LL}}$), and Nadaraya–Watson ($\overline{\hat{m}_{h_N,NW}}$) estimates. Finally, coverage probabilities of both estimators ($\tilde{p}_{LL}$), ($\tilde{p}_{NW}$) are reported in Tables 4.1 and 4.2 for $\alpha = 0.05$ and $\alpha = 0.10$ respectively. Figures enclosed in brackets (.) under estimated values in the tables refer to their standard errors of the estimated values.

The average amount of oversampling ($\%Over$) in the two-stage procedure is increasing with increasing d. The average percentage difference between $\bar{n}$ and $\bar{n_1}$ decreases with decreasing d. Coverage probabilities of both Nadaraya–Watson ($\tilde{p}_{NW}$) and local linear estimators ($\tilde{p}_{LL}$) have achieved preset confidence coefficients 95% and 90% at $x_0 = 0.306$ in Model II. But the coverage probabilities for Model I shows a different picture as Nadaraya–Watson estimator fails to achieve required coverage probabilities whereas local linear method does. This noticeable difference is mainly due to the fact that Model I is harmonic. And the bias term of Nadaraya–Watson estimator is heavily depend on derivatives of the unknown function $m(\cdot)$. However, (4.8) shows that as d decreases required final sample size N increases and hence improve coverage probabilities. Both tables depict this result. According to Table 4.1, $\tilde{p}_{NW}$ for Model I when $\alpha = 0.05$ started decreasing with decreasing d from 0.14 to 0.09 and then improved due to fairly large sample sizes for small d.

Similar pattern appears in Table 4.2 but $\tilde{p}_{NW}$ improves after $d = 0.07$ as calculated sample sizes are small when $\alpha = 0.10$ and larger sample size occurred for much smaller values of d compared to those in $\alpha = 0.05$. How-

Table 4.1: Empirical Coverage of LL and NW Estimators for Model I and Model II for $\alpha = 0.05$ and $x_0 = 0.306$.

	$\varepsilon_i \sim N(0, 0.5^2)$					$\varepsilon_i \sim DoubleExp(0, 0.5/\sqrt{2})$				
d	.14	.12	.09	.07	.05	.14	.12	.09	.07	.05
n_{opt}	64.6	105.4	262.8	583.6	1698.2	64.6	105.4	262.8	583.6	1698.2
	Model I : $\mathbf{m(x_0) = 3.052}$									
$\bar{n}$	109.7	171.7	403.0	890.8	2535.0	114.7	180.6	422.6	942.2	2663.6
	(.53)	(.86)	(2.10)	(4.78)	(13.5)	(.80)	(1.29)	(3.06)	(7.07)	(19.3)
$\%Over$	69.7%	62.8%	53.4%	52.7%	49.3%	77%	71.3%	60.8%	59.5%	56.9%
$\bar{n_1}$	97.2	158.8	393.8	878.3	2521.1	102.2	168.2	410.1	929.7	2651.0
	(.53)	(.85)	(2.12)	(4.75)	(14.0)	(.80)	(1.28)	(3.06)	(7.07)	(19.3)
$\bar{T}$	4.39	6.87	16.12	35.63	101.4	4.59	7.23	16.91	37.69	106.5
	(.02)	(.03)	(.08)	(.19)	(.54)	(.03)	(.05)	(.12)	(.28)	(.77)
$\overline{\hat{m}_{LL}}$	3.070	3.070	3.070	3.068	3.066	3.071	3.070	3.070	3.069	3.066
	(.001)	(.001)	(.000)	(.000)	(.000)	(.001)	(.001)	(.000)	(.000)	(.000)
$\overline{\hat{m}_{NW}}$	3.103	3.103	3.099	3.089	3.076	3.104	3.103	3.098	3.090	3.078
	(.001)	(.000)	(.000)	(.000)	(.000)	(.001)	(.000)	(.000)	(.000)	(.000)
$\tilde{p}_{LL}$	.9484	.9521	.9649	.9737	.9819	.9421	.9466	.9519	.9628	.9738
	(.001)	(.000)	(.000)	(.000)	(.001)	(.002)	(.002)	(.002)	(.002)	(.001)
$\tilde{p}_{NW}$	.9290	.9174	.9037	.9277	.9277	.9219	.9145	.9097	.9222	.9258
	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.002)	(.003)	(.003)	(.003)
$\overline{\hat{\sigma}^2}$	.2502	.2512	.2486	.2503	.249	.2504	.2518	.2488	.2518	.249
	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.000)	(.000)	(.000)
	Model II : $\mathbf{m(x_0) = 3.024}$									
$\bar{n}$	109.6	171.9	400.7	880.5	2561.4	114.7	180.6	422.6	942.2	2692.7
	(.53)	(.86)	(2.10)	(4.78)	(13.8)	(.53)	(.86)	(2.10)	(4.65)	(20.7)
$\%Over$	69.6%	63.0%	52.5%	50.9%	50.8%	77.4%	71.0%	60.9%	59.2%	58.6%
$\bar{n_1}$	97.2	158.8	393.8	878.4	2521.5	102.2	167.9	410.2	916.2	2680.2
	(.53)	(.85)	(2.12)	(4.75)	(14.0)	(.80)	(1.27)	(3.01)	(7.00)	(20.7)
$\bar{T}$	4.38	6.88	16.03	35.22	102.5	4.59	7.21	16.91	37.15	107.7
	(.02)	(.03)	(.08)	(.19)	(.55)	(.03)	(.05)	(.12)	(.28)	(.83)
$\overline{\hat{m}_{LL}}$	3.031	3.031	3.030	3.028	3.025	3.032	3.031	3.030	3.029	3.026
	(.001)	(.001)	(.000)	(.000)	(.000)	(.001)	(.001)	(.000)	(.000)	(.000)
$\overline{\hat{m}_{NW}}$	2.993	2.994	2.996	2.991	3.006	2.994	2.994	2.996	2.999	3.006
	(.001)	(.000)	(.000)	(.000)	(.000)	(.001)	(.000)	(.000)	(.000)	(.000)
$\tilde{p}_{LL}$	.9522	.9565	.9721	.9840	.9936	.9469	.9508	.9611	.9736	.9855
	(.002)	(.002)	(.001)	(.001)	(.001)	(.002)	(.002)	(.002)	(.001)	(.001)
$\tilde{p}_{NW}$	.9508	.9564	.9597	.9638	.9703	.9519	.9519	.9490	.9513	.9545
	(.002)	(.002)	(.002)	(.002)	(.001)	(.002)	(.002)	(.002)	(.002)	(.002)
$\overline{\hat{\sigma}^2}$	.2502	.2515	.2476	.2488	.2500	.2504	.2517	.2492	.2494	.2501
	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)

Table 4.2: Empirical Coverage of LL and NW Estimators for Model I and Model II for $\alpha = 0.10$ and $x_0 = 0.306$.

	$\varepsilon_i \sim N(0, 0.5^2)$					$\varepsilon_i \sim DoubleExp(0, 0.5/\sqrt{2})$				
d	.14	.12	.09	.07	.05	.14	.12	.09	.07	.05
n_{opt}	64.6	105.4	262.8	583.6	1698.2	64.6	105.4	262.8	583.6	1698.2
	Model I : $\mathbf{m(x_0) = 3.052}$									
$\bar{n}$	63.3	96.8	221.9	477.7	1366.0	66.4	100.9	233.3	497.5	1448.0
	(.28)	(.46)	(1.13)	(2.52)	(7.29)	(.43)	(.67)	(1.66)	(3.68)	(10.9)
$\%Over$	70.8%	60.1%	47.3%	42.8%	40.3%	49.3%	79.1%	66.9%	54.9%	48.7%
$\bar{n_1}$	51.7	84.2	208.5	464.7	1360.0	53.9	88.5	220.8	485.0	1435.5
	(.28)	(.45)	(1.12)	(2.51)	(7.38)	(.42)	(.67)	(1.66)	(3.68)	(10.9)
$\bar{T}$	2.53	3.87	8.88	19.11	54.64	2.65	4.04	9.33	19.9	57.9
	(.01)	(.02)	(.05)	(.10)	.29	(.02)	(.03)	(.07)	(.15)	(.44)
$\overline{\hat{m}_{LL}}$	3.069	3.069	3.070	3.070	3.069	3.067	3.068	3.070	3.070	3.067
	(.001)	(.001)	(.000)	(.000)	(.001)	(.000)	(.001)	(.000)	(.000)	(.000)
$\overline{\hat{m}_{NW}}$	3.104	3.102	3.102	3.070	3.104	3.084	3.103	3.102	3.097	3.085
	(.001)	(.001)	(.000)	(.000)	(.000)	(.001)	(.001)	(.000)	(.000)	(.000)
$\tilde{p}_{LL}$	.8814	.89117	.9132	.9289	.9497	.8857	.8917	.9030	.9137	.9357
	(.003)	(.003)	(.002)	(.002)	.002	(.003)	(.003)	(.002)	(.002)	(.002)
$\tilde{p}_{NW}$	.8646	.8560	.8264	.8000	.8243	.8697	.8564	.8217	.7993	.8224
	(.003)	(.003)	(.003)	(.003)	(.003)	(.003)	(.003)	(.003)	(.003)	(.003)
$\overline{\hat{\sigma}^2}$	.2493	.2513	.2507	.2507	.2509	.2499	.2516	.2516	.2497	.2521
	(.008)	(.008)	(.008)	(.008)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)
	Model II : $\mathbf{m(x_0) = 3.024}$									
$\bar{n}$	63.3	96.9	222.0	477.1	1349.1	66.5	101.1	232.7	496.8	1433.3
	(.28)	(.46)	(1.13)	(2.51)	(7.05)	(.43)	(.67)	(1.66)	(3.64)	(10.76)
$\%Over$	70.8%	60.3%	47.4%	42.6%	38.6%	79.4%	67.2%	54.5%	48.5%	47.2%
$\bar{n_1}$	51.7	84.2	208.5	464.8	1360.0	54.0	88.7	220.2	484.3	1435.5
	(.28)	(.45)	(1.12)	(2.51)	(7.4)	(.42)	(.67)	(1.66)	(3.64)	(10.8)
$\bar{T}$	2.53	3.87	8.88	19.11	54.64	2.65	4.04	9.31	19.87	57.3
	(.01)	(.02)	(.05)	(.10)	(.28)	(.02)	(.03)	(.07)	(.15)	(.43)
$\overline{\hat{m}_{LL}}$	3.031	3.030	3.031	3.030	3.027	3.032	3.030	3.031	3.029	3.027
	(.001)	(.001)	(.000)	(.000)	(.000)	(.001)	(.001)	(.000)	(.000)	(.000)
$\overline{\hat{m}_{NW}}$	2.992	2.992	2.994	2.996	3.002	2.993	2.992	2.994	2.996	3.002
	(.001)	(.000)	(.000)	(.000)	(.000)	(.001)	(.000)	(.000)	(.000)	(.000)
$\tilde{p}_{LL}$	.8875	.9009	.9274	.9455	.9727	.8904	.8989	.9161	.9339	.9601
	(.001)	(.001)	(.000)	(.000)	(.001)	(.003)	(.003)	(.002)	(.002)	(.002)
$\tilde{p}_{NW}$	.8966	.9009	.9151	.9081	.9186	.9041	.8991	.9029	.8989	.9027
	(.001)	(.001)	(.000)	(.000)	(.002)	(.002)	(.003)	(.002)	(.003)	(.002)
$\overline{\hat{\sigma}^2}$	.2494	.2515	.2508	.2506	.2491	.2501	.2518	.2511	.2496	.2507
	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)	(.001)

ever, according to (4.8) very small d results in larger sample sizes which is not desirable in practice and also very high d value result in small sample size which is not enough to achieve a given coverage probability. Note that decreasing d means $\hat{m}(\cdot) \approx m(\cdot)$ and this happens when n is fairly large and how large a sample we have to take depends again on individual bias terms and rate of convergence. This is very likely the reason why the average sample size $\bar{n}$ is fairly large compared to corresponding optimal sample size n_{opt} for both Models I and II.

Secondly, we look at how well modified two-stage sequential procedure performs in the case of fixed equidistant design over two-stage sequential procedure. Here errors are generated only from normal distribution ($\varepsilon_i \sim N(0, 0.5^2)$) as we could not observe any significant differences on the performances of two-stage procedure due to the specific selection of error distribution. Table 4.3 and Table 4.4 display the simulation results for $\alpha = 0.05$ and $\alpha = 0.10$ respectively. The figures under estimated values refer to their standard errors.

As expected, use of modified two-stage procedure result in less oversampling compared to two-stage procedure. This procedure is very desirable in particular for small d values as resulting final sample sizes are very close to respective optimal value. Furthermore, as we mentioned in Section 4.3, $\bar{n_1}$ values are very close to optimal sample sizes n_{opt} regardless of the value of d. However, average final sample sizes $\bar{n}$ do not show the same consequences as rounding effects are more dominant. Hence, we can say that structural constraints inherit to fixed equidistant design data has more control over the performance of sequential procedures no matter how desirable they are in general or how well they perform elsewhere.

Table 4.3: Empirical Coverage of LL and NW Estimators for Model I and Model II for $\alpha = 0.05$ and $x_0 = 0.306$.

d	n_{opt}	$\bar{n_1}$	$\bar{n}$	% *over*	$\bar{T}$	$\tilde{p}_{LL}$	$\tilde{p}_{NW}$	$\overline{\hat{m}_{LL}}$	$\overline{\hat{m}_{NW}}$	$\overline{\hat{\sigma}^2}$
	Model I : $\mathbf{m(x_0) = 3.052}$									
.14	64.6	83.2 0.36	102.7 0.37	58.9	2.7 .01	3.070 .0005	3.103 .0005	.953 .002	.931 .002	.250 .001
.12	105.4	128.3 0.48	153.7 0.49	45.8	3.1 .01	3.070 .0004	3.103 .0004	.963 .002	.925 .002	.251 .001
.09	262.8	292.4 0.84	334.6 0.86	27.3	4.0 .01	3.070 .0003	3.100 .0003	.974 .001	.904 .002	.249 .000
.07	583.6	626.2 1.41	692.3 1.44	18.6	5.3 .01	3.069 .0002	3.09 .0002	.984 .001	.909 .002	.250 .000
.05	1698.2	1767.6 2.93	1889.5 2.98	11.3	7.8 .01	3.066 .0001	3.077 .0001	.990 .001	.952 .002	.250 .000
	Model II : $\mathbf{m(x_0) = 3.024}$									
.14	64.6	83.2 0.36	102.7 0.37	58.9	2.7 .01	3.031 .0005	2.994 .0005	.957 .002	.959 .002	.250 .001
.12	105.4	126.9 0.47	152.6 0.48	44.7	3.1 .01	3.031 .0004	2.994 .0004	.968 .001	.966 .002	.249 .001
.09	262.8	292.2 0.84	334.5 0.86	27.3	4.0 .01	3.030 .0003	2.995 .0003	.987 .001	.968 .001	.249 .000
.07	583.6	625.8 1.42	692.9 1.45	18.7	5.3 .01	3.029 .0002	2.999 .0002	.994 .001	.972 .001	.250 .000
.05	1698.2	1771.7 2.92	1893.3 2.98	11.5	7.8 .01	3.026 .0001	3.006 .0001	.999 .000	.985 .001	.251 .000

Table 4.4: Empirical Coverage of LL and NW Estimators for Model I and Model II for $\alpha = 0.10$ and $x_0 = 0.306$.

d	n_{opt}	$\bar{n}$	$\bar{n}$	% over	$\bar{T}$	$\tilde{p}_{LL}$	$\tilde{p}_{NW}$	$\overline{\hat{m}_{LL}}$	$\overline{\hat{m}_{NW}}$	$\overline{\hat{\sigma}^2}$
	Model I : $\mathbf{m(x_0) = 3.052}$									
.14	37.0	48.7 0.25	62.9 0.26	69.8	2.3 .01	3.070 0.001	3.105 0.001	.884 .003	.864 .003	.249 .001
.12	60.4	75.2 0.33	94.1 0.34	55.7	2.5 .01	3.069 0.001	3.102 0.001	.905 .002	.866 .003	.252 .001
.09	150.6	170.6 0.57	201.7 0.59	33.9	3.3 .01	3.071 0.000	3.104 0.000	.929 .002	.829 .003	.250 .001
.07	334.5	362.9 0.95	411.7 0.98	23.1	4.2 .01	3.070 0.000	3.099 0.000	.945 .002	.787 .003	.251 .000
.05	973.5	1014.9 1.97	1103.9 2.02	13.4	6.2 .01	3.068 0.000	3.084 0.000	.961 .002	.835 .003	.250 .000
	Model II : $\mathbf{m(x_0) = 3.024}$									
.14	37.0	48.7 0.25	62.9 0.26	69.8	2.3 .01	3.033 0.001	2.993 0.001	.891 .003	.9018 .0024	.249 .001
.12	60.4	75.2 0.33	94.1 0.34	55.7	2.5 .01	3.031 0.001	2.992 0.001	.919 .002	.9135 .0023	.252 .001
.09	150.6	170.2 0.57	201.3 0.58	33.7	3.3 .01	3.031 0.000	2.994 0.000	.944 .002	.9238 .0022	.250 .001
.07	334.5	362.6 0.97	411.7 1.00	23.1	4.2 .01	3.030 0.000	2.995 0.000	.966 .002	.9227 .0022	.250 .000
.05	973.5	1019 1.97	1107.8 2.01	13.8	6.3 .01	3.027 0.000	3.002 0.000	.988 .001	.9325 .0020	.251 .000

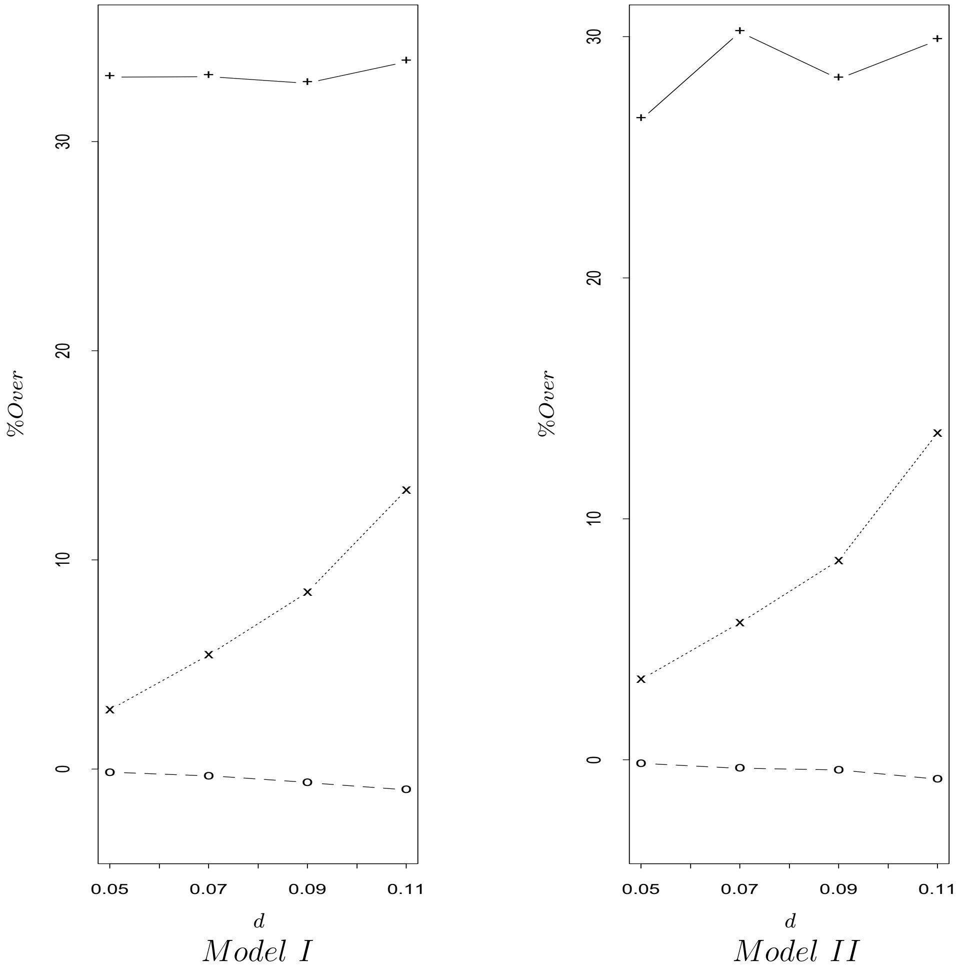
4.5.2 Random Design

We use the following two models to assess the performance of the confidence bands developed in Section 4.4:
Model I : $Y = \sqrt{4x+3} + \varepsilon$
Model II: $Y = 2\exp\{\frac{-x^2}{0.18}\} + 3\exp\{-\frac{(x-1)^2}{0.98}\} + \varepsilon$
where $\varepsilon \sim N(0, \sigma^2)$.
Half-widths of the interval $d = 0.05, 0.07, 0.09, 0.11, 0.13$ were used. The initial sample size n_0 and σ were chosen to be 25 and 0.5 respectively. The confidence bands were investigated for $\alpha = 0.05$. For all the data analysed, we used standard normal kernel $K(u) = (2\pi)^{-1/2}\exp(-u^2/2)$ and hence $B = 2\sqrt{\pi}$. In both models 15000 replicate samples for each experimental setting were carried out to obtain the final sample sizes required to estimate $m(x)$ at $x_0 = 0.306$ for a given fixed-width, $2d$.

We obtained 15000 random samples of $\{X_i\}_{i=1}^{25}$ from uniform distribution and then calculate corresponding y_i for each stated relation (Models I and II). Random errors ε were generated from $N(0, 0.5^2)$ distribution and added to the above y_i to obtained Y_i. First we considered two-stage sequential procedure for $\alpha = 0.05$ and then modified two-stage sequential procedure and purely sequential procedure. The average final sample size $\bar{n}$, average residual variance estimate $\overline{\hat{\sigma}^2}$, average local linear $\overline{\hat{m}_{LL}}$, average Nadaraya–Watson $\overline{\hat{m}_{NW}}$ estimates and coverage probability $\tilde{p}$ which is the proportion of the confidence intervals that contains the theoretical value, $m(x_0)$ estimated at the point $x_0 = 0.306$ are reported in Tables 4.5 and 4.6 for $\alpha = 0.05$.

Coverage probabilities of both Nadaraya–Watson ($\tilde{p}_{NW}$) and local linear estimators ($\tilde{p}_{LL}$) have achieved preset confidence coefficient 95% at $x_0 = 0.306$ in Model II except when $d = 0.13$. But the coverage probabilities for Model I shows a different picture as Nadaraya–Watson estimator fails to achieve required coverage probabilities except when $d = 0.05$ whereas local linear method does. This noticeable difference is mainly due to the structural differences in the selected models and also the bias terms which heavily depend on derivatives of the unknown function $m(\cdot)$ associated with each estimator. However, Nadaraya–Watson coverage probabilities ($\tilde{p}_{NW}$) for Model I increases with decreasing d due to large sample sizes. This is consistent with all the sequential procedures. The performance of Nadaraya–Watson estimator worsens as x increases as its bias highly depends on derivatives of $m(\cdot)$. For the interior point $x_0 = 0.306$, the Nadaraya–Watson estimator assigns symmetric weights to both sides of $x_0 = 0.306$. For a random design this will overweigh the points on right hand side and hence create large bias. In other words Nadaraya–Watson estimator is not design-adaptive. However, local linear method assigns asymmetrical weighting scheme while maintaining the same type of smooth weighting scheme as Nadaraya–Watson estimator. Hence, local linear method adapts automatically to this random design.

This simulation analysis clearly shows that the average sample sizes in two-stage procedure is much larger than corresponding values in both purely sequential and modified two-stage procedures for both models. This evidence clearly implies that the two-stage procedure is less efficient compared to purely and modified two-stage sequential procedures but at the same time one should note that it is also associated with the highest coverage probability which exceeds the target confidence coefficient 95%. Nevertheless modified two-stage procedure has reduced the amount of oversampling significantly and has achieved target confidence coefficient simultaneously. Further note that the advantage of using a modified two-stage procedure is reflected in computational time. The purely sequential procedure needs substantially more computations and hence during simulations it needs significantly more computational times than the two-stage procedure, particularly for small d. However purely sequential procedure at times fall somewhat short of the optimal sample size. Hence, the coverage probability falls short of the target especially when half-width of the interval d becomes larger as it result in small sample sizes. Nevertheless, it achieves values closer to target coverage probability for smaller d due to larger sample sizes.



Model I *Model II*

Figure 4.1: Over % Vs Half-Width of the Interval, d.

(*Legend*: '+' = Two-stage, 'x' = Modified Two-stage,'o' = Purely Sequential.)

Table 4.5: Empirical Coverage of LL and NW for Model I for $\alpha = 0.05,\ x_0 = 0.306$ and $m(x_0) = 2.055$.

d	n_{opt}	$\bar{n}$	$\tilde{p}_{LL}$	$\tilde{p}_{NW}$	$\overline{\hat{m}_{LL}}$	$\overline{\hat{m}_{NW}}$	$\overline{\hat{\sigma}^2}$
			Two – Stage Procedure				
0.13	81.77	109.33	0.9473	0.9019	2.0462	2.1080	0.2649
		(0.40)	(0.0018)	(0.0024)	(0.0012)	(0.0009)	(0.0012)
0.11	138.97	185.94	0.9653	0.9123	2.0477	2.0961	0.2617
		(0.69)	(0.0015)	(0.0030)	(0.0009)	(0.0011)	(0.0011)
0.09	262.78	348.93	0.9779	0.9214	2.0479	2.0912	0.2644
		(1.28)	(0.0012)	(0.0033)	(0.0007)	(0.0008)	(0.0010)
0.07	583.56	776.72	0.9894	0.9324	2.0469	2.0842	0.2650
		(2.83)	(0.0008)	(0.0024)	(0.0003)	(0.0004)	(0.0009)
0.05	1698.19	2259.69	0.9962	0.9582	2.0480	2.0763	0.2649
		(8.34)	(0.0005)	(0.0016)	(0.0001)	(0.0003)	(0.0008)
			Modified Two – Stage Procedure				
0.13	81.77	97.5	0.9493	0.8655	2.0459	2.1196	0.2598
		(.26)	(0.0018)	(0.0028)	(0.0005)	(0.0005)	(0.0004)
0.11	138.97	157.5	0.9633	0.8763	2.0463	2.1201	0.2565
		(.38)	(0.0015)	(0.0030)	(0.0004)	(0.0004)	(0.0004)
0.09	262.78	285.0	0.9800	0.8903	2.0464	2.1036	0.2535
		(.57)	(0.0011)	(0.0026)	(0.0003)	(0.0003)	(0.0003)
0.07	583.56	615.4	0.9901	0.9026	2.0475	2.1089	0.2526
		(.97)	(0.0008)	(0.0035)	(0.0002)	(0.0002)	(0.0002)
0.05	1698.19	1746.5	0.9973	0.9639	2.0480	2.0784	0.2513
		(2.06)	(0.0004)	(.0015)	(0.0001)	(0.0001)	(0.0002)
			Purely Sequential Procedure				
0.13	81.78	80.1	0.9184	0.8693	2.0463	2.2193	0.2425
		(0.00)	(0.0001)	(0.0001)	(0.001)	(0.0001)	(0.0001)
0.11	138.97	137.6	0.9348	0.9014	2.0462	2.1895	0.2468
		(0.00)	(0.0001)	(0.0001)	(0.001)	(0.0001)	(0.0001)
0.09	262.78	261.1	0.9409	0.9142	2.0474	2.1094	0.2489
		(0.00)	(0.0001)	(0.0001)	(0.000)	(0.0001)	(0.0001)
0.07	583.56	581.7	0.9417	0.9265	2.0476	2.0975	0.24975
		(0.00)	(0.0001)	(0.0001)	(0.000)	(0.0001)	(0.0001)
0.05	1698.19	1695.6	0.9489	0.9471	2.0517	2.0818	0.2503
		(0.00)	(0.0000)	(0.0001)	(0.000)	(0.0001)	(0.0001)

Table 4.6: Empirical Coverage of LL and NW for Model II for $\alpha = 0.05,\ x_0 = 0.306$ and $m(x_0) = 3.024$.

d	n_{opt}	$\bar{n}$	$\tilde{p}_{LL}$	$\tilde{p}_{NW}$	$\overline{\hat{m}_{LL}}$	$\overline{\hat{m}_{NW}}$	$\overline{\hat{\sigma}^2}$
			Two – Stage Procedure				
0.13	81.77	105.06	0.9464	0.9559	3.0375	3.0108	0.2580
		(0.40)	(0.0018)	(0.0017)	(0.0010)	(0.0010)	(0.0010)
0.11	138.97	180.46	0.9591	0.9667	3.0368	3.0044	0.2602
		(0.68)	(0.0016)	(0.0015)	(0.0009)	(0.0008)	(0.0009)
0.09	262.78	337.03	0.9731	0.9537	3.0306	2.9925	0.2583
		(1.27)	(0.0013)	(0.0017)	(0.0008)	(0.0003)	(0.0007)
0.07	583.56	759.76	0.9891	0.9761	3.0319	3.0029	0.2608
		(2.91)	(0.0008)	(0.0012)	(.0003)	(0.0004)	(0.0003)
0.05	1698.19	2149.35	0.9943	0.9540	3.0274	3.0012	0.2563
		(8.25)	(0.0006)	(0.0017)	(0.0001)	(0.0002)	(0.0002)
			Modified Two – Stage Procedure				
0.13	81.77	95.8	0.9506	0.9471	3.0276	2.9926	0.2568
		(0.26)	(0.0018)	(0.0018)	(0.0005)	(0.0005)	(0.0004)
0.11	138.97	157.8	0.9634	0.9274	3.0237	2.9806	0.2568
		(0.38)	(0.0015)	(0.0021)	(0.0004)	(0.0004)	(0.0004)
0.09	262.78	284.5	0.9784	0.9323	3.0280	2.9859	0.2532
		(0.57)	(0.0012)	(0.0021)	(0.0003)	(0.0003)	(0.0003)
0.07	583.56	616.7	0.9928	0.9429	3.0213	2.9951	0.2529
		(0.99)	(0.0007)	(0.0024)	(0.0002)	(0.0002)	(0.0003)
0.05	1698.19	1754.8	0 .9983	0.9803	3.0277	3.0041	0.2520
		(2.09)	(0.0003)	(0.0011)	(0.0001)	(0.0001)	(0.0002)
			Purely Sequential Procedure				
0.13	81.77	79.56	0.9362	0.9011	3.0211	2.9834	0.2410
		(0.40)	(0.0028)	(0.0030)	(0.0007)	(0.0008)	(0.0005)
0.11	138.97	137.85	0.9386	0.9162	3.0305	2.9946	0.2462
		(0.68)	(0.0020)	(0.0023)	(0.0005)	(0.0005)	(0.0013)
0.09	262.78	261.66	0.9468	0.9240	3.0332	2.9989	0.2481
		(1.27)	(0.0015)	(0.0019)	(0.0004)	(0.0004)	(0.0002)
0.07	583.56	581.47	0.9480	0.9257	3.0291	2.9970	0.2489
		(2.91)	(0.0009)	(0.0021)	(0.0002)	(0.0003)	(0.0001)
0.05	1698.19	1695.62	0.9470	0.9380	3.0252	3.0041	0.2496
		(8.25)	(0.0010)	(0.0036)	(0.0003)	(0.0003)	(0.0002)

Chapter 5

Symmetric Bootstrap Fixed-width confidence Interval for Nonparametric Kernel Regression

5.1 Introduction

An application of bootstrapping to the construction of fixed-width confidence bands is considered in the context of nonparametric regression estimation. The developed bootstrap confidence bands are then compared with those constructed in Sections 4.3 and 4.4 by the direct method based on asymptotic mean, variance and distribution of regression estimators. In general, the development of confidence intervals in nonparametric regression falls into two parts, the first being construction of a confidence interval for $m(x_0)$ and the second involving bias $\mathbf{E}[\hat{m}_{h_n}(x_0) - m(x_0)]$ correction. The effect of bias depends very much on how bias is corrected and there are different views as to how this should be done.

Bootstrapping is a resampling technique which has an appealing nonparametric approach for approximating sampling distribution of a statistic for small and also for moderate sample sizes. It will generate the frequency distribution of the statistic by resampling the given sample. There are many ways in which this sort of information could be of considerable assistance in nonparametric kernel regression. One application of interest is the construction of confidence interval for a given point of interest. However, special care must be taken to account for the bias terms encountered in nonparametric kernel estimation.

Several authors have developed bootstrap methods for constructing confidence intervals in nonparametric regression. Both pivotal and non-pivotal approaches have been employed. Recall that a distribution function F of both data and unknown parameter is said to be pivotal if it has the same distribution for all values of the unknowns. Hardle and Bowman (1988) used non-pivotal technique for constructing confidence intervals for in nonparametric regression whereas Hall (1992) drew attention to some of the theoretical advantages of a pivotal approach in the context of nonparametric regression.

McDonald (1982) has suggested the use of bootstrap methods for assessing variability bands in nonparametric regression and the used method was based on resampling from the empirical distribution of the pairs of observations. Dikta (1988) investigated McDonald's approach and showed that up to a bias term, a type of pointwise bootstrap confidence interval is asymptotically correct. Bickel and Freedman (1981) have argued that resampling should be done from estimated residuals when the predictor variables are fixed and non-random in the setting of linear regression. Hardle and Bowman (1988) applied this estimated residual resampling scheme in the case of random predictor to the problems of local adaptive choice of bandwidth by approximating the mean squared error of the nonparametric estimate at some point of interest and then to the construction of confidence bands.

Most of the published work on bootstrap confidence intervals is based on equal-tailed two-sided intervals (Efron, 1981; 1982; 1987). An equal-tailed $(1-\alpha)$ level of confidence interval for an unknown regression function at a given point x_0, $m(x_0)$ would be of the form

$$[\hat{m}_{h_n,q}(x_0) - d_1, \hat{m}_{h_n,q}(x_0) + d_2]$$

where $\hat{m}_{h_n,q}(x_0)$ is a point estimate of $m(x_0)$ and d_1, d_2 are chosen so that

$$Pr\left[m(x_0) < \hat{m}_{h_n,q}(x_0) - d_1\right] = Pr\left[m(x_0) > \hat{m}_{h_n,q}(x_0) + d_2\right] = \frac{\alpha}{2}.$$

However, symmetric two-sided percentile bootstrap interval has a form of

$$[\hat{m}_{h_n,q}(x_0) - d, \hat{m}_{h_n,q}(x_0) + d]$$

where d is chosen so that

$$Pr(|\hat{m}_{h_n,q}(x_0) - m(x_0)| > d) = \alpha.$$

Let (X_i, Y_i); $i = 1, \dots, n$ be identical and independently distributed pairs of observations with unknown bivariate distribution. Note that X_i design points could be either from fixed equidistant or random designed data. Here we consider the case of symmetric two-sided bootstrap type confidence intervals. Motivation behind using symmetrized version is Hall (1988) who showed that symmetric intervals have better coverage accuracy than equal tailed intervals. A fixed-width confidence interval procedure for $m(x_0)$ based on $\hat{m}_{h_n,q}(x_0)$ having length $2d$ and coverage probability $(1-\alpha)$ is a stopping random variable N such that

$$Pr\left[\hat{m}_{h_N,q}(x_0) - d \le m(x_0) \le \hat{m}_{h_N,q}(x_0) + d\right] \ge 1 - \alpha. \tag{5.1}$$

Essentially, the problem of constructing an interval $I_{N,d} = (\hat{m}_{h_N,q}(x_0) - d, \hat{m}_{h_N,q}(x_0) + d)$ is translated into a problem of determining the sample size. Indeed, if the sample size N is too small then the interval $I_{N,d}$ will not achieve preset coverage probability $1 - \alpha$.

In principle, confidence intervals can be obtained from asymptotic distribution results for $\hat{m}_{h_n,q}(x_0)$

$$\frac{(\hat{m}_{h_n,q}(x_0) - \mathbf{E}[\hat{m}_{h_n,q}(x_0)])}{\sqrt{\mathbf{Var}[\hat{m}_{h_n,q}(x_0)]}} \to \zeta \tag{5.2}$$

that is

$$\frac{\sqrt{nh_n}\,(\hat{m}_{h_n,q}(x_0) - \mathbf{E}[\hat{m}_{h_n,q}(x_0)])}{\sigma\sqrt{B(f(x_0))^{-1}}} \to \zeta.$$

In Chapter 4 we discussed how to determine sample size N such that the coverage probability $(1 - \alpha)$ attains the preset confidence coefficient using the asymptotic normal approximation i.e, $\zeta \equiv N(0,1)$. However, in practice residual variance σ^2 is unknown and replaced by a suitable estimator $\hat{\sigma}^2$. Thus, the construction of these asymptotic confidence intervals involves assessing the distribution of $\frac{\hat{m}_{h_n,q}(x_0)}{\hat{\sigma}}$ which is not always possible to derive as it depends not only on the individual distributions of $\hat{m}_{h_n,q}(x_0)$ and $\hat{\sigma}^2$ but also the distributional effects when they are considered together in case $f(x_0)$ is known. On the other hand, when $f(x_0)$ is unknown this problem becomes more critical since then we have to look at joint effect of three terms $\hat{f}(x_0)$, $\hat{m}_{h_n,q}(x_0)$ and $\hat{\sigma}$. Also it is difficult to come across the distribution of residual variance estimator $\hat{\sigma}^2$ in nonparametric regression regardless of whether $f(x_0)$ is known or unknown. Even if it is possible, we have only approximation or asymptotic distribution. To resolve this we could employ bootstrap method which is considered as an alternative method of estimating properties of unknown distributions. Resampling could be done from a suitably estimated residual distribution and then utilizes the percentiles of the approximate distribution to construct confidence intervals for the curve at a given design point.

Here we suggests a novel approach using bootstrap ideas which is first introduced by Swanepoel et. al. (1984) for the case of sample mean. Here, the distribution of $\hat{m}_{h_n,q}(x_0)$ is not approximated by the estimated asymptotic distribution but by an unknown distribution obtained from resampling and whose quantiles can therefore be computed. Approximate fixed-width confidence intervals can then be constructed by employing these quantiles.
Define

$$\zeta_n(c) = Pr\left\{|\hat{m}_{h_n,q}(x_0) - m(x_0)| \leq c\right\} \tag{5.3}$$

where $\zeta_n(\cdot)$ is the distribution function of the symmetrised estimator of $\hat{m}_{h_n,q}(x_0)$.
Now the optimal sample size n_{opt} which assures the interval $I_{n_{opt},d}$ producing at least coverage probability $(1-\alpha)$ satisfies

$$Pr\left\{\left|\hat{m}_{h_{n_{opt}},q}(x_0) - m(x_0)\right| \leq d\right\} \geq 1-\alpha \tag{5.4}$$

where

$$n_{opt} = \min\left\{n : Pr\left[|\hat{m}_{h_n,q}(x_0) - m(x_0)| \leq d\right] \geq 1-\alpha\right\}$$

and any sample size $n \geq n_{opt}$ will satisfy inequality in (5.4).
Since

$$Pr\left\{\sqrt{n_{opt}h_{n_{opt}}}\left|\hat{m}_{h_{n_{opt}},q}(x_0) - m(x_0)\right| \leq \sqrt{n_{opt}h_{n_{opt}}}d\right\} \geq 1-\alpha,$$

$$\begin{aligned}\zeta_{n_{opt}}\left(d\sqrt{n_{opt}h_{n_{opt}}}\right) &= Pr\left\{\sqrt{n_{opt}h_{n_{opt}}}\left|\hat{m}_{h_{n_{opt}},q}(x_0) - m(x_0)\right| \leq d\sqrt{n_{opt}h_{n_{opt}}}\right\} \\ &\geq 1-\alpha\end{aligned}$$

which implies

$$d\sqrt{n_{opt}h_{n_{opt}}} \geq \xi_{n_{opt}}(\alpha)$$

where $\xi_n(\alpha)$ is the $(1-\alpha)^{th}$ quantile of the $\zeta_n(\cdot)$ distribution.

Since we take bandwidth $h_n = n^{-r}$, $n_{opt}h_{n_{opt}} = n_{opt}^{1-r}$

$$d\sqrt{n_{opt}^{1-r}} \geq \xi_{n_{opt}}(\alpha)$$

which implies

$$n_{opt} \geq \left[\frac{\xi_{n_{opt}}(\alpha)}{d}\right]^{\frac{2}{1-r}}. \tag{5.5}$$

In order to estimate n_{opt}, we use

$$\hat{n}_{opt} \geq \left[\frac{\hat{\xi}_{n_0}(\alpha)}{d}\right]^{\frac{2}{1-r}} \tag{5.6}$$

where we suggest using bootstrap critical point $\hat{\xi}_{n_0}(\alpha)$ based on smaller available sample $(X_1, Y_1), \ldots, (X_{n_0}, Y_{n_0})$ of size n_0 as a estimator of $\xi_{n_{opt}}$ which is the only unknown quantity in (5.6). Note that in practice n_0 is the pilot sample size which is a subjective choice.

Even though much research has been done in sequential analysis, sequential procedures are not commonly employed in practice. But they are of great importance as we can find many situations where we do not know in advance how many observations or sample size will be required to reach a decision. We consider the Stein two-stage sequential procedure as given in Sections 4.3.1 and 4.4.1 which requires only two sampling operations. However, it turned out that this two-stage procedure lacks efficiency (Section 4.5). Methodology is developed for determining whether it is advantageous to use the bootstrap method to reduce the extent of oversampling that is normally known to plague Stein's two-stage sequential procedure. It is a well known fact that for a wide class of statistics, the bootstrap approximation has a high degree of accuracy. Therefore, we would expect the proposed procedure will perform better than the classical two-stage sequential procedure based on student-t approximation given in Chapter 4.

In this chapter we propose a method which combines bootstrap ideas with the two-stage sequential procedure to estimate unknown regression function at any given point using the smallest possible sample size to achieved preassigned level of accuracy. The outline of this chapter is as follows: Section 5.2 introduces bootstrapping in nonparametric kernel regression. Section 5.3 explains the implementation of bootstrap methodology to two-stage procedure. An extensive simulation study is conducted to illustrate the application of the developed procedure. The simulation results on confidence intervals and their coverage probabilities are discussed in Section 5.4. We show that the amount of oversampling cause by two-stage procedure can be reduced significantly by employing a bootstrap technique.

5.2 Bootstrapping in Nonparametric Kernel Regression Estimation

Efron (1979,1982) explained and explored the bootstrap method in detail. Let $\hat{\theta}$ is an estimate of a parameter θ based on a sample $X_1, \ldots, X_n$ drawn from unknown distribution F. The standard bootstrap technique is to estimate θ by sampling method but with the samples being drawn not from F itself but from the empirical distribution function F_n of the observed data $X_1, \ldots, X_n$. A sample from F_n is generated by successively selecting uniformly with replacement from $X_1, \ldots, X_n$ to construct a bootstrap sample $X_1^*, \ldots, X_n^*$. For each bootstrap sample an estimator of θ is calculated. Since arbitrarily large numbers of bootstrap samples can be constructed θ can easily be estimated to any reasonable required accuracy from the simulated samples. The estimator $\hat{\theta}$ is then used as an estimate of θ. All the members of drawn bootstrap samples consist of the observations from original sample and nearly every sample will contain repeated values. Samples drawn from empirical distribution F_n in the bootstrap simulations will have some rather odd properties as empirical distribution F_n is a discrete distribution. However Efron (1979) came up with the idea of smoothed bootstrap as a modification to the bootstrap procedure to avoid bootstrap samples with these odd properties. In a smoothed bootstrap, the resampling is conducted not from the empirical distribution F_n but from a smoothed version $\hat{F}$ of F_n. Some properties of the smoothed bootstrap and also some insights into circumstances when the smoothed bootstrap will give better results than the standard bootstrap is described comprehensively by Silverman and Young (1987).

In general, bootstrap technique used in nonparametric regression replace any occurrence of the unknown distribution F in the definition of the statistical function of interest by the empirical distribution function F_n of the observed errors $\{\varepsilon_i\}_{i=1}^n$. We can not observe F_n because these residuals are not directly observed in a context of regression analysis although they can be estimated from the respective fitted model $\hat{\varepsilon}_i = Y_i - \hat{m}_{h_n,q}(x_i)$. Here we employ nonparametric kernel regression estimators, Nadaraya–Watson estimator $\hat{m}_{h_n,NW}(x_0)$ and local linear estimator $\hat{m}_{h_n,LL}(x_0)$ which are respectively defined by (2.18) and (2.19) in Section 2.3 as the fitted values of $\hat{m}_{h_n,q}(x_0)$. It is important however to note that $\hat{m}_{h_n,q}(x_0)$ is a biased estimator of $m(x_0)$ and that if its bandwidth h_n is chosen to balance this bias against the standard deviation of $\hat{m}_{h_n,q}(x_0)$, then the variance and squared bias will have the same speed of convergence to 0 (Wand and Jones, 1995).

There are several ways of bootstrapping in the context of nonparametric regression. Hardle and Marron (1991) did not resample from the entire set of residuals as in Hardle and Bowman (1988). They used the idea of wild bootstrapping which is developed based on literature by Rosenblueth (1975). In wild bootstrapping, each bootstrap residual ε_i^* is drawn from the two-point distribution which has mean zero $\mathbf{E}[\varepsilon_i^*] = 0$, variance equal to the square of the residual $\mathbf{Var}[\varepsilon_i^*] = \hat{\varepsilon_i}^2$ and third moment equal to the cube of the residual $\mathbf{E}[\varepsilon_i^{*^3}] = \hat{\varepsilon_i}^3$ where $\hat{\varepsilon}_i = Y_i - \hat{m}_{h_n,q}(X_i)$.

The standard bootstrap approach of resampling from the pairs $\{(X_i, Y_i)\}_{i=1}^n$ is not recommended because the bootstrap bias will be 0. For more details on bootstrapping regression models we refer to Hardle and Bowman (1988) and Hall (1992). We consider the regression model (2.1) as defined in Section 2.1. Depending on the structure of the design points of explanatory variable X, considered regression model takes the form of (2.2) or (2.3) as explained in Section 2.1 with density function $f(x)$ where $f(x) = 1$ in the case of fixed equidistant design. Without loss of generality, we assume that $X_i \in (0, 1)$ as we can make any type of data set to be in $(0, 1)$ by taking $t = \frac{x-a}{b-a}$ where a and b are minimum and maximum values of the data set respectively and reverting back to original scale by $x = (b - a)t + a$. In the regression model given in (2.1), ε s are independent and identically distributed (iid) random variable with zero mean $\mathbf{E}(\varepsilon) = 0$ and constant variance $\mathbf{Var}[\varepsilon] = \sigma^2$. Also there is one-to-one correspondence between ε_i and the data pair (X_i, Y_i). Therefore our approach to the bootstrapping in the nonparametric regression is to first use the estimated residual $\hat{\varepsilon}_i$

$$\hat{\varepsilon}_i = Y_i - \hat{m}_{g_n,q}(x_i); \quad i = 1, \ldots, n \tag{5.7}$$

where g_n is a bandwidth (different from h_n to be introduced later). The idea is to resample from the estimated residuals which are the differences between the observed values of response variable and the kernel type estimators and then use this data to construct an estimator whose distribution will approximate the distribution of the original estimator. The estimated residuals need not necessarily have mean 0, so to let the resampled residuals reflect the behaviour of the true observed errors they should first be centred as

$$\tilde{\varepsilon}_i = \hat{\varepsilon}_i - \frac{\sum_{j=1}^{n} \hat{\varepsilon}_j}{n}; \quad i = 1, \ldots, n. \tag{5.8}$$

This form of bootstrapping preserves the error distribution in the data and guarantees that the bootstrap observations have errors with zero mean. One of the main advantages of this approach is, it correctly accounts for the bias of the estimator.

Since $\mathbf{Var}[\tilde{\varepsilon}_i] < \sigma^2$ it is desirable to adjust the centred residuals to incorporate $\mathbf{Var}[\tilde{\varepsilon}_i] = \sigma^2$. The adjusted residuals are defined as

$$\tilde{\varepsilon}_i \rightarrow \frac{\tilde{\varepsilon}_i}{\sqrt{1 - 1/n}} \qquad i = 1, \ldots, n. \tag{5.9}$$

The bootstrap samples are then constructed by adding to the observed estimate errors which are randomly chosen with replacement from the collection of centered residuals from the original data. Let $\varepsilon_i^*, \ldots, \varepsilon_n^*$ be a sample of bootstrap residuals drawn randomly with replacement from the set $\tilde{\varepsilon}_1, \ldots, \tilde{\varepsilon}_n$. Resampled Y^* are then constructed by

$$Y_i^* = \hat{m}_{g_n,q}(x_i) + \varepsilon_i^*; \quad i = 1, \ldots, n \tag{5.10}$$

where $\hat{m}_{g_n,q}(x_i)$ is a kernel type estimator with bandwidth g_n chosen to be larger than h_n. The reason why we take $g_n > h_n$ is based on asymptotic analysis of mean of $[\hat{m}_{h_n,q}(x_0) - m(x_0)]$ under the conditional distribution of $Y_1, \ldots, Y_n | X_1, \ldots, X_n$ and mean of $[\hat{m}^*_{h_n,q}(x_0) - \hat{m}_{g_n,q}(x_0)]$ under the conditional distribution of $Y_1^*, \ldots, Y_n^* | (X_1, Y_1), \ldots, (X_n, Y_n)$ in the simple situation when the marginal density $f(x)$ is constant in a neighbourhood of x_0. Rosenblatt (1969) showed that

$$\mathbf{E}_{Y|X}\left[\hat{m}_{h_n,q}(x_0) - m(x_0)\right] \approx \frac{h_n^2}{2}\left(\int_t t^2 K(t)dt\right) m''(x_0) \tag{5.11}$$

$$\mathbf{E}_{Y^*|(X,Y)}\left[\hat{m}^*_{h_n,q}(x_0) - \hat{m}_{g_n,q}(x_0)\right] \approx \frac{h_n^2}{2}\left(\int_t t^2 K(t)dt\right) m''_{g_n,q}(x_0). \tag{5.12}$$

From (5.11) and (5.12) for the two distributions, which are $[\hat{m}_{h_n,q}(x_0) - m(x_0)]$ and $[\hat{m}^*_{h_n,q}(x_0) - \hat{m}_{g_n,q}(x_0)]$, to have the same bias, $m''_{g_n,q}(x_0) \rightarrow m''(x_0)$. This requires choosing g_n tending to 0 at a rate slower than the h_n for estimating $m(x)$ (Hardle, 1990).

Recall that our ultimate objective is to estimate distribution of $\hat{m}_q(x_0) - m(x_0)$ by $\hat{m}^*_q(x_0) - \hat{m}_q(x_0)$. However, $\hat{m}_q(x_0)$ is a biased estimator of $m(x_0)$ and hence according to (5.11) and (5.12) not only different bandwidths required for $\hat{m}_q(x_0)$ and $\hat{m}^*_q(x_0)$ estimators but also they need to be chosen such that $\hat{m}_q(x_0) - m(x_0)$ and $\hat{m}^*_q(x_0) - \hat{m}_q(x_0)$ have the same speed of convergence to zero. This condition is satisfied by selecting different bandwidths g_n and h_n such that $g_n > h_n$. To be precise, distribution of $\hat{m}_{h_n,q}(x_0) - m(x_0)$ is estimated by distribution of $\hat{m}^*_{h_n,q}(x_0) - \hat{m}_{g_n,q}(x_0)$. Moreover, rationale for selecting different bandwidths for $\hat{m}_q(x_0)$ and $\hat{m}^*_q(x_0)$ is also elucidated by Theorem 8.

Finally, we use bootstrap sample $(X_1, Y_1^*), \ldots, (X_n, Y_n^*)$ to estimate $\hat{m}^*_{h_n,q}(x_0)$. Corresponding bootstrap estimates for Nadaraya–Watson and local linear methods are

$$\begin{aligned}\hat{m}^*_{h_n,NW}(x_0) &= \frac{\sum_{i=1}^n K\left(\frac{x_0 - X_i}{h_n}\right) Y_i^*}{\sum_{i=1}^n K\left(\frac{x_0 - X_i}{h_n}\right)} \\ &= \frac{\sum_{i=1}^n K\left(\frac{x_0 - X_i}{h_n}\right)\left(\hat{m}_{g_n,NW}(X_i) + \varepsilon_i^*\right)}{\sum_{i=1}^n K\left(\frac{x - X_i}{h_n}\right)} \end{aligned} \tag{5.13}$$

and

$$\begin{aligned}\hat{m}^*_{h_n,LL}(x_0) &= \frac{\sum_{i=1}^n w_i Y_i^*}{\sum_{i=1}^n w_i} \\ &= \frac{\sum_{i=1}^n w_i \left(\hat{m}_{g_n,LL}(X_i) + \varepsilon_i^*\right)}{\sum_{i=1}^n w_i}. \qquad (5.14)\end{aligned}$$

The choice of bandwidths h_n and g_n are crucial to the performance of the bootstrap nonparametric estimators (5.13) and (5.14). Here we estimate bandwidths as discussed in Section 2.5. In order to conform with the condition $g_n > h_n$ which leads to $n^{-r_1} > n^{-r}$ we select values r_1 and r in such a way $r_1 < r$ and $r_1 \in \left(\frac{-\ln[\min(x_0, 1-x_0)]}{\ln n}, 1\right)$ where x_0 is the point that estimation is taking place.

Hardle (1990) came up with the following theorem suggesting the distribution of $\hat{m}_{h_n,q}(x) - m(x)$ is approximated by the distribution of $\hat{m}^*_{h_n,q}(x) - \hat{m}_{g_n,q}(x)$.

Theorem 8 *Suppose that $\int_u |K(u)|^{2+\eta} du < \infty$ for some $\eta > 0$, $m(x)$ and $f(x)$ twice differentiable and $\boldsymbol{E}[|Y|^{2+\eta}|X = x]$, $g_n > h_n$ and $f(x_i) > 0$ for $i = 1, \ldots, n$. Then for almost all sample sequences and for all $c \in \mathbb{R}$*

$$\left| Pr^{Y|X}\left\{\triangle_{x,n}\right\} - Pr^{Y^*|X}\left\{\sqrt{nh_n}\left[\hat{m}^*_{h_n}(x) - \hat{m}_{g_n}(x)\right] < c\right\}\right| \to 0$$

where $\triangle_{x,n} = \sqrt{nh_n}\left[\hat{m}_{h_n}(x) - m(x)\right] < c$.

Here we use the symbol $Y|X$ to denote the conditional distribution of $Y_1, \ldots, Y_n|X_1, \ldots, X_n$ and $Y^*|X$ to denote the bootstrap distribution of $Y_1^*, \ldots, Y_n^*|(X_1, Y_1), \ldots, (X_n, Y_n)$. Hence, $\hat{m}^*_{h_n,q}(x_0)$ can be used as the basis for constructing a confidence interval for $m(x_0)$.

5.3 Bootstrap Sequential Confidence Bands in Nonparametric Kernel Regression Estimation

One of main objective of bootstrapping is to gain information on the distribution of an estimator. Thus fixed-width confidence bands for the unknown regression function at specific design point or points can also be derived from using the percentiles of the bootstrap distribution. Faraway (1990) developed a bootstrap method to estimate average squared error of a kernel based nonparametric regression estimator for a given bandwidth and also proposed simultaneous $100(1-\alpha)\%$ bootstrap confidence bands for $m(x)$ to be constructed as $[\hat{m}_{h_n}(x) - \bar{c}_{(\alpha)}, \hat{m}_{h_n}(x) + \underline{c}_{(\alpha)}]$ where $\bar{c}_{(\alpha)}$ and $\underline{c}_{(\alpha)}$ are the appropriate α level sample percentile of

$$\bar{c}_j = \max_{1\le i\le n} \left[(\hat{m}^*_{h_n}(x_i))_j - \hat{m}_{g_n}(x_i) \right]$$

and

$$\underline{c}_j = \max_{1\le i\le n} \left[\hat{m}_{g_n}(x_i) - (\hat{m}^*_{h_n}(x_i))_j \right]; \qquad j = 1, \ldots, n_b$$

respectively, where $\left(\hat{m}^*_{h_n}(x)\right)_j$ is the j^{th} bootstrap estimate based on bootstrap sample $\{(X_i, Y_i^*), \ldots, (X_n, Y_n^*)\}$ where $Y_i^* = \hat{m}_{g_n}(x_i) + \varepsilon_i^*$ and n_b is number of bootstrap samples. This method is an extension of work on bootstrap bandwidth selection for density estimates in Faraway (1988).

Given $(X_i, Y_i), \ldots, (X_n, Y_n)$, $(X_i, Y_i^*), \ldots, (X_n, Y_n^*)$ be a bootstrap sample of size n drawn with replacement and Pr_n^* denotes its corresponding distribution which is the empirical distribution based on the original sample.

Define

$$Pr_n^*\left\{\sqrt{nh_n}\left|\hat{m}^*_{h_n,q}(x_0)-\hat{m}_{g_n,q}(x_0)\right|\leq c\right\}$$

where c is positive constant and

$$\hat{m}_{g_n,q}(x_0)=\begin{cases}\hat{m}_{g_n,LL}(x_0) & \text{if q=LL;}\\ \hat{m}_{g_n,NW}(x_0) & \text{if q=NW.}\end{cases}$$

Here we advocate a particular bootstrap distribution which uses centered but not standardised estimator. Since

$$\left|Pr_n^*\{\cdot\}-Pr\left\{\sqrt{nh_n}\left[m(x_0)-\hat{m}_{h_n,q}(x_0)\right]\leq c\right\}\right|\to 0 \tag{5.15}$$

where

$$Pr_n^*\{\cdot\}=Pr_n^*\left\{\sqrt{nh_n}\left[\hat{m}^*_{h_n,q}(x_0)-\hat{m}_{g_n,q}(x_0)\right]\leq c\right\}\quad \text{as}\quad n\to\infty.$$

From the theory of bootstrapping, this suggests the way in which the distribution of the nonparametric estimate about the true curve at some point of interest may be approximated by suitable centering of the nonparametric estimates based on bootstrap samples.

The proposed stopping rules in Chapter 4 were involved with an asymptotic approximation of residual variance, bias and variance of kernel estimators, estimation of density function of explanatory variable X for random design data, the normal approximation and chi-square distribution. This section investigates the use of the bootstrap in providing approximations to a suitably centered distribution of kernel estimators of nonparametric regression estimation. On the other hand, we expect the application of bootstrapping to two-stage sequential procedure will reduce the amount of oversampling for some extent. Here we replace all the unknown quantities by the corresponding bootstrap critical value. In this method, one first draws a bootstrap sample $\{(X_j, Y_j^*)\}_{j=1}^{n_0}$ with replacement from a set of initial observations of size n_0 and Y_j^*s are calculated as explained in Section 5.2. Define

$$U_i^* = \left| \sqrt{n_0 h_{n_0}} \left[(\hat{m}^*_{h_{n_0},q}(x_0))_i - \hat{m}_{g_{n_0},q}(x_0) \right] \right|; \qquad i = 1, \dots, n_b \tag{5.16}$$

where $\hat{m}^*_{h_{n_0},q}(x_0)$ is the i^{th} bootstrap nonparametric regression estimate based on i^{th} bootstrap sample $\left\{(X_1, Y_1^*), \dots, \left(X_{n_0}, Y_{n0}^*\right)\right\}$ and n_b is the number of bootstrap replications. Using bootstrap approximation:

$$Pr^*_{n_0} \left\{ \left| \hat{m}^*_{h_{n_0},q}(x_0) - \hat{m}_{g_{n_0},q}(x_0) \right| \le d \right\} \ge 1 - \alpha$$

the equivalent bootstrap optimal sample size n^*_{opt} is given by

$$n^*_{opt} \ge \left\{ \frac{\xi^*_{\alpha,n_0}}{d} \right\}^{\frac{2}{1-r}} \tag{5.17}$$

where ξ^*_{α,n_0} is the $[\lfloor n_b(1-\alpha) \rfloor]^{th}$ largest value of $\{U^*_{(1)}, \dots, U^*_{(n_b)}\}$ and $U^*_{(i)}$s are the ordered values of U_i^* for $i = 1, \dots, n_b$. Then ξ^*_{α,n_0} is the estimated bootstrap critical value. Hence, two-stage sampling procedure based on bootstrap critical point is given by

$$N_q^* \equiv N_q^*(d) = \max \left\{ n_0, \left\lfloor \left(\frac{[\xi^*_{\alpha,n_0}]_q}{d} \right)^{\frac{2}{1-r}} \right\rfloor + 1 \right\} \tag{5.18}$$

where

$$\left[\xi^*_{\alpha,n_0}\right]_q = [\lfloor n_b(1-\alpha) \rfloor]^{th} \quad \text{value of} \quad \{U^*_{q(i)}\}_{i=1}^{n_b}$$

and

$$U^*_{q(i)} = \begin{cases} U^*_{NW(i)} = \left| \sqrt{n_0 h_{n_0}} \left[\hat{m}^*_{h_{n_0},NW}(x_0)_i - \hat{m}_{g_{n_0},NW}(x_0) \right] \right| & \text{if q=NW} \\ U^*_{LL(i)} = \left| \sqrt{n_0 h_{n_0}} \left[\hat{m}^*_{h_{n_0},LL}(x_0)_i - \hat{m}_{g_{n_0},LL}(x_0) \right] \right| & \text{if q=LL.} \end{cases}$$

If $N_q^* > n_0$ then we take further $(N_q^* - n_0)$ observations, otherwise no more observations are required in the second stage. Note that this approach does not require an estimators for σ or/and $f(x)$ regardless of the data design since this estimation part is already included in the estimator ξ^*_{α,n_0}. The accuracy of those approximations can be improved by increasing number of bootstrap samples n_b. Increasing the number of bootstrap replicates is limited by the computational time though.

According to Section 5.2, it is obvious that unlike the stopping rules developed in Chapter 4 here we have two stopping rules N^*_{LL} and N^*_{NW} depending on which nonparametric kernel estimator has been used in attaining bootstrap distribution of $U = \sqrt{nh_n}[\hat{m}^*_{h_n,q}(x_0) - \hat{m}_{g_n,q}(x_0)]$. Finally, we use both N^*_{LL} and N^*_{NW} final sample sizes to estimate Nadaraya–Watson estimator and local linear estimator. Hence we have two estimates $\hat{m}_{h_{N^*_{NW}},NW}(\cdot)$ and $\hat{m}_{h_{N^*_{LL}},NW}(\cdot)$ for Nadaraya–Watson method based on the samples $(X_1, Y_1), \ldots, (X_{N^*_{NW}}, Y_{N^*_{NW}})$ and $(X_1, Y_1), \ldots, (X_{N^*_{LL}}, Y_{N^*_{LL}})$ respectively. Same with the local linear method and the two estimates are $\hat{m}_{h_{N^*_{NW}},LL}(\cdot)$ and $\hat{m}_{h_{N^*_{LL}},LL}(\cdot)$.

It is important to note that if the initial sample size n_0 is too small the approximated bootstrap critical point ξ^*_{α,n_0} may differ too much from the optimal value $\xi_{\alpha,n_{opt}}$. There is a certain region of small sample sizes n_0 in which the approximated bootstrap critical points ξ^*_{α,n_0} are subject to a lot of variation (with extreme large values). This instability and overestimation is carried over (in a quadratic way) to the final sample size N^*_q as $\left(\xi^*_{\alpha,n_0}\right)^{\frac{2}{1-r}}$ is used in calculating (5.18). It is clear that this region should be avoided and that the choice of the initial sample size n_0 is more sensitive when using the bootstrap critical values.

Thus, in an application of above stopping rule (5.18), it is important to select suitable values for the design constants r and n_0 for fixed predesigned values of d and α. Note that value for r is chosen as described in Section 2.7 and pilot sample size n_0 is an arbitrary sensible number to start up the sequential procedures.

5.3.1 Fixed Equidistant Designed data

From equation (5.18) we propose the following rule for $N^*_{1,q}$:

$$N^*_{1,q} = \max\left\{ n_0, \left\lfloor \left(\frac{[\xi^*_{\alpha,n_0}]_q}{d} \right)^{\frac{2}{1-r}} \right\rfloor + 1 \right\} \tag{5.19}$$

where $r \in \left(\frac{-\ln[min(x_0, 1-x_0)]}{\ln n_0}, 1 \right)$. Therefore from (5.19), ξ^*_{α,n_0} bootstrap critical value replaces $t_{\nu,\alpha/2}\sigma\sqrt{B}$ in the stopping rule given in (4.13).

In order to comply with the fixed equidistant design data and to continually use the observed data in the initial sample, take the final sample size $N^*_q = n_0 T^*_q$ where T^*_q is a positive integer given by

$$T^*_q = \left\lfloor \frac{N^*_{1,q}}{n_0} \right\rfloor = \max\left\{ 1, \left\lfloor \frac{\left(\frac{[\xi^*_{\alpha,n_0}]_q}{d} \right)^{\frac{2}{1-r}}}{n_0} \right\rfloor \right\} \tag{5.20}$$

and $N^*_q \geq N^*_{1,q}$. As discussed in Section 4.3 the constraint inherent in the structure of fixed equidistant design data leads to an overestimated final sample size N^*_q regardless of which nonparametric estimator has been used to attain bootstrap critical point from distribution of U. Clearly if $T^*_q = 1$, no additional observations are required in the second stage and $N^*_q = n_0$. However, if $T^*_q > 1$ we take extra sample of size $N^*_q - n_0 = n_0(T^*_q - 1)$ in the second stage with

$$x_i = \frac{i}{n_0 T^*_q} \quad \text{for} \quad i = n_0 + 1, \ldots, (n_0 T^*_q - 1) \quad \text{and} \quad i \neq T^*_q, 2T^*_q, \ldots, n_0 T^*_q. \tag{5.21}$$

Note that the initial sample data corresponds to (x_i, Y_i) for $i = T^*_q, 2T^*_q, \ldots,$ $n_0 T^*_q$. Finally, we use the sample $\{(x_1, Y_1), \ldots, (x_{N^*_q}, Y_{N^*_q})\}$ with $x_i = i/N^*_q$ to compute Nadaraya–Watson (2.18) and local linear (2.19) estimates for $m(x_0)$ and construct the confidence band given in (5.1).

5.3.2 Random Designed data

Two-stage sampling procedure is started by taking a pilot bivariate sample $\{X_i, Y_i\}_{i=1}^{n_0}$ and then estimate the required final sample size N_q^*. From (4.26), bootstrap critical value ξ^*_{α,n_0} replaces $t_{n,\alpha/2}\sigma\sqrt{\frac{B}{f(x)}}$. Hence, two-stage sampling procedure based on bootstrap critical point is given by

$$N_q^* \equiv N_q^*(d) = \max\left\{ n_0, \left\lfloor \left(\frac{[\xi^*_{n_0,\alpha}]_q}{d} \right)^{\frac{2}{1-r}} \right\rfloor + 1 \right\}. \qquad (5.22)$$

If $N_q^* = n_0$ then we need no more observations in the second stage. But if $N_q^* > n_0$ then we take additional bivariate sample $\{X_i, Y_i\}_{i=n0+1}^{N_q^*}$ of size $N_q^* - n_0$ in the second stage. Finally, we use the sample $\{(X_1, Y_1), \ldots, (X_{N_q^*}, Y_{N_q^*})\}$ to compute Nadaraya–Watson (2.18) and local linear (2.19) estimates for $m(x_0)$ and construct the confidence band given in (5.1).

5.4 Simulation Results

A simulation study was conducted to compare 95% ($\alpha = 0.05$) fixed-width confidence intervals using Nadaraya–Watson and local linear estimators with and without bootstrapping. The performance of the bootstrap confidence interval for the unknown regression function at a specific point x_0 is compared with confidence interval based on asymptotic distribution. Bootstrap requires more computational effort in an attempt to reflect features of underlying distribution whereas the direct method is simpler but based on asymptotic distribution. Fortran programs were used to carry through all computations.

The initial sample size n_0 was chosen to be 25. For all simulations, we used standard normal kernel, $K(u) = (2\pi)^{-1/2}\exp(-u^2/2)$; $-\infty < u < \infty$. The parameter r of the bandwidth was computed as previously discussed in Section 2.7. The bootstrap resampling procedure given in Sections 5.2 and 5.3 is carried out for $n_b = 500$. Given a sample $(X_1, Y_1), \ldots, (X_{n_0}, Y_{n_0})$ we take n_b bootstrap samples $(X_1, Y_1^*), \ldots, (X_{n_0}, Y_{n_0}^*)$ of size $n_0 = 25$. For each bootstrap sample we calculate the appropriate bootstrap statistic for i^{th} bootstrap sample i.e., $U_i^* = \left|\sqrt{n_0 h_{n_0}}\left(\hat{m}^*_{h_{n_0},q}(x_0) - \hat{m}_{g_{n_0},q}(x_0)\right)\right|$; $i = 1, \ldots, n_b$. The sampling distribution of the resulting $n_b = 500$ values of statistic U_i^* is taken as an approximation to the actual bootstrap distribution. The $(1-\alpha)^{th}100\%$ percentile of this sampling distribution ξ^*_{α,n_0} provides an appropriate approximation to the actual bootstrap critical value ξ_{α,n_0}.

During the simulation, first the sample size required to achieve the preset confidence coefficient is estimated. The final sample sizes i.e. without bootstrap N and with bootstrap N_q^*, are respectively obtained from the stopping rules given in Section 4.3 and Section 5.3. Tables 5.1 and 5.2 give the summary results obtained from the simulation study with and without bootstrap. Here $\tilde{p}$ is the coverage probability, $\bar{n}$ is the average final sample size and the standard errors of these quantities are calculated in the same way as defined in Section 4.5. To quantify the accuracy of each estimate, the associated standard error is included in the tables underneath the corresponding value. Estimate of nonparametric regression function at a given point from each method and their coverage probabilities are calculated using final sample sizes based on with and without bootstrapping and given in columns 6-17. For each selected value of half-width of the interval d, the performance of procedures (with and without bootstrapping) can be evaluated by looking at the average sample sizes $\bar{n}$, $\overline{n^*_{LL}}$ and $\overline{n^*_{NW}}$ and coverage proportion of the confidence intervals $[I_N - d, I_N + d]$ resulting form 15000 simulations of original data followed in each case by 500 bootstrap simulations.

Sections 5.4.1 and 5.4.2, respectively, present the results of the simulation study for the case of fixed equidistant designed data discussed in Section 5.3.1 and the case of random designed data discussed in Section 5.3.2. The results present in both Tables 5.1 and 5.2 are the average of those 15000 iterations. In the following sections we consider three main points. The first is investigation of how much difference there is between average sample sizes based on pointwise bootstrap and direct confidence intervals. Second, we compute and compare the coverage probabilities of the bootstrap confidence intervals with those of without bootstrapping. Third, we compare average values of Nadaraya–Watson estimator and local linear estimator with $\overline{\hat{m}_{h_{N_q^*},q}(x_0)}$ and without bootstrapping $\overline{\hat{m}_{h_N,q}(x_0)}$ with the theoretical value $m(x_0)$ for a given point x_0.

The simulation results show the benefit in using confidence bands based on centered bootstrap approximation instead of those based on asymptotic distribution.

5.4.1 Fixed Equidistant Design

We use the following two models to assess the performance of the confidence bands developed in Section 5.3.1:

Model I : $Y = \sin^2(0.75x) + 3 + \varepsilon_i$

Model II: $Y = 2\exp\{-x^2/0.18\} + 3\exp\{-(x-1)^2/0.98\} + \varepsilon_i$

where $\varepsilon_i \sim N(0, \sigma^2)$ with $\sigma^2 = 0.25$.

Half-widths of the interval are chosen to be $d = 0.05, 0.07, 0.09, 0.12, 0.14$. In both models 15000 replicate samples for each experimental setting are carried out to obtain the final sample sizes required to estimate $m(x_0)$ at

$x_0 = 0.306$ for a given fixed-width, $2d$. We take an initial sample of size 25. Hence, design points of explanatory variable X take the form $\left\{x_i = \frac{i}{25}\right\}_{i=1}^{25}$. Then corresponding y_i values are calculated for each stated relation (Models I and II). Random errors ε_i are generated from Normal distribution $N\left(0, 0.5^2\right)$ and added to the above y_i to obtain Y_i.

Results of Table 5.1 looks impressive since the observed coverage probabilities are all close to or well above desired value of 95% for most of the d values. Also bootstrap average sample sizes $\overline{n_{NW}^*}$ and $\overline{n_{LL}^*}$ are lower than $\overline{n}$ for every d value. However average sample sizes based on Nadaraya–Watson estimator show somewhat disappointing results since the $\overline{n_{NW}^*}$ values are all significantly below than the optimal sample size n_{opt} and the difference $n_{opt} - \overline{n_{NW}^*}$ is becoming large for decreasing half-width of the interval d. Whereas sample sizes based on local linear method $\overline{n_{LL}^*}$ show better result as every single average sample size is less than $\bar{n}$ and close to the corresponding optimal sample size n_{opt}. The difference between $\overline{n_{LL}^*}$ and $\overline{n_{NW}^*}$ show clearly the bias that is inherent to nonparametric regression estimation as both values depend on the distribution of $U_i^* = \left|\sqrt{n_0 h_{n_0}}\left[(\hat{m}_{h_{n_0},q}^*(x_0))_i - \hat{m}_{g_{n_0},q}(x_0)\right]\right|$.

Observe that most of the coverage probabilities based on the local linear estimator $\tilde{p}_{LL}$ have achieved 95% compared with those are based on Nadaraya–Watson estimator $\tilde{p}_{NW}$ which are slightly lower. Careful investigation reveals that this was due to problems with the estimated bias. Of course this bias effect goes away asymptotically but in the models considered here shows that it is not negligible especially in the Model I (Column 17) and we believe this problem will occur quite often. Note that because $\hat{m}_{NW}$ has larger bias term than $\hat{m}_{LL}$ the estimated bias will typically be bigger than the bias of $\hat{m}_{LL}$. The effect does not look very large in the average estimates $\overline{\hat{m}_{NW}}$ but simultaneous coverage turns out to be a very sensitive quantity. As expected from the previous analysis, the local linear coverage probabilities $\tilde{p}_{LL}$ based on $\bar{n}$ are slightly bigger (due to the large sample sizes) than those based on n^*_{LL} (not a significant difference in most d values though). However, average local linear estimators $\overline{\hat{m}_{LL}}$ based on n^*_{LL} are quite close to the theoretical value $m(x_0)$. The difference of local linear coverage probabilities $\tilde{p}_{LL}$ between those based on $\bar{n}$ and n^*_{LL} is really quite small while those based on n^*_{NW} are quite distant.

Of course coverage probabilities of these confidence intervals, with and without bootstrap all fall below the preset confidence coefficient 95% for large d values due to the fact that when there are less data available, the estimates are less accurate. The one surprising feature is that the average Nadaraya–Watson estimator $\overline{\hat{m}_{NW}}$ estimated using n^*_{NW} are close to the theoretical value $m(x_0)$ compare to other estimates based on n and n^*_{LL} in the case of Model I. Also as the sample size is much larger, it seems reasonable to hope that the asymptotic negligibility of the bias problem is closer to being realized.

- $\overline{n^*_{LL}} = \frac{\sum_{j=1}^{n_{sim}} (N^*_{LL})_j}{n_{sim}}$

where $(N^*_{LL})_j$ is final sample size given in (5.18) calculated from j^{th} simulated sample and n_{sim} (= 15000) is number of simulated samples.

- $SE\left(\overline{n^*_{LL}}\right) = \sqrt{\frac{\sum_{j=1}^{n_{sim}} \left((N^*_{LL})_j - \overline{n^*_{LL}}\right)^2}{(n_{sim}-1)n_{sim}}}$
- $\overline{n^*_{NW}} = \frac{\sum_{j=1}^{n_{sim}} (N^*_{NW})_j}{n_{sim}}$

where $(N^*_{NW})_j$ is final sample size given in (5.18) calculated from j^{th} simulated sample and n_{sim} (= 15000) is number of simulated samples.

- $SE\left(\overline{n^*_{NW}}\right) = \sqrt{\frac{\sum_{j=1}^{n_{sim}} \left((N^*_{NW})_j - \overline{n^*_{NW}}\right)^2}{(n_{sim}-1)n_{sim}}}$

Table 5.1: Empirical Coverage of LL and NW Estimators with and without Bootstrapping: Fixed Equidistant Design Data for $\alpha = 0.05$.

d	n_{opt}	$\bar{n}$	$\bar{n}^*_{LL}$	$\bar{n}^*_{NW}$	$\overline{\hat{m}_{LL}(x_0)}$			$\tilde{p}_{LL}$			$\overline{\hat{m}_{NW}(x_0)}$			$\tilde{p}_{NW}$		
					n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	n^*_{NW}
	$\mathbf{m(x) = \sin^2(.75x) + 3;\quad m(x_0) = 3.052}$ at $\mathbf{x_0 = 0.306}$															
.14	64.6	109.6	80.6	61.6	3.070	3.068	3.063	.945	.941	.910	3.104	3.084	3.064	.922	.932	.926
		.53	.33	.24	.001	.002	.001	.002	.002	.002	.001	.001	.001	.002	.002	.002
.12	105.4	170	123.4	92.7	3.070	3.067	3.063	.951	.951	.920	3.103	3.082	3.061	.919	.937	.933
		.85	.53	.38	.001	.002	.001	.002	.002	.002	.000	0.001	.001	.002	.002	.002
.09	262.8	407.4	290.6	213.8	3.070	3.068	3.062	.965	.965	.942	3.101	3.081	3.058	.895	.932	.949
		2.15	1.34	.96	.001	.002	.001	.002	.002	.002	.001	.000	.000	.003	.002	.000
.07	583.6	882.7	625.2	456.8	3.069	3.067	3.061	.974	.976	.958	3.093	3.078	3.056	.878	.924	.952
		4.68	2.89	2.08	.000	.001	.000	.001	.001	.002	.000	.000	.000	.003	.002	.002
.05	1698.2	2545.6	1800.6	1306.5	3.067	3.065	3.060	.980	.982	0.973	3.080	3.072	3.054	.902	.932	.957
		13.59	8.44	6.05	.000	.001	.000	.001	.001	.001	.000	.000	.000	.002	.002	.002
	$\mathbf{m(x) = 2\exp(-x^2/.18) + 3\exp(-(x-1)^2/.98);\quad m(x_0) = 3.024}$ at $\mathbf{x_0 = 0.306}$															
.14	64.6	109.0	80.6	61.7	3.031	3.023	3.021	.949	.944	.912	2.992	3.003	3.031	.952	.943	.911
		.52	.33	.24	.001	.002	.001	.002	.002	.002	.001	.001	.001	.002	.002	.002
.12	105.4	170.6	124	93.4	3.030	3.022	3.020	.959	.955	.927	2.992	3.003	3.033	.955	.947	.907
		.85	.53	.38	.001	.002	.001	.002	.002	.002	.001	.001	.001	.002	.002	.002
.09	262.8	408.3	291.5	214.4	3.029	3.022	3.020	.976	.974	.944	2.994	3.005	3.040	.961	.956	.878
		2.13	1.31	.94	.001	.001	.000	.001	.001	.002	.001	.000	.001	.002	.002	.003
.07	583.6	888.8	627.9	458.9	3.029	3.022	3.021	.985	.982	.946	2.998	3.007	3.042	.962	.951	.822
		4.74	2.91	2.1	.000	.001	.000	.001	.001	.002	.000	.000	.000	.002	.002	.003
.05	1698.2	2586.9	1820.8	1325.3	3.026	3.021	3.022	.992	.992	.957	3.004	3.012	3.045	.967	.942	.813
		13.78	8.49	6.19	.000	.001	.000	.001	.001	.002	.000	.000	.000	.002	.002	.003

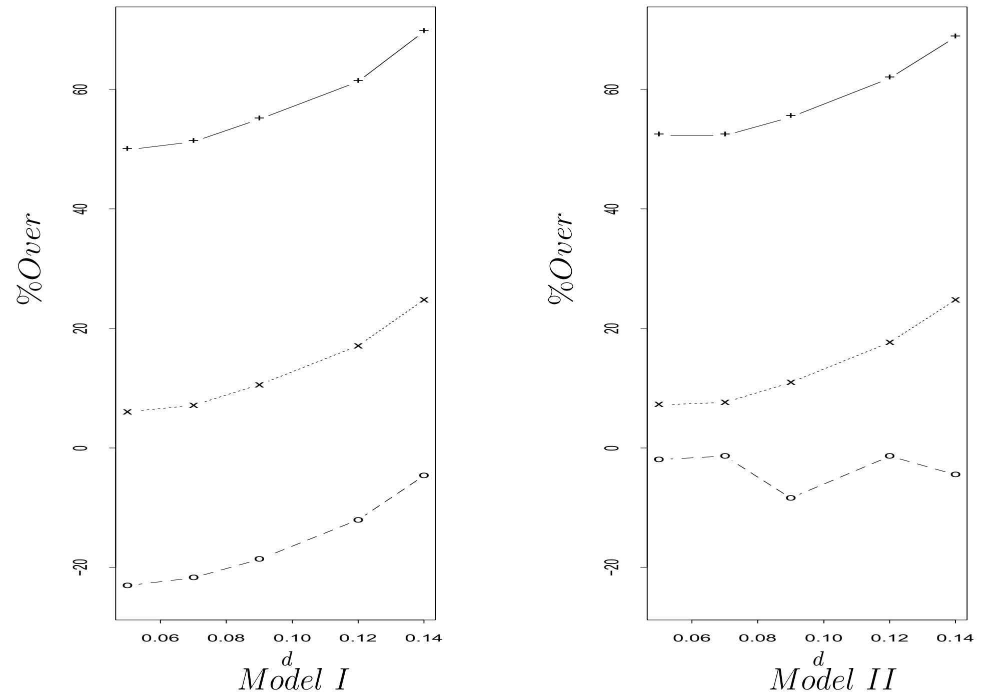


Figure 5.1: Over % Vs Half-Width of the Interval, d:

Fixed Equidistant Data.

(*Legend*: '+' = $(\frac{\bar{n}-n_{opt}}{n_{opt}})100\%$, 'x' = $(\frac{\overline{n_{LL}}^{*}-n_{opt}}{n_{opt}})100\%$, 'o' =$(\frac{\overline{n_{NW}}^{*}-n_{opt}}{n_{opt}})100\%$).

Figure 5.1 reflects the amount of over sampling ($\%Over$) which is calculated by either $\left(\frac{\overline{n_q^*}-n_{opt}}{n_{opt}}\right)100\%$ where $q = NW$, LL or $\left(\frac{\bar{n}-n_{opt}}{n_{opt}}\right)100\%$. Even both $\bar{n}$ and $\overline{n_{LL}^*}$ are over sampling, average sample sizes based on local linear method $\overline{n_{LL}^*}$ show less amount of over sampling. Whereas, average sample sizes based on Nadaraya-Watson estimation method $\overline{n_{NW}^*}$ are undersampling.

5.4.2 Random Design

Simulations are performed using the

- Model I: $y = m(x) + \varepsilon \; = \; \sqrt{4x+3} + \varepsilon$
- Model II: $y = m(x) + \varepsilon \; = \; 2\exp\{\frac{-x^2}{.18}\} + 3\exp\{-(x-1)^2/.98\} + \varepsilon$

where $\varepsilon \sim N(0, \sigma^2)$ with $\sigma^2 = 0.25$.

In both cases, 15000 simulation replications are carried out to obtain the final sample sizes required to estimate $m(x)$ at $x_0 = 0.756$ for a given fixed-width, $2d$. Values for half-width of the interval are chosen to be $d = 0.07, 0.09, 0.11, 0.13, 0.15$. We obtain random samples of $\{X_i\}_{i=1}^{25}$ from uniform distribution $X_i \sim U(0, 1)$ and then calculate corresponding y_i for each stated relation (Models I and II). Random errors ε_i are generated from Normal distribution $N(0, 0.5^2)$ and added to the above y_i to get Y_i.

It is clear from the simulation results that the average sample size estimated using two-stage sequential procedure with the application of bootstrap are very close to the optimum values whereas, the average sample sizes $\bar{n}$ calculated using two-stage procedure are very high in comparison to the optimal sample size n_{opt}. Although both local linear $\overline{n^*_{LL}}$ and Nadaraya–Watson methods $\overline{n^*_{NW}}$ record smaller average sample sizes compared to $\bar{n}$, average sample sizes based on local linear method $\overline{n^*_{LL}}$ are very close to the optimal sample sizes n_{opt} in both models.

Local linear method has near or above nominal coverage probabilities in most of the d values. Coverage probabilities of Nadaraya–Watson estimator $\tilde{p}_{NW}$ for Model I decrease with decreasing d whereas model II is not. This is consistent with both procedures i.e. with and without bootstrapping.

The performance of Nadaraya–Watson estimator worsens $\hat{m}_{NW}$ as x increases as its bias highly depends on derivatives of $m(x)$. For the interior point $x_0 = 0.756$, the Nadaraya–Watson estimator assigns symmetric weights to both sides of $x_0 = 0.756$. For a random design this will overweigh the points on left hand side and hence creates a large bias. In other words, Nadaraya–Watson estimator is not design-adaptive. However, local linear method assigns asymmetrical weighting scheme while maintaining the same type of smooth weighting scheme as Nadaraya–Watson estimator. Hence local linear method adapts automatically to this random design.

Using the theoretical and simulation results presented here, we conclude that the bootstrapping will reduce the oversampling of the two-stage sequential procedure significantly while constructing the fixed-width confidence interval for unknown regression function at a given point using local linear method.

Figure 5.2 reflects the over sampling percentage ($\%Over$) of average sample sizes compared to optimum sample sizes for each half-width of the interval d. Bootstrap average sample sizes $\overline{n^*_{LL}}$ and $\overline{n^*_{NW}}$ from both methods (local linear and Nadaraya–Watson) are oversampling. However, $\overline{n^*_{LL}}$ shows less amount compared to other two average sample sizes $\bar{n}$ and $\overline{n^*_{NW}}$.

Table 5.2: Empirical coverage of LL and NW Estimators with and without Bootstrapping: Random Data for $\alpha = 0.05$.

d	n_{opt}	$\bar{n}$	$\bar{n}^*_{LL}$	$\bar{n}^*_{NW}$	$\overline{\hat{m}_{LL}(x_0)}$			$\tilde{p}_{LL}$			$\overline{\hat{m}_{NW}(x_0)}$			$\tilde{p}_{NW}$		
					n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	n^*_{NW}	n	n^*_{LL}	$\bar{n}^*_{NW}$
	$\mathbf{m(x) = \sqrt{4x+3};\quad m(x_0) = 2.454}$ at $\mathbf{x_0 = 0.756}$															
.15	85.1	123.6 .51	87.7 .44	98.9 .480	2.447 .001	2.448 .001	2.449 .001	.970 .001	.939 .001	.952 .002	2.422 .001	2.429 .001	2.427 .001	.967 .002	.933 .002	.947 .002
.13	141.8	203.8 .84	144.8 .75	163.5 .817	2.450 .001	2.450 .001	2.451 .001	.980 .001	.957 .001	.966 .002	2.411 .001	2.420 .001	2.418 .001	.964 .002	.942 .002	.949 .002
.11	257.5	371.1 1.54	262.3 1.35	296.6 1.477	2.450 .000	2.451 .001	2.451 .000	.990 .001	.968 .001	.975 .001	2.408 .000	2.413 .000	2.411 .000	.960 .002	.935 .002	.944 .002
.09	527.3	756.5 3.11	534.7 2.74	605.3 2.973	2.450 .000	2.451 .001	2.451 .000	.993 .001	.983 .001	.988 .001	2.402 .000	2.405 .000	2.404 .000	.936 .002	.913 .002	.919 .002
.07	1293.8	1864.7 7.71	1312.9 6.72	1488.2 7.334	2.451 .000	2.451 .000	2.451 .000	.998 .000	.991 .000	.995 .001	2.403 .000	2.401 .000	2.401 .000	.847 .003	.791 .003	.805 .003
	$\mathbf{m(x) = 2\exp(-x^2/.18) + 3\exp(-(x-1)^2/.98);\quad m(x_0) = 2.907}$ at $\mathbf{x_0 = 0.756}$															
.15	85.1	116.8 .50	87.8 .45	99.4 .489	2.931 .001	2.933 .002	2.933 .001	.961 .002	.925 .002	.939 .002	2.927 .001	2.930 .001	2.929 .001	.966 .002	.930 .002	.945 .002
.13	141.8	194.2 .82	145.2 .72	164.8 .785	2.926 .000	2.929 .002	2.928 .001	.972 .001	.945 .002	.956 .002	2.921 .000	2.924 .001	2.923 .001	.979 .001	.952 .002	.964 .002
.11	257.5	348.0 1.49	261.0 1.34	296.2 1.466	2.925 .000	2.926 .002	2.925 .000	.981 .001	.960 .002	.971 .001	2.919 .000	2.920 .000	2.919 .000	.987 .001	.969 .001	.978 .001
.09	527.3	716.2 3.07	537.8 2.76	611.0 3.038	2.923 .000	2.924 .001	2.923 .000	.989 .001	.973 .001	.980 .001	2.916 .000	2.917 .000	2.919 .000	.994 .001	.982 .001	.988 .001
.07	1293.8	1750.5 7.5	1310.2 6.7	483.4 7.245	2.919 .000	2.921 .001	2.920 .000	.995 .001	.983 .001	.989 .001	2.911 .000	2.913 .000	2.912 .000	.998 .000	.993 .001	.995 .001

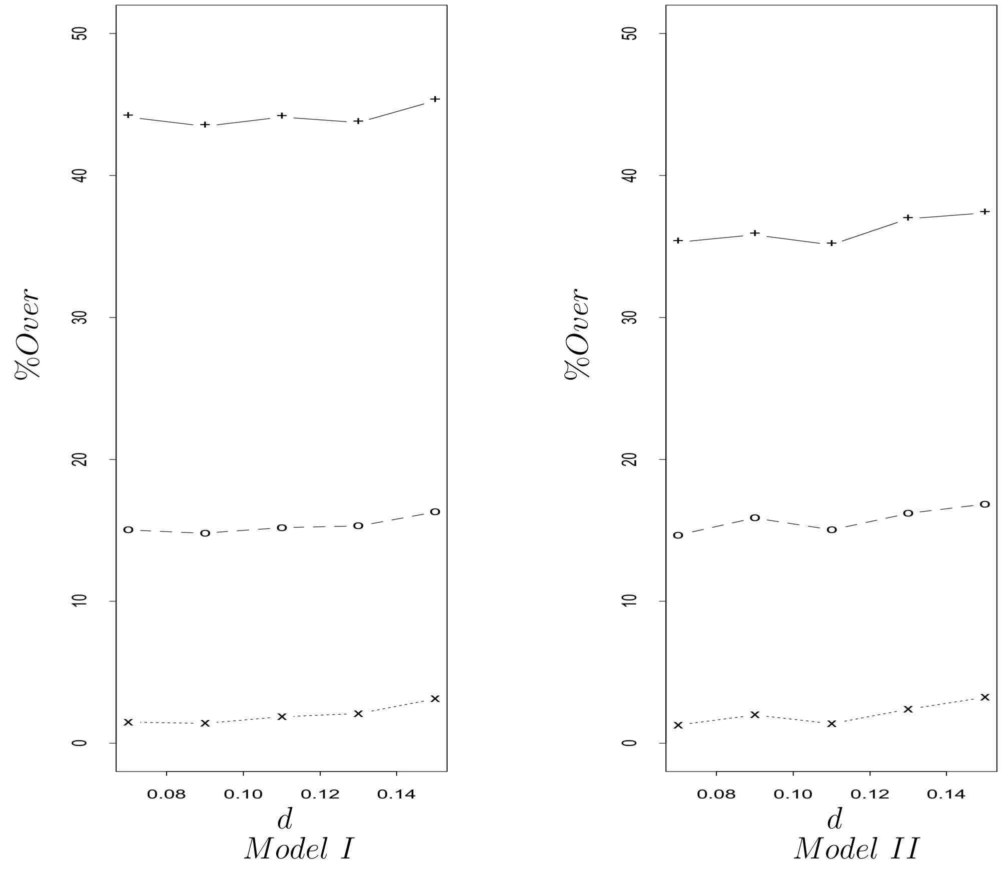


Figure 5.2: Over % Vs Half-Width of the Interval, d:

Random Data.

(*Legend*: '+' = $(\frac{\bar{n}-n_{opt}}{n_{opt}})100\%$, 'x' = $(\frac{\overline{n_{LL}}^{*}-n_{opt}}{n_{opt}})100\%$, 'o' =$(\frac{\overline{n_{NW}}^{*}-n_{opt}}{n_{opt}})100\%$).

From the practical point of view, the focus is mostly on final sample sizes as close as possible to optimal sample size n_{opt} with a reasonable coverage probability. Therefore, we conclude that bootstrap confidence bands based on local linear method is more desirable and satisfy the required goal of this study.

Chapter 6

Application of Sequential Nonparametric Curve Estimation

It is desirable to estimate unknown regression function over some known range of values of explanatory variable. The construction of confidence intervals extends the use of nonparametric smoothing beyond its role as a point estimator often constructed with the sole purpose of giving visual information on the shape of the underlying regression curve. It would be very helpful to obtain through confidence intervals an impression of the variability of the estimator providing a useful scale against which unusual features of the estimated curve may be assessed. The nonparametric kernel regression estimation method developed has a wide application in the estimation of curves or surfaces where no parametric regression models are known.

Here we consider the problem of sequentially selecting bivariate data points (X_i, Y_i); $i = 1, \ldots, n$ for a nonparametric regression curve estimation. Required number of observations or measurements to estimate the underlying regression curve are chosen on the basis of past observations or observations from a pilot sample. Hence, fewer observations or measurements may be required to obtain some specific accuracy compared to those of fixed sampling procedures. The value of using these sequential procedures to nonparametric curve fitting should be obvious. Observations of interest are not always easy to collect. In addition, there is a high risk and cost involved in using outdated data in forecasting and prediction, and most importantly is the time taken for collecting data. As a result, any procedure that will assist data analysts to estimate the sample size required to achieve a good prediction of their nonparametric regression curves will be a useful tool to have.

In this chapter, a sequential procedure, which is adapted from Stein's two-stage procedure (Section 4.3.1 and Section 4.4.1), will be employed to obtain fixed-width confidence interval of unknown regression function. Two-stage sequential procedure is a simple and attractive procedure: select a sample size that one can afford to begin with n_0, and then compute final sample size N. If final sample size equals to size of pilot sample size i.e. $N = n_0$, stop sampling; otherwise, add more samples to make up the difference $N - n_0$. We take $n_0 = 25$ as this choice seems to work well in the simulation study given in Section 4.6. By construction, the value N guarantees that the confidence level (4.7) is satisfied with the estimators achieving their desired accuracy. Also we use standard normal kernel $K(\cdot)$, for all computations. This sequential nonparametric kernel curve estimation method also requires a selection of bandwidth for the kernel regression estimates as well as sample size consideration. We proceed in a similar manner as explained in Chapter 4 by selecting a value for r which suits for all the points of interest in making an estimation. As it is essential to be consistent with the model assumptions stated in Chapter 4 i.e. values of explanatory variable is within 0 and 1 ($X \in [0,1]$), we pre-processed design points to be within $(0,1)$. That is, for a sample of size n, we define $x_i = (u_i - a)/(b - a)$ where $u_1, \ldots, u_n$ are sample values, $a = \min(u_1, \ldots, u_n)$ and $b = \max(u_1, \ldots, u_n)$ which leads to $x_i \in [0,1]$. Also where necessary, we can revert x_i into u_i by taking $u_i = x_i(b - a) + a$.

In Section 6.1, we apply nonparametric kernel curve fitting to four real software data taken from four major releases of a software product (Wood, 1996) and compare the estimated values obtained from the local linear and Nadaraya–Watson methods. From the results obtained, it is suggested that a much more accurate prediction of software reliability growth curve values will be obtained if one used either Nadaraya–Watson or local linear predicted values as predictor. As expected, it can be seen that the fixed evenly spaced design requires a substantially larger sample sizes to attain the given degree of accuracy.

In Section 6.2, we use kernel regression procedure to estimate row average intensity of a digital photo of Leonardo da Vinci's painting, "Mona Lisa" for a given row of the image. We also employ two-stage sequential procedure to compute final sample size required in estimation which guarantees a preassigned accuracy.

In Section 6.3, we develop a nonparametric kernel regression approach using the smallest possible sample in estimating capital asset pricing model (CAPM) when the underlying assumptions (most importantly assumption of existence of linear relationship) fails, within a given level of accuracy. The CAPM can be useful for applications requiring a measure of expected returns. Some applications include cost of capital estimation, portfolio performance evaluation and even-study analysis. The observed results appear to be promising as the period of the statistical analysis should be more desirable to be as short as possible.

Note that in the first two applications given in Sections 6.1 and 6.2, data points of explanatory variable of interest are in the form of fixed equally spaced design whereas application given in Section 6.3 corresponds to random design data.

6.1 Fixed Equidistant Design

6.1.1 Application 1

In software reliability analysis, a standard approach of assessing the reliability of a piece of software is to plot the cumulative number of failures observed during testing, $N(t)$, against execution or calendar time t. It is anticipated that with prolong testing, there will be a growth in the number of faults uncovered. Starting about three decades ago with the seminal work by Jelinski and Moranda (1972), there have been many models, aptly named Software Reliability Growth Models (SRGMs), which have been proposed to fit software failure data to the growth curve $m(t) = \mathbf{E}[N(t)]$. For a review of some of these models, we refer the reader to Pham (2000). Unfortunately, many of these SRGMs are very complex and standard estimation techniques, such as Maximum Likelihood (ML) or least squares methods, fail to estimate the parameters of these models accurately if at all. In this section, we investigate the potential benefits of using nonparametric kernel regression methods to fit SRGMs i.e. Nadaraya–Watson estimator $\hat{m}_{h_n,NW}$ and the local linear estimator $\hat{m}_{h_n,LL}$.

Wald A. based his work on the philosophy that when testing for product defect, the decision on when to stop sampling should depend on the evidence accumulated by the experimenter. In estimation, a sequential approach would involve repeated sampling, usually twice, with successive sample added to the samples already selected, terminating when a desired level of error of estimation is reached. This approach can be used to great advantage in software reliability analysis, where it is often expensive and time consuming to obtain test cases; therefore a rule of determining the optimal sample size to achieve a fixed level of accuracy in estimating SRGMs would be of value.

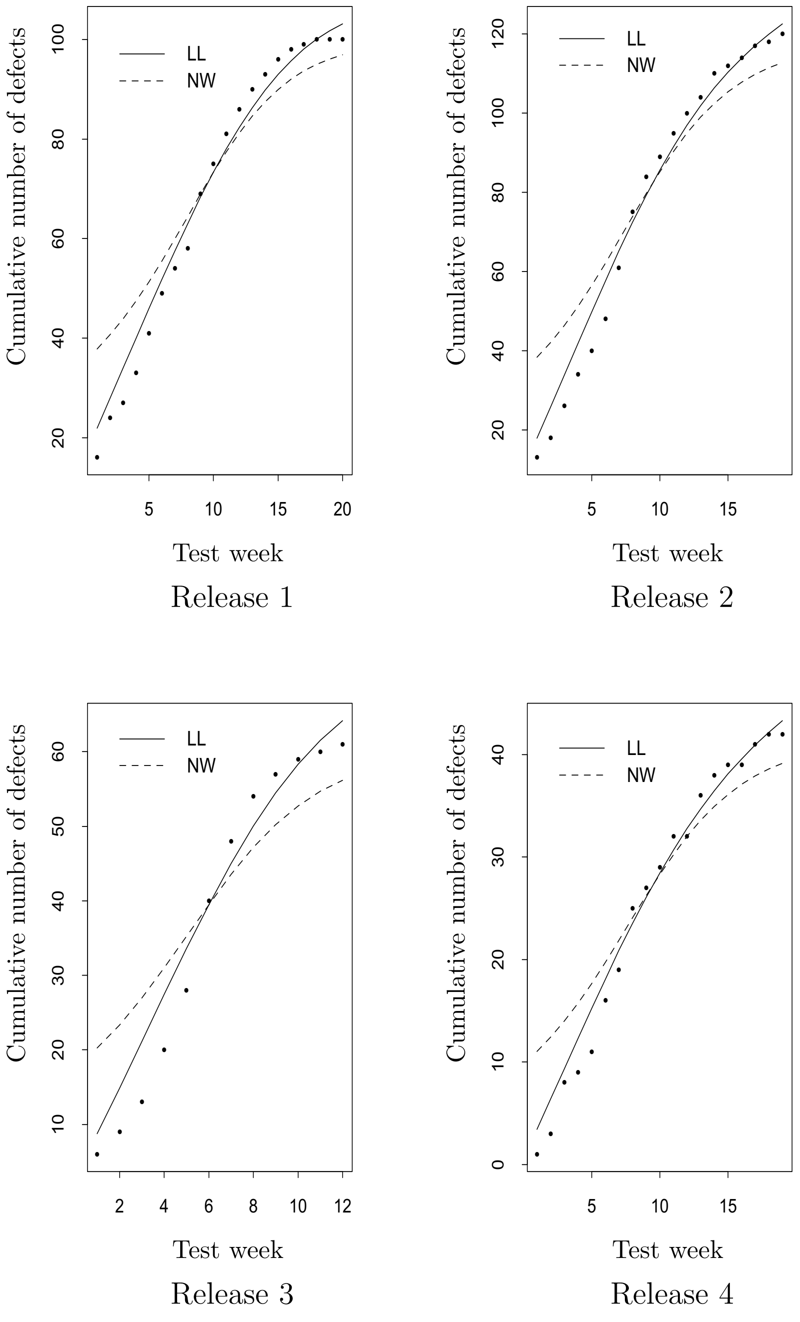
Numerical Examples

In this section, we will apply nonparametric kernel regression estimators to four sets of software failure data taken from Wood (1996). These data sets arise from tests performed on four major releases at Tandem Computers. To avoid confidentiality issues, these data have been transformed from the original data (refer to details of this transformation in Wood, (1996). Table 6.1 below displays the data where the numbers under the column headed by Release i, $i = 1, 2, 3, 4$ are the cumulative number of failures detected at the end of each Test Week. The graphs obtained using the Nadaraya–Watson and local linear estimators to fit the four sets of data are displayed in Figure 6.1. It is apparent from the figures that local linear estimator provides much better fit to the cumulative failure data than Nadaraya–Watson estimator. This observation is confirmed by the Mean-Squared Error (MSE) value table below.

We next compare the predicted values made by the $\hat{m}_{h_n,NW}$ and $\hat{m}_{h_n,LL}$ estimators at some point $x = x_0$. Our approach is to use the defect data records of the first n_0 weeks to predict the value for week $n_0 + 1$,

Table 6.1: Cumulative Defect Data Based on Four Software Releases.

Test Week	Release 1	Release 2	Release 3	Release 4
1	16	13	6	1
2	24	18	9	3
3	27	26	13	8
4	33	34	20	9
5	41	40	28	11
6	49	48	40	16
7	54	61	48	19
8	58	75	54	25
9	69	84	57	27
10	75	89	59	29
11	81	95	60	32
12	86	100	61	32
13	90	104		36
14	93	110		38
15	96	112		39
16	98	114		39
17	99	117		41
18	100	118		42
19	100	120		42
20	100			



Release 1

Release 2

Release 3

Release 4

Figure 6.1: SRGM Curve Fitting Using Nadaraya–Watson (NW) and Local Linear (LL) Estimators.

Table 6.2: Mean-Squared Error (MSE) Values.

$NW = \hat{m}_{h_n,NW}$ and $LL = \hat{m}_{h_n,LL}$

	Release 1		Release 2		Release 3		Release 4	
	NW	LL	NW	LL	NW	LL	NW	LL
MSE	85.93	14.62	149.60	26.53	81.83	19.92	21.00	3.60

then add this prediction to the original set of records to predict the value for week $n_0 + 2$ and so on. Table 6.3 to Table 6.6 display these predicted values, based on Nadaraya–Watson and local linear estimators, when $n_0 = 13, 14, 9$ and 14 for Release $1, 2, 3$ and 4 data respectively. We also compare between the predictions using MSE values and it appears that Nadaraya–Watson estimator tend to produce more accurate predicted values than local linear estimator. In conjunction with the results in Table 6.2 therefore, it seems that $\hat{m}_{h_n,LL}$ estimator does a better job at *interpolating* data whereas $\hat{m}_{h_n,NW}$ does better at *extrapolating* them. Whether this is true in general remains to be investigated.

Another interesting observation from Table 6.3 to Table 6.4 is that $\hat{m}_{h_n,NW}$ estimator always under-estimate the observed values whereas $\hat{m}_{h_n,LL}$ estimator always over-estimate them. Therefore, if we predict using the *average* of the two predictors, i.e.

$$\hat{m}_{AV}(t) = \frac{\hat{m}_{h_n,NW}(t) + \hat{m}_{h_n,LL}(t)}{2}, \tag{6.1}$$

we would expect the resulting predictions to be very close to the true values. This is confirmed by the results in the two tables. Therefore, it is highly recommended that $\hat{m}_{AV}(t)$ be used to predict SRGM values.

In the next example, we illustrate our two-stage sequential procedure discussed in Section 4.3 by applying it to an exponential SRGM (cf. Goel

Table 6.3: Predicted Values - Release 1.

Release 1 ($n_0 = 13$)				
Test week t	True value	$\hat{m}_{h_n,NW}(t)$	$\hat{m}_{h_n,LL}(t)$	$\hat{m}_{AV}(t)$
14	93	89.07	94.19	91.63
15	96	89.19	98.36	93.78
16	98	89.18	102.53	95.86
17	99	89.18	106.69	97.94
18	100	89.18	110.86	100.02
19	100	89.18	115.03	102.11
20	100	89.18	119.20	104.19
MSE		83.89	114.16	4.94

Table 6.4: Predicted Values - Release 2.

Release 1 ($n_0 = 13$)				
Test week t	True value	$\hat{m}_{h_n,NW}(t)$	$\hat{m}_{h_n,LL}(t)$	$\hat{m}_{AV}(t)$
15	112	104.07	114.99	109.53
16	114	104.80	120.15	112.48
17	117	105.03	125.32	115.18
18	118	105.07	130.49	117.78
19	120	105.07	135.66	120.37
MSE		136.18	103.44	2.39

Table 6.5: Predicted Values - Release 3.

Release 3 ($n_0 = 13$)				
Test week t	True value	$\hat{m}_{h_n,NW}(t)$	$\hat{m}_{h_n,LL}(t)$	$\hat{m}_{AV}(t)$
10	59	56.27	60.55	58.41
11	60	56.36	64.04	60.20
12	61	56.35	67.53	61.94
MSE		14.11	20.46	0.42

Table 6.6: Predicted Values - Release 4.

Release 4 ($n_0 = 14$)				
Test week t	True value	$\hat{m}_{h_n,NW}(t)$	$\hat{m}_{h_n,LL}(t)$	$\hat{m}_{AV}(t)$
15	39	37.51	40.32	38.92
16	39	37.57	42.61	40.09
17	41	37.57	44.90	41.24
18	42	37.57	47.18	42.38
19	42	37.57	49.47	43.52
MSE		11.06	22.52	0.74

and Okumoto, 1979) defined by

$$Y = 20(1 - e^{-.2t}) + \varepsilon$$

where $\varepsilon \sim N(0, 0.25)$. Note that in this case, explanatory variable takes equally spaced design points. We will evaluate the performance of the procedure based on an initial sample size $n_0 = 25$ to test the accuracy of estimating $m(t_0) = 2.0115$ at $t_0 = 0.53$. Fixed-width confidence intervals defined by (4.3) at level $\alpha = 0.05$ are obtained for $d = 0.10, 0.11, 0.12, 0.15$ and 0.17 using the steps of the procedure. Altogether, 15000 simulations were performed to obtained some of the following statistics which are displayed in Table 6.7:

- Optimal sample size n_{opt} given by equation (4.8);
- average final sample size $\bar{n}$ where N is given by equation (4.14);
- average $\bar{T}$ value where T is given by equation (4.15);
- average Nadaraya–Watson estimated value $\overline{\hat{m}_{h_n,NW}(t_0)}$ at $t_0 = 0.53$ using equation (2.18);
- average local linear estimated value $\overline{\hat{m}_{h_n,LL}(t_0)}$ at $t_0 = 0.53$ using equation (2.19);
- coverage probabilities $\tilde{p}_{NW}$ and $\tilde{p}_{LL}$, i.e. proportion of intervals defined in (4.3) which contain $m(t_0)$.

(In Table 6.7, figures enclosed in brackets under estimated values refer to their standard errors.)

As would be expected, the optimal sample size that is required to produce a 95% confidence interval increases as d decreases. Comparing the values

Table 6.7: Summary of Results for Two-Stage Sequential Procedure at $t_0 = 0.53$ and $n_0 = 25$.

Statistics	Values				
d	0.17	0.15	0.12	0.11	0.10
n_{opt}	73.98	119.72	282.42	394.68	569.44
$\bar{n}$	130.75	202.93	455.69	624.14	895.76
	(0.7062)	(1.1316)	(2.6775)	(3.6971)	(5.4058)
$\bar{T}$	4.36	6.76	15.19	20.80	29.86
	(0.0235)	(0.0377)	(0.0892)	(0.1232)	(0.1802)
$\overline{\hat{m}_{h_n,LL}(t_0)}$	1.9828	1.9822	1.9828	1.9823	1.9825
	(0.0004)	(0.0004)	(0.0002)	(0.0002)	(0.0002)
$\overline{\hat{m}_{h_n,NW}(t_0)}$	1.9016	1.8949	1.8888	1.8867	1.8857
	(0.0004)	(0.0004)	(0.0002)	(0.0002)	(0.0002)
$\tilde{p}_{LL}$	0.9905	0.9917	0.9947	0.9955	0.9972
	(0.0008)	(0.0007)	(0.0006)	(0.0005)	(0.0004)
$\tilde{p}_{NW}$	0.8920	0.8035	0.4447	0.2413	0.0997
	(0.0025)	(0.0032)	(0.0041)	(0.0035)	(0.0024)

of $\bar{n}$ with n_{opt} indicates the phenomenon of oversampling, where $\bar{n} > n_{opt}$, which is common with two-stage procedure. Both Nadaraya–Watson and local linear methods underestimate the true value of $m(t_0)$ with local linear producing more accurate values. Finally, it is significant that local linear method produces far superior coverage probabilities, i.e. closer to 0.95 than Nadaraya–Watson.

6.1.2 Application 2

The aim of this application is to explain how we can employ nonparametric kernel regression estimation to estimate row average intensity of a digital photo of Leonardo da Vinci's painting, "Mona Lisa" for a given row number of the image using the smallest possible sample size for a pre-assigned accuracy.

The data were measured as arithmetic average of the values in each row of the image. These row averages can be used to correct for lighting effect especially when there is a top-to-bottom lighting variation. In that case, robust smoothing of row averages may be a good way to estimate the lighting effect.

Initial data set consists of average intensities measured at each of 425 rows. Thus, the response variable is row average intensity and the explanatory variable is the row number. The given row numbers are ordered nonrandom numbers of the form $|x_{i+1} - x_i| = 1$ for all i where $i = 1, \ldots, 424$. Hence, we consider an fixed equally spaced design points and two-stage stopping rule developed in Section 4.3 is employed. The explanatory variable, row number was rescaled to be within $(0, 1)$ to comply with our data design using a relation $\left\{\frac{x_i - a}{b-a}\right\}$; $i = 1, \ldots, 425$ where $a = 1$ and $b = 425$.

Since two-stage sequential procedure initiates with taking an initial sample of size n_0, we take 25 bivariate data points as our pilot sample i.e. $n_0 = 25$.

Half width of the interval d, is chosen to be 2.5 as a range of row average intensities is $\in (72, 225)$. The performance of two-stage procedure is examined for two confidence coefficients $\{1 - \alpha_i\}_{i=1}^{2}$ where $\alpha_1 = 0.1$ and $\alpha_2 = 0.05$. Then using sampling stopping rule given in (4.14), we determine final sample size N for both $\alpha = 0.05, 0.10$. Finally, $\hat{m}_{h_N,LL}(x_i)$ and $\hat{m}_{h_N,NW}(x_i)$ are estimated at each design point x_i using $\{x_i, Y_i\}_{i=1}^{N}$.

Figure 6.2 shows the Nadaraya–Watson and local linear kernel regression estimates of row average intensity of a digital photo of Leonardo da Vinci's painting, "Mona Lisa" for a given row of the image. The local linear regression estimation shown by the solid line and the nonparametric regression estimate of Nadaraya–Watson shown by the dotted line. In both graphs, there is no noticeable difference between local linear and Nadaraya–Watson estimators as both final sample sizes $N_{\alpha=.05} = 250$ and $N_{\alpha=.10} = 150$ are fairly large. Both final sample sizes are able to highlight an important structure in the original data hence, produce a better estimate of average intensity for a given row number.

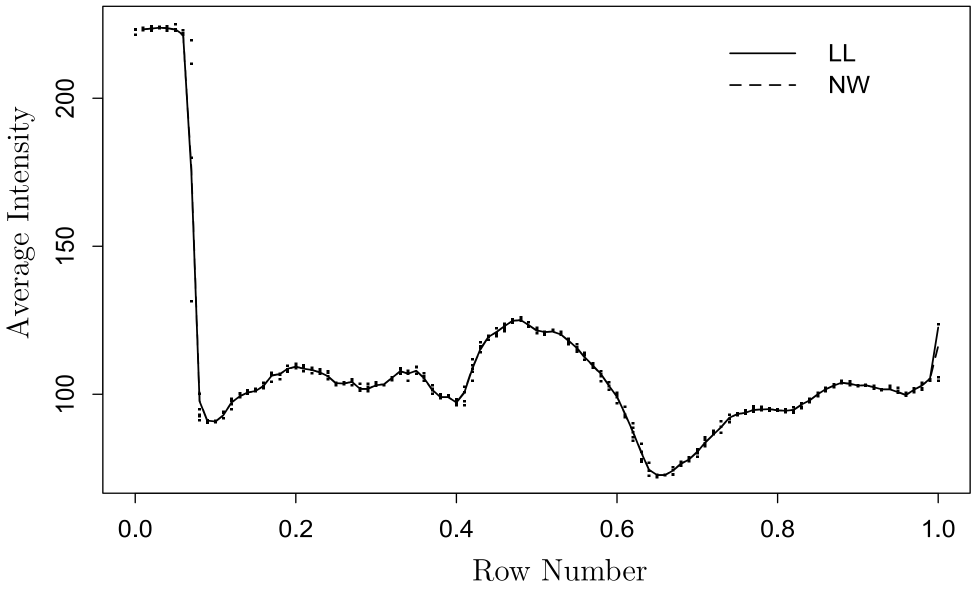


Fig 6.2A: Final sample size $N = 250$ for $d = 2.5$, $\alpha = 0.05$

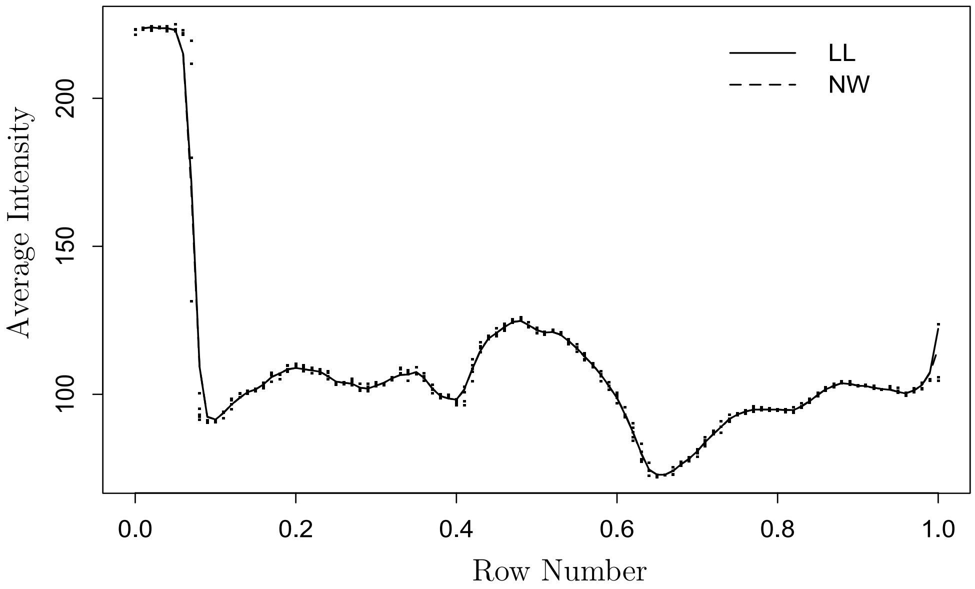


Fig 6.2B: Final sample size $N = 150$ for $d = 2.5$, $\alpha = 0.10$

Figure 6.2: Nonparametric Kernel Regression Estimation of Row Averages of a Digital Photo of “Mona Lisa”.

6.2 Random Design

6.2.1 Application 3

The capital asset pricing model (CAPM) implies that the expected return of an asset must be linearly related to the covariance of its return with the return of the market portfolio. The development of CAPM helps economists to quantify risk and reward for bearing risky investments. Markowitz's (1959) mean-variance portfolio theory laid the groundwork for the CAPM. Sharpe (1964) and Lintner (1965b) extend Markowitz's work to develop economy-wide implications. The usual CAPM equation is a direct implication of the mean-variance efficiency of market portfolio. The CAPM assumes the existence of lending and borrowing at a risk free rate of interest. Under this assumption the CAPM we have for the expected return of asset i,

$$\mathbf{E}[R_i] = R_f + \beta\,(\mathbf{E}[R_m] - R_f)\,; \qquad \beta = \frac{\mathbf{Cov}[R_i, R_m]}{\mathbf{Var}[R_m]} \tag{6.2}$$

where R_i is the return of asset i, R_m is the return on the market portfolio, R_f is the return on the risk free asset. β is defined as the gradient of the least squares linear regression where the excess return on the market over the risk-free rate is the predictor and the excess return on the asset over the risk-free rate is the response variable. However, the possibility that there exist a nonlinear relationship between the excess returns of an asset and a market is justified in the discussions in Long (1990), Luenberger (1993, 1998) and Efromovich (2004). This section explores a statistical analysis of historical data and develops a nonparametric kernel regression estimation of the CAPM that can be used when the underlying assumptions given in Campbell (1997) fail.

Generally a fixed sample size is used to calculate the β 's. The sample size may be too large for some periods of time and too small for others. The larger the period the more outdated is the information that is being used. The current situation in the Australian market is different from what it was in the early 1990's. The period of the statistical analysis should be as short as possible to minimize the effect of factors such as size of the institution, dividend per share, business environment, union actions etc. This is why sequential analysis appears to be promising as its primary goal is to achieve a given accuracy by using the smallest possible sample size. Efromovich (2004) proposed a method of finding the optimal stopping time based on the empirical risk approximation procedure suggested in Chaudhuri et. al. (1997) and Efromovich (1989, 1994, 1995). Here we proposed the use of two-stage sequential procedure along with nonparametric kernel regression estimation which is explained in Section 4.4. The suggested procedure allows an investor to analyse the relationship between the excess rate of returns on an asset i $(R_i - R_f)$ and the excess rate of returns on the market $(R_M - R_f)$ using the shortest period of historical data without any assumption on the underlying distribution.

Let us examine now why (6.2) is called a pricing model. Suppose that an asset is purchased at price P_0 and later sold at price P_1. The rate of return is then $R = \frac{(P_1 - P_0)}{P_0}$. Here P_1 is random so the CAPM implies

$$\begin{aligned} P_0 &= \frac{\mathbf{E}[P_1]}{1 + \mathbf{E}[R]} \\ &= \frac{\mathbf{E}[P_1]}{1 + R_f + \beta(R_m - R_f)}, \end{aligned} \tag{6.3}$$

where $\mathbf{E}[R]$ is given by (6.2).

Suppose that we observe n_0 pairs $\{(r_{M_1}, r_{A_1}), \ldots, (r_{M_{n_0}}, r_{A_{n_0}})\}$ where $r_{M_i} = R_{M_i} - R_{f_i}$ is the excess rate of return from the market during the i^{th} period and $r_{A_i} = R_{A_i} - R_{f_i}$ is the excess rate of return from an asset during the i^{th} period. Then, the regression model we consider here is of the form:

$$r_{A_i} = m(r_{M_i}) + \varepsilon_i; \quad i = 1, \ldots, n_0 \tag{6.4}$$

where $m(\cdot)$ is a regression function and errors ε_i are independent and identically distributed with zero mean $\mathbf{E}[\varepsilon_i] = 0$ and constant variance $\mathbf{Var}[\varepsilon_i] = \sigma^2$.

The methodology developed in Section 4.3.2 has been applied to estimate CAPM using the excess rate of monthly returns of the Microsoft stock (r_{A_i}) and the excess rate of monthly returns of the market (r_{M_i}) whose proxy is Standard and Poor's 500 index which is a capital-weighted portfolio of most of the United State's largest stocks. The 13-weeks Treasury bill serves as the proxy for the risk-free asset R_f.

We begin the analysis of data with visualization of a scatter diagram of the original data. Figure 6.3 exhibits a scatter diagram for the excess rate of monthly returns of the Microsoft stock versus the excess rate of monthly returns of the market during a 55-month period that ended on May 1, 2007. If the CAPM is correct, then a linear relationship between these two rates with zero y-intercept should be observed. If the model is incorrect or its assumptions are invalid then a more complicated relationship may be visible.

Here we examine whether a classical parametric regression analysis helps in our understanding of this data set. A most commonly used parametric regression model is $r_{A_i} = \alpha + \beta r_{M_i} + \varepsilon_i$. The least squares regression line is shown next in Figure 6.4. The fitted line shows the fact that larger returns from the market imply larger returns from the stock and vice a versa. The parameters of the fitted regression model are as follows: the slope, $\hat{\beta} = 1.4133$ and $\hat{\alpha} = -1.5845$. Standard error of $\hat{\alpha}$, $SE(\hat{\alpha}) = 1.13804$ and corresponding $p - value = 0.016954$ confirms non-zero y-intercept. This result contradicts CAPM since CAPM implies that the intercept α is equal to 0 i.e. $\alpha = 0$.

The nonparametric kernel curve estimation using Nadaraya–Watson and local linear methods are shown in Figure 6.5. As it is essential to be consistent with the model assumptions stated in Chapter 4 which is range of explanatory variable is within 0 and 1 ($X_i \in [0,1]$), we transformed market excess rate to be within $(0,1)$.

Next we examined the two-stage sequential nonparametric approach. As in the case of our simulation study given in Section 4.5.2 we start the sampling procedure with an initial sample of size $n_0 = 30$ with $\alpha = 0.05$ and values being selected for half-width of the interval d, are $d = 0.75, 1.0$. Finally, we determine the final sample size N, using the proposed two-stage stopping rule (4.26). Figure 6.6 and Figure 6.7 display nonparametric kernel estimates i.e. local linear and Nadaraya–Watson estimates for the data of Microsoft stock. The solid lines are either local linear (LL) or Nadaraya–Watson (NW) estimates and dotted lines are least squares linear regression estimates with $\hat{\alpha} = -1.3480$, $\hat{\beta} = 1.2917$ and $\hat{\alpha} = -0.8686$, $\hat{\beta} = 1.3057$ when $d = 0.75$ and $d = 1.0$ respectively.

Nadaraya–Watson curve estimation shows larger absolute values of the market returns imply smaller absolute values of the asset returns compared to those predicted by local linear method. Local linear curve estimation shows smaller values of the market returns imply smaller values of the asset returns and vice a versa. From this, it appears that nonparametric kernel regression estimation can be used in assessing return on risky assets. Rather than relying on beta's disclosed by companies this could be used as an additional analytical tool for investors to know more about assets in their portfolio.

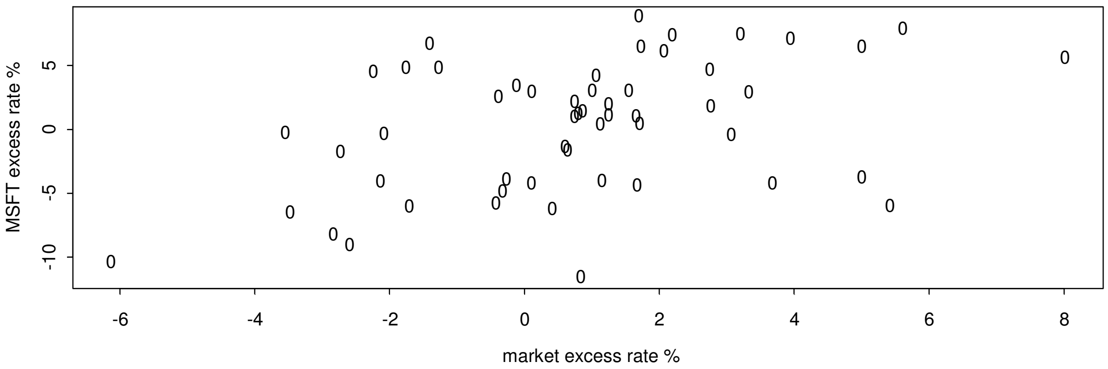


Figure 6.3: Scatter Diagram for Microsoft-SP500 Monthly Data, n =55.

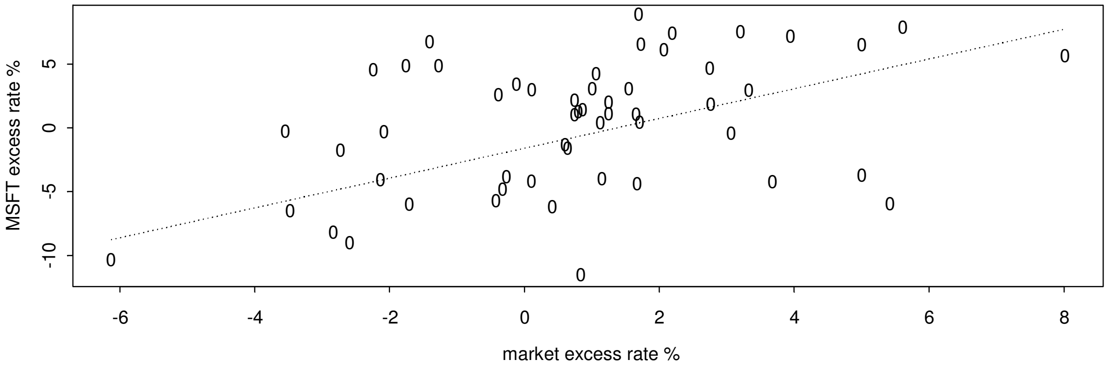


Figure 6.4: Linear Regression, β =1.1672, α = -1.6110, n =55.

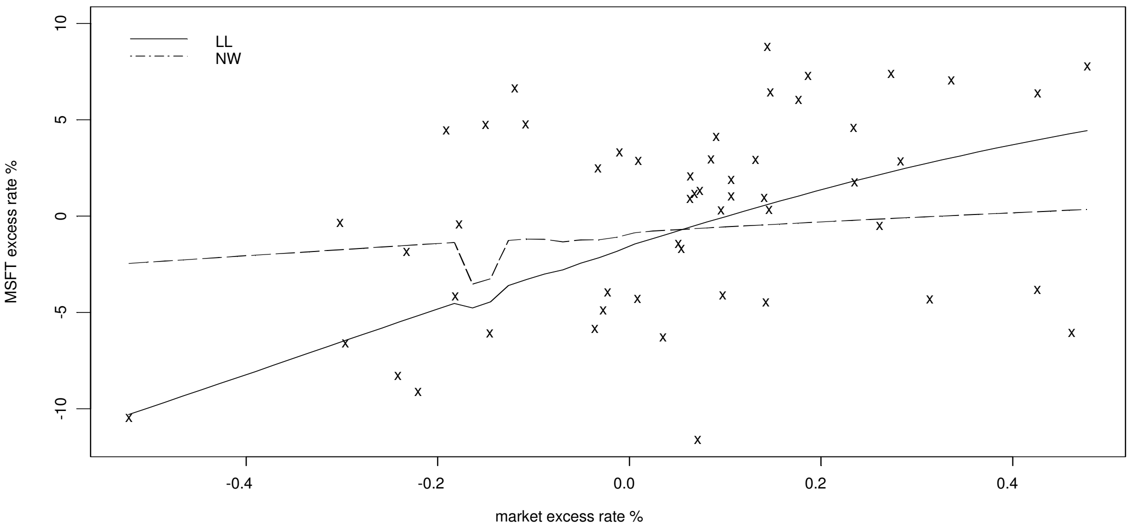
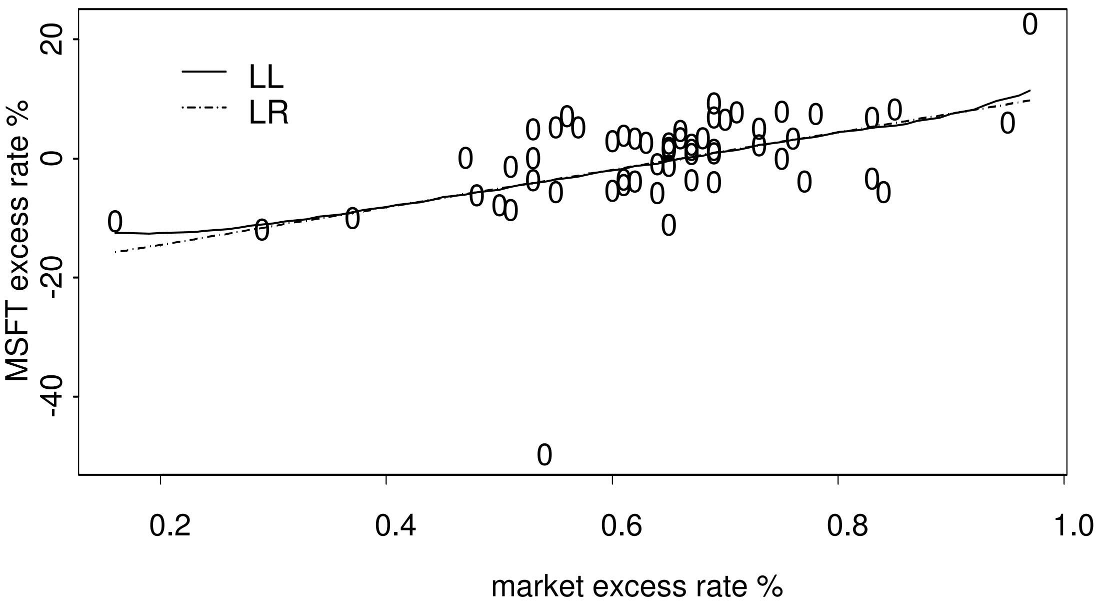


Figure 6.5: Nonparametric Kernel Regression, n =55.



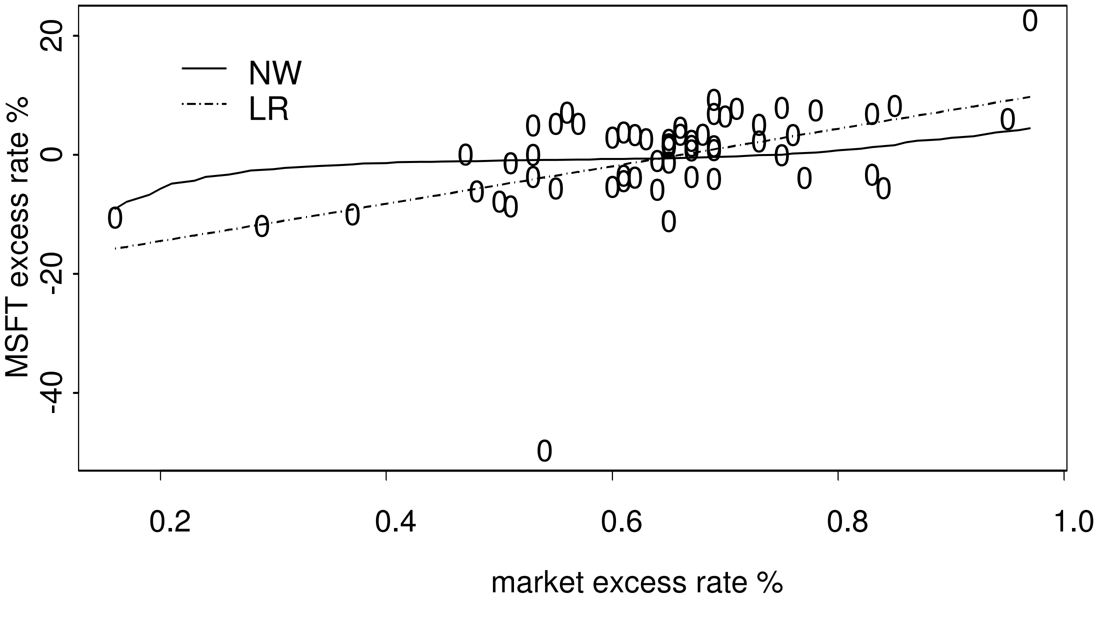


Figure 6.6: Two-Stage Sequential Nonparametric Kernel Regression, d =1.0, n =59.

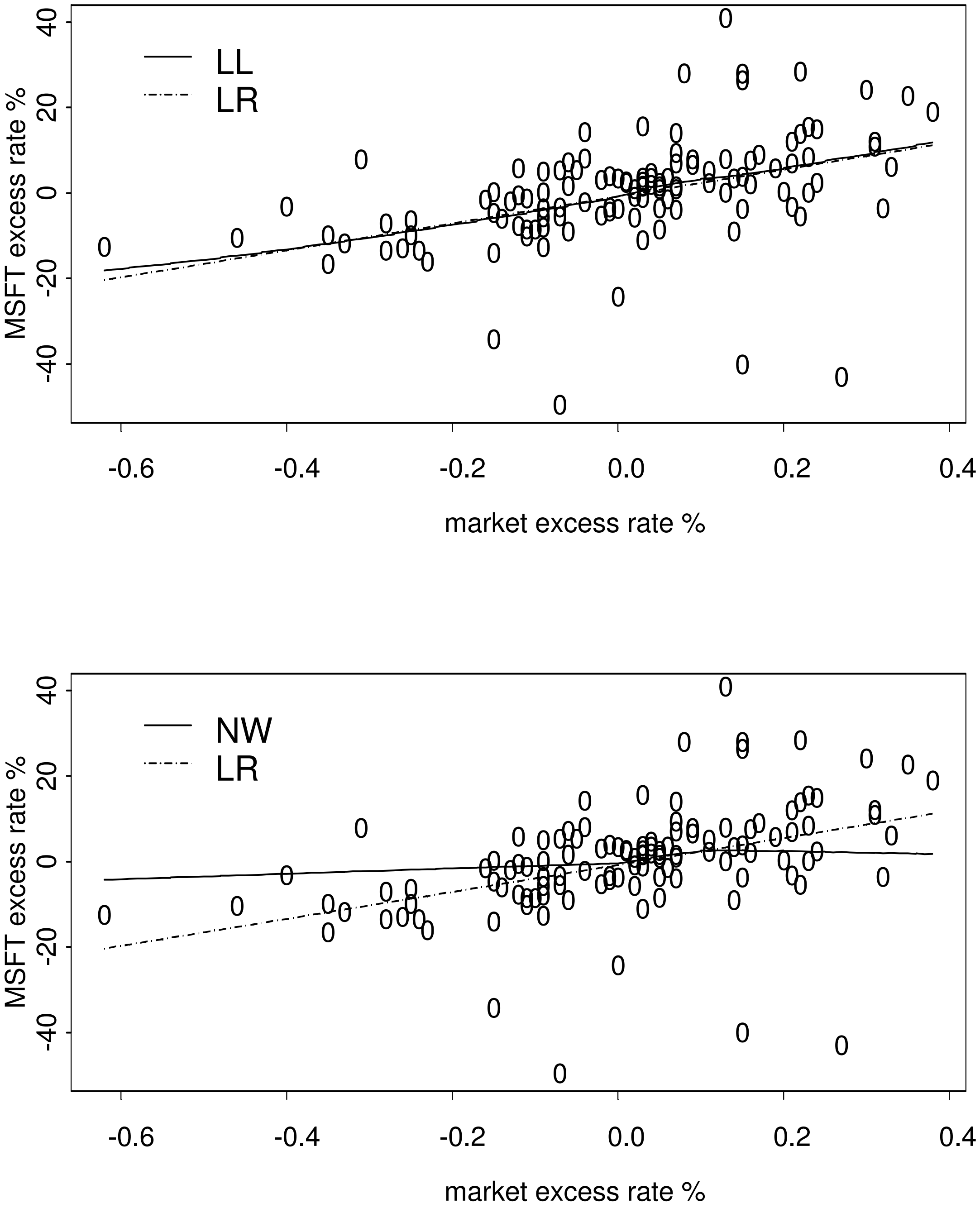


Figure 6.7: Two-Stage Sequential Nonparametric Kernel Regression, d =0.75, n =55.

Chapter 7

Conclusions and Further Work

Nonparametric regression models can be used for the same types of applications such as estimation, prediction, calibration and optimization that traditional regression models are used for. Nonparametric regression techniques have become increasingly popular with practitioners due to the fact that they don't make many assumptions on the distributional form of the underlying distributions, except that they should be reasonably smooth functions, and are also very easy to use. In a nonparametric setting the aim is to produce a reasonable approximation to the unknown function $m(x)$ when we have no precise information about the form of the true regression function, $m(x)$. When using nonparametric regression methods, one of the key objectives is to ensure that the fitted value $\hat{m}(x)$, based on a sample of size n, achieves a reasonably good fit to the true but unknown regression function $m(x)$ at a given point. The main objective of this thesis is to apply data driven sequential approach to analyse nonlinear relationship between two variables using the smallest possible sample size. This is primary goal of sequential analysis which is to achieve a given accuracy by using the smallest possible sample sizes.

Chapter 2 is devoted to a brief introduction of nonparametric regression estimation. Special interest was on investigating the potential benefits of using kernel type nonparametric regression methods namely, Nadaraya–Watson method and local linear method due to their popularity among available kernel type estimators. Both methods estimate regression function as the weighted average using kernel function as a weighting function. Even though a variety of kernel functions are possible in general, practical and theoretical considerations restrict the selection. The weights of the estimators are determined by the bandwidth. Quick and simple bandwidth selector was employed. Properties of these nonparametric regression methods have been explored under both fixed design and random design contexts. It was shown that specification of bandwidth is very important on the performance of each estimator. There is very little to choose between the different kernels on the basis of simulation results. The analysis also showed that local linear method is superior over Nadaraya–Watson method especially in its ability of design adaption which adapted to both random and fixed designs and even to both interior and boundary points. Performance of proposed bandwidth selection method and effect on selecting various kernel functions for local linear and Nadaraya–Watson methods were illustrated via an extensive simulation study.

It is natural to ask what the residual variance estimator σ^2 is when fitting a nonparametric regression function to a data set. Residual variance estimators are broadly divided into difference-based estimators and curve fitting estimators depending on how these estimators are formulated. This task was addressed in Chapter 3 by comparing several estimators of residual variance for different circumstances in terms of different types of error distributions, diverse data design types, different sample sizes and finally, for various type of regression functions. Under difference-based residual variance estimators, estimators proposed by Rice (1984) $\hat{\sigma}_R^2$, Gasser et. al. (1986) $\hat{\sigma}_{GSJ}^2$ and Müller et. al. (2002) $\hat{\sigma}_{MSW}^2$ were considered. Whereas in the case of curve fitting type of residual variance estimator, estimator proposed by Hall and Marron (1990) $\hat{\sigma}_{HM}^2$ was considered. Performances of these estimators were investigated in a simulation study, including a comparison with different cases as previously mentioned. For fixed design points, $\hat{\sigma}_{GSJ}^2$ was the proper choice as $\hat{\sigma}_R^2$ should not be used because it did not always behave well. In particular, if design points are expected to be random, difference-based estimator $\hat{\sigma}_{MSW}^2$ over curve fitting estimator $\hat{\sigma}_{HM}^2$ was preferable as it achieves asymptotic optimal efficiency.

The analytical work in this thesis starts from Chapter 4. We studied data-driven fixed-width confidence bands for nonparametric regression function estimation using local linear and Nadaraya–Watson estimators in both fixed and random design contexts. We considered a nonparametric regression model based on independent and identically distributed pairs of observations (X_i, Y_i); $i = 1, \ldots, n$, where the regression function $m(x)$ is given by $m(x) = E(Y_i \mid X_i = x)$ with one independent variable. We described an estimation procedure of nonparametric regression model at a specified point of the independent variable by some appropriately constructed fixed-width $(2d)$ confidence interval with the confidence coefficient of at least $1-\alpha$. Here, $d(> 0)$ and $\alpha \in (0, 1)$ were two preassigned values. In the case of fixed designed data we employed two-stage and modified two-stage sequential procedures. Whereas for random design regression model, the sample sizes for a preset confidence coefficient were optimized using sequential procedures namely two-stage, modified two-stage and purely sequential procedures. As would be expected, the optimal sample size that was required to produce a given level of confidence interval increased as half width of the interval d decreases. Comparing the values of average sample sizes $\bar{n}$ with optimal sample sizes n_{opt} indicated the phenomenon of oversampling, i.e. $\bar{n} > n_{opt}$, which is common with two-stage procedure. As anticipated almost all average sample sizes $\bar{n}$ under modified two-stage procedure were lower than those of two-stage procedure. However, slightly higher than their corresponding optimum sample sizes n_{opt}. Coverage probabilities of both two-stage and modified two-stage procedures were close or above the preset confidence coefficients 95% and 90%. The performance of the purely sequential procedure was better than that of the two-stage procedure. However operationally, two-stage procedure reduces computational costs associated with the corresponding purely sequential schemes by a substantial margin. It was significant that local linear method produces far superior coverage probabilities, i.e. closer to preset confidence coefficient than Nadaraya–Watson. However, both estimators were shown to have asymptotically correct coverage properties.

In Chapter 5, a bootstrap method was developed to estimate average sample sizes for kernel based nonparametric regression estimation for a given accuracy. The proposed bootstrap technique uses the percentiles of approximate distribution of unknown regression function to construct confidence intervals for the curve at specific design points. Particular attention was devoted to the problem of minimising the amount of oversampling in the two-stage sequential procedure. The numerical results indicated that the confidence bands based on the local linear estimator had the best performance than those constructed by using Nadaraya–Watson estimator. The coverage probability of Nadaraya–Watson method was found to be generally below the preset confidence coefficients. On the other hand, local linear method had near-nominal coverage probabilities in most of the cases. Using the theoretical and simulation results presented, we showed that the bootstrapping reduces the oversampling of the two-stage procedure significantly while constructing the fixed-width confidence interval for unknown regression function at a given point using local linear method. From a practical point of view, mostly the focus was on final sample size as close as possible to optimal sample size with a reasonable coverage probability. Therefore, we concluded that results obtained from local linear method satisfied the required goal of this study.

Finally, in Chapter 6, we employed proposed sequential stopping rules together with nonparametric kernel regression methods to predict the software reliability growth model (SRGM) and to estimate the regression curve of capital asset pricing model (CAPM). A sequential procedure, which is adapted from Stein's two-stage procedure was employed to obtain fixed-width confidence interval. The main advantage of using Nadaraya–Watson method and local linear method in predicting growth of software reliability is that they place minimum requirement on the distributional form of the stochastic process which gave rise to software failure data and hence dispense with the need to estimate parameters from complex models. Numerical examples involving four sets of real software data were presented to illustrate the developed techniques and compared the estimated values obtained from the two nonparametric regression methods. From the results obtained, it is suggested that a much more accurate prediction of SRGM values will be obtained if one used the average of the Nadaraya–Watson and local linear predicted values as predictor. The key advantage of using sequential procedures in CAPM approach was the use of the shortest period of historical data to deduce correct price of a risky asset. This is because there is a price to be paid for the use of outdated information, especially in analysing prices of risky assets due to dynamic nature of financial markets. Parametric estimation of CAPM can only be applied for an analysis of markets in equilibrium. This assumption underlying the CAPM approach holds in a highly volatile security market is very difficult to expect though. Nonparametric regression allows investor to accepted any shape of estimated capital asset pricing model.

Further Work

Optimal estimation of the bandwidth is an interesting and practically important problem. Because the choice of a bandwidth can lead to better results for nonparametric kernel type regression estimators. Besides the quality of estimation of the bandwidth is important for the quality of the fitted regression function even though the bandwidth is an auxiliary quantity. In general practice, the bandwidths of regression estimators are chosen to minimize their asymptotic mean integrated squared error. However, as in sequential framework we employed bandwidth as $h_n = n^{-r}$ for $a < r < b$ where $0 < a, b < 1$. de Silva and Mukhopadhyay (2002) employed bootstrap bandwidth selection method to find optimal value for r in nonparametric kernel density curve estimation. Hence it is appealing to scrutinize relevance of bootstrap method to compute optimal bandwidth such that $h_{n,opt} = n^{-r_{opt}}$ where r_{opt} is optimal value of r in nonparametric regression estimation.

Hall (1981) explored asymptotic theory of three-stage sequential procedure in the case of estimation of sample mean and claimed that the proposed procedure is more efficient than Stein's two-stage procedure as it uses a significantly smaller sample size to achieve a confidence interval with nearly the same coverage probability. Hence it is worthwhile to examine applicability of triple sampling sequential procedure in the context of nonparametric kernel regression estimation.

Even though this study has been confined to the case of univariate design points there are many practical applications where we seek to identify how a response variable Y_i is related to b fixed design variables $x_i = (x_{i1}, \ldots, x_{ib})^T$ or random design variables $X_i = (X_{i1}, \ldots, X_{ib})^T$. This is the multivariate regression analogue of the univariate kernel type regression estimation problem treated in this study. Hence implementation of sequential procedures for constructing a simultaneous fixed-size spherical confidence region $\mathcal{R}$ for regression function $\mathbf{m}(\mathbf{x})$ is a worthwhile field deserving further studies.

Bibliography

[1] N. S. Altman. An introduction to kernel and nearest-neighbour nonparametric regression. *Amer. Statist.*, **46**, 1992, 175–185.

[2] M. Aerts and I. Gijbels. A three-stage procedure based on bootstrap critical points. *Sequential Anal.*, **12**, 1993, 93–112.

[3] P. J. Bickel and M. Rosenblatt. On some global measures of the deviations of density function estimates. *Ann. Statist.*, **1**, 1973, 1071–1095.

[4] P. L. Bickel and D. A. Freedman. Some asymptotic theory for the bootstrap. *Ann. Statist.*, **9**, 1981, 1196–1217.

[5] J. Boularan, L. Ferré and P. Vieu. Location of particular points in nonparametric regression analysis. *Aust. N. Z. J. Stat.*, **37**, 1995, 161–168.

[6] G. E. P. Box. Some theorems on quadratic forms applied in the study of analysis of variance problems, II. Effects of inequality of variance and of correlation between errors in the two-way classification. *Ann. Math. Statist.*, **25**, 1954, 484–498.

[7] L. Breiman and W. S. Meisel. General estimates of the intrinsic variability of data in nonlinear regression models. *J. Amer. Statist. Assoc.*, **71**, 1976, 301–307.

[8] M. J. Buckley and G. K. Eagleson. A graphical method for estimating the residual variance in nonparametric regression. *Biometrika*, **76**, 1989, 203–210.

[9] J. Y. Campbell. The econometrics of financial market. Princeton university press, New Jersey, 1997.

[10] R. J. Carroll. On sequential density estimation. *Z. Wahrsch. Verw. Geb.*, **36**, 1976, 137–151.

[11] C. K. Carter and G. K. Eagleson. A comparison of variance estimators in nonparametric regression. *J. Roy. Statist. Soc., Ser. B*, **54**, 1992, 773–780.

[12] S. K. Chatterjee. On an extension of Stein's two-sample procedure for the multinormial mean. *Cal. Stat. Assoc. Bull.*, **9**, 1959, 20–28.

[13] P. Chaudhuri, K. Doksum and A. Samarov. On Average Derivative Quantile Regression. *Ann. Statist.*, **25**, 1997, 715–744.

[14] Y. S. Chow and H. Robbins. On the asymptotic theory of fixed-width sequential confidence intervals of the mean. *Ann. math. Statist.*, **36**, 1965, 457–462.

[15] Y. S. Clark. A calibration curve for radiocarbon dates. *Antiquity.*, **49**, 1975, 251–266.

[16] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.*, **74**, 1979, 829–836.

[17] G. Collomb. Estimation Nonparamétrique de la Régression par la Méthode du Noyau. Thesis at University Paul Sabatier, Toulouse, France, 1976.

[18] D. R. Cox. Estimation by double sampling. *Biometrika*, **39**, 1952, 217–227.

[19] B. M. de Silva and N. Mukhopadhyay. Kernel density estimation of wool fibre Diameter. *Univ. of RMIT. Tech. Rep.*, **14**, 2002.

[20] B. M. de Silva, A. Roy and N. Mukhopadhyay. Two-stage nonparametric density estimation with bootstrapping. *An international Conference on Recent Development in Statistics and Probability and their Applications*, Dec 30-Jan 2, New Delhi, India, 2001.

[21] B. M. de Silva and N. Mukhopadhyay. Sequential procedures for nonparametric kernel density estimation. *Bulletin of the international Statistical Institute*, 53rd Session, Contributed Papers, Book 2, 2001, 8–9.

[22] L. S., Dharmasena, P. Zeephongsekul and B. M. de Silva. Two-Stage Sequential Procedure for Nonparametric Regression Estimation. *ANZIAM J.*, **49**, 2008, C699–C716.

[23] L. S., Dharmasena, B. M. de Silva and P. Zeephongsekul. Sequential Fixed-width Confidence Bands for Kernel Regression Estimation. *Proceedings of The 2008 IAENG International Conference on Data Mining and Applications, IMECS 2008*, 19-21 March 2008, Kowloon, Hong Kong, 432–436.

[24] L. S., Dharmasena, B. M. de Silva and P. Zeephongsekul. Application of Sequential Nonparametric Confidence Bands in Finance. *Proceedings of 36th Australian Conference of Economists ACE 2007*, 24-26 September 2007, Tasmania, Australia, CDROM.

[25] L. S., Dharmasena, B. M. de Silva and P. Zeephongsekul. Fixed-width Sequential Confidence Bands in Kernel Regression Using Bootstrap

Approximation. *Proceedings of International Conference on Data Management ICDM 2008*, 25-26 February 2008, New Delhi, India, 488–494.

[26] J. Diebolt. A nonparametric test for the regression function: asymptotic theory. *J. Statist. Plann. Inference*, **44**, 1995, 1-17.

[27] G. Dikta. Approximation of nearest neighbour regression function estimators. Technical report, Univ. Giessen, 1988.

[28] D. L. Donoho and I. M. Johnstone. Idea spatial adaption by walvet shrinkage. *Biometrika*, **81**, 1994, 425–455.

[29] N. R. Draper and H. Smith. Applied Regression Analysis. 3rd ed., John Wiley & Sons, Inc., 1998.

[30] S. Efromovich and M. S. Pinkser. Estimation of square inegral probability density of a random variable. *Probl. Inf. Transm.*, **18**, 1982, 175–189.

[31] S. Efromovich. Nonparametric Curve Estimation: Methods, Theory and Applications. Springer-Verlag, New York, 1999.

[32] S. Efromovich. Financial application sequential nonparametric curve estimation. Applied Sequential Methodologies. Marcel Dekker, New York, 2004.

[33] B. Efron. Nonparametric standrd errors and confidence intervals. *Can. J. Statist.*, **9**, 1981, 139-172.

[34] B. Efron. The jackknife, the bootstrap and other resampling plans. Philadelphia: SIAM, 1982.

[35] B. Efron. Better bootstrap confidence intervals. *J. Amer. Statist. Ass.*, **82**, 1987, 171–200.

[36] B. Efron. Better methods: another look at the jackknife. *Ann. Statist.*, **7**, 1979, 1–26.

[37] U. Einmahl and D. M. Mason. An emperical process approach to the uniform consistency of kernel-type function estimators. *J. Theoret. Probab.*, **13**, 2000, 1-37.

[38] R. L. Eubank. Spline smoothing and nonparametric regression. New York, Marcel and Dekker, 1988.

[39] R. L. Eubank and P. L. Speckman. Confidence bands in nonparametric regression. *J. Amer. Statist. Assoc.*, **88**, 1993, 1287–1301.

[40] J. Fan. Design-adaptive nonparametric regression. *J. Amer. Statist. Assoc.*, **87**, 1992, 998–1004.

[41] J. Fan. Local linear regression smoothers and their minimax efficiency. *Ann. Statist.*, **21**, 1993, 196–216.

[42] J. Fan and I. Gijbels. Local Polynomial Modelling and its Applications. Chapman and Hall, London, 1996.

[43] J. Faraway. Bootstrap selection of bandwidth and confidence bands for nonparametric regression. *J. Statist. Comp. Simul.*, **37**, 1990, 37–44.

[44] J. Faraway and M. Jhun. Bootstrap choice of bandwidth for density estimation. *J. Amer. Statist. Assoc.*, **85**, 1990, 1119–1122.

[45] J. Fox. Nonparametric simple regression: Smoothing scatterplots. Sage, Thousand Oaks CA, 2000a.

[46] T. Gasser and H. G. Müller. Kernel estimation of regression functions. In smoothing Techniques for Curve Estimation. Lecture notes in Mathematics, **757**, Springer-Verlag, New York, 1979, 23–68.

[47] T. Gasser, L. Sroka and C. Jennen-Steinmetz. Residual variance and residual pattern in nonlinear regression. *Biometrika*, **73**, 1986, 625–633.

[48] M. Ghosh, N. Mukhopadhyay and P.K. Sen. Sequential Estimation. Wiley, New York, 1997.

[49] M. Ghosh and N. Mukhopadhyay. Sequential estimation of the percentiles of exponential and normal distributions. *S. Afr. Statist. J.*, **23**, 1989, 251–268.

[50] M. Ghosh and P.K. Sen. On two-stage James-Stein estimators. *Sequential Anal.*, **2**, 1983, 359–367.

[51] M. Ghosh and N. Mukhopadhyay. Consistency and asymptotic efficiency of two-stage and sequential estimation procedures. *Sankhya Ser. A*, **43**, 1981, 220–227.

[52] M. Ghosh and N. Mukhopadhyay. On two fundamental problems of sequential estimation. *Sankhya Ser. B*, **38**, 1976, 203-218.

[53] M. Ghosh and N. Mukhopadhyay. Sequential point estimation of the mean when the distribution is unspecified. *Commun. Ann. Statist.*, **8**, 1979, 637–652.

[54] M. Ghosh and N. Mukhopadhyay. On two fundamental problems of sequential estimation. *Sankhya Ser. B*, **38**, 1976, 203–218.

[55] A. L. Goel and K. Okumoto. A time dependent error detection rate model for software reliability and other performance measures, *IEEE Trans. Reliab.*, **28**, 1979, 206–211.

[56] P. J. Green and B. W. Silverman. Nonparametric regression and generalized linear models: a roughness penalty approach. London, Chapman and Hall, 1994.

[57] P. Hall. Large sample optimality of least square cross-validation in density estimation. *Ann. Statist.*, **11**, 1983, 1156–1174.

[58] P. Hall and J. S. Marron. On variance estimation in nonparametric regression. *Biometrika*, **77**, 1990, 415–419.

[59] P. Hall and D. M. Titterington. On confidence bands in nonparametric density estimation and regression. *J. Multivariate Anal.*, **27**, 1988, 228–254.

[60] P. Hall. On bootstrap confidence intervals in nonparametric regression. *Ann. Statist.*, **20**, 1992, 695–711.

[61] P. Hall. On symmetric bootstrap confidence intervals. *J. Roy. Statist. Soc., Ser. B*, **50**, 1988, 35–45.

[62] P. Hall. The Bootstrap and Edgeworth Expansion. Springer-Verlag, New York, Inc., 1992.

[63] P. Hall. Asymptotic theory of triple sampling for sequential estimation of a mean. *Ann. Statist.*, **9**, 1981, 1229–1238.

[64] P. Hall. Empirical functionals and efficient smoothing parameter selection. *J. Roy. Statist. Soc., Ser. B*, **54**, 1992, 475–530.

[65] P. Hall, S. J. Sheather, M. C. Jones and J. S. Marron. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika*, **78**, 1991, 263–269.

[66] P. Hall, J. W. Kay and D. M. Titterington. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, **77**, 1990, 521–528.

[67] P. Hall and R. J. Carroll. Variance function estimation in regression: The effect of estimating the mean. *J. Roy. Statist. Soc., Ser. B*, **51**, 1989, 3–14.

[68] W. Härdle and J. S. Marron. Optimal bandwidth selection in nonparametric regression function estimation. *Ann. Statist.*, **13**, 1985, 1465–1481.

[69] W. Härdle and A. W. Bowman. Bootstrapping in nonparametric regression: Local adaptive smoothing and confidence bands. *J. Amer. Statist. Assoc.*, **83**, 1988, 102–110.

[70] W. Härdle and E. Mammen. Comparison nonparametric regression: Local adaptive smoothing and confidence bands. *J. Amer. Statist. Assoc.*, **83**, 1989, 102–110.

[71] W. Härdle and J. S. Marron. Bootstrap Simultaneous Error Bars for Nonparametric Regression. *Ann. Statist.*, **19**, 1991, 778–796.

[72] W. Härdle and E. Mammen. Comparing Nonparametric Vs. parametric regression fits, Discussion paper, A-177, Univ. Bonn., 1989.

[73] W. Härdle. Smoothing techniques: with implementation in S. Springer-Verlag, New York, 1991.

[74] E. Isogai. A note on sequential density estimation. *Sequential Anal.*, **7**, 1988, 11–21.

[75] E. Isogai. The convergence rate of fixed-width sequential confidence intervals for a probability density function. *Sequential Anal.*, **6**, 1987a, 55–69.

[76] E. Isogai. On the asymptotic normality for nonparametric sequential density estimation. *Bull. Inform. Cybernet.*, **22**, 1987b, 215–224.

[77] E. Isogai. Joint asymptotic normality of nonparametric recursive density estimators at a finite number of distinct points. *J. Japan Statist. Soc.*, **14**, 1984, 125–135.

[78] E. Isogai. Stopping rules for sequential density estimation. *Bull. Math. Statist.*, **19**, 1980/81, 53–67.

[79] Z. Jelinski and P. B. Moranda. Software reliability research. *Statistical Computer Performance Evaluation*, Academic Press, New York, 1972, 465–497.

[80] S. Kundu. Using stopping rules to bound the L_p distance in nonparametric estimation. *Univ. of Conn. Statist. Tech. Rep.*, 1994, 94–14.

[81] S. Kundu, and A. T. Martinsek. Bounding the L_1 distance in nonparametric estimation. *Univ. of Conn. Statist. Tech. Rep.*, 1994, 94–13.

[82] J. Lintner. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Rev. Economics and Statist.*, **47**, 1965, 13–47.

[83] D. G. Luenberger. A preference foundation for log mean-variance criteria in portfolio choice problems. *Journal of Econom. Dynam. Control.*, **17**, 1993, 887–906.

[84] D. G. Luenberger. Products of trees for investment analysis. *Journal of Econom. Dynam. Control.*, **22**, 1998, 1403-1417.

[85] H. Markowitz. Portfolio selection: Efficient diversification of investment. New York: Wiley, 1959.

[86] J. D. McDonald. Projection pursuit regression with the Orion I workstation, a 20 min 16 mm colour sound film, available for loan from Jerome H. Friedman, Compuatation research group, Bin 88, p.o. Box 4349, Stanford, Claif. 94305.

[87] A. T. Martinsek. Fixed width confidence bands for density functions. *Sequential Anal.*, **12**, 1993, 169–177.

[88] N. Mukhopadhyay. An overview of sequential nonparametric density estimation. *Nonlinear Anal.*, **30**, 1997, 4395–4402.

[89] N. Mukhopadhyay and W. T. Duggan. On a two-stage procedure having second-order properties with applications. *Ann. Inst. Statist. Math.*, **51**, 1999, 621–636.

[90] N. Mukhopadhyay and W. T. Duggan. Can a two-stage procedure enjoy second-order properties?. *Sankhya Ser. A*, **59**, 1997, 435–448.

[91] N. Mukhopadhyay and T. K. S. Solanky. Multistage Selection and Ranking Procedures. Marcel Deker, New York, 1994.

[92] N. Mukhopadhyay. Sequential estimation problems for negative exponential population. *Commun. Statist.* **17**, 1988, 2471–2506.

[93] N. Mukhopadhyay and J. Diaz. Two-stage sampling for estimating the mean of a negative binomial distribution. *Sequential Anal.*, **4**, 1985, 1–18.

[94] N. Mukhopadhyay. A consistent and asymptotically efficient two-stage procedure to construct fixed-width confidence intervals for mean. *Metrika*, **27**, 1980, 281–284.

[95] H. G. Müller. Weighted local regression and Kernel methods for nonparametric curve fitting. *J. Amer. Statist. Assoc.*, **82**, 1987, 231–238.

[96] H. G. Müller. Nonparametric analysis of longitudinal data. Lecture notes in statistics, **46**, Berlin, Springer, 1988.

[97] H. G. Müller and U. Stadtmuller. Estimation of heteroscedasticity in regression analysis. *Ann. Statist.*, **15**, 1987, 610–625.

[98] U. U. Müller, A. Schick and W. Wefelmeyer. Estimating the error variance in nonparametric regression by a covariate-matched u-statistic. *Statistics*, **37**, 2003, 79–188.

[99] U. U. Müller, A. Schick and W. Wefelmeyer. Estimating linear functionals of the error distribution in nonparametric regression. *J. Statist. Plann. Inference.*, **119**, 2004, 75-93.

[100] E. A. Nadaraya. On estimating regression. *Theory Probab. Appl.*, **9**, 1964, 141–142.

[101] D. Nychka. Splines as local smoothers. *Ann. Statist.*, **23**, 1995, 1175–1197.

[102] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, **33**, 1962, 1065-1976.

[103] H. Pham, L. Nordmann and X. Zhang. A general imperfect-debugging with S-Shaped fault- detection rate. *IEEE Trans. Reliab.*, **48**, 1999, 169–175.

[104] H. Pham. Software Reliability. Springer- Verlag, Singapore, 2000.

[105] M. B.Priestley and M. T. Chao. Nonparametric curve fitting. *J. Roy. Statist. Soc. Ser.B*, **34**, 1972, 385–392.

[106] W. Quian and V. Mammitzsch. Strong uniform convergence for the estimator of the regression function under φ mixing conditions. *Metrika*, **52**, 2000, 45–61.

[107] Ramkaran. The robustness of Stein's two-stage procedure. *Ann. Statist.*, **38**, 1982, 395–402.

[108] J. Rice. Boundary modifications for kernel regression. *Commun. Statist., Part A–Theroy and Methods*, **13**, 1984, 893–900.

[109] J. Rice. Bandwidth choice for nonparametric regression. *Ann. Statist.*, **12**, 1984, 1215–1230.

[110] H. Robbins. Sequential estimation of the mean of a normal population. In probability and Statistics, 235-245 (The Harald Cramer Volume), Almqvist and Wiksell, Stockholm, 1959.

[111] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, **27**, 1956, 832–837.

[112] M. Rosenblatt. Conditional probability density and regression estimators. In Multivariate Analysis II 25-31. Academic, New York, 1969.

[113] E. Rosenblueth. Point Estimates for Probability Moments. *Proc. Nat. Acad. Sci.*, **72**, 1975, 3812–3814.

[114] M. Rudemo. Emperical choice of histograms and kernel density estimators. *Scand. J. Statist.*, **9**, 1982, 65–78.

[115] D. Ruppert, S. J. Sheather and M. P. Wand. An effective bandwidth selector for local least squares regression *J. Amer. Statist. Assoc.*, **90**, 1995, 1257–1270.

[116] P. Sarda and P. Vieu. Kernel regression. In: Schimek, M.G. (Ed.), Smoothing and Regression: Approaches Computation and Application. Wiley Series in Probability and Statistics, 2000, 43–70.

[117] E. F. Schuster. Joint asymptotic distribution of the estimated regression function at a finite number of distinct points. *Ann. Math. Statist.*, **43**, 1972, 84–88.

[118] D. W. Scott and G. R. Terrell. Biased and unbiased cross-validation in density estimation. *J. Amer. Statist. Assoc.*, **82**, 1987, 1131–1146.

[119] W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *J. Finance*, **19**, 1964, 425–442.

[120] B. W. Silverman. Density Estimation for Statistics and Data Analysis. Chapman and Hall, London, 1986.

[121] B. W. Silverman and G. A. Young. The bootstrap: To smooth or not to smooth?, *Biometrika*, **74**, 1987, 469–479.

[122] B. W. Silverman. Some Aspects of the spline smoothing approach to non-Parametric regression curve fitting, *J. Roy. Statist. Soc. Ser.B*, **47**, 1985, 1–52.

[123] J. Shao and D. Tu. The Jackknife and Bootstrap. Springer-Verlag, New York, 1995.

[124] C. Stein. Some problems in sequentail estimation. *Econometrics*, **17**, 1949, 498–499.

[125] C. Stein. A two-sample test for linear hypothesis whose power is independent of the variance. *Ann. Math. Statist.*, **16**, 1945, 243–258.

[126] C. J. Stone. Consistent nonparametric regression. *Ann. Statist.*, **5**, 1977, 595–645.

[127] C. J. Stone. Cross-validatory choice and assessment of statistical prediction. *J. Roy. Statist. Soc. Ser.B*, **36**, 1974, 111–147.

[128] W. Stute. Modified cross-validation in density estimation. *J. Staist. Plann. Inference*, **30**, 1992, 293–305.

[129] W. Stute. Sequental fixed-width confidence intervals for a nonparametric density function. *Z. Wahrsch. Verw. Geb.*, **62**, 1983, 113–123.

[130] J. W. V. Swanepoel, J. W. J van Wijik and J. H. Venter. Fixed-width confidence intervals based on bootstrap procedures. *Commun. Statist.,-Seq. Analys.*, **2**, 1984, 289–310.

[131] K. Takezawa. Introduction to nonparametric regression. John Wiley & Sons, Inc., 1990.

[132] B. Vidakovic. Statistical modelling by wavelets. Wiley, New York, 1999.

[133] G. Vik and N. Mukhopadhyay. Triple sampling to construct fixed-width confidence intervals for estimatable parameters based on U-statistics. *Metron, Intern. J. Statist.*, **XLVI**, 1988, 165–174.

[134] G. Wahba. A survey of some smoothing problems and the method of generalized cross-validation for solvinf them, Applications of statistics, ed. P. R. Krisnaiah, Amsterdam: North Holland, 1977, 507–523.

[135] G. Wahba. Spline models for observational data. Philadelphia, SIAM, 1990.

[136] G. Wahba. Bayesian confidence intervals for the cross-validated smoothing spline. *J. Roy. Statist. Soc. Ser.B*, **45**, 1983, 133–150.

[137] G. Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *J. Roy. Statist. Soc. Ser.B*, **40**, 1978, 364–372.

[138] A. Wald. Sequential Analysis. Wiley, New York, 1947.

[139] M. P. Wand and M. C. Jones. Kernel Smoothing. Chapman and Hall, London, 1995.

[140] G. S. Watson. Smooth regression analysis. *Sankhyā Ser. A*, **26**, 1964, 359–372.

[141] E. J. Wegman and H. I. Davies. Sequential nonparametric density estimation. *IEEE Trans. Information Theory*, **21**, 1975, 619–628.

[142] A. Wood. Predicting software reliability. *IEEE Computer*, **11**, 1996, 69–77.

[143] S. Yamada, M. Ohba and M. Osaki. S-shaped reliability growth modelling for software error detection. *IEEE Trans. Reliab.*, **32**, 1983, 475–478.

[144] S. Yamada, M. Ohba and M. Osaki. S-shaped reliability growth models and their applications. *IEEE Trans. Reliab.*, **33**, 1984, 289–291.

[145] H. Yamato. Sequential estimate of a continuous probability density function and mode. *Bull. Math. Statist.*, **14**, 1971, 1–12.

[146] Y. Xia. Bia-corrected confidence bands in nonparametric regression. *J. Roy. Statist. Soc. Ser.B*, **4**, 1998, 797–811.

[147] Y. Xu and A. T. Martinsek. Sequential confidence bands for densities. *Ann. Statist.*, **23**, 1995, 2318–2240.

[148] P. Zeephongsekul, G. Xia and S. Kumar. Software-reliability growth model: primary-failures generate secondary-faults under imperfect debugging. *IEEE Trans. Reliab.*, **43**, 1994, 408–413.

[149] P. Zeephongsekul and L. S. Dharmasena, Predicting Software Reliability Growth Using Nonparametric Regression. *Proceedings of 14TH ISSAT International Conference on Reliability and Quality in Design 2008*, 7-9 August 2008, Florida, USA, 337–341.

Appendices

Appendix A

Fortran programme to estimate nonparametric kernel regression function using different kernel functions for fixed equidistant data.

```
program DiffKSSFDEq
use rnset_int
use rnnoa_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
integer, parameter :: mdata=10,Nsim=15000,iseed=123479,x0data=1
real, parameter :: sigma=0.5
real, dimension(x0data)::x00
integer, dimension(mdata)::m0
real, dimension(5000):: x,y
real,dimension(nsim)::yksim1,ynwsim1,yksim2,ynwsim2,yksim3,ynwsim3
real,dimension(nsim)::yksim4,ynwsim4,yksim5,ynwsim5
real :: yk1,NNsim,r0,yk_bar1,ynw_bar1,SE_ykbar1,SE_ynwbar1,mx0,x0
real :: ynw1,Z0(1),m00,yk_bar4,ynw_bar4,SE_ykbar4,SE_ynwbar4,yk2
real :: SE_ynwbar5,yk_bar5,ynw_bar5,yk3,yk4,yk5,ynw4,ynw5,SE_ykbar5
real :: yk_bar2,ynw_bar2,SE_ykbar2,SE_ynwbar2,yk_bar3,ynw_bar3,ynw3
real :: yk3,yk4,yk5,ynw4,ynw5,SE_ykbar5,SE_ykbar3,SE_ynwbar3,ynw2
integer:: im,m9,isim,ix,ix0
open (2, file='FDEqM3LL1.dat', status = 'unknown')
open (3, file='FDEqM3NW1.dat', status = 'unknown')
open (4, file='FDEqM3Se1.dat', status = 'unknown')
```

```
m0=(/10,20,30,50,100,200,300,400,500,1000/)
x00=(/.375/)
call rnset(iseed)
write (4,5) nsim,sigma
write (4,6)
write (3,5) nsim,sigma
write (3,6)
write (2,5) nsim,sigma
write (2,6)
do ix0=1, x0data
x0=x00(ix0)
mx0=4.0*x0+3.0
mx0=2.*exp(-(x0*x0)/(.3*.3*2.))+3.*exp(-((x0-1.)**2/(.7*.7*2.))
mx0=sin(.75*x0)*sin(.75*x0)+3.
write (4,8) x0,mx0
write (3,8) x0,mx0
write (2,8) x0,mx0
write (3,40)
write (2,35)
write (4,45)
do im = 1, mdata
m9=m0(im)
m00=float(m9)
call rcal(m9,x0,r0)
do isim = 1, Nsim
do ix = 1, m9
call rnnoa(z0)
x(ix) = float(ix)/float(m9)
y(ix)=4.0*x(ix)+3.0+ sigma*z0(1)
```

```
y(ix)=2.*exp(-(x(ix)*x(ix))/(.3*.3*2.))+3.*exp(-((x(ix)-1.)**2)
                                      /(.7*.7*2.) + sigma*z0(1)
y(ix)=sin(0.75*x(ix))*sin(0.75*x(ix))+3.0+ sigma*z0(1)
end do
call Ker_Est1(x,y,m9,r0,x0,yk1,ynw1)
yksim1(isim)= yk1
ynwsim1(isim)= ynw1
call Ker_Est2(x,y,m9,r0,x0,yk2,ynw2)
yksim2(isim)= yk2
ynwsim2(isim)= ynw2
call Ker_Est3(x,y,m9,r0,x0,yk3,ynw3)
yksim3(isim)= yk3
ynwsim3(isim)= ynw3
call Ker_Est4(x,y,m9,r0,x0,yk4,ynw4)
yksim4(isim)= yk4
ynwsim4(isim)= ynw4
call Ker_Est5(x,y,m9,r0,x0,yk5,ynw5)
yksim5(isim)= yk5
ynwsim5(isim)= ynw5
end do
NNsim=real(nsim)
yk_bar1=sum(yksim1)/NNsim
ynw_bar1=sum(ynwsim1)/NNsim
SE_ykbar1=sqrt(sum((yksim1-yk_bar1)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar1=sqrt(sum((ynwsim1-ynw_bar1)**2)/(NNsim-1.0))/sqrt(NNsim)
yk_bar2=sum(yksim2)/NNsim
ynw_bar2=sum(ynwsim2)/NNsim
SE_ykbar2=sqrt(sum((yksim2-yk_bar2)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar2=sqrt(sum((ynwsim2-ynw_bar2)**2)/(NNsim-1.0))/sqrt(NNsim)
```

```
yk_bar3=sum(yksim3)/NNsim
ynw_bar3=sum(ynwsim3)/NNsim
SE_ykbar3=sqrt(sum((yksim3-yk_bar3)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar3=sqrt(sum((ynwsim3-ynw_bar3)**2)/(NNsim-1.0))/sqrt(NNsim)
yk_bar4=sum(yksim4)/NNsim
ynw_bar4=sum(ynwsim4)/NNsim
SE_ykbar4=sqrt(sum((yksim4-yk_bar4)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar4=sqrt(sum((ynwsim4-ynw_bar4)**2)/(NNsim-1.0))/sqrt(NNsim)
yk_bar5=sum(yksim5)/NNsim
ynw_bar5=sum(ynwsim5)/NNsim
SE_ykbar5=sqrt(sum((yksim5-yk_bar5)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar5=sqrt(sum((ynwsim5-ynw_bar5)**2)/(NNsim-1.0))/sqrt(NNsim)
end do
end do

contains
!------------------------------------------------------------------
 subroutine rcal(ndata,x0,r0)
 integer, intent(in) :: ndata
 real, intent(in) :: x0
 real, intent(out)::r0
 integer::D
 real::r00,A,B,C,r
 A= min(x0,(1-x0))
 C=float(ndata)
 B=((-100.0)*log(A))/log(C)
 D= ceiling(B)
 r00=0.01*float(min(D,99))
 r=0.0
 r0=max(r,r00)
end subroutine rcal
!-----------------------------------------------------------------
subroutine Ker_Est1(X,Y,m,r,x0,yk1,ynw1)
implicit none
integer, intent(in) :: m
real, intent(in) :: r, x0
real, dimension(m), intent(in) :: X,Y
real, intent(out) :: yk1,ynw1
integer :: j
real :: s0, s1, s2, t0, t1,an, xx0, hn, xhn, khn
an = real(m)
hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,m
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk01(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk1 = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw1=t0/s0
    end subroutine Ker_Est1
!________________________________________________________________
    subroutine Ker_Est2(X,Y,m,r,x0,yk2,ynw2)
    implicit none
    integer, intent(in) :: m
    real, intent(in) :: r, x0
    real, dimension(m), intent(in) :: X,Y
    real, intent(out) :: yk2,ynw2
    integer :: j
    real :: s0, s1, s2, t0, t1,an, xx0, hn, xhn, khn
    an = real(m)
    hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,m
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk02(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk2 = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw2=t0/s0
    end subroutine Ker_Est2
!-------------------------------------------------------------
    subroutine Ker_Est4(X,Y,m,r,x0,yk4,ynw4)
    implicit none
    integer, intent(in) :: m
    real, intent(in) :: r, x0
    real, dimension(m), intent(in) :: X,Y
    real, intent(out) :: yk4,ynw4
    integer :: j
    real :: s0, s1, s2, t0, t1,an, xx0, hn, xhn, khn
    an = real(m)
    hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,m
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk04(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk4 = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw4=t0/s0
    end subroutine Ker_Est4
!---------------------------------------------------------------
    subroutine Ker_Est5(X,Y,m,r,x0,yk5,ynw5)
    implicit none
    integer, intent(in) :: m
    real, intent(in) :: r, x0
    real, dimension(m), intent(in) :: X,Y
    real, intent(out) :: yk5,ynw5
    integer :: j
    real :: s0, s1, s2, t0, t1,an, xx0, hn, xhn, khn
    an = real(m)
    hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,m
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk05(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk5 = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw5=t0/s0
    end subroutine Ker_Est5
! Std normal Kernel
        real function nk01(xk)
        real, intent(in) :: xk
        real:: pi, d00, up
        pi = 2.0*asin(1.0)
        d00 = sqrt(2.0*pi)
        up=exp(-0.5*xk*xk)
        nk01=up/d00
        end function nk01
! Epanechnikov Kernel
       real function nk02(xk)
       real, intent(in) :: xk
       real:: d00
        if (abs(xk) .lt. sqrt(5.0))then
        d00=3.0/(4.0*sqrt(5.0))
        nk02=d00*(1.0-0.2*xk*xk)
        else
        nk02=0.0
        endif
        end function nk02
! Double Expo Kernel
       real function nk04(xk)
       real, intent(in) :: xk
       nk04=0.5*exp(-abs(xk))
        end function nk04
! Uniform Kernel
       real function nk05(xk)
```

```
      real, intent(in) :: xk
       if (abs(xk) .lt. 1.0)then
       nk05=0.5
       else
       nk05=0.0
       endif
       end function nk05
end program DiffKSSFDEq
```

Appendix B

Fortran programme to estimate nonparametric residual variance for different residual distributions for fixed equidistant design data.

```
program RSS
use rnset_int
use rnun_int
use rnexp_int
use rnund_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
integer, parameter :: mdata=6, rdata=3, iseed=123479,nsim=15000
                      ,vdata=3
real, dimension(500):: x,y,e
real, parameter ::  a=3.0, b=4.0
real, dimension(vdata)::v0
integer, dimension(mdata)::m0
real, dimension(NSIM):: VUsim,VPHsim
integer :: ir,isim,nmiss,ix,m9,im,ie(500),NR,iv
real :: sigma,pi,k0,z0(1),z1(1)
real :: r0(rdata),m00,an,hn,theta ,max_U, min_PH,max_PH
real:: NNsim,V_U,V_PH,VU_bar,VPH_bar,SE_Ubar,SE_PHbar,min_U
r0=(/.201,0.51,.76 /)
m0=(/10,30,50,100,200,500 /)
v0=(/0.05,0.25,0.90 /)
call rnset(iseed)
write(2,45)
write(2,10)
pi = 2.0*asin(1.0)
K0=1.0/SQRT(2.0*pi)
write (2,5) nsim
write (2,6)
do iv=1, vdata
sigma= sqrt(v0(iv))
!EXPONENTIAL
theta=sigma
!LAPLACE
theta=sigma/sqrt(2.0)
write (2,8) v0(iv)
write (2,30)
do im = 1, mdata
m9=m0(im)
m00=float(m9)
write (2,2) m9
    do ir=1, rdata
       do isim=1,nsim
          call rnund (2,ie)
            do ix = 1, m9
                call rnun(z1)
                x(ix)=z1(1)
               ! call rnnoa(z0)
                call rnexp(z0)
                IF (ie(ix).eq.1) then
                    e(ix)=-1.0*z0(1)
                else
                    e(ix)=z0(1)
```

```
                 endif
!!!!!!!!! Fixed Equidistant  !!!!!!!!!!!!!!!!!!!!!
!$$$$$$$$$ e~N(0,0.5^2) $$$$$$$$$$$$$$$$$
y(ix) = a + b*x(ix) + sigma*z0(1)
y(ix)= 2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)-1.0)
                        *(x(ix)-1.0))/(0.7*0.7*2.0))+ sigma*z0(1)
y(ix) = sin(0.75*x(ix))*sin(0.75*x(ix)) + 3.0 + sigma*z0(1)
!$$$$$$$$$ e~Expo(lamda=sigma) $$$$$$$$$$$$$$$$$
y(ix) = sin(0.75*x(ix))*sin(0.75*x(ix)) + 3.0 + theta*z0(1)
y(ix) = a + b*x(ix) + theta*z0(1)
y(ix)= 2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)-1.0)
                        *(x(ix)-1.0))/(0.7*0.7*2.0))+ theta*z0(1)
! $$$$$$$$$ e~Laplace(lamda=sigma/sqrt(2)) $$$$$$$$$$$$$$$$$
y(ix) = sin(0.75*x(ix))*sin(0.75*x(ix)) + 3.0 + theta*e(ix)
y(ix) = a + b*x(ix) + theta*e(ix)
y(ix)= 2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)-1.0)
                        *(x(ix)-1.0))/(0.7*0.7*2.0))+ theta*e(ix)
end do
call RSU(x,y,m9,r0(ir),V_U)
VUsim(isim)=V_U
call RSS_PH(x,y,m9,pi,r0(ir),V_PH )
VPHsim(isim)=V_PH
end do
NNsim=real(nsim)
VU_bar=sum(VUsim)/NNsim
VPH_bar=sum(VPHsim)/NNsim
SE_Ubar=sqrt(sum((VUsim-VU_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_PHbar=sqrt(sum((VPHsim-VPH_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
             min_U = VUsim(1)
```

```
            max_U = VUsim(1)
            min_PH = VPHsim(1)
            max_PH = VPHsim(1)
            do isim = 1,Nsim
                if(min_U .gt. VUsim(isim)) min_U=VUsim(isim)
                if(max_U .lt. VUsim(isim)) max_U=VUsim(isim)
                if(min_PH .gt. VPHsim(isim)) min_PH=VPHsim(isim)
                if(max_PH .lt. VPHsim(isim)) max_PH=VPHsim(isim)
            end do
end do
end do
end do

contains
-------------------------------------------------------------------
  Subroutine RSU(x,y,n,r,V_U)
    implicit none
    integer, intent(in)::n
    real, intent(in)::r
    real, dimension(n):: x,y
    real, intent(out):: V_U
    integer:: i,j,k
    real :: an0,hn,xx0,xhn,khn,s0,s2,g(n),w

    an0 =real(2*n*(n - 1))
    hn= real(n)**(-r)
    s2 = 0.0
    do i = 1, n
    s0 = 0.0
```

```
       do k = 1, n
         if (i .ne. k) then
          xx0 = x(k) - x(i)
          xhn = xx0/hn
          khn = nk0(xhn)/hn
          s0 = s0 + khn
         endif
       enddo
       g(i)=s0/real(n-1)
     enddo
    do i = 1, n
       do j = 1, n
           if (i .ne. j) then
              xx0 = x(j) - x(i)
              xhn = xx0/hn
              khn = nk0(xhn)/hn
              w = 0.5*(1.0/g(i)+1.0/g(j))*khn
              s2= s2+(y(i)-y(j))*(y(i)-y(j))*w
           endif
        end do
    end do
    V_U = s2/an0
   end Subroutine RSU
-------------------------------------------------------------------
Subroutine RSS_PH(x,y,ndata,pi,r,V_PH)
    integer, intent(in)::ndata
    real, dimension(ndata):: x,y
    real, intent(in)::r,pi
    real, intent(out):: V_PH
```

```
integer:: i,j,k
real :: s0, s1, s2, t0, t1,t2,k0
real :: an, xx0, hn, xhn, khn,x0,y0
real::xij,xijh,kij,wij
K0=1.0/SQRT(2.0*pi)
an = real(ndata)
hn = an**(-r)
    s2 = 0.0
    t1 = 0.0
    t2 = 0.0
do i=1,ndata
    s0 = 0.0
    t0 = 0.0
    s1 = 0.0
    x0=x(i)
    y0=y(i)
    do k=1,ndata
        xx0 =  x0-x(k)
        xhn = xx0/hn
        khn = nk0(xhn)
        s0 = s0 + khn
    end do
    do j=1,ndata
        xij = x0 - x(j)
        xijh=xij/hn
        kij = nk0(xijh)
        wij= kij/s0
        t0 = t0 + wij*y(j)
        s1 = s1 + wij*wij
        end do
        s2= s2+ k0/s0
        t1 = t1 + s1
        t2 = t2+ (y0-t0)*(y0-t0)
     end do
     V_PH = t2/(an-2.0*s2 + t1)
end Subroutine RSS_PH
    end program RSS
```

Appendix C.1

Fortran programme to compute final sample size N using two-stage procedure for fixed equidistant design data.

```
program TWOSTAGE
use rnset_int
use rnnof_int
use anorin_int
use rnnoa_int
use tin_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
! Variables
integer, parameter: mdata=1,ddata=5,Nsim=15000 iseed=123479,x0data=1
real, parameter :: sigma=0.5, a=3.0, b=4.0, alpha =0.10
real, dimension(ddata)::d0
real, dimension(x0data)::x00
integer, dimension(mdata)::m0
real, dimension(90000):: x,y,xx,yy
common /xy/ x,y,xx,yy
integer, dimension(nsim)::N_sim, Tsim
real,dimension(nsim)::NN_sim, TTsim,yksim,ynwsim,vsim
real,dimension(1)::z0
integer:: n_opt, m2,min_N,max_N,min_T,max_T
real ::  f_x,N_bar,T_bar,SE_Tbar,n4,SE_Nbar,cp,SE_cp,mx,yk,m1,SE_cpnw
real:: ZBS,ADF,u0,u1,u2,diff,oversam,T_hat,NNsim,r0,r2,yk_bar,ynw_bar
real:: v_bar,SE_vbar,SE_ykbar,SE_ynwbar,Var_gsj,t_h,BT2,BTD,Tc0,x0,ynw
real :: z,z2,pi,pi_root,mx0,Bn,BB,Z2B,S2,r1,d2,n_star,m00, T_star,cpnw
```

```
integer::  id,im,m9,isim,ix,n3,Tc,ic,ix0,icnw,dn
open (2, file='2Rel.out', status = 'unknown')
open (3, file='2Rel.dat', status = 'unknown')
open (4, file='sin.dat', status = 'unknown')
    d0=(/ 0.14,0.12,0.09,.07,0.05/)
    m0=(/15/)
    x00=(/.533/)
    call rnset(iseed)
    write(2,45)
    write(3,45)
    z= anorin(1.0-0.5*alpha)
    z2=z*z
    pi = 2.0*asin(1.0)
    pi_root = sqrt(pi)
    BB = 2.0*pi_root
    Bn = 1.0/BB
    Z2B=Z2*Bn
    S2 = sigma*sigma
do ix0=1, x0data
x0=x00(ix0)
mx0=2.*exp(-(x0*x0)/(.3*.3*2.))+3.*exp(-((x0-1.)*(x0-1.))
                                              /(.7*.7*2.))
mx0=sin(0.75*x0)*sin(0.75*x0)+3.0
do im = 1, mdata
       m9=m0(im)
       m00=float(m9)
       call rcal(m9,x0,r0)
       r1=1.0/(1.0-r0)
do id = 1, ddata
```

```
    write (3,15) d0(id)
     d2 = d0(id)*d0(id)
     n_star= (Z2B*S2/d2)**r1
     T_star= n_star/m00
    ic=0
    icnw=0
     call TCal(m9,alpha,t_h)
do isim = 1, Nsim
do ix = 1, m9
call rnnoa(z0)
x(ix) = float(ix)/float(m9)
y(ix)=2.*exp(-(x(ix)*x(ix))/(.3*.3*2.))+3.*exp(-((x(ix)-1.)**2)/
                                        .7*.7*2.))+ sigma*z0(1)
y(ix)=sin(0.75*x(ix))*sin(0.75*x(ix))+3.0+ sigma*z0(1)
y(ix)= 10*(1.0-exp(-0.2*x(ix)))+ sigma*z0(1)
xx(ix)=x(ix)
yy(ix)=y(ix)
end do
            call RSS(m9,Var_gsj)
            vsim(isim)=Var_gsj
            BT2 = t_h*t_h*Bn
            BTD = BT2/d2
            Tc0 = (BTD*Var_gsj)**r1
            Tc = ceiling(Tc0/m00)
            Tsim(isim) = max(1,Tc)
            n3 = Tsim(isim)*m9
            N_sim(isim)=n3
            dn=n3-m9
          call rcal(n3,x0,r2)
          call Ker_Est(m9,n3,dn,Tsim(isim),r2,x0,a,b,sigma,yk,ynw)
          yksim(isim)= yk
           ynwsim(isim)= ynw
            if (abs(mx0-yk).LT. d0(id)) ic=ic+1
            if (abs(mx0-ynw).LT. d0(id)) icnw=icnw+1
      end do
          NNsim = real(nsim)
         NN_sim = real(N_sim)
         N_bar = sum(NN_sim)/NNsim
         yk_bar = sum(yksim)/NNsim
         v_bar=sum(vsim)/NNsim
         ynw_bar = sum(ynwsim)/NNsim
SE_Nbar  = sqrt(sum((NN_sim-N_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_vbar   = sqrt(sum((vsim-v_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ykbar  = sqrt(sum((yksim-yk_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
SE_ynwbar  = sqrt(sum((ynwsim-ynw_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
            min_N = N_sim(1)
            max_N = N_sim(1)
            min_T = Tsim(1)
            max_T = Tsim(1)
     do isim = 1,Nsim
         if(min_N .gt. N_sim(isim)) min_N=N_sim(isim)
         if(max_N .lt. N_sim(isim)) max_N=N_sim(isim)
         if(min_T .gt. Tsim(isim)) min_T=Tsim(isim)
         if(max_T .lt. Tsim(isim)) max_T=Tsim(isim)
     end do
     diff = n_bar-n_star
     oversam = diff/n_star*100.0
     TTsim = real(Tsim)
     T_bar = sum(TTsim)/NNsim
     SE_Tbar = sqrt(sum((TTsim-T_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
     cp = float(ic)/float(Nsim)
     cpnw =float(icnw)/float(Nsim)
     SE_cp = sqrt((cp*(1.0-cp))/NNsim)
     SE_cpNW = sqrt((cpNW*(1.0-cpNW))/NNsim)
end do
end do
end do

contain
!______________________________________________________________________
subroutine Ker_Est(m,n3,dn,t,r,x0,A0,B0,sigma,yk,ynw)
implicit none
integer, intent(in) :: m,n3,dn,t
real, intent(in) :: r, x0,A0,B0,sigma
real, dimension(90000):: x,y,xx,yy
common /xy/ x,y,xx,yy
real, intent(out) :: yk,ynw
integer :: i,j, ir,j0,j1
real ::an,xx0,hn,xhn,khn,stdn(dn),s0,s1,s2,t0,t1,pi,zt(1)
    an = real(n3)
    hn = an**(-r)
    pi = 2.0*asin(1.0)
    If (n3 .gt. m) then
       call rnnoa(stdn)
       j=0
       j1=0
    do i=1,n3
    if (mod(i,t) .gt. 0) then
     x(i)=real(i)/an
     j=j+1
     if (j .gt. dn) then
        print *, 'Error1'
        stop
     endif
y(i)=2.0*exp(-(x(i)*x(i))/(0.3*0.3*2.0))+3.0*exp(-(x(i)-1.0)
                   *(x(i)-1.0)/(2.0*0.7*0.7))+ sigma*stdn(j)
y(i)=sin(0.75*x(i))*sin(0.75*x(i))+3.0+sigma*stdn(j)
y(i)= 10*(1.0-exp(-0.2*x(i)))+ sigma*stdn(j)
   else
    j1=j1+1
    if (j1 .gt. m) then
        print *, 'Error2'
        stop
    endif
    x(i)=xx(j1)
    y(i)=yy(j1)
 endif
 end do
 endif
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,n3
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk0(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw=t0/s0
    end subroutine Ker_Est
    end program TWOSTAGE
```

Appendix C.2

Fortran programme to compute final sample size N using two-stage procedure for random design data.

```
program RD_2Stage
use rnset_int
use rnnof_int
use anorin_int
use rnnoa_int
use TIN_INT
use rnun_INT
use ordst_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
! Variables
integer, parameter::mdata=1, ddata=1,Nsim=15000,iseed=12347,x0data=1
real, parameter ::sigma=0.5, a=3.0, b=4.0, alpha =0.05
real, dimension(ddata)::d0
real, dimension(x0data)::x00
integer, dimension(mdata)::m0
real, dimension(90000):: x,y
common /xy/ x,y
integer, dimension(nsim)::N_sim
real,dimension(nsim)::NN_sim,var,yllsim,ynwsim
integer:: min_N,max_N,nmiss,nx,m9,id,im,isim,ix,n3,ic, dnsim, icnw,ix0
real :: N_bar,SE_Nbar,cp,SE_cp,SE_cpNW,yk,diff,oversam,NNsim,r0,fx,p,df
real :: z,z2,pi,pi_root,mx0,Bn,BB,Z2B,S2,r1,d2,n_star,m00,t_h,BT2,BTD
real::fxhat,Var_PH,r2,ynw,cpnw,x0,yll_bar,ynw_bar,z0(1),z1(1),n4,V_bar
```

```
open (2, file='RD2S_NL6.out', status = 'unknown')
open (3, file='RD2S_NL6.dat', status = 'unknown')
d0=(/ 0.15,0.13,.11,.09,0.07,0.05/)
m0=(/25 /)
x00=(/.306/)
call rnset(iseed)
write(2,45)
write(3,45)
z= anorin(1.0-0.5*alpha)
z2=z*z
pi = 2.0*asin(1.0)
pi_root = sqrt(pi)
BB = 2.0*pi_root
Bn = 1.0/BB
fx=1.0
Z2B=Z2*Bn
p = (1.0 - 0.5*alpha)
S2 = sigma*sigma
write (2,5) nsim
write (3,5) nsim
write (2,6)
write (3,6)
write (2,7) alpha, sigma
write (3,7) alpha, sigma
do ix0=1, x0data
x0=x00(ix0)
mx0= sqrt(a + b*x0)
mx0=2.0*exp(-(x0*x0)/(0.3*0.3*2.0))+3.0*exp(-((x0-1.0)**2)
                                             /(0.7*0.7*2.0))
```

```
mx0=sin(2.0*pi*(x0-0.5))*sin(2.0*pi*(x0-0.5))
mx0=5.0*sqrt(x0*x0-0.02)
write (2,8) x0,mx0
write (3,8) x0,mx0
write (3,35)
write (2,30)
do im = 1, mdata
       m9=m0(im)
       m00=float(m9)
       call rcal(m9,x0,r0)
       r1=1.0/(1.0-r0)
       df = m00
       t_h = TIN(p,df)
do id = 1, ddata
       d2 = d0(id)*d0(id)
       n_star= (Z2B*S2/(d2*fx))**r1
    ic=0
    icnw=0
    nx=m9
        do ix=1,nx
        call rnun(z1)
        x(ix)=z1(1)
        end do
do isim = 1, Nsim
do ix = 1, m9
call rnnoa(z0)
y(ix) = sqrt(a + b*x(ix)) + sigma*z0(1)
y(ix)=2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)-1.0)**2)
                                         /(0.7*0.7*2.0))+ sigma*z0(1)
```

```
y(ix)=sin(2.0*pi*(x(ix)-0.5))*sin(2.0*pi*(x(ix)-0.5))+sigma*z0(1)
y(ix)=5.0*sqrt(x(ix)*x(ix)-0.02)+sigma*z0(1)
end do
call RSS_PH(m9,pi,r0,Var_PH)
var(isim)=Var_PH
!Estimate the sample size N using two-stage procedure
            BT2 =  t_h* t_h*Bn
            BTD = BT2/d2
            fxhat=1.0
            n4 = (BTD*Var_PH/fxhat)**r1
            n3=ceiling(n4)
            N_sim(isim)=max(m9,n3)
            n3=N_sim(isim)
            if (n3 .gt. nx) then
               do ix=nx+1,n3
                call rnun(z1)
                x(ix)=z1(1)
               end do
                 nx=n3
            end if
          call rcal(n3,x0,r2)
          call Ker_Est(m9,n3,r2,x0,a,b,sigma,yk,ynw)
          yllsim(isim)=yk
          ynwsim(isim)=ynw
            if (abs(mx0-yk).LT. d0(id)) ic=ic+1
            if (abs(mx0-ynw).LT. d0(id)) icnw=icnw+1
 end do
 NNsim = real(nsim)
 NN_sim = real(N_sim)
 V_bar=sum(var)/NNsim
 yll_bar=sum(yllsim)/NNsim
 ynw_bar=sum(ynwsim)/NNsim
 N_bar=sum(NN_sim)/NNsim
 SE_Nbar=sqrt(sum((NN_sim-N_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
            min_N = N_sim(1)
            max_N = N_sim(1)
            do isim = 1,Nsim
                if(min_N .gt. N_sim(isim)) min_N=N_sim(isim)
                if(max_N .lt. N_sim(isim)) max_N=N_sim(isim)
            end do
            diff = n_bar-n_star
            oversam = diff/n_star*100.0
         cp = float(ic)/float(Nsim)
         cpnw=float(icnw)/float(Nsim)
         SE_cp = sqrt((cp*(1.0-cp))/NNsim)
          SE_cpNW = sqrt((cpNW*(1.0-cpNW))/NNsim)
end do
end do
end do

contains
!-----------------------------------------------------------------
Subroutine RSS_PH(ndata,pi,r,Var_PH)
integer, intent(in)::ndata
real, dimension(90000):: x,y
common /xy/ x,y
real, intent(in)::r,pi
real, intent(out):: Var_PH
```

```
integer:: i,j,k
real :: s0,s1,s2,t0,t1,t2,k0,an,xx0,hn,xhn,khn,x0,y0,xij,xijh
real :: kij,wij
K0=1.0/SQRT(2.0*pi)
an = real(ndata)
hn = an**(-r)
        s2 = 0.0
        t1 = 0.0
        t2 = 0.0
do i=1,ndata
        s0 = 0.0
        t0 = 0.0
        s1 = 0.0
        x0=x(i)
        y0=y(i)
do k=1,ndata
xx0 =  x0-x(k)
xhn = xx0/hn
khn = nk0(xhn)
s0 = s0 + khn
end do
do j=1,ndata
xij = x0 - x(j)
 xijh=xij/hn
 kij = nk0(xijh)
 wij= kij/s0
 t0 = t0 + wij*y(j)
 s1 = s1 + wij*wij
end do
```

```
s2= s2+ k0/s0
t1 = t1 + s1
t2 = t2+ (y0-t0)*(y0-t0)
 end do
 Var_PH = t2/(an-2.0*s2 + t1)
end Subroutine RSS_PH
!-----------------------------------------------------------------
   Subroutine Kernelfx(ndata,x0,r,fxhat)
    real, dimension(90000):: x,y
    common /xy/ x,y
    integer, intent(in)::ndata
    real, intent(in)::x0,r
    real, intent(out)::fxhat
    real:: an,hn,nhn,s0,xx0,kx0
    integer::  i
    an = real(ndata)
    hn = an**(-r)
    nhn=an*hn
    s0=0.0
    do i=1,ndata
    xx0=(x0-x(i))/hn
    kx0=nk0(xx0)
    s0=s0+ kx0
    end do
    fxhat=s0/nhn
    End Subroutine Kernelfx
!------------------------------------------------------------------
subroutine Ker_Est(m,n3,r,x0,A0,B0,sigma,yk,ynw)
integer, intent(in) :: m,n3
```

```
real, intent(in) :: r, x0,A0,B0,sigma
real, dimension(90000):: x,y
common /xy/ x,y
real, intent(out) :: yk,ynw
integer :: i,j,k
real :: s0, s1, s2, t0, t1,zt(1)
real :: an, xx0, hn, xhn, khn
an = real(n3)
 hn = an**(-r)
j=m+1
do i=j,n3
call rnnoa(zt)
y(i)=sin(2.0*pi*(x(i)-0.5))*sin(2.0*pi*(x(i)-0.5))+sigma*zt(1)
y(i)=2.0*exp(-(x(i)*x(i))/(0.3*0.3*2.0))+3.0*exp(-((x(i)-1.0)**2)
                                      /(0.7*0.7*2.0))+ sigma*zt(1)
end do
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,n3
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk0(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
       yk = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
       ynw=t0/s0
 end subroutine Ker_Est

end program RD_2Stage
```

Appendix D.1

Fortran programme to compute final sample size N using bootstrap two-stage procedure for fixed equidistant design data.

```
program FD_boot
use rnund_int
use rnset_int
use rnnof_int
use anorin_int
use TIN_INT
use rnnoa_int
use rnun_INT
use ordst_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
integer, parameter::x0data=1,mdata=1,ddata=5,Nsim=15000,iseed=12349
                                                          ,bs=500,mn=25
real, parameter :: sigma=0.5, a=3.0, b=4.0, alpha =0.05
real, dimension(x0data)::x00
real, dimension(ddata)::d0
integer, dimension(mdata)::m0
real, dimension(90000):: x,y,xx,yy
common /xy/ x,y,xx,yy
real,dimension(bs)::zll,znw,OZ_ll,OZ_nw
integer, dimension(nsim)::N_sim,N_ll,N_nw,Tsim
real,dimension(nsim)::NN_sim,ynw_b,yksim,ynwsim,yk_b,yk_b1,ynw_b1
real,dimension(mn)::ek_d,enw_d,ek,enw,ykstar,ynwstar,ekstar,enwstar
real,dimension(mn)::ykbhat,ynwbhat,ynwhat,ykhat
```

```
 real ::  N_bar,mx,yk,ynw,p,st_ll,st_nw,nhn,nopt_ll,nopt_nw,z0(1),x0
 real:: ek_bar,enw_bar,sum_e,r0,r2,ll_h,nw_h ,ll_s,nw_s,SE_ybnw,z1(1)
 real:: SE_nbar,cp_k,SE_k,cp_nw,SE_n,y_k,SE_yk,y_nw,SE_ynw,Nll_bar,r3
 integer::ib,i,NR,ie(mn),m9,nll,nnw,icnw_b,AB,nmiss,ic_b1,icnw_b1,dn
 real:: ZBS,ADF,u0,u1,u2,diff,oversam,NNsim,r2_ll,r2_nw,b_yk,b_ynw
 real:: k_bar,nw_bar,k_b,BD,r0b,cp_bk,SE_bk,SE_bn,yb_nw,cp_bn,t_h
 real :: z,z2,pi,pi_root,mx0,Bn,BB,Z2B,S2,r1,d2,n_opt,m00,V_u
 real :: SE_bn1,SE_nk,b_k,b_nw,BT2,BTD,hn,cp_bk1,SE_cbk1
 integer:: id,im,isim,ix,n3,ic,icnw,ix0,ic_b,min_N,max_N
 real::yb_nw1,SE_ybnw1,k_b1,SE_bk1,cp_bn1,z5(1),Nnw_bar,SE_nnw
 open (2, file=’M1WoB1.out’, status = ’unknown’)
 open (5, file=’M1WB3.out’, status = ’unknown’)
 open (6, file=’MWB4.out’, status = ’unknown’)
 d0=(/ 0.14,0.12,.09,0.07,.05/)
 m0=(/25/)
 x00=(/.306/)
    call rnset(iseed)
    z= anorin(1.0-0.5*alpha)
    z2=z*z
    p = (1.0 - 0.5*alpha)
    pi = 2.0*asin(1.0)
    pi_root = sqrt(pi)
    BB = 2.0*pi_root
    Bn = 1.0/BB
    Z2B=Z2*Bn
    S2 = sigma*sigma
do ix0=1, x0data
x0=x00(ix0)
mx0= a + b*x0
```

```
mx0=2.0*exp(-(x0*x0)/(0.3*0.3*2.0))+3.0*exp(-((x0-1.0)*(x0-1.0))
                                                     /(0.7*0.7*2.0))
mx0=sin(0.75*x0)*sin(0.75*x0)+3.0
mx0=sin(2.0*pi*(x0-0.5))*sin(2.0*pi*(x0-0.5))
mx0=sqrt(4.0*x0+3.0)
write (2,6) nsim, bs,x0, mx0,alpha
write (5,6) nsim, bs,x0, mx0,alpha
do im = 1, mdata
 m9=m0(im)
 m00=real(m9)
 call TCal(m9,alpha,t_h)
 BT2 = t_h*t_h*Bn
 call rcal(m9,x0,r0)
 hn=m00**(-r0)
 nhn=m00**(1.0-r0)
 write (2,8) m9,r0
 write (5,8) m9,r0
 r1=1.0/(1.0-r0)
 write (2,40)
 write (5,50)
 write (6,60)
 do id = 1, ddata
 call rnset(iseed)
 d2 = d0(id)*d0(id)
 BD = Bn/d2
 BTD = BT2/d2
 n_opt= (Z2B*S2/d2)**r1
 ic_b=0
 icnw_b=0
 ic_b1=0
 icnw_b1=0
 ic=0
 icnw=0
do isim = 1, Nsim
do ix = 1, m9
x(ix)=real(ix)/m00
call rnnoa(z0)
y(ix) = a + b*x(ix) + sigma*z0(1)
y(ix)=2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)-1.0)**2)
                                           (0.7*0.7*2.0))+ sigma*z0(1)
y(ix)=sin(2.0*pi*(x(ix)+0.2))*sin(2.0*pi*(x(ix)+0.2))+3.0+ sigma*z0(1)
y(ix)=sin(0.75*x(ix))+3.0+ sigma*z0(1)
y(ix)=sin(0.75*x(ix))*sin(0.75*x(ix))+3.0+ sigma*z0(1)
y(ix)=sin(2.0*pi*(x(ix)-0.5))*sin(2.0*pi*(x(ix)-0.5))+sigma*z0(1)
y(ix)=sqrt(4.0*x(ix)+3.0)+sigma*z0(1)
yy(ix)=y(ix)
xx(ix)=x(ix)
end do
!####          Without Bootstapping              #######
         call RSS(m9,V_u)
         Tsim(isim)=max(1,ceiling(((BTD*V_u)**r1)/m00))
         N_sim(isim)=m9*Tsim(isim)
         n3=N_sim(isim)
!####            Bootstapping                    #######
        do ix = 1, m9
           call rcal(m9,x(ix),r3)
           call Ker_Yhat(m9,r3,x(ix),ykhat(ix),ynwhat(ix))
            ek_d(ix) =  y(ix)-ykhat(ix)
```

```
          enw_d(ix) =  y(ix)-ynwhat(ix)
      end do
      ek_bar = sum(ek_d)/m00
      enw_bar = sum(enw_d)/m00
  do ix = 1, m9
  ek(ix) = (ek_d(ix)- ek_bar )/(1.0-1.0/m00)
  enw(ix) = (enw_d(ix)- enw_bar)/(1.0-1.0/m00)
  end do
     do ib=1,bs
        NR=m9
        call rnund (m9,ie)
     do i=1, m9
         ekstar(i)=ek(ie(i))
         enwstar(i)=enw(ie(i))
     end do
     do ix = 1, m9
          ykstar(ix)  = ykhat(ix) +ekstar(ix)
          ynwstar(ix) = ynwhat(ix)+enwstar(ix)
     end do
     call Ker_Ystar(m9,ykstar,ynwstar,r0,x0,ll_s,nw_s)
     call Ker_Yhat(m9,r0,x0,ll_h,nw_h)
      zll(ib) = abs((ll_s-ll_h)* sqrt(nhn))
      znw(ib) = abs((nw_s-nw_h)* sqrt(nhn))
   end do
call ordst (zll,bs,OZ_ll,NMISS)
call ordst (znw,bs,OZ_nw,NMISS)
AB=int(float(bs)*(1.0-alpha))
st_ll=OZ_ll(AB)
st_nw=OZ_nw(AB)
```

```
   nll =max( 1,ceiling(((st_ll/d0(id))**(2.0/(1.0-r0)))/m00) )
   nnw = max(1,ceiling(((st_nw/d0(id))**(2.0/(1.0-r0)))/m00) )
          N_ll(isim)=nll*m9
          N_nw(isim)=nnw*m9
      call rcal(n3,x0,r2)
      dn=n3-m9
      call Ker_Est(m9,n3,dn,Tsim(isim),r2,x0,a,b,sigma,yk,ynw)
      yksim(isim)= yk
      ynwsim(isim)= ynw
            if (abs(mx0-yk).LT. d0(id)) ic=ic+1
            if (abs(mx0-ynw).LT. d0(id)) icnw=icnw+1
          call rcal(N_ll(isim),x0,r2_ll)
          call rcal(N_nw(isim),x0,r2_nw)
          call Ker_Yhat(N_ll(isim),r2_ll,x0,b_yk,b_nw)
          call Ker_Yhat(N_nw(isim),r2_nw,x0,b_k,b_ynw)
          yk_b(isim)= b_yk
          ynw_b(isim)= b_ynw
          yk_b1(isim)= b_k
          ynw_b1(isim)= b_nw
           if (abs(mx0-b_yk).LT. d0(id)) ic_b=ic_b+1
           if (abs(mx0-b_ynw).LT. d0(id)) icnw_b=icnw_b+1
           if (abs(mx0-b_k).LT. d0(id)) ic_b1=ic_b1+1
           if (abs(mx0-b_nw).LT. d0(id)) icnw_b1=icnw_b1+1
      end do
      NNsim = real(nsim)
 !!!!!!!!!!!!!!!!!!!!!!! Without Boot^^^^^^^^^^^^^^^^^^^^^^^^^^^
NN_sim = real(N_sim)
N_bar = sum(NN_sim)/NNsim
SE_nbar = sqrt(sum((NN_sim-N_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
```

```
cp_k = real(ic)/NNsim
SE_k=sqrt((cp_k*(1.0-cp_k))/NNsim)
cp_nw = real(icnw)/NNsim
SE_n=sqrt((cp_nw*(1.0-cp_nw))/NNsim)
y_k=sum(yksim)/NNsim
SE_yk= sqrt(sum((yksim-yk)**2)/(NNsim-1.0))/sqrt(NNsim)
y_nw=sum(ynwsim)/NNsim
SE_ynw= sqrt(sum((ynwsim-ynw)**2)/(NNsim-1.0))/sqrt(NNsim)
write(2,45) d0(id),n_opt,N_bar,SE_nbar,cp_k,SE_k,cp_nw,SE_n,
                                       y_k,SE_yk,y_nw,SE_ynw
!!  !!!!!!!!!!!!!!!!!!!!!!!! With Boot^^^^^^^^^^^^^^^^^^^^^^^^^
Nll_bar =sum(real(N_ll))/NNsim
SE_nk= sqrt(sum((N_ll-Nll_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
Nnw_bar =sum(real(N_nw))/NNsim
SE_nnw= sqrt(sum((N_nw-Nnw_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
k_b=sum(yk_b)/NNsim
SE_bk= sqrt(sum((yk_b-k_b)**2)/(NNsim-1.0))/sqrt(NNsim)
yb_nw=sum(ynw_b)/NNsim
SE_ybnw= sqrt(sum((ynw_b-yb_nw)**2)/(NNsim-1.0))/sqrt(NNsim)
k_b1=sum(yk_b1)/NNsim
SE_bk1= sqrt(sum((yk_b1-k_b1)**2)/(NNsim-1.0))/sqrt(NNsim)
yb_nw1=sum(ynw_b1)/NNsim
SE_ybnw1= sqrt(sum((ynw_b1-yb_nw1)**2)/(NNsim-1.0))/sqrt(NNsim)
cp_bk = real(ic_b)/NNsim
SE_bk=sqrt((cp_bk*(1.0-cp_bk))/NNsim)
cp_bn = real(icnw_b)/NNsim
SE_bn = sqrt((cp_bn*(1.0-cp_bn))/NNsim)
cp_bk1 = real(ic_b1)/NNsim
SE_cbk1=sqrt((cp_bk1*(1.0-cp_bk1))/NNsim)
```

```
cp_bn1 = real(icnw_b1)/NNsim
SE_bn1 = sqrt((cp_bn1*(1.0-cp_bn1))/NNsim)
end do
end do
end do

contains
!-----------------------------------------------------------------
subroutine Ker_Ystar(ndata,ykstar,ynwstar,r,x0,yk_s,ynw_s)
implicit none
integer, intent(in) :: ndata
real, intent(in) :: r,x0,ykstar(ndata),ynwstar(ndata)
real, dimension(90000):: x,y,xx,yy
common /xy/ x,y,xx,yy
integer :: i, j, k
real :: s0, s1, s2, t0, t1,t2,an, xx0, hn, xhn, khn,yk_s,ynw_s
an = real(ndata)
hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        t2 = 0.0
        do j=1,ndata
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk0(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*ykstar(j)
            t1 = t1 + xx0*khn*ykstar(j)
            t2 = t2 + khn*ynwstar(j)
        end do
        yk_s = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw_s=  t2/s0
    end subroutine Ker_Ystar
!-----------------------------------------------------------------
    subroutine Ker_Yhat(ndata,r,x0,yk_h,ynw_h)
    implicit none
    integer, intent(in) :: ndata
    real, intent(in) :: r,x0
    real, dimension(90000):: x,y,xx,yy
    common /xy/ x,y,xx,yy
    real, intent(out):: yk_h,ynw_h
    integer :: i, j, k
    real :: s0, s1, s2, t0, t1,an, xx0, hn, xhn, khn
    an = real(ndata)
    hn = an**(-r)
        s0 = 0.0
        s1 = 0.0
        s2 = 0.0
        t0 = 0.0
        t1 = 0.0
        do j=1,ndata
            xx0 = x(j) - x0
            xhn = xx0/hn
            khn = nk0(xhn)
            s0 = s0 + khn
            s1 = s1 + xx0*khn
            s2 = s2 + xx0*xx0*khn
            t0 = t0 + khn*y(j)
            t1 = t1 + xx0*khn*y(j)
        end do
        yk_h = (s2*t0 - s1*t1)/(s0*s2 - s1*s1)
        ynw_h=  t0/s0
    end subroutine Ker_Yhat
end program FD_boot
```

Appendix D.2

Fortran programme to compute final sample size N using bootstrap two-stage procedure for random design data.

```
program RD_boot
use rnund_int
use rnset_int
use rnnof_int
use anorin_int
use TIN_INT
use rnnoa_int
use rnun_INT
use ordst_int
implicit none
include 'link_f90_dll.h'
include 'link_f90_static.h'
integer,parameter::x0data=1,mdata=1,ddata=5,Nsim=15000,iseed=12349
                                                        ,bs=500,mn=25
real, parameter :: sigma=0.5, a=3.0, b=4.0,alpha =0.05
real, dimension(x0data)::x00
real, dimension(ddata)::d0
integer, dimension(mdata)::m0
real, dimension(90000):: x,y
common /xy/ x,y
real,dimension(bs)::zll,znw,OZ_ll,OZ_nw
integer, dimension(nsim)::N_sim,N_ll,N_nw
real,dimension(nsim)::NN_sim,ynw_b,yksim,ynwsim,yk_b,yk_b1,ynw_b1
real,dimension(mn)::ek_d,enw_d,ek,enw,ykstar,ynwstar,ekstar
real,dimension(mn)::ykbhat,ynwbhat,ynwhat,ykhat,enwstar
```

```
real ::  N_bar,mx,yk,ynw,p,st_ll,st_nw,nhn,nopt_ll,nopt_nw,z0(1)
real:: ek_bar,enw_bar,sum_e,r0,r2,ll_h,nw_h ,ll_s,nw_s,z1(1),z5(1)
real:: SE_nbar,cp_k,SE_k,cp_nw,SE_n,y_k,SE_yk,y_nw,SE_ynw,Nll_bar
real::SE_nnw,cp_bn,SE_bn,yb_nw,SE_ybnw,k_bar,nw_bar,k_b,Nnw_bar
integer::ib,i,NR,ie(mn),m9,nx,nll,nnw,hn,BD,r0b,cp_bk,SE_bk min_N
real:: ZBS,ADF,u0,u1,u2,diff,oversam,NNsim,r2_ll,r2_nw,b_yk,b_ynw
real :: z,z2,pi,pi_root,mx0,Bn,BB,Z2B,S2,r1,d2,n_opt,m00,V_u,t_h
integer:: id,im,isim,ix,n3,ic,icnw,ix0,ic_b,icnw_b,AB,nmiss
real::yb_nw1,SE_ybnw1,k_b1,SE_bk1,cp_bk1,SE_cbk1,cp_bn1,SE_bn1
real::SE_nk,ic_b1,icnw_b1,b_k,b_nw,r3,max_N,BT2,BTD,x0
open (2, file=’outWoB1.out’, status = ’unknown’)
open (5, file=’outWB3.out’, status = ’unknown’)
open (6, file=’outWB4.out’, status = ’unknown’)
    d0=(/ 0.15,0.13,.11,.09,0.07/)
    m0=(/25/)
    x00=(/.756/)
    call rnset(iseed)
    WRITE (2,4)
    write (2,5)
    WRITE (5,4)
    write (5,5)
    z= anorin(1.0-0.5*alpha)
    z2=z*z
    p = (1.0 - 0.5*alpha)
    pi = 2.0*asin(1.0)
    pi_root = sqrt(pi)
    BB = 2.0*pi_root
    Bn = 1.0/BB
    Z2B=Z2*Bn
    S2 = sigma*sigma
do ix0=1, x0data
x0=x00(ix0)
mx0= a + b*x0
mx0=2.0*exp(-(x0*x0)/(0.3*0.3*2.0))+3.0*exp(-((x0-1.0)*(x0-1.0))/
                                                  (0.7*0.7*2.0))
mx0=sin(0.75*x0)*sin(0.75*x0)+3.0
mx0=sin(2.0*pi*(x0-0.5))*sin(2.0*pi*(x0-0.5))
mx0=sqrt(4.0*x0+3.0)
do im = 1, mdata
   m9=m0(im)
   m00=real(m9)
   t_h = TIN(p,(m00-1.0))
   BT2 = t_h*t_h*Bn
   call rcal(m9,x0,r0)
   hn=m00**(-r0)
   nhn=m00**(1.0-r0)
   write (2,8) m9,r0
   write (5,8) m9,r0
   r1=1.0/(1.0-r0)
   write (2,40)
   write (5,50)
   write (6,60)
   do id = 1, ddata
     call rnset(iseed)
       d2 = d0(id)*d0(id)
       BD = Bn/d2
       BTD = BT2/d2
    n_opt= (Z2B*S2/d2)**r1
    ic_b=0
    icnw_b=0
    ic_b1=0
    icnw_b1=0
    ic=0
    icnw=0
    nx=m9
    do ix=1,nx
        call rnun(z1)
        x(ix)=z1(1)
     end do
do isim = 1, Nsim
do ix = 1, m9
call rnnoa(z0)
y(ix) = a + b*x(ix) + sigma*z0(1)
y(ix)=2.0*exp(-(x(ix)*x(ix))/(0.3*0.3*2.0))+3.0*exp(-((x(ix)
                       -1.0)**2)/(0.7*0.7*2.0))+ sigma*z0(1)
y(ix)=sin(2.0*pi*(x(ix)+0.2))*sin(2.0*pi*(x(ix)+0.2))+3.0+
                                                sigma*z0(1)
y(ix)=sin(0.75*x(ix))+3.0+ sigma*z0(1)
end do
!####     Without Bootstapping          #########
         call RSS(m9,r0,V_U)
           N_sim(isim)=max(m9,ceiling((BTD*V_u)**r1))
           n3=N_sim(isim)
           if (n3 .gt. nx) then
               do ix=nx+1,n3
                call rnun(z1)
                x(ix)=z1(1)
               end do
                 nx=n3
            end if
          call rcal(n3,x0,r2)
          call Ker_Est(m9,n3,r2,x0,a,b,sigma,yk,ynw)
           yksim(isim)= yk
           ynwsim(isim)= ynw
            if (abs(mx0-yk).LT. d0(id)) ic=ic+1
            if (abs(mx0-ynw).LT. d0(id)) icnw=icnw+1
!####          Bootstapping              #########
        do ix = 1, m9
            !call rcal(m9,x(ix),r3)
            r3=0.84
            call Ker_Yhat(m9,r3,x(ix),ykhat(ix),ynwhat(ix))
            ek_d(ix) =  y(ix)-ykhat(ix)
            enw_d(ix) =  y(ix)-ynwhat(ix)
        end do
        ek_bar = sum(ek_d)/m00
        enw_bar = sum(enw_d)/m00
       do ix = 1, m9
            ek(ix) = (ek_d(ix)- ek_bar )/(1.0-1.0/m00)
            enw(ix) = (enw_d(ix)- enw_bar)/(1.0-1.0/m00)
       end do
       do ib=1,bs
          NR=m9
          call rnund (m9,ie)
        do i=1, m9
           ekstar(i)=ek(ie(i))
           enwstar(i)=enw(ie(i))
        end do
do ix = 1, m9
ykstar(ix)  = ykhat(ix) +ekstar(ix)
ynwstar(ix) = ynwhat(ix)+enwstar(ix)
end do
   call Ker_Ystar(m9,ykstar,ynwstar,r0,x0,ll_s,nw_s)
   call Ker_Yhat(m9,r0,x0,ll_h,nw_h)
zll(ib) = abs((ll_s-ll_h)* sqrt(nhn))
znw(ib) = abs((nw_s-nw_h)* sqrt(nhn))
end do
call ordst (zll,bs,OZ_ll,NMISS)
call ordst (znw,bs,OZ_nw,NMISS)
AB=int(float(bs)*(1.0-alpha))
st_ll=OZ_ll(AB)
st_nw=OZ_nw(AB)
nll = ceiling((st_ll/d0(id))**(2.0/(1.0-r0)))
nnw = ceiling((st_nw/d0(id))**(2.0/(1.0-r0)))
N_ll(isim)=max(m9,nll)
N_nw(isim)=max(m9,nnw)
call rcal(N_ll(isim),x0,r2_ll)
call rcal(N_nw(isim),x0,r2_nw)
call Ker_Yhat(N_ll(isim),r2_ll,x0,b_yk,b_nw)
call Ker_Yhat(N_nw(isim),r2_nw,x0,b_k,b_ynw)
yk_b(isim)= b_yk
ynw_b(isim)= b_ynw
yk_b1(isim)= b_k
ynw_b1(isim)= b_nw
if (abs(mx0-b_yk).LT. d0(id)) ic_b=ic_b+1
if (abs(mx0-b_ynw).LT. d0(id)) icnw_b=icnw_b+1
if (abs(mx0-b_k).LT. d0(id)) ic_b1=ic_b1+1
if (abs(mx0-b_nw).LT. d0(id)) icnw_b1=icnw_b1+1
end do
NNsim = real(nsim)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Without Boot^^^^^^^^^^^^^^^^
NN_sim = real(N_sim)
N_bar = sum(NN_sim)/NNsim
SE_nbar=sqrt(sum((NN_sim-N_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
cp_k = real(ic)/NNsim
SE_k=sqrt((cp_k*(1.0-cp_k))/NNsim)
cp_nw = real(icnw)/NNsim
SE_n=sqrt((cp_nw*(1.0-cp_nw))/NNsim)
y_k=sum(yksim)/NNsim
SE_yk= sqrt(sum((yksim-yk)**2)/(NNsim-1.0))/sqrt(NNsim)
y_nw=sum(ynwsim)/NNsim
SE_ynw= sqrt(sum((ynwsim-ynw)**2)/(NNsim-1.0))/sqrt(NNsim)
write(2,45) d0(id),n_opt,N_bar,SE_nbar,cp_k,SE_k,cp_nw,SE_n
                                     ,y_k,SE_yk,y_nw,SE_ynw
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! With Boot^^^^^^^^^^^^^^^^^^^
Nll_bar =sum(real(N_ll))/NNsim
SE_nk= sqrt(sum((N_ll-Nll_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
Nnw_bar =sum(real(N_nw))/NNsim
SE_nnw= sqrt(sum((N_nw-Nnw_bar)**2)/(NNsim-1.0))/sqrt(NNsim)
k_b=sum(yk_b)/NNsim
SE_bk= sqrt(sum((yk_b-k_b)**2)/(NNsim-1.0))/sqrt(NNsim)
yb_nw=sum(ynw_b)/NNsim
SE_ybnw= sqrt(sum((ynw_b-yb_nw)**2)/(NNsim-1.0))/sqrt(NNsim)
k_b1=sum(yk_b1)/NNsim
SE_bk1= sqrt(sum((yk_b1-k_b1)**2)/(NNsim-1.0))/sqrt(NNsim)
```

```
yb_nw1=sum(ynw_b1)/NNsim

SE_ybnw1=sqrt(sum((ynw_b1-yb_nw1)**2)/(NNsim-1.0))/sqrt(NNsim)

cp_bk = real(ic_b)/NNsim

SE_bk=sqrt((cp_bk*(1.0-cp_bk))/NNsim)

cp_bn = real(icnw_b)/NNsim

SE_bn = sqrt((cp_bn*(1.0-cp_bn))/NNsim)

cp_bk1 = real(ic_b1)/NNsim

SE_cbk1=sqrt((cp_bk1*(1.0-cp_bk1))/NNsim)

cp_bn1 = real(icnw_b1)/NNsim

SE_bn1 = sqrt((cp_bn1*(1.0-cp_bn1))/NNsim)

end do

end do

end do


end program RD_boot
```