Bias reduction studies in nonparametric regression with applications: an empirical approach

M Krugell

21102007

Dissertation submitted in partial fulfilment of the requirements for the degree *Magister Scientiae* in Statistics at the Potchefstroom Campus of the North-West University

Supervisor: Prof CJ Swanepoel

September 2014

It all starts here ™

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
®

Abstract

The purpose of this study is to determine the effect of three improvement methods on nonparametric kernel regression estimators. The improvement methods are applied to the Nadaraya-Watson estimator with cross-validation bandwidth selection, the Nadaraya-Watson estimator with plug-in bandwidth selection, the local linear estimator with plug-in bandwidth selection and a bias corrected nonparametric estimator proposed by Yao (2012). The different resulting regression estimates are evaluated by minimising a global discrepancy measure, i.e. the mean integrated squared error (MISE).

In the machine learning context various improvement methods, in terms of the precision and accuracy of an estimator, exist. The first two improvement methods introduced in this study are bootstrapped based. Bagging is an acronym for **b**ootstrap **agg**regat**ing** and was introduced by Breiman (1996a) from a machine learning viewpoint and by Swanepoel (1988, 1990) in a functional context. Bagging is primarily a variance reduction tool, i.e. bagging is implemented to reduce the variance of an estimator and in this way improve the precision of the estimation process. Bagging is performed by drawing repetitive bootstrap samples from the original sample and generating multiple versions of an estimator. These replicates of the estimator are then used to obtain an aggregated estimator. Bragging stands for **b**ootstrap **r**obust **agg**regat**ing**. A robust estimator is obtained by using the sample median over the B bootstrap estimates instead of the sample mean as in bagging.

The third improvement method aims to reduce the bias component of the estimator and is referred to as boosting. Boosting is a general method for improving the accuracy of any given learning algorithm. The method starts off with a sensible estimator and improves iteratively, based on its performance on a training dataset.

Results and conclusions verifying existing literature are provided, as well as new results for the new methods.

Keywords:

Kernel regression estimators, cross-validation bandwidth, plug-in bandwidth, bagging, bragging, boosting.

Opsomming

Die doel van die studie is om die effek van drie versterkingsmetodes op nieparametriese kernregressieberamers te bepaal. Die versterkingsmetodes word toegepas op die Nadaraya-Watson beramer met kruisgeldigheid-strookwydte, die Nadaraya-Watson beramer met inprop-strookwydte, die lokale linêere beramer met inprop-strookwydte en op Yao (2012) se beramer met kruisgeldigheid-strookwydte. Die drie verskillende kernregressieberamers word dan met mekaar vergelyk met behulp van 'n afwykingsverskilmaatstaf.

In die masjienleringvakgebied bestaan daar verskeie metodes om die noukeurigheid en akkuraatheid van 'n beramer te verbeter. Die doel van die studie is om te bepaal wat die mate van verbetering is, indien enige, wanneer *boosting*, *bagging* en *bragging* toegepas word op nieparametriese kernregressieberamers.

Bagging verwys na die verkryging van saamgestelde beramings uit skoenlussteekproewe. *Bagging* is primêr 'n hulpmiddel om die variansie van 'n beramer kleiner te maak. *Bagging* word geïmplementeer deur herhaaldelike skoenlussteekproewe te trek uit die oorspronklike steekproef en dan vir elke skoenlussteekproef 'n weergawe van die beramer te genereer. Op die manier word daar dan 'n saamgestelde beramer bereken. *Bragging* is 'n meer robuuste metode as *bagging*. 'n Meer robuuste beramer word verkry deur die mediaan te neem oor die B skoenlussteekproewe in plaas van die gemiddeld soos by *bagging*.

Die derde versterkingsmetode, *boosting* is 'n metode wat gebruik kan word om die akkuraatheid van 'n beramer te verbeter. *Boosting* het ook ontstaan in die masjienleringvakgebied.

Resultate en gevolgtrekkings wat bestaande literatuur bevestig, word getoon, asook nuwe resultate vir die nuwe metodes.

Sleutelwoorde:

Kernregressieberamers, kruisgeldigheid-strookwydte, inprop-strookwydte, bagging, bragging, boosting.

Preface

Regression analysis refers to the statistical technique used to study the relation between two or more quantitative variables. A regression curve describes a general relationship between a vector of explanatory variables $\mathbf{X}$ and a possible response variable $\mathbf{Y}$. Suppose that observations are realised on a continuous random variable $\mathbf{Y}$ at n predetermined values of a continuous independent variable $\mathbf{X}$. Let $(X_i, Y_i)\,,\ i = 1, \ldots, n$, be the values of the independent variable $\mathbf{X}$ and the dependent variable $\mathbf{Y}$ which result from this sampling scheme and assume that the X_i and Y_i are related by the regression model

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

where

$$m(x) = E\left[Y|X = x\right]$$

are the values of some unknown function m at the points $X_1, X_2, \ldots, X_n$. The function $m(x)$ is usually referred to as the *regression function* or regression curve, while the phrase *regression analysis* refers to methods for statistical inference about the regression function. The ε_i represent zero mean uncorrelated random variables with a common variance σ^2. The aim is then to obtain an estimate of $\hat{m}$ using the sample of size n, that tends to $m(x)$ as $n \to \infty$ (Watson, 1964:360). Two approaches for obtaining an estimate for $m(x)$ are *parametric regression* and *nonparametric regression*.

The *parametric approach* assumes that the structure of the regression function is known and depends only on finitely many parameters, and the data are used to estimate the unknown values of these parameters (Györfi, Kohler, *et al.*, 2002:9). On the other hand *nonparametric regression* is a collection of techniques for estimating a regression curve without making strong assumptions about the shape of the true regression function Altman (1992:175). A nonparametric regression model generally only assumes that $m(x)$ belongs to some infinite dimensional collection of functions (Eubank, 1988:5). Various methods to obtain a nonparametric regression estimate of m exist. The simplest nonparametric regression estimators are based on a *local averaging procedure*. This procedure can be defined as

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ni}(x) Y_i.$$

The basic idea for these methods is that large weight is given to observations in a small neighbourhood around x and small or no weight is given to points far away from x (Härdle, 1990:16). In kernel regression smoothing the

shape and size of the weights $\{W_{ni}(x)\}_{i=1}^{n}$ are defined by a density function with a scale parameter that adjusts the form and the size of the weights near x. The shape of the kernel weights is determined by a kernel K, whereas the size of the weights is parameterised by h, which is called the smoothing parameter or bandwidth. Every smoothing method discussed in this study is, at least asymptotically, of this form. In particular, in this study three kernel regression smoothers are considered: the Nadaraya-Watson estimator, the local linear estimator and Yao's (2012) estimator.

The choice of the bandwidth, h, plays a critical role in the performance of a kernel regression smoother. A local average over too large a neighbourhood produces an extremely oversmoothed curve which results in a biased estimate $\hat{m}(x)$ (Härdle, 1990:18). If the smoothing parameter is defined to correspond to a very small neighbourhood, then only a small number of observations contribute non-negligibly to the estimate $\hat{m}(x)$ at x making it very rough and wiggly (Härdle, 1990:18). In this case the variance of $\hat{m}(x)$ is inflated. A trade-off between reducing the variance by increasing h and keeping the bias low by decreasing h, exists. Various data-driven approaches to select the bandwidth exist. In this study leave-one-out cross-validation bandwidth selection and plug-in bandwidth selection methods are used. The aim of the study is to determine the effect of the choice of the bandwidth selection method on the performance of the nonparametric regression estimators.

In the machine learning context various improvement methods in terms of the precision and accuracy of an estimator exist. The purpose of this study is to determine the effect of improvement methods such as boosting, bagging and bragging on nonparametric kernel regression smoothers.

Bagging is an acronym for **b**ootstrap **agg**regat**ing** and was introduced by Breiman (1996a) from a machine learning viewpoint and by Swanepoel (1988, 1990) in a functional context. Bagging is primarily a variance reduction tool, i.e. bagging is implemented to reduce the variance of an estimator and in this way improve the precision of the estimation process. Bagging is performed by drawing repetitive bootstrap samples from the original sample and generating multiple versions of an estimator. These replicates of the estimator are then used to obtain an aggregated estimator. Bragging stands for **b**ootstrap **r**obust **agg**regat**ing**. A robust estimator is obtained by using the sample median over the B bootstrap estimates instead of the sample mean as in bagging. Hall and Robinson (2009) applied the bagging method to leave-one-out cross-validation bandwidth selection in kernel regression smoothing to produce better regression estimates in terms of reducing global discrepancy measures.

Boosting is a general method for improving the accuracy of an estimator. Boosting also originated in the machine learning context but the methodology and theory developed to be applicable over a wide range of fields. The method starts off with a sensible estimator and improves iteratively, based on its performance on a training dataset. Marzio and Taylor (2008) showed how to improve the Nadaraya-Watson kernel regression smoother by using the boosting method. The aim in the present study is to determine if these improvement methods will improve nonparametric kernel regression estimators and to quantify this improvement by means of simulation studies. A more detailed outline of the aims of study is now presented.

Aims of the study

The main aims of the dissertation are now presented.

- Present an overview of basic literature on nonparametric regression methods.
- Define the fundamental ideas and present applications of the bootstrap methodology, which underlie the bagging and bragging improvement methods.
- Present an overview of the basic literature of important aspects of the bagging, bragging and boosting improvement methods, such as the development of the methods in the machine learning context and examples of the application of the methods in the regression context.
- Develop and present algorithms to determine the three enhanced regression estimators with their respective smoothing parameter selection methods. The three nonparametric regression estimators of concern are the Nadaraya-Watson estimator, the local linear estimator and a new bias corrected nonparametric estimator proposed by Yao (2012). The appropriate smoothing parameter selection methods involve the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method.
- Develop and present algorithms for various ways in which the boosting, bagging and bragging improvement methods can be applied to the three regression estimators. This involves algorithms for three ways of determining bandwidths by using the bagging method, as well as three ways of determining bandwidths by using bragging methods. The applications involving plug-in methods are new contributions.
- Empirically evaluate and compare the three enhanced regression estimation methods with respect to the global error criteria measured, with a watchful eye on the behaviour of the bias.
- Determine the effect of the bandwidth selection method on the performance of the regression estimator with respect to the global discrepancy measured. In particular, the Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the Nadaraya-Watson estimator with plug-in bandwidth selection is of interest.
- Empirically determine and compare the performance of the boosted, bagged and bragged regression estimators with the respect to the global discrepancy measured for a variety of simulation setup scenarios.

Chapter layout

A basic outline of this dissertation is now presented.

- Chapter 1 will supply the reader with information needed to understand the methods used to construct nonparametric regression estimators and the role played by two important properties of kernel regression estimators, i.e. the kernel function and the bandwidth. Information regarding pointwise and global discrepancy measures for assessing the accuracy and precision of a regression estimator is also given.
- In Chapter 2 fundamental ideas of the bootstrap methodology are defined and illustrated. This serve as an introduction for the discussion of the bootstrap-based improvement methods, bagging and bragging.
- Chapter 3 contains a summary of existing literature on the development of the improvement methods in the machine learning context and the application of the improvement methods, bagging, bragging and boosting, in the regression context.
- Chapter 4 lays out the simulation setup and presents definitions and algorithms for the old and new procedures needed to achieve the aims of the study.
- Chapter 5 concludes with results of the conducted Monte Carlo studies, discussions and recommendations.
- Several appendices will provide empirical information regarding the simulation studies.
- A complete bibliography will be listed.

Bedankings

Aan my Hemelse Vader vir sy liefde, krag en genade.

Aan die volgende persone, my opregte dank:

- My studieleier, Prof. C.J. Swanepoel, vir u bekwame en professionele leiding, aanmoediging en ook vir die besondere persoon wat u is.
- Prof. J.W.H. Swanepoel vir sy kundigheid en bydrae tot hierdie studie.
- My man, Henry, vir jou oneindige liefde, belangstelling, motivering en geduld.
- My ouers, vir julle deurlopende opofferings, ondersteuning en bemoediging deur al my studiejare.
- My broers, Waldo, Johann en Cobus vir julle belangstelling en bemoediging.

Contents

Abstract i

Opsomming ii

Preface iii

Bedankings vii

List of Figures xv

List of Tables xv

List of Abbreviations xvi

1 Nonparametric regression 1

1.1 Introduction . . . 1

1.2 Regression methods: a brief review . . . 2

1.3 The stochastic nature of the observations . . . 4

1.3.1 Fixed design . . . 4

1.3.2 Random design . . . 5

1.4 Construction of nonparametric regression estimators . . . 5

1.4.1 Kernel estimators . . . 6

1.4.2 Nearest neighbour regression smoothing . . . 9

1.4.3 Local polynomial fitting . . . 10

1.4.4 Spline smoothing . . . 10

1.4.5 Robust regression . . . 12

1.5 Discrepancy measures . . . 15

1.5.1 Pointwise discrepancy measures . . . 15

1.5.2 Asymptotic properties of the MSE . . . 16

1.5.3 Global discrepancy measures . . . 19

1.6 Bias-variance trade-off . . . 20

1.7 Methods for choosing the smoothing parameter . . . 20

1.7.1 Cross-validation . . . 21
1.7.2 Penalising functions . . . 22
1.7.3 Plug-in method . . . 23
1.8 Selecting a kernel . . . 28
1.9 Kernel estimators of derivatives . . . 29
1.10 Multivariate kernel regression . . . 30
1.10.1 Bias, variance and asymptotics . . . 32
1.10.2 Bandwidth selection and practical aspects . . . 32

2 The bootstrap **34**
2.1 Introduction . . . 34
2.2 The bootstrap procedure . . . 35
2.2.1 The plug-in principle . . . 35
2.2.2 Choices of $\hat{F}$ . . . 36
2.3 Bootstrap estimation of the standard error and bias . . . 39
2.3.1 Bootstrap estimation of the standard error . . . 39
2.3.2 Bootstrap estimation of the bias . . . 40
2.3.3 The double bootstrap . . . 41
2.4 Bootstrap applied to regression analysis . . . 43
2.4.1 Bootstrapping residuals . . . 44
2.4.2 Bootstrapping pairs . . . 45
2.4.3 The wild bootstrap . . . 46

3 Improvement methods **49**
3.1 Introduction . . . 49
3.2 Ensemble methods . . . 50
3.3 Bagging and related methods . . . 51
3.4 Variants of bagging . . . 53
3.4.1 Subagging and moon-bagging . . . 53
3.4.2 Bragging and trimmed bagging . . . 54
3.5 Bagging and cross-validation bandwidth selection . . . 55
3.5.1 Background . . . 56
3.5.2 Methodology: bagging applied to h_{CV} . . . 56
3.5.3 Rescaling of the bandwidth . . . 57
3.6 Bagging and plug-in bandwidth selection . . . 58
3.7 Boosting in regression . . . 58
3.7.1 Functional gradient descent (FGD) view of boosting . . . 58
3.7.2 L_2-boosting . . . 60

4 New contributions and simulation studies **62**

4.1 Introduction: Aims of the study . . . 62

4.2 Revision of basic procedures and introduction of new additions . . . 63

4.2.1 The Nadaraya-Watson estimator and bandwidth selection methods . . . 64

4.2.2 The local linear estimator and bandwidth selection methods . . . 68

4.2.3 The BRNP estimator and bandwidth selection methods . . . 70

4.3 Improvement methods . . . 73

4.3.1 Bagging . . . 73

4.3.2 Corresponding bagging algorithms for methods Bag1, Bag2 and Bag3 . . . 76

4.3.3 Bragging . . . 79

4.3.4 Corresponding bragging algorithms for Brag1, Brag2 and Brag3 . . . 81

4.3.5 Boosting . . . 85

4.3.6 Measuring the discrepancies . . . 85

4.4 Matters surrounding the main algorithm . . . 86

4.4.1 Simulation setup and remarks regarding the main algorithm . . . 86

4.4.2 The main algorithm . . . 94

5 Results and conclusions **96**

5.1 Introduction . . . 96

5.2 Formulation of specific research questions . . . 97

5.2.1 Nadaraya-Watson estimator with cross-validation bandwidth selection (SimA) . . . 98

5.2.2 Nadaraya-Watson estimator with plug-in bandwidth selection (SimB) . . . 98

5.2.3 Local linear estimator with plug-in bandwidth selection (SimC) . . . 99

5.2.4 Yao's estimator with cross-validation bandwidth selection (SimD) . . . 100

5.2.5 Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the Nadaraya-Watson estimator with plug-in bandwidth selection (SimA compared to SimB) 100

5.2.6 Nadaraya-Watson estimator with plug-in bandwidth selection compared to the local linear estimator with plug-in bandwidth selection (SimB compared to SimC) . . . 101

5.3 Guidelines for reading Appendices A-D . . . 101

5.4 Guidelines for reading Appendices E and F . . . 103

5.5 Guidelines for reading Appendix G . . . 104

5.6 Guidelines for reading Appendix H . . . 105

5.7 Observations and conclusions for the Nadaraya-Watson estimator with cross-validation bandwidth selection . . . 106

5.7.1 Conclusions drawn for the Nadaraya-Watson estimator with cross-validation bandwidth selection . . . 111

5.8 Observations and conclusions for the Nadaraya-Watson estimator with plug-in bandwidth selection 112

5.8.1 Conclusions drawn for the Nadaraya-Watson estimator with plug-in bandwidth selection . 117

5.9 Observations and conclusions for the local linear estimator with plug-in bandwidth selection . . . 118
5.9.1 Conclusions drawn for the local linear estimator with plug-in bandwidth selection 122
5.10 Observations and conclusions for the BRNP estimator with cross-validation bandwidth selection 122
5.10.1 Conclusion drawn for the BRNP estimator with cross-validation bandwidth selection . . . 126
5.11 Observations and conclusions for comparing SimA to SimB 126
5.11.1 Conclusion drawn for comparing SimA to SimB 133
5.12 Observations and conclusions for comparing SimB to SimC 133
5.12.1 Conclusions drawn for comparing SimB to SimC 138
5.13 Discussion of computation time 139
5.14 Discussion of the graphs 140
5.15 Final remarks and recommendations 141

Bibliography **144**

A Results of Simulation Study A **151**

B Results of Simulation Study B **168**

C Results of Simulation Study C **181**

D Results of Simulation Study D **194**

E Comparing SimA to SimB **198**

F Comparing SimB to SimC **203**

G Computation time **208**

H Graphs and figures **212**

List of Figures

1.1 Examples of popular kernel functions 8

2.1 Bootstrap diagram 36

4.1 The underlying regression function $m(x)$ 87
4.2 Kernel functions used in the present study 88
4.3 Simulation scenarios for Model 1 93
4.4 Simulation scenarios for Model 2 93
4.5 Simulation scenarios for Model 3 93

5.1 The cross-validation function before and after transforming the response variable 123

H.1 Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$. 213
H.2 Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$. 214
H.3 Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$. 215
H.4 Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$. 216
H.5 Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$. 217
H.6 Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$. 218
H.7 Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$. 219
H.8 Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$. 220

List of Tables

1.1 Examples of popular kernel functions 8
1.2 Asymptotic MSE properties: Fixed design model 18
1.3 Asymptotic MSE properties: Random design model 19
1.4 Kernel functions minimising $V(K)B(K)$ 29
1.5 Some kernels and their efficiencies 30

4.1 The underlying regression function $m(x)$ 86
4.2 Kernels used in the present study 88
4.3 Procedures used in calculating the *N-W estimator* with *cross-validation bandwidth selection* . . . 91
4.4 Procedures used in calculating the *N-W estimator* with *plug-in bandwidth selection* 91
4.5 Procedures used in calculating the *local linear estimator* with *plug-in bandwidth selection* 92
4.6 Procedures used in calculating the *BRNP estimator* with *cross-validation bandwidth selection* . . 92

5.1 Summary of the four simulation studies 97
5.2 Summary of the two comparison studies 97
5.3 Comparison of the performance of NW_{CV} and NW_{PLUG} for the thirty-six simulation scenarios . 133
5.4 Comparison of the performance of LL_{PLUG} and NW_{PLUG} for the thirty-six simulation scenarios 139

A.1 N-W estimator with cross-validation bandwidth, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 152
A.2 N-W estimator with cross-validation bandwidth, Model 1, $X \sim U(-2,2)$, Triangular kernel . . . 153
A.3 N-W estimator with cross-validation bandwidth, Model 1, $X \sim N(0,1), N(0,1)$ kernel 154
A.4 N-W estimator with cross-validation bandwidth, Model 1, $X \sim N(0,1)$, Triangular kernel 155
A.5 N-W estimator with cross-validation bandwidth, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 156
A.6 N-W estimator with cross-validation bandwidth, Model 2, $X \sim U(-2,2)$, Triangular kernel . . . 157
A.7 N-W estimator with cross-validation bandwidth, Model 2, $X \sim N(0,1),\ N(0,1)$ kernel 158
A.8 N-W estimator with cross-validation bandwidth, Model 2, $X \sim N(0,1)$, Triangular kernel 159
A.9 N-W estimator with cross-validation bandwidth, Model 3, $X \sim U(-2,2),\ N(0,1)$, kernel 160
A.10 N-W estimator with cross-validation bandwidth, Model 3, $X \sim U(-2,2)$, Triangular kernel . . . 161
A.11 N-W estimator with cross-validation bandwidth, Model 3, $X \sim N(0,1),\ N(0,1)$ kernel 162
A.12 N-W estimator with cross-validation bandwidth, Model 3, $X \sim N(0,1)$, Triangular kernel 163

A.13 Bagging and bragging procedures producing the smallest variance and MISE values for Model 1 . 164
A.14 Bagging and bragging procedures producing the smallest variance and MISE values for Model 2 . 165
A.15 Bagging and bragging procedures producing the smallest variance and MISE values for Model 3 . 166
A.16 Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 1 167
A.17 Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 2 167
A.18 Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 3 167

B.1 N-W estimator with plug-in bandwidth, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 169
B.2 N-W estimator with plug-in bandwidth, Model 1, $X \sim U(-2,2)$, Triangular kernel 170
B.3 N-W estimator with plug-in bandwidth, Model 1, $X \sim N(0,1), N(0,1)$ kernel 171
B.4 N-W estimator with plug-in bandwidth, Model 1, $X \sim N(0,1)$, Triangular kernel 172
B.5 N-W estimator with plug-in bandwidth, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 173
B.6 N-W estimator with plug-in bandwidth, Model 2, $X \sim U(-2,2)$, Triangular kernel 174
B.7 N-W estimator with plug-in bandwidth, Model 2, $X \sim N(0,1),\ N(0,1)$ kernel 175
B.8 N-W estimator with plug-in bandwidth, Model 2, $X \sim N(0,1)$, Triangular kernel 176
B.9 N-W estimator with plug-in bandwidth, Model 3, $X \sim U(-2,2),\ N(0,1)$ kernel 177
B.10 N-W estimator with plug-in bandwidth, Model 3, $X \sim U(-2,2)$, Triangular kernel 178
B.11 N-W estimator with plug-in bandwidth, Model 3, $X \sim N(0,1),\ N(0,1)$ kernel 179
B.12 N-W estimator with plug-in bandwidth, Model 3, $X \sim N(0,1)$, Triangular kernel 180

C.1 LL estimator with plug-in bandwidth, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 182
C.2 LL estimator with plug-in bandwidth, Model 1, $X \sim U(-2,2)$, Triangular kernel 183
C.3 LL estimator with plug-in bandwidth, Model 1, $X \sim N(0,1), N(0,1)$ kernel 184
C.4 LL estimator with plug-in bandwidth, Model 1, $X \sim N(0,1)$, Triangular kernel 185
C.5 LL estimator with plug-in bandwidth, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 186
C.6 LL estimator with plug-in bandwidth, Model 2, $X \sim U(-2,2)$, Triangular kernel 187
C.7 LL estimator with plug-in bandwidth, Model 2, $X \sim N(0,1),\ N(0,1)$ kernel 188
C.8 LL estimator with plug-in bandwidth, Model 2, $X \sim N(0,1)$, Triangular kernel 189
C.9 LL estimator with plug-in bandwidth, Model 3, $X \sim U(-2,2),\ N(0,1)$ kernel 190
C.10 LL estimator with plug-in bandwidth, Model 3, $X \sim U(-2,2)$, Triangular kernel 191
C.11 LL estimator with plug-in bandwidth, Model 3, $X \sim N(0,1),\ N(0,1)$ kernel 192
C.12 LL estimator with plug-in bandwidth, Model 3, $X \sim N(0,1)$, Triangular kernel 193

D.1 BRNP estimator with cross-validation bandwidth, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 195
D.2 BRNP estimator with cross-validation bandwidth, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 196
D.3 BRNP estimator with cross-validation bandwidth, Model 3, $X \sim U(-2,2), N(0,1)$ kernel 197

E.1 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 199
E.2 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2,2)$, Triangular kernel 199
E.3 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1), N(0,1)$ kernel 199
E.4 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1)$, Triangular kernel 200
E.5 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 200
E.6 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2)$, Triangular kernel 200
E.7 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1),\ N(0,1)$ kernel 201
E.8 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, Triangular kernel 201
E.9 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2),\ N(0,1)$ kernel 201
E.10 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2)$, Triangular kernel 202
E.11 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0,1),\ N(0,1)$ kernel 202
E.12 Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0,1)$, Triangular kernel 202

F.1 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2,2), N(0,1)$ kernel 204
F.2 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2,2)$, Triangular kernel 204
F.3 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1), N(0,1)$ kernel 204
F.4 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1)$, Triangular kernel 205
F.5 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2), N(0,1)$ kernel 205
F.6 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2)$, Triangular kernel 205
F.7 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1),\ N(0,1)$ kernel 206
F.8 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, Triangular kernel 206
F.9 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2),\ N(0,1)$ kernel 206
F.10 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2)$, Triangular kernel 207
F.11 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0,1),\ N(0,1)$ kernel 207
F.12 Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0,1)$, Triangular kernel 207

G.1 Computer time (in seconds) for standard estimation methods 209
G.2 Computer time (in seconds) for Bag1 estimation methods 209
G.3 Computer time (in seconds) for Bag2 estimation methods 210
G.4 Computer time (in seconds) for Bag3 estimation methods 210
G.5 Computer time (in seconds) for Boost estimation methods 211
G.6 Computer time (in seconds) for the BRNP estimator 211

List of Abbreviations

AMISE	Asymptotic mean integrated squared error
AMSE	Asymptotic mean squared error
ASE	Averaged squared error
Bagging	Bootstrap aggregating
Bragging	Bootstrap robust aggregating
BRNP	Bias reduction nonparametric estimator
BRNP_{CV}	Bias reduction nonparametric estimator with cross-validation bandwidth selection
CV	Cross-validation bandwidth selection
DGP	Data generating process
DPI	Direct plug-in bandwidth selector
EDF	Empirical distribution function
FGD	Functional gradient descent
GM	Gasser-Müller estimator
ISE	Integrated squared error
k-NN	k-nearest neighbour estimator
LL	Local linear estimator
LL_{plug}	Local linear estimator with plug-in bandwidth selection
MC	Monte Carlo
MISE	Mean integrated squared error
Moon-bagging	m-out-of-n bootstrap aggregating
MSE	Mean squared error
NW	Nadaraya-Watson estimator
NW_{CV}	Nadaraya-Watson estimator with cross-validation bandwidth selection
NW_{plug}	Nadaraya-Watson estimator with plug-in bandwidth selection
OLS	Ordinary least squares estimator
PC	Priestley-Chao estimator
Plug	Plug-in bandwidth selection
ROT	Rule-of-thumb bandwidth selector
Subagging	subsample aggregating
WB	Wild bootstrap

Chapter 1

Nonparametric regression

1.1 Introduction

The specific aims of this study are stated and laid out in full in Chapter 4, where the new contributions of this study to the literature are discussed. The three fold aim of the present study can briefly be summarised as follows:

- Firstly, three enhanced regression estimation methods based on kernel regression estimation methods, involving bandwidth selection procedures, are studied, evaluated and compared with respect to well-known global error criteria. The three nonparametric regression estimators of concern are the Nadaraya-Watson estimator, the local linear estimator and a new bias corrected nonparametric estimator proposed by Yao (2012). The appropriate smoothing parameter selection methods involve the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method.
- Secondly, three improvement methods are applied to the three nonparametric regression estimators, and evaluated with respect to the global error criteria measured. The effects caused by a bias reduction method, i.e., boosting, as well as bootstrap-based variance reduction methods, i.e., bagging and bragging, are studied, measured and interpreted. Specifically, three ways of determining bandwidths by using the bagging method, as well as three ways of determining bandwidths by using the bragging method, will be executed and the effect of these bandwidths as it will be applied in the various regression estimation methods, will be evaluated in terms of global error criteria.
- Thirdly, the effect of the smoothing parameter selection methods involving the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection on the performance of the nonparametric regression estimator are evaluated with respect to the global error criteria.

The behaviour of the bias component of the estimators is of interest throughout the study.

Chapters 1-3 concentrates on the fundamental elements necessary to fulfil the tasks set in the aims of this study. Chapter 1 provides a brief introduction to the main aspects of nonparametric regression estimators. The

bagging and bragging improvement methods are based on bootstrap principles. In Chapter 2 fundamental ideas of the bootstrap methodology are defined and illustrated. Existing literature on aspects of the boosting, bagging and bragging improvement methods that are relevant for this study are summarised in Chapter 3. Chapter 4 lays out the simulation setup and presents definitions and algorithms for the old and new procedures needed to achieve the aims of the study. In Chapter 5 results of the conducted Monte Carlo studies are discussed and conclusions are drawn.

The objective set for this chapter is to give the reader a brief overview from the literature of the main aspects of nonparametric regression analysis. The initial objective in Section 1.2 is to distinguish between the philosophies which underlie parametric regression analysis and nonparametric regression analysis. The data used in regression analysis can originate from either fixed or random design settings. These data generation schemes are discussed in Section 1.3. Three paradigms for nonparametric regression, i.e., local averaging, local modelling and penalised modelling are described in Section 1.4. Estimators utilising the idea of local averaging are the Nadaraya-Watson estimator, Priestley-Chao estimator, Gasser-Müller estimator and the k-nearest neighbour estimator. Two important components of kernel estimators, namely the choice of the kernel function and of matters surrounding the choice of bandwidth, are also discussed. The most popular example of local modelling estimates is the local polynomial kernel estimate. In the case of the penalised modelling paradigm, the example provided displays the method of spline smoothing. This section concludes with methods for dealing with outliers via robust regression. The performance of an estimator can be evaluated by utilising various discrepancy measures. In Section 1.5 some important discrepancy measures are discussed such as the mean squared error (MSE) and the mean integrated squared error (MISE), with specific reference to the asymptotic properties of kernel and nearest neighbour estimators, as it appear in the literature. The bias-variance trade-off problem is introduced in Section 1.6. Three methods for selecting the smoothing parameter are presented in Section 1.7, i.e., cross-validation, penalising functions and the plug-in method. Possible choices for the kernel and the role of the kernel in the performance of the estimator receive attention in Section 1.8. In Section 1.9 it is shown how kernel estimation techniques can be utilised to estimate the derivatives of a regression function. This chapter concludes with remarks regarding the extension of the above information to the multivariate case.

1.2 Regression methods: a brief review

Regression analysis refers to the statistical technique used to study the relation between two or more quantitative variables. A regression curve describes a general relationship between a vector of explanatory variables $\mathbf{X}$ and a possible response variable $\mathbf{Y}$. The aim of regression analysis is to investigate the relationship between dependent and independent variables, to assess the contribution of the independent variables and to identify the impact of the independent variables on the dependent variable (Fan & Gijbels, 1996:1). The aim of this section is to distinguish between the philosophies which underlie parametric and nonparametric regression analysis. Therefore this section focuses on a simple model that provides a convenient framework for the discussion of parametric and nonparametric regression methods. For simplicity the study concentrates on the one-dimensional case, i.e., one

independent variable and one dependent variable are available. Note that most methods discussed below have been extended to the multivariate situation and are discussed in Section 1.10. The formulation of the regression relationship is based on Eubank (1988:1-2), unless stated differently.

Suppose that observations are realised on a continuous random variable $\mathbf{Y}$ at n predetermined values of a continuous independent variable $\mathbf{X}$. Let (X_i, Y_i), $i = 1, \ldots, n$, be the values of the independent variable $\mathbf{X}$ and the dependent variable $\mathbf{Y}$ which result from this sampling scheme and assume that the X_i and Y_i are related by the regression model

$$Y_i = m(X_i) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1.1}$$

where

$$m(x) = E\left[Y|X = x\right] \tag{1.2}$$

are the values of some unknown function m at the points $X_1, X_2, \ldots, X_n$. The function $m(x)$ is usually referred to as the *regression function* or regression curve, while the phrase *regression analysis* refers to methods for statistical inference about the regression function. In (1.1) the ε_i represent zero mean uncorrelated random variables with a common variance σ^2. The aim is then to obtain an estimate of $\hat{m}$ using the sample of size n, that tends to $m(x)$ as $n \to \infty$ (Watson, 1964:360). Two approaches for obtaining an estimate for $m(x)$ are *parametric regression* and *nonparametric regression*. The assumptions that are possible to make about $m(x)$ will determine the suitable inferential methodology to use for model (1.1).

The parametric approach assumes that the structure of the regression function is known and depends only on finitely many parameters, and the data are used to estimate the unknown values of these parameters (Györfi, Kohler, *et al.*, 2002:9). Since parametric estimates usually depend only on a few parameters, they are suitable even for small sample sizes n if the parametric model is appropriately chosen (Györfi, Kohler, *et al.*, 2002:9). The estimate of m, obtained by using the parametric approach, is a curve that has been selected from the family of curves allowed under the model assumptions and conforms to the data in some fashion. The parametric approach, however, suffers from the drawback that regardless of the data, a parametric estimate of m cannot approximate the regression function more successful than the best function which possesses the assumed parametric structure (Györfi, Kohler, *et al.*, 2002:9). For example, a linear regression estimate will produce a large error for every sample size if the true underlying regression function is not linear and can therefore not be well approximated by a linear function (Györfi, Kohler, *et al.*, 2002:9).

There are, however, other methods of fitting regression curves to data, for example *nonparametric regression techniques*. According to Altman (1992:175) nonparametric regression is a collection of techniques for estimating a regression curve without making strong assumptions about the shape of the true regression function. A nonparametric regression model generally only assumes that $m(x)$ belongs to some infinite dimensional collection of functions (Eubank, 1988:5). Various methods to obtain a nonparametric regression estimate of m exist. The simplest nonparametric regression estimators are based on a *local averaging procedure*. This procedure can be

defined as

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ni}(x)Y_i. \tag{1.3}$$

The basic idea for these methods is that large weight is given to observations in a small neighbourhood around x and small or no weight is given to points far away from x (Härdle, 1990:16). Every smoothing method discussed in this chapter is, at least asymptotically, of the form (1.3). Often the regression estimator $\hat{m}(x)$ is called a *smoother* and the outcome of the smoothing procedure is simply called the *smooth* (Härdle, 1990:17).

To summarise, parametric and nonparametric regression techniques represent two different approaches to the problem of regression analysis. Parametric methods require very specific, quantitative information from the researcher about the form of $m(x)$ that places restrictions on what the data can reveal about the regression function (Eubank, 1988:5). According to Eubank (1988:5) parametric techniques are most appropriate when theory, past experience and/or other sources are available that provide detailed knowledge about the process under study. In contrast, *nonparametric regression* techniques rely on the researcher to supply only qualitative information about $m(x)$ and allow the data to speak for itself concerning the actual form of the regression curve (Eubank, 1988:5). These methods are best suited for inference in situations where there is little or no prior information available about the regression curve. It should be noted that even though parametric and nonparametric regression models represent noticeably different approaches to regression analysis, this does not mean that the use of one approach prohibits the use of the other. Nonparametric regression techniques can be used to assess the validity of a proposed parametric model (Eubank, 1988:6). Conversely, it may be that the form of a fitted regression curve obtained by nonparametric techniques will suggest an appropriate parametric model for use in future studies. Therefore, nonparametric regression procedures may represent the final stage of data analysis or merely an exploratory or confirmatory step in the modelling process (Eubank, 1988:6).

1.3 The stochastic nature of the observations

According to Härdle (1990:21) and Chu and Marron (1991:407) the data $\{(X_i, Y_i)\}_{i=1}^n$ can be generated from two possible design schemes, i.e., the fixed design setting or the random design setting. In the following section these two possible scenarios for the origin of the data are discussed.

1.3.1 Fixed design

The fixed design model is concerned with controlled, nonstochastic X-variables (Härdle, 1990:21). According to Chu and Marron (1991:407) the fixed design model can be defined as

$$Y_i = m(x_i) + \varepsilon_i, \tag{1.4}$$

for $i = 1, \ldots, n$, where $m(x)$ is the regression function, the x_i's are nonrandom design points with $a \leq x_1 \leq \cdots \leq x_n \leq b$. The x-values are usually chosen by the researcher and in many cases the points are taken to be equidistributed on an interval $[a, b]$. Without loss of generality one can assume $[a, b] = [0, 1]$ and $x_i = \frac{i}{n}$. The

ε_i's are independent random variables with mean 0 and variance σ^2. Experimental studies give rise to fixed design regression.

1.3.2 Random design

In the random design setting the data points are thought of as being realisations from a bivariate probability distribution, where the (X_i, Y_i)'s are independent, identically distributed random variables. According to Chu and Marron (1991:407) the random design model can be defined as

$$Y_i = m(X_i) + \varepsilon_i, \tag{1.5}$$

for $i = 1, \ldots, n$, where the ε_i's are defined by $\varepsilon_i = Y_i - m(X_i)$ and assumed to have mean 0 and variance σ^2. In this design setting the X-values are usually not chosen by the researcher. Observational studies and sample surveys often result in random design settings.

The regression curve stated in (1.2) is well defined if $E(|Y|) < \infty$. If the joint density $f(x, y)$ exists, then $m(x)$ can be calculated as

$$m(x) = \frac{\int y f(x, y)\, dy}{f(x)}, \tag{1.6}$$

where $f(x) = \int f(x, y)\, dy$ denotes the marginal density of X (Härdle, 1990:21).

Although the stochastic mechanism is different, the basic idea of smoothing is the same for both random and nonrandom X-variables (Härdle, 1990:21). One might think there is little practical difference between these models, because the regression function only depends on the conditional distribution, where the X-values are given (Chu & Marron, 1991:407). Even though this is correct, it will be seen that the nature of the X-values will greatly influence the performance of the estimators.

1.4 Construction of nonparametric regression estimators

This section gives an overview of various ways to define nonparametric regression estimators. Three paradigms are described surrounding nonparametric regression, i.e., local averaging, local modelling and penalised modelling. Estimators utilising the idea of local averaging are the Nadaraya-Watson kernel estimator, the Priestley-Chao estimator, the Gasser-Müller estimator and the k-nearest neighbour estimator. From a function approximation point of view, the Nadaraya-Watson and the Gasser-Müller estimators both use local constant approximations (Fan & Gijbels, 1996:17). A generalisation of this leads to the local modelling paradigm. It is suggested that instead of locally fitting a constant to the data, rather locally fit a more general function, which depends on several parameters (Györfi, Kohler, *et al.*, 2002:20). The most popular example of local modelling estimates is the local polynomial kernel estimate. Here one locally fits a polynomial to the data. Instead of restricting the set of functions over which one minimises, one can rather add a penalty term to the functional to be minimised. This leads to the penalised modelling paradigm. The example provided displays the method of spline smoothing.

1.4.1 Kernel estimators

Recall from (1.1) that the regression relationship can be formulated as

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

where $\varepsilon_i = Y_i - m(X_i)$ satisfies $E(\varepsilon_i|X_i) = 0$. Therefore, Y_i can be considered as the sum of the value of the regression function at X_i and some error term ε_i, where the expected value of the error is zero. According to Györfi *et al.* (2002:18) this motivates the construction of the estimates by local averaging, i.e., estimation of $m(x)$ by the average of those Y_i where X_i is close to x. Such an estimate can be written as

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ni}(x)Y_i, \tag{1.7}$$

where $\{W_{ni}(x)\}_{i=1}^{n}$ denotes a sequence of weights which may depend on the whole vector $\{X_i\}_{i=1}^{n}$ (Härdle, 1990:16). The shape of the weight function is described by a density function with a scale parameter that adjusts the size and the form of the weights near x (Härdle, 1990:24). This shape function is referred to as a kernel K. The weight sequence for kernel smoothers is defined by

$$W_{hi}(x) = \frac{K_h(x - X_i)}{\hat{f}_h(x)}, \tag{1.8}$$

where

$$K_h(u) = h^{-1}K(u/h)$$

is the kernel with scale factor h and $\hat{f}_h(x)$ denotes some estimate of the marginal density $f(x)$ of X.

Three popular kernel estimators are now discussed: the Nadaraya-Watson estimator, the Priestley-Chao estimator and the Gasser-Müller estimator. In Chapter 4 more estimators are introduced, which involve the aims of the present study. The Nadaraya-Watson estimator is a generally applicable method, while the Priestley-Chao estimator and the Gasser-Müller estimator have specific fields of application which is not suitable for the present study. Attention is also given to two important properties of kernel estimators, i.e., the kernel function and the bandwidth.

a) The Nadaraya-Watson estimator

Nadaraya (1964) and Watson (1964) proposed the following estimator

$$\hat{m}_{NW}(x) = \frac{n^{-1}\sum_{i=1}^{n} K_h(x - X_i)Y_i}{n^{-1}\sum_{i=1}^{n} K_h(x - X_i)}. \tag{1.9}$$

For the Nadaraya-Watson estimator

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)$$

in (1.8). The function $\hat{f}_h(\cdot)$ is referred to as the Rosenblatt-Parzen kernel density estimator and was introduced by Rosenblatt (1956) and Parzen (1962) to estimate the marginal distribution of the X-values, $f(x)$, which is the denominator in (1.6). The choice of the denominator and numerator ensures that the Nadaraya-Watson weights sum to one. The shape of the kernel weights is determined by K, whereas the size of the weights is

parameterised by h, which is referred to as the bandwidth (Härdle, 1990:25). The above definition is stated for the random design setting, but also holds for the fixed design setting where the X_i-values, $i = 1, 2, \ldots, n$, are fixed, controlled and nonstochastic.

b) The Priestley-Chao estimator

For the fixed design setting with equidistant x-values chosen within the interval $[0, 1]$, Priestley and Chao (1972) defined the following estimator:

$$\hat{m}_{PC}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - X_i) Y_i. \tag{1.10}$$

For the random design model, with X-values chosen between 0 and 1, the Priestley-Chao estimator is defined by:

$$\hat{m}_{PC}(x) = \sum_{i=1}^{n} (X_i - X_{i-1}) K_h(x - X_i) Y_i, \tag{1.11}$$

where $\{(X_i, Y_i)\}_{i=1}^n$ are assumed to be ordered by the X-values. For the Priestley-Chao estimator the weights need not necessarily add up to one.

c) The Gasser-Müller estimator

The Gasser-Müller estimator is a modification of an earlier version of Priestley and Chao (1972). Gasser and Müller (1979) proposed the following estimator

$$\hat{m}_{GM}(x) = \sum_{i=1}^{n} Y_i \int_{s_i - 1}^{s_i} K_h(x - u)\, du, \tag{1.12}$$

where $\{(X_i, Y_i)\}_{i=1}^n$ is assumed to be ordered by the X-values and $s_i = (X_i + X_{i+1})/2$, $X_0 = -\infty$, $X_{n+1} = \infty$. Note that the sum of the weights in (1.12) is one. This estimator was originally proposed for equispaced designs, but can also be used for non-equispaced designs. The Gasser-Müller and Priestley-Chao estimators are conveniently defined without the random denominator of the Nadaraya-Watson estimator in (1.9). These definitions make both estimators easier to handle, for example when taking derivatives of the estimator or when deriving asymptotic properties.

d) The kernel function

Kernel functions are usually symmetric, real-valued probability density functions that are continuous and bounded, with $\int K(u)\, du = 1$ (Härdle, 1990:24). A variety of different kernel functions are applied in literature. Four of the popular kernel functions are particular cases of the family

$$K(u; p) = \left\{2^{2p+1} B(p + 1, p + 1)\right\}^{-1} (1 - u^2)^p I(|u| \leq 1), \tag{1.13}$$

where $B(\cdot, \cdot)$ is the Beta function which can be defined as

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x + y)} \tag{1.14}$$

and Γ is the Gamma function defined by

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt. \tag{1.15}$$

Now, using $p = 0$ in (1.13) the uniform kernel is obtained. Substituting $p = 1$ results in the Epanechnikov kernel, whereas $p = 2$ delivers the biweight kernel and $p = 3$ the triweight kernel. The Gaussian kernel can be seen as the limiting case where $p \to \infty$. Table 1.1 shows the definitions of these popular kernel while Figure 1.1 displays the forms of the kernel functions.

Kernel	Expression	p
Triangular	$(1 - \|u\|)I(\|u\| \leq 1)$	-
Uniform	$\frac{1}{2}I(\|u\|) \leq 1$	0
Epanechnikov	$\frac{3}{4}(1 - u^2)I(\|u\| \leq 1)$	1
Biweight	$\frac{15}{16}(1 - u^2)^2 I(\|u\| \leq 1)$	2
Triweight	$\frac{35}{32}(1 - u^2)^3 I(\|u\| \leq 1)$	3
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-u^2/2}$	∞

Table 1.1: Examples of popular kernel functions

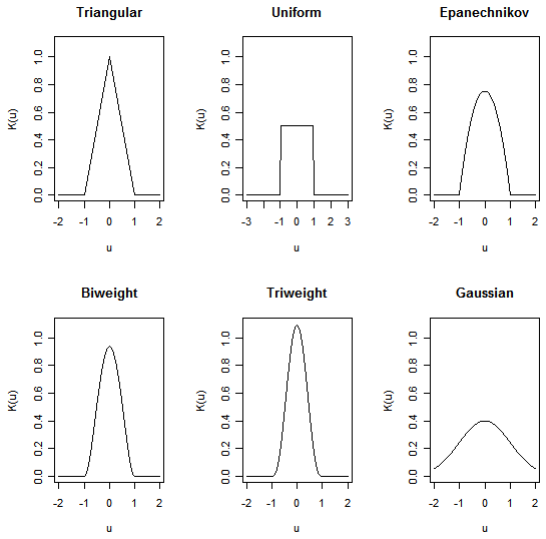

Figure 1.1: Examples of popular kernel functions

e) The bandwidth

The bandwidth h, also referred to as the *smoothing parameter*, is a non-negative number controlling the size of the neighbourhood around x (Härdle, 1990:18). The shape of the smooth depends greatly on the choice of h. A local average over too large a neighbourhood produces an extremely oversmoothed curve which results in a

biased estimate $\hat{m}(x)$ (Härdle, 1990:18). If one defines a smoothing parameter that corresponds to a very small neighbourhood, then only a small number of observations contribute non-negligibly to the estimate $\hat{m}(x)$ at x making it very rough and wiggly (Härdle, 1990:18). In this case the variance of $\hat{m}(x)$ is inflated. A trade-off between reducing the variance by increasing h and keeping the bias low by decreasing h, exists. This is called the bias-variance trade-off problem, which is further discussed in Section 1.6. Methods for choosing the smoothing parameter are discussed in Section 1.7.

1.4.2 Nearest neighbour regression smoothing

All of the kernel estimators, defined in the previous section, are a weighted average of the response variables in a fixed neighbourhood around x, determined in shape by the kernel K and the bandwidth h (Härdle, 1990:42). Note that the number of data points varies from neighbourhood to neighbourhood. The k-nearest neighbour estimator (k-NN) makes use of a varying neighbourhood in contrast to previous methods where the neighbourhood was fixed. This neighbourhood consists of those X-variables which are among the k-nearest neighbours of x in Euclidean distance. The k-NN estimate in the point x is then calculated as the weighted average of the response variables whose corresponding X-values fall in the neighbourhood. Now, the number of data points in each neighbourhood stays constant. Härdle (1990:42) defines the k-NN estimator as

$$\hat{m}_k(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ki}(x)Y_i, \tag{1.16}$$

where $\{W_{ki}(x)\}_{i=1}^{n}$ is a weight sequence defined through the set of indexes

$$J_x = \{i : X_i \text{ is one of the } k \text{ nearest observations to } x\}.$$

With this set of indexes of neighbouring observations the k-NN weight sequence is constructed:

$$W_{ki}(x) = \begin{cases} n/k, & \text{if } i \in J_x; \\ 0, & \text{otherwise.} \end{cases}$$

The smoothing parameter k regulates the degree of smoothness of the estimated curve. According to Härdle (1990:43) one can consider the following arguments regarding the size of the value of k. Firstly, consider for fixed n the case where k becomes larger than n. The k-NN estimator is then equal to the average of the dependent variable. Secondly, consider the case when $k = 1$ in which observations are reproduced at X_i, and for an x between two adjacent predictor variables a step function is obtained with a jump in the middle between the two observations. A smoothing parameter selection problem is observed. The first aim is to reduce the variance of the estimator by letting k tend to infinity as a function of the sample size. The second aim is to reduce the bias of the estimator. This can be achieved by letting the neighbourhood around x asymptotically shrink to zero. These two aims are conflicting. A trade-off situation between the reduction of the observational noise and a good approximation of the regression function arises. A bias-variance trade-off problem is faced, which is further discussed in Section 1.6.

1.4.3 Local polynomial fitting

This section is based on Fan and Gijbels (1996:57-58) and Fan (1992:1000). Suppose that the $(p+1)^{th}$ derivative of $m(x)$ at the point x_0 exists. The unknown regression function $m(x)$ is then approximated locally by a polynomial of order p. A Taylor expansion gives, for x in a neighbourhood of x_0,

$$m(x) \approx m(x_0) + m'(x_0)(x - x_0) + \frac{m''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{m^{(p)}(x_0)}{p!}(x - x_0)^p. \tag{1.17}$$

This polynomial is fitted locally by a weighted least squares regression method, i.e., by minimising

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_{j=0}^{p} \beta_j (X_i - x_0)^j \right\}^2 K_h(X_i - x_0), \tag{1.18}$$

where $K_h(\cdot)$ denotes a kernel function and h is a bandwidth. With $p = 1$ the estimator $\hat{m}_{LL}$ is termed a *local linear regression smoother* or a *local linear fit*. The problem of estimating $\hat{m}_{LL}$ is equivalent to estimating the intercept β_0. Now consider a weighted local linear regression: finding β_0 and β_1 to minimise

$$\sum_{i=1}^{n} \{Y_i - \beta_0 - \beta_1(X_i - x_0)\}^2 K_h(X_i - x_0). \tag{1.19}$$

Let $\hat{\beta}_0$ and $\hat{\beta}_1$ be the solution to the weighted least squares problem in (1.19). Simple calculations yield

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}. \tag{1.20}$$

The local linear regression smoother is defined by

$$\hat{m}_{LL}(x) = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}, \tag{1.21}$$

with

$$w_i = K\left(\frac{x - X_i}{h}\right)[s_{n,2} - (x - X_i)s_{n,1}], \tag{1.22}$$

where

$$s_{n,l} = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)(x - X_i)^l, \ l = 1, 2. \tag{1.23}$$

Local polynomial fitting is an attractive method both from theoretical and practical points of view. This method adapts easily to the random design setting and the fixed design setting (Fan, 1992:998). *A very important advantage of local polynomial fitting is that the bias at the boundary stays automatically of the same order as in the interior, without use of specific boundary kernels.*

1.4.4 Spline smoothing

For the sake of completeness, spline smoothing is briefly discussed, although it will not be used in the simulation studies. According to Fan and Gijbels (1996:39), a polynomial function, possessing all derivatives at all locations, is not very flexible for approximating curves with different degrees of smoothness at different locations. A method to improve the flexibility is to allow the derivatives of the approximating function to have discontinuities at certain

locations. This can be done by fitting piecewise polynomials or splines, resulting in the spline method. The points where derivatives of the approximating function could have discontinuities are called knots. Consider the homoskedastic model

$$Y_i = m(X_i) + \varepsilon_i,\ i = 1, \ldots, n,$$

where ε_i are independent, identically distributed with zero mean and common variance σ^2. The homoskedasticity of the model is important to the development of spline smoothing techniques, although those techniques can also be applied to heteroskedastic models. Now a procedure for an automatic selection of knots is introduced, called spline smoothing. Spline smoothing is discussed using the input of Fan and Gijbels (1996:43-44).

To motivate the spline smoothing procedure, consider again the least squares problem of finding a function m that minimises

$$\sum_{i=1}^{n} \{Y_i - m(X_i)\}^2 .$$

The solution to this naive least squares problem can be any function m which interpolates the data. Such a solution is undesirable for most statistical applications since it is usually not unique and too wiggly and not structure-orientated. It does therefore not describe the data in an elegant way and often produces a model as complex as the original data. From a statistical modelling point of view, it over-parameterised the model, resulting in large variability of the estimated parameters. The residuals $\hat{\varepsilon}_i = Y_i - \hat{m}(X_i)$ from this naive approach are usually close to zero. This strongly contradicts a homoskedastic model. One cannot expect that the realisations of the uncorrelated random noise ε_i are all zero. A clear shortcoming of this approach is that we did not impose a penalty for over-parametrisation. A convenient way for introducing such a penalty is via the roughness, popularly measured by $\int \{m''(x)\}^2 dx$. This leads to the following penalised least squares regression: find $\hat{m}_\lambda$ that minimises

$$\sum_{i=1}^{n} \{Y_i - m(X_i)\}^2 + \lambda \int \{m''(x)\}^2 \, dx, \tag{1.24}$$

for a non-negative real number $\lambda > 0$, called a smoothing parameter. Expression (1.24) consists of two parts. The first part penalizes the lack of fit, which is in some sense the modelling bias. The second part puts a penalty on the roughness, which relates to the over-parametrisation. It is clear that $\lambda = 0$ corresponds to interpolation whereas $\lambda = +\infty$ results in a linear regression. As λ ranges from zero to infinity, the estimate ranges from the most complex model (interpolation) to the simplest model (linear model). Thus, the model complexity of the smoothing spline approach is effectively controlled by the smoothing parameter λ. The estimator $\hat{m}_\lambda$ is referred to as the smoothing spline estimator.

It is well-known that a solution to the minimisation of (1.24) is a cubic spline on the interval $[X_{(1)}, X_{(n)}]$. This solution is also unique in this data range. Moreover it can easily be argued that $\hat{m}_\lambda$ is linear in the responses:

$$\hat{m}_\lambda(x) = \frac{1}{n} \sum_{i=1}^{n} W_{\lambda i}(x) Y_i. \tag{1.25}$$

The connections between kernel regression and smoothing splines have been established theoretically by Silverman (1984). In particular, Silverman (1984) pointed out that the smoothing spline is basically a local kernel

average with a variable bandwidth. For X_i away from the boundary, and for n large and λ relatively small,

$$W_{\lambda i}(x) \approx f(X_i)^{-1} h(X_i)^{-1} K_s \left(\frac{x - X_i}{h(X_i)} \right),$$

where

$$h(X_i) = \left[\frac{\lambda}{n f(X_i)} \right]^{1/4}$$

and

$$K_s(t) = 0.5 \exp(-|t|/\sqrt{2}) \sin(|t|/\sqrt{2} + \pi/4).$$

The smoothing parameter λ can be chosen objectively by the data. One approach is to select λ via the minimisation of the cross-validation criterion (Section 1.7.1).

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \{Y_i - \hat{m}_{\lambda,i}(X_i)\}^2, \tag{1.26}$$

where $\hat{m}_{\lambda,i}$ is the estimator arising from (1.24) without using the i^{th} observation.

1.4.5 Robust regression

For the sake of completeness, some robust regression methods are briefly discussed, although none of it will be used in the simulation studies. One can identify two kinds of robustness for an estimator: model and distributional robustness. Model robustness implies that an estimator can adjust well to departures from an assumed model (Eubank, 1988:173). Distributional robustness implies that the estimator is not sensitive to outliers (Eubank, 1988:173). If data points are identified as outliers it does not mean that they are not part of the joint distribution of the data or that they contain no information for estimating the regression curve. It means that these data points are too small a fraction of the data to be allowed to dominate the small-sample behaviour of the statistics to be calculated (Härdle, 1990:190). Kernel estimators can be viewed as being model robust because they are capable of fitting rather general regression curves. However, they are not totally resistant to the effects of outliers. Methods for handling data sets with outliers exist and are referred to as robust or resistant methods. Härdle (1990:69,193-195) discussed a variety of robust regression methods which are briefly stated below.

a) Median smoothing

In the case of median smoothing the aim of approximation is the conditional median curve $\text{med}(Y|X = x)$ rather than the conditional mean curve. A sequence of local medians of the response variables defines the median smoother. Härdle (1990:69) defines the median smoother as

$$\hat{m}_{med}(x) = \text{med}\{Y_i : i \in J_x\},$$

where

$$J_x = \{i : X_i \text{ is one of the } k\text{-nearest observations to } x\}.$$

A local average smoothing technique is not robust against outliers. Moving a response observation to infinity would drag the smooth to infinity as well. Therefore, local averaging smoothing has unbounded capacity to

be influenced by far out observations (Härdle, 1990:191). By downweighting large residuals, resistance against outliers can be achieved. Since only the median response value is used in median smoothing, responses leading to large residuals will carry no weight. A disadvantage of median smoothing is that it produces rough and wiggly curves.

b) L-smoothing

The discussion of this smoothing technique is based on Härdle (1990:193-194). This smoothing technique is based on the local trimmed averages of the response variables. If $Z_{(1)}, Z_{(2)}, \ldots, Z_{(n)}$ denotes the order statistic from n observations $\{Z_j\}_{j=1}^n$, a trimmed average is defined by

$$\bar{Z}_\alpha = (n - 2[\alpha n])^{-1} \sum_{j=[\alpha n]}^{n-[\alpha n]} Z_{(j)}, \quad 0 < \alpha < 1/2,$$

the mean of the inner $100(1-2\alpha)$ percent of the data. A local trimmed average at the point x from the regression data $\{(X_i, Y_i)\}_{i=1}^n$ is defined as a trimmed mean of the response variables Y_i such that X_i is in a neighbourhood of x. L-smoothing is a resistant technique since the extreme values at a point x do not enter the local averaging procedure. In general one considers a conditional L-functional

$$l(x) = \int_0^1 J(v) F^{-1}(v|x)\, dv, \tag{1.27}$$

where $F^{-1}(v|x) = \inf\{y : F(y|x) \geq v\},\ 0 < v < 1$, denotes the conditional quantile function associated with $F(\cdot|x)$, the conditional distribution function of Y given $X = x$. Consider the following choices for $J(v)$:

- J(v)$\equiv$ 1
$$l(x) = \int_0^1 F^{-1}(v|x)\, dv = \int_{F^{-1}(0|x)}^{F^{-1}(1|x)} y\, dF(y|x) = m(x),$$
where $y = F^{-1}(v|x)$. In this case $l(x)$ reduces to the regression function $m(x)$.

- $J(v) \equiv I(\alpha \leq v \leq 1-\alpha)/(1-2\alpha)$ where $0 < \alpha < 1/2$, with symmetric conditional distribution function,
$$l(x) = \frac{1}{1-2\alpha} \int_\alpha^{1-\alpha} F^{-1}(v|x)\, dv = \int_{F^{-1}(\alpha|x)}^{F^{-1}(1-\alpha|x)} y\, dF(y|x),$$
where the substitution $y = F^{-1}(v|x)$ was used. Median smoothing is a special case of L-smoothing with $\alpha = 1/2$.

  In practice $F(\cdot|x)$ is unknown and has to be estimated. Let $\hat{F}(\cdot|x)$ denote an estimator of $F(\cdot|x)$. Consider the following choices of $\hat{F}(\cdot|x)$:

- Take $\hat{F}(\cdot, x) = F_n(\cdot|x)$, the empirical conditional distribution function. Then
$$\hat{l}(x) = \int_{F_n^{-1}(\alpha|x)}^{F_n^{-1}(1-\alpha|x)} y\, dF_n(y|x) = (n - 2[\alpha n])^{-1} \sum_{j=[\alpha n]}^{n-[\alpha n]} Y_{(j)} = \bar{Y}_\alpha,$$
is the trimmed average of the response observations such that the corresponding X_i's are in a neighbourhood of x.

- Estimate $F(\cdot|x)$ by the kernel technique:

$$\hat{F}_h(t|x) = \frac{n^{-1}\sum_{i=1}^n K_h(x - X_i)I(Y_i \leq t)}{\hat{f}_h(x)},$$

to obtain

$$\hat{m}_h^L(x) = \int_0^1 J(v)\hat{F}_h^{-1}(v|x)\, dv.$$

Asymptotic results for L-smoothers were derived by Stute (1984), Owen (1987) and Härdle, Janssen and Serfling (1988b).

c) R-smoothing

The motivation for R-smoothing techniques stems from rank tests. This smoothing method is derived from R-estimates of location. Assume $F(\cdot|x)$ is symmetric around $m(x)$ and that J is a non-decreasing function defined on $(0, 1)$ such that $J(1 - s) = -J(s)$. Then the score

$$T(\theta, F(\cdot|x)) = \int_{-\infty}^{\infty} J\left(\frac{1}{2}(F(v|x) + 1 - F(2\theta - v|x))\right) dF(v|x)$$

is zero for $\theta = m(x)$. Since $F(\cdot|x)$ is an unknown quantity it needs to be estimated. Let $F_n(\cdot|x)$ denote such an estimate of the conditional distribution function $F(\cdot|x)$, then this score should be roughly zero for a good estimate of $m(x)$. In general the solution of $T(\theta, \hat{F}(\cdot|x))$ is not unique or may have irregular behaviour. Cheng and Cheng (1987) therefore suggested

$$\hat{m}_h^R(x) = \frac{1}{2}\left[\sup\{\theta : T(\theta, F_n(\cdot|x)) > 0\} + \inf\{\theta : T(\theta, F_n(\cdot|x)) < 0\}\right].$$

d) M-smoothing

In Section 1.4.1 we have seen that we can construct an estimate for $m(x)$ by local averaging. Such an estimate can be written as

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^n W_{ni}(x)Y_i.$$

Estimators of this form can also be seen as the solution to a local least squares problem

$$\hat{m}(x) = \arg\min_\theta \left\{\frac{1}{n}\sum_{i=1}^n W_{ni}(x)(Y_i - \theta)^2\right\}. \tag{1.28}$$

The basic idea of M-smoothers is to reduce the influence of outlying observations by the use of a non-quadratic loss function instead of the quadratic loss function used in (1.28). Assume that the conditional distribution $F(\cdot|x)$ is symmetric. This assumption ensures that we are still estimating $m(x)$, the conditional mean curve. Now, a robust kernel M-smoother is defined as

$$\hat{m}_h^M(x) = \arg\min_\theta \left\{\frac{1}{n}\sum_{i=1}^n W_{hi}(x)\rho(Y_i - \theta)\right\}, \tag{1.29}$$

where ρ is a loss function and $\{W_{hi}(x)\}_{i=1}^n$ denotes a positive kernel weight sequence. An example of such a loss function is

$$\rho(u) = \begin{cases} \frac{1}{2}u^2, & \text{if } |u| \leq c; \\ c|u| - \frac{1}{2}c^2, & \text{if } |u| > c. \end{cases}$$

The constant c regulates the degree of resistance. For large values of c one obtains the ordinary quadratic loss function. For small values one achieves more robustness. To solve (1.29), the expression must be differentiated to θ and set equal to zero,

$$\frac{1}{n}\sum_{i=1}^{n} W_{hi}(x)\varphi(Y_i - \theta) = 0,$$

where $\varphi = \rho'$. Since the kernel M-smoother is implicitly defined, it requires iterative numerical methods. A fast algorithm based on the Fast Fourier Transform and a "one-step" approximation to $\hat{m}_h^M$ are given in Härdle (1987). A wide variety of possible φ-functions yield consistent estimators. The choice $\varphi(u) = u$ yields the ordinary kernel smoother $\hat{m}_h(x)$. In the setting of spline smoothing an M-type spline is defined as

$$\arg\min_{m} \left\{ \frac{1}{n}\sum_{i=1}^{n} \rho(Y_i - m(X_i)) + \lambda \int [m''(x)]^2 dx \right\}, \tag{1.30}$$

where ρ is a loss function with lighter tails than the usual quadratic form.

1.5 Discrepancy measures

In Section 1.4 various nonparametric regression estimates were discussed, all depending on some bandwidth. If the smoothing parameter is chosen as a suitable function of the sample size n, all of the above smoothers converge to the true curve if the number of observations increases (Härdle, 1990:89). However, the convergence of an estimator is not enough. *Several distance measures are available for assessing the extent of uncertainty and the speed of convergence of the above defined smoothers.* The aim of this section is to introduce various methods developed for assessing the accuracy and precision of a regression estimator. In the present study discrepancy measures between the regression function estimator and the true regression function are defined. The measure can be evaluated pointwise (Section 1.5.1), or globally (Section 1.5.3).

1.5.1 Pointwise discrepancy measures

A popular pointwise discrepancy measure is the mean squared error (MSE) (Wand & Jones, 1995). The MSE is a pointwise discrepancy measure since it quantifies the accuracy and precision of the estimator in a single point x. The MSE is defined as the sum of the variance and the squared bias of the estimator. First, the two components, bias and variance, of the MSE are defined and asymptotic expressions for the MSE for kernel and k-NN regression estimators are presented.

a) Bias

The pointwise bias of a regression function estimator $\hat{m}$, measures how close the expected estimated value is to the true regression function m. The bias is a measure of the difference between the expected value of an estimator and the parameter it is attempting to estimate. Therefore the bias is a measure of the accuracy of an estimator. The pointwise bias is defined as:

$$\text{Bias}[\hat{m}(x)] = E\left[\hat{m}(x)\right] - m(x). \tag{1.31}$$

If the estimator overestimates the parameter value it results in a positive value for the bias, or the estimator might underestimate the parameter value and this results in a negative value for the bias. If the bias is equal to zero the estimator is called unbiased. This means that the expected value of the estimator in the point x is exactly equal to the parameter value in the point x, i.e., $E[\hat{m}(x)] = m(x)$. Note that, in general, the squared bias term is increasing in h, meaning that if the value of h increases, the value of the squared bias will increase as well.

b) Variance

The pointwise variance of a regression function estimator $\hat{m}$ measures the spread of the estimator around the expected value of the estimator. The variance is a measure of the precision of the estimator. The pointwise variance is defined as

$$\text{Var}[\hat{m}(x)] = E\left[(\hat{m}(x) - E\left[\hat{m}(x)\right])^2\right]. \tag{1.32}$$

Note that, the variance term is always positive and should be as small as possible. A small variance term indicates that the regression function estimator is precise when estimating the true regression function. Note that the variance term is decreasing in h, which means that if the value of h increases, the value of the variance term will decrease.

c) Mean squared error

The bias is a measure of the accuracy of the estimator and the variance is a measure of the precision of the estimator. The mean squared error, or MSE, is a measure that incorporates both these discrepancy measures. The MSE is defined as

$$\text{MSE}[\hat{m}(x)] = E[(\hat{m}(x) - m(x))^2]. \tag{1.33}$$

It is now shown how the bias and variance combine to give the value of the MSE.

$$\begin{aligned}
\text{MSE}[\hat{m}(x)] &= E[(\hat{m}(x) - m(x))^2] \\
&= E[\{\hat{m}(x) - E[\hat{m}(x)] + E[\hat{m}(x)] - m(x)\}^2] \\
&= E[\{\hat{m}(x) - E[\hat{m}(x)]\}^2 + 2\{\hat{m}(x) - E[\hat{m}(x)]\}\{E[\hat{m}(x)] - m(x)\} + \{E[\hat{m}(x)] - m(x)\}^2] \\
&= \text{Var}[\hat{m}(x)] + 2E[\hat{m}(x) - E[\hat{m}(x)]]\{E[\hat{m}(x)] - m(x)\} + E[\{E[\hat{m}(x)] - m(x)\}^2] \\
&= \text{Var}[\hat{m}(x)] + 2\{E[\hat{m}(x)] - m(x)\}E[\{\hat{m}(x) - E[\hat{m}(x)]\}] + \{E[\hat{m}(x)] - m(x)\}^2 \\
&= \text{Var}[\hat{m}(x)] + 2\{E[\hat{m}(x)] - m(x)\}E[\{\hat{m}(x) - E[\hat{m}(x)]\}] + \{\text{Bias}[\hat{m}(x)]\}^2.
\end{aligned}$$

Therefore we have the following decomposition of the MSE:

$$\text{MSE}[\hat{m}(x)] = \text{Var}[\hat{m}(x)] + [\text{Bias}(\hat{m}(x))]^2. \tag{1.34}$$

1.5.2 Asymptotic properties of the MSE

The aim is to choose an estimator $\hat{m}(x)$ that minimises the discrepancy measure of interest. These measures depend on the kernel K and the smoothing parameter h. An objective of this study is to study the behaviour

of these discrepancy measures as a function of the smoothing parameter. One difficulty in this regard is that the dependence on h cannot be explicitly expressed in most situations. Therefore it is required to obtain simple expressions of these measures in order to see the dependence of these measures on h. This is achieved by studying measures in an asymptotic sense, that is in cases where the sample size n increases to infinity, $n \to \infty$. These approximations have relatively simple expressions that allow better understanding of the role of the smoothing parameter on the performance of the estimator.

The asymptotic bias and variance of the estimators under consideration have been studied extensively in the literature. Tables 1.2 and 1.3 provide theoretical approximation results of the fixed and random design cases respectively, given that the following list of assumptions hold. The rest of the discussion is based on Härdle (1990:29-30).

Assume the fixed design model with a one-dimensional predictor variable X and define

$$c_K = \int K^2(u)\,du. \tag{1.35}$$

$$d_K = \int u^2 K(u)\,du. \tag{1.36}$$

If the following assumptions hold, the asymptotic properties of the MSE are valid for the fixed and random designs as it is provided in Tables 1.2 and 1.3 below.

(A1) K has support $[-1, 1]$ with $K(-1) = K(1) = 0$.

(A2) m is twice continuously differentiable on a neighbourhood of the point x.

(A3) $\max_i |X_i - X_{i-1}| = O(n^{-1})$.

(A4) $\mathrm{Var}[\varepsilon_i] = \sigma^2,\ i = 1, \ldots, n$.

(A5) For kernel smoothers: $n \to \infty, h \to 0,\ nh \to \infty$.
For k-NN smoother: $n \to \infty, k \to 0,\ nk \to \infty$.

Chu and Marron (1991:411) added two additional assumptions in order to derive theoretical approximations in the random design case.

(A6) The marginal density f of X_j has a bounded and continuous first derivative and is bounded above 0 on a neighbourhood of x.

(A7) X_j and ε_j are uncorrelated.

The theoretical approximation results under the assumptions stated above are provided in Table 1.2 for the fixed design case and in Table 1.3 for the random design case. One can clearly see that in the fixed design case the Nadaraya-Watson, Priestley-Chao and the Gasser-Müller estimators have the same mean squared error properties. In the fixed design setting the k-NN weight sequence can be seen as a kernel sequence if one assumes the identification $h = \frac{k}{2n}$ (Härdle, 1990:76). The bias and variance for the random design case are drastically different. Pointwise bias and variance are complicated functionals not only of the regression curve m but also

of the marginal density f (Härdle, 1990:76). The aim is to select an estimator that minimises the discrepancy measure, i.e. the MSE in this case. It is obvious in all of the bias-variance combinations that the choice of h that will cause a reduction in the bias, will also be responsible for increasing the variance and vice versa. This is known as the bias-variance trade-off dilemma and is a mathematical quantification for the critical role of the bandwidth (Wand & Jones, 1995:22). The bias-variance trade-off in the context of nonparametric regression analysis is discussed in Härdle (1990). If one compares the local linear fit with the Nadaraya-Watson estimator one sees that the Nadaraya-Watson estimator uses one less parameter, but without reducing the asymptotic variance. In the case of the local linear fit this extra parameter makes it possible to reduce the bias. The Nadaraya-Watson estimator suffers from large bias at the region where the derivative of the regression function of the design density is large. The Nadaraya-Watson estimator is prone to large bias even when the true regression curve is linear. The Nadaraya-Watson estimator has a large order bias when estimating a curve at the boundary region. Many proposals such as reflection methods and boundary kernel methods are proposed to handle this problem, but they are less efficient than the automatic boundary correction of the local linear fit. The local linear fit is efficient in correcting boundary bias. Both the Nadaraya-Watson estimator and the local linear estimator are used in the present study.

Estimator	Bias	Variance	Reference
Nadaraya-Watson	$\frac{h^2}{2} d_K m''(x)$	$\frac{\sigma^2}{nh} c_K$	Gasser and Müller (1984:175)
Priestley-Chao	$\frac{h^2}{2} d_K m''(x)$	$\frac{\sigma^2}{nh} c_K$	Bias: Priestley and Chao (1972:389) Variance: Chu and Marron (1991:410)
Gasser-Müller	$\frac{h^2}{2} d_K m''(x)$	$\frac{\sigma^2}{nh} c_K$	Mack and Müller (1989:230)
k-NN	$\frac{1}{8} \left(\frac{k}{n}\right)^2 d_K m''(x)$	$\frac{2\sigma^2}{k} c_K$	Härdle (1990:76)

Table 1.2: Asymptotic MSE properties: Fixed design model

Estimator	Bias	Variance	Reference
Nadaraya-Watson	$\frac{h^2}{2} d_K \left(m''(x) + \frac{2m'(x)f'(x)}{f(x)} \right)$	$\frac{\sigma^2}{f(x)nh} c_K$	Härdle (1990:77)
Gasser-Müller	$\frac{h^2}{2} d_K m''(x)$	$\frac{3\sigma^2}{2f(x)nh} c_K$	Härdle (1990:77)
k-NN	$\frac{1}{24} \left(\frac{k}{n}\right)^2 \frac{m''(x)f(x)+2m'(x)f'(x)}{f^3(x)}$	$\frac{\sigma^2}{k}$	Härdle (1990:43)
Local linear	$m''(x)\frac{h^2}{2} d_K$	$\frac{\sigma^2}{f(x)nh} c_K$	Fan and Gijbels (1996:17)

Table 1.3: Asymptotic MSE properties: Random design model

1.5.3 Global discrepancy measures

A pointwise discrepancy measure such as the MSE gives an indication of the accuracy and precision of an estimator in a single point x. A measure that quantifies the quality of an estimator over the entire range of x-values is often desired. Some global measures of discrepancy are proposed by Härdle (1990:90) and are listed below.

1. The integrated absolute deviation:

$$d_L(\hat{m}, m) = \int |\hat{m}(x) - m(x)| f(x) w(x)\, dx. \tag{1.37}$$

2. The integrated squared error (ISE):

$$d_I(\hat{m}, m) = \int (\hat{m}(x) - m(x))^2 f(x) w(x)\, dx. \tag{1.38}$$

3. The averaged squared error (ASE):

$$d_A(\hat{m}, m) = \frac{1}{n} \sum_{i=1}^{n} (\hat{m}(X_i) - m(X_i))^2 w(X_i). \tag{1.39}$$

4. The maximal absolute deviation:

$$d_{L_\infty}(\hat{m}, m) = \sup_{x \in \mathcal{X}} |\hat{m}(x) - m(x)|, \tag{1.40}$$

   where the $\sup_x$ ranges over a set $\mathcal{X} \in \mathbb{R}^d$ of interest.

5. A conditional version of d_A:

$$d_C(\hat{m}, m) = E[d_A(\hat{m}, m) | X_1, \ldots, X_n]. \tag{1.41}$$

6. The mean integrated squared error (MISE):

$$\text{MISE} = E[d_I(\hat{m}, m)] = \int \text{MSE}[\hat{m}(x)]\, dx. \tag{1.42}$$

The measures in (1.37) to (1.41) are global in the sense that they cover the whole range of x-values. However, these measures are based on only the dataset at hand. Each dataset is merely one realisation of a random sample from the population. To recover the whole curve over the entire population, the expected value of the integrated squared error (ISE) could be taken. The resulting measure is the mean integrated squared error (MISE), given in (1.42).

1.6 Bias-variance trade-off

It is shown in the tables of Section 1.5 that the bias is asymptotically proportional to h^2, so for this quantity to decrease one needs to select h small. However, taking h small means an increase in the leading term of the variance since this quantity is proportional to $\frac{1}{nh}$. Therefore, as n increases, h should vary in such a way that each of the components of the MSE becomes smaller (Wand & Jones, 1995:22). It is clear that the choice of the smoothing parameter plays a very important role in the performance of nonparametric regression estimators (Altman, 1992:180). According to Altman (1992:180) one can consider two extreme scenarios: In the first scenario a small bandwidth is chosen. This will cause the estimator to follow the data very closely (small bias) but with much wigglyness (large variance). This is called overfitting. In the second scenario a large bandwidth is chosen. This will result in a very smooth curve (small variance) that lies near the mean of the original data (large bias). This is called underfitting. Since the MSE is the sum of the squared bias and the variance, the choice of smoothing parameter that minimises the MSE will be a good local choice. Choosing a smoothing parameter that will be optimal over the entire curve also seems feasible. To achieve this, a smoothing parameter that minimises the ASE, ISE, or MISE is demanded. Such choices of h will be theoretically optimal. A vast literature exists regarding the choice of smoothing parameter. Brief discussions of the basic ideas follow below.

1.7 Methods for choosing the smoothing parameter

Section 1.6 showed that the problem of how much to smooth is of great importance in nonparametric regression. In this section three bandwidth selection methods are introduced using the input of several well-known authors on this topic.

The basic idea surrounding smoothing parameter selection algorithms is to find an expression for an estimator of the discrepancy measure. According to Härdle (1990:150) the hope is then that the smoothing parameter minimising this estimate will also be a good minimiser for the discrepancy measure itself. Consider the average squared error (ASE) and expand the ASE as

$$\begin{aligned} d_A(\hat{m}, m) &= \frac{1}{n}\sum_{j=1}^{n}(\hat{m}_h(X_j) - m(X_j))^2 w(X_j) \\ &= \frac{1}{n}\sum_{j=1}^{n} m^2(X_j)w(X_j) + \frac{1}{n}\sum_{j=1}^{n}\hat{m}_h^2(X_j)w(X_j) - 2\frac{1}{n}\sum_{j=1}^{n} m(X_j)\hat{m}_h(X_j)w(X_j). \end{aligned} \tag{1.43}$$

Note that the first term is independent of the smoothing parameter and the second term can be computed entirely from the data. If the third term could be estimated and if it vanished faster than d_A itself tends to zero, then indeed a device for selecting the bandwidth could be established quite easily. A naive estimate of the third term would be

$$\frac{1}{n}\sum_{j=1}^{n} Y_j \hat{m}_h(X_j) w(X_j),$$

where $m(X_j)$ is replaced by the observation Y_j at X_j. This is equivalent to considering the so-called resubstitution estimate of the prediction error

$$p(h) = \frac{1}{n}\sum_{j=1}^{n} [Y_j - \hat{m}_h(X_j)]^2 w(X_j) \tag{1.44}$$

as a device for selecting h. Unfortunately the prediction error $p(h)$ is an increasing function of h, therefore the best bandwidth will always be the smallest bandwidth. This is due to the fact that $p(h)$ is a biased estimate of the ASE, because it uses the same dataset to construct and assess the estimate. To see this, Härdle (1990:151-152) shows that expanding $p(h)$ leads to a last term

$$C_{1n}(h) = -\frac{2}{n}\sum_{j=1}^{n} \varepsilon_j \left[\frac{1}{n}\sum_{i=1}^{n} W_{hi}(X_j) Y_i - m(X_j)\right] w(X_j), \tag{1.45}$$

where W_{hi} denotes the Nadaraya-Watson weight sequence defined in (1.9). $C_{1n}(h)$ has expectation

$$-\frac{2}{n}\sum_{j=1}^{n} \left[\frac{1}{n} W_{hj}(X_j)\sigma^2(X_j)\right] w(X_j).$$

This quantity tends to zero at the same rate as the variance component of the ASE, which explains why $p(h)$ is a biased estimate for the ASE. Ways to obtain an unbiased estimate of the ASE should be pursued. There are at least three possible ways to find an unbiased estimate of d_A:

1. a leave-out technique to obtain zero expectation for (1.45);
2. a modification of $p(h)$ such that bias terms like that of (1.45) cancel asymptotically;
3. a plug-in method, using asymptotics of optimal bandwidth sequences.

These three well-known methods are now introduced using the input of several well-known authors on this topic.

1.7.1 Cross-validation

The following is based on Härdle and Marron (1985b:1466-1467). One of the crucial points in applying $\hat{m}(x)$ is the choice of the bandwidth h. Suppose that h is in some set $H_n \subseteq R_n^+$ of interest. A bandwidth-selection rule $\hat{h}$ is an H_n-valued function of $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$. Let the distance $d(\hat{m}, m)$ denote a given measure of accuracy for the estimator $\hat{m}(x)$. In this case the integrated squared error is used as distance measure. Define the integrated squared error as

$$\begin{aligned} d_I(\hat{m}, m) &= \int [\hat{m}(x) - m(x)]^2 w(x) f(x)\, dx \\ &= \int \hat{m}^2(x) w(x) f(x)\, dx - 2\int \hat{m}(x) m(x) w(x) f(x)\, dx + \int m^2(x) w(x) f(x)\, dx. \end{aligned} \tag{1.46}$$

Since the last summand is independent of h, the goal of minimising this loss is equivalent to that of minimising

$$\int \hat{m}^2(x)w(x)f(x)\,dx - 2\int \hat{m}(x)m(x)w(x)f(x)\,dx. \tag{1.47}$$

However, this cannot be realised in practice because this quantity depends on the unknowns m and f. Observe also that the second term in (1.47) may be written as

$$\int \hat{m}(x)m(x)w(x)f(x)\,dx = E[\hat{m}(X)Yw(X)]. \tag{1.48}$$

This motivates estimating the second term by

$$\frac{1}{n}\sum_{j=1}^{n}[\hat{m}_j(X_j)Y_jw(X_j)], \tag{1.49}$$

where $\hat{m}_j$ is the leave-one-out estimator given by

$$\hat{m}_j(x) = \frac{1}{n-1}\sum_{i\neq j}\frac{1}{h}K\left(\frac{x-X_i}{h}\right)Y_i/\hat{f}_j(x), \tag{1.50}$$

$$\hat{f}_j(x) = \frac{1}{n-1}\sum_{i\neq j}\frac{1}{h}K\left(\frac{x-X_i}{h}\right). \tag{1.51}$$

Similarly, the first term in (1.47) may be approximated by

$$\frac{1}{n}\sum_{j=1}^{n}[\hat{m}_j^2(X_j)w(X_j)]. \tag{1.52}$$

Therefore, it seems reasonable to take h to minimise the sum of the estimates of the first two terms. Adding a term which is independent of h does not change the bandwidth-selection rule, which is then: Choose $\hat{h}$ to minimise

$$CV(h) = \frac{1}{n}\sum_{j=1}^{n}[Y_j - \hat{m}_j(X_j)]^2w(X_j). \tag{1.53}$$

1.7.2 Penalising functions

The following is based on Härdle (1990:154). It is clear that $h \to 0$ leads to interpolation, i.e. $\hat{m}_h(X_j) \to Y_j$, so that the function to be minimised, namely $p(h)$ defined in (1.44), could become arbitrarily small. On the other hand, this would surely cause a very large variance of $\hat{m}_h$ which indicates that such a criterion function would not balance the bias and variance. Consequently, the corrected ASE penalises when choosing h too small in a reasonable sense. Define

$$G(h) = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{m}(X_j))^2\Xi(n^{-1}W_{hj}(X_j))w(X_j), \tag{1.54}$$

where $\Xi(\cdot)$ is a penalising function with first-order Taylor expansion

$$\Xi(u) = 1 + 2u + O(u^2), u \to 0. \tag{1.55}$$

The smaller the bandwidth h the larger $W_{hj}(X_j)$ and the penalising factor $\Xi(n^{-1}W_{hj}(X_j))$ increases. By conducting a first-order Taylor expansion of G and disregarding lower order terms it is easy to show that $G(h)$ is roughly equal to $ASE(h)$, up to a shift that is independent of h. The following list presents a number of proposed penalising functions that satisfy the expansion $\Xi(u) = 1 + 2u + O(u^2), u \to 0$ (Härdle, 1990:166-167):

- Generalised cross-validation (Craven & Wahba, 1979):

$$\Xi_{GCV}(n^{-1}h^{-1}) = (1 - n^{-1}h^{-1}K(0))^{-2}.$$

- Akaike's information criterion (Akaike, 1970):

$$\Xi_{AIC}(n^{-1}h^{-1}) = \exp(2n^{-1}h^{-1}K(0)).$$

- Finite prediction error (Akaike, 1974):

$$\Xi_{FPE}(n^{-1}h^{-1}) = (1 + n^{-1}h^{-1}K(0))/(1 - n^{-1}h^{-1}K(0)).$$

- Shibata's model selector (1981):

$$\Xi_{S}(n^{-1}h^{-1}) = 1 + 2n^{-1}h^{-1}K(0).$$

- Rice's bandwidth selector (1984):

$$\Xi_{T}(n^{-1}h^{-1}) = (1 - 2n^{-1}h^{-1}K(0))^{-1}.$$

1.7.3 Plug-in method

The discussion on the plug-in method is based on Ruppert *et al.* (1995) and Fan and Gijbels (1996). In contrast to the cross-validation and penalising function methods, the plug-in methods try to minimise the MISE or the AMISE. Then the idea is to replace the unknown quantities by mainly three different strategies:

1. Rule-of-thumb bandwidth selector h_{ROT}:
   The unknown quantities are replaced by parametric estimators. In this strategy parametric polynomial regression is used.

2. Direct plug-in bandwidth selector h_{DPI}:
   The unknown quantities are replaced by nonparametric estimates, where pilot bandwidths for nonparametric estimators need to be chosen.

3. Bootstrap based bandwidth selection h_{WB}:
   The unknown expression is estimated by the wild bootstrap method. More detail on bootstrap methods is provided in Chapter 2. The wild bootstrap method (h_{WB}) directly estimates the MISE of $\hat{m}_h$ and then minimises with respect to h. This method requires a pilot bandwidth.

a) Rule-of-thumb bandwidth selection

In this section rule-of-thumb bandwidth selection methods are discussed for the Nadaraya-Watson estimator and the local linear estimator. In Chapter 4 algorithms corresponding with the bandwidth selection methods are stated to enable the calculation of the smoothers.

- **Nadaraya-Watson estimator**

  Hansen (2009), shows that explicit expressions exist for the asymptotic variance and bias of the Nadaraya-Watson estimator. This can be simplified to derive the optimal bandwidth for the conditional mean integrated square error (MISE):

$$h_{opt} = \left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\int \sigma^2(x) w_0(x)\, dx}{\left\{\int \frac{1}{2} m''(x) + m'(x)\frac{f'(x)}{f(x)}\right\}^2 w_0(x) f(x)\, dx}\right]^{1/5} n^{-1/5}, \tag{1.56}$$

  where $c_K = \int u^2 K(u)\, du$ and $d_K = \int K^2(u)\, du$. Unfortunately, this optimal bandwidth expression contains unknown quantities $\sigma^2(\cdot), m'(\cdot), m''(\cdot), f(\cdot)$ and $f'(\cdot)$, which need to be estimated. Assuming the conditional variance $\sigma^2(x)$ is constant and substituting the pilot estimates $\hat{m}'(\cdot)$, $\hat{m}''(\cdot)$, $\hat{f}'(\cdot)$, $\hat{f}''(\cdot)$ and σ^2 into the expression for the optimal bandwidth in (1.56), the following expression is obtained

$$\left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\hat{\sigma}^2 \int w_0(x)\, dx}{\{\int \frac{1}{2}\hat{m}''(x) + \hat{m}'(x)\frac{\hat{f}'(x)}{\hat{f}(x)}\}^2 w_0(x)\hat{f}(x)\, dx}\right]^{1/5} n^{-1/5}.$$

  The denominator in the above expression can be estimated by

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\hat{m}''(X_i) + \hat{m}'(X_i)\frac{\hat{f}'(X_i)}{\hat{f}(X_i)}\right\}^2 w_0(X_i)$$

  which leads to the rule-of-thumb bandwidth selector

$$h_{ROT} = \left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\hat{\sigma}^2 \int w_0(x)\, dx}{\frac{1}{n}\sum_{i=1}^{n}\left\{\hat{m}''(X_i) + \hat{m}'(X_i)\frac{\hat{f}'(X_i)}{\hat{f}(X_i)}\right\}^2 w_0(X_i)}\right]^{1/5}. \tag{1.57}$$

  The basic idea of this bandwidth selection method is to replace the unknown quantities by parametric estimators. In this case parametric polynomial regression is used i.e., fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{1.58}$$

  where k is an appropriate integer. Utilising the standardised residual sum of squares from this parametric fit leads to an estimator for σ^2, i.e., $\hat{\sigma}^2$. Furthermore, this choice of a global fit enables calculations of the derivative functions $\hat{m}^{(k)}(x)$. The definition of the weight function, $w_0(x)$ and the estimation of the density function, $f(x)$ are discussed in Chapter 4. Also, an algorithm corresponding with the bandwidth selection method is stated to enable the calculation of the Nadaraya-Watson estimator.

- **Local linear estimator**

  Fan (1992:1000) showed that the optimal bandwidth for the conditional mean integrated square error (MISE) is

$$h_{opt} = \left(\frac{d_K \int f^{-1}(x)\sigma^2(x) w(x)\, dx}{c_K^2 \int (m''(x))^2 w(x)\, dx}\right)^{1/5} n^{-1/5}, \tag{1.59}$$

where $c_K = \int u^2 K(u)\,du$ and $d_K = \int K^2(u)\,du$. The optimal bandwidth defined in (1.59) contains the unknown quantities $\sigma^2(\cdot), m''(\cdot)$ and $f(\cdot)$, which needs to be estimated. Suppose one can take $w(x) = f(x)w_0(x)$ for some specific function w_0 and assume the conditional variance $\sigma^2(x)$ is constant. Substituting the pilot estimates $\hat{m}(\cdot)$ and σ^2 into the expression for the optimal bandwidth in (1.59), the following expression is obtained

$$\left(\frac{d_K\hat{\sigma}^2\int w_0(x)\,dx}{c_K^2\int(\hat{m}''(x))^2 w_0(x)f(x)\,dx}\right)^{1/5} n^{-1/5}. \tag{1.60}$$

The denominator in the above expression can be estimated by

$$\frac{1}{n}\sum_{j=1}^{n}\hat{m}''(X_i)^2 w_0(X_i),$$

which leads to the rule-of-thumb bandwidth selector

$$\hat{h}_{plug} = \left(\frac{d_K\hat{\sigma}^2\int w_0(x)\,dx}{c_K^2\sum_{j=1}^{n}\{\hat{m}''(X_i)\}^2 w_0(X_i)}\right)^{1/5}. \tag{1.61}$$

Again, the unknown quantities are replaced by parametric estimators, where parametric polynomial regression is used i.e., fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{1.62}$$

where k is an appropriate integer. Utilising the standardised residual sum of squares from this parametric fit leads to an estimator for σ^2, i.e., $\hat{\sigma}^2$. Furthermore, this choice of a global fit enables calculations of the derivative functions $\hat{m}^{(k)}(x)$. The definition of the weight function, $w_0(x)$ is discussed in Chapter 4 and an algorithm corresponding with the bandwidth selection method is stated to enable the calculation of the local linear estimator.

b) Direct plug-in bandwidth selection

For completeness sake an example of a direct plug-in bandwidth selector is discussed according to Ruppert *et al.* (1995). Fan (1992:1000) showed in the case of the local linear estimator that the optimal bandwidth for the conditional mean integrated square error (MISE) is

$$h_{opt} = \left(\frac{d_K\int f^{-1}(x)\sigma^2(x)w(x)\,dx}{c_K^2\int(m''(x))^2 w(x)\,dx}\right)^{1/5} n^{-1/5}. \tag{1.63}$$

Ruppert *et al.* (1995:1258) rewrites this optimal bandwidth as

$$h_{opt} = C_1(K)\left[\frac{\sigma^2(b-a)}{\theta_{22}n}\right]^{1/5}, \tag{1.64}$$

where $C_1(K)$ is a kernel dependent constant and $\theta_{rs} = \int m^{(r)}(x)m^{(s)}(x)f(x)\,dx$. For this approach a parametric pre-estimate in some stage is required. In this case piece-wise polynomial regression is used. Let N be the number of subsamples and let χ_j denote the j^{th} subsample of the ordered X_i's. If N divides the sample n and $t = n/N$, then $\chi_j = \{X_{((j-1)t+1}, \ldots, X_{(jt)}\}$. Let $\hat{m}_j^Q(x)$ be the least squares quartic fit obtained from data

having X_i values in χ_j. The term *quartic fit* refers to the process of fitting a polynomial of the fourth order. The *blocked quartic estimator* for θ_{rs} is

$$\hat{\theta}_{rs}^{Q}(N) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{N}(\hat{m}_j^{Q})^{(r)}(X_i)(\hat{m}_j^{Q})^{(s)}(X_i)1_{\{X_i\in\chi_j\}}. \tag{1.65}$$

Similarly, the blocked quartic estimator for σ^2 is

$$\hat{\sigma}_{Q}^{2}(N) = \frac{1}{n-5N}\sum_{i=1}^{n}\sum_{j=1}^{N}\{Y_i - \hat{m}_j^{Q}(X_i)\}^2 1_{\{X_i\in\chi_j\}}. \tag{1.66}$$

These estimators require a rule for choosing N. In this example Mallows' C_p (Mallows, 1973) is used. For blocked quartic fits, this involves choosing $\hat{N}$ from the set $\{1, 2, \ldots, N_{\max}\}$ to minimise

$$C_p(N) = \frac{RSS(N)\cdot(n-5N_{\max})}{RSS(N_{\max})} - (n-10N), \tag{1.67}$$

where $RSS(N)$ is the residual sum of squares on a blocked quartic fit over N blocks and take $N_{\max}$ to be of the form

$$N_{\max} = \max\{\min(\lfloor n/20\rfloor, N^*), 1\} \tag{1.68}$$

for some positive integer N^*. In this example $N^* = 5$. Though the combination of blocked quartic fits and Mallows' C_p offers a simple and effective way of obtaining an initial estimate for m, many other possibilities could be used instead. An optional adjustment to the kernel functional estimates is the truncation of data within $100\alpha\%$ of the boundaries, for some small value of α. In the case where the $X_i's$ are supported on $[a, b]$, this involves replacement of $\hat{\theta}_{rs}(g)$ by

$$\hat{\theta}_{rs}^{\alpha}(g) = \frac{1}{n}\sum_{i=1}^{n}\hat{m}^{(r)}(X_i)\hat{m}^{(s)}(X_i)1_{\{(1-\alpha)a+\alpha b<X_i<\alpha a+(1-\alpha)b\}}. \tag{1.69}$$

For the nonparametric estimator of θ_{22} a bandwidth g is needed. A possible contender is the bandwidth g_{AMSE} that minimises the AMSE of the nonparametric estimator of θ_{22}. Furthermore, a prior bandwidth λ_{AMSE} has to be determined for the nonparametric estimator of σ^2. These prior bandwidths are calculated with a parametric OLS-block-fit. A possible nonparametric estimator $\hat{\theta}_{22}(g_{\text{AMSE}})$ can be defined by

$$\hat{\theta}_{22}(g) = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{m}_g^{(2)}(X_i)\right]^2. \tag{1.70}$$

The prior bandwidth g_{AMSE}, i.e., the minimiser of the conditional asymptotic mean squared error of $\hat{\theta}_{22}(g)$ is given by

$$g_{\text{AMSE}} = C_2(K)\left[\frac{\sigma^2\cdot(b-a)}{|\theta_{24}|n}\right]^{1/7}, \tag{1.71}$$

where $C_2(K)$ is a kernel dependent constant. The two unknown quantities are replaced by block-wise quartic parametric fits. For the prior estimation of σ^2 one uses the same as for the rule-of thumb bandwidth selector. An estimator for θ_{24} is based on a blocked quartic fit defined in (1.65). A nonparametric estimator for σ^2 is defined by

$$\hat{\sigma}^2(\lambda) = v^{-1}\sum_{i=1}^{n}[Y_i - \hat{m}_{\lambda_{\text{AMSE}}}(X_i)]^2, \tag{1.72}$$

where $v = -2\sum_i w_{ii} + \sum\sum_{ij} w_{ij}^2$ with $\{w_{ij}\}_{i,j=1}^n$ is the hat-matrix of $\hat{m}_{\lambda_{\text{AMSE}}}$. The use of $\hat{\sigma}(\lambda)$ in practice requires a selection of λ. It follows that the asymptotically MSE (AMSE)-optimal choice of λ is

$$\hat{\lambda}_{\text{AMSE}} = C_3(K)\left[\frac{\hat{\sigma}_Q^4(\hat{N})(b-a)}{\hat{\theta}_{22}^{.05}(\hat{g}_{\text{AMSE}})^2 n^2}\right]^{1/9}, \tag{1.73}$$

where $C_3(K)$ is a kernel dependent constant.

Algorithm 1.1. The direct plug-in bandwidth selector $\hat{h}_{DPI}$

1. Find $\hat{\theta}_{24}^Q(\hat{N})$ and $\sigma_Q^2(\hat{N})$ based on a blocked quartic fit with $\hat{N}$ chosen by Mallow's C_p and N_{max} given by (1.68) with $N^* = 5$.

2. Estimate θ_{22} using $\hat{\theta}_{22}^{.05}(\hat{g}_{\text{AMSE}})$, where

$$\hat{g}_{\text{AMSE}} = C_2(K)\left[\frac{\hat{\sigma}_Q^2(\hat{N})(b-a)}{|\hat{\theta}_{24}^Q(\hat{N})|n}\right]^{1/7}, \tag{1.74}$$

   and estimate $\hat{\sigma}^2$ using $\hat{\sigma}^2(\hat{\lambda}_{\text{AMSE}})$, where

$$\hat{\lambda}_{\text{AMSE}} = C_3(K)\left[\frac{\hat{\sigma}_Q^4(\hat{N})(b-a)}{\hat{\theta}_{22}^{.05}(\hat{g}_{\text{AMSE}})^2 n^2}\right]^{1/9}. \tag{1.75}$$

3. The selected bandwidth is

$$\hat{h}_{DPI} = C_1(K)\left[\frac{\hat{\sigma}^2(\hat{\lambda}_{\text{AMSE}})(b-a)}{\hat{\theta}_{22}^{.05}(\hat{g}_{\text{AMSE}})n}\right]^{1/5}. \tag{1.76}$$

c) Wild bootstrap method

Other popular methods for selecting bandwidths are based on the bootstrap method. The wild bootstrap has been proposed by Manteiga *et al.* (2004). Here a bootstrap bandwidth selector is considered to estimate the optimal bandwidth, in the sense of minimising the MSE of the nonparametric regression estimator,

$$\text{MSE}(x,h) = E[(\hat{m}_h(x) - m(x))^2]. \tag{1.77}$$

The bootstrap bandwidth choice comes from an estimation of MSE based on the following resampling scheme.

Algorithm 1.2. The wild bootstrap bandwidth selector

1. Suppose the sample $\mathbf{S}_n = ((X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n))$ is available.

2. Choose a pilot bandwidth g, by searching a grid, G, of possible bandwidth values, and use the sample data to construct the kernel smoother $\hat{m}_g(x)$.

3. Calculate the sample residuals as usual from the expression:

$$e_i = Y_i - \hat{m}_g(X_i),\ i = 1, \ldots, n.$$

4. Draw a bootstrap random sample $\mathbf{e}_n^* = (e_1^*, e_2^*, \ldots, e_n^*)$.

$$e_i^* = \begin{cases} e_i, & \text{with probability } \frac{1}{2} \\ -e_i, & \text{with probability } \frac{1}{2} \end{cases}$$

so that $E(e_i^*) = 0, E(e_i^*)^2 = e_i^2$ and $E(e_i^*)^3 = e_i^3$.

5. Obtain the bootstrap sample $\mathbf{S}_n^* = ((X_1, Y_1^*), (X_2, Y_2^*), \ldots, (X_n, Y_n^*))$ by using the relationship:

$$Y_i^* = \hat{m}_g(X_i) + e_i^*, \ i = 1, \ldots, n.$$

6. Calculate the bootstrap nonparametric estimator $\hat{m}_h^*(x)$ similar to $\hat{m}_g(x)$, but based on the bootstrap samples generated in the previous step. The bandwidth h is obtained by searching a grid, H, of possible bandwidth values. Finally define the bootstrap approximation of the MSE given in (1.77) as follows

$$\text{MSE}^*(x, h) = E^*[(\hat{m}_h^*(x) - \hat{m}_g(x))^2]. \tag{1.78}$$

The MSE* can be exactly calculated, since it can be written as

$$\text{MSE}^*(x, h) = (B_{h,g}^*(x))^2 + Var_{h,g}^*(x), \tag{1.79}$$

with bootstrap bias and bootstrap variance of (1.78) given, respectively, by

$$B_{h,g}^*(x) = \sum_{i=1}^n W_{hi}(x)\hat{m}_g(X_i) - \hat{m}_g(x)$$

and

$$\text{Var}_{h,g}^*(x) = \sum_{i=1}^n (W_{hi}(x))^2 (Y_i - \hat{m}_g(X_i))^2.$$

Having chosen the pilot bandwidth g, expression (1.78) only depends on h, so that minimising (1.78) gives the optimal wild bootstrap bandwidth $\hat{h}_{WB}$. It can easily be seen, however, that the necessity of choosing a pilot bandwidth, is the main disadvantage of the bootstrap method.

1.8 Selecting a kernel

The weight function $\{W_{hi}(x)\}$ of the kernel smoothers defined in Section 1.4.1 is determined by the kernel K and the bandwidth h. The accuracy of the estimated curve $\hat{m}(x)$ is dependent upon the pair (K, h). In this section the behaviour of the MSE is studied as a function of the kernel K. This section is based on Härdle (1990:133-137). It is known, in the fixed design case, from Table 1.2 that the asymptotic MSE of $m(x)$ can be written as

$$C_V c_K n^{-1} h^{-1} + C_B^2 d_K^2 h^4, \tag{1.80}$$

where C_V, C_B are constants depending on the joint distribution of (X, Y) and c_K and d_K are defined in (1.35) and (1.36) respectively. The bandwidth minimising (1.80) is of the form

$$h_0 = \left(\frac{C_V}{4C_B^2}\right)^{1/5} \left(\frac{c_K}{d_K^2}\right)^{1/5} n^{-1/5}. \tag{1.81}$$

This smoothing parameter results in the following MSE

$$\begin{aligned} \text{MSE}_{opt} &= n^{-4/5}(4C_B^2)^{1/5}C_V^{4/5}c_K^{4/5}d_K^{2/5} + n^{-4/5}(C_V/4)^{4/5}C_B^{2/5}c_K^{4/5}d_K^{2/5} \\ &= n^{-4/5}(C_V)^{4/5}C_B^{2/5}(4^{1/5} + 4^{-4/5})c_K^{4/5}d_K^{2/5}. \end{aligned}$$

This minimal MSE depends on the kernel through the factor

$$V(K)B(K) = c_K^2 d_K = \left(\int K^2(u)\,du\right)^2 \int u^2 K(u)\,du. \tag{1.82}$$

More generally, consider the case of estimating the k^{th} derivative, $m^{(k)}$, of a p-times differentiable m. The functional then takes the form

$$V(K)B(K) = \left(\int K^{(k)}(u)^2\,du\right)^{p-k} \left(\int K^{(k)}(u)u^p\,du\right)^{2k+1}. \tag{1.83}$$

Gasser *et al.* (1985) derived optimal kernels with respect to the MSE for the estimation of the v^{th} derivative of m where m is assumed to be p-times differentiable. Table 1.4 contains some of these optimal kernels.

v	p	kernel $K(u)$
0	2	$(3/4)(-u^2+1)I(\lvert u\rvert \le 1)$
0	4	$(15/32)(7u^4-10u^2+3)I(\lvert u\rvert \le 1)$
1	3	$(15/4)(u^3-u)I(\lvert u\rvert \le 1)$
1	5	$(105/32)(-9u^5+14u^3-5u)I(\lvert u\rvert \le 1)$
2	4	$(105/16)(-5u^4+6u^2-1)I(\lvert u\rvert \le 1)$
2	6	$(315/64)(77u^6-135u^4+63u^2-5)I(\lvert u\rvert \le 1)$

Table 1.4: Kernel functions minimising $V(K)B(K)$

Benedetti (1977) showed that the Epanechnikov kernel is optimal with respect to the MSE. A natural question to ask is how suboptimal are non-optimal kernels, that is by how much the expression $V(K)B(K)$ is increased for non-optimal kernels. Table 1.5 lists some commonly used kernels ($v = 0,\ p = 2$). Their deficiencies with respect to the Epanechnikov kernel are defined as

$$D(K_{opt}, K) = \frac{c_K^2 d_K}{c_{K_{opt}}^2 d_{K_{opt}}},$$

where K_{opt} denotes the optimal kernel, Epanechnikov kernel, and K is the suboptimal kernel in question. In each case the efficiency ratio does not differ significantly from one, which implies that *the choice of kernel function on the basis of the MSE is not very important. Other properties, such as computational efficiency for example, deserve more consideration than hunting for the optimal kernel. The optimal bandwidth is of greater importance.*

1.9 Kernel estimators of derivatives

This section is mainly based on the content of Härdle (1990:32). Kernel estimation techniques are handy tools which can also be used to estimate the derivatives of the regression function. Kernel derivative estimators

Kernel	$K(u)$	$D(K_{opt}, K)$
Epanechnikov	$(3/4)(-u^2+1)I(\|u\| \leq 1)$	1
Quartic	$(15/16)(1-u^2)^2 I(\|u\| \leq 1)$	1.005
Triangular	$(1-\|u\|)I(\|u\| \leq 1)$	1.011
Gauss	$(2\pi)^{-1/2}\exp(-u^2/2)$	1.041
Uniform	$(1/2)I(\|u\| \leq 1)$	1.060

Table 1.5: Some kernels and their efficiencies

are defined by differentiating the weight function sequence with respect to x. If the weights are sufficiently smooth and the bandwidth sequence is correctly tuned, then these estimators will converge to the corresponding derivatives of m. Assume the equidistant design setting and consider, for example, the Priestley-Chao weights

$$W_{hi}(x) = n(X_i - X_{i-1})K_h(x - X_i).$$

The k^{th} derivative with respect to x is

$$\hat{m}_h^{(k)}(x) = n^{-1}h^{-(k+1)} \sum_{i=1}^{n} K^{(k)} \left(\frac{x - X_i}{h} \right) Y_i. \tag{1.84}$$

The kernel estimate of the k^{th} derivative of m is therefore a local average of the response variables in which the k^{th} derivatives of the kernel weights have been used as weights. In the case of non-equally spaced and random X-variables the weight sequence becomes more complicated. According to Härdle (1990:33) the principle of differentiating the kernel weights to obtain kernel estimates for derivatives of the regression function can also be applied in the case of non-equally spaced and random X-variables. For instance, the first derivative $m'(x)$ could be estimated using the effective weight sequence

$$W_{hi}(x) = \frac{K_h^{(1)}(x - X_i)}{\hat{f}_h(x)} - \frac{K_h(x - X_i)\hat{f'}_h(x)}{(\hat{f}_h(x))^2}, \tag{1.85}$$

where

$$K_h^{(1)}(u) = h^{-2}K^{(1)}(u/h),$$

and

$$\hat{f}'(x) = n^{-1} \sum_{i=1}^{n} K_h^{(1)}(x - X_i)$$

denotes an estimate of the first derivative of the marginal density $f(x)$. The topic of derivative estimation is closely linked to the local polynomial fitting method.

1.10 Multivariate kernel regression

The discussion so far was confined to the case of univariate design points. The extension of nonparametric kernel regression to the multivariate case is discussed according to Härdle and Müller (2000). The notation of Härdle and Müller (2000) is adapted for this section. The multivariate Nadaraya-Watson estimator and the

multivariate local linear estimator are presented as well as their respective asymptotic conditional variance and bias. The extension of the cross-validation bandwidth selection method to the multivariate case is also discussed.

The aim of multivariate nonparametric regression is to estimate the functional relation between a response variable Y and a multivariate explanatory variable T, i.e., the conditional expectation

$$E(Y|T) = E(Y|T_1, \ldots, T_q) = m(T), \tag{1.86}$$

where $T = (T_1, \ldots, T_d)^T$. The relation

$$E(Y|T) = \int y f(y|t)\, dy = \frac{\int y f(y,t)\, dy}{f(x)} \tag{1.87}$$

leads by replacing the multivariate densities $f(y,t)$ by the kernel density estimate

$$\hat{f}_{h,\mathbf{H}}(y,t) = \frac{1}{n}\sum_{i=1}^{n} K_h(Y_i - y)\mathcal{K}_{\mathbf{H}}(t_i - t), \tag{1.88}$$

where $\mathbf{H}$ is a nonsingular, symmetric bandwidth matrix. The multivariate Nadaraya-Watson estimator is then defined as

$$\hat{m}_{\mathbf{H}}(t) = \frac{\sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(T_i - t)Y_i}{\sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(T_i - t)}. \tag{1.89}$$

Hence, the multivariate kernel regression estimator is just a weighted sum of the observed responses Y_i. The denominator ensures that the weights sum up to one. Depending on the choice of the kernel, $\hat{m}_{\mathbf{H}}(t)$ is a weighted average of those Y_i where T_i lies in a ball or cube around t.

The multivariate Nadaraya-Watson estimator is a local constant estimator, i.e., the solution of

$$\min_{\beta_0} \sum_{i=1}^{n} \{Y_i - \beta_0\}^2 \mathcal{K}_{\mathbf{H}}(T_i - t). \tag{1.90}$$

Replacing β_0 by a polynomial in $T_i - t$ yields a local polynomial kernel regression estimator. This definition of local polynomial kernel regression is a straightforward generalisation of the univariate case. Details of this process can be found in Ruppert and Wand (1994). This can be illustrated with the example of a local linear regression estimate. The minimisation problem is then

$$\min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \{Y_i - \beta_0 - (T_i - t)^T \beta_1\}^2 \mathcal{K}_{\mathbf{H}}(T_i - t). \tag{1.91}$$

The solution of the problem can hence equivalently be written as

$$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1^T)^T = (\mathbf{T}^T\mathbf{W}\mathbf{T})^{-1}\mathbf{T}^T\mathbf{W}\mathbf{Y} \tag{1.92}$$

using the notations

$$\mathbf{T} = \begin{pmatrix} 1 & (T_1 - t)^T \\ \vdots & \vdots \\ 1 & (T_n - t)^T \end{pmatrix},$$

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

and $\mathbf{W} = \text{diag}(\mathcal{K}_{\mathbf{H}}(T_1 - t), \ldots, \mathcal{K}_{\mathbf{H}}(T_n - t))$. In (1.92) $\hat{\beta}_0$ estimates the regression function itself, whereas $\hat{\beta}_1$ estimates the partial derivatives with respect to the components T. For the following section the multivariate local linear estimator is denoted as

$$\hat{m}_{1,\mathbf{H}}(t) = \hat{\beta}_0(t). \tag{1.93}$$

1.10.1 Bias, variance and asymptotics

The conditional asymptotic bias and variance of the multivariate Nadaraya-Watson kernel regression estimator are

$$E\{\hat{m}_{\mathbf{H}}|T_1, \ldots, T_n\} - m(t) \approx \mu_2(\mathcal{K}) \frac{\nabla_m(t)^T \mathbf{H}\mathbf{H}^T \nabla_f(t)}{f(t)} + \frac{1}{2}\mu_2(\mathcal{K}) tr\{\mathbf{H}^T \mathcal{H}_m(t)\mathbf{H}\} \tag{1.94}$$

$$\text{Var}\{\hat{m}_{\mathbf{H}}|T_1, \ldots, T_n\} \approx \frac{1}{n \det(\mathbf{H})} ||\mathcal{K}||_2^2 \frac{\sigma^2(t)}{f(t)}, \tag{1.95}$$

in the interior of the support of f_T. Note that ∇ refers to the first derivative of the function.

The conditional asymptotic bias and variance of the multivariate local linear regression estimator are

$$E\{\hat{m}_{1,\mathbf{H}}|T_1, \ldots, T_n\} - m(t) \approx \frac{1}{2}\mu_2(\mathcal{K}) tr\{\mathbf{H}^T \mathcal{H}_m(t)\mathbf{H}\} \tag{1.96}$$

$$\text{Var}\{\hat{m}_{1,\mathbf{H}}|T_1, \ldots, T_n\} \approx \frac{1}{n \det(\mathbf{H})} ||\mathcal{K}||_2^2 \frac{\sigma^2(t)}{f(t)}, \tag{1.97}$$

in the interior of the support of f_T.

1.10.2 Bandwidth selection and practical aspects

Principally, the methods to choose a smoothing parameter in multivariate nonparametric regression are the same as in the case of univariate nonparametric regression. Again, plug-in and resampling ideas are employed for finding a global bandwidth $\mathbf{H}$ or a local bandwidth $\mathbf{H}(t)$. For the multivariate case only the cross-validation bandwidth selection method is discussed.

In the multivariate case the cross-validation function is defined as

$$CV(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\{Y_i - \hat{m}_{\mathbf{H},-i}(X_i)\}^2. \tag{1.98}$$

The computation of $CV(\mathbf{H})$ does not need more computational effort than the computation of $m_{\mathbf{H}}(X_1), \ldots, m_{\mathbf{H}}(X_n)$. However the optimisation over a matrix $\mathbf{H}$ might be more difficult, hence diagonal bandwidth matrices, $\mathbf{H} = h\mathbf{I}_q$, where $\mathbf{I}_q$ denotes the q x q identity matrix, with appropriate standardisation of the data, are still preferred in practice.

The general problem in multivariate nonparametric estimation is the *curse of dimensionality*. Recall that the nonparametric regression estimators are based on the idea of local, weighted averaging. In higher dimensions the observations are usually sparsely distributed for reasonable sample sizes, and consequently estimators based on local averaging perform unsatisfactorily in this situation. One can explain this effect by looking at the MISE. Consider a multivariate regression estimator with the same bandwidth h for all components, e.g. a Nadaraya-Watson or local linear estimator with bandwidth matrix $\mathbf{H} = h\mathbf{I}_q$. Here the asymptotic MISE depends on q:

$$\text{AMISE}(n, h) = \frac{1}{nh^q} C_1 + h^4 C_2, \tag{1.99}$$

where C_1 and C_2 are constants that neither depend on n nor h. If we derive the optimal bandwidth we find that $h_{opt} \sim n^{-1/(4+q)}$ and hence the rate of convergence for AMISE is $n^{-4/(4+q)}$. One can clearly see that the speed of convergence decreases dramatically for higher dimensions q.

In this chapter a brief overview from the literature of the main aspects of nonparametric regression analysis was given. In the next chapter the fundamental ideas of the bootstrap methodology, which underlie the bagging and bragging improvement methods, are presented.

Chapter 2

The bootstrap

2.1 Introduction

The aim of Chapter 2 is to provide the reader with basic bootstrap knowledge needed for the application of the new methods stated in Chapters 3 and 4. The new nonparametric regression estimators are bootstrap intensive. Therefore a thorough discussion of the basic definitions and ideas surrounding the application of the bootstrap in the regression context is needed. To motivate the bootstrap procedure, the following introductory ideas may be of use.

Suppose a sample $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$, is drawn independently from an unknown probability distribution F. This sample is to be used to make inferences about a population characteristic, denoted by θ, using a statistic T (Davison & Hinkley, 1997:11). One would like to determine various properties related to the distribution of the statistic, such as the bias and the standard error. These values give an indication of the accuracy of the estimate. One way to derive these discrepancy measures is to make some assumptions about the underlying distribution of the population. These assumptions will greatly impact the results and the conclusions drawn. In some cases one would like to carry out statistical inference in problems without deriving analytic solutions. The bootstrap is a resampling method that does not make any assumptions about the underlying distribution and can be applied in such cases to almost any statistic. According to Efron and Tibshirani (1986:54) the bootstrap is a computer-based method, which substitutes considerable amounts of computation in place of theoretical analysis. The bootstrap can routinely answer questions which are far too complicated for traditional statistical analysis. Boos (2003:168) describes the bootstrap in terms of the "bootstrap world" which is created by the statistician and where the statistician knows everything. In this parallel world the true sampling design of the data is reproduced as closely as possible and unknown aspects of the statistical models are replaced by sample estimates. In this world, the data analyst can determine any quantity of interest by simulation. For example, if a discrepancy measure such as the variance of a complicated parameter estimate in this world is desired, generate B bootstrap samples by using computer methods, compute the estimate for each bootstrap sample and then use the sample variance of the B estimates as an approximation of the variance of the parameter estimate. As B grows large,

this sample variance converges to the true variance in the bootstrap world. In terms of the estimator based on the original data, this limiting sample variance is just an estimator of the true variance of the estimate. Therefore a bootstrap world is created where anything can be computed, at least up to Monte Carlo error. Those true quantities calculated in the bootstrap world are estimates of the parallel quantities in the real world (Boos, 2003:169).

In this chapter a brief effort is made to introduce some key aspects of the bootstrap method as it is applied in this study. In Section 2.2 the bootstrap methodology is described with regard to the plug-in principle and choices of $\hat{F}$ in particular. In Section 2.3 two examples of applications of the bootstrap are considered, namely the bootstrap estimation of the standard error (Section 2.3.1) and the bootstrap estimation of the bias (Section 2.3.2). The reliability of such a bootstrap estimate is also of interest. The consistency of a bootstrap estimate can be assessed by bootstrapping the bootstrap. The double bootstrap is discussed in Section 2.3.3. In the last section the bootstrap method is illustrated as it is applied to regression analysis in the literature. Efron and Tibshirani (1993) and Davison and Hinkley (1997) form the main sources for the following discussion.

2.2 The bootstrap procedure

Let F denote the probability distribution of a random variable X. Suppose a random sample of size n is observed from this unknown probability distribution F, $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$. As said before, the idea of the bootstrap is to shift from an environment with unknown parameters to an environment where everything can be estimated by resampling from the observed sample. This environment is referred to as "the bootstrap world". One moves from the real world to the bootstrap world by applying the plug-in principle. The plug-in principle is discussed in Section 2.2.1. The plug-in principle requires some approximation of the unknown distribution function F. Choices of $\hat{F}$ is discussed in Section 2.2.2.

2.2.1 The plug-in principle

The following discussion of the plug-in principle is based on Efron and Tibshirani (1993:35). The plug-in principle is a simple method of estimating parameters from samples. The parameter of interest is a function of the probability distribution F. Let θ denote the parameter of interest. The parameter can be written directly as a function of F:

$$\theta = t(F). \tag{2.1}$$

This notation implies that the value of θ is obtained by applying some functional $t(\cdot)$ to the distribution function F. Let $\hat{\theta} = \hat{\theta}(X_1, X_2, \ldots, X_n)$ denote an estimator of the parameter θ based on a sample of size n. The plug-in estimate of the parameter $\theta = t(F)$ is defined as:

$$\hat{\theta} = t(\hat{F}). \tag{2.2}$$

The plug-in principle requires plugging in some approximation of the unknown distribution function F into the functional. The bootstrap method is a direct application of the plug-in principle. Figure 2.1 demonstrates the plug-in principle in the case of the bootstrap (Efron, 2003:136). One travels from the real world to the bootstrap

world simply by plugging in a point estimate $\hat{F}$ for F. All arrows on the right are exact analogues of those on the left (Efron, 2003:136).

$$\begin{array}{ccc} \text{Real world} & & \text{Bootstrap world} \\ F \to \mathbf{X}_n & \Longrightarrow & \hat{F} \to \mathbf{X}_n^* \\ \downarrow & & \downarrow \\ \hat{\theta} & & \hat{\theta}^* \end{array}$$

Figure 2.1: Bootstrap diagram

The plug-in principle can then be repeated by using $\hat{F}$ as an estimator of F and defining the bootstrap estimator as:

$$\hat{\theta}^* = t(\hat{F}). \tag{2.3}$$

This implies applying the functional $t(\cdot)$ to the estimate $\hat{F}$. To create a bootstrap sample, $\mathbf{X}_n^* = (X_1^*, \ldots, X_n^*)$ one draws a random sample from $\hat{F}$. The following general bootstrap algorithm explains the bootstrap procedure.

Algorithm 2.0. The basic bootstrap procedure

1. Draw a bootstrap random sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ from $\hat{F}$.

2. Repeat step 1 B times, where B denotes the number of bootstrap samples.

$$\begin{array}{ll} X_1^*, X_2^*, \ldots, X_n^* & (1) \\ X_1^*, X_2^*, \ldots, X_n^* & (2) \\ & \vdots \\ X_1^*, X_2^*, \ldots, X_n^* & (B) \end{array}$$

3. For each of these B bootstrap samples the bootstrap statistic $\hat{\theta}_i^* = \hat{\theta}\,(X_1^*, X_2^*, \ldots, X_n^*)$ is calculated, $i = 1, 2, \ldots, B$.

Once all B bootstrap replicates $\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*$ have been calculated, the distribution of the statistic $\hat{\theta}^*$ can be estimated. As $B \to \infty$ the accuracy of the distribution approximation of $\hat{\theta}^*$ increases. This distribution of $\hat{\theta}^*$ therefore approximates the distribution of the statistic $\hat{\theta}$.

2.2.2 Choices of $\hat{F}$

An important concept in the bootstrap methodology is that the distribution of the statistic of interest, say G, can be approximated by the distribution of repeated estimates from replicated samples from an approximation to the unknown population distribution F. The population distribution can be calculated in different ways, each of which leads to a different type of bootstrap method, such as the parametric and nonparametric bootstrap methods. A less frequently used bootstrap method is the smoothed bootstrap, originating from kernel distribution functions, i.e. where F_h is an estimate of F. In the case of the nonparametric bootstrap the empirical

distribution function F_n is used to estimate F. The parametric bootstrap is applied when one can assume that the sample data were generated from some well-known parametric distribution $F_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ denotes a vector of the distribution's parameters, which is unknown. More detailed information regarding the choices of $\hat{F}$ follows.

a) The empirical distribution function, F_n

The empirical distribution function, EDF, denoted by F_n, is a discrete approximation of the distribution function F, constructed entirely from the observed sample, or empirical data. Let $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ be a random sample from a continuous distribution function F. The EDF is defined as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I\left(X_i \leq x\right), \tag{2.4}$$

where I is an indicator function defined as:

$$I(A) = \begin{cases} 1, & \text{if the event A occurs.} \\ 0, & \text{if the event A does not occur.} \end{cases}$$

F_n assigns an equal probability n^{-1} to each of the observed values $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ and is a step function. The properties of F_n can be found in Rice (2007:380) and Jacod and Protter (2004:185).

Remarks:

1. $E[F_n(x)]$=$F(x)$, that is F_n is an *unbiased* estimator of F.
2. $\text{Var}[F_n(x)]$=$\frac{1}{n}$ $[F(x)(1 - F(x))]$.
3. The *Glivenko-Cantelli* Lemma states that:

$$P\left(\limsup_{n\to\infty} |F_n(x) - F(x)| = 0\right) = 1.$$

F_n converges *almost-surely (a.s)* to F as n tends to infinity.

If $\hat{F} = F_n$, the bootstrap sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ is a random sample of size n drawn with replacement from the original sample. Each data point in the original sample has probability n^{-1} to be drawn as the next data point in the bootstrap sample. Since F_n only depends on the data and not on any distribution assumptions, this nonparametric bootstrap method is used in the simulation studies.

b) A known parametric distribution, $F_{\boldsymbol{\theta}}$

The parametric bootstrap is used when it is assumed that the sample data were generated from some well-known parametric distribution $F_{\boldsymbol{\theta}}$, where $\boldsymbol{\theta}$ denotes a vector of the distribution's parameters, which is unknown. The unknown parameters can be estimated by using the sample data. The estimated distribution $F_{\hat{\boldsymbol{\theta}}}$ is then obtained. The next step is to generate the bootstrap sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ as follows: instead of sampling with replacement from the observed data, one draws B samples of size n from the *assumed parametric distribution* and proceeds as described before in step 3 of Algorithm 2.0 to calculate bootstrap statistics $\hat{\theta}_i^*,\ i = 1, 2, \ldots, B$,

of the B generated samples, from which the distribution of $\hat{\theta}^*$ is then determined. This defines the parametric bootstrap procedure. The parametric bootstrap is useful in problems where some knowledge about the form of the underlying population is available. However, a main reason for making parametric assumptions in traditional statistical analysis is to enable the use of textbook formulas for discrepancy measures such as the standard error of a statistic (Efron & Tibshirani, 1993:56). Since the aim is to try to avoid restrictive formulas in the bootstrap approach, it is preferred to minimise parametric assumptions.

c) The kernel distribution function, F_h

Since the empirical distribution F_n is a discrete distribution, samples constructed from F_n in the bootstrap simulations, have some peculiar properties. For example, since all the elements of the bootstrap samples are drawn from the original sample values, nearly every sample contains repeated values. The *smoothed bootstrap* is a modification of the bootstrap procedure, designed to avoid samples with these properties (Silverman & Young, 1987:470). When it is reasonable to assume that F has a continuous probability distribution function, one possibility is to use kernel density estimation (Davison & Hinkley, 1997:79).

If $f(y)$ denotes the density function of a continuous random variable X, then the distribution function of a continuous random variable X in the point x is defined by:

$$F(x) = \int_{-\infty}^{x} f(y)\, dy. \tag{2.5}$$

Define the kernel density estimator of f by:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \tag{2.6}$$

The kernel distribution function estimator F_h is defined by substituting the kernel density estimator $\hat{f}_h$ in equation (2.5). The result is the following:

$$\begin{aligned} F_h(x) &= \int_{-\infty}^{x} \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{t - X_i}{h}\right) dt \\ &= \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \end{aligned} \tag{2.7}$$

Bootstrap samples can now be generated by sampling independently from F_h. The following expression shows a way in which bootstrap samples are drawn from the kernel distribution function estimator F_h:

$$Y_i^* = X_i^* + hZ_i. \tag{2.8}$$

Remarks:

1. In (2.8) $X_1^*, X_2^*, \ldots, X_n^*$ is a bootstrap sample created by sampling with replacement from $X_1, X_2, \ldots, X_n$, in other words created from F_n.

2. The value Z_i denotes a random value drawn from the kernel function K, and is drawn independently of X_i^*.

3. h denotes the smoothing parameter used in the kernel distribution function estimator, F_h.

It can be shown that (2.8) is equivalent to sampling independently from F_h. The expression (2.8) is useful in that one now has a smooth, continuous function from which one can generate bootstrap samples. This method is referred to as the *smoothed bootstrap*. However, the limitations of (2.8) become clear when the variance term of the different bootstrap observations is studied:

$$\begin{aligned} \text{Var}_{F_h}(Y^*) &= \text{Var}_{F_h}(X^* + hZ) \\ &= \text{Var}_{F_n}(X^*) + h^2\text{Var}(Z) \\ &= S_n^2 + h^2\text{Var}(Z). \end{aligned} \tag{2.9}$$

Unfortunately, the variances calculated using these smoothed bootstrap samples are not the same as the true variances. Smooth bootstrap observations obtained by using (2.8) lead to bootstrap samples with inflated variance. It is desirable for the samples to have the same variances as for the unsmoothed bootstrap (Davison & Hinkley, 1997:79). This is implemented via the *shrunk smoothed bootstrap* (Davison & Hinkley, 1997:79). The following sampling scheme has the same expected value and variance as the naive bootstrap procedure, therefore resulting in a better approximation of F.

$$Y_i^* = \bar{X}_n + \left(X_i^* - \bar{X}_n + hZ_i\right)\left(1 + \frac{\sigma_k^2 h^2}{S_n^2}\right)^{-\frac{1}{2}}. \tag{2.10}$$

Remarks:

1. S_n^2 above denotes the sample variance defined by $S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}_n\right)$.

2. σ_k^2 denotes the variance of the kernel function K.

2.3 Bootstrap estimation of the standard error and bias

In this section the bootstrap methodology is illustrated by presenting the bootstrap algorithms to estimate a statistic's standard error and bias. Suppose a sample $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ is drawn independently from an unknown probability distribution F. Let θ denote the parameter of interest and let $\hat{\theta} = \hat{\theta}(X_1, X_2, \ldots, X_n)$ denote an estimator of the parameter θ based on a sample of size n. The aim of the bootstrap is to estimate the distributional properties of the estimator $\hat{\theta}$.

2.3.1 Bootstrap estimation of the standard error

Efron and Tibshirani (1993:45-46) describe the bootstrap estimation of the standard error as follows: Let the standard error of the estimator $\hat{\theta} = \hat{\theta}(X_1, X_2, \ldots, X_n)$ be denoted by:

$$\text{se}_F(\hat{\theta}) = \sqrt{\text{Var}_F(\hat{\theta})}. \tag{2.11}$$

Now, by applying the plug-in principle, the bootstrap estimate of the standard error of $\hat{\theta}$ is given by:

$$\text{se}_{\hat{F}}(\hat{\theta}^*) = \sqrt{\text{Var}_{\hat{F}}(\hat{\theta}^*)}. \tag{2.12}$$

Equation (2.12) is referred to as the *ideal bootstrap estimate of the standard error* and can be calculated explicitly in some cases. In the cases where the expression cannot be determined explicitly, one has to approximate it by using Monte Carlo simulations based on a large number of bootstrap samples, i.e. by B independent identically distributed replications. The approximated bootstrap standard error of $\hat{\theta}$ is denoted by $\widehat{\text{se}}_B$ and is determined from the expression:

$$\widehat{\text{se}}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)\right)^2}, \tag{2.13}$$

where

$$\hat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b).$$

To obtain the ideal bootstrap estimate of the standard error, one needs $B \to \infty$. But only a finite number of bootstrap samples can be generated, and therefore the simulated standard error is merely an estimate of the ideal standard error. It can be shown that $\widehat{\text{se}}_B \to \text{se}_{\hat{F}}$ as $B \to \infty$. The following algorithm presented by Efron and Tibshirani (1993:47) describes a Monte Carlo algorithm to obtain an approximation of the ideal bootstrap estimate of the standard error.

Algorithm 2.1. Bootstrap estimate of the standard error

1. Suppose the sample $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ has been observed. Then construct the EDF F_n.
2. Draw a bootstrap random sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ from the empirical distribution function, F_n. In other words sample with replacement from $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ with equal probability n^{-1}.
3. Calculate the bootstrap replication of the statistic $\hat{\theta}$, $\hat{\theta}^*$=$\hat{\theta}(X_1^*, X_2^*, \ldots, X_n^*)$, by using the bootstrap random sample generated in step 2.
4. Independently repeat steps 2 and 3 B times. The following bootstrap replications are obtained: $\hat{\theta}^*(1)$, $\hat{\theta}^*(2)$,..., $\hat{\theta}^*(B)$.
5. Calculate the bootstrap estimation of the standard error:

$$\widehat{\text{se}}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)\right)^2}, \tag{2.14}$$

where

$$\hat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b).$$

2.3.2 Bootstrap estimation of the bias

Efron and Tibshirani (1993:124) describe the bootstrap estimation of the bias as follows: Bias can be seen as an indication of the accuracy of an estimator. Bias is a measure of the difference between the expected value of an estimator ($\hat{\theta}$) and the parameter (θ) it is attempting to estimate. Therefore the bias of the estimator $\hat{\theta} = \hat{\theta}(X_1, X_2, \ldots, X_n)$ is defined by

$$\text{bias}_F = E_F(\hat{\theta}) - \theta. \tag{2.15}$$

In order to obtain the ideal bootstrap estimate of the bias given in (2.15) the plug-in principle needs to be applied. The ideal bootstrap estimate of the bias is then:

$$\text{bias}_{\hat{F}} = E_{\hat{F}}(\hat{\theta}^*) - \hat{\theta}. \tag{2.16}$$

Equation (2.16) is the *ideal bootstrap estimate of the bias* and in some cases can be calculated explicitly. But in most cases the ideal bootstrap estimate of the bias needs to be approximated by simulation. This value needs to be estimated by making use of a Monte Carlo algorithm. The following algorithm approximates the true bootstrap estimate of the bias:

Algorithm 2.2. Bootstrap estimate of the bias

1. As in Algorithm 2.1 suppose the sample $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ is observed. Then construct the EDF F_n.
2. Draw a bootstrap random sample $\mathbf{X}_n^*$=(X_1^*, X_2^*,..., X_n^*) from the empirical distribution function, F_n. In other words sample with replacement from $\mathbf{X}_n$=(X_1, X_2,..., X_n) with equal probability n^{-1}.
3. Calculate the bootstrap replication of the statistic $\hat{\theta}$, $\hat{\theta}^*$=$\hat{\theta}$(X_1^*, X_2^*,..., X_n^*), by using the bootstrap random sample generated in step 2.
4. Independently repeat steps 2 and 3 B times. The following bootstrap replications are obtained: $\hat{\theta}^*(1)$, $\hat{\theta}^*(2)$,..., $\hat{\theta}^*(B)$.
5. Calculate the bootstrap estimation of the bias:

$$\widehat{\text{bias}}_B = \hat{\theta}^*(\cdot) - \hat{\theta}, \tag{2.17}$$

where

$$\hat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^*(b).$$

To obtain the ideal bootstrap estimate of the bias, one needs $B \to \infty$. But only a finite number of bootstrap samples can be generated, and therefore the simulated bias is just an estimate of the ideal bias. It can be shown that $\widehat{\text{bias}}_B \to \text{bias}_{\hat{F}}$ as $B \to \infty$.

2.3.3 The double bootstrap

When calculating bootstrap approximations of the distributional properties of an estimator $\hat{\theta}$ the question arises of how accurate and precise these bootstrap estimates are. The reliability of a bootstrap algorithm can be assessed by bootstrapping the bootstrap (Hinkley, 1994:403). To illustrate this idea an algorithm for the estimation of the *standard error of the bootstrap standard error* is discussed.

Let the standard error of the estimator $\hat{\theta} = \hat{\theta}(X_1, X_2, \ldots, X_n)$ be defined by:

$$\text{se}_F(\hat{\theta}) = \sqrt{\text{Var}_F(\hat{\theta})}.$$

Now, by applying the plug-in principle, the ideal bootstrap estimate of the standard error of $\hat{\theta}$ is given by:

$$\mathrm{se}_{\hat{F}}(\hat{\theta}^*) = \sqrt{\mathrm{Var}_{\hat{F}}(\hat{\theta}^*)}.$$

Then the ideal bootstrap estimate of the standard error of $\mathrm{se}_{\hat{F}}(\hat{\theta}^*)$ is given by

$$\mathrm{se}_{\hat{F}}(\mathrm{se}_{\hat{F}^*}(\hat{\theta}^{**})) = \sqrt{\mathrm{Var}_{\hat{F}}(\sqrt{\mathrm{Var}_{\hat{F}^*}(\hat{\theta}^{**})})}, \tag{2.18}$$

where $\hat{F}^*$ is defined in step 2(a) of Algorithm 2.3 below. Here equation (2.18) denotes the bootstrap estimate of the standard error based on the second phase bootstrap sample $\mathbf{X}_n^{**} = (X_1^{**}, X_2^{**}, \ldots, X_n^{**})$, which is drawn with replacement from a first phase bootstrap sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ and which is used to determine $\hat{\theta}^{**} = \hat{\theta}(X_1^{**}, X_2^{**}, \ldots, X_n^{**})$. The Monte Carlo algorithm for this double bootstrap is now presented.

Algorithm 2.3. The double bootstrap

1. Having observed $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$, construct the EDF F_n.

2. Draw a bootstrap random sample $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ from F_n. In other words, sample with replacement from $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$ with equal probability n^{-1}. For the rest of step 2, consider the bootstrap sample $\mathbf{X}_n^*$ as the original sample.

   (a) Having observed $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$, construct the bootstrap EDF F_n^*.

   (b) Draw a double bootstrap random sample $\mathbf{X}_n^{**} = (X_1^{**}, X_2^{**}, \ldots, X_n^{**})$ from F_n^*. In other words, sample with replacement from $\mathbf{X}_n^* = (X_1^*, X_2^*, \ldots, X_n^*)$ with equal probability n^{-1}.

   (c) Calculate the first double bootstrap replication of the statistic $\hat{\theta}^*$, i.e. $\hat{\theta}_{(1)}^{**} = \hat{\theta}(X_1^{**}, X_2^{**}, \ldots, X_n^{**})$, by using the double bootstrap sample.

   (d) Independently repeat steps (b) to (c) R times, obtaining double bootstrap replications, $\hat{\theta}^{**}(1), \hat{\theta}^{**}(2), \ldots, \hat{\theta}^{**}(R)$.

   (e) Calculate the first double bootstrap approximated standard error by

$$\hat{\mathrm{se}}_R^*(1) = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(\hat{\theta}^{**}(r) - \hat{\theta}^{**}(\cdot)\right)^2},$$

   where

$$\hat{\theta}^{**}(\cdot) = \frac{1}{R}\sum_{r=1}^{R}\hat{\theta}^{**}(r).$$

3. Independently repeat step 2 B times, obtaining double bootstrap standard error approximations $\widehat{\mathrm{se}}_R^*(1), \widehat{\mathrm{se}}_R^*(2), \ldots, \widehat{\mathrm{se}}_R^*(B)$.

4. Calculate in a similar way as in (2.14)

$$\widehat{\mathrm{se}}_B = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\mathrm{se}}_R^*(b) - \hat{\mathrm{se}}_R^*(\cdot)\right)^2},$$

   where

$$\hat{\mathrm{se}}_R^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\mathrm{se}}_R^*(b).$$

2.4 Bootstrap applied to regression analysis

In regression analysis a regression curve describes a general relationship between a vector of explanatory variables $\mathbf{X}$ and a response variable Y (Härdle, 1990:3). Suppose n data points $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ have been collected. The regression relationship can be formulated as:

$$Y_i = m(\mathbf{X}_i) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{2.19}$$

where

$$m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$$

is the unknown regression function and the ε_i terms are the independent random variables with mean 0 and variance σ^2, usually referred to as errors. The aim is to estimate the unknown regression function $m(\mathbf{x})$, using the available data. The task of approximating this unknown regression function m can be approached in two ways: parametrically or nonparametrically.

In the parametric approach it is assumed that the mean curve has some prespecified functional form (Härdle, 1990:4). A parametric regression model assumes that the functional form of m is fully described by a finite set of parameters. More specifically it is assumed that $m(\mathbf{x}) = m(\mathbf{x}, \boldsymbol{\beta})$, with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p),\ 0 < p < \infty$. By applying an appropriate estimation methodology such as least squares, estimators $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ can be determined by utilising the data, i.e.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} [Y_i - m(\mathbf{X}_i; \boldsymbol{\beta})]^2.$$

Thereby m can be estimated. The least squares theory of regression estimation and inference provides clean, exact methods of analysis if one has a regression model with normal random error terms and constant variance (Davison & Hinkley, 1997:256). In the absence of normality and constant variance, exact methods seldom exist. The bootstrap can be used to get insight into the sampling distribution of $\hat{\boldsymbol{\beta}}$.

Utilising the bootstrap in regression analysis can be done in one of two ways: by bootstrapping the residuals or by bootstrapping data pairs, which are discussed in Sections 2.4.1 and 2.4.2. In the case of non-constant variance the wild bootstrap is used, which is discussed in Section 2.4.3. These methods are illustrated by applying it to the estimated parametric model $m(X_i, \hat{\boldsymbol{\beta}})$ where the distributional properties of $\hat{\boldsymbol{\beta}}$ are of interest. In this chapter the focus is on the standard errors of $\hat{\boldsymbol{\beta}}$.

Nonparametric regression methods contain a collection of techniques for estimating a regression curve without making strong assumptions about the shape of the regression function (Altman, 1992:175). There are various methods to obtain a nonparametric regression estimate of m. Consider a univariate predictor variable. In equation (2.19) Y_i can be considered as the sum of the value of the regression function at X_i and some error ε_i, where the expected value of the error is zero. According to Györfi *et al.* (2002:18) this motivates the construction of estimates by local averaging, i.e. estimation of $m(x)$ by the average of those Y_i where X_i is close

to x. Such an estimate can be written as

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ni}(x)Y_i, \tag{2.20}$$

where $\{W_{ni}(x)\}_{i=1}^{n}$ denotes a sequence of weights, usually depending on a smoothing parameter. This weight sequence is tuned by the smoothing parameter which regulates the size of the neighbourhood around x (Härdle, 1990:24). The choice of the smoothing parameter plays an important role in the performance of the nonparametric kernel regression estimators. Bootstrap methods can be used to improve the choice of the smoothing parameter. These bootstrap-based improvement methods are discussed in Chapter 3.

2.4.1 Bootstrapping residuals

The discussion on the method of bootstrapping the residuals is based on Davison and Hinkley (1997:260). Suppose that the errors in (2.19) have an underlying distribution G. The idea of the bootstrap residuals approach is to calculate the sample residuals:

$$\hat{\varepsilon}_i = Y_i - m(\mathbf{X}_i, \hat{\boldsymbol{\beta}}),\ i = 1, \ldots, n, \tag{2.21}$$

where $\hat{\boldsymbol{\beta}}$ was determined by some estimation technique, for example least squares, *to center these residuals and then to construct the empirical distribution G_n of the centred residuals. Bootstrap residuals are obtained by sampling with replacement from the centred residuals.* These resampled residuals are used to construct the bootstrap sample from which the bootstrap estimates of the regression coefficients $\hat{\boldsymbol{\beta}}^*$ are obtained. The distributions of these bootstrap estimates of the regression coefficients serve as approximations of the sampling distributions of the estimated regression coefficients $\hat{\boldsymbol{\beta}}$. Davison and Hinkley (1997:262) present a Monte Carlo algorithm that describes the process of using bootstrapping residuals to estimate the standard errors of $\hat{\boldsymbol{\beta}}$.

Algorithm 2.4. Bootstrapping residuals

1. Suppose the sample $\mathbf{S}_n = ((\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n))$ is observed.

2. Use the sample data to calculate the vector of estimated regression parameters $\boldsymbol{\beta}$, with some estimation technique, say least squares:
$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[Y_i - m\left(\mathbf{X}_i, \boldsymbol{\beta}\right)\right]^2.$$

3. Calculate the sample residuals from the expression:
$$e_i = Y_i - m(\mathbf{X}_i, \hat{\boldsymbol{\beta}}),\quad i = 1, \ldots, n.$$

4. Center the residuals calculated in step 3.
$$\hat{e}_i = e_i - \bar{e},\quad i = 1, \ldots, n,$$
where $\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i$ to obtain $\hat{\mathbf{e}}_n = (\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n)$ and construct the EDF G_n of the centred residuals.

5. Draw a bootstrap random sample $\hat{\mathbf{e}}_n^* = (\hat{e}_1^*, \hat{e}_2^*, \ldots, \hat{e}_n^*)$ from G_n. In other words, sample with replacement from $\hat{\mathbf{e}}_n = (\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n)$ with equal probability n^{-1}.

6. Obtain the bootstrap sample $\mathbf{S}_n^* = ((\mathbf{X}_1, Y_1^*), (\mathbf{X}_2, Y_2^*), \ldots, (\mathbf{X}_n, Y_n^*))$ by using the relationship:

$$Y_i^* = m\left(\mathbf{X}_i, \hat{\boldsymbol{\beta}}\right) + \hat{e}_i^*, \quad i = 1, \ldots, n.$$

7. Calculate the bootstrap estimate of the coefficient vector $\hat{\boldsymbol{\beta}}$ by using the bootstrap random sample $\mathbf{S}_n^*$. In the case of the least squares estimation this implies

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} \left[Y_i^* - m\left(\mathbf{X}_i, \hat{\boldsymbol{\beta}}\right)\right]^2.$$

8. Independently repeat steps 5 to 7 B times, obtaining bootstrap replications $\hat{\boldsymbol{\beta}}^*(1)$, $\hat{\boldsymbol{\beta}}^*(2)$,..., $\hat{\boldsymbol{\beta}}^*(B)$.

9. Calculate the bootstrap estimation of the standard error of the j^{th} estimated regression coefficient $\hat{\beta}_j$ as

$$\hat{\sigma}_{j,B} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}_j^*(b) - \hat{\beta}_j^*(\cdot)\right)^2}, \quad j = 0, 1, \ldots, p,$$

where

$$\hat{\beta}_j^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_j^*(b).$$

Note that $\hat{\sigma}_{j,B} \to \hat{\sigma}_j$ as $B \to \infty$ where $\hat{\sigma}_j$ denotes the ideal bootstrap estimate of the standard error of the j^{th} estimated regression coefficient.

2.4.2 Bootstrapping pairs

The resampling of residuals in the previous section has the disadvantage that it relies greatly on the assumption that the error terms ε_i have equal variances. A different approach is to assume the data are sampled from some bivariate distribution F of $(\mathbf{X}, \mathbf{Y})$. The regression coefficients are viewed as statistical functions of F. The model in (2.19) still holds, the only restriction on the random errors ε_i is that they are independent. Since this method does not assume equal error variance it can be used in the case of heteroskedasticity. In the case of constant variance the bootstrapping of the residuals will be more effective than bootstrapping pairs. (Davison & Hinkley, 1997:264).

Let F denote the bivariate distribution of $(\mathbf{X}, \mathbf{Y})$. Resampling will be done from the EDF of the data pairs, F_n. This involves resampling with replacement from $\mathbf{S}_n = ((\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), ..., (\mathbf{X}_n, Y_n))$. The bootstrap sample is then denoted by $\mathbf{S}_n^* = ((\mathbf{X}_1^*, Y_1^*), (\mathbf{X}_2^*, Y_2^*), ..., (\mathbf{X}_n^*, Y_n^*))$. The bootstrap replicates $\hat{\boldsymbol{\beta}}^*$ of the coefficient estimates are computed from $\mathbf{S}_n^*$ by using the same least squares algorithm used to obtain $\hat{\boldsymbol{\beta}}$ from $\mathbf{S}_n$. The Monte Carlo algorithm that describes the process of using bootstrapping pairs to estimate the standard errors of $\hat{\boldsymbol{\beta}}$ is as follows:

Algorithm 2.5. Bootstrapping pairs

1. Suppose the sample $\mathbf{S}_n = ((\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots (\mathbf{X}_n, Y_n))$ is observed.

2. Draw a bootstrap random sample $\mathbf{S}_n^* = ((\mathbf{X}_1^*, Y_1^*), (\mathbf{X}_2^*, Y_2^*), \ldots, (\mathbf{X}_n^*, Y_n^*))$ from F_n. In other words, sample pairs of data with replacement from $\mathbf{S}_n = ((\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n))$.

3. Calculate the bootstrap estimate of the coefficient vector $\hat{\boldsymbol{\beta}}$ by using the bootstrap random sample $\mathbf{S}_n^*$. In the case of the least squares estimation this implies

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} \left[Y_i - m\left(\mathbf{X}_i^*, \hat{\boldsymbol{\beta}}\right) \right]^2.$$

4. Independently repeat steps 2 and 3 B times, obtaining bootstrap replications $\hat{\boldsymbol{\beta}}^*(1)$, $\hat{\boldsymbol{\beta}}^*(2)$,..., $\hat{\boldsymbol{\beta}}^*(B)$.

5. Calculate the bootstrap estimation of the standard error of the j^{th} estimated regression coefficient $\hat{\beta}_j$ as

$$\hat{\sigma}_{j,B} = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\beta}_j^*(b) - \hat{\beta}_j^*(\cdot) \right)^2}, \quad j = 0, 1, \ldots, p,$$

where

$$\hat{\beta}_j^*(\cdot) = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}_j^*(b).$$

Note that $\hat{\sigma}_{j,B} \to \hat{\sigma}_j$ as $B \to \infty$ where $\hat{\sigma}_j$ denotes the ideal bootstrap estimate of the standard error of the j^{th} estimated regression coefficient.

2.4.3 The wild bootstrap

The use of the wild bootstrap as a smoothing parameter selection method was already illustrated in Chapter 1, Section 1.7.3. For the sake of completeness the wild bootstrap is discussed below based on Davidson and Flachaire (2008:162-164). Inference on the parameters of the linear regression model

$$Y_i = m(\mathbf{X}_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

requires special precautions when the errors $\boldsymbol{\varepsilon}$ are heteroskedastic. With heteroskedastic errors, the usual ordinary least squares estimator, OLS, of the covariance of the OLS estimates $\hat{\beta}$ is in general asymptotically biased and conventional t and F tests do not have t and F distributions respectively. Bootstrap methods normally rely on simulation to approximate the finite sample distribution of test statistics, under the appropriate null hypothesis. In order for such methods to be reasonably accurate, it is desirable that the data generating process (DGP) used for drawing bootstrap samples should be as close as possible to the true DGP that generated the observed data, assuming the DGP satisfies the null hypothesis. This presents a problem if the null hypothesis admits heteroskedasticity of unknown form: if the form is unknown, it cannot be imitated in the bootstrap DGP. In the face of this difficulty, the wild bootstrap was developed. The wild bootstrap uses a bootstrap DGP of the form:

$$Y_i^* = m\left(\mathbf{X}_i, \hat{\boldsymbol{\beta}}\right) + e_i^*, \ i = 1, \ldots, n, \tag{2.22}$$

where $\hat{\boldsymbol{\beta}}$ is a vector of parameter estimates, and the bootstrap error terms are

$$e_i^* = f_t(e_i)\eta_i \tag{2.23}$$

where $f_t(e_i)$ is a transformation of the OLS residual e_i and the η_i are mutually independent drawings, completely independent of the original data, from some auxiliary distribution such that

$$E(\eta_i) = 0 \text{ and } E(\eta_i^2) = 1.$$

Mammen (1993:257) proposed the following two-point distribution for this purpose:

$$\eta_i = \begin{cases} \frac{(1-\sqrt{5})}{2}, & \text{with probability } \frac{\sqrt{5}+1}{2\sqrt{5}} \\ \frac{(1+\sqrt{5})}{2}, & \text{with probability 1- } \frac{\sqrt{5}+1}{2\sqrt{5}} \end{cases}$$

Liu (1988:1705) proposed the Rademacher distribution:

$$\eta_i = \begin{cases} 1, & \text{with probability } \frac{1}{2} \\ -1, & \text{with probability } \frac{1}{2} \end{cases}$$

Suppose that $f_t(e_i) = e_i$ and the Rademacher distribution is used, then the following Monte Carlo algorithm can be used to estimate the standard errors of $\hat{\boldsymbol{\beta}}$.

Algorithm 2.6. The wild bootstrap

1. Suppose the sample $\mathbf{S}_n = ((\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n))$ is available.

2. Use the sample data to calculate the vector of estimated regression parameters $\boldsymbol{\beta}$, with some estimation technique, say least squares:
$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left[Y_i - m\left(\mathbf{X}_i, \boldsymbol{\beta}\right)\right]^2.$$

3. Calculate the sample residuals as usual from the expression:
$$e_i = Y_i - m(\mathbf{X}_i, \hat{\boldsymbol{\beta}}), \quad i = 1, \ldots, n.$$

4. Draw a bootstrap random sample $\mathbf{e}_n^* = (e_1^*, e_2^*, \ldots, e_n^*)$.
$$e_i^* = \begin{cases} e_i, & \text{with probability } \frac{1}{2} \\ -e_i, & \text{with probability } \frac{1}{2} \end{cases}$$
so that $E(e_i^*) = 0, E(e_i^*)^2 = e_i^2$ and $E(e_i^*)^3 = e_i^3$

5. Obtain the bootstrap sample $\mathbf{S}_n^* = ((\mathbf{X}_1, Y_1^*), (\mathbf{X}_2, Y_2^*), \ldots, (\mathbf{X}_n, Y_n^*))$ by using the relationship:
$$Y_i^* = m\left(\mathbf{X}_i, \hat{\boldsymbol{\beta}}\right) + e_i^*, \quad i = 1, \ldots, n.$$

6. Calculate the bootstrap estimate of the coefficient vector $\hat{\boldsymbol{\beta}}$ by using the bootstrap random sample $\mathbf{S}_n^*$. In the case of the least squares estimation this implies
$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} \left[Y_i^* - m\left(\mathbf{X}_i, \hat{\boldsymbol{\beta}}\right)\right]^2.$$

7. Independently repeat steps 4 to 6 B times, obtaining bootstrap replications $\hat{\boldsymbol{\beta}}^*(1)$, $\hat{\boldsymbol{\beta}}^*(2)$,..., $\hat{\boldsymbol{\beta}}^*(B)$.

8. Calculate the bootstrap estimation of the standard error of the j^{th} estimated regression coefficient $\hat{\beta}_j$ as

$$\hat{\sigma}_{j,B} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}_j^*(b) - \hat{\beta}_j^*(\cdot)\right)^2}, \quad j = 0, 1, \ldots, p,$$

where

$$\hat{\beta}_j^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_j^*(b).$$

The bootstrap method is applicable in various disciplines. In this chapter some fundamental ideas of the bootstrap methodology were defined and illustrated. This serve as an introduction for the discussion of the bagging and bragging methods in the next chapter. Bagging and bragging are bootstrap procedures developed to reduce the variance of estimators by utilising bootstrap samples generated by resampling data pairs. Henceforth in this study the pairs bootstrap approach is used.

Chapter 3

Improvement methods

3.1 Introduction

The aims of this study are closely related to the contents of Chapter 3, i.e. the so-called improvement methods. Below, a general discussion follows regarding the main aspects of these methods. In Chapter 4 a more specific way of applying these methods are displayed in the form of specific algorithms. The aim of Chapter 3 is therefore to provide a general theoretical framework regarding these improvement methods.

The first two improvement methods are bootstrap-based. In Chapter 2 the bootstrap method, which underlie improvement methods, i.e. the bagging and bragging methods, was introduced. Bagging and bragging are variance reduction techniques. The basic principles of bagging and bragging were stated by Swanepoel (1988, 1990) in a functional context and by Breiman (1996a) from a machine learning viewpoint. A brief introduction to ensemble methods is provided in Section 3.2. In the statistical literature Swanepoel (1988, 1990) first introduced the bagging and bragging methods as special choices of a functional when constructing estimates for a parameter $\theta = \psi(F)$ for some suitable functional ψ and distribution function F. These concepts are discussed in Section 3.3. A very general bagging algorithm is also presented. In Section 3.4 variants on the basic bagging method are discussed. In Chapter 1 it was shown that the choice of the smoothing parameter plays a critical role in the performance of the nonparametric regression estimator. Hall and Robinson (2009) showed that the cross-validation bandwidth selection method suffers from high variability. Hall and Robinson (2009) aimed to reduce this variability by applying bagging to the cross-validation bandwidth selection method. In Section 3.5 the application of bagging to cross-validation bandwidth selection, according to Hall and Robinson (2009), is reviewed briefly.

The third improvement method aims to reduce the bias component of the estimator and is referred to as boosting. Boosting is a general method for improving the accuracy of any given learning algorithm (Schapire, 1999:1). The method originated in the machine learning context but the methodology and theory developed to be applicable over a wide range of fields. The method starts off with a sensible estimator and improves

iteratively, based on its performance on a training dataset (Bühlmann & Yu, 2003:324). Bauer and Kohavi (1999) showed that in a variety of classification problems, boosting techniques proved to be an effective method for reducing bias and variance, and improving misclassification rates. At first the application of boosting to regression problems received little attention. Freund and Schapire (1997) showed how boosting can be applied to regression analysis using the *AdaBoost.R algorithm*. The AdaBoost.R algorithm is defined and explained thoroughly in Freund and Schapire (1997) and is omitted in this study. This method reduced regression problems to classification problems, in a complicated way, involving the mapping of all paired samples to an infinite set of binary questions, called *instances*.

A variety of other boosting algorithms in the regression context were proposed, but eventually Breiman (1999) showed that boosting can be viewed as a so-called *functional gradient descent algorithm*, and practical boosting algorithms for regression became possible (Lutz *et al.*, 2008:3331). This gradient descent view connected boosting to the more common optimisation view of statistical inference (Bühlmann & Yu, 2003:423). Friedman (2001) developed boosting methods for regression which involves minimising the squared error loss function, known as L_2*-boosting*. L_2-boosting uses a pre-chosen fitting method, called the learner, iteratively. This comes down to iterative refitting of the residuals. In this study we are interested in using nonparametric regression estimates of the unknown regression curve as base learners. In this study the functional gradient descent view of boosting and L_2-boosting will be discussed in Section 3.7.

3.2 Ensemble methods

Improvements in data collection methods and computing technology have led to an increase in the availability of large datasets and the methods to analyse these datasets (Bühlmann & Yu, 2002:927). The successful development of computationally intensive methods, by using high-speed computers, for example, is extremely useful when analysing large, high dimensional datasets when solving problems where finding a good model or classifier in one step is almost impossible due to the complexity and scale of the problem (Bühlmann & Yu, 2002:927). Some of these developments are referred to as *ensemble learning algorithms*. The aim of ensemble methods is to improve the predictive performance of a given statistical learning or model fitting technique (Bühlmann, 2004:985). The term learner is used by the machine learning community. However, the term estimator is well established in statistics. In the text below some sections will refer specifically to machine learning literature and the term learner will be used in such cases. Ensemble methods train multiple component learners and then combine their predictions, and this generalisation ability is often significantly better than that of a single learner (Zhou & Yang, 2005:48). The general principle of ensemble methods is to construct a linear combination of some model fitting method, instead of using a single fit of the method (Bühlmann, 2004:985). Bühlmann (2004:985-986) explains ensemble methods at the hand of a framework for function estimation.

Suppose estimating a real-valued function,

$$g : \mathbb{R}^d \to \mathbb{R} \tag{3.1}$$

based on data $(X_1, Y_1), \ldots, (X_n, Y_n)$ where $\mathbf{X}$ is a d-dimensional predictor variable and $\mathbf{Y}$ a univariate response variable, is of interest. Generalisations to other functions $g(\cdot)$ and other data-types are possible. Assume there is a specified base procedure which, given some input data, yields an estimated function $\hat{g}(\cdot)$ for g in (3.1). For example, the base procedure could be a nonparametric kernel estimator if d is small. This base procedure can be run many times when changing the input data. The original idea of ensemble methods is to use reweighted original data to obtain different estimates $\hat{g}_1(\cdot),\ \hat{g}_2(\cdot),\ \hat{g}_3(\cdot), \ldots$ based on different reweighted input data. An ensemble-based function estimate $g_{ens}(\cdot)$ is then constructed by taking linear combinations of the individual function estimates $\hat{g}_b(\cdot)$:

$$\hat{g}_{ens}(\cdot) = \sum_{b=1}^{B} c_b \hat{g}_b(\cdot), \tag{3.2}$$

where the $\hat{g}_b(\cdot)$ are obtained from the base procedure based on the b^{th} reweighted dataset. For some ensemble methods, e.g. for bagging (see Section 3.3), the linear combination coefficients $c_b = 1/B$ are averaging weights. According to Bühlmann (2004:986) ensemble methods are relatively simple devices to improve the predictive performance of a base procedure. In Section 3.3 bagging as a variance reduction scheme is discussed. The bagging algorithm, variants of bagging and the application of bagging in the case of cross-validation smoothing parameter selection are also discussed.

3.3 Bagging and related methods

The basic idea of the terms bagging and bragging was introduced by Swanepoel (1988, 1990). The problem of constructing estimates for a parameter θ, is the problem of interest. The parameter is written in the form

$$\theta = \psi(F)$$

for some suitable functional ψ, where ψ depends on the distribution function F of the population. From this functional approach it is shown that if $\psi(F)$ can be approximated by a sequence of functionals, namely

$$\psi_m(F) \approx \psi(F),$$

with the approximation becoming increasingly accurate as $m \to \infty$, then $\psi_m(F_n)$ can be taken as an estimator of θ, with $m = m(n)$ suitably chosen. Here F_n refers to the empirical distribution function defined in (2.4). Suppose $T_m(X_1, X_2, \ldots, X_m)$ is some known estimator of θ, such as the sample mean. Then two possible choices of $\psi_m(F)$ are

$$\psi_{m,1}(F) = E[T_m(X_1, X_2, \ldots, X_m)] \tag{3.3}$$

and

$$\psi_{m,2}(F) = \text{median}[T_m(X_1, X_2, \ldots, X_m)] \tag{3.4}$$

In this case we have that

$$\psi_{m,1}(F_n) = E_*[T_m(X_1^*, X_2^*, \ldots, X_m^*)] \tag{3.5}$$

and

$$\psi_{m,2}(F_n) = \text{median}_*[T_m(X_1^*, X_2^*, \ldots, X_m^*)] \tag{3.6}$$

respectively, where $(X_1^*, X_2^*, \ldots, X_m^*)$ denotes a bootstrap random sample of size m taken from F_n. The ideal bootstrap estimates $\psi_{m,1}(F_n)$ and $\psi_{m,2}(F_n)$ are approximated by Monte Carlo algorithms which are stated in Chapter 4. The choice of $\psi_{m,1}(F_n)$ is known as *bagging* in the statistical literature and $\psi_{m,2}(F_n)$ is known as *bragging*.

Bagging and bragging procedures also developed from the theory of ensemble methods as mentioned before. *Bagging* is an acronym for **b**ootstrap **agg**regat**ing** and was introduced by Breiman (1996a). It developed as an ensemble method for reducing the prediction error of learning algorithms. Bagging is performed by drawing repetitive bootstrap samples from the training sample, applying the learning algorithm to each of these samples, and by averaging/aggregating the resulting prediction rules, that is, averaging or otherwise aggregating the predicted values for test observations. Bagging is a primarily variance reduction tool, i.e bagging is implemented to reduce the variance of a predictor and in this way improve the accuracy of this estimation process. For bagging to succeed in this goal, the procedure (basic learners) that constructs each predictor must be unstable, in the sense that, in the training set changes in the constructed predictor should be significant (Peterson *et al.*, 2007:1696). The resemblance with Swanepoel (1988, 1990) is evident.

The next section considers the bagging methodology and points out how it is applied to reduce the variability of unstable predictors. Some modifications of the method, such as bragging are also discussed. First, an algorithm for bagging is presented according to Bühlmann (2004:986-987).

Algorithm 3.0. General bagging algorithm

Suppose data pairs (X_i, Y_i) $(i = 1, \ldots, n)$ are available. The target function of interest is $m(x) = E[Y|X = x]$. The function estimator, which is the result from a given base procedure, is

$$\hat{g}(\cdot) = q_n((X_1, Y_1), \ldots, (X_n, Y_n))(\cdot) : \ \mathbb{R}^d \to \mathbb{R},$$

where the function $q_n(\cdot)$ defines the estimator as a function of the data. A basic bagging algorithm is defined as follows.

1. Construct a bootstrap sample $(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$ by randomly drawing n times with replacement from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$.

2. Compute the bootstrapped estimator $\hat{g}^*(\cdot)$ by applying the plug-in principle:

$$\hat{g}^*(\cdot) = q_n((X_1^*, Y_1^*), \ldots (X_n^*, Y_n^*))(\cdot).$$

3. Repeat steps 1 and 2 B times, yielding $\hat{g}^{*b}(\cdot)$ $(b = 1, \ldots, B)$. The bagged estimator is then

$$\hat{g}_{bag} = \frac{1}{B} \sum_{b=1}^{B} g^{*b}(\cdot). \tag{3.7}$$

The finite number B in practice governs the accuracy of the Monte Carlo approximation and depends on the sample size n and the computational cost involved in evaluating the predictor.

Bühlmann (2004:988) describes the interplay between bias and variance in nonparametric statistics as follows: The following equality indicates the somewhat unusual approach of using the bootstrap methodology:

$$\hat{g}_{bag} = \hat{g}(\cdot) + (E^*[\hat{g}^*(\cdot)] - \hat{g}(\cdot)) = \hat{g}(\cdot) + \text{Bias}^*(\cdot),$$

where $\text{Bias}^*(\cdot)$ is the bootstrap bias estimate of $\hat{g}(\cdot)$. Usually the bias correction term has a negative sign. In the case of bagging the bootstrap bias estimate is added. Therefore, it is expected that bagging has a higher bias than $\hat{g}(\cdot)$. But according to the usual bias-variance trade-off in nonparametric regression, the hope is to gain more by reducing the variance than increasing the bias. Therefore overall, bagging may pay-off in terms of the MSE.

Together with the *boosting* algorithm presented in Section 3.7.2, which is a bias reduction technique, bias may diminish to some extent. Sometimes, the bagging algorithm is not very functional. Modifications of the bagging method can solve problems that arise when applying bagging.

3.4 Variants of bagging

In the bagging algorithm described in Section 3.3 the bootstrap samples from the training set have empirical distribution functions which are in a "neighbourhood" of the empirical distribution function corresponding to the training sample (Pino-Mejias *et al.*, 2008:266). The resulting procedure does not work in all situations (Bühlmann & Yu, 2002:929). A more general bagging methodology can be defined, where other classes of neighbourhoods of the empirical distribution of the training sample are considered, or where aggregation is done by another rule than averaging (Pino-Mejias *et al.*, 2008:266). These modifications include sampling without replacement, using resample sizes that differ from the original training set's size and aggregating using the median over the multiple versions of the predictor, rather than the mean.

3.4.1 Subagging and moon-bagging

Ordinary bagging is performed by drawing bootstrap samples from the training sample, then applying the learning algorithm to each bootstrap sample, and averaging/aggregating the resulting prediction rules. The sizes of these samples are generally chosen to coincide with the number of observations in the original training dataset. Subsampling and applying the m-out-of n bootstrap method with $m < n$ have been proposed in the statistical literature as alternatives to the standard bootstrap. In general, the motivation of these studies is to repair inconsistencies of the bootstrap estimates obtained by sampling with replacement and $m = n$, or to improve the efficiency of these estimates. In the next two paragraphs subagging and moon-bagging are presented as examples of methods with different resampling schemes than the traditional bagging method.

a) Subagging

For *subagging* (**s**ubsample **agg**regat**ing**) a sample size $m = [an]$ is used, with $0 < a < 1$ and resampling is done *without* replacement (Bühlmann & Yu, 2002:929). Subagging is proposed as a computationally less intensive variant of bagging that is expected to have similar accuracy in regression problems. In some cases m is chosen as $m = [n/2]$. This method is generally known as "half subagging". It saves more than half of the computing time because the computational order of an estimator $\hat{g} = \hat{g}_{(n)}$ is usually at least linear in n (Bühlmann, 2004:993). In Section 3.5 it is shown that it is important to use subagging when implementing cross-validation as the method for choosing the smoothing parameter, since sampling with replacement could result in ties which might cause the cross-validation method to break down (Hall & Robinson, 2009:177).

b) Moon-bagging

Moon-bagging is an acronym for **m**-**o**ut-**o**f-**n** **b**ootstrap **agg**regat**ing**. For this procedure a sample size $m = [an]$ is used, with $0 < a < 1$, where resampling is done *with* replacement. The difference between subagging and moon-bagging essentially disappears if m is small relative to n (Bühlmann & Yu, 2002:946).

3.4.2 Bragging and trimmed bagging

Instead of calculating the mean arithmetic in (3.7) above, a robust location estimator can be used. Bragging and trimmed bagging are examples of such robust methods.

a) Bragging

Swanepoel (1988, 1990) derived the bragging procedure as another choice for the functional $\psi(F)$, as it is defined in (3.4). Bühlmann (2004:994) also provides a brief description of bragging. *Bragging* stands for **b**ootstrap **r**obust **agg**regat**ing**. The *sample median* is used over the B bootstrap estimates $\hat{g}^{*b}(\cdot)$, instead of the sample mean in step 3 of the bagging algorithm.

Algorithm 3.1. A general bragging algorithm

1. Construct a bootstrap sample $(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)$ by randomly drawing n times with replacement from the data $(X_1, Y_1), \ldots, (X_n, Y_n)$.

2. Compute the bootstrapped estimator $\hat{g}^*(\cdot)$ by applying the plug-in principle:

$$\hat{g}^*(\cdot) = q_n((X_1^*, Y_1^*), \ldots (X_n^*, Y_n^*))(\cdot).$$

3. Repeat steps 1 and 2 B times, yielding $\hat{g}^{*b}(\cdot)$ $(b = 1, \ldots, B)$. The bagged estimator is

$$\hat{g}_{bag} = \text{median}_{1 \leq b \leq B} \hat{g}^{*b}(\cdot).$$

This algorithm corresponds to Swanepoel's results in (3.6).

b) Trimmed bagging

Croux *et al.* (2007) proposed trimmed bagging, a procedure where aggregation is done over only the "most accurate" predictors. The *out-of-bag* or *out-of-bootstrap* error rate is used to find the most accurate predictors. Out-of-bag error measures involve using out-of-bag samples and corresponding out-of-bag predictions. Croux *et al* (2007) describes out-of-bag methods as follows: suppose that $\mathbf{S} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ is the training sample that should be used to determine the prediction $g(\cdot)$. In bagging B bootstrap training sets $\mathbf{S}_1^*, \mathbf{S}_2^*, \ldots, \mathbf{S}_B^*$ are used to determine B bootstrap predictions $\hat{g}^{*1}(\cdot), \hat{g}^{*2}(\cdot), \ldots \hat{g}^{*B}(\cdot)$. In each bootstrap sample, some data points will appear multiple times, while other observations will be missing. The observations that are missing from each bootstrap sample, form the out-of-bag sample for that particular bootstrap sample. Denote the array of out-of-bag samples by $\mathbf{S}^{-*1}, \mathbf{S}^{-*2}, \ldots, \mathbf{S}^{-*B}$. The out-of-bag error rate, $\mathrm{ER}(\hat{g}^{*b})$, is defined as the proportion of data points in the out-of-bag sample that are misclassified.
Trimmed bagging involves the following: first, order the bootstrap versions of the classifier by increasing out-of-bag error rate (ER), i.e.

$$\mathrm{ER}(g(\cdot, \mathbf{S}^{*(1)})) \leq \mathrm{ER}(g(\cdot, \mathbf{S}^{*(2)})) \leq \cdots \leq \mathrm{ER}(g(\cdot, \mathbf{S}^{*(B)})).$$

Trim off the $\alpha\%$ bootstrap estimates that perform worst, i.e. that have the largest error rates, and aggregate over only the best performing estimates. Then calculate the trimmed bagged estimator

$$\hat{g}_{trimbag}(x) \approx \mathrm{average}_{1 \leq b \leq [(1-\alpha)B]} \hat{g}(x, \mathbf{S}^{*(b)}),$$

where $[a]$ denotes the largest integer smaller than or equal to a.

Since bagging is a variance reduction technique the trimming portion α should be chosen small enough, such that averaging is still done over a substantial number of learners. However, choosing α too small implies that the risk of including "bad" learners with large error rates in the trimmed sum increases. Croux *et al.* (2007) decided to choose $\alpha = 0.25$ as a rule-of-thumb.

Bagging is applied to hopefully reduce the mean squared error (MSE) of learning algorithms. Breiman (1996a) showed empirically that bagging can indeed be successful in this expectation and provided the following intuitive explanation: for learning methods that are "unstable" functions of the data, averaging over the bootstrap samples reduces the variance component of the MSE. In other words, the variance of the bagged estimator $\hat{g}_{bag}(\cdot)$ is smaller than or equal to the variance of the base learner $\hat{g}(\cdot)$. If the original predictor is indeed "unstable", this reduction is drastic. Instability is defined as follows: a predictor is unstable if small changes in the training sample can cause large changes in the predicted values.

3.5 Bagging and cross-validation bandwidth selection

One of the advantages of cross-validation, as a tool for bandwidth choice, is that it can be employed to a wide variety of estimation scenarios. However, a disadvantage of the cross-validation method is the high variance

of the cross-validated smoothing parameter and the undesirable resulting effects on the performance of the estimators such as high tail variance. Hall and Robinson (2009) show that this variability can be reduced by implementing bagging. In this section arguments from Hall and Robinson (2009) are presented.

3.5.1 Background

According to Hall and Robinson (2009:175) the negative effects on the performance of estimators, caused by the relatively high variance of the cross-validated smoothing parameter, are evident from theoretical analysis. The plug-in bandwidth selectors, here denoted by $\hat{h}_{plug}$, are relatively root-n consistent for the optimal bandwidth h_0 in the sense that

$$\frac{\hat{h}_{plug} - h_0}{h_0} \to 0$$

at a rate of $n^{-1/2}$. In comparison, for the cross-validation bandwidth, here denoted by $\hat{h}_{CV}$, it holds that

$$\frac{\hat{h}_{CV} - h_0}{h_0} \to 0$$

at a much slower rate of $n^{-1/10}$, due to its high variability. According to Hall and Robinson (2009) this stochastic variability of cross-validation can be reduced significantly by using bagging. The bagging algorithm implemented by Hall and Robinson (2009:176) uses without-replacement m-out-of-n resampling with $m < n$. Sampling without replacement is applied since sampling with replacement could result in ties which might cause the cross-validation procedure to break down. They provide an intuitive explanation of the procedure's performance: consider the case in which kernel methods are used to estimate a function, such as probability density or a regression function. The bandwidth h_0 which minimises the mean integrated squared error, typically satisfies

$$h_0 \sim Cn^{-a}, \tag{3.8}$$

where $a > 0$ depends on the 'regularity' of the problem and C is a functional of the target function. The value of a is generally known. Both plug-in and cross-validation estimators of h_0 attempt to estimate C: plug-in estimators in an explicit way and cross-validation implicitly. Let $\hat{C}$ denote the cross-validation estimator of C. Then the cross-validation bandwidth is given by $\hat{h}_{CV} = \hat{C}n^{-a}$. Bagging reduces the variability of $\hat{h}_{CV}$ by substituting $\hat{C}$ by $E^*[\hat{C}^*] = E[\hat{C}^*|\mathbf{S}]$, where $\mathbf{S}$ denotes the original training sample and $\hat{C}^*$ is the bootstrap version of $\hat{C}$. When applied in this way, bagging will mostly reduce the variability of the bandwidth selector, provided that $\hat{C}$ is a highly nonlinear function of the data, which is the case in this setting.
In the next section the application of bagging to cross-validation smoothing parameter selection in kernel regression is considered.

3.5.2 Methodology: bagging applied to h_{CV}

Hall and Robinson (2009:177) explains the methodology as follows: consider the regression problem where the conditional mean response $E[Y|X = x] = m(x)$ is to be estimated using the data $\mathbf{S}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$. Let $\hat{m}_h$ denote a kernel estimator of m with bandwidth h. Let $\hat{m}_{h,(j)}(X_j)$ be the leave-one-out estimator in

the point X_j where the j^{th} observation is deleted when determining the estimator in that point. Recall the cross-validation function

$$CV(h) = \frac{1}{n}\sum_{j=1}^{n}[Y_j - \hat{m}_{h,(j)}(X_j)]^2 \tag{3.9}$$

stated in (1.53). Note that when the cross-validation function was introduced in (1.53) weights were included in the expression. Hall and Robinson (2009) chose the weights to be equal to one for $j = 1, 2, \ldots n$. Therefore the weights are omitted in (3.9). The cross-validation method chooses the bandwidth that minimises $CV(h)$. Denote the value by $\hat{h}_{CV}$. $CV(h)$ can be seen as an empirical approximation to the L_2 error criterion $d_I(h) = \int E[\hat{m}_h(x) - m(x)]^2 f(x)\,dx$ (f denotes the density of the explanatory variables), up to terms that do not depend on h. In this sense $\hat{h}_{CV}$ can be interpreted as an approximation to $h_0 = h_{0n} = \arg\min_h d_I(h)$.

There are at least two ways in which the smoothing parameter can be estimated when using bagged cross-validation. Each method involves using independent sampling operations to draw samples of size $m < n$ repeatedly by sampling randomly, without replacement, from the original n-sample, and calculating $CV^*(h)$, i.e. the cross-validation function, for each of the bootstrap samples,

$$CV^*(h) = \frac{1}{m}\sum_{j=1}^{m}[Y_j^* - \hat{m}^*_{h,(j)}(X_j^*)]^2. \tag{3.10}$$

In this case m refers to the sample size and $\hat{m}^*_{h,(j)}(X_j)$ to the bootstrap leave-one-out estimator in the point X_j^* where the j^{th} observation is deleted when determining the estimator in that point. Now, apply bagged cross-validation in one of the following ways:

1. Bagging $CV(h)$:
   For the first approach $CV_{bag}(h)$ is taken to be the average of a large number of values of the function $CV^*(h)$:
   $$CV_{bag}(h) = E^*[CV^*(h)].$$
   Choose $\hat{h}_{bag}$ to minimise $CV_{bag}(h)$, *after rescaling the bandwidth to take into account the smaller sample size*, i.e. m rather than n, on which $CV_{bag}(h)$ is based Hall and Robinson (2009). Rescaling of the bandwidth is discussed in Section 3.5.3 below.

2. Bagging $\hat{h}_{CV}$:
   The second approach chooses $\hat{h}^*_{CV}$ to minimise $CV^*(h)$ for each given bootstrap sample, and defines $\hat{h}_{bag}$ by averaging $\hat{h}^*_{CV}$ over all bootstrap samples, again rescaling for sample size. More versions exist in the literature, which will be discussed in Chapter 4.

3.5.3 Rescaling of the bandwidth

Hall and Robinson (2009:177) suggest a rescaling of the bandwidth to take into account the smaller sample size, i.e. m rather than n, on which CV_{bag} is based. The rescaling step is very simple, being based on standard

formulae for bandwidth selectors. In the case of kernel methods where the theoretically optimal bandwidth is given by (3.8) the bagged bandwidth based on resample size m should be multiplied by the factor $(m/n)^a$. The aim is to give optimal performance for functions with two derivatives, such as when the regression estimator is of second order, $a = 1/5$ is chosen.

3.6 Bagging and plug-in bandwidth selection

Application of the bootstrap to the cross-validation method of bandwidth selection may not be effective. The fact that sampling must be done without replacement to prevent ties from causing the breakdown of the cross-validation method and the rescaling that is necessary for m-out-of-n without replacement sampling, may influence the results. If plug-in methods are used then bootstrapping with replacement can be applied for all the above methods for $m = n$ and $m < n$. If $m = n$ then the rescaling of the bandwidth is not needed.

Next, the improvement method boosting is discussed.

3.7 Boosting in regression

The learning problem for regression is based on a dataset $\mathbf{S} = (X_i, Y_i)_{i=1}^n$, where $X_i \in \mathbf{X}$, the range of possible values which the predictor variable can attain, and $Y_i \in \mathbb{R}$. This is in contrast to the classification case. In the classification case the variable Y is restricted to a finite set (Meir & Rätsch, 2003:165). Freund and Schapire (1997) provided a suggestion as to how boosting might produce regression models using their algorithm *AdaBoost.R*, but this method involved the reduction of the regression problem to a classification problem. Practical regression boosting algorithms could only emerge after Breiman (1999) showed that boosting can be viewed as a functional gradient descent algorithm (Lutz et al., 2008:3331). This gradient descent view connected boosting to the more common optimisation view of statistical inference (Bühlmann & Yu, 2003:423). Friedman (2001) developed boosting methods for regression which involves minimising the squared error loss function, known as L_2-boosting. Two viewpoints are presented: the gradient descent view of boosting is discussed in Section 3.7.1 and L_2-boosting is discussed in Section 3.7.2.

3.7.1 Functional gradient descent (FGD) view of boosting

Suppose one has a set of independent, identically distributed random variables $\mathbf{S}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$, where $X_1, X_2, \ldots, X_n$ are one-dimensional predictor variables and $Y_1, Y_2, \ldots, Y_n$ are one-dimensional response variables. We adapt the notation of Bühlmann and Yu (2003). According to Bühlmann and Yu (2003:325) the aim is to determine the function $m : \mathbb{R} \to \mathbb{R}$ such that the expectation of some cost function is minimised:

$$m(\cdot) = \arg\min_{m(\cdot)} E[C(Y, m(X))],\ C(\cdot,\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+, \tag{3.11}$$

where (X, Y) follows the same distribution as each of the $\{(X_i, Y_i)\}_{i=1}^n$. The cost function, C, is assumed to be differentiable and convex in the second argument, to ensure that the gradient method works well.

The most prominent cost functions are:

1. $C(y, f) = \exp(yf)$ with $y \in \{-1, 1\}$, which is the cost function for AdaBoost;
2. $C(y, f) = \log_2(1 + \exp(-2yf))$ with $y \in \{-1, 1\}$, which is the cost function for LogitBoost; and
3. $C(y, f) = (y - f)^2/2$ with $y \in \mathbb{R}$ or $y \in \{-1, 1\}$, which is the cost function for L_2Boost.

The population minimisers of (3.11) are then

$$m(x) = \frac{1}{2} \log \left(\frac{P[Y = 1|X = x]}{P[Y = -1|X = x]} \right)$$

for AdaBoost and LogitBoost and

$$m(x) = E[Y|X = x]$$

for L_2Boost, i.e. the regression relation. $m(\cdot)$ can be estimated from the data via a constrained minimisation of the empirical risk,

$$\frac{1}{n} \sum_{i=1}^{n} C(Y_i, m(X_i)), \tag{3.12}$$

applying the FGD iteratively. Bühlmann and Hothorn (2007:480) present the generic FGD or boosting algorithm as follows:

Algorithm 3.2. Generic FGD boosting

1. Fit an initial learner, $\hat{m}_0(\cdot)$. Possible choices are $\hat{m}_0(\cdot) \equiv \arg\min_c \sum_{i=1}^{n} C(Y_i, c)$, or $\hat{m}_0(\cdot) \equiv 0$. Set $t = 0$.
2. Let t=t+1. Compute the negative gradient vector,
$$U_i = -\frac{\delta C(Y_i, m)}{\delta m}\Big|_{m=\hat{m}_{t-1}(X_i)}, i = 1, \ldots, n.$$
3. Fit the negative gradient vector $U_1, U_2, \ldots, U_n$ to $X_1, X_2, \ldots, X_n$ by the base procedure. Call this fit $\tilde{m}_t(\cdot)$. Thus $\tilde{m}_t(x)$ can be viewed as an approximation of the negative gradient vector.
4. Update
$$\hat{m}_t(\cdot) = \hat{m}_{t-1}(\cdot) + \nu \tilde{m}_t(\cdot),$$
where $0 < \nu \leq 1$ is a real-valued step-length factor, i.e. proceed along an estimate of the negative gradient vector.
5. Repeat steps 2 to 4 until $t = T$, where T is some stopping value for the number of iterations.

In the algorithm, two values need specification: T, the number of iterations and ν, the step-length factor. These serve as regularisation parameters, ensuring that the data are not over-fitted. According to Bühlmann and Hothorn (2007:480) the stopping iteration, T, which is the main tuning parameter, can be determined by cross-validation or some information criterion. Next, the step-length factor ν is considered. Friedman (2001) suggested using an additional line search between steps 3 and 4. Bühlmann and Hothorn (2007:480) states that

this additional line search is rather unnecessary for achieving a good estimator. The choice of the step-length factor ν is of minor importance, as long as the value is small, such as $\nu = 0.1$ (Bühlmann & Horton, 2007:480). In their application of boosting to the Nadaraya-Watson estimator, Marzio and Taylor (2008) chose $\nu = 1$. In the present study $\nu = 1$ is used.

Consider now L_2-boosting, the boosting algorithm that in the FGD view uses the squared error loss function.

3.7.2 L_2-boosting

According to Bühlmann and Hothorn (2007:483) the L_2-boosting algorithm is the simplest and most instructive boosting algorithm. This algorithm is very useful for regression, in particular in presence of many predictor variables. L_2-boosting applies the functional gradient descent view of boosting by using the squared error loss which amounts to repeated fitting of ordinary residuals. Applying the general description of the FGD algorithm from Section 4.2.1 the following algorithm is obtained:

Algorithm 3.3. L_2**-boosting**

1. Suppose the data $\mathbf{S}_n = (X_1, Y_1), \ldots, (X_n, Y_n)$ is given.
2. Fit an initial learner, $\hat{m}_0(\cdot)$. The default value is $\hat{m}_0(\cdot) \equiv \bar{Y}$. Set $t = 0$.
3. Let $t = t + 1$. Compute the residuals,
$$U_i = Y_i - \hat{m}_{t-1}(X_i),\ i = 1, \ldots, n.$$
4. Fit the residual vector $U_1, U_2, \ldots, U_n$ to $X_1, X_2, \ldots, X_n$ by the base procedure. Call this fit $\tilde{m}_t(\cdot)$.
5. Update
$$\hat{m}_t(\cdot) = \hat{m}_{t-1}(\cdot) + \nu \tilde{m}_t(\cdot),$$
where $0 < \nu \leq 1$ is a real-valued step-length factor, as in the general FGD algorithm.
6. Repeat steps 3 to 5 until $t = T$, where T is some stopping value for the number of iterations.

According to Bühlmann and Yu (2003:328) boosting until theoretical convergence with $T = \infty$ is typically not a good strategy. The MSE with $T = \infty$ is not smaller than the noise level σ^2. The reason is that boosting infinitely often yields the fully saturated model that fits the data exactly. This implies that the bias is equal to zero, but the variance is equal to σ^2. In practice, the data-driven optimal boosting iteration can be selected through cross-validation. In Bühlmann and Yu (2003:327) they show a bias-variance trade-off that is in contrast to the standard algebraic bias-variance trade-off commonly seen in nonparametric regression estimation. As the boosting iteration T varies, both the bias and variance terms change exponentially, with the bias decreasing exponentially fast and the variance increasing with exponentially diminishing terms eventually. The advantage of this new exponential trade-off is a flatter near-optimal region for the optimal T: the trade-off not only gets very close to the optimal MSE, but also stays quite flat afterward. Bühlmann and Yu (2003:336) concludes that

the computationally simple L_2 boosting is only successful if the learner is sufficiently weak, that is the base learner has low variance. If the learner is too strong then even after the first boosting iteration, the MSE is not improved over the original learner. It is also likely that the generalisation error will then increase with the number of boosting steps, stabilising eventually due to the numerical convergence of the boosting algorithm to the fully saturated model.

In this study the L_2-boosting algorithm will be applied with the Nadaraya-Watson estimator with cross-validation bandwidth selection, the Nadaraya-Watson with plug-in bandwidth selection, the local linear estimator with plug-in bandwidth selection and Yao's (2012) estimator with cross-validation bandwidth selection, the respective learners.

This chapter provided a general theoretical framework regarding the bagging, bragging and boosting improvement methods. In the following chapter a more specific way of applying these methods to nonparametric regression estimators are displayed in the form of specific algorithms.

Chapter 4

New contributions and simulation studies

4.1 Introduction: Aims of the study

For the sake of the reader it is important to summarise the aims set for the present study.

- Firstly, three enhanced regression estimation methods based on kernel regression estimation methods, involving bandwidth selection procedures are studied, evaluated and compared with respect to well-known global error criteria. Performances throughout are evaluated in terms of the MISE, integrated squared bias and integrated variance of the method, which are defined in Section 4.3.6. The three nonparametric regression estimators of concern are the Nadaraya-Watson estimator, the local linear estimator and a new bias corrected nonparametric estimator proposed by Yao (2012). The appropriate smoothing parameter selection methods involve the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method.

- Secondly, three improvement methods are applied to the three nonparametric regression estimators, and evaluated with respect to the global error criteria measured. The effects caused by a bias reduction method, i.e. boosting, as well as bootstrap-based variance reduction methods, i.e. bagging and bragging, are studied, measured and interpreted. Specifically, three ways of determining bandwidths by using the bagging method, as well as three ways of determining bandwidths by using the bragging method, will be executed and the effect of these bandwidths as it will be applied in the various regression estimation methods, will be evaluated in terms of global error criteria. Boshoff (2009) did a limited study, confirming some influence of boosting, bagging, bragging and combinations thereof, on the Nadaraya-Watson estimator with cross-validation bandwidth selection. The effect of boosting, bagging and bragging on the Nadaraya-Watson estimator with plug-in bandwidth selection, the local linear estimator with plug-in bandwidth selection and Yao's estimator with cross-validation bandwidth selection seem to be new contributions to the literature.

- Thirdly, the effect of the smoothing parameter selection methods involving the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection on the performance of the nonparametric regression estimator are evaluated with respect to the global error criteria. Specifically, the Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the Nadaraya-Watson estimator with plug-in bandwidth selection, especially with regard to bias behaviour.
- Computer time for the various methods is a matter of concern and will be reported on in general.
- The behaviour of the bias component of the estimators is of interest throughout the study.

It is also important to again restate important key definitions and formulas regarding the study, and to make reading easier, by revising specific procedures very briefly as it will be applied. The remainder of Chapter 4 consists of detailed descriptions of the methods and algorithms applied in this study.

4.2 Revision of basic procedures and introduction of new additions

To make reading easier, this section focuses on a brief review of the basic components of nonparametric regression estimation. In Section 4.2.1 the Nadaraya-Watson smoother used in the present study is presented together with appropriate smoothing parameter selection methods which involves the leave-one-out cross-validation bandwidth selection method and the plug-in method selection method. In Section 4.2.2 the local linear estimator as well as possible plug-in bandwidth selection methods are discussed, while Section 4.2.3 contains definitions and discussions regarding Yao's estimator and the application of the leave-one-out cross-validation bandwidth selection method to this estimator. The construction of nonparametric kernel regression estimators is briefly reviewed.

Let $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample. The regression relationship can be formulated as

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

where $\varepsilon_i = Y_i - m(X_i)$ satisfies $E(\varepsilon_i | X_i) = 0$ (Chu & Marron, 1991:407). Therefore, Y_i can be considered as the sum of the value of the regression function at X_i and some error term ε_i, where the expected value of the error is zero. According to Györfi *et al.* (2002:18) this motivates the construction of the estimates by local averaging, i.e. estimation of $m(x)$ by the average of those Y_i where X_i is close to x. Such an estimate can be written as

$$\hat{m}(x) = \frac{1}{n} \sum_{i=1}^{n} W_{ni}(x) Y_i, \tag{4.1}$$

where $\{W_{ni}(x)\}_{i=1}^{n}$ denotes a sequence of weights which may depend on the whole vector $\{X_i\}_{i=1}^{n}$ (Härdle, 1990:16). The shape of the weight function is described by a density function with a scale parameter that adjusts the size and the form of the weights near x (Härdle, 1990:24). This shape function is referred to as a kernel K. The weight sequence for kernel smoothers is defined by

$$W_{hi}(x) = \frac{K_h(x - X_i)}{\hat{f}_h(x)}, \tag{4.2}$$

where

$$K_h(u) = h^{-1}K(u/h)$$

is the kernel with scale factor h and $\hat{f}_h(x)$ denotes some estimate of the marginal density $f(x)$ of X.

Now, the three regression estimators used in this study are reviewed, as well as applicable methods of bandwidth selection, suitable for each estimator.

4.2.1 The Nadaraya-Watson estimator and bandwidth selection methods

In this section the Nadaraya-Watson kernel regression estimator is briefly restated from Section 1.4.1 as well as the appropriate bandwidth selection methods to be used, i.e. the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method. Furthermore, two basic algorithms corresponding with the two bandwidth selection methods are outlined to enable the calculation of the Nadaraya-Watson estimator.

a) Nadaraya-Watson estimator

Nadaraya (1964) and Watson (1964) proposed what is known as the Nadaraya-Watson estimator, i.e.

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} K_h(x - X_i)Y_i}{\sum_{i=1}^{n} K_h(x - X_i)}, \tag{4.3}$$

where

$$K_h(u) = h^{-1}K(u/h) \tag{4.4}$$

represents the kernel with scale factor h. Härdle (1990:77) defines the pointwise asymptotic bias for the Nadaraya-Watson smoother as

$$\text{Bias}[\hat{m}(x)] = \left(m''(x) + \frac{2m'(x)f'(x)}{f(x)}\right)\frac{h^2}{2}\int u^2K(u)\,du. \tag{4.5}$$

The pointwise asymptotic variance for the Nadaraya-Watson smoother is defined by

$$\text{Var}[\hat{m}(x)] = \frac{\sigma^2(x)}{f(x)nh}\int K^2(u)\,du. \tag{4.6}$$

b) Cross-validation bandwidth selection for the Nadaraya-Watson estimator

According to Hall and Robinson (2009) the choice of the bandwidth plays a critical role in the performance of nonparametric kernel regression estimators. In Section 1.7 various ways to choose the bandwidth were discussed. For the Nadaraya-Watson smoother the leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method are used in this study. *The plug-in application seems to be a new contribution to the literature since no applications in literature could be found. These two methods are now repeated for easy reading purposes.*

Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be given as above. Leave-one-out cross-validation utilises

the regression smoother after deleting the j^{th} observation, to estimate $\hat{m}$ in the point X_j. The Nadaraya-Watson estimator where the j^{th} observation has been left out is then defined by

$$\hat{m}_{NW,(j)}(X_j) = \frac{\sum_{i=1,i\neq j}^{n} K_h(X_j - X_i)Y_i}{\sum_{i=1,i\neq j}^{n} K_h(X_j - X_i)}, \tag{4.7}$$

where $K_h(u)$ is defined in (4.4). The cross-validation function is then calculated as the sum of the squared deleted residuals

$$CV(h) = \frac{1}{n}\sum_{j=1}^{n}[Y_j - \hat{m}_{NW,(j)}(X_j)]^2. \tag{4.8}$$

Note that when the cross-validation function was introduced in (1.53), weights were included in the expression. In the present study the weights are chosen to be 1 for $j = 1, 2, \ldots, n$ and therefore the weights are omitted in (4.8) as was done by Hall and Robinson (2009). The cross-validation method chooses the bandwidth that minimises $CV(h)$ over a grid of possible h values. Denote this bandwidth by $\hat{h}_{CV}$.

A general algorithm is now presented for determining the standard Nadaraya-Watson estimator using the cross-validation method for bandwidth selection.

Algorithm 4.0. The standard Nadaraya-Watson regression estimator using the cross-validation method for bandwidth selection

1. Let $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$ be the given sample.

2. Determine a grid of possible values, i.e. H, for the bandwidth h. See Section 4.4.1 regarding grid issues.

3. Calculate the leave-one-out estimator for a given bandwidth h, i.e.

$$\hat{m}_{NW,(j)}(X_j) = \frac{\sum_{i=1,i\neq j}^{n} K_h(X_j - X_i)Y_i}{\sum_{i=1,i\neq j}^{n} K_h(X_j - X_i)}, \tag{4.9}$$

   where K_h is the kernel of interest, with scale factor h.

4. Determine the cross-validation function for the particular bandwidth h, i.e.

$$CV(h) = \frac{1}{n}\sum_{j=1}^{n}[Y_j - \hat{m}_{(j)}(X_j)]^2,$$

   where $\hat{m}_{(j)}(\cdot)$ is the proposed regression estimate obtained by leaving out the j^{th} observation.

5. Repeat steps 3 and 4 for the complete grid of bandwidths, to obtain $CV(h)$ for each h value.

6. Select $\hat{h}_{CV}$ as the bandwidth that minimises $CV(h)$ in step 4.

7. Calculate the Nadaraya-Watson estimator using the selected bandwidth $\hat{h}_{CV}$, i.e.

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} K_{\hat{h}_{CV}}(x - X_i)Y_i}{\sum_{i=1}^{n} K_{\hat{h}_{CV}}(x - X_i)}, \tag{4.10}$$

   where $K_{\hat{h}_{CV}}$ is the kernel of interest, with scale factor $\hat{h}_{CV}$.

c) Plug-in bandwidth selection for the Nadaraya-Watson estimator

Hansen (2009), shows that explicit expressions exist for the asymptotic variance and bias of the Nadaraya-Watson estimator. This can be simplified to derive the optimal bandwidth for the conditional mean integrated square error (MISE):

$$h_{opt} = \left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\int \sigma^2(x) w_0(x)\, dx}{\left\{\int \frac{1}{2} m''(x) + m'(x)\frac{f'(x)}{f(x)}\right\}^2 w_0(x) f(x)\, dx}\right]^{1/5} n^{-1/5}, \tag{4.11}$$

where $c_K = \int u^2 K(u)\, du$ and $d_K = \int K^2(u)\, du$. Unfortunately, this optimal bandwidth expression contains unknown quantities $\sigma^2(\cdot), m'(\cdot), m''(\cdot), f(\cdot)$ and $f'(\cdot)$, which need to be estimated. Assuming the conditional variance $\sigma^2(x)$ is constant and substituting the pilot estimates $\hat{m}'(\cdot)$, $\hat{m}''(\cdot)$, $\hat{f}'(\cdot)$, $\hat{f}''(\cdot)$ and σ^2 into the expression for the optimal bandwidth in (4.11), the following expression is obtained

$$\left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\hat{\sigma}^2 \int w_0(x)\, dx}{\{\int \frac{1}{2}\hat{m}''(x) + \hat{m}'(x)\frac{\hat{f}'(x)}{\hat{f}(x)}\}^2 w_0(x)\hat{f}(x)\, dx}\right]^{1/5} n^{-1/5}.$$

The denominator in the above expression can be estimated by

$$\frac{1}{n}\sum_{i=1}^{n} \left\{\hat{m}''(X_i) + \hat{m}'(X_i)\frac{\hat{f}'(X_i)}{\hat{f}(X_i)}\right\}^2 w_0(X_i),$$

which leads to the plug-in bandwidth selector

$$h_{plug} = \left[\frac{d_K}{4c_K^2}\right]^{1/5} \left[\frac{\hat{\sigma}^2 \int w_0(x)\, dx}{\frac{1}{n}\sum_{i=1}^{n} \left\{\hat{m}''(X_i) + \hat{m}'(X_i)\frac{\hat{f}'(X_i)}{\hat{f}(X_i)}\right\}^2 w_0(X_i)}\right]^{1/5}. \tag{4.12}$$

The unknown quantities are replaced by parametric estimators, where parametric polynomial regression is used i.e., fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{4.13}$$

where k is an appropriate integer. Utilising the standardised residual sum of squares from this parametric fit leads to an estimator for σ^2, i.e. $\hat{\sigma}^2$. Furthermore, this choice of a global fit enables calculations of the derivative functions $\hat{m}^{(k)}(x)$, which are of quadratic form, allowing for a certain flexibility in estimating the curvature.
A general algorithm can now be stated to calculate the Nadaraya-Watson regression estimator by using the plug-in bandwidth selection method.

Algorithm 4.1. The Nadaraya-Watson estimator using the plug-in method for bandwidth selection

1. Let $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$ be the given sample.

2. Fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{4.14}$$

   where k is an appropriate integer.

3. Assume that the conditional variance $\sigma^2(x)$ is a constant. Calculate the standardised residual sum of squares of the polynomial fit as an estimate for σ^2, i.e.,

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n}(\hat{\varepsilon}_j - \hat{\bar{\varepsilon}})^2/(n-k+1), \tag{4.15}$$

where $\hat{\bar{\varepsilon}}$ denotes the mean of the residuals and k denotes the order of the polynomial used in step 2. This was suggested as $\hat{\sigma}^2$ by Fan and Gijbels (1996:111).

4. Derive expressions for the first and second derivatives of the k^{th} order polynomial defined in (4.14). Evaluate $\hat{m}'(\cdot)$ and $\hat{m}''(\cdot)$ in each data point, $i = 1, 2, \ldots, n$.

5. Determine kernel estimators of $f(\cdot)$ and $f'(\cdot)$, and evaluate it in each data point, i.e.

$$\hat{f}(X_i) = \frac{1}{n\hat{h}}\sum_{j=1}^{n} K\left(\frac{X_i - X_j}{\hat{h}}\right) \tag{4.16}$$

and

$$\hat{f}'(X_i) = \frac{1}{n\hat{h}^2}\sum_{j=1}^{n} K'\left(\frac{X_i - X_j}{\hat{h}}\right) \tag{4.17}$$

where K is an appropriate kernel function. The bandwidth $\hat{h}$ is defined as

$$\hat{h} = 1.059 s_n n^{-1/5} \tag{4.18}$$

and

$$s_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2}. \tag{4.19}$$

6. Calculate the constants c_K and d_K, i.e.

$$c_K = \int u^2 K(u)\, du \tag{4.20}$$

$$d_K = \int K^2(u)\, du, \tag{4.21}$$

where K denotes the kernel function used.

7. Calculate the plug-in bandwidth

$$\hat{h}_{plug} = \left[\frac{d_K}{4c_K^2}\right]^{1/5}\left[\frac{\hat{\sigma}^2 \int w_0(x)\, dx}{\frac{1}{n}\sum_{i=1}^{n}\left\{\hat{m}''(X_i) + \hat{m}'(X_i)\frac{\hat{f}'(X_i)}{\hat{f}(X_i)}\right\}^2 w_0(X_i)}\right]^{1/5}, \tag{4.22}$$

where $w_0(x) > 0$ is a sensible weight function.

8. Calculate the Nadaraya-Watson estimator using the selected bandwidth $\hat{h}_{plug}$, i.e. evaluate

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} K_{\hat{h}_{plug}}(x - X_i)Y_i}{\sum_{i=1}^{n} K_{\hat{h}_{plug}}(x - X_i)}. \tag{4.23}$$

4.2.2 The local linear estimator and bandwidth selection methods

In this section the local linear regression estimator is briefly revised and the appropriate bandwidth selection method to be used is discussed, i.e. the plug-in bandwidth selection method. Furthermore, an algorithm corresponding with the plug-in bandwidth selection method is stated to enable the calculation of the local linear smoother.

a) Local linear estimator

This section is based on Fan and Gijbels (1996:57-58). Suppose that the $(p+1)^{th}$ derivative of $m(x)$ at the point x_0 exists. The unknown regression function $m(x)$ is then approximated locally by a polynomial of order p. A Taylor expansion yields, for x in a neighbourhood of x_0,

$$m(x) \approx m(x_0) + m'(x_0)(x - x_0) + \frac{m''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{m^{(p)}(x_0)}{p!}(x - x_0)^p. \tag{4.24}$$

This polynomial is fitted locally by a weighted least squares regression method, i.e. by minimising

$$\sum_{i=1}^{n} \left\{ Y_i - \sum_{j=0}^{p} \beta_j (X_i - x_0)^j \right\}^2 K_h(X_i - x_0), \tag{4.25}$$

where $K_h(\cdot)$ denotes a kernel function and h is a bandwidth. With $p = 1$ the estimator $\hat{m}_{LL}$ is termed a *local linear regression smoother* or a *local linear fit*. The problem of estimating $\hat{m}_{LL}$ is equivalent to estimating the intercept β_0. Now consider a weighted local linear regression: finding β_0 and β_1 to minimise

$$\sum_{i=1}^{n} \{Y_i - \beta_0 - \beta_1(X_i - x_0)\}^2 K_h(X_i - x_0) \tag{4.26}$$

Let $\hat{\beta}_0$ and $\hat{\beta}_1$ be the solution to the weighted least squares problem in (4.26). Simple calculations yields

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}. \tag{4.27}$$

The local linear regression smoother is defined by

$$\hat{m}_{LL}(x) = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i}, \tag{4.28}$$

with

$$w_i = K\left(\frac{x - X_i}{h}\right)[s_{n,2} - (x - X_i)s_{n,1}], \tag{4.29}$$

where

$$s_{n,l} = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)(x - X_i)^l, \quad l = 1, 2. \tag{4.30}$$

Furthermore, Fan and Gijbels (1996:17) defines the pointwise asymptotic bias of the local linear smoother as

$$\text{Bias}[\hat{m}(x)] = m''(x)\frac{h^2}{2}\int u^2 K(u)\, du, \tag{4.31}$$

and the pointwise asymptotic variance of the local linear smoother as

$$\text{Var}[\hat{m}(x)] = \frac{\sigma^2(x)}{f(x)nh}\int K^2(u)\, du. \tag{4.32}$$

b) Plug-in bandwidth selection for the local linear estimator

From Fan (1992), it is known that the optimal bandwidth for the conditional mean integrated squared error (MISE) is

$$h_{opt} = \left(\frac{d_K \int f^{-1}(x)\sigma^2(x)w(x)\,dx}{c_K^2 \int (m''(x))^2 w(x)\,dx} \right)^{1/5} n^{-1/5}, \tag{4.33}$$

where $c_K = \int u^2 K(u)\,du$ and $d_K = \int K^2(u)\,du$. The optimal bandwidth defined in (4.33) contains the unknown quantities $\sigma^2(\cdot), m''(\cdot)$ and $f(\cdot)$, which needs to be estimated. Suppose one can take $w(x) = f(x)w_0(x)$ for some specific function w_0 and assume the conditional variance $\sigma^2(x)$ is constant. Substituting the pilot estimates $\hat{m}(\cdot)$ and σ^2 into the expression for the optimal bandwidth in (4.33), the following expression is obtained

$$\hat{h}_{opt} = \left(\frac{d_K \hat{\sigma}^2 \int w_0(x)\,dx}{c_K^2 \int (\hat{m}''(x))^2 w_0(x) f(x)\,dx} \right)^{1/5} n^{-1/5}. \tag{4.34}$$

The denominator in the above expression can be estimated by

$$\frac{1}{n}\sum_{i=1}^{n} \hat{m}''(X_i)^2 w_0(X_i),$$

which leads to the plug-in bandwidth selector

$$\hat{h}_{plug} = \left(\frac{d_K \hat{\sigma}^2 \int w_0(x)\,dx}{c_K^2 \sum_{i=1}^{n} \{\hat{m}''(X_i)\}^2 w_0(X_i)} \right)^{1/5}. \tag{4.35}$$

The unknown quantities are replaced by parametric estimators, where parametric polynomial regression is used i.e., fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{4.36}$$

where k is an appropriate integer. Utilising the standardised residual sum of squares from this parametric fit leads to an estimator for σ^2, i.e. $\hat{\sigma}^2$. Furthermore, this choice of a global fit enables calculations of the derivative functions $\hat{m}^{(k)}(x)$, which are of quadratic form, allowing for a certain flexibility in estimating the curvature. A general algorithm can now be stated.

Algorithm 4.2. The local linear estimator using the plug-in bandwidth selection method

1. Let $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$ be the given sample.

2. Fit a polynomial of order k globally to $m(x)$, leading to the parametric fit

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \cdots + \hat{\beta}_k x^k, \tag{4.37}$$

   where k is an appropriate integer.

3. Assume that the conditional variance $\sigma^2(x)$ is a constant. Calculate the standardised residual sum of squares of the polynomial fit as an estimate for σ^2, i.e.,

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{j=1}^{n} (\hat{\varepsilon}_j - \hat{\bar{\varepsilon}})^2/(n-k+1), \tag{4.38}$$

   where $\hat{\bar{\varepsilon}}$ denotes the mean of the residuals and k denotes the order of the polynomial used in step 2. This was suggested as $\hat{\sigma}^2$ by Fan and Gijbels (1996:111).

4. Derive expressions for the first and second derivatives of the k^{th} order polynomial defined in (4.37). Evaluate $\hat{m}''(\cdot)$ in each data point, $i = 1, 2, \ldots, n$.

5. Calculate the constants c_K and d_K, i.e.

$$c_K = \int u^2 K(u)\, du \tag{4.39}$$

$$d_K = \int K^2(u)\, du, \tag{4.40}$$

where K denotes the kernel used.

6. Calculate the plug-in bandwidth

$$\hat{h}_{plug} = \left(\frac{d_K \hat{\sigma}^2 \int w_0(x)\, dx}{c_K^2 \sum_{i=1}^{n} \{\hat{m}''(X_i\}^2 w_0(X_i)} \right)^{1/5}, \tag{4.41}$$

where $w_0(x) > 0$ is a sensible weight function.

7. Calculate the local linear estimator using the selected bandwidth $\hat{h}_{plug}$, i.e. evaluate

$$\hat{m}_{LL} = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{j=1}^{n} w_i}, \tag{4.42}$$

with w_i and $s_{n,l}$ defined in (4.29) and (4.30) above.

4.2.3 The BRNP estimator and bandwidth selection methods

In this section Yao's estimator is introduced and discussed, as well as the appropriate bandwidth selection method to be used, i.e. the leave-one-out cross-validation bandwidth selection method. Furthermore, an algorithm corresponding with the cross-validation bandwidth selection method is stated to enable the calculation of the new smoother. This is a very computer time consuming method. However, Yao (2012) found this bandwidth selection method the best in his limited study.

a) BRNP estimator

Yao (2012) proposed a new bias reduction nonparametric regression estimator by employing the bias-correction idea of Chung and Lindsay (2011), which was developed for density estimation. They proposed a bias-corrected density estimator via a nonparametric mixture model using a two-step EM algorithm. The new estimator proposed by Yao (2012) is based on the bias cancellation idea of Chung and Lindsay's (2011) proof. The new nonparametric regression estimator has asymptotic bias of order h^4 in contrast to the local linear estimators' asymptotic bias of h^2. It has the same order of asymptotic variance as the local linear estimator. However, the bias expression depends on the design density $f(x)$. Yao (2012) claims that the BRNP estimator is not adapted to non-uniform designs. The new nonparametric regression estimator is defined below. The mathematics surrounding this method is extensive and are omitted. It appears in Yao (2012) in an extensive appendix.

First, let $f(x)$ be the *density of the predictor* X. A usual *kernel density estimator* has the asymptotic bias

of order h^2. Chung and Lindsay (2011) proposed a new *bias-corrected density estimate* which has the asymptotic bias of order h^4. The bias-corrected density estimator given $(X_1, \ldots, X_n)$ is

$$\hat{f}(x) = n^{-2}c(h)\sum_i \sum_j w_j K_{\sqrt{3}h}(X_i - X_j)K_{\sqrt{3}h}(X_i - x)K_{\sqrt{3}h}(X_j - x), \tag{4.43}$$

where

$$c(h) = 3\sqrt{\pi h} \tag{4.44}$$

and

$$w_j = \left\{ \frac{1}{n}\sum_{i=1}^{n} K_{\sqrt{2}h}(X_i - X_j) \right\}^{-1}. \tag{4.45}$$

Here w_j is the regular *kernel density estimator* of $f(x_j)$ with bandwidth $\sqrt{2}h$.

Based on the bias-correction idea of Chung and Lindsay's (2011) proof, Yao (2012) proposed the following *bias-corrected nonparametric regression estimate*:

$$\hat{m}_{BRNP}(x) = \frac{n^{-2}c(h)\sum_i \sum_j v_j Y_i Y_j K_{\sqrt{3}h}(X_i - X_j)K_{\sqrt{3}h}(X_i - x)K_{\sqrt{3}h}(X_j - x)}{n^{-2}c(h)\sum_i \sum_j w_j \sqrt{3}h(X_i - X_j)K_{\sqrt{3}h}(X_i - x)K_{\sqrt{3}h}(X_j - x)} = \frac{N}{D} \tag{4.46}$$

where

$$v_j = \left\{ \frac{1}{n}\sum_{i=1}^{n} K_{\sqrt{2}h}(X_i - X_j)Y_i \right\}^{-1}, \tag{4.47}$$

$c(h) = 3\sqrt{2\pi}h$ and w_j is defined in (4.45).

Furthermore, Yao (2012) defines the pointwise asymptotic bias of the smoother as

$$\text{Bias}[\hat{m}(x)] = \frac{h^4\{A - m(x)B\}}{f(x)}, \tag{4.48}$$

and the pointwise asymptotic variance as

$$\text{Var}[\hat{m}(x)] = \frac{\sigma^2(x)}{nh\sqrt{\pi}f(x)}\left(\sqrt{2} + \frac{1}{4} - \frac{2}{\sqrt{3}}\right) \tag{4.49}$$

where

$$A = \left[-g^{(4)}(x) + g^{-1}(x)g'(x)g'''(x) + g^{-1}(x)g''(x)^2 - g'(x)^2 g^{-2}(x)g''(x)\right] \tag{4.50}$$

and

$$B = \left[-f^{(4)}(x) + f^{-1}(x)f'(x)f'''(x) + f^{-1}(x)f''(x)^2 - f'(x)^2 f^{-2}(x)f''(x)\right] \tag{4.51}$$

with $g(x) = m(x)f(x)$. From (4.48) it is clear that the bias depends on $f(x)$. The estimator in (4.46) does not belong to the class of linear smoothers $\hat{m}(x) = \sum_{i=1}^{n} W_{ni}(x)Y_i$ where $W_{ni}(\cdot)$ is a weight function not being dependent on Y_i. Also note that the denominator D is the same expression as (4.43), i.e. the density estimator of Chung and Lindsay (2011). Regarding bandwidth selection, Yao (2012) used two methods, i.e. cross-validation bandwidth selection method and plug-in bandwidth selection method. Since the cross-validation method performed better, the present study focuses only on the cross-validation bandwidth selection method which is discussed below.

b) Cross-validation bandwidth selection for the BRNP estimator

As before, the cross-validation method selects the bandwidth that minimises the cross-validation function. Denote this bandwidth by $\hat{h}_{CV}$. A general algorithm is now presented for determining the BRNP estimator using the cross-validation bandwidth selection.

Algorithm 4.3. The BRNP estimator using cross-validation bandwidth selection

1. Let $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$ be the given sample.

2. Determine a grid of possible bandwidth values, i.e. H, for the bandwidth h.

3. Calculate the leave-one-out estimator for a given bandwidth h, i.e.

$$\hat{m}_{\text{BRNP},(k)}(X_k) = \frac{n^{-2}c(h)\sum_{\substack{i=1\\i\neq k}}^{n}\sum_{\substack{j=1\\j\neq k}}^{n} v_j Y_i Y_j K_{\sqrt{3}h}(X_i - X_j)K_{\sqrt{3}h}(X_i - X_k)K_{\sqrt{3}h}(X_j - X_k)}{n^{-2}c(h)\sum_{\substack{i=1\\i\neq k}}^{n}\sum_{\substack{j=1\\j\neq k}}^{n} w_j\sqrt{3}h(X_i - X_j)K_{\sqrt{3}h}(X_i - X_k)K_{\sqrt{3}h}(X_j - X_k)} \tag{4.52}$$

   where K_h is the kernel of interest, with scale factor h, $c(h)$ is defined in (4.44) and $k = 1, 2, \ldots, n$.

4. Determine the cross-validation function for the particular bandwidth h, i.e.

$$CV(h) = \frac{1}{n}\sum_{k=1}^{n}[Y_k - \hat{m}_{(k)}(X_k)]^2,$$

   where $\hat{m}_{(k)}(\cdot)$ is the proposed regression estimate obtained by leaving out the k^{th} observation.

5. Repeat steps 3 and 4 for the complete grid of bandwidths, to obtain $CV(h)$ for each h value.

6. Select $\hat{h}_{CV}$ as the bandwidth that minimises $CV(h)$ in step 4.

7. Calculate the BRNP estimator using the selected bandwidth $\hat{h}_{CV}$, i.e.

$$\hat{m}_{\text{BRNP}}(x) = \frac{n^{-2}c(\hat{h}_{CV})\sum_i\sum_j v_j Y_i Y_j K_{\sqrt{3}\hat{h}_{CV}}(X_i - X_j)K_{\sqrt{3}\hat{h}_{CV}}(X_i - x)K_{\sqrt{3}\hat{h}_{CV}}(X_j - x)}{n^{-2}c(\hat{h}_{CV})\sum_i\sum_j w_j\sqrt{3}\hat{h}_{CV}(X_i - X_j)K_{\sqrt{3}\hat{h}_{CV}}(X_i - x)K_{\sqrt{3}\hat{h}_{CV}}(X_j - x)} \tag{4.53}$$

   where $K_{\hat{h}_{CV}}$ is the kernel of interest, with scale factor $\hat{h}_{CV}$.

Remark

- Note that the denominator of each of these three estimators can become very small. To overcome problems that arise due to division by very small values, a correction factor will be used in all the calculations: whenever the denominator turns out to be less than 0.001, the value n^{-2} is added to the denominator.

- In the rest of the chapter $\hat{m}(x)$ will refer to any of the three above defined nonparametric regression estimators, whichever of them will be of interest in the discussions to follow.

4.3 Improvement methods

This section refers to Marzio and Taylor (2008) and Hall and Robinson (2009) regarding the so-called improvement methods. In this study three improvement methods are explored for the three regression estimators. Remarks on each method are given below.

In Chapter 3 the basic bagging and bragging methodology was presented according to Swanepoel (1988, 1990) and Breiman (1996a). Swanepoel (1988, 1990) proved satisfactory asymptotic properties of these procedures. Furthermore, Hall and Robinson (2009) showed that bagging can be used to reduce the variability existing in cross-validation bandwidth selection in kernel regression estimation. Hall and Robinson (2009) suggested that bagging can be applied to the cross-validation bandwidth selection method in at least two ways, namely bagging the cross-validation function and bagging $\hat{h}_{CV}$. In this study bagging is applied to the cross-validation bandwidth selection method and plug-in bandwidth selection methods. Three ways are proposed in which the bagging of $\hat{h}$ can be approached. These procedures are discussed more informatively in Section 4.3.1 below.

Bragging involves that the median of bootstrap versions of the required estimator is calculated, rather than the average as in bagging (Swanepoel, 1990, Bühlmann 2003). Further details regarding the bragging procedure are discussed in Section 4.3.3.

Boshoff (2009) also applied various bagging and bragging methods to the cross-validation bandwidth selection method. *However, the combined bagging and plug-in methods as well as combined bragging and plug-in methods are new contributions to nonparametric regression results.* In Chapter 3 the development of the boosting method was discussed. In Section 3.7.2 it was shown that the L_2 boosting algorithm is very useful in the regression environment. Marzio and Taylor (2008) presented a multi-step kernel regression smoother generated by learning the Nadaraya-Watson estimator. Marzio and Taylor showed that the bias decreases exponentially fast towards zero, while the variance increases exponentially slow towards σ^2. In the present study, boosting as an improvement method regarding bias, will be applied to the Nadaraya-Watson smoother, local linear smoother and Yao's smoother. The boosting algorithm used in this study is presented in Section 4.3.5.

4.3.1 Bagging

As mentioned before, the basic principles of bootstrap aggregating (bagging) were presented according to Swanepoel (1988, 1990) in a functional context and by Breiman (1996a) from a machine learning viewpoint. One of the popular attractions of the cross-validation bandwidth selection method is its applicability to a wide variety of estimator types. However, the cross-validation method suffers from relatively high variability and this compromises the performance of the estimators in which cross-validation selected smoothing parameters are used. Hall and Robinson (2009) showed that this variability can be reduced by employing bootstrap aggregation or bagging. In Chapter 3, a discussion of the bagging methodology is stated. In Section 3.5.2, the application of the bagging procedure to the cross-validation bandwidth selection method is considered. In the present section

more refined algorithms are provided to show how to implement the bagging method in the simulation studies. Algorithms 4.4 to 4.6 show how to implement bagging when the cross-validation bandwidth selection method is employed. In Algorithms 4.7 to 4.9 it is clarified how the bagging algorithms changes when the plug-in bandwidth selection method is used.

Regarding the bagging algorithms the following points are of importance for the simulation studies.

1. **Bagging method**

   Hall and Robinson (2009) mentioned at least two approaches to bagged cross-validation when determining the smoothing parameter. By recalling Section 3.5.2, it is repeated that these approaches are based on *bagging* the cross-validation function and on *bagging* h_{CV}, respectively.

   - In the first approach bootstrap replications of the cross-validation function are combined to obtain CV_{bag}. The bandwidth, $\hat{h}_{bag}$, is then chosen to be the value of h that minimises CV_{bag}.
   - In the second approach, for each bootstrap sample, a bandwidth $\hat{h}^*_{CV}$ is obtained to minimise $CV^*(h)$ defined in (3.10). The estimated bandwidth $\hat{h}_{bag}$ is now defined as the average value of $\hat{h}^*_{CV}$ over all the bootstrap samples.

   Hall and Robinson (2009) showed and concluded that the second method generally leads to greater improvements in the variance and mean squared error than the first method. Therefore only the *second method* is pursued in this study. Keeping in mind the results obtained from Boshoff (2009), for this estimation method, *at least three different possibilities* are worth while investigating, in order to perform the bootstrap aggregation. Throughout these methods are referred to as:

   - Bag 1
   - Bag 2
   - Bag 3

   The three bootstrap aggregation methods are applied to two smoothing parameter selection methods, i.e. leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method. A very brief and general description of the three methods is given below.

   - The first approach roughly involves the following: generate bootstrap samples from the original dataset, *determine the smoothing parameter for each bootstrap sample and calculate the mean of these bootstrap smoothing parameters and finally applying the resulting mean smoothing parameter value to the original sample to obtain* $\hat{m}(x)$.
   - In the second approach *the mean smoothing parameter value over the bootstrap samples is applied to each of the bootstrap samples to obtain bootstrap regression estimates. The mean of the bootstrap regression estimates are then calculated to obtain* $\hat{m}(x)$.
   - The third approach involves *determining the smoothing parameter for each bootstrap sample and then applying each bootstrap sample's smoothing parameter to that particular bootstrap sample to obtain*

bootstrap regression estimates. The mean of the bootstrap regression estimates are then calculated to obtain $\hat{m}(x)$. More detailed steps follow in the algorithms below.

In the present study the three bagging methods are applied, based on the leave-one-out cross-validation bandwidth selection method, as well as the plug-in bandwidth selection method. In the case of the *cross-validation bandwidth selection method*, the *Nadaraya-Watson smoother* and the *BRNP smoother* are calculated and improved via the three bagging methods. The *Nadaraya-Watson smoother* and a *local linear smoother* are determined and improved via bagging methods using *plug-in bandwidth selection methods*. The reason for this is that the BRNP method is already very time consuming and complicated. Applying plug-in methods for the BRNP will involve a later study on itself in order to develop rules-of-thumb similar to those rules discussed in Fan and Gijbels (1996:111). Also, the local linear theory is based only on the plug-in method of Fan and Gijbels (1996:111). Later studies may involve cross-validation methods for selecting bandwidths.

The algorithms to determine the final estimators for these three bagging methods are described in Algorithms 4.4 to 4.6 and 4.7 to 4.9. Results and conclusions about these three methods are presented in the next chapter and in Appendix A to D.

2. Resampling scheme

- In all algorithms where the bandwidth is chosen by employing the leave-one-out cross-validation method, *resampling is done* ***without*** *replacement*. Hall and Robinson (2009) notes that one should resample without replacement to avoid difficulties caused by ties, which can lead to the cross-validation method breaking down.
- In all algorithms where the bandwidth is chosen by employing the plug-in bandwidth selection method, *resampling is done* ***with*** *replacement*.

3. Resample size

- In the cases where the leave-one-out cross-validation bandwidth selection method is used, resamples of size $m = [an], 0 < a < 1$, are drawn, where $[y]$ denotes the largest integer value smaller than or equal to y, for some suitable a.
- In all algorithms where the bandwidth is chosen by employing the plug-in bandwidth selection method, resamples equal to the original sample size n are used.

4. Rescaling of the bandwidth

- In Algorithm Bag1$_{CV}$ the final bandwidth estimator needs to be rescaled. The bagged bandwidth is based on a smaller sample size, i.e. m rather than n, and is then applied to determine $\hat{m}(x)$ using the original n data points. In Algorithms Bag2$_{CV}$ and Bag3$_{CV}$ the rescaling step is not necessary, since the bootstrap bandwidths are used to obtain bootstrap estimates $\hat{m}(x)$ when using the bootstrap resamples with m data points.

- When plug-in bandwidth selection methods are used, no rescaling of the bandwidth is needed, since the bagged bandwidth is based on the same sample size as the original sample size, n.

4.3.2 Corresponding bagging algorithms for methods Bag1, Bag2 and Bag3

In this section the bagging algorithms used in this study are presented. The first three algorithms shows how the three bagging methods are implemented when the cross-validation bandwidth selection method is used. These methods were also used in Boshoff (2009) who proposed a basic study using only leave-one-out cross-validation bandwidth selection and the Nadaraya-Watson estimator with the basic improvement methods. Algorithms 4.4 to 4.6 are similar to the algorithms used by Boshoff (2009).

In the present study a much broader study is attempted involving a variety of estimation methods, the plug-in method and two kernels. The last three Algorithms 4.7 to 4.9 show how to utilise the bagging methods when the bandwidth is chosen by plug-in methods and are new contributions.

Algorithm 4.4. Bag1 using cross-validation bandwidth selection ($\mathbf{Bag1_{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.
3. Determine, over an appropriate grid of h values, a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample $\mathbf{S}_m^*$ to obtain $CV^*(h)$. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.
4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the cross-validation bandwidth $\hat{h}_{CV}^*(1)$, $\hat{h}_{CV}^*(2), \ldots, \hat{h}_{CV}^*(B)$.
5. Calculate the average of the bootstrap replications of the bandwidth
$$\hat{h}_{bag_m} = \frac{1}{B}\sum_{b=1}^{B} \hat{h}_{CV}^*(b).$$
6. Rescale the bandwidth in step 5: $\hat{h}_{bag} = (\hat{h}_{bag_m})\left(\frac{m}{n}\right)^{\frac{1}{5}}$.
7. Fit the estimator $\hat{m}(x)$ using the original sample $\mathbf{S}_n$ and the bandwidth $\hat{h}_{bag}$.

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.5. Bag2 using cross-validation bandwidth selection ($\mathbf{Bag2_{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.

3. Determine, over an appropriate grid of h values, a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample $\mathbf{S}_m^*$ to obtain $CV^*(h)$. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.

4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the cross-validation bandwidth $\hat{h}_{CV}^*(1)$, $\hat{h}_{CV}^*(2), \ldots, \hat{h}_{CV}^*(B)$.

5. Calculate the average of the bootstrap replications of the bandwidth

$$\hat{h}_{bag_m} = \frac{1}{B}\sum_{b=1}^{B} \hat{h}_{CV}^*(b).$$

6. Use each bootstrap sample $\mathbf{S}_m^*$ and the aggregated bandwidth $\hat{h}_{bag_m}$ and fit the estimator $\hat{m}(x)$ to obtain bootstrap versions of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.

7. Calculate the average of the bootstrap replications of the regression estimate

$$\hat{m}^*(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.6. Bag3 using cross-validation bandwidth selection ($\mathbf{Bag3_{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.

2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.

3. Determine, over an appropriate grid of h values, a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample S_m^* to obtain $CV^*(h)$. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.

4. Use the bootstrap sample $\mathbf{S}_m^*$ and corresponding bandwidth $\hat{h}_{CV}^*$ and fit the estimator $\hat{m}$ to obtain a bootstrap version of the regression estimate $\hat{m}^*(x)$.

5. Repeat steps 2, 3 and 4 B times to obtain bootstrap replications of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.

6. Calculate the average of the bootstrap replications of the regression estimate

$$\hat{m}^*(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.7. Bag1 using plug-in bandwidth selection ($\mathbf{Bag1_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample n data pairs *with replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.
3. Calculate a bootstrap version of the plug-in bandwidth in (4.22) for the Nadaraya-Watson estimator or (4.41) for the local linear estimator by using the bootstrap sample $\mathbf{S}_n^*$.
4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the plug-in bandwidth $\hat{h}^*_{plug}(1)$, $\hat{h}^*_{plug}(2), \ldots, \hat{h}^*_{plug}(B)$.
5. Calculate the average of the bootstrap replications of the bandwidth
$$\hat{h}_{bag_n} = \frac{1}{B}\sum_{b=1}^{B} \hat{h}^*_{plug}(b).$$
6. Fit the estimator $\hat{m}(x)$ using the original sample $\mathbf{S}_n$ and the bandwidth $\hat{h}_{bag_n}$.

Here $\hat{m}(x)$ refers to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

Algorithm 4.8. Bag2 using plug-in bandwidth selection ($\mathbf{Bag2_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample n data pairs *with replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.
3. Calculate a bootstrap version of the plug-in bandwidth in (4.22) for the Nadaraya-Watson estimator or (4.41) for the local linear estimator by using the bootstrap sample $\mathbf{S}_n^*$.
4. Repeat steps 2 and 3 B times to obtain bootstrap replications of plug-in bandwidth $\hat{h}^*_{plug}(1)$, $\hat{h}^*_{plug}(2), \ldots, \hat{h}^*_{plug}(B)$.
5. Calculate the average of the bootstrap replications of the bandwidth
$$\hat{h}_{bag_n} = \frac{1}{B}\sum_{b=1}^{B} \hat{h}^*_{plug}(b).$$
6. Use each bootstrap sample $\mathbf{S}_n^*$ and the aggregated bandwidth $\hat{h}_{bag_n}$ and fit the estimator $\hat{m}(x)$ to obtain bootstrap versions of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.
7. Calculate the average of the bootstrap replications of the regression estimate
$$\hat{m}^*(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ refers to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

Algorithm 4.9. Bag3 using plug-in bandwidth selection ($\mathbf{Bag3_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.

2. Randomly sample n data pairs *with replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.

3. Calculate a bootstrap version of the plug-in bandwidth in (4.22) for the Nadaraya-Watson estimator or (4.41) for the local linear estimator by using the bootstrap sample $\mathbf{S}_n^*$.

4. Use the bootstrap sample $\mathbf{S}_n^*$ and corresponding bandwidth $\hat{h}_{plug}^*$ and fit the estimator $\hat{m}$ to obtain a bootstrap version of the regression estimate $\hat{m}^*(x)$.

5. Repeat steps 2, 3 and 4 B times to obtain bootstrap replications of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.

6. Calculate the average of the bootstrap replications of the regression estimate

$$\hat{m}^*(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ refers to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

4.3.3 Bragging

Swanepoel (1990) and Bühlmann (2003) suggested that the median of bootstrap versions of an estimator can be computed instead of the arithmetic mean. *This procedure is referred to as bragging.* In this way, a more robust estimator is obtained. In this section algorithms are provided to show how to implement the bragging method in the simulation studies. Algorithms 4.10 to 4.12 show how to implement bragging when the cross-validation bandwidth selection method is employed. In Algorithms 4.13 to 4.15 it is clarified how the bragging algorithms changes when the plug-in bandwidth selection method is used instead of leave-one-out cross-validation methods. Regarding the bragging algorithms the following points are of importance for the simulation studies.

1. **Bragging method**

   Similar to bagging, there are two ways in which bragging can be applied to bandwidth selection. By recalling Section 4.3.1 we repeat that these approaches are based on *bragging* the cross-validation function and on *bragging* h_{CV}, respectively.

   - In the first approach bootstrap replications of the cross-validation function are combined to obtain CV_{brag}. The bandwidth, $\hat{h}_{brag}$, is then chosen to be the value of h that minimises CV_{brag}.
   - In the second approach, for each bootstrap sample, a bandwidth $\hat{h}_{CV}^*$ is obtained to minimise $CV^*(h)$ defined in (3.10). The estimated bandwidth $\hat{h}_{brag}$ is now defined as the average value of $\hat{h}_{CV}^*$ over all the bootstrap samples.

Hall and Robinson (2009) showed and concluded that the second method generally leads to greater improvements in the variance and mean squared error than the first method. Therefore only the *second method* is pursued in this study. For this estimation method, *at least three different possibilities* are worth while investigating, in order to perform the bootstrap robust aggregation. Throughout theses methods are referred to as:

- Brag 1
- Brag 2
- Brag 3

The three bragging methods are applied to two smoothing parameter selection methods, i.e. leave-one-out cross-validation bandwidth selection method and the plug-in bandwidth selection method. A very brief and general description of the three methods is given below.

- The first approach roughly involves the following: generate bootstrap samples from the original dataset, determine the smoothing parameter for each bootstrap sample and calculate the median of these bootstrap smoothing parameters and finally applying the resulting median smoothing parameter value to the original sample to obtain $\hat{m}(x)$.
- In the second approach the median smoothing parameter value over the bootstrap samples is applied to each of the bootstrap samples to obtain bootstrap regression estimates. The median of the bootstrap regression estimates are then calculated to obtain $\hat{m}(x)$.
- The third approach involves determining the smoothing parameter for each bootstrap sample and then applying each bootstrap sample's smoothing parameter to that particular bootstrap sample to obtain bootstrap regression estimates. The median of the bootstrap regression estimates are then calculated to obtain $\hat{m}(x)$. More detailed steps follow in the algorithms below.

In the present study the three bragging methods are applied based on the leave-one-out cross-validation bandwidth selection method as well as the plug-in bandwidth selection method. In the case of the cross-validation bandwidth selection method, the Nadaraya-Watson smoother and the BRNP smoother are calculated and improved via the three bragging methods. The Nadaraya-Watson smoother and a local linear smoother are determined and improved via bragging methods using plug-in bandwidth selection methods.

The algorithms to determine the final estimators for these three bragging methods are described in Algorithms 4.10 to 4.12 and 4.13 to 4.15. Results and conclusions about these three methods are presented in the next chapter and in Appendix A to D.

2. Resampling scheme

- In all algorithms where the bandwidth is chosen by employing the leave-one-out cross-validation method, *resampling is done* ***without*** *replacement.* Hall and Robinson (2009) notes that one should

resample without replacement to avoid difficulties caused by ties, which can lead to the cross-validation method breaking down.

- In all algorithms where the bandwidth is chosen by employing the plug-in bandwidth selection method, *resampling is done **with** replacement.*

3. Resample size

- In the cases where the leave-one-out cross-validation bandwidth selection method is used, resamples of size $m = [an], 0 < a < 1$, are drawn, where $[y]$ denotes the largest integer value smaller than or equal to y, for some suitable a.
- In all algorithms where the bandwidth is chosen by employing the plug-in bandwidth selection method, resamples equal to the original sample size n are used.

4. Rescaling of the bandwidth

- In Algorithm Brag1$_{CV}$ the final bandwidth estimator needs to be rescaled. The bragged bandwidth is based on a smaller sample size, i.e. m rather than n, and is then applied to determine $\hat{m}(x)$ using the original n data points. In Algorithms Brag2$_{CV}$ and Brag3$_{CV}$ the rescaling step is not necessary, since the bootstrap bandwidths are used to obtain bootstrap estimates $\hat{m}(x)$ when using the bootstrap resamples with m data points.
- When plug-in bandwidth selection methods are used, no rescaling of the bandwidth is needed, since the bragged bandwidth is based on the same sample size as the original sample size, n.

4.3.4 Corresponding bragging algorithms for Brag1, Brag2 and Brag3

In this section the bragging algorithms used in this study are presented. The first three algorithms show how the three bragging methods are implemented when the cross-validation bandwidth selection method is used. These methods were also used in Boshoff (2009) who proposed a basic study using only leave-one-out cross-validation bandwidth selection and the Nadaraya-Watson estimator with the basic improvement methods. Therefore, Algorithms 4.11 to 4.13 are similar to those of Boshoff (2009).

In the present study a much broader study is attempted. The last three Algorithms 4.14 to 4.16 show how to utilise the bragging methods when the bandwidth is chosen by plug-in methods and are new contributions.

Algorithm 4.10. Brag1 using cross-validation bandwidth selection (Brag1$_{\mathbf{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.

3. Calculate a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample $\mathbf{S}_m^*$ to obtain $CV^*(h)$ over a grid of h-values. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.

4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the cross-validation bandwidth $\hat{h}_{CV}^*(1)$, $\hat{h}_{CV}^*(2), \ldots, \hat{h}_{CV}^*(B)$.

5. Calculate the median of the bootstrap replications of the bandwidth

$$\hat{h}_{brag_m} = \text{median}_{1 \leq b \leq B} \hat{h}_{CV}^*(b).$$

6. Rescale the bandwidth in step 5: $\hat{h}_{bag} = (\hat{h}_{bag_m})(\frac{m}{n})^{\frac{1}{5}}$.

7. Fit the estimator $\hat{m}(x)$ using the original sample $\mathbf{S}_n$ and the bandwidth $\hat{h}_{brag}$.

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.11. Brag2 using cross-validation bandwidth selection (Brag2$_{\mathbf{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.

2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.

3. Calculate a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample $\mathbf{S}_m^*$ to obtain $CV^*(h)$ over a grid of h-values. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.

4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the cross-validation bandwidth $\hat{h}_{CV}^*(1)$, $\hat{h}_{CV}^*(2), \ldots, \hat{h}_{CV}^*(B)$.

5. Calculate the median of the bootstrap replications of the bandwidth

$$\hat{h}_{brag_m} = \text{median}_{1 \leq b \leq B} \hat{h}_{CV}^*(b).$$

6. Use each bootstrap sample $\mathbf{S}_m^*$ and median bandwidth $\hat{h}_{bag_m}$ and fit the estimator $\hat{m}(x)$ to obtain bootstrap versions of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.

7. Calculate the median of the bootstrap replications of the regression estimate

$$\hat{m}^*(x) = \text{median}_{1 \leq b \leq B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.12. Brag3 using cross-validation bandwidth selection ($\mathbf{Brag3_{CV}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample m data pairs *without replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $S_m^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_m^*, Y_m^*)\}$.
3. Calculate a bootstrap version of the leave-one-out cross-validation function as it is defined in (3.10) by using the bootstrap sample $\mathbf{S}_m^*$ to obtain $CV^*(h)$ over a grid of h-values. Obtain the value $\hat{h}_{CV}^*$ that minimises $CV^*(h)$.
4. Use the bootstrap sample $\mathbf{S}_m^*$ and corresponding bandwidth $\hat{h}_{CV}^*$ and fit the estimator $\hat{m}$ to obtain a bootstrap version of the regression estimate $\hat{m}^*(x)$.
5. Repeat steps 2, 3 and 4 B times to obtain bootstrap replications of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.
6. Calculate the median of the bootstrap replications of the regression estimate

$$\hat{m}^*(x) = \text{median}_{1 \leq b \leq B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the BRNP estimator defined in (4.46).

Algorithm 4.13. Brag1 using plug-in bandwidth selection ($\mathbf{Brag1_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample n data pairs *with replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.
3. Calculate a bootstrap version of the plug-in bandwidth in (4.22) for the Nadaraya-Watson estimator or (4.41) for the local linear estimator by using the bootstrap sample $\mathbf{S}_n^*$.
4. Repeat steps 2 and 3 B times to obtain bootstrap replications of the plug-in bandwidth $\hat{h}_{plug}^*(1)$, $\hat{h}_{plug}^*(2), \ldots, \hat{h}_{plug}^*(B)$.
5. Calculate the median of the bootstrap replications of the bandwidth

$$\hat{h}_{brag_n} = \text{median}_{1 \leq b \leq B} \hat{h}_{plug}^*(b).$$

6. Fit the estimator $\hat{m}(x)$ using the original sample $\mathbf{S}_n$ and the bandwidth $\hat{h}_{bag_n}$.

Here $\hat{m}(x)$ refers to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

Algorithm 4.14. Brag2 using plug-in bandwidth selection ($\mathbf{Brag2_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample n data pairs *with replacement* from S_n to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.
3. Calculate a bootstrap version of the plug-in bandwidth in (4.22) for the Nadaraya-Watson estimator or (4.41) for the local linear estimator by using the bootstrap sample $\mathbf{S}_n^*$.
4. Repeat steps 2 and 3 B times to obtain bootstrap replications of plug-in bandwidth $\hat{h}_{plug}^*(1), \ldots, \hat{h}_{plug}^*(B)$.
5. Calculate the median of the bootstrap replications of the bandwidth
$$\hat{h}_{bag_n} = \text{median}_{1 \le b \le B} \hat{h}_{plug}^*(b).$$
6. Use each bootstrap sample $\mathbf{S}_n^*$ and the median bandwidth $\hat{h}_{bag_n}$ and fit the estimator $\hat{m}(x)$ to obtain bootstrap versions of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.
7. Calculate the median of the bootstrap replications of the regression estimate
$$\hat{m}^*(x) = \text{median}_{1 \le b \le B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

Algorithm 4.15. Brag3 using plug-in bandwidth selection ($\mathbf{Brag3_{plug}}$)

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be the given sample.
2. Randomly sample n data pairs *with replacement* from $\mathbf{S}_n$ to obtain a bootstrap sample $\mathbf{S}_n^* = \{(X_1^*, Y_1^*), (X_2^*, Y_2^*), \ldots, (X_n^*, Y_n^*)\}$.
3. Calculate a bootstrap version of plug-in bandwidth in (4.8) by using the bootstrap sample $\mathbf{S}_n^*$.
4. Use the bootstrap sample $\mathbf{S}_n^*$ and corresponding bandwidth $\hat{h}_{plug}^*$ and fit the estimator $\hat{m}$ to obtain a bootstrap version of the regression estimate $\hat{m}^*(x)$.
5. Repeat steps 2, 3 and 4 B times to obtain bootstrap replications of the regression estimate $\hat{m}^*(x)^{(1)}, \hat{m}^*(x)^{(2)}, \ldots, \hat{m}^*(x)^{(B)}$.
6. Calculate the median of the bootstrap replications of the regression estimate
$$\hat{m}^*(x) = \text{median}_{1 \le b \le B} \hat{m}^*(x)^{(b)}.$$

Here $\hat{m}(x)$ can refer to the Nadaraya-Watson estimator defined in (4.3) or the local linear estimator in defined (4.28).

4.3.5 Boosting

In Chapter 3 the boosting method, i.e. the development, applications and features of the method was discussed. According to Bühlmann and Hothorn (2007:483) the L_2-boosting algorithm is the simplest and most instructive boosting algorithm. This algorithm is especially very useful in the regression environment. For the sake of completeness the algorithm stated in Section 3.7.2 is repeated here, specifying the choice of parameters in the algorithm where relevant.

Algorithm 4.16. **L_2-boosting**

1. Let the sample $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ be given.
2. Fit an initial learner, $\hat{m}_0(x)$ by fitting one of the regression estimators defined in Section 4.2 to the original data: $\hat{m}_0(x) \equiv \hat{m}(x)$, using some bandwidth, h. Set $t = 0$.
3. Let $t = t + 1$. Compute the residuals,
$$U_i = Y_i - \hat{m}_{t-1}(X_i), i = 1, \ldots, n.$$
4. Fit the residual vector $U_1, U_2, \ldots, U_n$ to $X_1, X_2, \ldots, X_n$ by one of the regression estimators defined in Section 4.2. Call this fit $\tilde{m}_t(x)$.
5. Update the previous estimator $\tilde{m}_{t-1}(x)$
$$\hat{m}_t(x) = \hat{m}_{t-1}(x) + \tilde{m}_t(x).$$
6. Repeat steps 3 to 5 until $t = T$, where T is some stopping value for the number of boosting iterations.

In this algorithm $\hat{m}(x)$ refers to the Nadaraya-Watson smoother, local linear smoother or Yao's smoother. The bandwidth can be chosen by the cross-validation method or plug-in bandwidth selection methods.

4.3.6 Measuring the discrepancies

When different regression estimation methods are explored, one would strive towards evaluating the performance of each method, to assess how close the smooth is to the true curve. In Section 1.5 different discrepancy measures were discussed. In this study the global mean integrated squared error (MISE) is used as a measure the performance of each estimator. Recall from Section 1.5, that a well-known feature of the MSE is that it can be written as the sum of the variance and the squared bias of the estimator:

$$\begin{aligned} \text{MSE}[\hat{m}(x)] &= E[\hat{m}(x) - m(x)]^2 \\ &= E[(\hat{m}(x) - E[\hat{m}(x)])^2] + [E[\hat{m}(x)] - m(x)]^2 \\ &= \text{Var}[\hat{m}(x)] + \text{Bias}^2[\hat{m}(x)]. \end{aligned} \tag{4.54}$$

Also, integrating over all x-values yields

$$\text{MISE} = \int \text{Var}[\hat{m}(x)]dx + \int \text{Bias}^2[\hat{m}(x)]dx, \tag{4.55}$$

which is the preferred discrepancy measure since it is a global measure and covers the entire population by taking the expected squared differences between the estimated values and their corresponding true values for each fixed point x by means of the MSE and it covers the whole range of x-values by integrating over the MSE. The question of concern is how to choose the parameters embedded in $\hat{m}(x)$ so that the MISE is minimised. The variance and squared bias in (4.55) are approximated by averaging over a suitable number of Monte Carlo replications of finite independent samples drawn randomly from the underlying distribution. These calculations are repeated for a fixed grid of chosen x-values. Numerical integration is then used to integrate over the entire range of x-values to obtain approximations to the global measures. The MISE, integrated variance and integrated squared bias values are reported and compared for the various estimation procedures that are under investigation in this study.

4.4 Matters surrounding the main algorithm

To enable the MISE calculations, the main algorithm of this study is now presented. This main algorithm includes calculating the approximated MISE value, for each of the specific smoothing procedures developed to determine the regression estimate $\hat{m}(x)$ for $m(x)$. The main algorithm is therefore applied to each of the different smoothing procedures, and the approximated MISE, integrated variance and integrated squared bias values, resulting from the main algorithm, are then noted and compared. The remarks below should first be noted since it defines the simulation setup.

4.4.1 Simulation setup and remarks regarding the main algorithm

The remarks below are necessary to assist in understanding the structure of the main algorithm. The main algorithm must be studied, keeping in mind the goals of this study, which are captured in remarks n) and o) below.

a) The underlying regression function $m(x)$:

This main algorithm is performed to determine the behaviour of all procedures of interest in estimating three specific models, which are used frequently in the literature, and which are well- known for its difficulty to be estimated. These models are defined in Table 4.1 below and represented graphically in Figure 4.1.

Model	$m(x)$
1	$x + 2\exp(-16x^2)$
2	$\sin(2x) + 2\exp(-16x^2)$
3	$0.3\exp\{-4(x+1)^2\} + 0.7\exp\{-16(x-1)^2\}$

Table 4.1: The underlying regression function $m(x)$

b) Underlying distribution of the covariate data:

Two distributions are considered for the covariate data, i.e. X-values are drawn from the $U(-2, 2)$ and

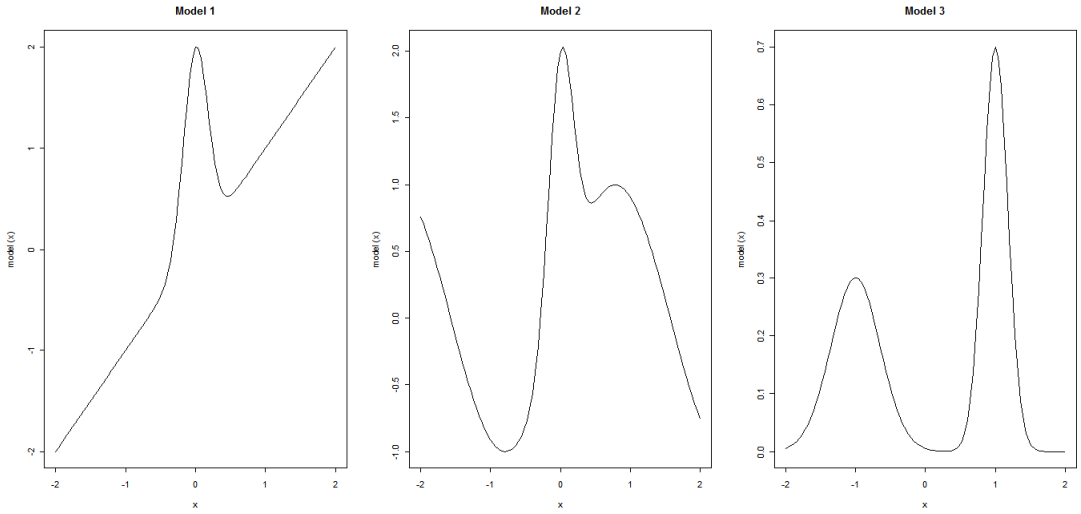

Figure 4.1: The underlying regression function $m(x)$

$N(0,1)$ distributions respectively. This is also popular choices throughout the literature, for example Fan and Gijbels (1992).

c) Construction of the response data:

Response data Y are constructed from $Y = m(X) + \sigma\varepsilon$, where:

- ε's are random errors drawn from the $N(0,1)$ distribution in this study.
- m is chosen as one of the first three simulation models of Fan and Gijbels (1996:111), listed in Table 4.1.
- σ is chosen as 0.2, 0.6 and 1.0 for models 1 and 2 and as 0.1, 0.3 and 0.5 for model 3.

d) Sample size:

Marzio and Taylor (2008) used samples of sizes $n = 50$, $n = 100$ and $n = 200$ in their study. The same sample size values are used in the present study.

e) Kernel functions:

Two kernels are used for executing the procedures, for each model defined above. Various other kernels are possible but Marzio and Taylor (2008) showed that boosting works properly only if the kernel yields strictly positive characteristic roots. This is the case for the $N(0,1)$ and triangular kernels, but not for many other popular kernels, such as the Epanechnikov, biweight and triweight kernels. Therefore the present study is restricted to the $N(0,1)$ and triangular kernels. Future studies will include more kernels when boosting is not a research issue. The two kernels are defined in the Table 4.2 below and presented graphically in Figure 4.2.

The above points are illustrated in Figures 4.3 to 4.5.

Kernel	K(u)
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-u^2/2}$
Triangular	$(1-\|u\|)I(\|u\| \leq 1)$

Table 4.2: Kernels used in the present study

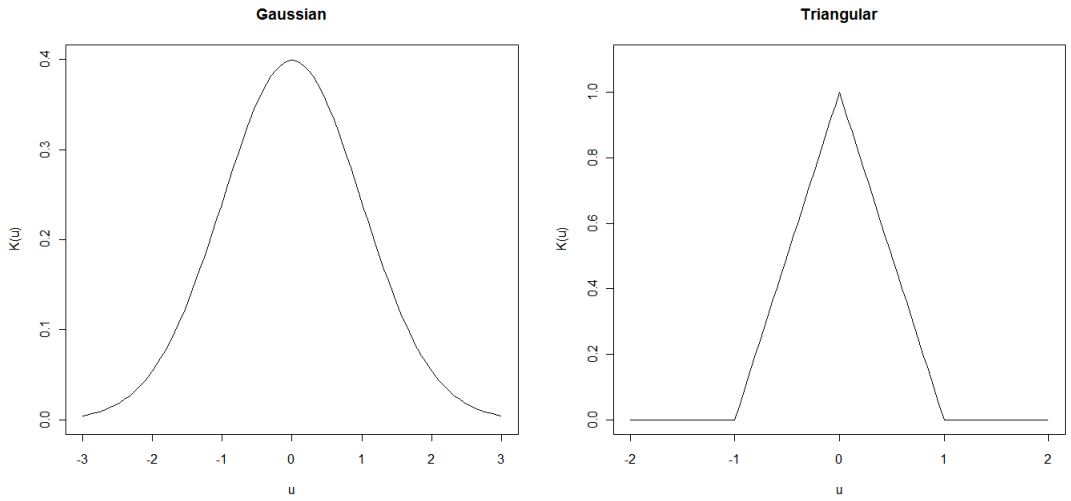

Figure 4.2: Kernel functions used in the present study

f) Cross-validation bandwidth selection:

In order to utilise the cross-validation bandwidth selection algorithm a grid of possible bandwidth values needs to be defined. In the present study a grid between 0.01 and 2 is constructed in increments of 0.01. In a preliminary study it was confirmed that this specific choice of the grid is effective, in the sense that chosen optimal h-value was not chosen as the boundary value of the grid. A larger grid search will result in more computation time.

g) Plug-in bandwidth selection:

The plug-in bandwidth selection algorithms contain two unknown values that needs to be defined, the order of the polynomial regression function and the weight function used.

The weight function depends on the chosen underlying distribution of the covariate data. If $X \sim U(-2,2)$, the weight function is defined as $w_0(x) = I(-1.8 \leq x \leq 1.8)$. This is the same weight function used by Fan and Gijbels (1996:112). If $X \sim N(0,1)$, the weight function is defined as $w_0(x) = I(-1.8 \leq x \leq 1.8)\phi(x)$.

For sample sizes $n = 100$ and $n = 200$ a 12^{th} order polynomial regression function is used and for $n = 50$ a polynomial regression function of order five is used. Polynomials of a larger order results in exact collinearity and then regression coefficients are not estimable.

h) Number of bootstrap samples:

Hall and Robinson (2009) used 50 bootstrap samples in their study. This choice is used as guideline in the present and 50 bootstrap samples are drawn for the application of the bagging and bragging methods throughout.

i) Number of data points in the bootstrap samples:

In the bagging and bragging algorithms presented in Chapter 3 it is indicated that bootstrap resamples of size $m = [an], 0 < a < 1$, are drawn, where $[y]$ denotes the largest integer value smaller than or equal to y. Bühlmann (2004:884) suggests that $m = n/2$ i.e. $a = 0.5$, is often a good choice in practice. In the present study bootstrap samples of size $m = n/2$ are used.

j) Bootstrap resampling scheme:

- In all algorithms where the bandwidth is chosen by employing the leave-one-out cross-validation method, resampling is done *without* replacement. Hall and Robinson (2009) notes that one should resample without replacement to avoid difficulties caused by ties, which can lead to the cross-validation method breaking down.
- In all algorithms where the bandwidth is chosen by employing the plug-in bandwidth selection method, resampling is done *with* replacement.

k) Number of boosting iterations:

Marzio and Taylor (2008) allowed a maximum of six boosting iterations. However, the improvement regarding the minimisation of the MISE for these additional boosting iterations was very small. Considering the extra computation time that would be needed for extra boosting iterations and the relatively small reduction in the corresponding MISE-values, the present study allows for a maximum of one boosting iteration.

l) Grid of x-values:

To determine the approximated MISE, as described in the main algorithm, the pointwise MSE has to be calculated for a grid of x-values. A fixed grid is constructed between the values -2 and +2 and a 100 grid points are used.

m) Number of Monte Carlo iterations:

Both Marzio and Taylor (2008) and Hall and Robinson (2009) used $MC = 200$ in their simulation studies. This value is used as a guideline in the present study and 200 Monte Carlo samples ($MC = 200$) are generated for each of the sample sizes: $n = 50$, $n = 100$ and $n = 200$.

n) Estimation methods:

Keeping the goals set for this study in mind, note that for a particular combination of model choice, underlying distribution of the predictor variable X and kernel choice, the methods of determining regression estimators for the models in Table 4.1 are executed. These methods respectively are:

- The Nadaraya-Watson with cross-validation bandwidth selection (Algorithms 4.0).

- The Nadaraya-Watson estimator with plug-in bandwidth selection (Algorithms 4.1).
- The local linear estimator with plug-in bandwidth selection (Algorithm 4.2).
- The BRNP estimator with cross-validation bandwidth selection (Algorithm 4.3).

These methods are reflected in step 3 of the main algorithm.

o) Improvement methods:

For each model and each estimation method mentioned above, the various improvement methods discussed in Section 4.3 are applied. These improvement methods respectively are:

- Bagging: Bag1, Bag2, Bag3.
- Bragging: Brag1, Brag2, Brag3.
- Boosting.

This is all included in step 3 of the main algorithm. Step 3 is therefore a comprehensive process.

p) Repeatability of results:

In order to ensure that the results can be repeated and compared to one another, a seed setting is used. This ensures that the same dataset is generated for each estimation method and improvement method.

q) Computer package

All calculations are done using R 2.15.1. The computer code is available on request.

Additional remarks:

Before the main algorithm is stated, it is necessary to summarise the aims of this study, as it is captured in four structural procedures. Tables 4.3 to 4.6 present the main procedures, i.e.

a) Eight procedures used in calculating the *Nadaraya-Watson estimator* with *cross-validation bandwidth* selection.

b) Eight procedures used in calculating the *Nadaraya-Watson estimator* with *plug-in bandwidth* selection.

c) Eight procedures used in calculating the *local linear estimator* with *plug-in bandwidth* selection.

d) Eight procedures used in calculating the *BRNP estimator* with *cross-validation bandwidth* selection.

Note that 32 different estimation procedures are listed in Tables 4.3 to 4.6. Each estimation procedure is performed for different simulation setup scenarios.

Furthermore, Figures 4.3 to 4.5 are graphical representations of the possible simulation setup scenarios for Models 1 to 3 defined in Table 4.1. For each model defined in Table 4.1 this results in 36 simulation setup scenarios. The tables in Figures 4.3 to 4.5 can be interpreted as follows:

- The first row shows the three sample size values, i.e. $n = 50$, $n = 100$ and $n = 200$.

- The second row shows the two options for the underlying distribution of the covariate data. X-values are drawn from the $U(-2, 2)$ and $N(0, 1)$ distributions respectively.
- In the third row the two kernel functions used are shown. In the present study the Gaussian kernel and the triangular kernel are used, for reasons provided in e) above.
- The last row shows the three options for σ, which is chosen as 0.2, 0.6 and 1.0 for Models 1 and 2 and as 0.1, 0.3 and 0.5 for Model 3.

Basic description of sub-algorithm	Algorithm no.
Standard N-W estimator with cross-validation bandwidth selection	4.0
Bagging is applied to the cross-validation choice of h in the N-W estimator, approach 1	4.4
Bagging is applied to the cross-validation choice of h in the N-W estimator, approach 2	4.5
Bagging is applied to the cross-validation choice of h in the N-W estimator, approach 3	4.6
Bragging is applied to the cross-validation choice of h in the N-W estimator, approach 1	4.10
Bragging is applied to the cross-validation choice of h in the N-W estimator, approach 2	4.11
Bragging is applied to the cross-validation choice of h in the N-W estimator, approach 3	4.12
Boosting is applied to the N-W estimator with cross-validation bandwidth selection.	4.16

Table 4.3: Procedures used in calculating the *N-W estimator* with *cross-validation bandwidth selection*

Basic description of sub-algorithm	Algorithm no.
Standard N-W estimator with plug-in bandwidth.	4.1
Bagging is applied to the plug-in choice of h in the N-W estimator, approach 1	4.7
Bagging is applied to the plug-in choice of h in the standard N-W estimator, approach 2	4.8
Bagging is applied to the plug-in choice of h in the standard N-W estimator, approach 3	4.9
Bragging is applied to the plug-in choice of h in the standard N-W estimator, approach 1	4.13
Bragging is applied to the plug-in choice of h in the standard N-W estimator, approach 2	4.14
Bragging is applied to the plug-in choice of h in the standard N-W estimator, approach 3	4.15
Boosting is applied to the N-W estimator with plug-in bandwidth selection.	4.16

Table 4.4: Procedures used in calculating the *N-W estimator* with *plug-in bandwidth selection*

Basic description of sub-algorithm	Algorithm no.
Standard LL estimator with plug-in bandwidth selection.	4.2
Bagging is applied to the plug-in choice of h in the LL estimator, approach 1	4.7
Bagging is applied to the plug-in choice of h in the LL estimator, approach 2	4.8
Bagging is applied to the plug-in choice of h in the LL estimator, approach 3	4.9
Bragging is applied to the plug-in choice of h in the LL estimator, approach 1	4.13
Bragging is applied to the plug-in choice of h in the LL estimator, approach 2	4.14
Bragging is applied to the plug-in choice of h in the LL estimator, approach 3	4.15
Boosting is applied to the LL estimator with plug-in bandwidth selection.	4.16

Table 4.5: Procedures used in calculating the *local linear estimator* with *plug-in bandwidth selection*

Basic description of sub-algorithm	Algorithm no.
Standard BRNP estimator with cross-validation bandwidth selection.	4.3
Bagging is applied to the cross-validation choice of h in the BRNP estimator, approach 1	4.4
Bagging is applied to the cross-validation choice of h in the BRNP estimator, approach 2	4.5
Bagging is applied to the cross-validation choice of h in the BRNP estimator, approach 3	4.6
Bragging is applied to the cross-validation choice of h in the BRNP estimator, approach 1	4.10
Bragging is applied to the cross-validation choice of h in the BRNP estimator, approach 2	4.11
Bragging is applied to the cross-validation choice of h in the BRNP estimator, approach 3	4.12
Boosting is applied to the BRNP estimator with cross-validation bandwidth selection.	4.16

Table 4.6: Procedures used in calculating the *BRNP estimator* with *cross-validation bandwidth selection*

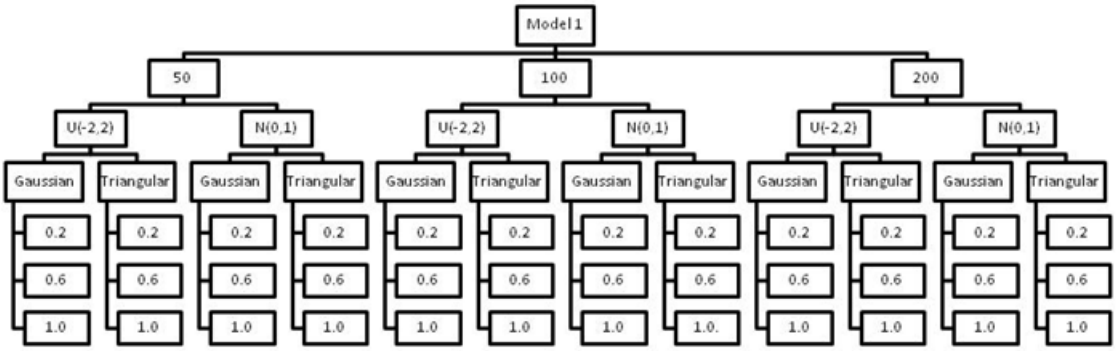

Figure 4.3: Simulation scenarios for Model 1

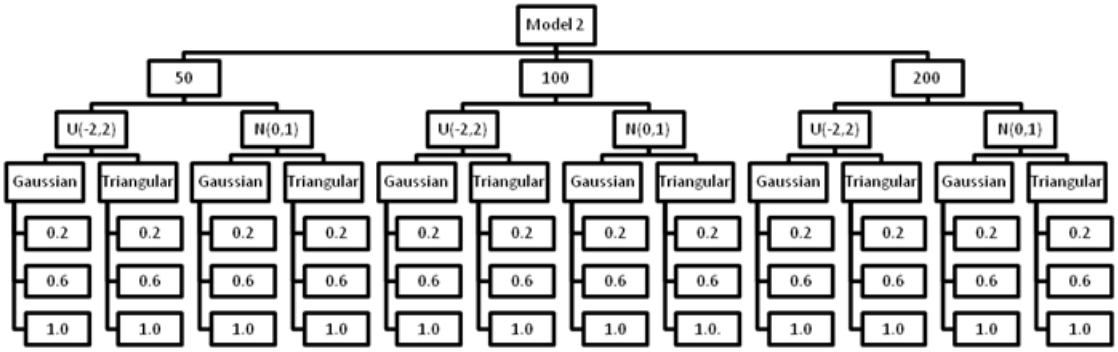

Figure 4.4: Simulation scenarios for Model 2

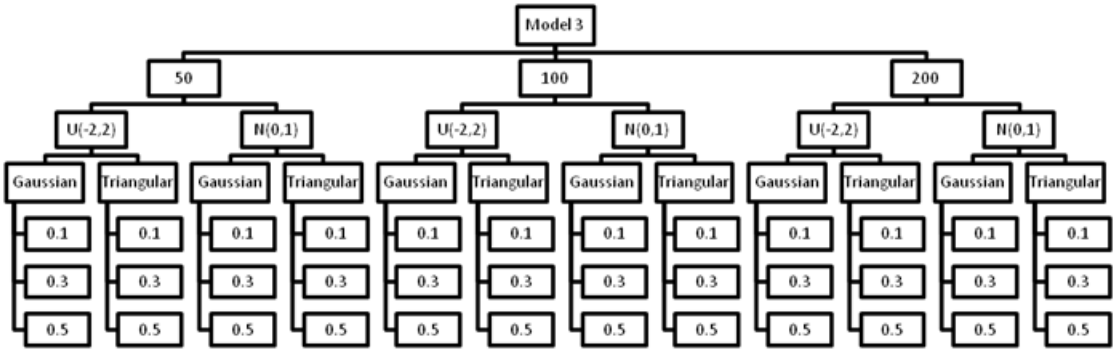

Figure 4.5: Simulation scenarios for Model 3

4.4.2 The main algorithm

The main algorithm is now presented with the following steps and covers the complete process from data generation to evaluating the discrepancy measures as it was explained above.

Algorithm 4.17. The main algorithm

1. Draw a random sample of size n from the underlying distribution and denote the sample by $\mathbf{S}_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$. The construction of the X-values and the Y-values are discussed in remarks b) and c). Possible choice of n is given in remark d).

2. Construct a grid of x-values, which is defined in remark l). Denote the minimum grid value by x_{min} and the maximum grid value by x_{max}.

3. In each grid point, x, estimate $m(x)$, one of the three models defined in Table 4.1, by utilising one of the various specific estimation methods listed in Tables 4.3 to 4.6.

4. Repeat the above steps Monte Carlo (MC) times and denote the estimates by $\hat{m}_i(x)$, $i = 1, 2, \ldots, MC$, where $x_{min} = x_1$ and $x_{max} = x_{100}$ The following matrix is obtained

$$\begin{bmatrix} \hat{m}_1(x_{min}) & \hat{m}_2(x_{min}) & \cdots & \hat{m}_{MC}(x_{min}) \\ \hat{m}_1(x_2) & \hat{m}_2(x_2) & \cdots & \hat{m}_{MC}(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{m}_1(x_{max}) & \hat{m}_2(x_{max}) & \cdots & \hat{m}_{MC}(x_{max}) \end{bmatrix}$$

5. For the chosen model and estimation method, calculate Monte Carlo approximations to the MSE, variance and squared bias in each grid point x:

$$\widehat{\text{MSE}}[\hat{m}(x)] = \frac{1}{MC}\sum_{i=1}^{MC}[\hat{m}_i(x) - m(x)]^2,$$

$$\widehat{\text{Var}}[\hat{m}(x)] = \frac{1}{MC}\sum_{i=1}^{MC}\left[\hat{m}_i(x) - \frac{1}{MC}\sum_{i=1}^{MC}[\hat{m}_i(x)]\right]^2$$

and

$$\widehat{\text{Bias}}^2[\hat{m}(x)] = \left[\frac{1}{MC}\sum_{i=1}^{MC}[\hat{m}_i(x)] - m(x)\right]^2.$$

6. Use numerical integration to integrate over the entire range of x-values to obtain approximated global measures:

$$\widehat{\text{MISE}} = \int_{x_{min}}^{x_{max}} \widehat{\text{MSE}}[\hat{m}(x)]dx, \tag{4.56}$$

$$\text{Approximate Integrated Variance} = \int_{x_{min}}^{x_{max}} \widehat{\text{Var}}[\hat{m}(x)]dx \tag{4.57}$$

and

$$\text{Approximate Integrated Bias}^2 = \int_{x_{min}}^{x_{max}} \widehat{\text{Bias}}^2[\hat{m}(x)]dx. \tag{4.58}$$

7. Repeat for all models in Table 4.1 and for all estimation method procedures listed in Tables 4.3 to 4.6.

The results of the various simulation executions are summarised in Appendices A to D. Appendix E displays a selection of graphical representations. In Chapter 5 the results of the simulation studies will be discussed.

Chapter 5

Results and conclusions

5.1 Introduction

In this chapter the results of the Monte Carlo studies, conducted to assess and compare the enhanced regression estimators with their respective smoothing parameter selection methods and the evaluation of the effect of the improvement methods on the regression estimators, are presented. The specific improvement methods are:

- Bagging
- Bragging
- Boosting

Chapter 3 provided a general theoretical framework regarding the bagging, bragging and boosting improvement methods. In Chapter 4 a more specific way of applying these methods to nonparametric regression estimators were displayed in the form of specific algorithms. In order to draw useful conclusions about the different methods, it is necessary to evaluate the performance of each method in various simulation settings. In the present study the following aspects of the simulation setup are varied:

- the underlying regression function $m(x)$,
- the underlying distribution of the covariate data,
- the variance of the error term used to generate the response data,
- the sample size, and
- the kernel function.

In literature, only limited studies exist with regard to the matters mentioned above. The present study aims to shed light on some of these issues. A discussion of the aspects in the list above and other aspects of the simulation setup, is provided in Section 4.4.1 in Chapter 4. The different simulation setup scenarios are graphically presented in Figures 4.3 to 4.5 in Chapter 4.

The Monte Carlo study consists of four parts. Table 5.1 provides a summary of the four simulation studies. Procedures and algorithms used in calculating the estimators are summarised in Table 4.3 to 4.6. Monte Carlo studies are also compared directly to one another. Firstly, the Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the Nadaraya-Watson estimator with plug-in bandwidth selection. Secondly, the Nadaraya-Watson estimator with plug-in bandwidth selection is compared to the local linear estimator with plug-in bandwidth selection. Table 5.2 provides a summary of the two comparison studies.

Simulation Study	Estimator	Section	Appendix
A	Nadaraya-Watson estimator with cross-validation bandwidth selection	5.7	A
B	Nadaraya-Watson estimator with plug-in bandwidth selection	5.8	B
C	Local linear estimator with plug-in bandwidth selection	5.9	C
D	Yao's estimator with cross-validation bandwidth selection	5.10	D

Table 5.1: Summary of the four simulation studies

Simulation Study	Comparison	Section	Appendix
A and B	Nadaraya-Watson estimator with cross-validation bandwidth selection compared to Nadaraya-Watson estimator with plug-in bandwidth selection	5.11	E
B and C	Nadaraya-Watson estimator with plug-in bandwidth selection compared to local linear estimator with plug-in bandwidth selection	5.12	F

Table 5.2: Summary of the two comparison studies

5.2 Formulation of specific research questions

The aims of the study stated in Section 4.1 in Chapter 4 can now be formulated as specific questions. These questions are ordered according to the four simulation studies. In each simulation study (Section 5.7 to 5.12), these questions will be answered and discussed where appropriate.

5.2.1 Nadaraya-Watson estimator with cross-validation bandwidth selection (SimA)

a) Does *bagging* indeed reduce the variance component of the Nadaraya-Watson estimator with cross-validation bandwidth selection as is expected in literature, Boshoff (2009), and is the reduction worth the calculation effort? The behaviour of the bias component is also of interest.

b) Is there a *bagged Nadaraya-Watson estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with cross-validation bandwidth selection, which is generally assumed to be the standard way of estimation?

c) Does *bragging* indeed reduce the variance component of the Nadaraya-Watson estimator with cross-validation bandwidth selection as is expected in literature, Boshoff (2009), and is the reduction worth the calculation effort? The behaviour of the bias component is also of interest.

d) Is there a *bragged Nadaraya-Watson estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with cross-validation bandwidth selection, which is generally assumed to be the standard way of estimation?

e) Does *boosting* indeed reduce the bias component of the Nadaraya-Watson estimator with cross-validation bandwidth selection as is expected in literature, Marzio and Taylor (2008) and Boshoff (2009), and is the reduction worth the calculation effort?

f) Does the *boosted Nadaraya-Watson estimator* perform better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with cross-validation bandwidth selection, which is generally assumed to be the standard way of estimation?

g) What is the influence of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the standard, boosted, bagged and bragged nonparametric regression estimators? In literature only limited studies exist with regard to the matters mentioned above. The present study will hopefully shed light on some of these issues.

Answers to the specific questions stated above, are given in Section 5.7.1.

5.2.2 Nadaraya-Watson estimator with plug-in bandwidth selection (SimB)

a) Does *bagging* indeed reduce the variance component of the Nadaraya-Watson estimator with plug-in bandwidth selection? *The bagged Nadaraya-Watson estimator with plug-in bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

b) Is there a *bagged Nadaraya-Watson estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with plug-in bandwidth selection?

c) Does *bragging* reduce the variance component of the Nadaraya-Watson estimator with plug-in bandwidth selection? *The bragged Nadaraya-Watson estimator with plug-in bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

d) Is there a *bragged Nadaraya-Watson estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with plug-in bandwidth selection?

e) Does *boosting* reduce the bias component of the Nadaraya-Watson estimator with plug-in bandwidth selection? *The boosted Nadaraya-Watson estimator with plug-in bandwidth selection seems to be a new contribution to the literature.*

f) Does the *boosted Nadaraya-Watson estimator* perform better, in the sense that the MISE is smaller overall, compared to the *standard Nadaraya-Watson estimator* with plug-in bandwidth selection?

g) What is the influence of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the standard, boosted, bagged and bragged nonparametric regression estimators?

In literature only limited studies exist with regard to the matters mentioned above. The present study aims to shed light on some of these issues.

Answers to the specific questions stated above, are given in Section 5.8.1.

5.2.3 Local linear estimator with plug-in bandwidth selection (SimC)

a) Does *bagging* reduce the variance component of the local linear estimator with plug-in bandwidth selection? *The bagged local linear estimator with plug-in bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

b) Is there a *bagged local linear estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard local linear estimator* with plug-in bandwidth selection?

c) Does *bragging* reduce the variance component of the local linear estimator with plug-in bandwidth selection? *The bragged local linear estimator with plug-in bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

d) Is there a *bragged local linear estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard local linear estimator* with plug-in bandwidth selection?

e) Does *boosting* reduce the bias component of the local linear estimator with plug-in bandwidth selection? *The boosted local linear estimator with plug-in bandwidth selection seems to be a new contribution to the literature.*

f) Does the *boosted local linear estimator* perform better, in the sense that the MISE is smaller overall, compared to the *standard local linear estimator* with plug-in bandwidth selection?

g) What is the influence of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the standard, boosted, bagged and bragged nonparametric regression estimators?

In literature only limited studies exist with regard to the matters mentioned above. The present study aims to shed light on some of these issues.

Answers to the specific questions stated above, are given in Section 5.9.1.

5.2.4 Yao's estimator with cross-validation bandwidth selection (SimD)

a) Does *bagging* reduce the variance component of the BRNP estimator with cross-validation bandwidth selection? *The bagged BRNP with cross-validation bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

b) Is there a *bagged* BRNP estimator that performs better, in the sense that the MISE is smaller overall, compared to the *standard BRNP estimator* with cross-validation bandwidth selection?

c) Does *bragging* reduce the variance component of Yao's estimator with cross-validation bandwidth selection? *The bragged BRNP estimator with cross-validation bandwidth selection seems to be a new contribution to the literature.* The behaviour of the bias component is also of interest.

d) Is there a *bragged BRNP estimator* that performs better, in the sense that the MISE is smaller overall, compared to the *standard BRNP estimator* with cross-validation bandwidth selection?

e) Does *boosting* reduce the bias component of the BRNP estimator with cross-validation bandwidth selection? *The boosted BRNP estimator with cross-validation bandwidth selection seems to be a new contribution to the literature.*

f) Does the *boosted BRNP estimator* perform better, in the sense that the MISE is smaller overall, compared to the *standard BRNP estimator* with cross-validation bandwidth selection?

g) What is the influence of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the standard, boosted, bagged and bragged nonparametric regression estimators?

Due to the computer time intensiveness of the BRNP estimator only limited studies exist with regard to the matters mentioned above. The present study aims to shed light on some of these issues.

Answers to the specific questions stated above, are given in Section 5.10.1.

5.2.5 Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the Nadaraya-Watson estimator with plug-in bandwidth selection (SimA compared to SimB)

a) Does the Nadaraya-Watson estimator with plug-in bandwidth selection perform better, in the sense that the MISE is smaller overall, compared to the Nadaraya-Watson estimator with cross-validation bandwidth selection? The comparison of the two estimators in standard, bagged, bragged and boosted form seems to be a new contribution to the literature.

Answers to the specific question stated above, are given Section 5.11.1.

5.2.6 Nadaraya-Watson estimator with plug-in bandwidth selection compared to the local linear estimator with plug-in bandwidth selection (SimB compared to SimC)

a) Does the Nadaraya-Watson estimator with plug-in bandwidth selection perform better, in the sense that the MISE is smaller overall, compared to the local linear estimator with plug-in bandwidth selection? The comparison of the two estimators in standard, bagged, bragged and boosted forms, seems to be a new contribution to the literature.

Answers to the specific question stated above, are given in Section 5.12.1.

First, guidelines to read the respective tables in Appendix A to H are now presented. This will assist the reader to make sense of the large output of results in this study.

5.3 Guidelines for reading Appendices A-D

The results where the effect of bagging, bragging and boosting on the three kernel regression estimators with their respective smoothing parameter selection methods are explored, are reported in Appendix A to D. The following remarks can be used as guidelines when consulting these appendices:

- Appendix A: Nadaraya-Watson estimator with cross-validation bandwidth selection (SimA).
- Appendix B: Nadaraya-Watson estimator with plug-in bandwidth selection (SimB).
- Appendix C: Local linear estimator with plug-in bandwidth selection (SimC).
- Appendix D: Yao's estimator with cross-validation bandwidth selection (SimD).
- All four appendices consist of twelve tables. Each of the twelve tables displays the results of a particular combination of model choice, underlying distribution of the predictor variable X and kernel choice. The specific combination is indicated in the caption of the table.
- Each table consists of nine blocks of results. The nine blocks represent the different simulation setup scenarios regarding the sample size n and the error variance σ. The different simulation setup scenarios are presented graphically in Figures 4.3 to 4.5 in Chapter 4.
- Each block consists of four columns and eight rows. The four columns respectively display the MISE, the integrated squared bias, the integrated variance and, the integrated bias obtained when the main algorithm, Algorithm 4.17, is applied.
- Each of the eight rows presents the results when a particular method is applied to obtain the estimate $\hat{m}(x)$ in step 3 of the main algorithm. The rows respectively display the following:

a) For the Nadaraya-Watson with cross-validation bandwidth selection (SimA):

- The first row contains the results when cross-validation is applied to select h for the *standard* Nadaraya-Watson estimator via Algorithm 4.0.
- Rows 2 to 4 contain the results when *bagging* is applied to cross-validation bandwidth selection in the standard Nadaraya-Watson estimator via Algorithms 4.4 to 4.6 respectively.
- Rows 5 to 7 contain the results when *bragging* is applied to cross-validation bandwidth selection in the standard Nadaraya-Watson estimator via Algorithms 4.10 to 4.12 respectively.
- Row 8 contains the results when *boosting* is applied to the standard Nadaraya-Watson estimator with cross-validation bandwidth selection via Algorithm 4.16.

b) For the Nadaraya-Watson with plug-in bandwidth selection (SimB):

- The first row contains the results when a plug-in method is applied to select h for the *standard* Nadaraya-Watson estimator via Algorithm 4.1.
- Rows 2 to 4 contain the results when *bagging* is applied to plug-in bandwidth selection in the standard Nadaraya-Watson estimator via Algorithms 4.7 to 4.9 respectively.
- Rows 5 to 7 contain the results when *bragging* is applied to plug-in bandwidth selection in the standard Nadaraya-Watson estimator via Algorithms 4.13 to 4.15 respectively.
- Row 8 contains the results when *boosting* is applied to the standard Nadaraya-Watson estimator with plug-in bandwidth selection via Algorithm 4.16.

c) For the local linear estimator with plug-in bandwidth selection (SimC):

- The first row contains the results when a plug-in method is applied to select h for the *standard* local linear estimator via Algorithm 4.2.
- Rows 2 to 4 contain the results when *bagging* is applied to plug-in bandwidth selection in the standard local linear estimator via Algorithms 4.7 to 4.9 respectively.
- Rows 5 to 7 contain the results when *bragging* is applied to plug-in bandwidth selection in the standard local linear estimator via Algorithms 4.13 to 4.15 respectively.
- Row 8 contains the results when *boosting* is applied to the standard local linear estimator with plug-in bandwidth selection via Algorithm 4.16.

d) For Yao's estimator with cross-validation bandwidth selection (SimD):

- The first row contains the results when cross-validation is applied to select h for the *standard* BRNP estimator via Algorithm 4.3.
- Rows 2 to 4 contain the results when *bagging* is applied to cross-validation bandwidth selection in the standard BRNP estimator via Algorithms 4.4 to 4.6 respectively.
- Rows 5 to 7 contain the results when *bragging* is applied to cross-validation bandwidth selection in the standard BRNP estimator via Algorithms 4.10 to 4.12 respectively.
- Row 8 contains the results when *boosting* is applied to the standard BRNP estimator with cross-validation bandwidth selection via Algorithm 4.16.

5.4 Guidelines for reading Appendices E and F

Appendix E and F contain the results where two simulation studies are compared. Firstly, the Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the Nadaraya-Watson estimator with plug-in bandwidth selection, the results are shown in Appendix E. Secondly, the Nadaraya-Watson estimator with plug-in bandwidth selection is compared to the local linear estimator with plug-in bandwidth selection, the results are shown in Appendix F.

In order to compare the results of the *Nadaraya-Watson estimator with cross-validation bandwidth selection* to the results of the *Nadaraya-Watson estimator with plug-in bandwidth selection*, a ratio of the obtained MISE values is used:

$$\text{ratio} = \frac{\text{MISE}_{NW_{CV}}}{\text{MISE}_{NW_{plug}}} \tag{5.1}$$

- If this ratio is larger than one, it implies the Nadaraya-Watson estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.
- If the ratio is smaller than one, the Nadaraya-Watson estimator with cross-validation bandwidth selection has the smaller MISE value for a specific simulation scenario.

Also, to compare the results of the *local linear estimator with plug-in bandwidth selection* to the results of the *Nadaraya-Watson estimator with plug-in bandwidth selection* a similar ratio of the MISE values is used:

$$\text{ratio} = \frac{\text{MISE}_{LL_{plug}}}{\text{MISE}_{NW_{plug}}} \tag{5.2}$$

- If this ratio is larger than one, it implies the Nadaraya-Watson estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.
- If the ratio is smaller than one, the local linear estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.

The following remarks can be used as guidelines when consulting appendix E and F:

- Appendix E and F consist of twelve tables. Each of the twelve tables displays the results of a particular combination of model choice, underlying distribution of the predictor variable X and kernel choice. The specific combination is indicated in the caption of the table.
- Each table consists of three blocks of results. The various blocks represent the different simulation setup scenarios regarding the error variance σ. Each block consists of three columns and eight rows. The three columns represent the different simulation setup scenarios regarding the sample size n. The different simulation setup scenarios are graphically presented in Figures 4.3 to 4.5 in Chapter 4.
- Each of the eight rows presents the results of the *ratio of MISE values* when a particular method is applied to obtain the estimate $\hat{m}(x)$ in step 3 of the main algorithm. The rows respectively display the following:

a) For the Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the Nadaraya-Watson estimator with plug-in bandwidth selection (SimA vs. SimB)

- The first row contains the results when the *standard* Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Rows 2 to 4 contain the results when the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Rows 5 to 7 contain the results when the *bragged* Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the *bragged* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Row 8 contains the results when the *boosted* Nadaraya-Watson estimator with cross-validation bandwidth selection is compared to the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection.

b) For the Nadaraya-Watson estimator with plug-in bandwidth selection compared to the local linear estimator with plug-in bandwidth selection (SimB vs. SimC)

- The first row contains the results when the *standard* local linear estimator with plug-in bandwidth selection is compared to the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Rows 2 to 4 contain the results when the *bagged* local linear estimator with plug-in bandwidth selection is compared to the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Rows 5 to 7 contain the results when the *bragged* local linear estimator with plug-in bandwidth selection is compared to the *bragged* Nadaraya-Watson estimator with plug-in bandwidth selection.
- Row 8 contains the results when the *boosted* local linear estimator with plug-in bandwidth selection is compared to the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection.

5.5 Guidelines for reading Appendix G

Appendix G shows the computation time, in seconds, for the standard nonparametric regression estimators and the improved nonparametric regression estimators. The following remarks can be used as guidelines when consulting Appendix G.

- This appendix consists of five tables. There is a table for each improvement method.
  - The first table shows the computation time, in seconds, when *no improvement method* is applied to the nonparametric regression estimators.

  - The second table shows the computation time, in seconds, when the *boosting improvement method* is applied to the nonparametric regression estimators.
  - The third table shows the computation time, in seconds, when the *Bag1 improvement method* is applied to the nonparametric regression estimators.
  - The fourth table shows the computation time, in seconds, when the *Bag2 improvement method* is applied to the nonparametric regression estimators.
  - The fifth table shows the computation time, in seconds, when the *Bag3 improvement method* is applied to the nonparametric regression estimators.

- Each table consists of two blocks. The first block shows the simulation setup scenario and consists of four columns. The four columns respectively display the error variance, the sample size, kernel choice and the underlying distribution of the predictor variable.

- The second block shows the estimation method and consists of three columns. The three columns respectively display the computation time, in seconds, for the Nadaraya-Watson estimator with cross-validation bandwidth selection (NW_{CV}), the Nadaraya-Watson estimator with plug-in bandwidth selection (NW_{PLUG}) and the local linear estimator with plug-in bandwidth selection (LL_{PLUG}).

- The last table shows the computation time, in seconds for the standard BRNP estimator and the improved BRNP estimator.

- The table for the computation time for the BRNP estimator also consists of two blocks. The first block shows the simulation setup scenario and consists of four columns. The four columns respectively display the error variance, the sample size, kernel choice and the underlying distribution of the predictor variable.

- The second block shows the improvement methods applied to the BRNP estimator and consists of five columns. The five columns respectively display the computation time, in seconds, for the standard BRNP estimator, Bag1 BRNP estimator, Bag2 BRNP estimator, Bag3 BRNP estimator and the boosted BRNP estimator.

The computation time, of the standard nonparametric regression estimators and the improved nonparametric regression estimators, is discussed in Section 5.13.

5.6 Guidelines for reading Appendix H

In Appendix H a limited number of graphs are presented to graphically illustrate general aspects of the simulation studies, such as the bias and variance behaviour of the various methods. The presented graphs are now briefly explained.

- In particular, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$ and $n = 100$ scenario is considered for Model 1 and Model 2 respectively. The specific choice of scenario is arbitrary, any other scenario could have been chosen for this illustration.

- For each nonparametric regression estimator used in this study four graphs are drawn.
- The first three graphs respectively display the standard, boosted and bagged nonparametric regression estimators. Each of the 200 Monte Carlo samples are drawn on one figure.
- On the fourth graph the average estimator over the 200 Monte Carlo samples is presented for the standard, boosted and bagged nonparametric regression estimators.
- Figures H.1 to H.2 illustrate the results for the Nadaraya-Watson estimator with cross-validation bandwidth selection. These are graphical representations of the results displayed in Appendix A, Table A.1 and Table A.5 respectively.
- Figures H.3 to H.4 illustrate the results for the Nadaraya-Watson estimator with plug-in bandwidth selection. These are graphical representations of the results displayed in Appendix B, Table B.1 and Table B.5 respectively.
- Figures H.5 to H.6 illustrate the results for the local linear estimator with plug-in bandwidth selection. These are graphical representations of the results displayed in Appendix C, Table C.1 and Table C.5 respectively.
- Figures H.7 to H.8 illustrate the results of the BRNP estimator with cross-validation bandwidth selection. These are graphical representations of the results displayed in Appendix D, Table D.1 and Table D.2 respectively. Note that the simulation study for the BRNP estimator is done on twenty Monte Carlo iterations. This is explained in Section 5.10.

5.7 Observations and conclusions for the Nadaraya-Watson estimator with cross-validation bandwidth selection

- The first aim of simulation study A is to determine if the *improvement methods* provide better estimators for the mean regression curve than the *standard Nadaraya-Watson estimator with cross-validation bandwidth selection*. The effect of bagging and bragging on the standard Nadaraya-Watson estimator with cross-validation bandwidth selection was studied by Boshoff (2009). The effect of boosting on the standard Nadaraya-Watson estimator with cross-validation bandwidth selection was studied by Marzio and Taylor (2008) and confirmed by Boshoff (2009).
- Secondly, the influence of sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the nonparametric regression estimator involved in this simulation setup, is determined.
- The results of SimA, where the effect of bagging, bragging and boosting on the standard Nadaraya-Watson estimator with cross-validation bandwidth selection are explored, are reported in Tables A.1 to A.12 in Appendix A. Guidelines for reading Appendix A are given in Section 5.3.

- Tables A.13 to A.15 and Tables A.16 to A.18 are supplementary to explaining the results of the bagging and bragging methods:

  a) Tables A.13 to A.15 provide a summary of the simulation setup, *bagging method* producing the *smallest integrated variance value*, *bagging method* producing the *smallest MISE value*, *bragging method* producing the *smallest integrated variance value* and the *bragging method* producing the *smallest MISE value*, compared to the integrated variance and MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. Simulation setup scenarios where the smallest MISE value is obtained using the standard Nadaraya-Watson estimator, is marked with a "*".

  b) Table A.16 to A.18 shows, for each model choice, in how many simulation setup scenarios a bagging and bragging method performed the best in terms of the integrated variance value and the MISE value.

The results of Simulation Study A are now discussed and conclusions are drawn.

1. **The effect of bagging and bragging:**

   In Chapter 3, bagging and bragging were introduced as variance reduction techniques. Three bagging as well as three bragging methods are applied to the cross-validation selection of h, used in the Nadaraya-Watson estimator. The corresponding algorithms were presented in Chapter 4. The aim of the implementation of the improvement methods, bagging and bragging, is to reduce the variability in the application of cross-validation bandwidth selection method.

   The results contained in Tables A.1 to A.12 enable the following observations regarding the effect of the bagging and bragging procedures as improvement methods on the standard Nadaraya-Watson estimator with cross-validation bandwidth selection:

   - For Models 1 and 2, in all *thirty-six* simulation scenarios, all three bagging and all three bragging procedures led to a smaller integrated variance value compared to the integrated variance value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.
   - For Model 3, at least one bagging and one bragging method led to a smaller integrated variance value compared to the integrated variance value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection, for each simulation setup scenario.
   - At least one bagging and one bragging method in each simulation setup scenario also led to smaller MISE values, except in the following three scenarios:
     - Model 3 for $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 200$.
     - Model 3 for $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 50$.
     - Model 3 for $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 200$.
   - In the simulation setup scenarios where a bagged Nadaraya-Watson estimator or a bragged Nadaraya-Watson estimator has the smallest MISE value, compared to the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection, the squared integrated bias value is still

higher compared to the integrated squared bias value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

2. **Comparison between Bag1, Bag2 and Bag3 methods in terms of the MISE:**
   Table A.16 to A.18 shows how many times, out of the *thirty-six* simulation setup scenarios, for a specific model, a certain bagging method performed the best, in the sense that the integrated variance value and the MISE values are smaller overall, compared to the integrated variance value and the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. The aim of this discussion is to determine if there is a specific bagging method that consistently produces the smaller MISE value compared to the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

   - For Model 1, Bag1 performed better, in *twenty-nine* (80.56%) cases, than Bag2 and Bag3.
   - Bag2 produced the smallest MISE value for *one* simulation setup scenario:
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$.
   - Bag3 performed better than Bag1 and Bag2 for Model 1 in the following *six* simulation setup scenarios:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$.
     - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 100.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 100$.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 100$, 200.
   - For Model 2, Bag1 performed better, in *twenty-two* (61.11%) cases, than Bag2 and Bag3.
   - Bag2 produced the smaller MISE values for Model 2 in the following *five* simulation setup scenarios:
     - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 200$.
     - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 200$.
   - Bag3 produced the smaller MISE values for Model 2 in the following *nine* simulation setup scenarios:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100.
     - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100.
   - For Model 3, Bag3 performed better in *twenty-eight* (61.11%) cases, than Bag1 and Bag2.
   - Bag1 produced the smaller MISE values for Model 3 in the following *seven* simulation setup scenarios:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 100$, 200.
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.3$, $n = 200$.
     - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 50$, 100, 200.
     - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, $n = 200$.
   - Bag2 produced the smaller MISE value for Model 3 in *one* scenario:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 50$.

Remarks:

- From Table A.16 to A.18 it is clear that in most cases Bag2 or Bag3 resulted in the smallest integrated variance value and Bag1 resulted in the smallest MISE value.
- Recall from Chapter 4 that the Bag2 and Bag3 algorithms obtain estimates of $m(x)$ for each bootstrap sample separately and then aggregate these estimates. Aggregation is done over the bootstrap estimates to obtain the final estimates $\hat{m}(x)$.
- In the algorithm for Bag1, a bagged bandwidth is obtained by aggregating over the cross-validation bandwidths of the bootstrap samples. After rescaling, the bagged bandwidth is applied to the original Monte Carlo sample.
- In Bag2 and Bag3 the focus is on reducing the variability of $\hat{m}(x)$, whereas the focus in Bag1 is to reduce the variability of the cross-validation bandwidth used to estimate $\hat{m}(x)$. Therefore, Bag2 and Bag3 estimators will have smaller variance values than Bag1 estimators. However, accuracy is compromised and the Bag2 and Bag3 estimates will lie further from the true curve $m(x)$ resulting in higher bias values than Bag1 estimates. Therefore smaller MISE values are obtained by Bag1, compared to that of Bag2 and Bag3.

3. Comparison between Brag1, Brag2 and Brag3 methods in terms of the MISE:

Table A.16 to A.18 shows how many times, out of the *thirty-six* simulation setup scenarios, for a specific model, a certain bragging method performed better, in the sense that the integrated variance value and the MISE values are smaller overall, compared to the integrated variance value and the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. The aim of this discussion is to determine if there is a specific bragging method that consistently produces the smaller MISE value compared to the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

- For Model 1, Brag1 performed better, in *thirty-two* (88.89%) cases, than Brag2 and Brag3.
- There are no cases where Brag2 performed better than Brag1 and Brag3 in terms of the MISE.
- Brag3 outperformed Brag1 and Brag2 in terms of the MISE in the following *four* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel $\sigma = 1.0$, $n = 50$.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 100$.
- For Model 2, Brag1 performed better, in *twenty-three* (63.89%) cases, than Brag2 and Brag3.
- Brag2 performed the best in terms of the MISE in *one* simulation setup scenario:
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$.
- For Model 2, Brag3 performed the best in terms of the MISE in the following *twelve* simulation setup scenarios:

- $X \sim N(0,1)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.

- For Model 3, Brag3 performed better, in *twenty-two* (61.11%) cases, than Brag1 and Brag2.
- There are no cases where Brag2 performed better than Brag1 and Brag3 in terms of the MISE.
- For Model 3, Brag1 performed the best in terms of the MISE in the following *fourteen* simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.1$, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.3$, $n = 100$, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.5$, $n = 200$.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 50$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, $n = 200$.

Remarks:

- It is clear from Table A.16 to A.18 that Brag3 consistently improved the variance component of the Nadaraya-Watson estimator with cross-validation bandwidth selection.
- The Brag3 algorithm obtains estimates of $m(x)$ for each bootstrap sample separately and then aggregates these estimates. The aim of the Brag3 method is to reduce the variability of $\hat{m}(x)$. Brag3 estimators will therefore have smaller variance values, however accuracy is compromised and the Brag3 estimates in some cases will lie further from the true curve $m(x)$ resulting in higher bias values.
- However, the Brag1 algorithm aggregates over the bootstrap bandwidths and uses the resulting bandwidth to determine $\hat{m}(x)$. The focus of the Brag1 method is to reduce the variability of the cross-validation bandwidth used to estimate $\hat{m}(x)$. Therefore smaller MISE values are obtained by Brag1, compared to Brag2 and Brag3.

4. The effect of boosting:

In Chapter 3 boosting was introduced as a bias reduction technique. The results contained in Tables A.1 to A.12 enable the following observations regarding the overall effect of the boosting procedure as an improvement method on the standard Nadaraya-Watson estimator with cross-validation bandwidth selection:

- For all three models, in all *thirty-six* simulation setup scenarios, the boosting procedure led to a smaller integrated squared bias value compared to the integrated squared bias value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.
- However, in most of the simulation setup scenarios there was a significant increase in the integrated squared variance value of the boosted regression estimator. This increase in the variance component resulted in the boosting improvement method not being effective in terms of the MISE.
- There were a few cases where the boosting method improved the MISE value. For Models 1 and 2 the boosting procedure reduced the MISE for the following *six* simulation setup scenarios:

- $X \sim N(0,1)$, $N(0,1)$ kernel, triangular kernel, $n = 50$, 100, 200 and $\sigma = 0.2$.

- However, for Model 3, the boosting procedure fared consistently worse than the standard Nadaraya-Watson estimator with cross-validation bandwidth selection regarding the MISE.

5. **The influence of the sample size, error variance, kernel and the covariate distribution:** With regards to the sample size n, the error variance σ, the kernel and the covariate distribution, the following observations can be made when studying Tables A.1 to A.12:

   - The effect of the sample size:
     - Large samples contain more information about the true underlying regression relationship $m(x)$ than small samples and will therefore result in estimators with less bias and variability. Therefore, as n increased, the MISE, integrated squared bias and integrated variance values decreased.
   - The effect of the error variance:
     - Samples with large error variance resulted in worse estimators for $m(x)$. This is reflected in the increase in the MISE, approximated integrated variance and approximated integrated squared bias as σ increases.
   - The effect of the kernel:
     - By using the $N(0,1)$ kernel, in most of the cases, smaller MISE, integrated squared bias and integrated variance values are obtained, compared to the triangular kernel. This is to be expected, since the $N(0,1)$ kernel has longer tails and therefore more data are included in the estimation process than when using the restricted triangular kernel.
   - The effect of the covariate distribution:
     - Smaller MISE, integrated squared bias, and integrated variance are obtained when the covariate data are drawn from the $U(-2,2)$ distribution than for the scenarios where data are drawn from the $N(0,1)$ distribution. This event can be ascribed to an increased boundary effect when data are drawn from the unevenly distributed $N(0,1)$ distribution. More specifically data drawn from a $N(0,1)$ distribution contain less observations near the boundary of the grid of x values. This results in large variability and bias near the boundary points.

5.7.1 Conclusions drawn for the Nadaraya-Watson estimator with cross-validation bandwidth selection

Answers are now provided for the questions stated in Section 5.2.1

a) Bagging did reduce the variance component of the Nadaraya-Watson estimator with cross-validation bandwidth selection. This conclusion is also observed by Boshoff (2009). In most of the simulation scenarios Bag3 performed better than Bag1 and Bag2 in terms of reducing the variance component of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

b) In most simulation scenarios the bagged Nadaraya-Watson estimator performed better, in the sense that the MISE is smaller overall, compared to the standard Nadaraya-Watson estimator. Bag1 performed the best, compared to Bag2 and Bag3, in most simulation scenarios, in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

c) Bragging did reduce the variance component of the Nadaraya-Watson estimator with cross-validation bandwidth selection. Brag3 performed better than Brag1 and Brag2, in most simulation scenarios, in terms of reducing the variance component of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

d) The bragged Nadaraya-Watson estimator performed better, in the sense that the MISE is smaller overall, compared to the standard Nadaraya-Watson estimator. In most simulation scenarios Brag1 performed the best, compared to Brag2 and Brag3, in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

e) The boosted Nadaraya-Watson estimator always has a smaller integrated squared bias value compared to the integrated squared bias value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. This conclusion is also confirmed by Marzio and Taylor (2008) and Boshoff (2009).

f) The boosted Nadaraya-Watson estimator performed better than the Nadaraya-Watson estimator with cross-validation bandwidth selection in terms of the MISE value, only in limited cases. The integrated squared bias of the boosted estimator in these scenarios is considerably lower than the bias of the standard Nadaraya-Watson estimator. The boosted estimator performed the best in terms of the MISE when $X \sim N(0,1)$ and the error variance is small.

g) The MISE decreases as the sample size increases. This is to be expected, since the underlying regression relationship $m(x)$ will be more evident from large samples carrying a lot of information, than smaller samples, which are used to estimate $m(x)$. Increasing values of the error variance, σ, resulted in larger MISE values, since $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance compared to data generated to follow $m(x)$ closely, i.e. data constructed with small error variance. The $N(0,1)$ kernel performs slightly better than the triangular kernel with regard to the minimised MISE. The underlying distribution of the covariate data influences the MISE values. Smaller MISE values are observed when $X \sim U(-2,2)$, compared to $X \sim N(0,1)$.

5.8 Observations and conclusions for the Nadaraya-Watson estimator with plug-in bandwidth selection

- The first aim of simulation study B is to establish if the *improvement methods* provide better estimators for the mean regression curve than the *standard Nadaraya-Watson estimator with plug-in bandwidth selection.*

The bagged, bragged and boosted Nadaraya-Watson estimators with plug-in bandwidth selection seem to be new contributions to the literature.

- Secondly, the effects of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the nonparametric regression estimator involved in this simulation setup, are determined.

- The results of SimB, where the effect of bagging, bragging and boosting on the standard Nadaraya-Watson estimator with plug-in bandwidth selection are explored, are reported in Tables B.1 to B.12 in Appendix B. Guidelines for reading Appendix B are given in Section 5.3.

The results of Simulation Study B are now discussed and conclusions are drawn.

1. **The effect of bagging and bragging:**
   The aim of the implementation of the bagging and bragging methods is to determine if these methods can reduce the variance of the Nadaraya-Watson estimator with plug-in bandwidth selection. Three bagging as well as three bragging methods are applied to the plug-in bandwidth selection of h, used in the Nadaraya-Watson estimator. The results contained in Tables B.1 to B.12 enables the following observations regarding the effect of the bagging and bragging procedures as improvement methods on the standard Nadaraya-Watson estimator with plug-in bandwidth selection:

   - For Model 1, the *bagged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *five* simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, Bag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Bag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, $n = 50$, Bag3.
     - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 50$, Bag3.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Bag3.
   - For Model 1, the *bragged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *four* simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, Brag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 200$, Brag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Brag1.
     - $X \sim N(0, 1)$, triangular kernel, 1.0, $n = 50$, Brag1.
   - For Model 2, the *bagged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *fourteen* simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, Bag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 200$, Bag1.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Bag1.

  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 200$, Bag2.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Bag3.

- For Model 2, the *bragged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *six* simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 50$, Brag1.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Brag1.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 200$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Brag1.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Brag3.

- For Model 3, the *bagged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *eleven* simulation setup scenarios:
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 50$, Bag3.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, Bag3.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.5$, $n = 50$, Bag2.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 50$, Bag1.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 200$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 100$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.3$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.5$, $n = 50$, Bag3.

- For Model 3, the *bragged* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *six* simulation setup scenarios:

- $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.1$, $n = 50$, Brag1.
- $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, Brag1.
- $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 50$, Brag1.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, $n = 50$, Brag3.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, $n = 50$, Brag3.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.5$, $n = 50$, Brag3.

Remarks:

- In the above cases where the *bagged* or *bragged* Nadaraya-Watson estimator with plug-in bandwidth selection performed better than the standard Nadaraya-Watson estimator with plug-in bandwidth selection, in terms of the MISE, it was due to a smaller variance component. Hence, bagging and bragging indeed reduced the variance of the Nadaraya-Watson estimator with plug-in bandwidth selection.
- Bag1 and Bag3 are the best bagging methods in terms of the MISE value.
- Brag1 and Brag3 are the best bragging methods in terms of the MISE value.
- From the above observations, it is clear that the bagging and bragging methods are the most effective, in terms of the MISE, in small sample sizes.
- Most of the simulation setup scenarios where bagging and bragging led to improvements in terms of the MISE value, are where $X \sim N(0, 1)$ and the triangular kernel is used.

2. The effect of boosting:

The aim of the implementation of boosting is to determine if this improvement method can reduce the bias of the Nadaraya-Watson estimator with plug-in bandwidth selection. The results contained in Tables B.1 to B.12 enable the following observations regarding the overall effect of the boosting procedure as an improvement method on the standard Nadaraya-Watson estimator with plug-in bandwidth selection:

- For Model 1, the *boosted* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *sixteen* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, $n = 50$, 100 and 200.
  - $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.6$, $n = 50$.
- For Model 2, the *boosted* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *seventeen* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, $n = 50$, 100 and 200.
  - $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.6$, $n = 50$.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$.

  For Model 3, the *boosted* Nadaraya-Watson estimator has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator, for the following *twelve* simulation setup scenarios:

- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.1$, $n = 50, 200$.
- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 50$.
- $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.1$, $n = 50$, 100 and 200.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, $n = 50$.

Remarks:

- For all three models, in all *thirty-six* simulation setup scenarios, the boosting procedure led to a smaller integrated squared bias value compared to the integrated squared bias value of the standard Nadaraya-Watson estimator with plug-in bandwidth selection. Boosting indeed reduced the bias of the Nadaraya-Watson estimator with plug-in bandwidth selection.
- However, there are only a few cases where boosting led to a smaller MISE value compared to the standard Nadaraya-Watson estimator with plug-in bandwidth selection.
- Boosting was the most effective in simulation setup scenarios where the error variance is small.

3. **The influence of the sample size, error variance, kernel and the covariate distribution:** With regards to the sample size n, the error variance σ, the kernel and the covariate distribution, the following observations can be made when studying Tables B.1 to B.12:

- The effect of the sample size:
  - The MISE value decreased significantly as n increased. This is to be expected, since larger sample sizes give a better indication of the underlying relationship between the predictor variable and the outcome variable.
- The effect of the kernel:
  - In the case of the Nadaraya-Watson estimator with plug-in bandwidth selection, the $N(0, 1)$ kernel did not always outperform the triangular kernel in terms of the MISE, integrated variance and integrated squared bias values obtained. In the following simulation setup scenarios the triangular kernel performed better than the $N(0, 1)$ kernel:
    - For Models 1 and 2: $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $\sigma = 0.6$, $n = 50$, 100 and 200.
    - For Model 3: $X \sim U(-2, 2)$, $X \sim N(0, 1)$, $n = 50$, 100, 200, $\sigma = 0.3$.
- The effect of the covariate distribution:
  - Smaller MISE, integrated squared bias, and integrated variance are obtained in all cases when the covariate data are drawn from the $U(-2, 2)$ distribution than for the scenarios where data are drawn from the $N(0, 1)$ distribution. A possible reason for this is that data drawn from a $N(0, 1)$ distribution contains less observations near the boundary of the grid of x values. This results in large variability and bias near the boundary points.

5.8.1 Conclusions drawn for the Nadaraya-Watson estimator with plug-in bandwidth selection

Answers are now provided for the questions stated in Section 5.2.2

a) Bagging reduced the variance component of the Nadaraya-Watson estimator with plug-in bandwidth selection only in limited cases. Bagging was the most effective in the case of small sample sizes, $X \sim N(0,1)$ and when the triangular kernel is used.

b) In limited cases the bagged Nadaraya-Watson estimator performed better, in the sense that the MISE is smaller overall, compared to the standard Nadaraya-Watson estimator. Bag3 performed the best, compared to Bag1 and Bag2, in most simulation scenarios, in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with plug-in bandwidth selection.

c) Bragging reduced the variance component of the Nadaraya-Watson estimator with plug-in bandwidth selection only in limited cases. Bragging was the most effective in the case of small sample sizes, $X \sim N(0,1)$ and when the triangular kernel is used.

d) In limited cases the bragged Nadaraya-Watson estimator performed better, in the sense that the MISE is smaller overall, compared to the standard Nadaraya-Watson estimator. In most simulation scenarios Brag1 or Brag3 performed the best, in the sense of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with plug-in bandwidth selection.

e) The boosted Nadaraya-Watson estimator always has a smaller integrated squared bias value compared to the integrated squared bias value of the standard Nadaraya-Watson estimator with plug-in bandwidth selection.

f) The boosted Nadaraya-Watson estimator performed better than the Nadaraya-Watson estimator with plug-in bandwidth selection in terms of the MISE value, only in limited cases. In general, the boosted Nadaraya-Watson estimator performed better than the Nadaraya-Watson estimator with plug-in bandwidth selection for small error variance.

g) The MISE decreases as the sample size increases, since the underlying regression relationship $m(x)$ is more evident from large samples carrying a lot of information, than smaller samples, which are used to estimate $m(x)$. Larger MISE values are observed for large error variance, since $m(x)$ is more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance compared to data generated to follow $m(x)$ closely, i.e. data constructed with small error variance. The $N(0,1)$ kernel performs slightly better than the triangular kernel with regard to the minimised MISE. Smaller MISE values are observed when $X \sim U(-2,2)$, compared to $X \sim N(0,1)$.

5.9 Observations and conclusions for the local linear estimator with plug-in bandwidth selection

- The first aim of simulation study C is to establish if the *improvement methods* provide better estimators for the mean regression curve than the *standard local linear estimator with plug-in bandwidth selection*. The bagged, bragged and boosted local linear estimators with plug-in bandwidth selection seem to be new contributions to the literature.

- The effects of the sample size n, the error variance σ, the kernel and the covariate distribution on the performance of the nonparametric regression estimator involved in this simulation setup, are determined.

- The results of Simulation Study C, where the effect of bagging, bragging and boosting on the standard local linear estimator with plug-in bandwidth selection are explored, are reported in Tables C.1 to C.12 in Appendix C. Guidelines for reading Appendix C are given in Section 5.3.

The results of Simulation Study C are now discussed and conclusions are drawn.

1. **The effect of bagging and bragging:**
   Three bagging as well as three bragging methods are applied to the plug-in bandwidth selection of h, used in the local linear estimator. The aim of the implementation of the bagging and bragging methods is to determine if these methods can reduce the variance of the local linear estimator with plug-in bandwidth selection. The results contained in Tables C.1 to C.12 enable the following observations regarding the effect of the bagging and bragging procedures as improvement methods on the standard local linear estimator with plug-in bandwidth selection:

   - For Model 1, the *bagged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *nine* simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, Bag3.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, Bag3.
     - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, Bag3.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, Bag3.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Bag3.
   - For Model 1, the *bragged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *three* simulation setup scenarios:
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, Brag3.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, $n = 50$, Brag3.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 1.0$, $n =$, Brag3.
   - For Model 2, the *bagged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *fifteen* simulation setup scenarios:

- $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6, 1.0$, $n = 50$, Bag3.
- $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6, 1.0$, $n = 50$, Bag3.
- $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 50$, Bag3.
- $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 50, 100$, Bag3.
- $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50, 100$, Bag3.
- $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Bag3.
- $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 50, 100, 200$, Bag3.
- $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 50, 100$, Bag3.

- For Model 2, the *bragged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *twelve* simulation setup scenarios:
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6, 1.0$, $n = 50$, Brag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 50, 100$, Brag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50, 100$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 50, 100, 200$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 50, 100$, Brag3.
- For Model 3, the *bagged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *nineteen* simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 100$, Bag2.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1, 0.3$, $n = 50$, Bag3.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50$, Bag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50, 100$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 200$, Bag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50, 100$, Bag3.
- For Model 3, the *bragged* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *seventeen* simulation setup scenarios:
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 100$, Brag3.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50$, Brag3.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50, 100$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 200$, Brag3.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1, 0.3, 0.5$, $n = 50, 100$, Brag3.

Remarks:

- In the above cases where the *bagged* or *bragged* local linear estimator with plug-in bandwidth selection performed better than the standard local linear estimator with plug-in bandwidth selection in terms of

the MISE, it was due to a smaller variance component. Hence, bagging and bragging indeed reduced the variance of the local linear estimator with plug-in bandwidth selection.

- From the information stated above, it is clear that Bag3 and Brag3 are the most effective improvement methods. The Bag3 and Brag3 algorithms obtain estimates of $m(x)$ for each bootstrap sample separately and then aggregate these estimates. The aim of the Bag3 and Brag3 methods is to reduce the variability of $\hat{m}(x)$.
- From the above observations, it is clear that the bagging and bragging methods are the most effective, in terms of the MISE, in small sample sizes.

2. The effect of boosting:

Boosting is implemented as a bias reduction technique. The results contained in Tables C.1 to C.12 enable the following observations regarding the overall effect of the boosting procedure as an improvement method on the standard local linear estimator with plug-in bandwidth selection:

- For Model 1, the *boosted* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *eight* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, 100, 200.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 200.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, 200.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 50$.
- For Model 2, the *boosted* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *six* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 200.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 50$, 200.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 50$.
- For Model 3, the *boosted* local linear estimator has a smaller MISE, compared to the MISE of the *standard* local linear estimator, for the following *three* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 50$.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, $n = 50$.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 50$.

Remarks:

- For all three models, in all *thirty-six* simulation setup scenarios, the boosting procedure led to a smaller integrated squared bias value compared to the integrated square bias of the standard local linear estimator with plug-in bandwidth selection. Boosting indeed reduced the bias of the standard local linear estimator with plug-in bandwidth selection.

- Unfortunately, the boosting improvement method is not effective in terms of the MISE. In most of the simulation setup scenarios, there was a significant increase in the integrated squared variance value. This increase in the variance component resulted in the MISE value for the boosted local linear estimator with plug-in bandwidth to be larger than the MISE value for the standard local linear estimator with plug-in bandwidth selection.
- There were a few cases where the boosted local linear estimator had a smaller MISE, compared to the MISE of the standard local linear estimator. From the above it is clear that the boosting method only improved the MISE value for small error variance values.

3. **The influence of the sample size, error variance, kernel and the covariate distribution:** With regards to the sample size n, the error variance σ, the kernel and the covariate distribution, the following observations can be made when studying Tables C.1 to C.12:

   - The effect of the sample size:
     - In most of the cases an increase in the sample size value n, involved a decrease in the MISE, integrated bias and integrated variance values. In general, larger samples will give a better indication of the underlying relationship between the X variable and the Y variable. Large samples contain more information about the true underlying regression relationship $m(x)$ than small samples and will therefore result in estimators with less bias and variability.
   - The effect of the error variance:
     - An increase in the error variance led to an increase in the values of the discrepancy measures. This is to be expected, since the underlying regression relationship $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance, than from data that follows $m(x)$ closely, i.e, small error variance.
   - The effect of the kernel:
     - By using the $N(0,1)$ kernel with longer tails, smaller MISE, integrated squared bias and integrated variance values are obtained, compared to the restricted triangular kernel.
   - The effect of the covariate distribution:
     - From Tables C.1 to C.12 it is evident that the underlying distribution of the covariate data influences the values of the discrepancy measures. Simulations where $X \sim U(-2,2)$ produce smaller MISE values than simulations where $X \sim N(0,1)$. This phenomenon can be ascribed to an increased boundary effect. More specifically, when X is drawn from the $N(0,1)$ distribution, less observations are drawn for values near the boundaries of the x-grid than when X is drawn from the $U(-2,2)$ distribution, which increases the boundary effect. This could result in larger MISE values.

5.9.1 Conclusions drawn for the local linear estimator with plug-in bandwidth selection

Answers are now provided for the questions stated in Section 5.2.3

a) Bagging reduced the variance component of the local linear estimator with plug-in bandwidth selection only in limited cases.

b) In most cases Bag3 performed the best in terms of producing lower MISE values than the standard local linear estimator with plug-in bandwidth selection.

c) Bragging reduced the variance component of the local linear estimator with plug-in bandwidth selection only in limited cases.

d) In most cases Brag3 performed the best in terms of producing lower MISE values than the standard local linear estimator with plug-in bandwidth selection.

e) Boosting always produces smaller integrated squared bias values compared to the standard local linear estimator with plug-in bandwidth selection.

f) However, there are only limited cases where boosting improved the local linear estimator with plug-in bandwidth selection, in terms of the MISE values. In general the boosted local linear estimator with plug-in bandwidth selection performed better than the standard local linear estimator, in terms of the MISE, for small error variance and small sample sizes.

g) The MISE decreases as the sample size increases. The is to be expected, since the underlying regression relationship $m(x)$ will be more evident from large samples carrying a lot of information, than smaller samples, which are used to estimate $m(x)$. Increasing values of the error variance, σ, resulted in larger MISE values, since $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance, compared to data generated to follow $m(x)$ closely, i.e. data constructed with small error variance. $N(0,1)$ kernel performs marginally better than the triangular kernel with regard to minimising the MISE. The underlying distribution of the covariate data greatly influenced the MISE values. Smaller MISE values were observed when $X \sim U(-2,2)$ compared to $X \sim N(0,1)$.

5.10 Observations and conclusions for the BRNP estimator with cross-validation bandwidth selection

- Yao's (2012) bias reduction nonparametric regression estimator (BRNP) is evaluated in a very limited simulation study, since various computational difficulties arose. Yao (2012) states that the proposed bias reduction nonparametric regression is not adaptive to non-uniform design. Therefore, the BRNP estimator is only tested for $X \sim U(-2,2)$. Yao only used the Gaussian kernel, hence only the Gaussian kernel is used in this study. The BRNP estimator is therefore tested for three models, three sample sizes and three error variances. This results in nine simulation setup scenarios for each model.

- The first aim of simulation study D is to establish if the *improvement methods* provide better estimators for the mean regression curve than the *standard BRNP estimator with cross-validation bandwidth selection*. The bagged, bragged and boosted BRNP estimator with plug-in bandwidth selection seems to be new contributions to the literature since no reference in the literature was found.

- The effects of the sample size n and the error variance σ on the performance of the nonparametric regression estimator involved in this simulation setup, are determined.

- The results of SimD, where the effect of bagging, bragging and boosting on the standard BRNP estimator with cross-validation bandwidth selection are explored, are reported in Tables D.1 to D.3 in Appendix D. Guidelines for reading Appendix D are given in Section 5.3.

- Note that v_j defined in (4.47) converges to $\{m(x_j)f(x_j)\}^{-1}$. Hence if $m(x_j)$ is close to zero, there will be some numerical issues to calculate v_j^{-1}. Yao (2012) recommended to add (or subtract) a large enough positive constant to all the responses to make all the transformed responses well above (or below) zero when the original response Y_i's change signs. Figure 5.1 is a graphical presentation of the effect of the untransformed and transformed responses on the cross-validation function. If the response variable is left unchanged it creates a difficulty to identify the bandwidth value that minimises the cross-validation function. In the case of the transformed response, it is easy to identify the bandwidth value that minimises the cross-validation function. Yao suggested to add $c + \max\{\text{abs}(Y_1), (Y_2), \ldots, (Y_n)\}$, for some constant c, to all the response values.

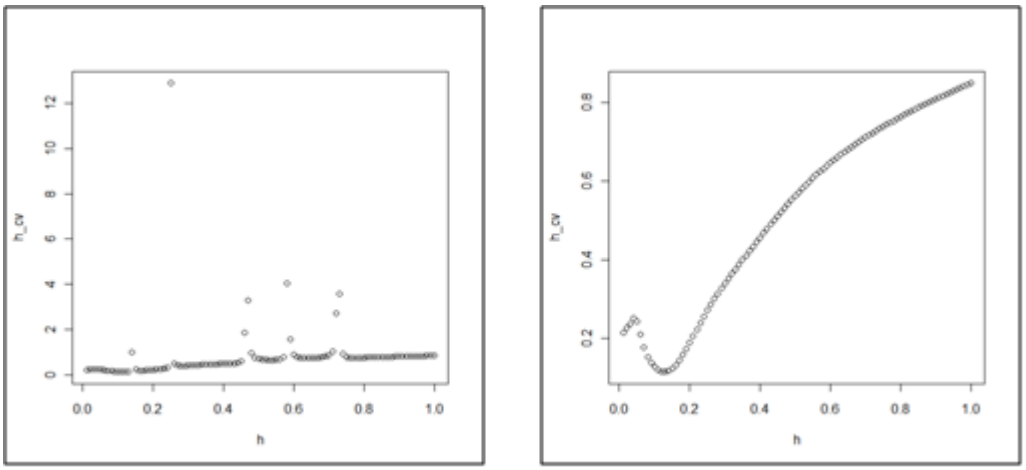

Figure 5.1: The cross-validation function before and after transforming the response variable

The results of Simulation Study D are now discussed and conclusions are drawn.

1. **The effect of bagging and bragging:**

   The aim of the implementation of the bagging and bragging methods is to determine if these methods can reduce the variance of the BRNP estimator with cross-validation bandwidth selection. Three bagging

as well as three bragging methods are applied to the cross-validation bandwidth selection of h, used in the BRNP estimator. The results contained in Tables D.1 to D.3 enables the following observations regarding the effect of the bagging and bragging procedures as improvement methods on the standard BRNP estimator with cross-validation bandwidth selection:

- For Model 1, the *bagged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for *one* simulation setup scenario:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, Bag1.
- For Model 1, the *bragged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for *one* simulation setup scenario:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, Brag1.
- For Model 2, the *bagged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *nine* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200, Bag1.
- For Model 2, the *bragged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *nine* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200, Brag1.
- For Model 3, the *bagged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *seven* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 50$, 100, 200, Bag1.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 200$, Bag3.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 200$, Bag3.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 100, Bag1.
- For Model 3, the *bragged* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *six* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 50$, 100, 200, Brag1.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 200$, Brag1.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 200, Brag1.

Remarks:

- In the above cases where the *bagged or bragged* BRNP estimator with cross-validation bandwidth selection performed better than the *standard* BRNP estimator with cross-validation bandwidth selection in terms of the MISE, it was due to a smaller variance component. Hence, bagging and bragging indeed reduced the variance of the BRNP estimator with cross-validation bandwidth selection.

- From the information stated above, it is clear that Bag1 and Brag1 are the most effective improvement methods. The bagging and bragging methods are the most effective, in terms of the MISE, for Model 2 and Model 3, for almost all choices of sample size and error variance.

2. The effect of boosting:

The aim of the implementation of boosting is to determine if this improvement method can reduce the bias of the BRNP estimator with cross-validation bandwidth selection. The results contained in Tables D.1 to D.3 enable the following observations regarding the overall effect of the boosting procedure as an improvement method on the standard BRNP estimator with cross-validation bandwidth selection:

- For all three models, for the simulation setup scenarios, the *boosted* BRNP estimator has a smaller integrated squared bias value compared to the integrated squared bias value of the *standard* BRNP estimator. The only exception is for Model 3, $X \sim U(-2, 2)$, $\sigma = 0.5$ and $n = 100$.
- For Model 1, the *boosted* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *seven* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 50$, 100, 200.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
- For Model 2, the *boosted* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *nine* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
- For Model 3, the *boosted* BRNP estimator has a smaller MISE, compared to the MISE of the *standard* BRNP estimator, for the following *eight* simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 200.

Remarks:

- Boosting was very effective in reducing the bias component of the BRNP estimator.
- In most cases boosting improved the BRNP estimator with cross-validation bandwidth selection, in terms of the MISE values.

3. The influence of the sample size and error variance:

With regards to the sample size n and the error variance σ, the following observations can be made when studying Tables D.1 to D.3:

- The effect of the sample size:
  - In most of the cases an increase in the sample size value n involved a decrease in the MISE, integrated bias and integrated variance values. In general, larger samples will give a better

indication of the underlying relationship between the X variable and the Y variable. Large samples contain more information about the true underlying regression relationship $m(x)$ than small samples and will therefore result in estimators with less bias and variability.

- The effect of the error variance:
  - An increase in the error variance led to an increase in the values of the discrepancy measures. This is to be expected, since the underlying regression relationship $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance, than from data that follows $m(x)$ closely.

5.10.1 Conclusion drawn for the BRNP estimator with cross-validation bandwidth selection

Answers are now provided for the questions stated in Section 5.2.4

a) Bagging reduced the variance component of the BRNP estimator with cross-validation bandwidth selection only in limited cases.

b) In almost all cases Bag1 performed the best in terms of producing lower MISE values than the standard BRNP estimator with cross-validation bandwidth selection.

c) Bragging reduced the variance component of the BRNP estimator with cross-validation bandwidth selection only in limited cases.

d) In almost all cases Brag1 performed the best in terms of producing lower MISE values than the standard BRNP estimator with cross-validation bandwidth selection.

e) Boosting always produces smaller integrated squared bias values compared to the standard BRNP estimator with cross-validation bandwidth selection.

f) In most cases boosting improved the BRNP estimator with cross-validation bandwidth selection, in terms of the MISE values.

g) The MISE decreases as the sample size increases. The is to be expected, since the underlying regression relationship $m(x)$ will be more evident from large samples carrying a lot of information, than smaller samples, which are used to estimate $m(x)$. Increasing values of the error variance, σ, resulted in larger MISE values, since $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance, compared to data generated to follow $m(x)$ closely, i.e. data constructed with small error variance.

5.11 Observations and conclusions for comparing SimA to SimB

- In this section the *Nadaraya-Watson estimator with cross-validation bandwidth selection* is compared to the *Nadaraya-Watson estimator with plug-in bandwidth selection*, the results are shown in Appendix E.

- In order to compare the results of the Nadaraya-Watson estimator with cross-validation bandwidth selection to the results of the Nadaraya-Watson estimator with plug-in bandwidth selection, a ratio of the obtained MISE values is used:

$$\text{ratio} = \frac{\text{MISE}_{NW_{CV}}}{\text{MISE}_{NW_{plug}}} \tag{5.3}$$

- If this ratio is larger than one, it implies the Nadaraya-Watson estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.

- If the ratio is smaller than one, the Nadaraya-Watson estimator with cross-validation bandwidth selection has the smaller MISE value for a specific simulation scenario.

- The aim of comparing these two estimators is to establish if the Nadaraya-Watson estimator with plug-in bandwidth selection provides better estimators for the mean regression curve than the Nadaraya-Watson estimator with cross-validation bandwidth selection.

- The results of the comparisons are reported in Tables E.1 to E.12 in Appendix E. Guidelines for reading Appendix E are given in Section 5.4.

The results of the comparisons are now discussed and conclusions are drawn.

1. **The standard Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the standard Nadaraya-Watson estimator with plug-in bandwidth selection:**
   For the standard case, i.e. no improvement method is applied to the estimator, the Nadaraya-Watson estimator with plug-in bandwidth selection performs better, regarding the MISE, in most simulation setup scenarios, than the Nadaraya-Watson estimator with cross-validation bandwidth selection.

   - For Model 1, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-seven* (75%) simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
     - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100, 200.
     - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 200.
   - For Model 2, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-five* (69%) simulation setup scenarios:

- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
- $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
- $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
- $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.

- For Model 3, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-nine* (81%) simulation setup scenarios:
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
  - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 200.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, 100, 200.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.5$, $n = 50$, 200.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 100$.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, $n = 100$.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
- From the above it seems that the standard Nadaraya-Watson estimator with plug-in bandwidth selection performs better than the standard Nadaraya-Watson estimator with cross-validation bandwidth selection for large error variance.

2. **The boosted Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the boosted Nadaraya-Watson estimator with plug-in bandwidth selection:**
   The boosted Nadaraya-Watson estimator with plug-in bandwidth selection performs better, regarding the MISE, in most simulation setup scenarios, than the boosted Nadaraya-Watson estimator with cross-validation bandwidth selection.

   - For Model 1, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-nine* (81%) simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 50$, 100, 200.
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, 0.6, $n = 50$, 100, 200.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.

  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 200.

- For Model 2, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *thirty-two* (89%) simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.2$, 0.6, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6, 1.0$, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2, 0.6, 1.0$, $n = 50$, 100, 200.

- For Model 3, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *thirty* (83%) simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.5$, $n = 50$.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, $n = 50$.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 100$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.

3. **The bagged Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the bagged Nadaraya-Watson estimator with plug-in bandwidth selection:**
   In the case of the bagging methods, the Nadaraya-Watson estimator with plug-in bandwidth does not consistently outperform the Nadaraya-Watson estimator with cross-validation bandwidth selection.

   - Bag1
     - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *seventeen* (47%) simulation setup scenarios:

  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6$, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, 0.6, $n = 200$.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, 0.6, $n = 50$, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 200$.

- For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *thirteen* (36%) simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 200$.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6$, $n = 200$.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6, 1.0$, $n = 50$, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, $n = 200$.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.

- For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-three* (64%) simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 50$, 100.
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 100$.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.5$, $n = 50$, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.1$, $n = 100$.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.5$, $n = 200$.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 200.

- Bag 2
  - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-four* (67%) simulation setup scenarios:
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, 0.6, $n = 200$.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 200.
    * $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 200.
  - For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *nineteen* (53%) simulation setup scenarios:
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, $n = 100$, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 200$.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
    * $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, 200.
    * $X \sim N(0, 1)$, triangular kernel, $\sigma = 1.0$, $n = 200$.
  - For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-seven* (75%) simulation setup scenarios:
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
    * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, 100, 200.
    * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.5$, $n = 50$, 200.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.1$, $n = 100$.
    * $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.3$, $n = 50$, 100, 200.

  - $X \sim N(0,1)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.5$, $n = 50$, 200.

- Bag 3
  - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *twenty-four* (67%) simulation setup scenarios:
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, 200.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6$, $n = 50$, 100, 200.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
    - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 50$, 100, 200.
    - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 200.
    - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 200.
  - For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *seventeen* (47%) simulation setup scenarios:
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, triangular kernel, $\sigma = 1.0$, $n = 50$, 200.
    - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 200$.
    - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, 200.
    - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 200$.
  - For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* Nadaraya-Watson estimator with cross-validation bandwidth selection, for the following *eighteen* (50%) simulation setup scenarios:
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.1$, 0.3, $n = 50$, 100, 200.
    - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.5$, $n = 200$.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, $n = 50$, 100, 200.
    - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.5$, $n = 200$.
    - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, 0.5, $n = 200$.
    - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, $n = 100$.
    - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 200$.

Note that the results for the bragged Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the bragged Nadaraya-Watson estimator with plug-in bandwidth selection is not shown, since it is the same as the results of the bagged estimators.

5.11.1 Conclusion drawn for comparing SimA to SimB

Answers are now provided for the question stated in Section 5.2.5

- Recall that for the three models there are *thirty-six* simulation setup scenarios, i.e. three sample sizes, three error variances, two kernels and two covariate distributions. Table 5.3 provides a summary of the percentage of times the Nadaraya-Watson with cross-validation bandwidth selection (NW_{CV}) performed the best in terms of the MISE and the percentage of times the Nadaraya-Watson estimator with plug-in bandwidth selection (NW_{PLUG}) performed the best in terms of the MISE.

	Model 1		Model 2		Model 3	
	NW_{CV}	NW_{PLUG}	NW_{CV}	NW_{PLUG}	NW_{CV}	NW_{PLUG}
Standard	25%	75%	31%	69%	19%	81%
Bag1	53%	47%	64%	36%	36%	64%
Bag2	33%	67%	47%	53%	25%	75%
Bag3	33%	67%	53%	47%	50%	50%
Boost	19%	81%	11 %	89%	17%	83%

Table 5.3: Comparison of the performance of NW_{CV} and NW_{PLUG} for the thirty-six simulation scenarios

- From Table 5.3 it is clear that in the standard case, i.e. no improvement method is applied to the estimator, the Nadaraya-Watson estimator with plug-in bandwidth selection outperforms the Nadaraya-Watson with cross-validation bandwidth selection in terms of the MISE.

- For all three models the boosted Nadaraya-Watson estimator with plug-in bandwidth selection outperformed the Nadaraya-Watson estimator with cross-validation in terms of the MISE.

- In the case of the bagging method one estimator does not consistently outperform the other estimator in terms of the MISE. Note that the results for the bragged Nadaraya-Watson estimator with cross-validation bandwidth selection compared to the bragged Nadaraya-Watson estimator with plug-in bandwidth selection is not shown, since it is the same as the results of the bagged estimators.

5.12 Observations and conclusions for comparing SimB to SimC

- In this section the *local linear estimator with plug-in bandwidth selection* is compared to the *Nadaraya-Watson estimator with plug-in bandwidth selection*, the results are shown in Appendix F.

- In order to compare the results of the local linear estimator with plug-in bandwidth selection to the results of the Nadaraya-Watson estimator with plug-in bandwidth selection, a ratio of the obtained MISE values is used:

$$\text{ratio} = \frac{\text{MISE}_{LL_{PLUG}}}{\text{MISE}_{NW_{PLUG}}} \tag{5.4}$$

- If this ratio is larger than one, it implies the Nadaraya-Watson estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.

- If the ratio is smaller than one, the local linear estimator with plug-in bandwidth selection has the smaller MISE value for a specific simulation scenario.

- The aim of comparing these two estimators is to establish if the Nadaraya-Watson estimator with plug-in bandwidth selection provides better estimators for the mean regression curve than the local linear estimator with plug-in bandwidth selection.

- The results of the comparisons are reported in Tables F.1 to F.12 in Appendix F. Guidelines for reading Appendix F are given in Section 5.4. The results of the comparisons are now discussed and conclusions are drawn.

1. **Standard local linear estimator with plug-in bandwidth compared to the standard Nadaraya-Watson estimator with plug-in bandwidth:**
   For the standard case, i.e. no improvement method is applied to the estimator, the Nadaraya-Watson estimator with plug-in bandwidth selection performs better, regarding the MISE, in most simulation setup scenarios, than the local linear estimator with plug-in bandwidth selection.

   - For Model 1, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* local linear estimator with plug-in bandwidth selection, for the following *thirty-three* (92%) simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 100$, 200.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
   - For Model 2, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* local linear estimator with plug-in bandwidth selection, for the following *twenty-five* (69%) simulation setup scenarios:
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
     - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
     - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6, 1.0$, $n = 50$, 100, 200.

- $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, 0.6, $n = 100$.
- $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
- $X \sim N(0,1)$, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.

- For Model 3, the *standard* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *standard* local linear estimator with plug-in bandwidth selection, for the following *twenty-eight* (78%) simulation setup scenarios:
  - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.3$, 0.5 $n = 50$, 100, 200.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.1$, $n = 50$, 100.
  - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.3$, $n = 50$, 100.
  - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.5$, $n = 50$, 100, 200.
  - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.1$, 0.3, 0.5, $n = 50$, 100, 200.

2. **Boosted local linear estimator with plug-in bandwidth compared to the boosted Nadaraya-Watson estimator with plug-in bandwidth:**
   The boosted Nadaraya-Watson estimator with plug-in bandwidth selection performs better, regarding the MISE, in most simulation setup scenarios, than the boosted local linear estimator with plug-in bandwidth selection.

   - For Model 1, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* local linear estimator with plug-in bandwidth selection, for the following *thirty-five* (97%) simulation setup scenarios:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim U(-2,2)$, $X \sim N(0,1)$, $N(0,1)$ kernel, triangular kernel $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
   - For Model 2, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* local linear estimator with plug-in bandwidth selection, for the following *thirty-four* (94%) simulation setup scenarios:
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$, 200.
     - $X \sim U(-2,2)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim U(-2,2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.2$, $n = 100$,200.
     - $X \sim N(0,1)$, $N(0,1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - $X \sim N(0,1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
   - For Model 3, the *boosted* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *boosted* local linear estimator with plug-in bandwidth selection, for the following *thirty-four* (94%) simulation setup scenarios:

- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 100$, 200.
- $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
- $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
- $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 100$, 200.
- $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
- $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.

3. **Bagged local linear estimator with plug-in bandwidth compared to the bagged Nadaraya-Watson estimator with plug-in bandwidth:**
   The bagged Nadaraya-Watson estimator with plug-in bandwidth selection performs better, regarding the MISE, in most simulation setup scenarios, than the bagged local linear estimator with plug-in bandwidth selection.

   - Bag 1
     - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *thirty-four* (94%) simulation setup scenarios:
       * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
       * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
       * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100.
       * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
       * $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
     - For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-five* (69%) simulation setup scenarios:
       * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
       * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
       * $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
       * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
       * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
       * $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
       * $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, 200.
       * $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
     - For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-nine* (81%) simulation setup scenarios:
       * $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.

  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.1$, $n = 50$, 100.
  - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
  - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, 0.3, 0.5, $n = 50$, 100, 200.

- Bag 2
  - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *thirty-two* (89%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
  - For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-four* (67%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
  - For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-six* (72%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.1$, $n = 100$.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 100$,200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.

- Bag 3
  - For Model 1, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-six* (72%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.2$, $n = 50$, 100.
    - $X \sim U(-2, 2)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.2$, 0.6, 1.0, $n = 50$, 100, 200.
  - For Model 2, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-three* (64%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 0.6$, $n = 100$, 200.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel, $\sigma = 1.0$, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.6$, 1.0, $n = 100$, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.2$, $n = 100$, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.6$, 1.0, $n = 50$, 100, 200.
  - For Model 3, the *bagged* Nadaraya-Watson estimator with plug-in bandwidth selection has a smaller MISE, compared to the MISE of the *bagged* local linear estimator with plug-in bandwidth selection, for the following *twenty-six* (72%) simulation setup scenarios:
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.1$, $n = 100$.
    - $X \sim U(-2, 2)$, $N(0, 1)$ kernel, triangular kernel $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.3$, $n = 100$, 200.
    - $X \sim N(0, 1)$, $N(0, 1)$ kernel, $\sigma = 0.5$, $n = 50$, 100, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.1$, $n = 100$, 200.
    - $X \sim N(0, 1)$, triangular kernel, $\sigma = 0.3$, 0.5, $n = 50$, 100, 200.

Note that the results for the bragged local linear estimator with plug-in bandwidth selection compared to the bragged Nadaraya-Watson estimator with plug-in bandwidth selection is not shown, since it is the same as the results of the bagged estimators.

5.12.1 Conclusions drawn for comparing SimB to SimC

Answers are now provided for the question stated in Section 5.2.6

- Recall that for the three models there are *thirty-six* simulation setup scenarios, i.e. three sample sizes, three error variances, two kernels and two covariate distributions. Table 5.4 provides a summary of the percentage of times the local linear estimator with plug-in bandwidth selection (LL_{PLUG}) performed the best in terms of the MISE and the percentage of times the Nadaraya-Watson estimator with plug-in bandwidth selection (NW_{PLUG}) performed the best in terms of the MISE.

- From Table 5.4 it is clear that in the standard case, i.e. no improvement method is applied to the estimator, the Nadaraya-Watson estimator with plug-in bandwidth selection outperforms the local linear estimator with plug-in bandwidth selection in terms of the MISE.

- For all three models the boosted Nadaraya-Watson estimator with plug-in bandwidth selection outperformed the boosted local linear estimator with plug-in bandwidth selection in terms of the MISE.

- For all three models the bagged Nadaraya-Watson estimator with plug-in bandwidth selection outperformed the bagged local linear estimator with plug-in bandwidth selection in terms of the MISE. Note that the results for the bragged local linear estimator with plug-in bandwidth selection compared to the bragged Nadaraya-Watson estimator with plug-in bandwidth selection is not shown, since it is the same as the results of the bagged estimators.

	Model 1		Model 2		Model 3	
	LL_{PLUG}	NW_{PLUG}	LL_{PLUG}	NW_{PLUG}	LL_{PLUG}	NW_{PLUG}
Standard	8%	92%	31%	69%	22%	78%
Bag1	6%	94%	31%	69%	19%	81%
Bag2	11%	89%	33%	67%	28%	72%
Bag3	28%	72%	36%	64%	28%	72%
Boost	3%	97%	6%	94%	6%	94%

Table 5.4: Comparison of the performance of LL_{PLUG} and NW_{PLUG} for the thirty-six simulation scenarios

5.13 Discussion of computation time

Appendix G shows the computation time, in seconds, for the standard nonparametric regression estimators and the improved nonparametric regression estimators. Guidelines when consulting Appendix G are given in Section 5.5.

- The first aim of this section is to confirm that the main advantage of utilising a nonparametric regression estimator with plug-in bandwidth selection, is the computation time.

- Secondly, to support the choice of only twenty Monte Carlo iterations for Yao's estimator.

- Tables G.1 to G.5 provide the computation time (in seconds) for 200 Monte Carlo iterations, for Model 1, for a limited number of simulation setup scenarios is given. Here one can clearly see that the Nadaraya-

Watson with plug-in bandwidth selection (NW_{PLUG}) and the local linear estimator with plug-in bandwidth selection (LL_{PLUG}) outperform the Nadaraya-Watson estimator with cross-validation bandwidth selection (NW_{CV}) in terms of computation time.

- In Section 5.11 it was shown that the Nadaraya-Watson estimator with plug-in bandwidth selection performs just as good as the Nadaraya-Watson estimator with cross-validation bandwidth selection in terms of the MISE. In Section 5.12 it was shown that the Nadaraya-Watson estimator with plug-in bandwidth selection outperforms the local linear estimator with plug-in bandwidth selection, in terms of the MISE.

- It is clear that a nonparametric regression estimator utilising plug-in bandwidth selection methods is computationally more efficient than a nonparametric regression estimator utilising cross-validation bandwidth selection methods.

- In this section it is confirmed that the Nadaraya-Watson estimator with plug-in bandwidth selection is a nonparametric regression estimator to consider, since it is not only a fairly precise and accurate estimator, but it is also computationally a lot faster than other nonparametric regression estimators considered in this study.

- Table G.6 presents the computation time for Yao's estimator for twenty Monte Carlo iterations, for Model 1, for a limited number of simulation setup scenarios. From this it is clear that Yao's estimator is a very computer time intensive estimator compared to the other three estimators.

- In future studies a plug-in bandwidth selection method for Yao's estimator needs to be considered to improve the computation time of this estimator.

5.14 Discussion of the graphs

In Appendix H a limited number of graphs are presented to graphically illustrate general aspects of the simulation studies, such as the bias and variance behaviour of the various methods. Guidelines for reading Appendix H are given in Section 5.6. The presented graphs are now briefly explained. In particular, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$ and $n = 100$ scenario is considered for Model 1 and Model 2 respectively. The specific choice of scenario is arbitrary, any other scenario could have been chosen for this illustration. The graphs are presented as follow:

- Figures H.1 to H.2 illustrate the results for the Nadaraya-Watson estimator with cross-validation bandwidth selection. These are graphical representations of the results displayed in Appendix A, Table A.1 and Table A.5 respectively.

- Figures H.3 to H.4 illustrate the results for the Nadaraya-Watson estimator with plug-in bandwidth selection. These are graphical representations of the results displayed in Appendix B, Table B.1 and Table B.5 respectively.

- Figures H.5 to H.6 illustrate the results for the local linear estimator with plug-in bandwidth selection. These are graphical representations of the results displayed in Appendix C, Table C.1 and Table C.5 respectively.

- Figures H.7 to H.8 illustrate the results of the BRNP estimator with cross-validation bandwidth selection. These are graphical representations of the results displayed in Appendix D, Table D.1 and Table D.2 respectively.

For each of the above mentioned scenarios, two types of graphs are drawn. Firstly, each procedure is illustrated in a graph where the estimated curves for each of the Monte Carlo samples are drawn on one figure. These graphs give insight in the variance involved for the particular method. Secondly, the procedures are illustrated in a graph where the average estimator over the Monte Carlo samples is presented for various procedures. These graphs show the effect of the various methods on the bias. The graphs in Figures H.1 to H.8 confirm the conclusions drawn from Simulation Study A to D. More specifically, the following conclusions are particularly clear from the graphs:

- The boosted estimators produce curves that are slightly closer to the underlying regression curve $m(x)$ than the standard estimators, i.e. the bias is reduced.

- The bagged estimators produce smoother curves, i.e. curves with less variability, than the standard estimators.

- As n increases, the estimates lie closer to $m(x)$ (i.e. the integrated bias decreases) and the variability becomes less (i.e. the integrated variance decreases).

5.15 Final remarks and recommendations

This study focused on bias reduction techniques in nonparametric kernel regression. The performance of the bias reduction methods were empirically determined in a variety of simulation setup scenarios. Specifically boosting as a bias reduction technique was investigated, also a new bias reduction nonparametric estimator proposed by Yao (2012) was introduced. Two variance reduction techniques, bagging and bragging, were also implemented and the effect of these improvement methods on the bias behaviour of the estimator was also observed. The following final remarks and recommendations regarding the study can be made:

- Boosting indeed reduced the bias component of the Nadaraya-Watson estimator with cross-validation bandwidth selection, in all simulation scenarios considered.

- Boosting was not effective in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. The boosted estimator performed better than the standard Nadaraya-Watson estimator with cross-validation bandwidth selection when $X \sim N(0, 1)$ and the error variance is small.

- Boosting indeed reduced the bias component of the Nadaraya-Watson estimator with plug-in bandwidth selection, in all simulation scenarios considered.
- The boosted Nadaraya-Watson estimator performed better than the standard Nadaraya-Watson estimator with plug-in bandwidth selection in terms of the MISE value, only in limited cases. In general, the boosted Nadaraya-Watson estimator performed better than the standard Nadaraya-Watson estimator with plug-in bandwidth selection for small error variance.
- Boosting indeed reduced the bias component of the local linear estimator with plug-in bandwidth selection, in all simulation scenarios considered.
- The boosted local linear estimator performed better than the standard local linear estimator with plug-in bandwidth selection in terms of the MISE value, only in limited cases. In general, the boosted local linear estimator performed better than the standard local linear estimator with plug-in bandwidth selection for small error variance and small sample sizes.
- For the BRNP estimator with cross-validation bandwidth selection, boosting reduced the bias component of the estimator in all simulation scenarios considered.
- In all simulation scenarios the boosted BRNP estimator was very effective in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard BRNP estimator with cross-validation bandwidth selection. This implies that the variance components were not inflated too much.
- However, this effect of boosting needs to be further investigated as this method was tested in a very limited simulation setup scenario due to the computer time intensiveness of the BRNP estimator.
- It is clear that boosting is a very effective method for reducing the bias component of a nonparametric regression estimator.
- Considering the extra computation time that would be needed for extra boosting iterations, the present study allowed for a maximum of only one boosting iteration. In future studies more boosting iterations need to be considered in a more specific boosting project.
- Marzio and Taylor (2008) showed that boosting works properly only if the kernel yields strictly positive characteristic roots. Therefore the present study was restricted to the $N(0,1)$ and triangular kernels. Future studies need to include more kernels when boosting is not a research issue.
- Since the BRNP estimator with cross-validation bandwidth selection is a very computer time intensive estimator, only twenty Monte Carlo iterations were performed. In future research more Monte Carlo iterations need to be performed. Much better bias reduction can be expected.
- The BRNP estimator with plug-in bandwidth selection will be investigated in further research as a method to reduce the computer time intensiveness of the estimator.

- Bagging methods are very effective in reducing the variance component of the Nadaraya-Watson estimator when cross-validation bandwidth selection is used. Specifically, Bag3 performed better than Bag1 and Bag2 in terms of reducing the variance component of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

- Bagging methods are very effective in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection. Specifically Bag1 performed the best, compared to Bag2 and Bag3, in terms of providing an estimator for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with cross-validation bandwidth selection.

- The same observations, in terms of the variance and MISE, hold for the bragging methods, with regard to the Nadaraya-Watson estimator with cross-validation bandwidth selection.

- Bagging and bragging methods were effective in reducing the variance component of the BRNP estimator in limited cases. However, this effect of the bagging and bragging methods need to be further investigated as these methods were tested in a very limited simulation setup scenario due to the computer time intensiveness of the BRNP estimator.

- Bagging and bragging were very effective in reducing the variability of the cross-validation bandwidths and the resulting regression estimates.

- In general the bias component of the estimators with cross-validation bandwidth selection increases when bagging and bragging improvement methods are applied to the estimators.

- Bagged and bragged estimators were not effective when plug-in bandwidth selection methods were used. For the Nadaraya-Watson estimator with plug-in bandwidth selection and the local linear estimator with plug-in bandwidth selection the bagged and bragged estimators provided estimators for the mean regression curve with a MISE value lower than the MISE value of the standard Nadaraya-Watson estimator with plug-in bandwidth selection and the local linear estimator with plug-in bandwidth selection in very limited cases.

- In this study the number of bootstrap samples, used when implementing the bagging and bragging improvement methods, are fixed at $B = 50$. In future studies the effect of the number of bootstrap samples must be investigated. Also, the bootstrap sample size is fixed at $n/2$. This also needs to be investigated further.

- The MISE decreased as the sample size increased. This is to be expected, since the underlying regression relationship $m(x)$ will be more evident from large samples carrying a lot of information, than smaller samples, which are used to estimate $m(x)$.

- Increasing values of the error variance, resulted in larger MISE values, since $m(x)$ will be more difficult to estimate from data that were generated to deviate from $m(x)$ a lot by means of a large error variance compared to data generated to follow $m(x)$ closely, i.e. data constructed with small error variance.

- By using the $N(0,1)$ kernel, in most cases, smaller MISE values are obtained, compared to the triangular kernel. This is to be expected, since the $N(0,1)$ kernel has longer tails and therefore more data are included in the estimation process than when using the restricted triangular kernel. Smaller MISE values were obtained when covariate data were from the U(-2,2) distribution than for the scenarios where data were drawn from the $N(0,1)$ distribution. This event can be ascribed to an increased boundary effect when data are drawn from the unevenly distributed $N(0,1)$ distribution. More specifically, data drawn from a $N(0,1)$ distribution contain less observations near the boundary of the grid of x values. This results in large variability and bias near the boundary points.

Bibliography

Akaike, H. (1970). Statistical predictor identification, *Annals of the institute of statistical mathematics* **22**: 203–217.

Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions of Automatic Control* **19**: 716–723.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician* **46**(3): 175–185.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: bagging, boosting and variants, *Machine Learning* **36**(1/2): 105–139.

Benedetti, J. K. (1977). On the nonparametric estimation of regression functions, *Journal of the Royal Statistical Society, Series B* **39**(2): 248–253.

Boos, D. D. (2003). Introduction to the bootstrap world, *Statistical Science* **18**(2): 168–174.

Boshoff, L. (2009). *Boosting, bagging, and bragging applied to nonparametric regression-an empirical approach*, Master's thesis, Potchefstroom: NWU.

Breiman, L. (1996a). Bagging predictors, *Machine Learning* **24**: 123–140.

Breiman, L. (1996b). Heuristics of instability and stabilization in model selection, *The Annals of Statistics* **24**(6): 2350–2383.

Breiman, L. (1996c). Out-of-bag estimation, *Technical report*, Statistics Department, University of California.

Breiman, L. (1999). Prediction games and arcing algorithms, *Neural Computation* **11**(7): 1493–1517.

Bühlmann, P. (2003). Bagging, subagging and bragging for improving some prediction algorithms, *in* M. G. Akritas and D. N. Politis (eds), *Recent Advances and Trends in Nonparametric Statistics*, Elsevier, Amsterdam, pp. 9–34.

Bühlmann, P. (2004). Bagging, boosting and ensemble methods, *in* J. E. Gentle, W. Härdle and Y. Mori (eds), *Handbook of Computational Statistics: Concepts and Methods*, Springer, Berlin, pp. 877–907.

Bühlmann, P. (2006). Boosting for high-dimensional linear models, *The Annals of Statistics* **34**(2): 559–583.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: regularization, prediction and model fitting, *Statistical Science* **22**(4): 477–505.

Bühlmann, P. and Yu, B. (2002). Analyzing bagging, *The Annals of Statistics* **30**(4): 927–961.

Bühlmann, P. and Yu, B. (2003). Boosting with the l_2 loss: Regression and classification, *Journal of the American Statistical Association* **98**(462): 324–339.

Cheng, P. E. and Cheng, K. F. (1987). Robust nonparametric estimation of a regression function, *Sankhya: The Indian Journal of Statistics, Series B* **49**(1): 9–22.

Chu, C.-K. and Marron, J. S. (1991). Choosing a kernel regression estimator, *Statistical Science* **6**(4): 404–419.

Chung, Y. and Lindsay, B. G. (2011). A likelihood-tuned density estimator via nonparametric mixture model, *Nonparametric Statistics and Mixture Models*, World Scientific Publishing, pp. 69–89.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions, *Numerische Mathematik* **31**: 377–403.

Croux, C., Joossens, K. and Lemmens, A. (2007). Trimmed bagging, *Computational Statistics and Data Analysis* **52**: 362–368.

Davidson, R. and Flachaire, E. (2008). The wild bootstrap, tamed at last, *Journal of Econometrics* **146**(1): 162–169.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and their Applications*, Cambridge University Press, Cambridge.

Efron, B. (1979). Bootstrap methods: another look at the jackknife, *The Annals of Statistics* **7**(1): 1–26.

Efron, B. (2003). Second thoughts on the bootstrap, *Statistical Science* **18**(2): 135–140.

Efron, B. and Tibshirani, R. J. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Statistical Science* **1**(1): 54–77.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*, Vol. 57 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, New York.

Eubank, R. L. (1988). *Spline Smoothing and Nonparametric Regression*, Vol. 90 of *Statistics: Textbooks and Monographs*, Marcel Dekker, New York.

Fan, J. (1992). Design-adaptive nonparametric regression, *Journal of the American Statistical Association* **87**(420): 998–1004.

Fan, J. (1993). Local linear regression smoothers and their minimax efficiencies, *The Annals of Statistics* **21**(1): 196–216.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*, Vol. 66 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1): 119–139.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine, *The Annals of Statistics* **29**(5): 1189–1232.

Gasser, T. and Müller, H.-G. (1979). Kernel estimation of regression functions, *in* T. Gasser and M. Rosenblatt (eds), *Smoothing Techniques for Curve Estimation*, Vol. 757 of *Lecture Notes in Mathematics*, Springer-Verlag, Heidelberg, pp. 23–68.

Gasser, T. and Müller, H.-G. (1984). Estimating regression functions and their derivatives by the kernel method, *Scandinavian Journal of Statistics* **11**(3): 171–185.

Gasser, T., Müller, H.-G. and Mammitzsch, V. (1985). Kernels for nonparametric curve estimation., *Journal of the Royal Statistical Society, Series B* **47**(2): 238–252.

Györfi, L., Kohler, M., Kryzak, A. and Walk, H. (2002). *A Distribution-free Theory of Nonparametric Regression*, New York: Springer.

Hall, P. and Robinson, A. P. (2009). Reducing variability of crossvalidation for smoothing-parameter choice, *Biometrika* **96**(1): 175–186.

Hansen, B. (2009). Lecture notes on nonparametrics: Nadaraya-Watson and local linear regression.

Härdle, W. (1987). Algorithm as 222: Resistant smoothing using the fast fourier transform, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **36**(1): 104–111.

Härdle, W. (1990). *Applied Nonparametric Regression*, Cambridge University Press, Cambridge.

Härdle, W. and Marron, J. S. (1985a). Asymptotic nonequivalence of some bandwidth selectors in nonparametric regression, *Biometrika* **72**(2): 481–484.

Härdle, W. and Marron, J. S. (1985b). Optimal bandwidth selection in nonparametric regression function estimation, *The Annals of Statistics* **13**(4): 1465–1481.

Härdle, W. and Müller, M. (2000). Multivariate and semiparametric kernel regression, *in* M. Schimek (ed.), *Smoothing and Regression*, Wiley, New York, pp. 357–391.

Härdle, W., Hall, P. and Marron, J. S. (1988a). How far are automatically chosen regression smoothing parameters from their optimum?, *Journal of the American Statistical Association* **83**(401): 86–95.

Härdle, W., Janssen, P. and Serfling, R. (1988b). Strong uniform consistency rates for estimators of conditional functionals, *The Annals of Statistics* **16**: 1428–1449.

Hinkley, D. (1994). Bootstrap: More than a stab in the dark?, *Statistical Science* **9**(3): 400–403.

Huber, P. J. (1979). Robust smoothing, *in* R. L. Launer and G. N. Wilkinson (eds), *Robustness in Statistics*, Academic Press, New York.

Jacod, J. and Protter, P. (2004). *Probability Essentials*, 2nd edn, Springer-Verlag, Berlin.

Kutner, M. H., Li, W., Nachtsheim, C. J. and Neter, J. (2005). *Applied Linear Statistical Models*, McGraw-Hill, New York.

Liu, R. Y. (1988). Bootstrap procedures under some non-i.i.d models, *The Annals of Statistics* **16**(4): 1696–1708.

Lutz, R. W. and Bühlmann, P. (2006). Conjugate direction boosting, *Journal of Computational and Graphical Statistics* **15**(2): 287–311.

Lutz, R. W., Kalisch, M. and Bühlmann, P. (2008). Robustified l-2 boosting, *Computational Statistics and Data Analysis* **52**: 3331–3341.

Mack, Y. P. and Müller, H.-G. (1989). Convolution type estimators for nonparametric regression, *Statistics and Probability Letters* **7**: 229–239.

Mallows, C. L. (1973). Some comments on c_p, *Technometrics* **15**(4): 661–675.

Mallows, C. L. (1980). Some theory of nonlinear smoothers, *The Annals of Statistics* **8**(4): 695–715.

Mammen, E. (1993). Bootstrap and wild bootstrap for high dimensional linear models, *The Annals of Statistics* **21**(1): 255–285.

Manteiga, W. G., Miranda, M. D. D. and González, A. (2004). The choice of smoothing parameter in nonparametric regression through wild bootstrap, *Computational Statistics and Data Analysis* **47**(1): 487–515.

Marron, J. S. and Nolan, D. (1988). Canonical kernels for density estimation, *Statistics and Probability Letters* **7**(3): 195–199.

Marzio, M. D. and Taylor, C. C. (2008). On boosting kernel regression, *Journal of Statistical Planning and Inference* **138**: 2483–2498.

Meir, R. and Rätsch, G. (2003). An introduction to boosting and leveraging, *in* S. Mendelson and A. Smola (eds), *Advanced Lectures on Machine Learning*, Vol. 2600 of *Lecture Notes in Artificial Intelligence*, Springer, New York, pp. 119–183.

Nadaraya, E. A. (1964). On estimating regression, *Theory of Probability and its Applications* **9**: 141–142.

Owen, A. B. (1987). Nonparametric conditional estimation, *Technical report*, Stanford University, Stanford, California.

Parzen, E. (1962). On estimation of a probability density function and mode, *The Annals of Mathematical Statistics* **33**(3): 1065–1076.

Peterson, M. L., Molinaro, A. M., Sinisi, S. E. and Van der Laan, M. J. (2007). Cross-validated bagged learning, *Journal of Multivariate Analysis* **98**: 1693–1704.

Pino-Mejías, R., Jiménez-Gamero, M.-D., Cubiles-de-la Vega, M.-D. and Pascual-Acosta, A. (2008). Reduced bootstrap aggregation of learning algorithms, *Pattern Recognition Letters* **29**: 265–271.

Priestley, M. B. and Chao, M. T. (1972). Nonparametric function fitting, *Journal of the Royal Statistical Society, Series B* **34**: 385–392.

Rice, J. A. (1984a). Bandwidth choice for nonparametric regression, *The Annals of Statistics* **12**: 1215–1230.

Rice, J. A. (1984b). Boundary modification for kernel regression, *Communications in Statistics: Theory and Methods* **13**(7): 893–900.

Rice, J. A. (2007). *Mathematical Statistics and Data Analysis*, 3rd edn, Thomson Brooks/Cole, Belmont, CA.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function, *The Annals of Mathematical Statistics* **27**(3): 832–837.

Ruppert, D. and Wand, M. (1994). Multivariate locally weighted least squares regression, *The Annals of Statistics* **22**(3): 807–817.

Ruppert, D., Sheather, S. J. and Wand, M. P. (1995). An effective bandwidth selector for local least squares, *Journal of the American Statistical Association* **90**(432): 1257–1270.

Schapire, R. (1990). The strength of weak learnability, *Machine Learning* **5**(2): 197–227.

Schapire, R. E. (1999). A brief introduction to boosting, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*.

Shibata, R. (1981). An optimal selection of regression variables, *Biometrika* **68**: 45–54.

Silverman, B. W. (1984). Spline smoothing: the equivalent variable kernel method, *The Annals of Statistics* **12**(3): 898–916.

Silverman, B. W. and Young, G. A. (1987). The bootstrap: To smooth or not to smooth?, *Biometrika* **74**(3): 469–479.

Skurichina, M. and Duin, R. P. W. (1998). Bagging for linear classifiers, *Pattern Recognition* **31**(7): 909–930.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society, Series B (Methodological)* **36**(2): 111–147.

Stute, W. (1984). Asymptotic normality of nearest neighbor regression function estimates, *The Annals of Statistics* **12**(3): 917–926.

Swanepoel, J. W. H. (1986). A note on proving that the (modified) bootstrap works, *Communications in Statistics: Theory and Methods* **15**(11): 3193–3203.

Swanepoel, J. W. H. (1988). Point estimation based on approximating functionals and the bootstrap, *Technical Report, Dept of Statistics, Potchefstroom University.*

Swanepoel, J. W. H. (1990). A review of bootstrap methods, *South African Statistical Journal* **24**: 1–34.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*, Vol. 60 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Watson, G. S. (1964). Smooth regression analysis, *Sankhyā Series A* **26**: 359–372.

Yao, W. (2012). A bias corrected nonparametric regression estimator, *Statistics and Probability Letters* **82**(2): 274–282.

Zhou, Z. and Yang, Y. (2005). Adapt bagging to nearest neighbor classifiers, *Journal of Computer Science and Technology* **20**(1): 48–54.

Appendix A

Results of Simulation Study A

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.02459	0.00392	0.02067	0.00127	0.05211	0.00774	0.04437	0.00206	0.13039	0.01856	0.11183	0.00423
Bag1	0.02405	0.00404	0.02001	0.00138	0.05207	0.01016	0.04191	0.00135	0.12037	0.03554	0.08483	-0.00539
Bag2	0.02579	0.00743	0.01836	0.00017	0.05963	0.02018	0.03944	0.00163	0.16834	0.07817	0.09017	-0.00119
Bag3	0.02548	0.00732	0.01817	0.00007	0.05923	0.02045	0.03879	0.00029	0.17301	0.07995	0.09306	-0.00739
Brag1	0.02418	0.00412	0.02006	0.00131	0.05189	0.00966	0.04223	0.00154	0.11778	0.02791	0.08987	-0.00224
Brag2	0.02578	0.00688	0.01890	0.00017	0.05674	0.01585	0.04090	-0.00043	0.14574	0.04695	0.09879	-0.00441
Brag3	0.02516	0.00678	0.01838	0.00052	0.05541	0.01610	0.03930	0.00134	0.14472	0.05068	0.09404	-0.00555
Boost	0.02561	0.00034	0.02527	-0.00065	0.05202	0.00115	0.05087	0.00331	0.12286	0.00580	0.11706	0.00952
	$\sigma = 0.6$											
Standard	0.12199	0.02406	0.09793	0.00144	0.23055	0.04877	0.18178	0.00444	0.43598	0.07035	0.36563	0.00398
Bag1	0.11855	0.02897	0.08958	0.00163	0.22887	0.07292	0.15595	0.00585	0.41513	0.15220	0.26293	-0.01080
Bag2	0.12729	0.04632	0.08097	-0.00179	0.25161	0.10961	0.14200	0.00919	0.46408	0.22250	0.24158	-0.00275
Bag3	0.12598	0.04358	0.08240	-0.00274	0.23809	0.09471	0.14338	0.00615	0.41520	0.17512	0.24008	-0.00809
Brag1	0.11854	0.02829	0.09024	0.00176	0.22686	0.06674	0.16013	0.00603	0.41541	0.12120	0.29421	-0.00889
Brag2	0.12655	0.04345	0.08310	-0.00027	0.24565	0.09605	0.14960	0.00597	0.44498	0.16615	0.27884	-0.00532
Brag3	0.12537	0.04303	0.08234	0.00148	0.23944	0.09286	0.14658	0.01139	0.42403	0.16239	0.26165	0.00005
Boost	0.13622	0.00414	0.13208	-0.00091	0.25692	0.01355	0.24337	0.00575	0.48888	0.02662	0.46225	0.00434
	$\sigma = 1.0$											
Standard	0.26776	0.05422	0.21354	0.00015	0.48805	0.11368	0.37438	0.01145	0.85797	0.16571	0.69226	0.01852
Bag1	0.26197	0.08184	0.18013	0.00101	0.47251	0.18913	0.28338	0.01646	0.76445	0.29535	0.46910	0.00211
Bag2	0.28225	0.11821	0.16404	-0.00351	0.51149	0.25281	0.25868	0.02177	0.81079	0.38432	0.42648	0.01051
Bag3	0.27087	0.10108	0.16979	-0.00556	0.47304	0.19967	0.27337	0.01797	0.74988	0.30282	0.44706	-0.00441
Brag1	0.26018	0.07645	0.18373	0.00118	0.47538	0.16699	0.30839	0.01693	0.78390	0.25387	0.53003	0.00312
Brag2	0.27719	0.10820	0.16899	-0.00085	0.50677	0.21896	0.28782	0.01954	0.81701	0.31935	0.49766	0.00792
Brag3	0.27317	0.10507	0.16810	0.00259	0.48863	0.21462	0.27400	0.02173	0.77834	0.32147	0.45687	0.00445
Boost	0.30995	0.01493	0.29502	-0.00237	0.56313	0.04600	0.51713	0.01083	0.99966	0.08667	0.91299	0.01258

Table A.1: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0261	0.0040	0.0221	0.0007	0.0654	0.0083	0.0571	0.0008	0.1743	0.0289	0.1455	0.0039
Bag1	0.0255	0.0043	0.0212	0.0008	0.0578	0.0146	0.0432	0.0011	0.1548	0.0609	0.0940	-0.0011
Bag2	0.0272	0.0079	0.0193	-0.0005	0.0713	0.0280	0.0433	0.0008	0.2261	0.1191	0.1069	-0.0032
Bag3	0.0269	0.0080	0.0189	-0.0011	0.0724	0.0296	0.0429	-0.0035	0.2286	0.1212	0.1074	-0.0153
Brag1	0.0254	0.0042	0.0212	0.0008	0.0576	0.0132	0.0444	0.0011	0.1507	0.0487	0.1020	0.0009
Brag2	0.0269	0.0071	0.0198	-0.0003	0.0651	0.0214	0.0437	-0.0004	0.2046	0.0797	0.1249	-0.0064
Brag3	0.0260	0.0072	0.0188	0.0000	0.0647	0.0223	0.0423	-0.0001	0.1981	0.0829	0.1153	-0.0072
Boost	0.0264	0.0004	0.0260	-0.0013	0.0642	0.0015	0.0627	0.0020	0.1591	0.0104	0.1487	0.0086
	$\sigma = 0.6$											
Standard	0.1240	0.0226	0.1014	0.0011	0.2392	0.0472	0.1920	0.0028	0.4570	0.0693	0.3877	-0.0009
Bag1	0.1180	0.0267	0.0912	0.0014	0.2302	0.0716	0.1586	0.0061	0.4276	0.1668	0.2608	-0.0101
Bag2	0.1261	0.0434	0.0827	-0.0021	0.2540	0.1087	0.1453	0.0087	0.4872	0.2432	0.2440	-0.0018
Bag3	0.1246	0.0409	0.0838	-0.0034	0.2385	0.0933	0.1452	0.0044	0.4337	0.1919	0.2418	-0.0142
Brag1	0.1187	0.0262	0.0925	0.0014	0.2279	0.0635	0.1644	0.0065	0.4184	0.1240	0.2944	-0.0084
Brag2	0.1264	0.0410	0.0854	-0.0009	0.2467	0.0930	0.1538	0.0061	0.4558	0.1728	0.2830	-0.0056
Brag3	0.1241	0.0406	0.0835	0.0007	0.2386	0.0901	0.1485	0.0095	0.4315	0.1704	0.2611	-0.0033
Boost	0.1385	0.0042	0.1343	-0.0010	0.2658	0.0142	0.2516	0.0052	0.5109	0.0281	0.4828	0.0016
	$\sigma = 1.0$											
Standard	0.2746	0.0483	0.2262	-0.0011	0.4991	0.1067	0.3924	0.0138	0.9038	0.1530	0.7509	0.0187
Bag1	0.2630	0.0790	0.1839	0.0004	0.4789	0.1866	0.2923	0.0160	0.7777	0.2955	0.4822	-0.0019
Bag2	0.2838	0.1156	0.1681	-0.0040	0.5186	0.2506	0.2680	0.0213	0.8227	0.3818	0.4409	0.0061
Bag3	0.2695	0.0957	0.1738	-0.0066	0.4728	0.1892	0.2836	0.0157	0.7554	0.2972	0.4583	-0.0098
Brag1	0.2617	0.0709	0.1908	0.0003	0.4794	0.1565	0.3229	0.0164	0.7924	0.2435	0.5489	0.0003
Brag2	0.2779	0.1023	0.1756	-0.0026	0.5085	0.2075	0.3010	0.0194	0.8199	0.3076	0.5123	0.0040
Brag3	0.2715	0.0999	0.1716	0.0016	0.4872	0.2045	0.2827	0.0209	0.7828	0.3061	0.4767	0.0037
Boost	0.3218	0.0140	0.3078	-0.0027	0.5812	0.0447	0.5366	0.0142	1.0542	0.0817	0.9725	0.0198

Table A.2: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 1, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0478	0.0181	0.0297	0.0096	0.1035	0.0333	0.0702	0.0197	0.2960	0.0744	0.2216	0.0270
Bag1	0.0399	0.0185	0.0214	0.0112	0.0836	0.0350	0.0486	0.0173	0.2090	0.0759	0.1330	0.0392
Bag2	0.0509	0.0301	0.0209	0.0144	0.1063	0.0592	0.0471	0.0234	0.2853	0.1537	0.1316	0.0508
Bag3	0.0521	0.0309	0.0212	0.0149	0.1191	0.0665	0.0526	0.0244	0.3330	0.1919	0.1411	0.0527
Brag1	0.0410	0.0165	0.0245	0.0100	0.0846	0.0308	0.0538	0.0174	0.2217	0.0703	0.1514	0.0371
Brag2	0.0506	0.0255	0.0252	0.0132	0.1016	0.0470	0.0546	0.0203	0.2813	0.1115	0.1698	0.0408
Brag3	0.0489	0.0270	0.0219	0.0154	0.1017	0.0504	0.0513	0.0239	0.2758	0.1171	0.1587	0.0438
Boost	0.0330	0.0030	0.0299	-0.0015	0.0797	0.0082	0.0715	-0.0009	0.2552	0.0323	0.2229	0.0000
	$\sigma = 0.6$											
Standard	0.1542	0.0347	0.1194	0.0165	0.2808	0.0533	0.2276	0.0138	0.5443	0.0986	0.4457	0.0665
Bag1	0.1450	0.0383	0.1068	0.0181	0.2601	0.0665	0.1936	0.0183	0.4749	0.1292	0.3456	0.0720
Bag2	0.1506	0.0585	0.0921	0.0237	0.2580	0.1013	0.1567	0.0309	0.4914	0.2138	0.2776	0.0842
Bag3	0.1487	0.0563	0.0924	0.0240	0.2564	0.0967	0.1597	0.0311	0.5087	0.2232	0.2855	0.0831
Brag1	0.1458	0.0360	0.1099	0.0169	0.2636	0.0616	0.2020	0.0168	0.4811	0.1133	0.3678	0.0699
Brag2	0.1535	0.0535	0.1000	0.0218	0.2691	0.0896	0.1795	0.0268	0.5095	0.1681	0.3414	0.0804
Brag3	0.1515	0.0560	0.0955	0.0261	0.2620	0.0928	0.1692	0.0362	0.5124	0.1853	0.3271	0.0978
Boost	0.1733	0.0080	0.1653	-0.0002	0.3274	0.0148	0.3127	-0.0115	0.6028	0.0366	0.5662	0.0233
	$\sigma = 1.0$											
Standard	0.3189	0.0633	0.2556	0.0271	0.6000	0.0965	0.5035	0.0229	1.0488	0.1652	0.8837	0.1276
Bag1	0.3036	0.0788	0.2248	0.0297	0.5231	0.1415	0.3816	0.0313	0.8754	0.2373	0.6380	0.1270
Bag2	0.3075	0.1140	0.1935	0.0388	0.5180	0.1977	0.3203	0.0554	0.8683	0.3518	0.5164	0.1522
Bag3	0.3016	0.1040	0.1976	0.0402	0.5041	0.1721	0.3319	0.0598	0.8827	0.3428	0.5400	0.1395
Brag1	0.3047	0.0747	0.2300	0.0285	0.5318	0.1300	0.4018	0.0309	0.9253	0.1991	0.7262	0.1198
Brag2	0.3129	0.1062	0.2068	0.0358	0.5334	0.1770	0.3564	0.0520	0.9377	0.2800	0.6577	0.1448
Brag3	0.3068	0.1077	0.1991	0.0426	0.5196	0.1802	0.3394	0.0691	0.9451	0.3428	0.6023	0.1685
Boost	0.3819	0.0193	0.3627	0.0037	0.7444	0.0347	0.7098	-0.0228	1.2433	0.0666	1.1767	0.0470

Table A.3: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 1, $X \sim N(0,1), N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0697	0.0270	0.0427	0.0122	0.1958	0.0538	0.1420	0.0174	0.5082	0.1380	0.3703	0.0285
Bag1	0.0574	0.0314	0.0259	0.0139	0.1344	0.0699	0.0645	0.0233	0.3694	0.1458	0.2236	0.0471
Bag2	0.0806	0.0514	0.0292	0.0181	0.1929	0.1194	0.0735	0.0311	0.4904	0.2803	0.2101	0.0651
Bag3	0.0869	0.0552	0.0317	0.0177	0.2197	0.1405	0.0792	0.0320	0.5394	0.3448	0.1946	0.0652
Brag1	0.0550	0.0259	0.0290	0.0118	0.1342	0.0573	0.0769	0.0188	0.3938	0.1322	0.2617	0.0479
Brag2	0.0812	0.0408	0.0403	0.0149	0.1772	0.0874	0.0899	0.0207	0.5251	0.2106	0.3145	0.0462
Brag3	0.0739	0.0418	0.0322	0.0182	0.1777	0.0916	0.0861	0.0226	0.5073	0.2218	0.2854	0.0466
Boost	0.0475	0.0061	0.0414	-0.0028	0.1641	0.0191	0.1449	-0.0070	0.4565	0.0751	0.3814	-0.0048
	$\sigma = 0.6$											
Standard	0.1681	0.0447	0.1234	0.0200	0.3295	0.0860	0.2435	0.0239	0.7197	0.1576	0.5621	0.0663
Bag1	0.1556	0.0561	0.0995	0.0222	0.2945	0.1078	0.1867	0.0269	0.5818	0.1963	0.3855	0.0847
Bag2	0.1746	0.0853	0.0893	0.0286	0.3285	0.1634	0.1651	0.0409	0.6504	0.3221	0.3283	0.0988
Bag3	0.1699	0.0800	0.0899	0.0309	0.3295	0.1623	0.1671	0.0423	0.6735	0.3599	0.3136	0.0975
Brag1	0.1519	0.0457	0.1062	0.0199	0.2928	0.0909	0.2019	0.0219	0.6113	0.1720	0.4393	0.0785
Brag2	0.1719	0.0687	0.1032	0.0256	0.3193	0.1315	0.1879	0.0315	0.6947	0.2564	0.4384	0.0869
Brag3	0.1655	0.0746	0.0909	0.0332	0.3168	0.1398	0.1770	0.0463	0.6935	0.2894	0.4041	0.0986
Boost	0.1758	0.0124	0.1635	0.0008	0.3506	0.0321	0.3186	-0.0086	0.7747	0.0765	0.6982	0.0230
	$\sigma = 1.0$											
Standard	0.3245	0.0733	0.2511	0.0313	0.6289	0.1212	0.5078	0.0275	1.1521	0.2199	0.9322	0.1204
Bag1	0.3138	0.1025	0.2113	0.0348	0.5453	0.1779	0.3674	0.0393	0.9321	0.2769	0.6552	0.1282
Bag2	0.3312	0.1470	0.1842	0.0446	0.5683	0.2473	0.3210	0.0633	0.9496	0.4120	0.5376	0.1500
Bag3	0.3175	0.1295	0.1880	0.0494	0.5533	0.2247	0.3286	0.0671	1.0020	0.4584	0.5436	0.1396
Brag1	0.3147	0.0898	0.2249	0.0322	0.5576	0.1571	0.4004	0.0377	1.0032	0.2419	0.7613	0.1264
Brag2	0.3325	0.1277	0.2048	0.0418	0.5744	0.2131	0.3613	0.0578	1.0664	0.3478	0.7186	0.1408
Brag3	0.3229	0.1320	0.1910	0.0513	0.5696	0.2258	0.3438	0.0765	1.0931	0.4420	0.6510	0.1713
Boost	0.3767	0.0254	0.3514	0.0075	0.7508	0.0518	0.6990	-0.0238	1.3419	0.1025	1.2394	0.0307

Table A.4: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 1, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0265	0.0039	0.0225	0.0008	0.0566	0.0078	0.0488	0.0009	0.1301	0.0211	0.1090	-0.0086
Bag1	0.0258	0.0041	0.0216	0.0010	0.0554	0.0095	0.0459	0.0004	0.1248	0.0249	0.0999	-0.0091
Bag2	0.0278	0.0076	0.0202	-0.0001	0.0634	0.0196	0.0437	0.0018	0.1605	0.0641	0.0963	-0.0102
Bag3	0.0274	0.0075	0.0199	-0.0002	0.0624	0.0202	0.0422	-0.0004	0.1661	0.0711	0.0950	-0.0265
Brag1	0.0258	0.0041	0.0216	0.0008	0.0554	0.0091	0.0462	0.0002	0.1242	0.0214	0.1028	-0.0084
Brag2	0.0276	0.0069	0.0207	0.0003	0.0604	0.0151	0.0452	0.0004	0.1472	0.0392	0.1081	-0.0091
Brag3	0.0271	0.0070	0.0201	0.0005	0.0595	0.0162	0.0434	0.0012	0.1543	0.0480	0.1063	-0.0147
Boost	0.0271	0.0004	0.0267	-0.0006	0.0554	0.0012	0.0542	0.0027	0.1195	0.0074	0.1121	-0.0041
	$\sigma = 0.6$											
Standard	0.1298	0.0253	0.1045	0.0016	0.2409	0.0487	0.1922	0.0035	0.4642	0.0796	0.3846	-0.0090
Bag1	0.1243	0.0284	0.0959	0.0019	0.2288	0.0586	0.1701	0.0039	0.4189	0.0984	0.3205	-0.0156
Bag2	0.1338	0.0465	0.0873	-0.0011	0.2475	0.0940	0.1536	0.0095	0.4562	0.1727	0.2835	-0.0118
Bag3	0.1333	0.0448	0.0886	-0.0018	0.2443	0.0894	0.1549	0.0069	0.4509	0.1757	0.2751	-0.0395
Brag1	0.1246	0.0285	0.0961	0.0019	0.2300	0.0598	0.1702	0.0044	0.4325	0.0870	0.3455	-0.0154
Brag2	0.1334	0.0444	0.0890	0.0005	0.2478	0.0888	0.1590	0.0074	0.4558	0.1328	0.3230	-0.0166
Brag3	0.1330	0.0444	0.0886	0.0015	0.2455	0.0880	0.1575	0.0088	0.4592	0.1564	0.3029	-0.0199
Boost	0.1417	0.0038	0.1378	-0.0010	0.2647	0.0121	0.2526	0.0050	0.5040	0.0280	0.4760	-0.0079
	$\sigma = 1.0$											
Standard	0.2828	0.0574	0.2254	0.0015	0.4968	0.1103	0.3865	0.0095	0.9948	0.1768	0.8179	-0.0070
Bag1	0.2702	0.0736	0.1966	0.0022	0.4756	0.1479	0.3277	0.0145	0.9165	0.2857	0.6308	-0.0012
Bag2	0.2908	0.1119	0.1789	-0.0018	0.5136	0.2157	0.2980	0.0236	0.9795	0.4028	0.5767	0.0039
Bag3	0.2876	0.1033	0.1842	-0.0028	0.4964	0.1920	0.3044	0.0172	0.8905	0.3522	0.5382	-0.0527
Brag1	0.2719	0.0739	0.1980	0.0024	0.4737	0.1440	0.3296	0.0135	0.9196	0.2064	0.7132	-0.0057
Brag2	0.2915	0.1086	0.1829	0.0006	0.5084	0.2018	0.3067	0.0194	0.9392	0.2827	0.6565	-0.0024
Brag3	0.2901	0.1074	0.1827	0.0006	0.5068	0.2026	0.3042	0.0175	0.9447	0.3686	0.5761	-0.0228
Boost	0.3171	0.0130	0.3041	-0.0027	0.5653	0.0354	0.5299	0.0090	1.1206	0.0754	1.0451	-0.0112

Table A.5: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 2, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0272	0.0039	0.0233	0.0004	0.0611	0.0078	0.0533	-0.0004	0.1481	0.0271	0.1210	-0.0114
Bag1	0.0262	0.0042	0.0219	0.0009	0.0577	0.0121	0.0456	0.0005	0.1381	0.0385	0.0996	-0.0093
Bag2	0.0286	0.0078	0.0208	-0.0001	0.0693	0.0238	0.0456	0.0003	0.1855	0.0832	0.1022	-0.0204
Bag3	0.0281	0.0080	0.0201	-0.0010	0.0684	0.0255	0.0429	-0.0054	0.1894	0.0926	0.0968	-0.0404
Brag1	0.0262	0.0042	0.0220	0.0007	0.0570	0.0112	0.0458	0.0007	0.1376	0.0347	0.1029	-0.0104
Brag2	0.0283	0.0071	0.0212	0.0001	0.0645	0.0187	0.0458	0.0010	0.1699	0.0593	0.1105	-0.0137
Brag3	0.0274	0.0073	0.0201	0.0004	0.0634	0.0201	0.0433	-0.0003	0.1691	0.0656	0.1035	-0.0196
Boost	0.0272	0.0004	0.0268	-0.0007	0.0593	0.0014	0.0580	0.0016	0.1323	0.0097	0.1226	-0.0059
	$\sigma = 0.6$											
Standard	0.1293	0.0231	0.1062	0.0020	0.2549	0.0453	0.2096	0.0013	0.4833	0.0785	0.4048	-0.0109
Bag1	0.1238	0.0263	0.0975	0.0016	0.2283	0.0534	0.1750	0.0037	0.4176	0.0937	0.3239	-0.0142
Bag2	0.1324	0.0435	0.0890	-0.0015	0.2449	0.0861	0.1589	0.0088	0.4477	0.1624	0.2852	-0.0145
Bag3	0.1318	0.0421	0.0898	-0.0026	0.2422	0.0841	0.1581	0.0026	0.4475	0.1746	0.2729	-0.0518
Brag1	0.1250	0.0262	0.0987	0.0014	0.2316	0.0535	0.1781	0.0034	0.4357	0.0805	0.3552	-0.0146
Brag2	0.1331	0.0415	0.0916	-0.0005	0.2473	0.0811	0.1662	0.0062	0.4503	0.1261	0.3242	-0.0180
Brag3	0.1316	0.0420	0.0896	0.0008	0.2437	0.0828	0.1609	0.0077	0.4565	0.1518	0.3047	-0.0233
Boost	0.1420	0.0037	0.1383	-0.0005	0.2812	0.0122	0.2690	0.0041	0.5304	0.0309	0.4995	-0.0041
	$\sigma = 1.0$											
Standard	0.2852	0.0526	0.2326	0.0000	0.5128	0.1021	0.4107	0.0126	1.1061	0.1613	0.9448	-0.0166
Bag1	0.2691	0.0692	0.1999	0.0017	0.4709	0.1295	0.3415	0.0126	0.8618	0.1637	0.6981	-0.0066
Bag2	0.2888	0.1065	0.1822	-0.0021	0.5013	0.1916	0.3097	0.0217	0.8555	0.2523	0.6031	-0.0064
Bag3	0.2849	0.0974	0.1875	-0.0035	0.4988	0.1787	0.3201	0.0082	0.8924	0.3353	0.5572	-0.0981
Brag1	0.2720	0.0677	0.2043	0.0020	0.4759	0.1257	0.3502	0.0121	1.0400	0.1325	0.9075	-0.0182
Brag2	0.2895	0.1010	0.1885	-0.0004	0.5034	0.1791	0.3243	0.0181	0.9911	0.1971	0.7940	-0.0222
Brag3	0.2884	0.1025	0.1860	-0.0003	0.5083	0.1914	0.3168	0.0164	0.9924	0.3506	0.6418	-0.0600
Boost	0.3221	0.0126	0.3095	-0.0031	0.5868	0.0349	0.5518	0.0121	1.2585	0.0748	1.1837	-0.0148

Table A.6: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 2, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0350	0.0056	0.0295	0.0090	0.0731	0.0094	0.0637	0.0061	0.1684	0.0240	0.1444	0.0135
Bag1	0.0341	0.0052	0.0289	0.0090	0.0709	0.0077	0.0632	0.0043	0.1569	0.0225	0.1344	0.0150
Bag2	0.0346	0.0098	0.0248	0.0124	0.0723	0.0208	0.0515	0.0085	0.1611	0.0606	0.1005	0.0260
Bag3	0.0348	0.0103	0.0245	0.0128	0.0728	0.0233	0.0495	0.0084	0.1619	0.0688	0.0931	0.0207
Brag1	0.0342	0.0049	0.0293	0.0088	0.0718	0.0073	0.0645	0.0038	0.1580	0.0210	0.1369	0.0141
Brag2	0.0369	0.0083	0.0286	0.0117	0.0792	0.0151	0.0641	0.0082	0.1739	0.0435	0.1304	0.0221
Brag3	0.0364	0.0091	0.0273	0.0133	0.0782	0.0175	0.0607	0.0099	0.1732	0.0536	0.1196	0.0248
Boost	0.0364	0.0008	0.0356	0.0022	0.0735	0.0018	0.0717	-0.0010	0.1623	0.0126	0.1497	0.0075
	$\sigma = 0.6$											
Standard	0.1623	0.0247	0.1377	0.0183	0.3073	0.0324	0.2749	0.0061	0.5974	0.0617	0.5356	0.0581
Bag1	0.1563	0.0255	0.1308	0.0188	0.2942	0.0320	0.2622	0.0044	0.5522	0.0541	0.4980	0.0501
Bag2	0.1497	0.0409	0.1088	0.0248	0.2654	0.0598	0.2056	0.0126	0.4716	0.1190	0.3526	0.0544
Bag3	0.1506	0.0409	0.1098	0.0243	0.2649	0.0618	0.2032	0.0126	0.4537	0.1344	0.3193	0.0438
Brag1	0.1564	0.0262	0.1302	0.0191	0.2974	0.0324	0.2651	0.0053	0.5791	0.0463	0.5328	0.0493
Brag2	0.1566	0.0399	0.1167	0.0238	0.2849	0.0529	0.2320	0.0083	0.5345	0.0832	0.4513	0.0608
Brag3	0.1546	0.0405	0.1141	0.0254	0.2767	0.0583	0.2184	0.0140	0.4864	0.1192	0.3671	0.0662
Boost	0.1985	0.0049	0.1936	0.0030	0.3838	0.0070	0.3767	-0.0084	0.6735	0.0210	0.6525	0.0335
	$\sigma = 1.0$											
Standard	0.3439	0.0599	0.2840	0.0304	0.6768	0.0740	0.6028	0.0071	1.2540	0.1299	1.1241	0.1087
Bag1	0.3250	0.0671	0.2579	0.0326	0.5870	0.0900	0.4970	0.0133	1.1141	0.1553	0.9588	0.1014
Bag2	0.3198	0.1009	0.2188	0.0425	0.5570	0.1473	0.4096	0.0305	1.0055	0.2787	0.7268	0.1142
Bag3	0.3221	0.0971	0.2251	0.0424	0.5523	0.1393	0.4130	0.0321	0.8971	0.2627	0.6344	0.0854
Brag1	0.3272	0.0665	0.2607	0.0324	0.6003	0.0845	0.5158	0.0105	1.2235	0.1013	1.1221	0.0876
Brag2	0.3304	0.0966	0.2337	0.0399	0.5799	0.1276	0.4523	0.0192	1.1087	0.1674	0.9414	0.1052
Brag3	0.3264	0.1001	0.2264	0.0457	0.5641	0.1446	0.4195	0.0362	0.9481	0.2759	0.6722	0.1280
Boost	0.4231	0.0154	0.4077	0.0054	0.8665	0.0227	0.8438	-0.0246	1.4853	0.0454	1.4398	0.0547

Table A.7: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 2, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0399	0.0074	0.0324	0.0128	0.0870	0.0136	0.0734	0.0067	0.1819	0.0315	0.1504	0.0144
Bag1	0.0364	0.0065	0.0299	0.0110	0.0794	0.0115	0.0679	0.0057	0.1680	0.0343	0.1338	0.0158
Bag2	0.0406	0.0135	0.0272	0.0154	0.0845	0.0300	0.0545	0.0101	0.1828	0.0787	0.1041	0.0248
Bag3	0.0406	0.0150	0.0256	0.0155	0.0828	0.0335	0.0494	0.0087	0.1785	0.0875	0.0910	0.0176
Brag1	0.0368	0.0062	0.0306	0.0107	0.0807	0.0109	0.0698	0.0060	0.1670	0.0314	0.1356	0.0146
Brag2	0.0434	0.0113	0.0321	0.0162	0.0926	0.0234	0.0692	0.0098	0.1943	0.0619	0.1325	0.0208
Brag3	0.0421	0.0124	0.0298	0.0173	0.0910	0.0266	0.0644	0.0110	0.1879	0.0719	0.1160	0.0236
Boost	0.0382	0.0013	0.0369	0.0044	0.0853	0.0045	0.0809	-0.0011	0.1720	0.0176	0.1543	0.0051
	$\sigma = 0.6$											
Standard	0.1689	0.0247	0.1442	0.0178	0.3336	0.0339	0.2997	0.0096	0.5830	0.0613	0.5216	0.0539
Bag1	0.1574	0.0242	0.1332	0.0186	0.2987	0.0318	0.2669	0.0041	0.5336	0.0589	0.4747	0.0468
Bag2	0.1507	0.0392	0.1115	0.0246	0.2676	0.0588	0.2088	0.0130	0.4500	0.1141	0.3358	0.0537
Bag3	0.1512	0.0407	0.1106	0.0245	0.2604	0.0631	0.1972	0.0116	0.4294	0.1337	0.2957	0.0344
Brag1	0.1589	0.0246	0.1343	0.0188	0.3043	0.0303	0.2740	0.0046	0.5497	0.0503	0.4994	0.0424
Brag2	0.1588	0.0384	0.1204	0.0241	0.2884	0.0518	0.2366	0.0096	0.5087	0.0901	0.4186	0.0514
Brag3	0.1559	0.0407	0.1152	0.0260	0.2789	0.0620	0.2169	0.0144	0.4668	0.1225	0.3443	0.0522
Boost	0.2046	0.0056	0.1991	0.0030	0.4055	0.0089	0.3966	-0.0025	0.6581	0.0237	0.6345	0.0345
	$\sigma = 1.0$											
Standard	0.3619	0.0553	0.3066	0.0278	0.7463	0.0674	0.6790	0.0016	1.3350	0.1048	1.2302	0.1008
Bag1	0.3305	0.0600	0.2705	0.0289	0.6180	0.0734	0.5445	0.0065	1.1184	0.1049	1.0135	0.0849
Bag2	0.3197	0.0912	0.2285	0.0379	0.5575	0.1197	0.4379	0.0203	0.9168	0.1859	0.7309	0.0893
Bag3	0.3212	0.0895	0.2317	0.0377	0.5490	0.1255	0.4234	0.0173	0.8545	0.2403	0.6141	0.0472
Brag1	0.3319	0.0601	0.2718	0.0291	0.6475	0.0693	0.5782	0.0036	1.2525	0.0805	1.1720	0.0792
Brag2	0.3290	0.0889	0.2401	0.0359	0.5980	0.1076	0.4904	0.0119	1.0773	0.1315	0.9458	0.0909
Brag3	0.3234	0.0938	0.2296	0.0406	0.5655	0.1341	0.4314	0.0294	0.9164	0.2453	0.6710	0.0931
Boost	0.4450	0.0158	0.4293	0.0032	0.9484	0.0227	0.9257	-0.0227	1.5802	0.0421	1.5381	0.0639

Table A.8: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 2, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0049	0.0009	0.0040	-0.0006	0.0094	0.0014	0.0080	0.0012	0.0211	0.0037	0.0174	-0.0058
Bag1	0.0048	0.0009	0.0039	-0.0006	0.0094	0.0017	0.0077	0.0004	0.0215	0.0054	0.0161	-0.0081
Bag2	0.0052	0.0015	0.0036	-0.0007	0.0100	0.0027	0.0073	0.0007	0.0213	0.0057	0.0155	-0.0088
Bag3	0.0051	0.0016	0.0035	-0.0011	0.0100	0.0032	0.0069	-0.0011	0.0227	0.0100	0.0128	-0.0277
Brag1	0.0048	0.0010	0.0039	-0.0006	0.0093	0.0017	0.0077	0.0006	0.0205	0.0036	0.0169	-0.0066
Brag2	0.0052	0.0015	0.0036	-0.0007	0.0100	0.0027	0.0073	0.0007	0.0213	0.0057	0.0155	-0.0088
Brag3	0.0051	0.0015	0.0036	-0.0005	0.0098	0.0027	0.0071	0.0006	0.0220	0.0072	0.0148	-0.0194
Boost	0.0052	0.0001	0.0051	-0.0003	0.0101	0.0002	0.0099	0.0016	0.0213	0.0012	0.0201	-0.0036
	$\sigma = 0.3$											
Standard	0.0261	0.0056	0.0205	-0.0020	0.0481	0.0096	0.0385	0.0036	0.0969	0.0231	0.0738	-0.0041
Bag1	0.0266	0.0081	0.0185	-0.0026	0.0628	0.0280	0.0348	0.0080	0.1143	0.0757	0.0386	0.0190
Bag2	0.0270	0.0102	0.0168	-0.0021	0.0501	0.0186	0.0315	0.0065	0.0959	0.0389	0.0570	0.0034
Bag3	0.0267	0.0099	0.0167	-0.0030	0.0478	0.0185	0.0293	0.0028	0.0813	0.0425	0.0388	-0.0365
Brag1	0.0252	0.0070	0.0182	-0.0020	0.0475	0.0135	0.0340	0.0044	0.0960	0.0321	0.0639	0.0010
Brag2	0.0270	0.0102	0.0168	-0.0021	0.0501	0.0186	0.0315	0.0065	0.0959	0.0389	0.0570	0.0034
Brag3	0.0267	0.0100	0.0167	-0.0007	0.0490	0.0186	0.0304	0.0082	0.0915	0.0500	0.0415	-0.0359
Boost	0.0296	0.0013	0.0283	-0.0015	0.0559	0.0031	0.0528	0.0020	0.1092	0.0121	0.0970	-0.0097
	$\sigma = 0.5$											
Standard	0.0581	0.0131	0.0450	-0.0026	0.1008	0.0215	0.0793	0.0114	0.1735	0.0549	0.1185	0.0053
Bag1	0.0848	0.0492	0.0356	0.0092	0.1224	0.0832	0.0392	0.0266	0.1563	0.1072	0.0491	0.0306
Bag2	0.0648	0.0285	0.0363	-0.0003	0.1059	0.0490	0.0569	0.0176	0.1611	0.0754	0.0858	0.0126
Bag3	0.0595	0.0267	0.0329	-0.0009	0.0891	0.0407	0.0484	0.0107	0.1296	0.0646	0.0650	-0.0310
Brag1	0.0620	0.0222	0.0398	-0.0014	0.1043	0.0415	0.0628	0.0125	0.1657	0.0694	0.0963	0.0140
Brag2	0.0648	0.0285	0.0363	-0.0003	0.1059	0.0490	0.0569	0.0176	0.1611	0.0754	0.0858	0.0126
Brag3	0.0627	0.0284	0.0344	0.0037	0.0968	0.0488	0.0480	0.0184	0.1367	0.0822	0.0546	-0.0331
Boost	0.0682	0.0047	0.0635	-0.0039	0.1224	0.0100	0.1123	0.0085	0.2021	0.0400	0.1621	-0.0031

Table A.9: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 3, $X \sim U(-2, 2)$, $N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0050	0.0009	0.0041	-0.0006	0.0098	0.0013	0.0085	0.0009	0.0240	0.0041	0.0199	-0.0113
Bag1	0.0048	0.0009	0.0039	-0.0005	0.0094	0.0016	0.0078	0.0001	0.0203	0.0050	0.0153	-0.0066
Bag2	0.0051	0.0014	0.0037	-0.0006	0.0100	0.0025	0.0075	-0.0004	0.0221	0.0065	0.0156	-0.0154
Bag3	0.0051	0.0015	0.0036	-0.0015	0.0101	0.0032	0.0069	-0.0064	0.0237	0.0113	0.0124	-0.0415
Brag1	0.0048	0.0009	0.0039	-0.0005	0.0094	0.0015	0.0079	0.0001	0.0208	0.0041	0.0166	-0.0070
Brag2	0.0051	0.0014	0.0037	-0.0006	0.0100	0.0025	0.0075	-0.0004	0.0221	0.0065	0.0156	-0.0154
Brag3	0.0050	0.0014	0.0036	-0.0008	0.0096	0.0026	0.0070	-0.0016	0.0224	0.0078	0.0145	-0.0279
Boost	0.0052	0.0001	0.0051	-0.0003	0.0105	0.0002	0.0103	0.0012	0.0244	0.0016	0.0229	-0.0089
	$\sigma = 0.3$											
Standard	0.0260	0.0052	0.0208	-0.0021	0.0492	0.0090	0.0402	0.0044	0.1083	0.0203	0.0880	-0.0214
Bag1	0.0252	0.0068	0.0184	-0.0025	0.0481	0.0132	0.0349	0.0038	0.0851	0.0233	0.0618	-0.0019
Bag2	0.0267	0.0093	0.0174	-0.0023	0.0502	0.0152	0.0350	0.0039	0.1053	0.0199	0.0853	-0.0287
Bag3	0.0264	0.0092	0.0172	-0.0037	0.0477	0.0163	0.0314	-0.0123	0.0825	0.0381	0.0444	-0.0859
Brag1	0.0250	0.0063	0.0188	-0.0021	0.0485	0.0106	0.0379	0.0030	0.1168	0.0145	0.1023	-0.0221
Brag2	0.0267	0.0093	0.0174	-0.0023	0.0502	0.0152	0.0350	0.0039	0.1053	0.0199	0.0853	-0.0287
Brag3	0.0264	0.0094	0.0170	-0.0009	0.0481	0.0162	0.0318	-0.0026	0.0942	0.0415	0.0527	-0.0995
Boost	0.0295	0.0013	0.0282	-0.0017	0.0571	0.0032	0.0539	0.0033	0.1233	0.0113	0.1120	-0.0262
	$\sigma = 0.5$											
Standard	0.0595	0.0121	0.0475	-0.0029	0.1012	0.0191	0.0821	0.0116	0.2271	0.0460	0.1811	-0.0399
Bag1	0.0633	0.0269	0.0364	0.0006	0.0948	0.0342	0.0606	0.0148	0.1712	0.0298	0.1414	0.0039
Bag2	0.0622	0.0258	0.0365	0.0002	0.1050	0.0300	0.0750	0.0123	0.2519	0.0337	0.2182	-0.0728
Bag3	0.0583	0.0237	0.0346	-0.0025	0.0931	0.0344	0.0587	-0.0263	0.1438	0.0592	0.0846	-0.1270
Brag1	0.0589	0.0194	0.0395	-0.0011	0.1081	0.0230	0.0851	0.0085	0.3192	0.0237	0.2955	-0.0549
Brag2	0.0622	0.0258	0.0365	0.0002	0.1050	0.0300	0.0750	0.0123	0.2519	0.0337	0.2182	-0.0728
Brag3	0.0604	0.0254	0.0350	0.0022	0.0965	0.0404	0.0561	-0.0204	0.1607	0.0770	0.0838	-0.1874
Boost	0.0702	0.0046	0.0656	-0.0042	0.1231	0.0089	0.1142	0.0088	0.2610	0.0329	0.2281	-0.0476

Table A.10: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 3, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0066	0.0008	0.0058	0.0044	0.0130	0.0013	0.0118	0.0054	0.0271	0.0034	0.0238	0.0148
Bag1	0.0063	0.0007	0.0056	0.0041	0.0124	0.0014	0.0110	0.0053	0.0274	0.0062	0.0212	0.0182
Bag2	0.0063	0.0013	0.0050	0.0057	0.0120	0.0023	0.0097	0.0083	0.0256	0.0054	0.0202	0.0161
Bag3	0.0059	0.0014	0.0045	0.0065	0.0113	0.0031	0.0082	0.0082	0.0245	0.0100	0.0145	0.0012
Brag1	0.0063	0.0008	0.0056	0.0042	0.0125	0.0014	0.0111	0.0057	0.0248	0.0033	0.0215	0.0176
Brag2	0.0063	0.0013	0.0050	0.0057	0.0120	0.0023	0.0097	0.0083	0.0256	0.0054	0.0202	0.0161
Brag3	0.0061	0.0013	0.0048	0.0058	0.0113	0.0024	0.0090	0.0073	0.0250	0.0063	0.0188	0.0030
Boost	0.0076	0.0001	0.0076	0.0014	0.0150	0.0002	0.0148	0.0027	0.0279	0.0012	0.0267	0.0132
	$\sigma = 0.3$											
Standard	0.0325	0.0059	0.0266	0.0127	0.0670	0.0094	0.0575	0.0119	0.1233	0.0198	0.1036	0.0320
Bag1	0.0338	0.0096	0.0242	0.0153	0.0758	0.0337	0.0421	0.0256	0.1249	0.0681	0.0568	0.0422
Bag2	0.0334	0.0111	0.0223	0.0178	0.0613	0.0188	0.0425	0.0192	0.1140	0.0314	0.0826	0.0437
Bag3	0.0319	0.0107	0.0213	0.0185	0.0541	0.0190	0.0351	0.0158	0.0862	0.0372	0.0491	0.0006
Brag1	0.0320	0.0074	0.0245	0.0139	0.0616	0.0134	0.0481	0.0131	0.1201	0.0247	0.0954	0.0362
Brag2	0.0334	0.0111	0.0223	0.0178	0.0613	0.0188	0.0425	0.0192	0.1140	0.0314	0.0826	0.0437
Brag3	0.0324	0.0110	0.0215	0.0212	0.0567	0.0196	0.0371	0.0197	0.0970	0.0419	0.0551	-0.0079
Boost	0.0382	0.0012	0.0369	0.0032	0.0827	0.0033	0.0794	0.0010	0.1457	0.0110	0.1346	0.0331
	$\sigma = 0.5$											
Standard	0.0718	0.0167	0.0551	0.0228	0.1354	0.0262	0.1092	0.0144	0.2231	0.0454	0.1777	0.0307
Bag1	0.0963	0.0572	0.0391	0.0376	0.1331	0.0916	0.0415	0.0350	0.1728	0.0972	0.0756	0.0400
Bag2	0.0769	0.0331	0.0439	0.0321	0.1195	0.0546	0.0649	0.0287	0.1977	0.0606	0.1370	0.0432
Bag3	0.0674	0.0292	0.0382	0.0251	0.0966	0.0425	0.0541	0.0171	0.1382	0.0549	0.0833	-0.0018
Brag1	0.0737	0.0253	0.0484	0.0273	0.1212	0.0461	0.0751	0.0248	0.2141	0.0529	0.1612	0.0376
Brag2	0.0769	0.0331	0.0439	0.0321	0.1195	0.0546	0.0649	0.0287	0.1977	0.0606	0.1370	0.0432
Brag3	0.0714	0.0322	0.0392	0.0316	0.1046	0.0539	0.0507	0.0151	0.1442	0.0704	0.0738	-0.0208
Boost	0.0841	0.0061	0.0780	0.0099	0.1718	0.0137	0.1581	0.0033	0.2732	0.0346	0.2386	0.0387

Table A.11: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 3, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0068	0.0007	0.0061	0.0039	0.0131	0.0012	0.0119	0.0035	0.0259	0.0031	0.0228	0.0000
Bag1	0.0064	0.0006	0.0058	0.0035	0.0122	0.0015	0.0107	0.0042	0.0244	0.0053	0.0191	0.0109
Bag2	0.0062	0.0011	0.0051	0.0043	0.0116	0.0022	0.0094	0.0016	0.0248	0.0061	0.0187	-0.0028
Bag3	0.0057	0.0012	0.0045	0.0030	0.0109	0.0031	0.0078	-0.0049	0.0236	0.0104	0.0132	-0.0268
Brag1	0.0065	0.0007	0.0058	0.0036	0.0124	0.0014	0.0110	0.0038	0.0238	0.0040	0.0197	0.0089
Brag2	0.0062	0.0011	0.0051	0.0043	0.0116	0.0022	0.0094	0.0016	0.0248	0.0061	0.0187	-0.0028
Brag3	0.0058	0.0011	0.0047	0.0038	0.0108	0.0023	0.0085	-0.0004	0.0235	0.0066	0.0169	-0.0179
Boost	0.0080	0.0001	0.0079	0.0013	0.0148	0.0002	0.0146	0.0010	0.0266	0.0010	0.0255	0.0004
	$\sigma = 0.3$											
Standard	0.0346	0.0048	0.0298	0.0101	0.0705	0.0071	0.0634	0.0081	0.1240	0.0167	0.1073	0.0212
Bag1	0.0320	0.0067	0.0253	0.0117	0.0609	0.0142	0.0468	0.0122	0.1056	0.0206	0.0851	0.0356
Bag2	0.0329	0.0095	0.0234	0.0149	0.0600	0.0128	0.0472	0.0107	0.1143	0.0154	0.0989	0.0091
Bag3	0.0313	0.0089	0.0224	0.0140	0.0525	0.0155	0.0370	-0.0039	0.0831	0.0316	0.0515	-0.0483
Brag1	0.0323	0.0062	0.0261	0.0115	0.0640	0.0088	0.0551	0.0078	0.1337	0.0111	0.1226	0.0211
Brag2	0.0329	0.0095	0.0234	0.0149	0.0600	0.0128	0.0472	0.0107	0.1143	0.0154	0.0989	0.0091
Brag3	0.0318	0.0094	0.0224	0.0171	0.0541	0.0155	0.0386	0.0030	0.0957	0.0326	0.0631	-0.0662
Boost	0.0418	0.0011	0.0407	0.0023	0.0886	0.0026	0.0860	-0.0009	0.1456	0.0093	0.1363	0.0253
	$\sigma = 0.5$											
Standard	0.0759	0.0135	0.0625	0.0208	0.1509	0.0200	0.1309	0.0085	0.2586	0.0353	0.2233	0.0035
Bag1	0.0742	0.0299	0.0443	0.0326	0.1157	0.0375	0.0783	0.0353	0.2268	0.0275	0.1992	0.0479
Bag2	0.0744	0.0267	0.0478	0.0299	0.1254	0.0300	0.0955	0.0205	0.2715	0.0238	0.2477	-0.0210
Bag3	0.0663	0.0242	0.0422	0.0229	0.0997	0.0335	0.0662	-0.0173	0.1447	0.0469	0.0978	-0.0809
Brag1	0.0719	0.0196	0.0523	0.0250	0.1371	0.0225	0.1146	0.0180	0.3506	0.0155	0.3350	-0.0002
Brag2	0.0744	0.0267	0.0478	0.0299	0.1254	0.0300	0.0955	0.0205	0.2715	0.0238	0.2477	-0.0210
Brag3	0.0691	0.0272	0.0419	0.0319	0.1033	0.0413	0.0620	-0.0196	0.1603	0.0614	0.0989	-0.1444
Boost	0.0919	0.0049	0.0869	0.0101	0.1936	0.0099	0.1837	-0.0004	0.3105	0.0256	0.2849	0.0191

Table A.12: Nadaraya-Watson estimator with cross-validation bandwidth selection, Model 3, $X \sim N(0, 1)$, Triangular kernel

				Bagging		Bragging	
X	Kernel	σ	n	Smallest $\int \text{Var}\, dx$	Smallest MISE	Smallest $\int \text{Var}\, dx$	Smallest MISE
$U(-2,2)$	Gaussian	0.2	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.2	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.2	50	Bag1	Bag1	Brag1	Brag1
$U(-2,2)$	Gaussian	0.6	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.6	100	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.6	50	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	100	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	50	Bag2	Bag3	Brag3	Brag3
$U(-2,2)$	Triangular	0.2	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.2	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.2	50	Bag1	Bag1	Brag1	Brag1
$U(-2,2)$	Triangular	0.6	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.6	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.6	50	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	100	Bag2	Bag3	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	50	Bag2	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.2	200	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.2	100	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.2	50	Bag2	Bag1	Brag1	Brag1
$N(0,1)$	Gaussian	0.6	200	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.6	100	Bag2	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.6	50	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	1.0	200	Bag2	Bag3	Brag3	Brag1
$N(0,1)$	Gaussian	1.0	100	Bag2	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	1.0	50	Bag2	Bag2	Brag3	Brag1
$N(0,1)$	Triangular	0.2	200	Bag1	Bag1	Brag1	Brag1
$N(0,1)$	Triangular	0.2	100	Bag1	Bag1	Brag1	Brag1
$N(0,1)$	Triangular	0.2	50	Bag3	Bag1	Brag1	Brag1
$N(0,1)$	Triangular	0.6	200	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	0.6	100	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	0.6	50	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	1.0	200	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	1.0	100	Bag2	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	1.0	50	Bag2	Bag1	Brag3	Brag1

Table A.13: Bagging and bragging procedures producing the smallest variance and MISE values for Model 1

				Bagging		Bragging	
X	Kernel	σ	n	Best $\int \mathrm{Var}\, dx$	Best MISE	Best $\int \mathrm{Var}\, dx$	Best MISE
$U(-2,2)$	Gaussian	0.2	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.2	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.2	50	Bag3	Bag1	Brag1	Brag1
$U(-2,2)$	Gaussian	0.6	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.6	100	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.6	50	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	100	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	1.0	50	Bag3	Bag3	Brag3	Brag1
$U(-2,2)$	Triangular	0.2	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.2	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.2	50	Bag3	Bag1	Brag1	Brag1
$U(-2,2)$	Triangular	0.6	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.6	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.6	50	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	200	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	100	Bag2	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	1.0	50	Bag3	Bag2	Brag3	Brag2
$N(0,1)$	Gaussian	0.2	200	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.2	100	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.2	50	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Gaussian	0.6	200	Bag2	Bag2	Brag3	Brag3
$N(0,1)$	Gaussian	0.6	100	Bag2	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.6	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	1.0	200	Bag2	Bag2	Brag3	Brag3
$N(0,1)$	Gaussian	1.0	100	Bag2	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	1.0	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.2	200	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	0.2	100	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	0.2	50	Bag3	Bag1	Brag3	Brag1
$N(0,1)$	Triangular	0.6	200	Bag3	Bag2	Brag3	Brag3
$N(0,1)$	Triangular	0.6	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.6	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	1.0	200	Bag2	Bag2	Brag3	Brag3
$N(0,1)$	Triangular	1.0	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	1.0	50	Bag3	Bag3	Brag3	Brag3

Table A.14: Bagging and bragging procedures producing the smallest variance and MISE values for Model 2

				Bagging		Bragging	
X	Kernel	σ	n	Best $\int \mathrm{Var}\, dx$	Best MISE	Best $\int \mathrm{Var}\, dx$	Best MISE
$U(-2,2)$	Gaussian	0.1	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.1	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Gaussian	0.1	50	Bag3	Bag2*	Brag3	Brag1
$U(-2,2)$	Gaussian	0.3	200	Bag3	Bag1*	Brag3	Brag1
$U(-2,2)$	Gaussian	0.3	100	Bag3	Bag3	Brag3	Brag1
$U(-2,2)$	Gaussian	0.3	50	Bag1	Bag3	Brag3	Brag3
$U(-2,2)$	Gaussian	0.5	200	Bag3	Bag3*	Brag3	Brag1*
$U(-2,2)$	Gaussian	0.5	100	Bag1	Bag3	Brag3	Brag3
$U(-2,2)$	Gaussian	0.5	50	Bag1	Bag3	Brag3	Brag3
$U(-2,2)$	Triangular	0.1	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.1	100	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.1	50	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.3	200	Bag3	Bag1	Brag3	Brag1
$U(-2,2)$	Triangular	0.3	100	Bag3	Bag3	Brag3	Brag3
$U(-2,2)$	Triangular	0.3	50	Bag3	Bag3	Brag3	Brag3
$U(-2,2)$	Triangular	0.5	200	Bag3	Bag3	Brag3	Brag1
$U(-2,2)$	Triangular	0.5	100	Bag3	Bag3	Brag3	Brag3
$U(-2,2)$	Triangular	0.5	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.1	200	Bag3	Bag3	Brag3	Brag1
$N(0,1)$	Gaussian	0.1	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.1	50	Bag3	Bag3	Brag3	Brag1
$N(0,1)$	Gaussian	0.3	200	Bag3	Bag3	Brag3	Brag1
$N(0,1)$	Gaussian	0.3	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.3	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.5	200	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.5	100	Bag1	Bag3	Brag3	Brag3
$N(0,1)$	Gaussian	0.5	50	Bag1	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.1	200	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.1	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.1	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.1	200	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.3	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.3	50	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.5	200	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.5	100	Bag3	Bag3	Brag3	Brag3
$N(0,1)$	Triangular	0.5	50	Bag3	Bag3	Brag3	Brag3

Table A.15: Bagging and bragging procedures producing the smallest variance and MISE values for Model 3

	Smallest $\int \text{Var}\, dx$		Smallest MISE	
	Frequency	Percentage	Frequency	Percentage
Bag1	4	11.11%	29	80.56%
Bag2	23	63.89%	1	2.78%
Bag3	9	25.00%	6	16.67%
Brag1	6	16.67%	32	88.89%
Brag2	0	0.00%	0	0.00%
Brag3	30	83.33%	4	11.11%

Table A.16: Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 1

	Smallest $\int \text{Var}\, dx$		Smallest MISE	
	Frequency	Percentage	Frequency	Percentage
Bag1	0	0.00%	22	61.11%
Bag2	12	33.33%	5	13.89%
Bag3	24	67.67%	9	25.00%
Brag1	2	5.56%	23	63.89%
Brag2	0	0.00%	1	2.78%
Brag3	34	94.44%	12	33.33%

Table A.17: Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 2

	Smallest $\int \text{Var}\, dx$		Smallest MISE	
	Frequency	Percentage	Frequency	Percentage
Bag1	5	13.89%	7	19.44%
Bag2	0	0.00%	1	2.78%
Bag3	31	86.11%	28	77.78%
Brag1	0	0.00%	14	38.89%
Brag2	0	0.00%	0	0.00%
Brag3	36	100.00%	22	61.11%

Table A.18: Percentage of bagging and bragging procedures producing the smallest variance and MISE values for Model 3

Appendix B

Results of Simulation Study B

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.02658	0.00844	0.01815	0.00127	0.05452	0.01482	0.03970	0.00163	0.26999	0.19486	0.07513	0.00252
Bag1	0.02611	0.00779	0.01832	0.00124	0.05303	0.01192	0.04111	0.00150	0.24070	0.16605	0.07465	0.00096
Bag2	0.02694	0.00847	0.01847	0.00279	0.05552	0.01434	0.04118	0.00070	0.26272	0.18554	0.07718	0.00313
Bag3	0.02702	0.00855	0.01846	0.00223	0.05634	0.01496	0.04138	-0.00042	0.26274	0.18627	0.07648	0.00067
Brag1	0.02612	0.00779	0.01833	0.00120	0.05312	0.01227	0.04085	0.00154	0.24145	0.16717	0.07428	0.00222
Brag2	0.02684	0.00805	0.01879	0.00309	0.05516	0.01290	0.04226	-0.00029	0.25100	0.17327	0.07772	-0.00084
Brag3	0.02687	0.00808	0.01879	0.00275	0.05496	0.01276	0.04220	-0.00119	0.25160	0.17389	0.07771	-0.00446
Boost	0.02062	0.00075	0.01988	-0.00067	0.04340	0.00222	0.04118	0.00222	0.15935	0.09562	0.06373	-0.00236
	$\sigma = 0.6$											
Standard	0.11472	0.02489	0.08983	0.00157	0.21327	0.03847	0.17479	0.00763	0.45988	0.25547	0.20441	0.00727
Bag1	0.11437	0.02137	0.09300	0.00155	0.22192	0.02695	0.19498	0.00771	0.42877	0.21699	0.21178	0.00350
Bag2	0.11599	0.02271	0.09328	0.00526	0.21769	0.02981	0.18789	0.00515	0.44690	0.23673	0.21017	0.00618
Bag3	0.11616	0.02269	0.09348	0.00445	0.21869	0.03042	0.18826	0.00426	0.44149	0.23242	0.20907	0.00529
Brag1	0.11441	0.02116	0.09326	0.00151	0.22096	0.02753	0.19343	0.00772	0.42350	0.21107	0.21243	0.00416
Brag2	0.11718	0.02199	0.09519	0.00483	0.22343	0.02890	0.19453	0.00407	0.43735	0.21983	0.21751	0.00569
Brag3	0.11724	0.02234	0.09490	0.00467	0.22375	0.02922	0.19453	0.00290	0.43877	0.22335	0.21542	0.00493
Boost	0.12453	0.00426	0.12027	-0.00125	0.24270	0.00995	0.23275	0.00842	0.38836	0.14496	0.24340	-0.00199
	$\sigma = 1.0$											
Standard	0.24763	0.03925	0.20839	0.00134	0.46093	0.05442	0.40651	0.01399	0.72461	0.29779	0.42682	0.01349
Bag1	0.25216	0.03114	0.22102	0.00131	0.50910	0.03574	0.47336	0.01385	0.70173	0.24894	0.45279	0.00850
Bag2	0.25496	0.03292	0.22205	0.00706	0.49317	0.03838	0.45479	0.00929	0.71619	0.26890	0.44729	0.01130
Bag3	0.25605	0.03257	0.22348	0.00633	0.49691	0.03889	0.45801	0.00874	0.71126	0.26066	0.45060	0.01233
Brag1	0.25291	0.03063	0.22228	0.00122	0.50719	0.03600	0.47119	0.01374	0.69794	0.23779	0.46014	0.00811
Brag2	0.25926	0.03196	0.22730	0.00670	0.50980	0.03775	0.47206	0.00785	0.71820	0.24829	0.46990	0.01138
Brag3	0.25879	0.03269	0.22610	0.00634	0.51033	0.03911	0.47122	0.00675	0.71684	0.25565	0.46119	0.01312
Boost	0.29756	0.00912	0.28844	-0.00187	0.58425	0.01777	0.56648	0.01517	0.73660	0.18372	0.55288	0.00307

Table B.1: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0286	0.0080	0.0206	-0.0003	0.0612	0.0140	0.0472	0.0012	0.2207	0.1360	0.0847	0.0043
Bag1	0.0282	0.0075	0.0207	0.0001	0.0616	0.0114	0.0503	0.0017	0.2059	0.1245	0.0814	0.0037
Bag2	0.0291	0.0080	0.0211	0.0014	0.0681	0.0147	0.0534	-0.0007	0.2353	0.1448	0.0905	0.0080
Bag3	0.0292	0.0081	0.0210	0.0008	0.0688	0.0161	0.0527	-0.0035	0.2392	0.1487	0.0905	0.0031
Brag1	0.0284	0.0078	0.0206	0.0002	0.0620	0.0124	0.0496	0.0019	0.2516	0.1684	0.0832	0.0027
Brag2	0.0289	0.0076	0.0213	0.0018	0.0638	0.0122	0.0516	-0.0004	0.2173	0.1310	0.0863	0.0020
Brag3	0.0289	0.0077	0.0212	0.0015	0.0655	0.0122	0.0533	-0.0011	0.2164	0.1307	0.0858	-0.0012
Boost	0.0213	0.0007	0.0206	-0.0011	0.0445	0.0020	0.0425	0.0020	0.1151	0.0475	0.0676	0.0024
	$\sigma = 0.6$											
Standard	0.1165	0.0241	0.0924	-0.0002	0.2187	0.0367	0.1820	0.0067	0.4255	0.1894	0.2361	0.0072
Bag1	0.1160	0.0209	0.0951	0.0002	0.2290	0.0256	0.2034	0.0081	0.4075	0.1701	0.2374	0.0034
Bag2	0.1178	0.0219	0.0958	0.0038	0.2277	0.0284	0.1993	0.0059	0.4270	0.1898	0.2372	0.0086
Bag3	0.1175	0.0220	0.0955	0.0028	0.2278	0.0303	0.1975	0.0023	0.4245	0.1892	0.2354	0.0065
Brag1	0.1151	0.0215	0.0936	0.0007	0.2254	0.0278	0.1977	0.0081	0.4346	0.2093	0.2253	0.0062
Brag2	0.1189	0.0213	0.0976	0.0033	0.2318	0.0274	0.2044	0.0039	0.4196	0.1751	0.2444	0.0046
Brag3	0.1188	0.0218	0.0970	0.0029	0.2322	0.0280	0.2042	0.0023	0.4160	0.1763	0.2397	0.0050
Boost	0.1273	0.0038	0.1235	-0.0018	0.2509	0.0089	0.2420	0.0081	0.3746	0.0795	0.2951	0.0007
	$\sigma = 1.0$											
Standard	0.2501	0.0384	0.2117	-0.0007	0.4713	0.0520	0.4193	0.0128	0.7286	0.2241	0.5045	0.0118
Bag1	0.2527	0.0306	0.2221	0.0001	0.5194	0.0340	0.4854	0.0137	0.7158	0.1988	0.5171	0.0062
Bag2	0.2561	0.0320	0.2241	0.0058	0.5075	0.0364	0.4711	0.0095	0.7279	0.2184	0.5095	0.0114
Bag3	0.2569	0.0319	0.2250	0.0046	0.5083	0.0384	0.4700	0.0070	0.7229	0.2137	0.5092	0.0126
Brag1	0.2518	0.0313	0.2205	0.0002	0.5099	0.0361	0.4739	0.0140	0.7191	0.2323	0.4868	0.0095
Brag2	0.2604	0.0312	0.2292	0.0056	0.5234	0.0359	0.4875	0.0073	0.7381	0.2020	0.5361	0.0090
Brag3	0.2601	0.0321	0.2279	0.0052	0.5228	0.0376	0.4851	0.0058	0.7266	0.2063	0.5203	0.0125
Boost	0.3055	0.0082	0.2974	-0.0024	0.6098	0.0158	0.5940	0.0148	0.7977	0.1060	0.6917	0.0007

Table B.2: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 1, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0347	0.0150	0.0197	0.0102	0.0848	0.0177	0.0670	0.0066	0.3099	0.2207	0.0892	0.0793
Bag1	0.0319	0.0112	0.0207	0.0085	0.0999	0.0117	0.0882	0.0051	0.2663	0.1733	0.0930	0.0669
Bag2	0.0323	0.0117	0.0205	0.0095	0.1148	0.0230	0.0918	0.0026	0.2885	0.1933	0.0952	0.0695
Bag3	0.0328	0.0119	0.0209	0.0090	0.1187	0.0296	0.0891	0.0025	0.2917	0.1959	0.0958	0.0653
Brag1	0.0320	0.0114	0.0206	0.0086	0.0998	0.0124	0.0874	0.0055	0.2697	0.1771	0.0926	0.0680
Brag2	0.0322	0.0115	0.0206	0.0092	0.1038	0.0133	0.0905	0.0059	0.2757	0.1808	0.0949	0.0677
Brag3	0.0321	0.0114	0.0207	0.0087	0.1114	0.0139	0.0974	0.0062	0.2703	0.1763	0.0940	0.0626
Boost	0.0248	0.0017	0.0231	0.0003	0.0780	0.0032	0.0748	-0.0055	0.1948	0.1078	0.0871	0.0124
	$\sigma = 0.6$											
Standard	0.1411	0.0365	0.1046	0.0179	0.2759	0.0395	0.2364	0.0055	0.5402	0.2926	0.2475	0.1147
Bag1	0.1416	0.0286	0.1130	0.0156	0.3118	0.0249	0.2869	-0.0006	0.5011	0.2330	0.2681	0.1003
Bag2	0.1374	0.0292	0.1082	0.0177	0.2832	0.0310	0.2522	-0.0037	0.5001	0.2518	0.2483	0.1016
Bag3	0.1375	0.0294	0.1081	0.0171	0.2868	0.0360	0.2508	-0.0027	0.4969	0.2495	0.2474	0.0974
Brag1	0.1413	0.0286	0.1127	0.0157	0.3108	0.0262	0.2846	-0.0001	0.5000	0.2308	0.2692	0.0999
Brag2	0.1413	0.0292	0.1121	0.0173	0.3069	0.0272	0.2797	-0.0008	0.5092	0.2375	0.2718	0.1031
Brag3	0.1411	0.0293	0.1118	0.0168	0.3116	0.0282	0.2835	0.0003	0.5050	0.2374	0.2676	0.1002
Boost	0.1545	0.0081	0.1465	0.0023	0.3416	0.0100	0.3315	-0.0135	0.4873	0.1564	0.3309	0.0289
	$\sigma = 1.0$											
Standard	0.2975	0.0624	0.2351	0.0251	0.5873	0.0622	0.5252	0.0028	0.8728	0.3647	0.5082	0.1514
Bag1	0.3061	0.0488	0.2574	0.0223	0.6874	0.0378	0.6496	-0.0060	0.8527	0.2892	0.5635	0.1340
Bag2	0.2981	0.0496	0.2485	0.0256	0.6085	0.0431	0.5654	-0.0105	0.8280	0.3078	0.5201	0.1343
Bag3	0.2978	0.0492	0.2486	0.0249	0.6113	0.0477	0.5637	-0.0096	0.8200	0.2996	0.5204	0.1320
Brag1	0.3059	0.0483	0.2576	0.0223	0.6840	0.0389	0.6451	-0.0059	0.8524	0.2827	0.5696	0.1325
Brag2	0.3057	0.0492	0.2564	0.0248	0.6725	0.0402	0.6323	-0.0067	0.8632	0.2913	0.5719	0.1355
Brag3	0.3047	0.0498	0.2549	0.0243	0.6709	0.0422	0.6288	-0.0038	0.8562	0.2943	0.5619	0.1345
Boost	0.3559	0.0191	0.3368	0.0043	0.7883	0.0208	0.7675	-0.0229	0.9412	0.2073	0.7339	0.0457

Table B.3: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 1, $X \sim N(0,1), N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0432	0.0147	0.0284	0.0067	0.1785	0.0236	0.1550	0.0069	0.3464	0.1619	0.1845	0.0572
Bag1	0.0459	0.0113	0.0347	0.0062	0.2535	0.0267	0.2268	0.0036	0.3361	0.1460	0.1901	0.0530
Bag2	0.0560	0.0143	0.0418	0.0036	0.3343	0.1021	0.2322	0.0022	0.3994	0.1987	0.2007	0.0531
Bag3	0.0547	0.0152	0.0395	0.0023	0.3165	0.1173	0.1992	0.0005	0.3958	0.2118	0.1840	0.0521
Brag1	0.0429	0.0125	0.0304	0.0071	0.2302	0.0253	0.2050	0.0054	0.3769	0.2181	0.1588	0.0712
Brag2	0.0463	0.0116	0.0347	0.0070	0.2660	0.0308	0.2352	0.0097	0.3448	0.1525	0.1923	0.0529
Brag3	0.0481	0.0116	0.0364	0.0064	0.3054	0.0396	0.2658	0.0077	0.3579	0.1541	0.2039	0.0511
Boost	0.0250	0.0016	0.0234	-0.0002	0.0861	0.0031	0.0830	-0.0055	0.1616	0.0548	0.1067	0.0100
	$\sigma = 0.6$											
Standard	0.1437	0.0360	0.1077	0.0151	0.3409	0.0403	0.3006	0.0071	0.5681	0.2127	0.3554	0.0866
Bag1	0.1468	0.0286	0.1182	0.0150	0.4074	0.0294	0.3781	0.0012	0.5579	0.1927	0.3651	0.0851
Bag2	0.1469	0.0295	0.1174	0.0156	0.4225	0.0694	0.3531	-0.0021	0.5806	0.2317	0.3489	0.0815
Bag3	0.1451	0.0301	0.1151	0.0142	0.4144	0.0859	0.3285	-0.0028	0.5695	0.2398	0.3297	0.0788
Brag1	0.1436	0.0315	0.1121	0.0158	0.3907	0.0326	0.3580	0.0023	0.5841	0.2753	0.3088	0.1060
Brag2	0.1456	0.0293	0.1163	0.0151	0.4223	0.0344	0.3879	0.0041	0.5735	0.2006	0.3730	0.0893
Brag3	0.1470	0.0293	0.1177	0.0164	0.4475	0.0436	0.4039	0.0071	0.5815	0.2047	0.3768	0.0909
Boost	0.1568	0.0074	0.1494	0.0009	0.3621	0.0089	0.3532	-0.0125	0.4848	0.0826	0.4022	0.0272
	$\sigma = 1.0$											
Standard	0.3014	0.0621	0.2393	0.0217	0.6619	0.0604	0.6016	0.0021	0.9352	0.2648	0.6704	0.1168
Bag1	0.3107	0.0489	0.2618	0.0212	0.7978	0.0399	0.7579	-0.0057	0.9321	0.2356	0.6965	0.1155
Bag2	0.3059	0.0495	0.2563	0.0243	0.7485	0.0710	0.6775	-0.0096	0.9131	0.2676	0.6455	0.1098
Bag3	0.3043	0.0496	0.2547	0.0233	0.7276	0.0884	0.6391	-0.0093	0.8942	0.2710	0.6233	0.1058
Brag1	0.3055	0.0533	0.2522	0.0227	0.7615	0.0444	0.7171	-0.0026	0.9146	0.3226	0.5920	0.1385
Brag2	0.3113	0.0497	0.2617	0.0229	0.8064	0.0459	0.7605	-0.0045	0.9552	0.2423	0.7130	0.1176
Brag3	0.3092	0.0499	0.2593	0.0229	0.8100	0.0571	0.7529	0.0015	0.9554	0.2515	0.7039	0.1203
Boost	0.3638	0.0174	0.3463	0.0024	0.8421	0.0180	0.8242	-0.0210	1.0266	0.1151	0.9114	0.0451

Table B.4: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 1, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0288	0.0091	0.0197	0.0010	0.0593	0.0158	0.0434	0.0010	0.2458	0.1584	0.0874	-0.0005
Bag1	0.0283	0.0084	0.0199	0.0010	0.0572	0.0128	0.0444	0.0010	0.2233	0.1367	0.0866	-0.0011
Bag2	0.0293	0.0092	0.0201	0.0024	0.0602	0.0156	0.0446	0.0003	0.2538	0.1617	0.0921	-0.0034
Bag3	0.0294	0.0093	0.0201	0.0019	0.0605	0.0160	0.0444	-0.0005	0.2551	0.1636	0.0915	-0.0011
Brag1	0.0283	0.0084	0.0199	0.0009	0.0574	0.0132	0.0442	0.0011	0.2264	0.1397	0.0867	-0.0008
Brag2	0.0292	0.0088	0.0204	0.0030	0.0598	0.0140	0.0457	-0.0010	0.2380	0.1459	0.0921	-0.0042
Brag3	0.0292	0.0088	0.0204	0.0027	0.0596	0.0139	0.0457	-0.0016	0.2388	0.1469	0.0920	-0.0040
Boost	0.0214	0.0008	0.0206	-0.0006	0.0451	0.0023	0.0428	0.0018	0.1279	0.0582	0.0697	-0.0027
	$\sigma = 0.6$											
Standard	0.1214	0.0277	0.0937	0.0011	0.2238	0.0419	0.1819	0.0069	0.4801	0.2410	0.2391	-0.0025
Bag1	0.1203	0.0238	0.0966	0.0011	0.2301	0.0295	0.2006	0.0070	0.4547	0.2091	0.2456	-0.0042
Bag2	0.1225	0.0255	0.0970	0.0046	0.2267	0.0331	0.1937	0.0044	0.4830	0.2363	0.2467	-0.0055
Bag3	0.1226	0.0255	0.0971	0.0039	0.2274	0.0335	0.1939	0.0036	0.4824	0.2369	0.2455	-0.0035
Brag1	0.1204	0.0236	0.0968	0.0010	0.2293	0.0303	0.1989	0.0071	0.4547	0.2091	0.2456	-0.0040
Brag2	0.1238	0.0249	0.0988	0.0055	0.2324	0.0324	0.2000	0.0027	0.4749	0.2206	0.2543	-0.0043
Brag3	0.1238	0.0253	0.0986	0.0050	0.2328	0.0325	0.2003	0.0024	0.4755	0.2225	0.2530	-0.0020
Boost	0.1266	0.0042	0.1224	-0.0014	0.2473	0.0102	0.2370	0.0072	0.3830	0.1063	0.2766	-0.0083
	$\sigma = 1.0$											
Standard	0.2598	0.0452	0.2146	0.0009	0.4764	0.0604	0.4161	0.0136	0.8114	0.3217	0.4897	-0.0007
Bag1	0.2623	0.0358	0.2265	0.0009	0.5201	0.0397	0.4804	0.0132	0.7852	0.2736	0.5116	-0.0030
Bag2	0.2659	0.0382	0.2277	0.0064	0.5051	0.0432	0.4619	0.0085	0.8104	0.3019	0.5085	-0.0041
Bag3	0.2669	0.0379	0.2290	0.0058	0.5086	0.0435	0.4651	0.0080	0.8079	0.2996	0.5084	-0.0013
Brag1	0.2629	0.0353	0.2276	0.0008	0.5180	0.0401	0.4778	0.0132	0.7834	0.2698	0.5136	-0.0033
Brag2	0.2699	0.0374	0.2325	0.0071	0.5211	0.0427	0.4784	0.0072	0.8136	0.2846	0.5290	0.0004
Brag3	0.2697	0.0383	0.2314	0.0061	0.5215	0.0440	0.4775	0.0055	0.8143	0.2906	0.5237	0.0019
Boost	0.3008	0.0091	0.2918	-0.0021	0.5914	0.0184	0.5730	0.0138	0.7790	0.1598	0.6191	-0.0102

Table B.5: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 2, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0292	0.0085	0.0207	0.0007	0.0603	0.0147	0.0456	0.0008	0.2184	0.1254	0.0931	0.0027
Bag1	0.0290	0.0084	0.0206	0.0011	0.0591	0.0126	0.0465	0.0009	0.2328	0.1409	0.0920	-0.0010
Bag2	0.0300	0.0090	0.0210	0.0025	0.0629	0.0152	0.0477	-0.0010	0.2582	0.1611	0.0971	-0.0035
Bag3	0.0298	0.0087	0.0210	0.0017	0.0624	0.0152	0.0472	-0.0033	0.2253	0.1325	0.0927	-0.0024
Brag1	0.0287	0.0080	0.0207	0.0010	0.0589	0.0123	0.0466	0.0010	0.2052	0.1134	0.0918	0.0009
Brag2	0.0296	0.0082	0.0213	0.0029	0.0613	0.0128	0.0485	-0.0014	0.2156	0.1180	0.0976	-0.0027
Brag3	0.0292	0.0088	0.0204	0.0027	0.0596	0.0139	0.0457	-0.0016	0.2388	0.1469	0.0920	-0.0040
Boost	0.0218	0.0007	0.0211	-0.0008	0.0459	0.0021	0.0438	0.0017	0.1104	0.0370	0.0734	-0.0009
	$\sigma = 0.6$											
Standard	0.1214	0.0265	0.0949	0.0004	0.2255	0.0394	0.1861	0.0065	0.4568	0.1964	0.2605	0.0026
Bag1	0.1199	0.0241	0.0958	0.0010	0.2300	0.0292	0.2008	0.0068	0.4642	0.2122	0.2520	-0.0028
Bag2	0.1223	0.0255	0.0968	0.0044	0.2290	0.0321	0.1969	0.0039	0.4885	0.2346	0.2539	-0.0037
Bag3	0.1223	0.0244	0.0979	0.0033	0.2296	0.0316	0.1980	0.0020	0.4532	0.1944	0.2588	-0.0027
Brag1	0.1201	0.0228	0.0973	0.0008	0.2310	0.0286	0.2024	0.0066	0.4374	0.1741	0.2633	-0.0021
Brag2	0.1235	0.0239	0.0995	0.0049	0.2353	0.0304	0.2049	0.0025	0.4537	0.1826	0.2711	-0.0020
Brag3	0.1238	0.0253	0.0986	0.0050	0.2328	0.0325	0.2003	0.0024	0.4755	0.2225	0.2530	-0.0020
Boost	0.1297	0.0037	0.1260	-0.0018	0.2554	0.0090	0.2463	0.0074	0.3866	0.0692	0.3174	-0.0042
	$\sigma = 1.0$											
Standard	0.2602	0.0435	0.2167	-0.0001	0.4825	0.0567	0.4258	0.0127	0.8043	0.2617	0.5427	0.0076
Bag1	0.2600	0.0363	0.2237	0.0005	0.5178	0.0392	0.4786	0.0127	0.7965	0.2712	0.5253	-0.0011
Bag2	0.2643	0.0384	0.2260	0.0060	0.5082	0.0420	0.4663	0.0080	0.8195	0.2947	0.5248	-0.0019
Bag3	0.2662	0.0365	0.2297	0.0048	0.5140	0.0412	0.4728	0.0063	0.7945	0.2454	0.5491	0.0002
Brag1	0.2619	0.0342	0.2277	0.0002	0.5233	0.0379	0.4853	0.0124	0.7851	0.2241	0.5610	-0.0021
Brag2	0.2693	0.0361	0.2332	0.0067	0.5285	0.0401	0.4884	0.0063	0.8105	0.2361	0.5744	0.0016
Brag3	0.2697	0.0383	0.2314	0.0061	0.5215	0.0440	0.4775	0.0055	0.8143	0.2906	0.5237	0.0019
Boost	0.3100	0.0080	0.3020	-0.0026	0.6168	0.0162	0.6006	0.0142	0.8322	0.1053	0.7269	-0.0019

Table B.6: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 2, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0396	0.0160	0.0236	0.0156	0.0732	0.0179	0.0553	0.0098	0.3110	0.1919	0.1191	0.0713
Bag1	0.0368	0.0122	0.0245	0.0136	0.0721	0.0109	0.0612	0.0052	0.2726	0.1531	0.1195	0.0621
Bag2	0.0372	0.0133	0.0239	0.0149	0.0705	0.0153	0.0552	0.0059	0.3004	0.1850	0.1154	0.0675
Bag3	0.0373	0.0135	0.0238	0.0145	0.0701	0.0162	0.0539	0.0057	0.2959	0.1831	0.1128	0.0681
Brag1	0.0369	0.0125	0.0244	0.0138	0.0721	0.0116	0.0605	0.0055	0.2778	0.1583	0.1195	0.0635
Brag2	0.0377	0.0128	0.0249	0.0146	0.0742	0.0124	0.0618	0.0046	0.2858	0.1634	0.1223	0.0652
Brag3	0.0376	0.0127	0.0249	0.0142	0.0739	0.0124	0.0615	0.0045	0.2805	0.1604	0.1201	0.0625
Boost	0.0272	0.0019	0.0254	0.0027	0.0612	0.0027	0.0585	-0.0013	0.1762	0.0739	0.1023	0.0226
	$\sigma = 0.6$											
Standard	0.1509	0.0405	0.1105	0.0248	0.2809	0.0405	0.2405	0.0092	0.5842	0.2718	0.3124	0.1044
Bag1	0.1512	0.0324	0.1188	0.0223	0.3055	0.0248	0.2807	0.0032	0.5507	0.2200	0.3307	0.0942
Bag2	0.1480	0.0343	0.1137	0.0245	0.2736	0.0295	0.2441	0.0020	0.5645	0.2589	0.3056	0.0977
Bag3	0.1479	0.0344	0.1135	0.0238	0.2722	0.0305	0.2417	0.0018	0.5563	0.2555	0.3008	0.0977
Brag1	0.1510	0.0326	0.1184	0.0224	0.3035	0.0261	0.2774	0.0037	0.5526	0.2224	0.3302	0.0947
Brag2	0.1519	0.0337	0.1182	0.0233	0.3012	0.0274	0.2738	0.0027	0.5640	0.2325	0.3315	0.1007
Brag3	0.1518	0.0339	0.1179	0.0231	0.2979	0.0280	0.2699	0.0034	0.5570	0.2321	0.3249	0.1009
Boost	0.1601	0.0082	0.1520	0.0049	0.3377	0.0090	0.3286	-0.0087	0.5046	0.1152	0.3894	0.0420
	$\sigma = 1.0$											
Standard	0.3140	0.0713	0.2428	0.0334	0.6017	0.0640	0.5378	0.0064	0.9864	0.3668	0.6196	0.1408
Bag1	0.3216	0.0568	0.2648	0.0302	0.6898	0.0375	0.6523	-0.0026	0.9658	0.2970	0.6688	0.1279
Bag2	0.3150	0.0596	0.2554	0.0336	0.6112	0.0424	0.5688	-0.0048	0.9592	0.3418	0.6174	0.1305
Bag3	0.3148	0.0594	0.2554	0.0329	0.6085	0.0431	0.5654	-0.0044	0.9439	0.3333	0.6106	0.1318
Brag1	0.3213	0.0567	0.2646	0.0303	0.6861	0.0390	0.6471	-0.0022	0.9657	0.2952	0.6705	0.1276
Brag2	0.3224	0.0588	0.2636	0.0314	0.6761	0.0404	0.6357	-0.0031	0.9806	0.3098	0.6708	0.1341
Brag3	0.3217	0.0593	0.2623	0.0314	0.6612	0.0419	0.6193	-0.0013	0.9685	0.3122	0.6563	0.1375
Boost	0.3653	0.0190	0.3463	0.0073	0.7897	0.0189	0.7708	-0.0189	1.0254	0.1682	0.8572	0.0605

Table B.7: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 2, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0406	0.0154	0.0253	0.0143	0.0807	0.0170	0.0637	0.0064	0.2546	0.1352	0.1194	0.0554
Bag1	0.0389	0.0131	0.0258	0.0144	0.0821	0.0121	0.0700	0.0039	0.2800	0.1628	0.1172	0.0632
Bag2	0.0405	0.0146	0.0259	0.0165	0.0823	0.0201	0.0622	0.0044	0.2942	0.1834	0.1107	0.0671
Bag3	0.0395	0.0138	0.0257	0.0148	0.0797	0.0213	0.0583	0.0031	0.2407	0.1387	0.1021	0.0571
Brag1	0.0382	0.0123	0.0260	0.0134	0.0827	0.0118	0.0709	0.0037	0.2412	0.1223	0.1189	0.0530
Brag2	0.0393	0.0126	0.0267	0.0142	0.0854	0.0129	0.0725	0.0032	0.2464	0.1250	0.1213	0.0546
Brag3	0.0376	0.0127	0.0249	0.0142	0.0739	0.0124	0.0615	0.0045	0.2805	0.1604	0.1201	0.0625
Boost	0.0275	0.0017	0.0258	0.0024	0.0633	0.0024	0.0609	-0.0015	0.1472	0.0430	0.1042	0.0163
	$\sigma = 0.6$											
Standard	0.1529	0.0394	0.1136	0.0223	0.2946	0.0370	0.2576	0.0083	0.5411	0.1953	0.3459	0.0859
Bag1	0.1521	0.0348	0.1172	0.0227	0.3144	0.0252	0.2893	0.0026	0.5620	0.2351	0.3269	0.0958
Bag2	0.1500	0.0366	0.1134	0.0258	0.2836	0.0311	0.2525	0.0011	0.5584	0.2592	0.2992	0.0973
Bag3	0.1491	0.0340	0.1150	0.0236	0.2778	0.0307	0.2471	-0.0004	0.4939	0.1932	0.3007	0.0838
Brag1	0.1527	0.0323	0.1204	0.0212	0.3201	0.0239	0.2963	0.0014	0.5295	0.1751	0.3543	0.0834
Brag2	0.1544	0.0333	0.1210	0.0220	0.3190	0.0252	0.2938	0.0002	0.5334	0.1812	0.3522	0.0870
Brag3	0.1518	0.0339	0.1179	0.0231	0.2979	0.0280	0.2699	0.0034	0.5570	0.2321	0.3249	0.1009
Boost	0.1628	0.0076	0.1552	0.0041	0.3547	0.0077	0.3470	-0.0076	0.5088	0.0681	0.4407	0.0329
	$\sigma = 1.0$											
Standard	0.3182	0.0700	0.2482	0.0300	0.6390	0.0582	0.5807	0.0054	0.9845	0.2676	0.7169	0.1174
Bag1	0.3214	0.0613	0.2601	0.0306	0.7056	0.0376	0.6680	-0.0027	0.9830	0.3156	0.6674	0.1273
Bag2	0.3168	0.0637	0.2531	0.0346	0.6291	0.0428	0.5863	-0.0057	0.9585	0.3446	0.6139	0.1281
Bag3	0.3171	0.0585	0.2586	0.0324	0.6212	0.0409	0.5803	-0.0064	0.8938	0.2530	0.6408	0.1113
Brag1	0.3244	0.0563	0.2681	0.0290	0.7257	0.0350	0.6907	-0.0022	0.9789	0.2344	0.7445	0.1151
Brag2	0.3261	0.0580	0.2681	0.0298	0.7177	0.0365	0.6812	-0.0034	0.9802	0.2437	0.7364	0.1202
Brag3	0.3217	0.0593	0.2623	0.0314	0.6612	0.0419	0.6193	-0.0013	0.9685	0.3122	0.6563	0.1375
Boost	0.3732	0.0176	0.3556	0.0060	0.8421	0.0159	0.8263	-0.0149	1.1004	0.1016	0.9988	0.0513

Table B.8: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 2, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0049	0.0011	0.0038	-0.0010	0.0092	0.0017	0.0075	0.0009	0.0362	0.0257	0.0105	0.0031
Bag1	0.0049	0.0010	0.0038	-0.0010	0.0092	0.0013	0.0078	0.0009	0.0316	0.0209	0.0107	0.0002
Bag2	0.0049	0.0011	0.0038	-0.0005	0.0092	0.0016	0.0076	0.0005	0.0338	0.0228	0.0110	0.0030
Bag3	0.0049	0.0011	0.0038	-0.0007	0.0091	0.0016	0.0075	0.0001	0.0334	0.0226	0.0108	0.0024
Brag1	0.0049	0.0010	0.0038	-0.0010	0.0091	0.0014	0.0078	0.0009	0.0321	0.0214	0.0107	0.0006
Brag2	0.0050	0.0011	0.0039	-0.0002	0.0093	0.0014	0.0079	0.0001	0.0331	0.0220	0.0112	0.0006
Brag3	0.0050	0.0011	0.0039	-0.0004	0.0093	0.0014	0.0079	-0.0002	0.0328	0.0216	0.0112	0.0004
Boost	0.0048	0.0001	0.0046	-0.0007	0.0092	0.0002	0.0089	0.0014	0.0237	0.0127	0.0111	-0.0049
	$\sigma = 0.3$											
Standard	0.0243	0.0040	0.0203	-0.0018	0.0435	0.0046	0.0389	0.0042	0.0786	0.0391	0.0395	0.0087
Bag1	0.0246	0.0033	0.0212	-0.0017	0.0474	0.0030	0.0444	0.0044	0.0751	0.0331	0.0420	0.0043
Bag2	0.0247	0.0034	0.0213	-0.0001	0.0459	0.0034	0.0426	0.0030	0.0769	0.0351	0.0419	0.0076
Bag3	0.0248	0.0034	0.0214	-0.0003	0.0461	0.0033	0.0427	0.0026	0.0757	0.0336	0.0421	0.0070
Brag1	0.0246	0.0033	0.0214	-0.0017	0.0471	0.0031	0.0440	0.0044	0.0745	0.0319	0.0426	0.0040
Brag2	0.0251	0.0034	0.0217	-0.0001	0.0474	0.0032	0.0442	0.0024	0.0768	0.0329	0.0438	0.0058
Brag3	0.0251	0.0034	0.0216	-0.0002	0.0474	0.0032	0.0442	0.0021	0.0766	0.0334	0.0433	0.0063
Boost	0.0288	0.0008	0.0280	-0.0014	0.0549	0.0011	0.0538	0.0044	0.0757	0.0228	0.0529	-0.0049
	$\sigma = 0.5$											
Standard	0.0547	0.0062	0.0486	-0.0022	0.1008	0.0059	0.0949	0.0076	0.1383	0.0472	0.0911	0.0133
Bag1	0.0570	0.0048	0.0522	-0.0021	0.1165	0.0037	0.1127	0.0075	0.1384	0.0399	0.0985	0.0075
Bag2	0.0574	0.0049	0.0525	0.0004	0.1122	0.0040	0.1081	0.0051	0.1399	0.0418	0.0981	0.0110
Bag3	0.0577	0.0048	0.0529	0.0004	0.1132	0.0040	0.1092	0.0048	0.1394	0.0394	0.1000	0.0101
Brag1	0.0574	0.0047	0.0528	-0.0021	0.1160	0.0037	0.1122	0.0075	0.1391	0.0378	0.1012	0.0063
Brag2	0.0586	0.0048	0.0539	0.0003	0.1163	0.0038	0.1124	0.0044	0.1428	0.0389	0.1039	0.0104
Brag3	0.0583	0.0050	0.0533	0.0001	0.1160	0.0039	0.1120	0.0036	0.1412	0.0397	0.1014	0.0116
Boost	0.0699	0.0017	0.0683	-0.0020	0.1368	0.0018	0.1350	0.0080	0.1561	0.0295	0.1265	-0.0034

Table B.9: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 3, $X \sim U(-2, 2)$, $N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0049	0.0011	0.0038	-0.0011	0.0092	0.0016	0.0076	0.0008	0.0271	0.0146	0.0126	-0.0035
Bag1	0.0049	0.0010	0.0039	-0.0010	0.0092	0.0012	0.0080	0.0009	0.0260	0.0137	0.0123	-0.0037
Bag2	0.0050	0.0011	0.0039	-0.0005	0.0093	0.0015	0.0078	-0.0004	0.0302	0.0182	0.0120	-0.0016
Bag3	0.0049	0.0011	0.0039	-0.0008	0.0092	0.0015	0.0077	-0.0019	0.0269	0.0150	0.0119	-0.0053
Brag1	0.0049	0.0010	0.0038	-0.0010	0.0092	0.0013	0.0079	0.0009	0.0263	0.0139	0.0124	-0.0037
Brag2	0.0050	0.0010	0.0039	-0.0003	0.0094	0.0013	0.0081	-0.0002	0.0272	0.0142	0.0129	-0.0042
Brag3	0.0050	0.0010	0.0039	-0.0004	0.0093	0.0013	0.0080	-0.0009	0.0268	0.0142	0.0126	-0.0059
Boost	0.0048	0.0001	0.0047	-0.0007	0.0093	0.0002	0.0091	0.0013	0.0182	0.0049	0.0133	-0.0036
	$\sigma = 0.3$											
Standard	0.0242	0.0039	0.0202	-0.0019	0.0438	0.0044	0.0393	0.0040	0.0765	0.0256	0.0508	-0.0027
Bag1	0.0244	0.0033	0.0211	-0.0017	0.0478	0.0028	0.0449	0.0042	0.0744	0.0245	0.0500	-0.0032
Bag2	0.0245	0.0035	0.0210	-0.0002	0.0461	0.0032	0.0429	0.0027	0.0763	0.0303	0.0460	0.0012
Bag3	0.0246	0.0033	0.0213	-0.0005	0.0464	0.0031	0.0433	0.0016	0.0736	0.0248	0.0487	-0.0025
Brag1	0.0244	0.0032	0.0212	-0.0017	0.0473	0.0029	0.0444	0.0042	0.0743	0.0235	0.0508	-0.0032
Brag2	0.0249	0.0033	0.0216	-0.0002	0.0478	0.0030	0.0448	0.0022	0.0763	0.0241	0.0522	-0.0017
Brag3	0.0248	0.0034	0.0215	-0.0003	0.0474	0.0030	0.0444	0.0012	0.0754	0.0247	0.0507	-0.0030
Boost	0.0293	0.0008	0.0285	-0.0014	0.0566	0.0010	0.0556	0.0043	0.0821	0.0114	0.0707	-0.0073
	$\sigma = 0.5$											
Standard	0.0547	0.0061	0.0486	-0.0025	0.1023	0.0057	0.0966	0.0074	0.1493	0.0331	0.1162	-0.0019
Bag1	0.0567	0.0047	0.0520	-0.0023	0.1179	0.0035	0.1144	0.0074	0.1473	0.0311	0.1162	-0.0022
Bag2	0.0568	0.0049	0.0518	0.0003	0.1128	0.0039	0.1089	0.0047	0.1441	0.0374	0.1067	0.0040
Bag3	0.0575	0.0047	0.0528	0.0001	0.1144	0.0037	0.1107	0.0037	0.1462	0.0306	0.1156	-0.0007
Brag1	0.0570	0.0046	0.0524	-0.0022	0.1172	0.0036	0.1136	0.0072	0.1487	0.0295	0.1193	-0.0028
Brag2	0.0583	0.0047	0.0537	0.0003	0.1180	0.0036	0.1144	0.0041	0.1524	0.0302	0.1222	0.0004
Brag3	0.0579	0.0048	0.0530	0.0001	0.1166	0.0037	0.1129	0.0027	0.1494	0.0311	0.1183	0.0003
Boost	0.0718	0.0016	0.0703	-0.0020	0.1427	0.0017	0.1410	0.0079	0.1858	0.0168	0.1690	-0.0103

Table B.10: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 3, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0076	0.0033	0.0043	0.0081	0.0127	0.0036	0.0091	0.0085	0.0523	0.0352	0.0171	0.0413
Bag1	0.0069	0.0024	0.0045	0.0069	0.0122	0.0020	0.0102	0.0062	0.0416	0.0255	0.0161	0.0358
Bag2	0.0069	0.0026	0.0044	0.0086	0.0115	0.0025	0.0090	0.0095	0.0456	0.0296	0.0160	0.0466
Bag3	0.0069	0.0026	0.0043	0.0083	0.0114	0.0026	0.0088	0.0084	0.0436	0.0283	0.0153	0.0430
Brag1	0.0069	0.0024	0.0045	0.0070	0.0122	0.0021	0.0101	0.0064	0.0413	0.0252	0.0161	0.0361
Brag2	0.0071	0.0025	0.0046	0.0077	0.0123	0.0023	0.0100	0.0067	0.0428	0.0261	0.0167	0.0373
Brag3	0.0070	0.0025	0.0045	0.0076	0.0121	0.0023	0.0098	0.0065	0.0425	0.0261	0.0164	0.0365
Boost	0.0055	0.0005	0.0050	0.0017	0.0117	0.0007	0.0110	0.0022	0.0388	0.0198	0.0190	0.0335
	$\sigma = 0.3$											
Standard	0.0304	0.0087	0.0217	0.0153	0.0550	0.0084	0.0467	0.0102	0.1054	0.0518	0.0536	0.0585
Bag1	0.0300	0.0063	0.0237	0.0131	0.0617	0.0044	0.0573	0.0056	0.0981	0.0408	0.0573	0.0537
Bag2	0.0297	0.0067	0.0230	0.0155	0.0549	0.0051	0.0498	0.0085	0.0998	0.0454	0.0544	0.0652
Bag3	0.0296	0.0066	0.0230	0.0153	0.0544	0.0050	0.0494	0.0077	0.0955	0.0419	0.0536	0.0610
Brag1	0.0300	0.0063	0.0237	0.0131	0.0615	0.0046	0.0569	0.0057	0.0967	0.0388	0.0579	0.0524
Brag2	0.0302	0.0066	0.0236	0.0147	0.0606	0.0049	0.0557	0.0065	0.0994	0.0405	0.0588	0.0547
Brag3	0.0301	0.0066	0.0235	0.0148	0.0590	0.0050	0.0540	0.0070	0.0980	0.0405	0.0575	0.0536
Boost	0.0322	0.0023	0.0298	0.0039	0.0700	0.0028	0.0672	-0.0018	0.1113	0.0337	0.0776	0.0559
	$\sigma = 0.5$											
Standard	0.0620	0.0128	0.0492	0.0203	0.1209	0.0108	0.1102	0.0090	0.1730	0.0603	0.1127	0.0696
Bag1	0.0640	0.0089	0.0551	0.0171	0.1485	0.0054	0.1432	0.0021	0.1734	0.0481	0.1253	0.0650
Bag2	0.0631	0.0093	0.0538	0.0203	0.1309	0.0060	0.1249	0.0043	0.1713	0.0528	0.1186	0.0763
Bag3	0.0630	0.0089	0.0540	0.0202	0.1303	0.0057	0.1245	0.0036	0.1670	0.0479	0.1191	0.0714
Brag1	0.0641	0.0087	0.0554	0.0170	0.1483	0.0054	0.1429	0.0023	0.1737	0.0456	0.1281	0.0632
Brag2	0.0641	0.0090	0.0551	0.0190	0.1455	0.0059	0.1396	0.0034	0.1770	0.0474	0.1296	0.0654
Brag3	0.0638	0.0091	0.0547	0.0198	0.1407	0.0060	0.1348	0.0039	0.1730	0.0472	0.1258	0.0635
Boost	0.0744	0.0044	0.0700	0.0065	0.1697	0.0045	0.1652	-0.0057	0.2125	0.0419	0.1706	0.0706

Table B.11: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 3, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0073	0.0028	0.0045	0.0067	0.0125	0.0028	0.0096	0.0057	0.0433	0.0246	0.0187	0.0292
Bag1	0.0068	0.0022	0.0047	0.0060	0.0123	0.0016	0.0108	0.0037	0.0357	0.0190	0.0167	0.0253
Bag2	0.0071	0.0027	0.0044	0.0079	0.0113	0.0021	0.0092	0.0023	0.0428	0.0267	0.0162	0.0294
Bag3	0.0068	0.0023	0.0044	0.0068	0.0107	0.0019	0.0088	-0.0019	0.0340	0.0196	0.0144	0.0194
Brag1	0.0069	0.0022	0.0046	0.0062	0.0124	0.0017	0.0107	0.0040	0.0353	0.0185	0.0168	0.0251
Brag2	0.0070	0.0023	0.0047	0.0066	0.0124	0.0018	0.0107	0.0024	0.0362	0.0188	0.0174	0.0239
Brag3	0.0069	0.0022	0.0046	0.0063	0.0117	0.0017	0.0100	0.0006	0.0352	0.0186	0.0166	0.0199
Boost	0.0056	0.0004	0.0053	0.0016	0.0121	0.0005	0.0116	0.0020	0.0312	0.0116	0.0196	0.0280
	$\sigma = 0.3$											
Standard	0.0308	0.0079	0.0229	0.0131	0.0576	0.0069	0.0507	0.0071	0.1029	0.0380	0.0649	0.0469
Bag1	0.0305	0.0060	0.0245	0.0119	0.0649	0.0036	0.0613	0.0033	0.0959	0.0318	0.0641	0.0422
Bag2	0.0301	0.0072	0.0230	0.0150	0.0559	0.0047	0.0513	0.0032	0.0986	0.0430	0.0556	0.0502
Bag3	0.0298	0.0062	0.0236	0.0137	0.0548	0.0039	0.0508	-0.0004	0.0871	0.0310	0.0561	0.0400
Brag1	0.0305	0.0060	0.0245	0.0119	0.0648	0.0037	0.0610	0.0031	0.0956	0.0304	0.0652	0.0415
Brag2	0.0308	0.0062	0.0246	0.0132	0.0638	0.0040	0.0599	0.0021	0.0968	0.0313	0.0655	0.0412
Brag3	0.0303	0.0062	0.0241	0.0131	0.0594	0.0040	0.0554	0.0007	0.0929	0.0308	0.0621	0.0383
Boost	0.0335	0.0018	0.0316	0.0034	0.0755	0.0020	0.0734	-0.0019	0.1101	0.0206	0.0894	0.0456
	$\sigma = 0.5$											
Standard	0.0638	0.0119	0.0520	0.0175	0.1309	0.0091	0.1219	0.0058	0.1880	0.0448	0.1432	0.0588
Bag1	0.0657	0.0085	0.0572	0.0156	0.1594	0.0044	0.1550	-0.0005	0.1843	0.0380	0.1463	0.0534
Bag2	0.0636	0.0101	0.0535	0.0197	0.1349	0.0056	0.1293	-0.0008	0.1736	0.0507	0.1230	0.0621
Bag3	0.0642	0.0085	0.0557	0.0183	0.1339	0.0047	0.1292	-0.0043	0.1668	0.0360	0.1308	0.0517
Brag1	0.0656	0.0084	0.0572	0.0157	0.1599	0.0045	0.1554	-0.0004	0.1858	0.0362	0.1496	0.0527
Brag2	0.0658	0.0086	0.0571	0.0172	0.1570	0.0048	0.1523	-0.0012	0.1871	0.0373	0.1498	0.0526
Brag3	0.0648	0.0087	0.0561	0.0176	0.1443	0.0049	0.1394	-0.0030	0.1773	0.0363	0.1409	0.0494
Boost	0.0780	0.0035	0.0745	0.0054	0.1867	0.0034	0.1833	-0.0057	0.2383	0.0260	0.2123	0.0571

Table B.12: Nadaraya-Watson estimator with plug-in bandwidth selection, Model 3, $X \sim N(0, 1)$, Triangular kernel

Appendix C

Results of Simulation Study C

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0256	0.0078	0.0178	-0.0012	0.0648	0.0140	0.0508	0.0042	0.2287	0.1607	0.0680	-0.0016
Bag1	0.0253	0.0072	0.0181	-0.0013	0.0726	0.0120	0.0606	0.0055	0.2124	0.1366	0.0758	0.0012
Bag2	0.0259	0.0075	0.0184	-0.0005	0.0878	0.0221	0.0657	0.0002	0.2334	0.1458	0.0876	0.0043
Bag3	0.0261	0.0077	0.0184	-0.0006	0.0884	0.0247	0.0637	-0.0045	0.2350	0.1491	0.0858	-0.0073
Brag1	0.0254	0.0072	0.0182	-0.0013	0.0706	0.0121	0.0585	0.0061	0.2122	0.1367	0.0755	0.0019
Brag2	0.0262	0.0073	0.0189	0.0001	0.0820	0.0138	0.0682	0.0032	0.2225	0.1349	0.0876	0.0061
Brag3	0.0262	0.0073	0.0189	0.0004	0.0825	0.0140	0.0685	0.0028	0.2189	0.1335	0.0854	0.0006
Boost	0.0241	0.0007	0.0234	-0.0021	0.0647	0.0030	0.0617	0.0041	0.1570	0.0786	0.0785	-0.0002
	$\sigma = 0.6$											
Standard	0.1226	0.0235	0.0991	0.0009	0.2442	0.0358	0.2083	0.0102	0.4627	0.2128	0.2499	-0.0099
Bag1	0.1229	0.0201	0.1029	0.0007	0.2655	0.0251	0.2404	0.0120	0.4409	0.1792	0.2617	-0.0090
Bag2	0.1228	0.0207	0.1021	0.0034	0.2571	0.0281	0.2290	0.0097	0.4378	0.1829	0.2549	-0.0043
Bag3	0.1233	0.0209	0.1025	0.0035	0.2563	0.0311	0.2252	0.0048	0.4301	0.1802	0.2499	-0.0167
Brag1	0.1232	0.0198	0.1034	0.0007	0.2638	0.0258	0.2380	0.0125	0.4391	0.1716	0.2674	-0.0066
Brag2	0.1268	0.0202	0.1066	0.0047	0.2719	0.0261	0.2457	0.0085	0.4439	0.1697	0.2742	-0.0043
Brag3	0.1264	0.0207	0.1057	0.0054	0.2713	0.0269	0.2444	0.0071	0.4405	0.1741	0.2663	-0.0105
Boost	0.1395	0.0041	0.1354	-0.0015	0.2931	0.0100	0.2831	0.0102	0.4633	0.1216	0.3418	-0.0066
	$\sigma = 1.0$											
Standard	0.2742	0.0373	0.2369	0.0026	0.5495	0.0507	0.4988	0.0181	0.8429	0.2474	0.5955	-0.0122
Bag1	0.2817	0.0294	0.2523	0.0026	0.6214	0.0336	0.5878	0.0204	0.8278	0.2052	0.6226	-0.0150
Bag2	0.2816	0.0303	0.2513	0.0068	0.5878	0.0352	0.5526	0.0172	0.8050	0.2064	0.5986	-0.0077
Bag3	0.2832	0.0302	0.2530	0.0071	0.5832	0.0378	0.5454	0.0118	0.7869	0.1991	0.5878	-0.0189
Brag1	0.2822	0.0289	0.2533	0.0023	0.6187	0.0339	0.5847	0.0207	0.8356	0.1937	0.6419	-0.0120
Brag2	0.2909	0.0295	0.2614	0.0083	0.6285	0.0340	0.5945	0.0141	0.8398	0.1922	0.6477	-0.0082
Brag3	0.2896	0.0305	0.2591	0.0087	0.6240	0.0358	0.5882	0.0112	0.8152	0.1994	0.6158	-0.0131
Boost	0.3334	0.0089	0.3245	-0.0007	0.7047	0.0174	0.6873	0.0180	0.9914	0.1561	0.8354	-0.0061

Table C.1: Local linear estimator with plug-in bandwidth selection, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0263	0.0076	0.0187	-0.0015	0.0719	0.0130	0.0589	0.0050	0.2442	0.1700	0.0742	-0.0004
Bag1	0.0260	0.0070	0.0190	-0.0017	0.0835	0.0109	0.0726	0.0062	0.2319	0.1437	0.0882	0.0028
Bag2	0.0279	0.0070	0.0208	-0.0010	0.1214	0.0263	0.0952	-0.0072	0.2573	0.1459	0.1114	0.0034
Bag3	0.0276	0.0071	0.0204	-0.0011	0.1191	0.0296	0.0895	-0.0120	0.2511	0.1491	0.1020	-0.0092
Brag1	0.0261	0.0070	0.0191	-0.0016	0.0815	0.0111	0.0704	0.0065	0.2320	0.1436	0.0884	0.0035
Brag2	0.0280	0.0069	0.0212	-0.0014	0.1271	0.0170	0.1101	0.0007	0.2702	0.1401	0.1300	0.0106
Brag3	0.0281	0.0069	0.0212	-0.0011	0.1254	0.0170	0.1084	0.0013	0.2549	0.1372	0.1177	0.0040
Boost	0.0254	0.0007	0.0247	-0.0024	0.0819	0.0031	0.0788	0.0020	0.1890	0.0928	0.0963	0.0029
	$\sigma = 0.6$											
Standard	0.1238	0.0239	0.0999	0.0018	0.2605	0.0360	0.2245	0.0142	0.5267	0.2282	0.2985	-0.0058
Bag1	0.1247	0.0203	0.1044	0.0016	0.2903	0.0248	0.2655	0.0171	0.5052	0.1917	0.3135	-0.0055
Bag2	0.1255	0.0203	0.1052	0.0036	0.2966	0.0282	0.2684	0.0123	0.5013	0.1858	0.3155	-0.0037
Bag3	0.1264	0.0206	0.1058	0.0036	0.2899	0.0322	0.2577	0.0042	0.4765	0.1821	0.2945	-0.0188
Brag1	0.1246	0.0201	0.1045	0.0014	0.2864	0.0254	0.2609	0.0168	0.5019	0.1833	0.3186	-0.0023
Brag2	0.1283	0.0200	0.1083	0.0044	0.3049	0.0268	0.2782	0.0107	0.5120	0.1766	0.3354	0.0059
Brag3	0.1282	0.0206	0.1076	0.0051	0.2992	0.0280	0.2712	0.0084	0.4923	0.1806	0.3117	-0.0039
Boost	0.1413	0.0046	0.1367	-0.0007	0.3289	0.0111	0.3178	0.0137	0.6115	0.1434	0.4681	0.0003
	$\sigma = 1.0$											
Standard	0.2767	0.0387	0.2380	0.0044	0.5755	0.0521	0.5234	0.0239	0.9850	0.2654	0.7196	-0.0055
Bag1	0.2853	0.0302	0.2551	0.0038	0.6725	0.0334	0.6391	0.0244	0.9789	0.2205	0.7585	-0.0084
Bag2	0.2860	0.0305	0.2555	0.0080	0.6655	0.0348	0.6307	0.0216	0.9490	0.2118	0.7371	-0.0062
Bag3	0.2890	0.0305	0.2585	0.0080	0.6485	0.0384	0.6101	0.0118	0.8871	0.2026	0.6845	-0.0194
Brag1	0.2855	0.0297	0.2558	0.0040	0.6675	0.0338	0.6337	0.0251	0.9828	0.2079	0.7749	-0.0040
Brag2	0.2934	0.0297	0.2637	0.0085	0.6737	0.0347	0.6390	0.0187	0.9559	0.2008	0.7551	0.0059
Brag3	0.2924	0.0309	0.2615	0.0090	0.6575	0.0372	0.6203	0.0141	0.8944	0.2074	0.6870	-0.0036
Boost	0.3323	0.0101	0.3222	0.0008	0.7523	0.0199	0.7324	0.0237	1.3420	0.1813	1.1607	0.0050

Table C.2: Local linear estimator with plug-in bandwidth selection, Model 1, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0488	0.0153	0.0334	-0.0007	0.3130	0.0514	0.2616	-0.0189	0.4141	0.2071	0.2070	-0.0020
Bag1	0.0583	0.0118	0.0465	-0.0035	0.5435	0.1162	0.4273	-0.0102	0.4283	0.1721	0.2562	0.0014
Bag2	0.0727	0.0168	0.0559	-0.0065	0.7231	0.3594	0.3637	-0.0149	0.4958	0.2402	0.2556	-0.0011
Bag3	0.0698	0.0180	0.0518	-0.0062	0.6743	0.3692	0.3051	-0.0190	0.4829	0.2643	0.2187	-0.0036
Brag1	0.0548	0.0119	0.0429	-0.0037	0.5153	0.1061	0.4092	-0.0133	0.4222	0.1759	0.2463	0.0045
Brag2	0.0693	0.0128	0.0565	-0.0019	0.7672	0.2381	0.5291	-0.0070	0.4823	0.1939	0.2885	0.0030
Brag3	0.0699	0.0126	0.0573	-0.0016	0.7888	0.2528	0.5360	-0.0082	0.4620	0.1838	0.2782	0.0034
Boost	0.0428	0.0019	0.0410	-0.0002	0.3246	0.0273	0.2973	-0.0211	0.3423	0.1120	0.2303	0.0025
	$\sigma = 0.6$											
Standard	0.1505	0.0367	0.1138	0.0030	0.4221	0.0442	0.3779	-0.0099	0.6739	0.2521	0.4218	-0.0041
Bag1	0.1584	0.0288	0.1296	0.0033	0.5727	0.0550	0.5177	-0.0189	0.6800	0.2053	0.4747	0.0060
Bag2	0.1589	0.0286	0.1303	0.0039	0.6359	0.1639	0.4720	-0.0199	0.6605	0.2387	0.4218	0.0001
Bag3	0.1572	0.0289	0.1283	0.0032	0.6031	0.1882	0.4149	-0.0203	0.6340	0.2553	0.3786	-0.0139
Brag1	0.1571	0.0291	0.1281	0.0036	0.5653	0.0537	0.5116	-0.0170	0.6818	0.2045	0.4773	0.0085
Brag2	0.1603	0.0289	0.1314	0.0047	0.6875	0.1083	0.5793	-0.0058	0.7015	0.2142	0.4873	0.0082
Brag3	0.1613	0.0288	0.1326	0.0048	0.7061	0.1260	0.5800	-0.0098	0.6764	0.2135	0.4630	0.0064
Boost	0.1678	0.0085	0.1594	0.0012	0.5199	0.0153	0.5046	-0.0112	0.7342	0.1481	0.5861	0.0111
	$\sigma = 1.0$											
Standard	0.3172	0.0619	0.2552	0.0053	0.7862	0.0581	0.7281	-0.0183	1.1290	0.2926	0.8364	-0.0074
Bag1	0.3348	0.0486	0.2861	0.0060	0.9972	0.0529	0.9443	-0.0287	1.1787	0.2362	0.9425	0.0083
Bag2	0.3325	0.0477	0.2848	0.0088	0.9469	0.1221	0.8249	-0.0279	1.0598	0.2540	0.8058	-0.0021
Bag3	0.3305	0.0472	0.2834	0.0086	0.8969	0.1539	0.7431	-0.0268	1.0058	0.2600	0.7459	-0.0221
Brag1	0.3337	0.0485	0.2852	0.0063	0.9943	0.0538	0.9405	-0.0256	1.1915	0.2307	0.9608	0.0122
Brag2	0.3368	0.0484	0.2884	0.0073	1.0465	0.0914	0.9551	-0.0135	1.1553	0.2355	0.9198	0.0116
Brag3	0.3360	0.0487	0.2873	0.0079	1.0348	0.1164	0.9183	-0.0140	1.0960	0.2406	0.8554	0.0055
Boost	0.3757	0.0194	0.3562	0.0039	1.0296	0.0221	1.0075	-0.0226	1.4216	0.1853	1.2363	0.0156

Table C.3: Local linear estimator with plug-in bandwidth selection, Model 1, $X \sim N(0,1), N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0851	0.0166	0.0686	-0.0026	0.5080	0.0879	0.4202	-0.0099	0.4990	0.2294	0.2695	-0.0053
Bag1	0.1089	0.0137	0.0952	-0.0037	0.7630	0.1713	0.5918	-0.0119	0.5324	0.1964	0.3360	-0.0019
Bag2	0.1635	0.0435	0.1200	-0.0117	1.0479	0.5930	0.4549	-0.0151	0.6875	0.3450	0.3426	-0.0082
Bag3	0.1557	0.0458	0.1099	-0.0117	0.9742	0.5916	0.3826	-0.0212	0.6443	0.3657	0.2786	-0.0113
Brag1	0.1057	0.0138	0.0919	-0.0031	0.7330	0.1575	0.5755	-0.0086	0.5223	0.1991	0.3232	-0.0032
Brag2	0.1828	0.0256	0.1572	-0.0141	1.2544	0.4926	0.7618	-0.0191	0.7950	0.2944	0.5006	-0.0152
Brag3	0.1777	0.0249	0.1529	-0.0098	1.2437	0.5012	0.7425	-0.0256	0.7377	0.2738	0.4639	-0.0105
Boost	0.1051	0.0028	0.1023	-0.0052	0.5957	0.0502	0.5455	-0.0227	0.4947	0.1315	0.3632	0.0023
	$\sigma = 0.6$											
Standard	0.2005	0.0385	0.1620	0.0007	0.7115	0.0600	0.6515	-0.0322	0.8924	0.2765	0.6159	-0.0157
Bag1	0.2488	0.0300	0.2189	-0.0007	0.9797	0.0840	0.8956	-0.0140	0.9553	0.2293	0.7260	-0.0104
Bag2	0.2393	0.0347	0.2046	-0.0023	0.9988	0.3198	0.6790	-0.0231	0.9154	0.3162	0.5992	-0.0153
Bag3	0.2234	0.0362	0.1872	-0.0029	0.8897	0.3343	0.5554	-0.0256	0.8099	0.3317	0.4783	-0.0290
Brag1	0.2295	0.0302	0.1993	0.0000	0.9635	0.0809	0.8826	-0.0202	0.9475	0.2278	0.7197	-0.0071
Brag2	0.2439	0.0314	0.2125	-0.0017	1.1462	0.2675	0.8787	-0.0133	1.0151	0.2882	0.7269	-0.0226
Brag3	0.2187	0.0310	0.1877	0.0019	1.0788	0.2829	0.7959	-0.0126	0.8805	0.2776	0.6029	-0.0223
Boost	0.3147	0.0103	0.3045	-0.0031	1.1836	0.0238	1.1598	-0.0378	1.2651	0.1730	1.0921	0.0011
	$\sigma = 1.0$											
Standard	0.3460	0.0658	0.2802	0.0054	1.2421	0.0708	1.1713	-0.0404	1.6892	0.3183	1.3710	-0.0237
Bag1	0.4495	0.0515	0.3981	0.0053	1.6447	0.0777	1.5670	-0.0456	1.6920	0.2637	1.4283	-0.0101
Bag2	0.4355	0.0506	0.3849	0.0066	1.4227	0.2462	1.1765	-0.0436	1.4518	0.3217	1.1301	-0.0254
Bag3	0.4034	0.0510	0.3524	0.0059	1.2249	0.2666	0.9583	-0.0386	1.2399	0.3220	0.9178	-0.0456
Brag1	0.4278	0.0511	0.3767	0.0034	1.6113	0.0784	1.5329	-0.0443	1.7065	0.2579	1.4486	-0.0063
Brag2	0.4375	0.0503	0.3871	0.0068	1.5028	0.2163	1.2866	-0.0197	1.5393	0.2947	1.2446	-0.0303
Brag3	0.3832	0.0505	0.3327	0.0075	1.3528	0.2382	1.1145	-0.0108	1.2968	0.2971	0.9997	-0.0321
Boost	0.4270	0.0226	0.4044	0.0046	2.1962	0.0287	2.1675	-0.0384	3.1854	0.2161	2.9693	-0.0034

Table C.4: Local linear estimator with plug-in bandwidth selection, Model 1, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0233	0.0078	0.0155	0.0005	0.0479	0.0133	0.0346	0.0031	0.1750	0.1234	0.0516	-0.0009
Bag1	0.0230	0.0072	0.0158	0.0004	0.0481	0.0105	0.0376	0.0027	0.1559	0.1024	0.0535	-0.0012
Bag2	0.0232	0.0076	0.0157	0.0016	0.0501	0.0130	0.0372	0.0005	0.1654	0.1096	0.0558	-0.0039
Bag3	0.0233	0.0076	0.0157	0.0015	0.0510	0.0140	0.0370	-0.0036	0.1664	0.1118	0.0546	-0.0069
Brag1	0.0230	0.0072	0.0158	0.0004	0.0479	0.0109	0.0370	0.0028	0.1592	0.1058	0.0534	-0.0009
Brag2	0.0236	0.0073	0.0163	0.0020	0.0497	0.0112	0.0385	0.0014	0.1598	0.1038	0.0560	-0.0014
Brag3	0.0236	0.0074	0.0162	0.0021	0.0500	0.0113	0.0387	0.0007	0.1606	0.1042	0.0563	-0.0047
Boost	0.0218	0.0007	0.0211	-0.0005	0.0482	0.0024	0.0458	0.0031	0.1115	0.0483	0.0632	0.0026
	$\sigma = 0.6$											
Standard	0.1220	0.0240	0.0979	0.0025	0.2375	0.0356	0.2019	0.0108	0.4467	0.1845	0.2622	-0.0079
Bag1	0.1225	0.0206	0.1019	0.0022	0.2523	0.0246	0.2276	0.0109	0.4266	0.1528	0.2737	-0.0083
Bag2	0.1226	0.0212	0.1013	0.0051	0.2405	0.0255	0.2150	0.0093	0.4167	0.1590	0.2577	-0.0107
Bag3	0.1230	0.0214	0.1016	0.0053	0.2392	0.0270	0.2122	0.0055	0.4130	0.1612	0.2518	-0.0188
Brag1	0.1225	0.0204	0.1021	0.0023	0.2504	0.0255	0.2249	0.0110	0.4262	0.1548	0.2713	-0.0074
Brag2	0.1258	0.0208	0.1050	0.0063	0.2519	0.0256	0.2263	0.0074	0.4240	0.1544	0.2697	-0.0071
Brag3	0.1257	0.0213	0.1044	0.0067	0.2510	0.0262	0.2248	0.0060	0.4217	0.1571	0.2647	-0.0118
Boost	0.1383	0.0040	0.1343	0.0000	0.2870	0.0096	0.2774	0.0106	0.4521	0.0858	0.3663	-0.0017
	$\sigma = 1.0$											
Standard	0.2756	0.0387	0.2370	0.0043	0.5461	0.0507	0.4954	0.0182	0.8723	0.2392	0.6331	-0.0131
Bag1	0.2829	0.0306	0.2523	0.0041	0.6111	0.0331	0.5780	0.0185	0.8581	0.1947	0.6634	-0.0152
Bag2	0.2832	0.0315	0.2516	0.0085	0.5774	0.0333	0.5440	0.0166	0.8282	0.2002	0.6280	-0.0151
Bag3	0.2848	0.0315	0.2533	0.0088	0.5732	0.0347	0.5385	0.0120	0.8150	0.2008	0.6142	-0.0255
Brag1	0.2831	0.0302	0.2529	0.0039	0.6084	0.0336	0.5748	0.0187	0.8628	0.1931	0.6697	-0.0128
Brag2	0.2915	0.0308	0.2607	0.0099	0.6102	0.0337	0.5765	0.0134	0.8555	0.1936	0.6619	-0.0105
Brag3	0.2905	0.0318	0.2588	0.0101	0.6016	0.0354	0.5662	0.0101	0.8362	0.2001	0.6361	-0.0155
Boost	0.3332	0.0088	0.3244	0.0008	0.7019	0.0170	0.6849	0.0179	1.0160	0.1246	0.8914	-0.0059

Table C.5: Local linear estimator with plug-in bandwidth selection, Model 2, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0233	0.0076	0.0156	0.0003	0.0516	0.0122	0.0395	0.0026	0.1864	0.1261	0.0603	-0.0001
Bag1	0.0230	0.0070	0.0160	0.0002	0.0541	0.0094	0.0447	0.0012	0.1680	0.1031	0.0649	0.0001
Bag2	0.0234	0.0069	0.0165	0.0013	0.0619	0.0120	0.0500	-0.0086	0.1665	0.1004	0.0661	-0.0078
Bag3	0.0235	0.0070	0.0165	0.0011	0.0619	0.0134	0.0486	-0.0131	0.1654	0.1033	0.0621	-0.0135
Brag1	0.0230	0.0070	0.0160	0.0002	0.0540	0.0099	0.0441	0.0012	0.1709	0.1068	0.0641	0.0003
Brag2	0.0237	0.0069	0.0168	0.0018	0.0627	0.0108	0.0520	-0.0003	0.1708	0.1027	0.0681	-0.0002
Brag3	0.0237	0.0069	0.0168	0.0020	0.0610	0.0107	0.0504	-0.0011	0.1688	0.1025	0.0663	-0.0036
Boost	0.0219	0.0007	0.0212	-0.0007	0.0590	0.0024	0.0566	0.0022	0.1384	0.0558	0.0826	0.0063
	$\sigma = 0.6$											
Standard	0.1237	0.0244	0.0993	0.0032	0.2540	0.0354	0.2186	0.0125	0.5017	0.1937	0.3079	-0.0052
Bag1	0.1243	0.0207	0.1036	0.0030	0.2732	0.0238	0.2494	0.0110	0.4841	0.1594	0.3246	-0.0051
Bag2	0.1247	0.0207	0.1039	0.0056	0.2687	0.0229	0.2458	0.0100	0.4575	0.1517	0.3058	-0.0126
Bag3	0.1250	0.0209	0.1040	0.0055	0.2627	0.0247	0.2380	0.0029	0.4451	0.1551	0.2900	-0.0223
Brag1	0.1242	0.0206	0.1036	0.0029	0.2702	0.0247	0.2455	0.0116	0.4830	0.1616	0.3214	-0.0038
Brag2	0.1268	0.0205	0.1062	0.0066	0.2660	0.0242	0.2418	0.0089	0.4589	0.1567	0.3022	-0.0030
Brag3	0.1266	0.0211	0.1055	0.0069	0.2608	0.0251	0.2357	0.0058	0.4471	0.1588	0.2883	-0.0092
Boost	0.1411	0.0045	0.1366	0.0006	0.3233	0.0104	0.3129	0.0119	0.5754	0.1015	0.4739	0.0032
	$\sigma = 1.0$											
Standard	0.2783	0.0400	0.2383	0.0059	0.5733	0.0518	0.5215	0.0225	1.0192	0.2528	0.7664	-0.0076
Bag1	0.2860	0.0313	0.2547	0.0053	0.6619	0.0327	0.6292	0.0204	1.0011	0.2050	0.7961	-0.0086
Bag2	0.2866	0.0315	0.2551	0.0098	0.6484	0.0308	0.6175	0.0194	0.9510	0.1963	0.7547	-0.0150
Bag3	0.2890	0.0316	0.2574	0.0100	0.6294	0.0324	0.5970	0.0107	0.9069	0.1980	0.7089	-0.0272
Brag1	0.2864	0.0309	0.2555	0.0055	0.6539	0.0332	0.6207	0.0205	0.9999	0.2034	0.7964	-0.0063
Brag2	0.2943	0.0309	0.2633	0.0108	0.6386	0.0327	0.6059	0.0159	0.9421	0.1993	0.7428	-0.0031
Brag3	0.2926	0.0321	0.2605	0.0109	0.6171	0.0345	0.5826	0.0119	0.8943	0.2053	0.6889	-0.0115
Boost	0.3329	0.0100	0.3229	0.0022	0.7519	0.0191	0.7328	0.0220	1.3904	0.1463	1.2441	0.0049

Table C.6: Local linear estimator with plug-in bandwidth selection, Model 2, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0345	0.0150	0.0194	0.0017	0.0775	0.0193	0.0581	-0.0041	0.2147	0.1448	0.0699	0.0126
Bag1	0.0338	0.0115	0.0223	0.0024	0.0925	0.0200	0.0725	-0.0057	0.1884	0.1149	0.0735	0.0107
Bag2	0.0346	0.0127	0.0219	0.0046	0.0986	0.0417	0.0568	-0.0030	0.1923	0.1291	0.0632	0.0127
Bag3	0.0342	0.0130	0.0212	0.0042	0.0959	0.0430	0.0529	-0.0055	0.1900	0.1306	0.0594	0.0109
Brag1	0.0335	0.0117	0.0218	0.0021	0.0900	0.0196	0.0704	-0.0048	0.1936	0.1208	0.0729	0.0122
Brag2	0.0353	0.0121	0.0231	0.0033	0.1074	0.0340	0.0735	-0.0053	0.1976	0.1248	0.0729	0.0147
Brag3	0.0351	0.0120	0.0231	0.0034	0.1081	0.0368	0.0713	-0.0030	0.1928	0.1216	0.0712	0.0101
Boost	0.0284	0.0019	0.0266	0.0005	0.0802	0.0053	0.0749	-0.0021	0.1431	0.0586	0.0845	0.0123
	$\sigma = 0.6$											
Standard	0.1498	0.0363	0.1135	0.0039	0.3187	0.0349	0.2838	-0.0119	0.5507	0.1951	0.3556	0.0181
Bag1	0.1553	0.0290	0.1263	0.0033	0.3553	0.0238	0.3315	-0.0124	0.5287	0.1576	0.3710	0.0200
Bag2	0.1509	0.0294	0.1216	0.0071	0.3033	0.0352	0.2680	-0.0151	0.4719	0.1662	0.3057	0.0182
Bag3	0.1503	0.0298	0.1205	0.0072	0.2856	0.0375	0.2481	-0.0136	0.4566	0.1674	0.2892	0.0072
Brag1	0.1543	0.0294	0.1249	0.0034	0.3512	0.0247	0.3265	-0.0123	0.5317	0.1618	0.3699	0.0205
Brag2	0.1549	0.0297	0.1253	0.0051	0.3291	0.0315	0.2976	-0.0103	0.5005	0.1656	0.3349	0.0233
Brag3	0.1544	0.0297	0.1247	0.0053	0.3134	0.0351	0.2783	-0.0100	0.4823	0.1637	0.3186	0.0163
Boost	0.1679	0.0081	0.1598	0.0025	0.4110	0.0086	0.4025	-0.0113	0.6002	0.0895	0.5107	0.0203
	$\sigma = 1.0$											
Standard	0.3216	0.0618	0.2598	0.0068	0.7178	0.0532	0.6645	-0.0189	1.1172	0.2485	0.8687	0.0211
Bag1	0.3381	0.0492	0.2889	0.0063	0.8394	0.0325	0.8070	-0.0248	1.1329	0.2008	0.9321	0.0291
Bag2	0.3339	0.0492	0.2847	0.0107	0.7027	0.0406	0.6621	-0.0275	0.9702	0.2068	0.7634	0.0195
Bag3	0.3329	0.0494	0.2834	0.0109	0.6522	0.0431	0.6091	-0.0233	0.9270	0.2064	0.7206	0.0008
Brag1	0.3366	0.0494	0.2872	0.0065	0.8291	0.0335	0.7956	-0.0222	1.1360	0.2018	0.9341	0.0293
Brag2	0.3385	0.0499	0.2886	0.0080	0.7586	0.0379	0.7208	-0.0176	1.0450	0.2051	0.8399	0.0292
Brag3	0.3373	0.0504	0.2869	0.0082	0.6950	0.0428	0.6522	-0.0182	0.9831	0.2060	0.7771	0.0217
Boost	0.3815	0.0185	0.3630	0.0050	0.9605	0.0169	0.9436	-0.0211	1.4188	0.1256	1.2932	0.0291

Table C.7: Local linear estimator with plug-in bandwidth selection, Model 2, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.0453	0.0155	0.0298	0.0025	0.1197	0.0240	0.0957	-0.0061	0.2578	0.1558	0.1020	0.0061
Bag1	0.0549	0.0119	0.0430	0.0016	0.1355	0.0263	0.1092	-0.0040	0.2302	0.1233	0.1069	0.0063
Bag2	0.0510	0.0155	0.0356	0.0053	0.1373	0.0596	0.0777	-0.0027	0.2209	0.1415	0.0794	0.0067
Bag3	0.0485	0.0158	0.0327	0.0047	0.1300	0.0604	0.0696	-0.0073	0.2122	0.1420	0.0702	0.0036
Brag1	0.0547	0.0122	0.0425	0.0023	0.1361	0.0253	0.1108	-0.0033	0.2370	0.1295	0.1075	0.0070
Brag2	0.0510	0.0143	0.0367	0.0061	0.1527	0.0567	0.0960	-0.0019	0.2411	0.1458	0.0953	0.0070
Brag3	0.0489	0.0145	0.0345	0.0068	0.1483	0.0593	0.0891	0.0005	0.2312	0.1427	0.0885	0.0029
Boost	0.0565	0.0022	0.0542	0.0021	0.1579	0.0095	0.1484	-0.0049	0.2353	0.0706	0.1647	0.0086
	$\sigma = 0.6$											
Standard	0.2001	0.0381	0.1620	0.0008	0.5298	0.0382	0.4916	-0.0203	0.7840	0.2101	0.5740	-0.0062
Bag1	0.2359	0.0306	0.2054	0.0016	0.6214	0.0302	0.5912	-0.0141	0.7766	0.1700	0.6066	-0.0008
Bag2	0.2008	0.0308	0.1700	0.0067	0.4407	0.0512	0.3894	-0.0159	0.6026	0.1801	0.4225	0.0058
Bag3	0.1864	0.0313	0.1550	0.0057	0.3762	0.0518	0.3244	-0.0148	0.5415	0.1803	0.3613	-0.0083
Brag1	0.2122	0.0307	0.1815	0.0009	0.6147	0.0304	0.5843	-0.0162	0.7939	0.1742	0.6198	-0.0023
Brag2	0.1939	0.0313	0.1626	0.0046	0.4410	0.0494	0.3916	-0.0092	0.6114	0.1862	0.4252	-0.0005
Brag3	0.1692	0.0320	0.1372	0.0088	0.3603	0.0526	0.3077	-0.0052	0.5452	0.1851	0.3600	-0.0056
Boost	0.3419	0.0099	0.3321	-0.0040	0.9732	0.0128	0.9604	-0.0254	1.1747	0.1076	1.0671	-0.0080
	$\sigma = 1.0$											
Standard	0.3551	0.0659	0.2892	0.0046	1.1320	0.0582	1.0738	-0.0329	1.6783	0.2684	1.4100	-0.0085
Bag1	0.4415	0.0523	0.3892	0.0051	1.3773	0.0389	1.3384	-0.0403	1.7252	0.2183	1.5069	-0.0036
Bag2	0.4155	0.0513	0.3642	0.0109	0.9894	0.0546	0.9348	-0.0380	1.2846	0.2213	1.0633	0.0036
Bag3	0.3896	0.0517	0.3379	0.0102	0.8226	0.0559	0.7667	-0.0300	1.0987	0.2205	0.8783	-0.0197
Brag1	0.4219	0.0521	0.3698	0.0030	1.3428	0.0393	1.3036	-0.0381	1.6879	0.2194	1.4685	0.0008
Brag2	0.4090	0.0528	0.3563	0.0091	0.9090	0.0497	0.8593	-0.0196	1.2480	0.2283	1.0197	-0.0031
Brag3	0.3528	0.0540	0.2988	0.0115	0.7375	0.0559	0.6816	-0.0121	1.0621	0.2309	0.8312	-0.0051
Boost	0.4586	0.0219	0.4367	0.0016	2.0600	0.0232	2.0368	-0.0387	3.1129	0.1522	2.9607	-0.0148

Table C.8: Local linear estimator with plug-in bandwidth selection, Model 2, $X \sim N(0, 1)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0048	0.0012	0.0036	-0.0006	0.0090	0.0017	0.0073	0.0009	0.0280	0.0177	0.0102	-0.0144
Bag1	0.0048	0.0011	0.0037	-0.0006	0.0091	0.0013	0.0078	0.0009	0.0251	0.0146	0.0105	-0.0109
Bag2	0.0047	0.0011	0.0036	-0.0002	0.0086	0.0014	0.0072	-0.0014	0.0253	0.0149	0.0103	-0.0146
Bag3	0.0047	0.0011	0.0036	-0.0003	0.0087	0.0015	0.0072	-0.0046	0.0250	0.0151	0.0099	-0.0150
Brag1	0.0048	0.0011	0.0037	-0.0006	0.0091	0.0013	0.0077	0.0010	0.0255	0.0149	0.0105	-0.0111
Brag2	0.0048	0.0011	0.0038	0.0001	0.0090	0.0013	0.0076	-0.0002	0.0251	0.0145	0.0106	-0.0110
Brag3	0.0048	0.0011	0.0038	0.0001	0.0089	0.0013	0.0075	-0.0012	0.0249	0.0146	0.0104	-0.0119
Boost	0.0051	0.0001	0.0050	-0.0006	0.0103	0.0003	0.0100	0.0012	0.0209	0.0079	0.0129	-0.0024
	$\sigma = 0.3$											
Standard	0.0267	0.0040	0.0227	-0.0008	0.0509	0.0045	0.0464	0.0028	0.0851	0.0294	0.0557	-0.0287
Bag1	0.0273	0.0033	0.0239	-0.0005	0.0562	0.0029	0.0533	0.0042	0.0840	0.0254	0.0587	-0.0222
Bag2	0.0272	0.0034	0.0238	0.0006	0.0528	0.0030	0.0498	0.0032	0.0819	0.0255	0.0564	-0.0271
Bag3	0.0272	0.0033	0.0239	0.0005	0.0523	0.0031	0.0492	0.0009	0.0795	0.0246	0.0549	-0.0222
Brag1	0.0273	0.0033	0.0240	-0.0005	0.0559	0.0030	0.0530	0.0042	0.0841	0.0243	0.0598	-0.0205
Brag2	0.0280	0.0033	0.0247	0.0013	0.0557	0.0029	0.0528	0.0024	0.0840	0.0240	0.0600	-0.0190
Brag3	0.0279	0.0034	0.0245	0.0011	0.0547	0.0030	0.0517	0.0009	0.0821	0.0248	0.0573	-0.0209
Boost	0.0320	0.0009	0.0311	-0.0001	0.0654	0.0012	0.0643	0.0043	0.0927	0.0161	0.0766	-0.0041
	$\sigma = 0.5$											
Standard	0.0620	0.0062	0.0558	-0.0009	0.1234	0.0057	0.1177	0.0061	0.1738	0.0373	0.1364	-0.0401
Bag1	0.0651	0.0048	0.0604	-0.0003	0.1429	0.0036	0.1393	0.0079	0.1784	0.0322	0.1462	-0.0307
Bag2	0.0651	0.0048	0.0604	0.0017	0.1342	0.0036	0.1306	0.0066	0.1724	0.0321	0.1403	-0.0355
Bag3	0.0655	0.0047	0.0608	0.0013	0.1331	0.0036	0.1294	0.0037	0.1683	0.0303	0.1380	-0.0290
Brag1	0.0654	0.0046	0.0608	-0.0002	0.1425	0.0036	0.1389	0.0080	0.1802	0.0303	0.1498	-0.0277
Brag2	0.0674	0.0046	0.0627	0.0027	0.1426	0.0035	0.1392	0.0051	0.1810	0.0299	0.1511	-0.0249
Brag3	0.0668	0.0048	0.0620	0.0020	0.1390	0.0036	0.1354	0.0025	0.1741	0.0312	0.1429	-0.0274
Boost	0.0785	0.0018	0.0767	0.0004	0.1653	0.0018	0.1634	0.0084	0.2080	0.0222	0.1858	-0.0103

Table C.9: Local linear estimator with plug-in bandwidth selection, Model 3, $X \sim U(-2, 2)$, $N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0048	0.0011	0.0037	-0.0004	0.0099	0.0016	0.0083	0.0008	0.0308	0.0185	0.0123	-0.0102
Bag1	0.0048	0.0010	0.0037	-0.0005	0.0105	0.0012	0.0093	0.0006	0.0276	0.0150	0.0126	-0.0069
Bag2	0.0048	0.0010	0.0038	-0.0002	0.0102	0.0012	0.0090	-0.0067	0.0257	0.0136	0.0121	-0.0115
Bag3	0.0048	0.0010	0.0038	-0.0001	0.0100	0.0013	0.0086	-0.0104	0.0253	0.0140	0.0113	-0.0138
Brag1	0.0048	0.0010	0.0037	-0.0005	0.0103	0.0012	0.0091	0.0007	0.0279	0.0153	0.0125	-0.0070
Brag2	0.0048	0.0010	0.0038	0.0001	0.0095	0.0011	0.0084	-0.0019	0.0259	0.0143	0.0116	-0.0070
Brag3	0.0048	0.0010	0.0038	0.0001	0.0094	0.0012	0.0083	-0.0035	0.0255	0.0143	0.0112	-0.0087
Boost	0.0052	0.0001	0.0051	-0.0005	0.0127	0.0003	0.0124	0.0005	0.0278	0.0092	0.0186	-0.0002
	$\sigma = 0.3$											
Standard	0.0270	0.0041	0.0229	0.0002	0.0545	0.0045	0.0499	0.0039	0.0977	0.0308	0.0669	-0.0221
Bag1	0.0277	0.0034	0.0244	0.0002	0.0612	0.0029	0.0584	0.0050	0.0992	0.0267	0.0725	-0.0145
Bag2	0.0276	0.0033	0.0243	0.0014	0.0593	0.0026	0.0566	0.0031	0.0927	0.0250	0.0677	-0.0172
Bag3	0.0277	0.0033	0.0244	0.0011	0.0578	0.0027	0.0550	-0.0004	0.0863	0.0240	0.0623	-0.0146
Brag1	0.0277	0.0033	0.0244	0.0003	0.0607	0.0029	0.0577	0.0049	0.0979	0.0255	0.0724	-0.0130
Brag2	0.0282	0.0033	0.0249	0.0019	0.0583	0.0027	0.0555	0.0029	0.0907	0.0245	0.0662	-0.0095
Brag3	0.0280	0.0034	0.0247	0.0017	0.0559	0.0028	0.0531	0.0005	0.0840	0.0252	0.0588	-0.0132
Boost	0.0322	0.0010	0.0312	0.0003	0.0733	0.0013	0.0720	0.0048	0.1300	0.0187	0.1113	0.0025
	$\sigma = 0.5$											
Standard	0.0627	0.0064	0.0563	0.0001	0.1302	0.0058	0.1244	0.0077	0.2007	0.0389	0.1618	-0.0329
Bag1	0.0660	0.0049	0.0611	0.0010	0.1543	0.0036	0.1507	0.0095	0.2113	0.0338	0.1775	-0.0218
Bag2	0.0658	0.0048	0.0610	0.0027	0.1507	0.0033	0.1474	0.0076	0.1999	0.0319	0.1680	-0.0253
Bag3	0.0663	0.0047	0.0617	0.0024	0.1466	0.0033	0.1433	0.0033	0.1849	0.0298	0.1551	-0.0202
Brag1	0.0662	0.0047	0.0614	0.0008	0.1533	0.0036	0.1497	0.0095	0.2145	0.0318	0.1827	-0.0183
Brag2	0.0678	0.0047	0.0631	0.0033	0.1486	0.0033	0.1452	0.0062	0.1991	0.0306	0.1684	-0.0120
Brag3	0.0670	0.0049	0.0622	0.0030	0.1419	0.0035	0.1384	0.0030	0.1778	0.0318	0.1460	-0.0191
Boost	0.0793	0.0020	0.0772	0.0006	0.1781	0.0020	0.1761	0.0091	0.2944	0.0252	0.2692	0.0003

Table C.10: Local linear estimator with plug-in bandwidth selection, Model 3, $X \sim U(-2, 2)$, Triangular kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0066	0.0028	0.0038	0.0001	0.0116	0.0027	0.0089	-0.0029	0.0382	0.0222	0.0160	0.0001
Bag1	0.0062	0.0020	0.0042	0.0000	0.0115	0.0014	0.0101	-0.0038	0.0306	0.0157	0.0149	0.0018
Bag2	0.0060	0.0020	0.0040	0.0005	0.0096	0.0015	0.0081	-0.0109	0.0295	0.0159	0.0137	-0.0052
Bag3	0.0060	0.0020	0.0039	0.0003	0.0096	0.0019	0.0077	-0.0177	0.0275	0.0154	0.0121	-0.0051
Brag1	0.0062	0.0020	0.0042	0.0000	0.0114	0.0015	0.0099	-0.0035	0.0305	0.0157	0.0148	0.0026
Brag2	0.0062	0.0020	0.0041	0.0003	0.0104	0.0015	0.0089	-0.0065	0.0290	0.0151	0.0139	0.0006
Brag3	0.0061	0.0020	0.0041	0.0004	0.0098	0.0015	0.0083	-0.0097	0.0283	0.0151	0.0132	-0.0002
Boost	0.0056	0.0004	0.0052	0.0007	0.0126	0.0005	0.0122	-0.0006	0.0309	0.0112	0.0197	0.0061
	$\sigma = 0.3$											
Standard	0.0300	0.0073	0.0228	0.0010	0.0638	0.0062	0.0576	-0.0070	0.1127	0.0318	0.0809	0.0022
Bag1	0.0309	0.0053	0.0257	0.0011	0.0734	0.0031	0.0702	-0.0086	0.1078	0.0246	0.0831	0.0028
Bag2	0.0305	0.0053	0.0252	0.0023	0.0611	0.0032	0.0579	-0.0115	0.0963	0.0252	0.0711	-0.0003
Bag3	0.0302	0.0052	0.0250	0.0024	0.0564	0.0034	0.0530	-0.0156	0.0886	0.0233	0.0653	0.0082
Brag1	0.0310	0.0053	0.0257	0.0011	0.0733	0.0032	0.0701	-0.0084	0.1076	0.0238	0.0838	0.0043
Brag2	0.0309	0.0053	0.0256	0.0017	0.0657	0.0032	0.0625	-0.0079	0.0992	0.0235	0.0757	0.0076
Brag3	0.0304	0.0053	0.0251	0.0022	0.0592	0.0034	0.0558	-0.0118	0.0937	0.0237	0.0700	0.0104
Boost	0.0333	0.0019	0.0314	0.0024	0.0838	0.0018	0.0820	-0.0038	0.1338	0.0190	0.1149	0.0161
	$\sigma = 0.5$											
Standard	0.0643	0.0106	0.0537	0.0016	0.1547	0.0081	0.1466	-0.0114	0.2357	0.0362	0.1994	0.0023
Bag1	0.0688	0.0074	0.0614	0.0023	0.1855	0.0039	0.1816	-0.0140	0.2403	0.0283	0.2120	0.0040
Bag2	0.0686	0.0073	0.0613	0.0039	0.1559	0.0040	0.1519	-0.0168	0.2082	0.0293	0.1789	0.0024
Bag3	0.0683	0.0071	0.0612	0.0044	0.1431	0.0040	0.1391	-0.0198	0.1913	0.0265	0.1648	0.0115
Brag1	0.0691	0.0073	0.0618	0.0022	0.1863	0.0039	0.1823	-0.0135	0.2393	0.0271	0.2122	0.0056
Brag2	0.0691	0.0073	0.0618	0.0029	0.1671	0.0039	0.1633	-0.0121	0.2191	0.0270	0.1921	0.0114
Brag3	0.0680	0.0074	0.0607	0.0039	0.1480	0.0042	0.1438	-0.0165	0.2017	0.0274	0.1743	0.0157
Boost	0.0782	0.0036	0.0745	0.0040	0.2163	0.0030	0.2133	-0.0079	0.3199	0.0233	0.2966	0.0191

Table C.11: Local linear estimator with plug-in bandwidth selection, Model 3, $X \sim N(0,1)$, $N(0,1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.0084	0.0029	0.0055	-0.0007	0.0188	0.0026	0.0162	-0.0062	0.0458	0.0231	0.0227	-0.0052
Bag1	0.0094	0.0020	0.0073	-0.0006	0.0189	0.0013	0.0176	-0.0052	0.0371	0.0159	0.0211	-0.0032
Bag2	0.0077	0.0019	0.0058	-0.0011	0.0133	0.0014	0.0119	-0.0211	0.0313	0.0146	0.0166	-0.0154
Bag3	0.0072	0.0020	0.0052	-0.0016	0.0122	0.0019	0.0103	-0.0286	0.0284	0.0144	0.0140	-0.0180
Brag1	0.0095	0.0021	0.0074	-0.0006	0.0189	0.0014	0.0175	-0.0047	0.0378	0.0161	0.0217	-0.0032
Brag2	0.0070	0.0020	0.0050	-0.0009	0.0137	0.0014	0.0124	-0.0149	0.0320	0.0149	0.0171	-0.0114
Brag3	0.0064	0.0020	0.0044	-0.0003	0.0111	0.0015	0.0096	-0.0173	0.0294	0.0149	0.0144	-0.0120
Boost	0.0114	0.0005	0.0110	-0.0004	0.0311	0.0006	0.0305	-0.0047	0.0505	0.0124	0.0381	0.0012
	$\sigma = 0.3$											
Standard	0.0329	0.0079	0.0250	0.0012	0.0914	0.0066	0.0848	-0.0107	0.1503	0.0329	0.1174	-0.0063
Bag1	0.0388	0.0057	0.0332	0.0005	0.1212	0.0033	0.1179	-0.0137	0.1528	0.0258	0.1270	-0.0037
Bag2	0.0372	0.0055	0.0317	0.0018	0.0856	0.0032	0.0825	-0.0210	0.1167	0.0246	0.0921	-0.0076
Bag3	0.0348	0.0054	0.0293	0.0013	0.0696	0.0035	0.0661	-0.0251	0.0979	0.0225	0.0754	0.0024
Brag1	0.0439	0.0057	0.0382	-0.0001	0.1163	0.0033	0.1129	-0.0126	0.1479	0.0252	0.1228	-0.0005
Brag2	0.0371	0.0056	0.0315	0.0017	0.0742	0.0032	0.0711	-0.0108	0.1135	0.0241	0.0894	0.0004
Brag3	0.0315	0.0056	0.0260	0.0022	0.0608	0.0035	0.0573	-0.0161	0.0947	0.0241	0.0705	0.0037
Boost	0.0398	0.0024	0.0373	0.0035	0.1568	0.0023	0.1545	-0.0078	0.2457	0.0205	0.2251	0.0075
	$\sigma = 0.5$											
Standard	0.0661	0.0116	0.0545	0.0012	0.2182	0.0089	0.2093	-0.0154	0.3347	0.0376	0.2971	-0.0088
Bag1	0.0792	0.0081	0.0710	0.0019	0.3069	0.0046	0.3023	-0.0251	0.3443	0.0296	0.3147	-0.0034
Bag2	0.0825	0.0078	0.0747	0.0032	0.2189	0.0042	0.2147	-0.0296	0.2601	0.0292	0.2309	-0.0058
Bag3	0.0762	0.0075	0.0687	0.0033	0.1749	0.0043	0.1706	-0.0306	0.2174	0.0262	0.1912	0.0073
Brag1	0.0774	0.0080	0.0694	0.0025	0.3002	0.0046	0.2956	-0.0252	0.3492	0.0287	0.3205	0.0000
Brag2	0.0748	0.0079	0.0670	0.0031	0.1862	0.0040	0.1822	-0.0135	0.2493	0.0279	0.2214	0.0040
Brag3	0.0694	0.0079	0.0615	0.0039	0.1488	0.0044	0.1444	-0.0203	0.2072	0.0282	0.1790	0.0101
Boost	0.0796	0.0045	0.0751	0.0044	0.3683	0.0040	0.3643	-0.0119	0.6042	0.0255	0.5786	0.0120

Table C.12: Local linear estimator with plug-in bandwidth selection, Model 3, $X \sim N(0, 1)$, Triangular kernel

Appendix D

Results of Simulation Study D

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.02051	0.00558	0.01493	-0.01220	0.04696	0.01070	0.03626	-0.00492	0.09468	0.00885	0.08582	-0.02470
Bag1	0.01913	0.00377	0.01536	-0.01375	0.04599	0.00883	0.03716	-0.00498	0.10340	0.01190	0.09149	-0.02023
Bag2	0.02648	0.00862	0.01786	-0.01115	0.06349	0.02188	0.04161	0.01076	0.13635	0.03199	0.10436	0.00911
Bag3	0.02558	0.00808	0.01749	-0.01168	0.06090	0.02069	0.04020	0.00961	0.13464	0.03357	0.10107	0.01509
Brag1	0.01862	0.00334	0.01528	-0.01375	0.04656	0.00916	0.03740	-0.00460	0.10808	0.01135	0.09673	-0.02056
Brag2	0.02607	0.00783	0.01824	-0.00740	0.06542	0.02060	0.04483	0.01335	0.14314	0.02314	0.12000	-0.01335
Brag3	0.02613	0.00796	0.01817	-0.00731	0.06394	0.01946	0.04448	0.01213	0.13366	0.02407	0.10959	-0.00858
Boost	0.01542	0.00105	0.01437	-0.01253	0.04147	0.00419	0.03728	-0.00674	0.09645	0.00646	0.08998	-0.03198
	$\sigma = 0.6$											
Standard	0.12714	0.04214	0.08499	-0.04242	0.23169	0.06407	0.16762	-0.01094	0.40095	0.09022	0.31072	-0.03951
Bag1	0.13005	0.05597	0.07409	-0.04484	0.30826	0.15042	0.15784	-0.01092	0.42349	0.12363	0.29986	-0.03250
Bag2	0.16577	0.08963	0.07614	-0.04271	0.36848	0.20838	0.16010	0.02046	0.50821	0.17569	0.33253	-0.00227
Bag3	0.15791	0.08274	0.07517	-0.04367	0.32334	0.16581	0.15752	0.01669	0.45032	0.12245	0.32787	-0.02696
Brag1	0.13314	0.05687	0.07627	-0.04525	0.30808	0.13117	0.17691	-0.00954	0.42417	0.12312	0.30105	-0.02131
Brag2	0.17599	0.09323	0.08276	-0.03105	0.38516	0.19217	0.19299	0.02676	0.52597	0.16975	0.35621	-0.02067
Brag3	0.17569	0.08934	0.08635	-0.04536	0.35804	0.16823	0.18982	0.00574	0.51785	0.13816	0.37969	-0.02510
Boost	0.10075	0.01812	0.08263	-0.04125	0.22593	0.03843	0.18749	-0.01327	0.39238	0.07330	0.31908	-0.06360
	$\sigma = 1.0$											
Standard	0.30312	0.11847	0.18465	-0.07490	0.56656	0.15501	0.41155	-0.02268	0.95762	0.19315	0.76447	-0.08288
Bag1	0.35422	0.21241	0.14181	-0.08508	0.66262	0.37304	0.28958	-0.01608	0.84549	0.36237	0.48312	-0.08560
Bag2	0.41556	0.27386	0.14170	-0.08342	0.74755	0.45303	0.29452	0.02139	0.99945	0.45523	0.54423	-0.04619
Bag3	0.36763	0.22644	0.14119	-0.08222	0.69001	0.39317	0.29684	0.02507	0.88261	0.30755	0.57506	-0.06617
Brag1	0.32997	0.17459	0.15538	-0.08667	0.67340	0.35654	0.31685	-0.01356	0.85287	0.32217	0.53070	-0.07439
Brag2	0.39261	0.23316	0.15944	-0.07873	0.79932	0.43622	0.36310	0.00999	1.03105	0.39284	0.63821	-0.08107
Brag3	0.39032	0.23444	0.15588	-0.07155	0.78148	0.41914	0.36234	0.00246	1.01184	0.36596	0.64588	-0.08396
Boost	0.25293	0.06552	0.18741	-0.07003	0.57860	0.14624	0.43236	-0.02284	0.83529	0.18103	0.65425	-0.09889

Table D.1: BRNP estimator with cross-validation bandwidth selection, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.2$											
Standard	0.03024	0.00504	0.02519	-0.01502	0.05901	0.01080	0.04821	-0.01423	0.12296	0.00927	0.11369	-0.01109
Bag1	0.01939	0.00339	0.01600	-0.01541	0.04763	0.00980	0.03783	-0.00970	0.10669	0.01151	0.09518	-0.01985
Bag2	0.02748	0.00815	0.01933	-0.01665	0.06838	0.02433	0.04405	0.00773	0.16281	0.03938	0.12344	-0.01576
Bag3	0.02659	0.00776	0.01883	-0.01718	0.06488	0.02306	0.04182	0.00686	0.1661	0.0711	0.0950	-0.0265
Brag1	0.01917	0.00339	0.01578	-0.01542	0.04889	0.00947	0.03943	-0.00922	0.11486	0.01285	0.10201	-0.02486
Brag2	0.02739	0.00806	0.01933	-0.01288	0.06733	0.02160	0.04573	0.00560	0.16940	0.03296	0.13644	-0.03996
Brag3	0.02680	0.00747	0.01933	-0.01508	0.06589	0.02084	0.04505	0.00260	0.1543	0.0480	0.1063	-0.0147
Boost	0.01575	0.00102	0.01473	-0.01381	0.04074	0.00468	0.03606	-0.01117	0.09538	0.00681	0.08857	-0.03312
	$\sigma = 0.6$											
Standard	0.17284	0.03783	0.13500	-0.04458	0.35562	0.09380	0.26181	-0.00353	0.62119	0.16627	0.45492	-0.04652
Bag1	0.13310	0.05618	0.07693	-0.05005	0.25016	0.11768	0.13247	-0.02800	0.44157	0.09656	0.34501	-0.06007
Bag2	0.17984	0.09456	0.08529	-0.05049	0.32759	0.18162	0.14597	0.01378	0.57471	0.15692	0.41779	-0.04751
Bag3	0.17522	0.08926	0.08596	-0.05186	0.30905	0.16258	0.14647	0.01662	0.4509	0.1757	0.2751	-0.0395
Brag1	0.13473	0.05388	0.08085	-0.04770	0.25093	0.10791	0.14302	-0.03150	0.44382	0.11083	0.33299	-0.05081
Brag2	0.18941	0.09578	0.09364	-0.04325	0.33807	0.16970	0.16838	0.00964	0.59064	0.16935	0.42129	-0.08798
Brag3	0.19406	0.09625	0.09781	-0.05353	0.33914	0.17161	0.16753	0.00701	0.4592	0.1564	0.3029	-0.0199
Boost	0.10389	0.01658	0.08731	-0.04389	0.22229	0.03364	0.18864	-0.02054	0.40097	0.06399	0.33698	-0.07514
	$\sigma = 1.0$											
Standard	0.50932	0.16509	0.34423	-0.06384	0.82912	0.26548	0.56364	-0.06929	1.45675	0.30446	1.15229	-0.09352
Bag1	0.31518	0.17138	0.14380	-0.08322	0.55964	0.30376	0.25589	-0.03870	1.03740	0.34136	0.69604	-0.07103
Bag2	0.40287	0.24540	0.15747	-0.08823	0.68693	0.40961	0.27732	0.01250	1.31621	0.46972	0.84649	-0.10168
Bag3	0.38870	0.22717	0.16153	-0.08677	0.63040	0.35208	0.27833	0.01532	1.19013	0.37079	0.81934	-0.12294
Brag1	0.30996	0.15827	0.15168	-0.08381	0.59224	0.30856	0.28368	-0.04114	1.00396	0.28557	0.71838	-0.06500
Brag2	0.40554	0.23543	0.17010	-0.08229	0.73995	0.41463	0.32531	0.00054	1.26972	0.38682	0.88290	-0.11528
Brag3	0.42809	0.24257	0.18552	-0.08852	0.72589	0.39440	0.33149	-0.01575	1.32077	0.41006	0.91071	-0.11643
Boost	0.24765	0.05763	0.19002	-0.07150	0.52598	0.11051	0.41547	-0.02160	0.88232	0.16913	0.71318	-0.12228

Table D.2: BRNP estimator with cross-validation bandwidth selection, Model 2, $X \sim U(-2, 2), N(0, 1)$ kernel

	$n = 200$				$n = 100$				$n = 50$			
	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$	MISE	$\int \text{Bias}^2\, dx$	$\int \text{Var}\, dx$	$\int \text{Bias}\, dx$
	$\sigma = 0.1$											
Standard	0.00541	0.00146	0.00395	-0.00650	0.01105	0.00286	0.00818	-0.00362	0.02725	0.00995	0.01730	-0.02264
Bag1	0.00387	0.00125	0.00262	-0.00682	0.00837	0.00159	0.00677	-0.00435	0.02111	0.00646	0.01465	-0.02793
Bag2	0.00517	0.00219	0.00297	-0.00682	0.01022	0.00332	0.00690	0.00258	0.02760	0.01061	0.01699	-0.01965
Bag3	0.00506	0.00207	0.00298	-0.00709	0.00984	0.00307	0.00677	0.00275	0.0227	0.0100	0.0128	-0.0277
Brag1	0.00383	0.00125	0.00258	-0.00692	0.00840	0.00143	0.00697	-0.00413	0.02090	0.00562	0.01528	-0.02776
Brag2	0.00543	0.00215	0.00329	-0.00500	0.01006	0.00259	0.00746	0.00032	0.02550	0.00753	0.01797	-0.01915
Brag3	0.00534	0.00199	0.00335	-0.00658	0.00978	0.00263	0.00715	0.00068	0.0220	0.0072	0.0148	-0.0194
Boost	0.00336	0.00034	0.00302	-0.00659	0.00823	0.00066	0.00757	-0.00377	0.01733	0.00328	0.01405	-0.02855
	$\sigma = 0.3$											
Standard	0.03437	0.01189	0.02247	-0.02578	0.05961	0.02314	0.03647	-0.01285	0.10652	0.04935	0.05717	-0.05519
Bag1	0.03875	0.02102	0.01774	-0.01857	0.08485	0.04904	0.03582	0.00170	0.12202	0.09753	0.02449	-0.03094
Bag2	0.04640	0.02646	0.01994	-0.01602	0.09192	0.05850	0.03342	0.01420	0.13207	0.10768	0.02438	-0.02567
Bag3	0.03363	0.01912	0.01451	-0.01987	0.06346	0.03537	0.02809	0.00406	0.10855	0.06897	0.03958	-0.03739
Brag1	0.02693	0.01463	0.01230	-0.02020	0.07174	0.03382	0.03791	-0.00364	0.11126	0.06799	0.04326	-0.03606
Brag2	0.03407	0.01931	0.01476	-0.01653	0.07888	0.03963	0.03925	0.00755	0.12171	0.07330	0.04841	-0.05430
Brag3	0.03405	0.01878	0.01527	-0.00628	0.07697	0.03873	0.03824	0.00140	0.12109	0.07700	0.04409	-0.03670
Boost	0.02256	0.00554	0.01701	-0.02381	0.04909	0.00642	0.04267	-0.00799	0.09530	0.03558	0.05972	-0.06043
	$\sigma = 0.5$											
Standard	0.08600	0.03801	0.04799	-0.02985	0.12194	0.03311	0.08883	-0.02792	0.19972	0.12962	0.07010	-0.14591
Bag1	0.13407	0.11024	0.02383	0.00102	0.10374	0.08208	0.02166	0.00067	0.16117	0.11705	0.04412	-0.06298
Bag2	0.11377	0.09261	0.02117	-0.00003	0.13950	0.11647	0.02303	0.01963	0.16666	0.12161	0.04505	-0.05706
Bag3	0.08071	0.05689	0.02382	-0.01802	0.11653	0.08324	0.03329	0.00985	0.16205	0.09799	0.06406	-0.06465
Brag1	0.08254	0.05331	0.02922	-0.01720	0.12409	0.08507	0.03903	-0.00516	0.15922	0.10458	0.05465	-0.07282
Brag2	0.08896	0.06093	0.02803	-0.00559	0.13044	0.08711	0.04333	0.00283	0.17224	0.10478	0.06746	-0.08718
Brag3	0.08814	0.06071	0.02743	0.00185	0.13271	0.08952	0.04319	-0.00519	0.17062	0.10260	0.06802	-0.06963
Boost	0.06496	0.02330	0.04166	-0.04012	0.13252	0.04537	0.08715	-0.00602	0.17123	0.10544	0.06579	-0.06428

Table D.3: BRNP estimator with cross-validation bandwidth selection, Model 3, $X \sim U(-2, 2), N(0, 1)$ kernel

Appendix E

Comparing SimA to SimB

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9250	0.9558	0.4829	1.0634	1.0810	0.9480	1.0813	1.0588	1.1840
Bag1	0.9209	0.9820	0.5001	1.0366	1.0313	0.9682	1.0389	0.9281	1.0894
Bag2	0.9573	1.0739	0.6408	1.0974	1.1558	1.0384	1.1070	1.0371	1.1321
Bag3	0.9432	1.0514	0.6585	1.0845	1.0887	0.9404	1.0579	0.9520	1.0543
Brag1	0.9257	0.9769	0.4878	1.0360	1.0267	0.9809	1.0287	0.9373	1.1232
Brag2	0.9606	1.0287	0.5806	1.0799	1.0994	1.0175	1.0691	0.9941	1.1376
Brag3	0.9364	1.0082	0.5752	1.0694	1.0701	0.9664	1.0555	0.9575	1.0858
Boost	1.2420	1.1988	0.7711	1.0938	1.0586	1.2588	1.0417	0.9638	1.3571

Table E.1: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9109	1.0694	0.7899	1.0642	1.0936	1.0740	1.0977	1.0591	1.2405
Bag1	0.9036	0.9374	0.7520	1.0174	1.0051	1.0491	1.0406	0.9221	1.0864
Bag2	0.9326	1.0480	0.9609	1.0705	1.1156	1.1410	1.1080	1.0218	1.1302
Bag3	0.9217	1.0533	0.9558	1.0604	1.0466	1.0215	1.0490	0.9301	1.0449
Brag1	0.8950	0.9286	0.5990	1.0311	1.0107	0.9628	1.0393	0.9401	1.1019
Brag2	0.9306	1.0215	0.9417	1.0631	1.0644	1.0864	1.0673	0.9716	1.1109
Brag3	0.9002	0.9868	0.9154	1.0441	1.0277	1.0373	1.0439	0.9319	1.0773
Boost	1.2401	1.4442	1.3824	1.0873	1.0597	1.3638	1.0531	0.9532	1.3215

Table E.2: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2, 2)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.3760	1.2213	0.9550	1.0929	1.0179	1.0076	1.0719	1.0216	1.2016
Bag1	1.2516	0.8368	0.7846	1.0239	0.8342	0.9476	0.9918	0.7610	1.0266
Bag2	1.5787	0.9262	0.9889	1.0959	0.9110	0.9825	1.0315	0.8513	1.0486
Bag3	1.5884	1.0028	1.1414	1.0812	0.8941	1.0238	1.0129	0.8246	1.0765
Brag1	1.2821	0.8474	0.8222	1.0319	0.8481	0.9622	0.9962	0.7775	1.0856
Brag2	1.5729	0.9788	1.0204	1.0860	0.8766	1.0004	1.0237	0.7932	1.0864
Brag3	1.5247	0.9136	1.0205	1.0739	0.8408	1.0146	1.0069	0.7745	1.1038
Boost	1.3298	1.0225	1.3099	1.1215	0.9587	1.2370	1.0732	0.9443	1.3211

Table E.3: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0, 1), N(0, 1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.6154	1.0966	1.4673	1.1701	0.9664	1.2669	1.0767	0.9501	1.2319
Bag1	1.2490	0.5303	1.0992	1.0601	0.7227	1.0429	1.0099	0.6835	1.0000
Bag2	1.4385	0.5770	1.2279	1.1883	0.7775	1.1202	1.0830	0.7593	1.0400
Bag3	1.5873	0.6942	1.3629	1.1704	0.7951	1.1828	1.0435	0.7605	1.1205
Brag1	1.2803	0.5830	1.0450	1.0577	0.7494	1.0466	1.0302	0.7322	1.0969
Brag2	1.7525	0.6661	1.5227	1.1807	0.7562	1.2114	1.0681	0.7123	1.1164
Brag3	1.5382	0.5819	1.4172	1.1258	0.7079	1.1926	1.0444	0.7032	1.1442
Boost	1.9040	1.9050	2.8252	1.1213	0.9684	1.5979	1.0356	0.8915	1.3072

Table E.4: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9180	0.9544	0.5294	1.0693	1.0764	0.9667	1.0886	1.0428	1.2259
Bag1	0.9110	0.9694	0.5590	1.0331	0.9942	0.9212	1.0300	0.9144	1.1672
Bag2	0.9486	1.0520	0.6322	1.0923	1.0918	0.9446	1.0936	1.0169	1.2087
Bag3	0.9330	1.0313	0.6512	1.0870	1.0744	0.9346	1.0775	0.9760	1.1021
Brag1	0.9116	0.9651	0.5488	1.0349	1.0033	0.9513	1.0343	0.9145	1.1739
Brag2	0.9446	1.0105	0.6187	1.0778	1.0661	0.9599	1.0799	0.9757	1.1543
Brag3	0.9276	0.9980	0.6459	1.0738	1.0546	0.9657	1.0758	0.9717	1.1602
Boost	1.2656	1.2273	0.9342	1.1193	1.0706	1.3161	1.0540	0.9560	1.4385

Table E.5: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2), N(0,1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9317	1.0137	0.6781	1.0656	1.1302	1.0579	1.0959	1.0629	1.3752
Bag1	0.9028	0.9768	0.5930	1.0325	0.9926	0.8996	1.0351	0.9095	1.0820
Bag2	0.9539	1.1030	0.7183	1.0828	1.0694	0.9164	1.0923	0.9863	1.0439
Bag3	0.9456	1.0963	0.8408	1.0778	1.0550	0.9874	1.0702	0.9703	1.1233
Brag1	0.9147	0.9693	0.6705	1.0403	1.0025	0.9961	1.0384	0.9095	1.3246
Brag2	0.9566	1.0528	0.7881	1.0777	1.0510	0.9925	1.0751	0.9525	1.2228
Brag3	0.9390	1.0636	0.7079	1.0628	1.0469	0.9601	1.0696	0.9746	1.2187
Boost	1.2464	1.2935	1.1985	1.0946	1.1011	1.3719	1.0393	0.9513	1.5122

Table E.6: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.8848	0.9982	0.5415	1.0752	1.0940	1.0225	1.0951	1.1247	1.2712
Bag1	0.9275	0.9835	0.5757	1.0334	0.9628	1.0026	1.0108	0.8509	1.1536
Bag2	0.9312	1.0264	0.5363	1.0117	0.9701	0.8354	1.0153	0.9113	1.0482
Bag3	0.9325	1.0379	0.5470	1.0183	0.9735	0.8156	1.0234	0.9077	0.9504
Brag1	0.9273	0.9957	0.5688	1.0357	0.9800	1.0479	1.0184	0.8750	1.2669
Brag2	0.9792	1.0667	0.6085	1.0308	0.9458	0.9478	1.0249	0.8577	1.1306
Brag3	0.9673	1.0575	0.6174	1.0184	0.9288	0.8731	1.0147	0.8531	0.9789
Boost	1.3366	1.2008	0.9211	1.2398	1.1364	1.3347	1.1581	1.0973	1.4484

Table E.7: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, $N(0,1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9810	1.0772	0.7146	1.1048	1.1324	1.0773	1.1373	1.1680	1.3560
Bag1	0.9345	0.9676	0.6002	1.0348	0.9499	0.9494	1.0284	0.8758	1.1378
Bag2	1.0033	1.0259	0.6216	1.0050	0.9434	0.8058	1.0093	0.8863	0.9564
Bag3	1.0281	1.0397	0.7414	1.0145	0.9373	0.8696	1.0128	0.8837	0.9560
Brag1	0.9622	0.9753	0.6924	1.0408	0.9505	1.0382	1.0231	0.8922	1.2794
Brag2	1.1063	1.0842	0.7888	1.0288	0.9038	0.9536	1.0089	0.8332	1.0991
Brag3	1.1202	1.2312	0.6699	1.0271	0.9362	0.8381	1.0053	0.8553	0.9461
Boost	1.3883	1.3478	1.1683	1.2573	1.1432	1.2935	1.1925	1.1262	1.4360

Table E.8: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, Triangular kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.0042	1.0277	1.7117	1.0729	1.1059	1.2318	1.0613	0.9999	1.2546
Bag1	0.9906	1.0257	1.4715	1.0831	1.3254	1.5215	1.4874	1.0511	1.1290
Bag2	1.0451	1.0936	1.5888	1.0908	1.0917	1.2466	1.1302	0.9444	1.1522
Bag3	1.0369	1.0986	1.4701	1.0760	1.0375	1.0750	1.0321	0.7871	0.9294
Brag1	0.9939	1.0216	1.5667	1.0223	1.0079	1.2881	1.0792	0.8994	1.1915
Brag2	1.0411	1.0742	1.5580	1.0748	1.0577	1.2496	1.1057	0.9111	1.1288
Brag3	1.0268	1.0517	1.4910	1.0646	1.0348	1.1933	1.0761	0.8346	0.9685
Boost	1.1017	1.0993	1.1139	1.0273	1.0187	1.4427	0.9750	0.8946	1.2947

Table E.9: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2)$, $N(0,1)$ kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.0192	1.0678	0.8831	1.0763	1.1239	1.4156	1.0884	0.9895	1.5214
Bag1	0.9891	1.0197	0.7798	1.0353	1.0077	1.1429	1.1169	0.8043	1.1623
Bag2	1.0347	1.0758	0.7334	1.0891	1.0890	1.3798	1.0965	0.9315	1.7486
Bag3	1.0273	1.0994	0.8797	1.0708	1.0291	1.1221	1.0144	0.8138	0.9837
Brag1	0.9925	1.0229	0.7900	1.0256	1.0242	1.5723	1.0333	0.9226	2.1457
Brag2	1.0329	1.0628	0.8151	1.0707	1.0505	1.3790	1.0672	0.8903	1.6529
Brag3	1.0162	1.0303	0.8343	1.0621	1.0138	1.2500	1.0441	0.8277	1.0758
Boost	1.0878	1.1281	1.3405	1.0079	1.0083	1.5011	0.9778	0.8627	1.4045

Table E.10: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2, 2)$, Triangular kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.8663	1.0235	0.5182	1.0699	1.2174	1.1704	1.1578	1.1195	1.2896
Bag1	0.9185	1.0180	0.6572	1.1261	1.2293	1.2740	1.5062	0.8963	0.9965
Bag2	0.9031	1.0417	0.5623	1.1264	1.1153	1.1426	1.2199	0.9133	1.1535
Bag3	0.8519	0.9968	0.5614	1.0809	0.9956	0.9029	1.0704	0.7413	0.8273
Brag1	0.9137	1.0262	0.6001	1.0652	1.0015	1.2411	1.1502	0.8167	1.2327
Brag2	0.8846	0.9725	0.5981	1.1052	1.0107	1.1472	1.1997	0.8215	1.1165
Brag3	0.8629	0.9382	0.5889	1.0776	0.9606	0.9893	1.1195	0.7430	0.8337
Boost	1.3869	1.2869	0.7205	1.1862	1.1812	1.3093	1.1306	1.0128	1.2854

Table E.11: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0, 1)$, $N(0, 1)$ kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9305	1.0528	0.5995	1.1209	1.2245	1.2046	1.1902	1.1528	1.3754
Bag1	0.9414	0.9925	0.6847	1.0490	0.9392	1.1010	1.1299	0.7260	1.2305
Bag2	0.8698	1.0247	0.5788	1.0915	1.0725	1.1588	1.1707	0.9297	1.5636
Bag3	0.8454	1.0161	0.6961	1.0536	0.9591	0.9541	1.0337	0.7450	0.8676
Brag1	0.9447	1.0028	0.6738	1.0593	0.9875	1.3993	1.0962	0.8574	1.8871
Brag2	0.8875	0.9354	0.6851	1.0694	0.9396	1.1805	1.1318	0.7989	1.4510
Brag3	0.8515	0.9228	0.6668	1.0516	0.9107	1.0299	1.0667	0.7159	0.9044
Boost	1.4095	1.2188	0.8514	1.2482	1.1745	1.3228	1.1787	1.0370	1.3028

Table E.12: Ratio of $\text{MISE}_{NW_{CV}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0, 1)$, Triangular kernel

Appendix F

Comparing SimB to SimC

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9622	1.1890	0.8471	1.0687	1.1449	1.0062	1.1071	1.1921	1.1632
Bag1	0.9688	1.3685	0.8826	1.0750	1.1964	1.0282	1.1170	1.2205	1.1796
Bag2	0.9626	1.5823	0.8886	1.0587	1.1810	0.9796	1.1045	1.1918	1.1241
Bag3	0.9643	1.5691	0.8943	1.0619	1.1721	0.9743	1.1060	1.1737	1.1063
Brag1	0.9714	1.3296	0.8789	1.0770	1.1941	1.0367	1.1159	1.2198	1.1973
Brag2	0.9756	1.4865	0.8863	1.0821	1.2169	1.0149	1.1222	1.2328	1.1694
Brag3	0.9770	1.5014	0.8701	1.0782	1.2124	1.0039	1.1189	1.2228	1.1372
Boost	1.1671	1.4907	0.9855	1.1201	1.2077	1.1931	1.1203	1.2062	1.3460

Table F.1: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2, 2), N(0, 1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9210	1.1759	1.1066	1.0627	1.1911	1.2377	1.1062	1.2211	1.3518
Bag1	0.9222	1.3551	1.1265	1.0755	1.2675	1.2395	1.1291	1.2948	1.3675
Bag2	0.9575	1.7844	1.0937	1.0659	1.3028	1.1739	1.1167	1.3112	1.3037
Bag3	0.9452	1.7321	1.0498	1.0756	1.2724	1.1225	1.1249	1.2758	1.2271
Brag1	0.9190	1.3145	0.9222	1.0823	1.2703	1.1548	1.1339	1.3090	1.3667
Brag2	0.9707	1.9936	1.2435	1.0795	1.3154	1.2204	1.1268	1.2872	1.2951
Brag3	0.9734	1.9137	1.1780	1.0788	1.2887	1.1834	1.1242	1.2577	1.2309
Boost	1.1914	1.8417	1.6422	1.1093	1.3112	1.6324	1.0876	1.2338	1.6823

Table F.2: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim U(-2, 2)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.4050	3.6927	1.3361	1.0670	1.5301	1.2477	1.0661	1.3385	1.2934
Bag1	1.8293	5.4388	1.6081	1.1182	1.8369	1.3570	1.0935	1.4507	1.3823
Bag2	2.2513	6.3014	1.7184	1.1561	2.2451	1.3206	1.1153	1.5562	1.2800
Bag3	2.1275	5.6796	1.6555	1.1431	2.1029	1.2758	1.1099	1.4672	1.2266
Brag1	1.7150	5.1635	1.5657	1.1117	1.8190	1.3636	1.0910	1.4538	1.3978
Brag2	2.1520	7.3935	1.7497	1.1342	2.2399	1.3776	1.1018	1.5561	1.3384
Brag3	2.1803	7.0834	1.7095	1.1437	2.2657	1.3395	1.1027	1.5423	1.2801
Boost	1.7264	4.1625	1.7568	1.0861	1.5222	1.5068	1.0556	1.3061	1.5105

Table F.3: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0, 1), N(0, 1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.9725	2.8456	1.4405	1.3954	2.0869	1.5709	1.1483	1.8765	1.8063
Bag1	2.3709	3.0100	1.5840	1.6954	2.4044	1.7124	1.4469	2.0615	1.8153
Bag2	2.9193	3.1344	1.7213	1.6288	2.3637	1.5766	1.4239	1.9008	1.5900
Bag3	2.8445	3.0776	1.6279	1.5395	2.1470	1.4223	1.3258	1.6836	1.3865
Brag1	2.4628	3.1837	1.3857	1.5983	2.4663	1.6222	1.4003	2.1161	1.8659
Brag2	3.9458	4.7151	2.3056	1.6754	2.7143	1.7700	1.4052	1.8636	1.6115
Brag3	3.6981	4.0725	2.0610	1.4878	2.4108	1.5142	1.2394	1.6701	1.3574
Boost	4.2085	6.9169	3.0620	2.0068	3.2688	2.6094	1.1739	2.6079	3.1030

Table F.4: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 1, $X \sim N(0,1)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.8094	0.8082	0.7120	1.0049	1.0612	0.9304	1.0609	1.1463	1.0750
Bag1	0.8143	0.8419	0.6982	1.0176	1.0964	0.9382	1.0786	1.1750	1.0928
Bag2	0.7928	0.8326	0.6516	1.0007	1.0607	0.8626	1.0648	1.1432	1.0220
Bag3	0.7935	0.8437	0.6522	1.0027	1.0520	0.8562	1.0669	1.1271	1.0087
Brag1	0.8143	0.8356	0.7032	1.0172	1.0921	0.9373	1.0771	1.1746	1.1014
Brag2	0.8087	0.8316	0.6714	1.0167	1.0839	0.8929	1.0799	1.1709	1.0515
Brag3	0.8077	0.8382	0.6724	1.0151	1.0782	0.8869	1.0775	1.1536	1.0269
Boost	1.0200	1.0668	0.8716	1.0929	1.1606	1.1805	1.1076	1.1870	1.3044

Table F.5: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2), N(0,1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.7972	0.8566	0.8535	1.0190	1.1262	1.0982	1.0694	1.1883	1.2672
Bag1	0.7935	0.9155	0.7217	1.0365	1.1877	1.0428	1.1001	1.2783	1.2568
Bag2	0.7792	0.9856	0.6447	1.0194	1.1733	0.9365	1.0842	1.2757	1.1605
Bag3	0.7883	0.9928	0.7343	1.0218	1.1446	0.9821	1.0856	1.2245	1.1415
Brag1	0.8021	0.9176	0.8330	1.0339	1.1698	1.1043	1.0936	1.2496	1.2735
Brag2	0.8019	1.0236	0.7925	1.0267	1.1306	1.0115	1.0929	1.2082	1.1624
Brag3	0.8107	1.0235	0.7067	1.0224	1.1204	0.9401	1.0851	1.1833	1.0982
Boost	1.0059	1.2850	1.2540	1.0882	1.2659	1.4884	1.0742	1.2190	1.6707

Table F.6: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim U(-2,2)$, Triangular kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.8699	1.0579	0.6904	0.9926	1.1344	0.9426	1.0240	1.1928	1.1326
Bag1	0.9184	1.2835	0.6909	1.0269	1.1629	0.9600	1.0515	1.2168	1.1730
Bag2	0.9302	1.3989	0.6400	1.0200	1.1085	0.8360	1.0602	1.1497	1.0115
Bag3	0.9186	1.3671	0.6422	1.0163	1.0494	0.8209	1.0575	1.0718	0.9821
Brag1	0.9080	1.2486	0.6971	1.0219	1.1572	0.9622	1.0478	1.2085	1.1763
Brag2	0.9364	1.4478	0.6915	1.0198	1.0926	0.8875	1.0501	1.1220	1.0656
Brag3	0.9337	1.4624	0.6875	1.0172	1.0522	0.8658	1.0484	1.0512	1.0151
Boost	1.0449	1.3108	0.8123	1.0487	1.2172	1.1894	1.0442	1.2163	1.3836

Table F.7: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, $N(0,1)$ kernel

	$\sigma = 0.2$			$\sigma = 0.6$			$\sigma = 1.0$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.1151	1.4823	1.0126	1.3087	1.7984	1.4489	1.1158	1.7716	1.7047
Bag1	1.4113	1.6503	0.8222	1.5511	1.9764	1.3818	1.3739	1.9520	1.7551
Bag2	1.2596	1.6679	0.7509	1.3387	1.5536	1.0791	1.3118	1.5728	1.3402
Bag3	1.2282	1.6320	0.8813	1.2500	1.3542	1.0966	1.2287	1.3242	1.2293
Brag1	1.4313	1.6452	0.9826	1.3900	1.9203	1.4995	1.3007	1.8504	1.7242
Brag2	1.2995	1.7888	0.9786	1.2563	1.3824	1.1462	1.2544	1.2666	1.2733
Brag3	1.3015	2.0064	0.8243	1.1149	1.2094	0.9787	1.0966	1.1155	1.0966
Boost	2.0518	2.4928	1.5987	2.1008	2.7436	2.3087	1.2288	2.4461	2.8288

Table F.8: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 2, $X \sim N(0,1)$, Triangular kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9758	0.9827	0.7735	1.0988	1.1710	1.0827	1.1322	1.2238	1.2567
Bag1	0.9773	0.9924	0.7945	1.1104	1.1860	1.1187	1.1418	1.2267	1.2892
Bag2	0.9589	0.9438	0.7479	1.0979	1.1499	1.0639	1.1351	1.1963	1.2328
Bag3	0.9602	0.9528	0.7489	1.0991	1.1344	1.0507	1.1350	1.1756	1.2072
Brag1	0.9775	0.9941	0.7940	1.1067	1.1877	1.1293	1.1398	1.2282	1.2956
Brag2	0.9767	0.9612	0.7576	1.1137	1.1762	1.0948	1.1485	1.2267	1.2678
Brag3	0.9762	0.9547	0.7608	1.1112	1.1546	1.0710	1.1462	1.1989	1.2330
Boost	1.0750	1.1198	0.8800	1.1113	1.1925	1.2255	1.1225	1.2081	1.3326

Table F.9: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2,2)$, $N(0,1)$ kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.9849	1.0810	1.1335	1.1176	1.2452	1.2772	1.1454	1.2724	1.3444
Bag1	0.9850	1.1384	1.0609	1.1383	1.2821	1.3321	1.1645	1.3082	1.4341
Bag2	0.9716	1.0954	0.8523	1.1261	1.2857	1.2154	1.1585	1.3365	1.3874
Bag3	0.9724	1.0827	0.9393	1.1240	1.2457	1.1736	1.1547	1.2814	1.2647
Brag1	0.9856	1.1208	1.0599	1.1367	1.2812	1.3186	1.1606	1.3085	1.4422
Brag2	0.9750	1.0128	0.9552	1.1315	1.2186	1.1883	1.1623	1.2592	1.3063
Brag3	0.9747	1.0125	0.9530	1.1282	1.1782	1.1142	1.1583	1.2171	1.1898
Boost	1.0816	1.3625	1.5280	1.0997	1.2935	1.5830	1.1034	1.2480	1.5844

Table F.10: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim U(-2, 2)$, Triangular kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	0.8748	0.9074	0.7305	0.9881	1.1595	1.0692	1.0360	1.2788	1.3622
Bag1	0.8979	0.9408	0.7349	1.0305	1.1896	1.0987	1.0753	1.2489	1.3856
Bag2	0.8649	0.8359	0.6480	1.0278	1.1127	0.9655	1.0881	1.1913	1.2153
Bag3	0.8615	0.8405	0.6302	1.0222	1.0378	0.9279	1.0841	1.0987	1.1454
Brag1	0.8965	0.9333	0.7376	1.0330	1.1931	1.1122	1.0781	1.2556	1.3780
Brag2	0.8734	0.8435	0.6774	1.0217	1.0834	0.9979	1.0781	1.1488	1.2375
Brag3	0.8658	0.8147	0.6657	1.0108	1.0031	0.9560	1.0662	1.0518	1.1664
Boost	1.0194	1.0813	0.7965	1.0351	1.1978	1.2029	1.0511	1.2752	1.5051

Table F.11: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0, 1),\ N(0, 1)$ kernel

	$\sigma = 0.1$			$\sigma = 0.3$			$\sigma = 0.5$		
	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$	$n = 200$	$n = 100$	$n = 50$
Standard	1.1410	1.5066	1.0574	1.0683	1.5869	1.4604	1.0361	1.6670	1.7800
Bag1	1.3703	1.5284	1.0393	1.2721	1.8672	1.5930	1.2051	1.9249	1.8683
Bag2	1.0837	1.1759	0.7303	1.2330	1.5312	1.1829	1.2975	1.6223	1.4979
Bag3	1.0564	1.1424	0.8372	1.1683	1.2703	1.1235	1.1869	1.3062	1.3030
Brag1	1.3870	1.5284	1.0713	1.4383	1.7953	1.5478	1.1799	1.8776	1.8796
Brag2	0.9988	1.1053	0.8850	1.2059	1.1630	1.1722	1.1377	1.1859	1.3326
Brag3	0.9265	0.9531	0.8347	1.0420	1.0223	1.0189	1.0707	1.0308	1.1686
Boost	2.0233	2.5662	1.6200	1.1889	2.0776	2.2321	1.0212	1.9728	2.5351

Table F.12: Ratio of $\text{MISE}_{LL_{PLUG}}/\text{MISE}_{NW_{PLUG}}$, Model 3, $X \sim N(0, 1)$, Triangular kernel

Appendix G

Computation time

Simulation setup scenario				Estimator		
σ	n	Kernel	X	NW_{CV}	NW_{PLUG}	LL_{PLUG}
0.2	200	Gaussian	$U(-2,2)$	1021.74	4.43	3.47
0.2	200	Gaussian	$N(0,1)$	1050.74	4.45	3.49
0.2	200	Triangular	$U(-2,2)$	2725.45	8.88	6.35
0.2	200	Triangular	$N(0,1)$	2734.65	9.03	6.38
0.2	100	Gaussian	$U(-2,2)$	334.56	2.26	2.38
0.2	100	Gaussian	$N(0,1)$	347.13	2.22	2.36
0.2	100	Triangular	$U(-2,2)$	871.36	4.75	4.57
0.2	100	Triangular	$N(0,1)$	865.39	4.73	4.56
0.2	50	Gaussian	$U(-2,2)$	126.75	1.39	1.76
0.2	50	Gaussian	$N(0,1)$	127.68	1.36	1.76
0.2	50	Triangular	$U(-2,2)$	304.03	3.02	3.54
0.2	50	Triangular	$N(0,1)$	305.88	3.03	3.52

Table G.1: Computer time (in seconds) for standard estimation methods

Simulation setup scenario				Estimator		
σ	n	Kernel	X	NW_{CV}	NW_{PLUG}	LL_{PLUG}
0.2	200	Gaussian	$U(-2,2)$	14379.16	153.37	37.49
0.2	200	Gaussian	$N(0,1)$	14502.82	154.94	36.49
0.2	200	Triangular	$U(-2,2)$	32935.96	309.12	39.67
0.2	200	Triangular	$N(0,1)$	32744.61	310.86	45.00
0.2	100	Gaussian	$U(-2,2)$	5382.32	69.22	34.79
0.2	100	Gaussian	$N(0,1)$	5445.76	69.20	33.76
0.2	100	Triangular	$U(-2,2)$	12824.17	129.34	36.68
0.2	100	Triangular	$N(0,1)$	12731.46	129.53	37.45
0.2	50	Gaussian	$U(-2,2)$	2193.08	38.96	28.21
0.2	50	Gaussian	$N(0,1)$	2189.79	39.04	28.20
0.2	50	Triangular	$U(-2,2)$	5065.29	62.93	29.63
0.2	50	Triangular	$N(0,1)$	4972.85	62.95	30.00

Table G.2: Computer time (in seconds) for Bag1 estimation methods

Simulation setup scenario				Estimator		
σ	n	Kernel	X	NW_{CV}	NW_{PLUG}	LL_{PLUG}
0.2	200	Gaussian	$U(-2,2)$	14495.90	262.15	260.15
0.2	200	Gaussian	$N(0,1)$	14612.41	256.61	257.80
0.2	200	Triangular	$U(-2,2)$	33092.35	701.10	509.43
0.2	200	Triangular	$N(0,1)$	32834.59	721.02	505.42
0.2	100	Gaussian	$U(-2,2)$	5402.52	133.02	160.04
0.2	100	Gaussian	$N(0,1)$	5401.44	134.61	159.23
0.2	100	Triangular	$U(-2,2)$	12703.59	354.08	358.98
0.2	100	Triangular	$N(0,1)$	12636.14	382.80	335.43
0.2	50	Gaussian	$U(-2,2)$	2203.06	83.32	123.14
0.2	50	Gaussian	$N(0,1)$	2201.23	83.27	121.63
0.2	50	Triangular	$U(-2,2)$	5057.74	339.18	247.56
0.2	50	Triangular	$N(0,1)$	5016.37	361.54	249.60

Table G.3: Computer time (in seconds) for Bag2 estimation methods

Simulation setup scenario				Estimator		
σ	n	Kernel	X	NW_{CV}	NW_{PLUG}	LL_{PLUG}
0.2	200	Gaussian	$U(-2,2)$	16802.25	240.20	268.79
0.2	200	Gaussian	$N(0,1)$	16994.83	255.09	235.88
0.2	200	Triangular	$U(-2,2)$	41072.16	528.41	504.80
0.2	200	Triangular	$N(0,1)$	40907.88	500.83	501.84
0.2	100	Gaussian	$U(-2,2)$	6294.51	128.23	160.45
0.2	100	Gaussian	$N(0,1)$	6282.62	127.89	163.27
0.2	100	Triangular	$U(-2,2)$	14604.62	251.30	357.98
0.2	100	Triangular	$N(0,1)$	14533.05	249.93	336.18
0.2	50	Gaussian	$U(-2,2)$	2501.71	80.50	122.53
0.2	50	Gaussian	$N(0,1)$	2530.32	80.67	122.49
0.2	50	Triangular	$U(-2,2)$	5430.09	161.29	251.58
0.2	50	Triangular	$N(0,1)$	5389.90	160.71	275.19

Table G.4: Computer time (in seconds) for Bag3 estimation methods

Simulation setup scenario				Estimator		
σ	n	Kernel	X	NW_{CV}	NW_{PLUG}	LL_{PLUG}
0.2	200	Gaussian	$U(-2,2)$	751.94	10.82	18.08
0.2	200	Gaussian	$N(0,1)$	770.83	10.67	18.19
0.2	200	Triangular	$U(-2,2)$	1611.13	23.35	42.70
0.2	200	Triangular	$N(0,1)$	1611.84	24.69	43.38
0.2	100	Gaussian	$U(-2,2)$	257.38	5.21	9.45
0.2	100	Gaussian	$N(0,1)$	260.31	5.10	9.50
0.2	100	Triangular	$U(-2,2)$	570.10	13.65	21.65
0.2	100	Triangular	$N(0,1)$	579.80	13.73	21.78
0.2	50	Gaussian	$U(-2,2)$	97.89	3.06	6.06
0.2	50	Gaussian	$N(0,1)$	98.19	3.04	6.04
0.2	50	Triangular	$U(-2,2)$	229.62	7.80	13.35
0.2	50	Triangular	$N(0,1)$	231.45	8.00	13.44

Table G.5: Computer time (in seconds) for Boost estimation methods

Simulation setup scenario				Improvement method				
σ	n	Kernel	X	Standard	Bag1	Bag2	Bag3	Boost
0.2	200	Gaussian	$U(-2,2)$	16009.51	8115.45	8364.47	7975.99	13592.29
0.2	100	Gaussian	$U(-2,2)$	1895.39	1029.12	1054.24	1044.85	1599.60
0.2	50	Gaussian	$U(-2,2)$	264.03	137.11	152.71	171.88	207.01

Table G.6: Computer time (in seconds) for the BRNP estimator

Appendix H

Graphs and figures

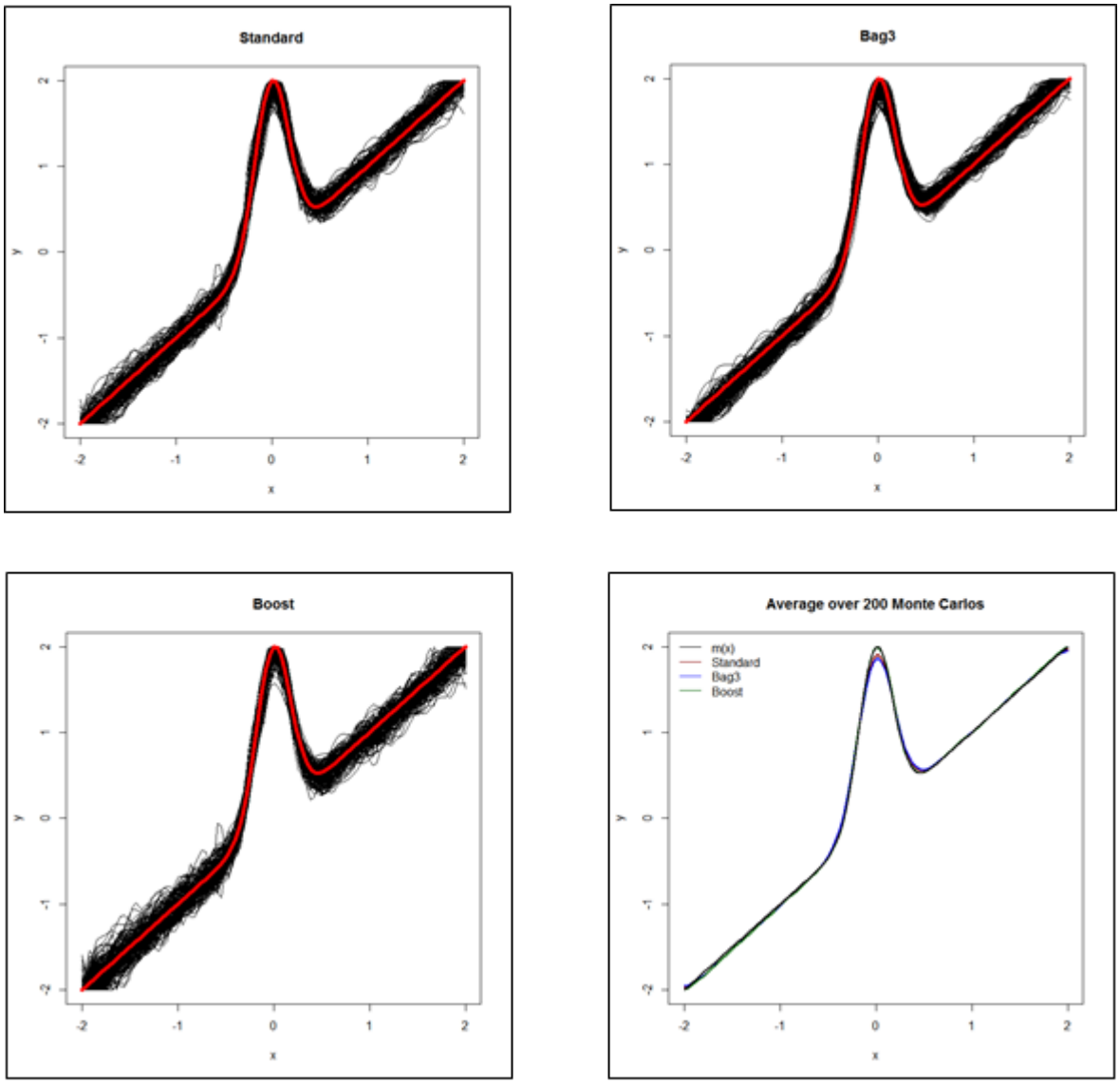

Figure H.1: Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$.

Top left: The standard Nadaraya-Watson estimator with cross-validation bandwidth selection. *Top right:* The bagged Nadaraya-Watson estimator with cross-validation bandwidth selection. *Bottom left:* The boosted Nadaraya-Watson estimator with cross-validation bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

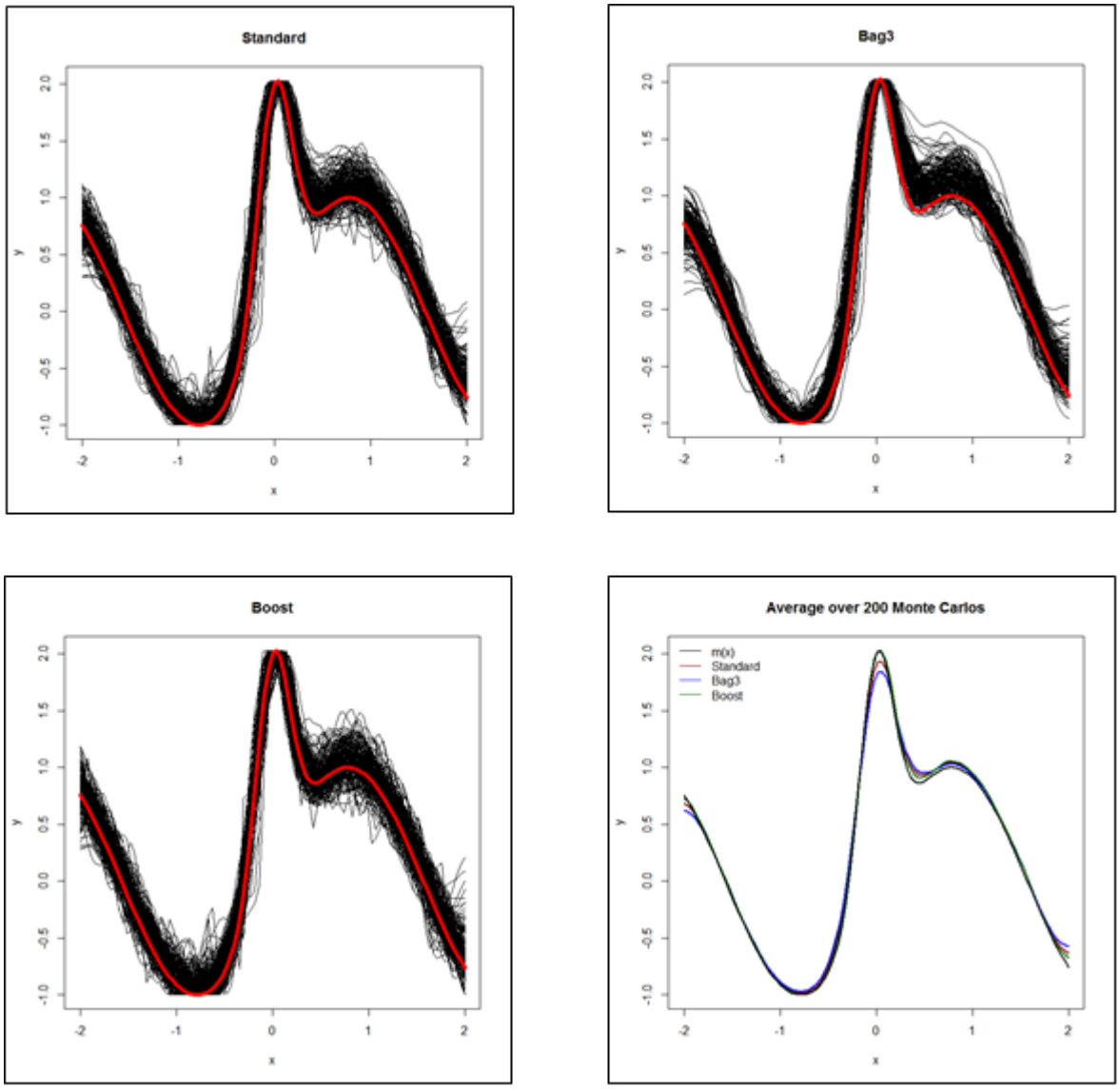

Figure H.2: Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.

Top left: The standard Nadaraya-Watson estimator with cross-validation bandwidth selection. *Top right:* The bagged Nadaraya-Watson estimator with cross-validation bandwidth selection. *Bottom left:* The boosted Nadaraya-Watson estimator with cross-validation bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

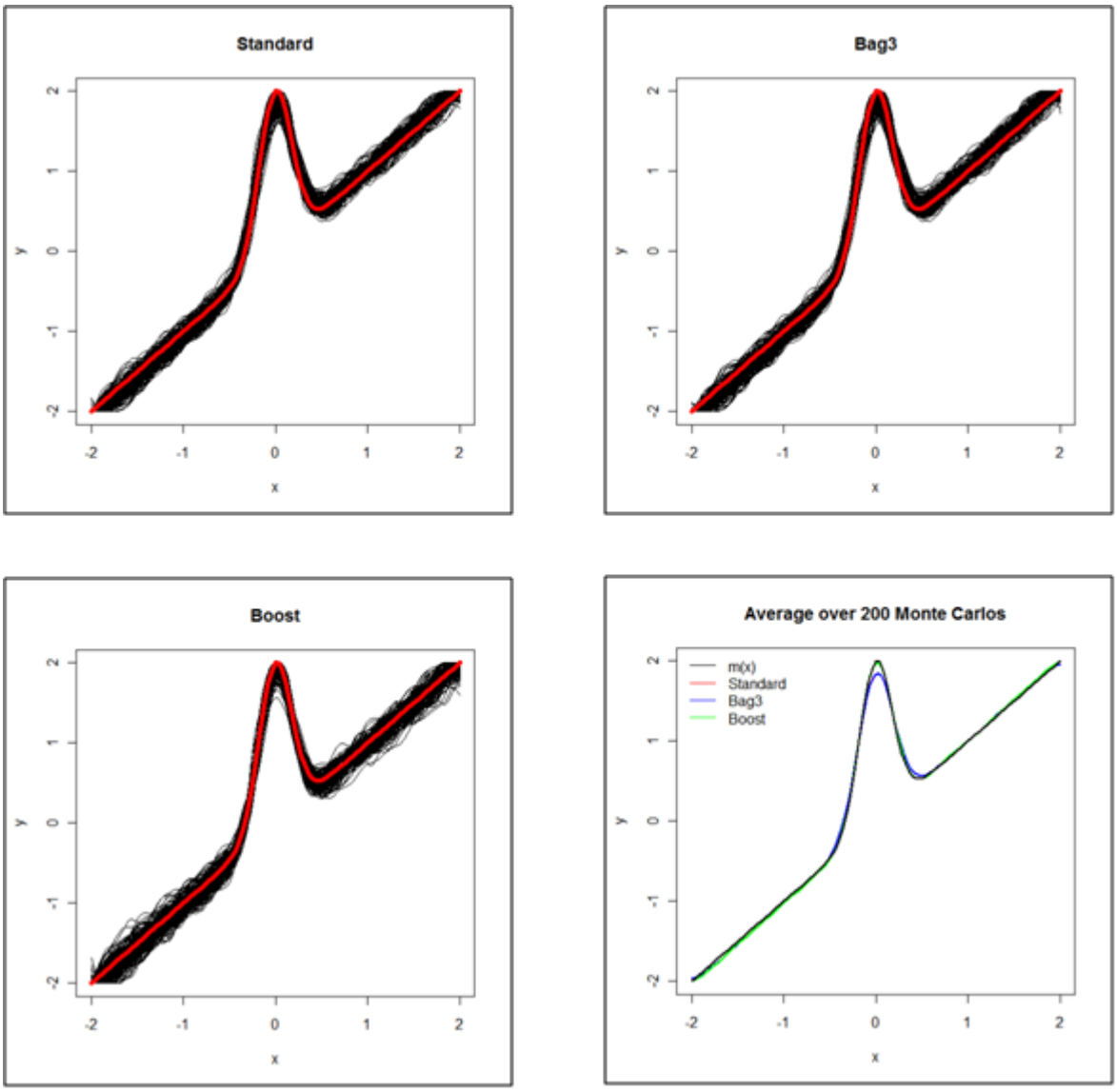

Figure H.3: Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$.

Top left: The standard Nadaraya-Watson estimator with plug-in bandwidth selection. *Top right:* The bagged Nadaraya-Watson estimator with plug-in bandwidth selection. *Bottom left:* The boosted Nadaraya-Watson estimator with plug-in bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

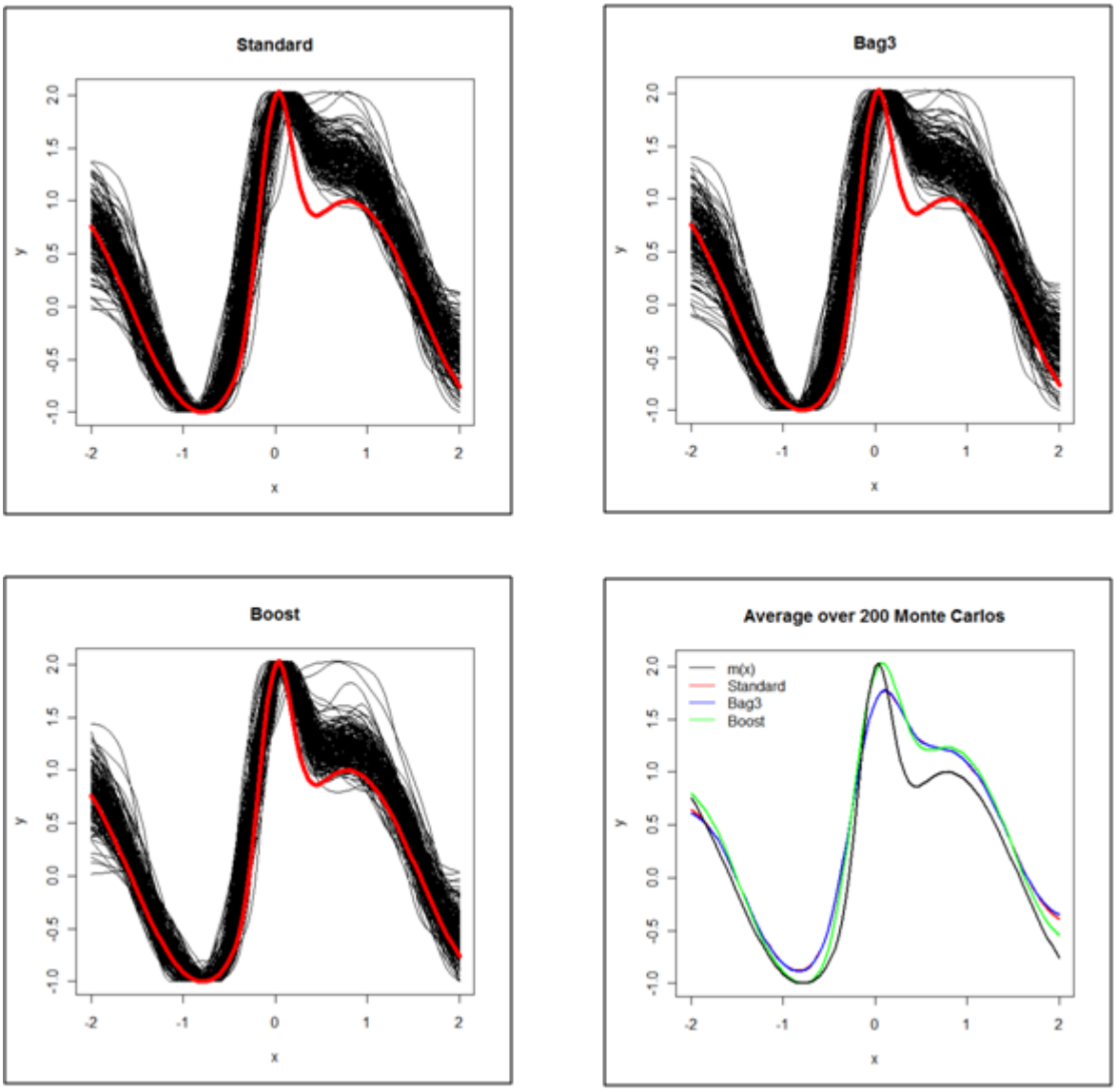

Figure H.4: Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.

Top left: The standard Nadaraya-Watson estimator with plug-in bandwidth selection. *Top right:* The bagged Nadaraya-Watson estimator with plug-in bandwidth selection. *Bottom left:* The boosted Nadaraya-Watson estimator with plug-in bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

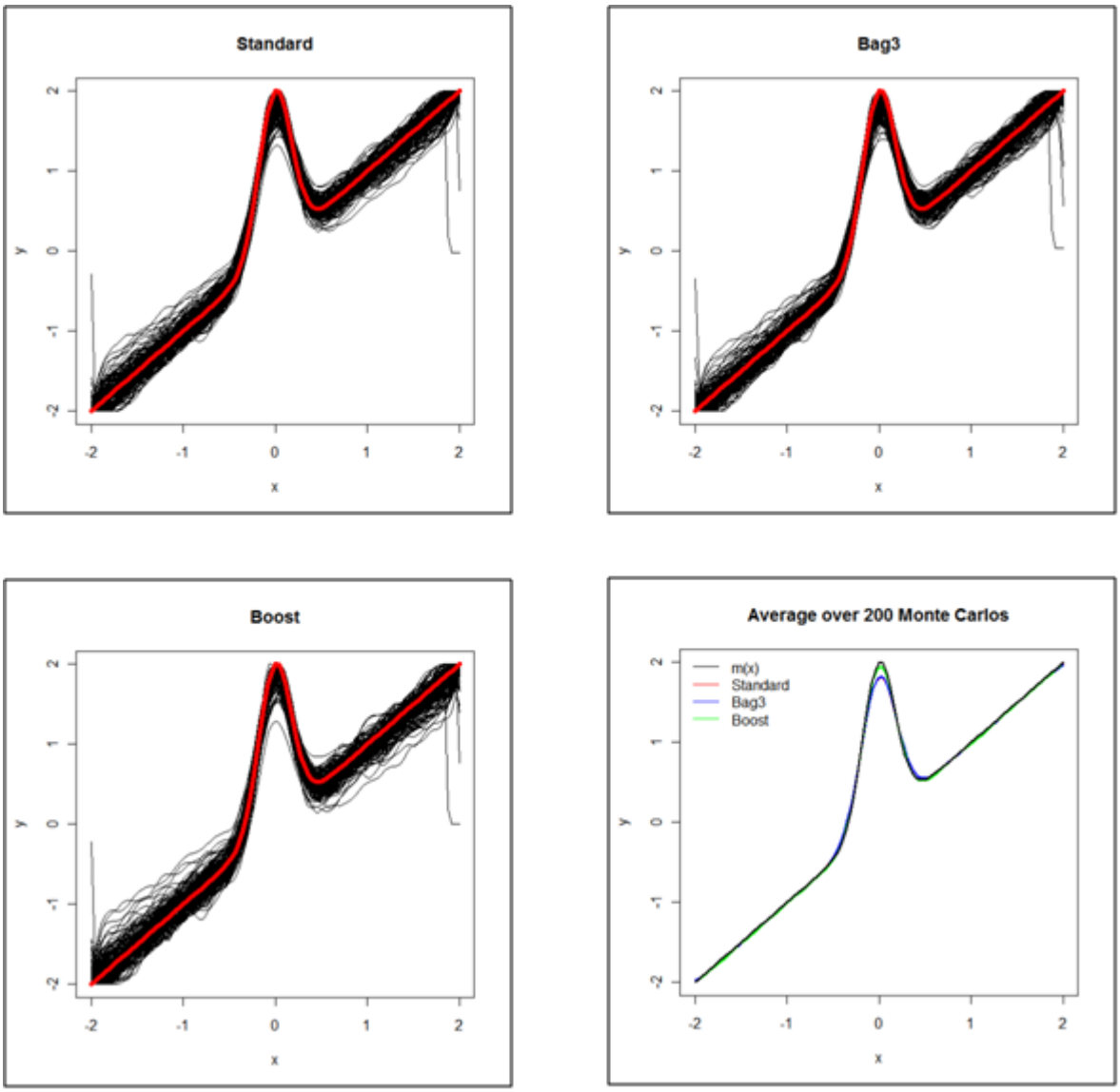

Figure H.5: Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$.

Top left: The standard local linear estimator with plug-in bandwidth selection. *Top right:* The bagged local linear estimator with plug-in bandwidth selection. *Bottom left:* The boosted local linear estimator with plug-in bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

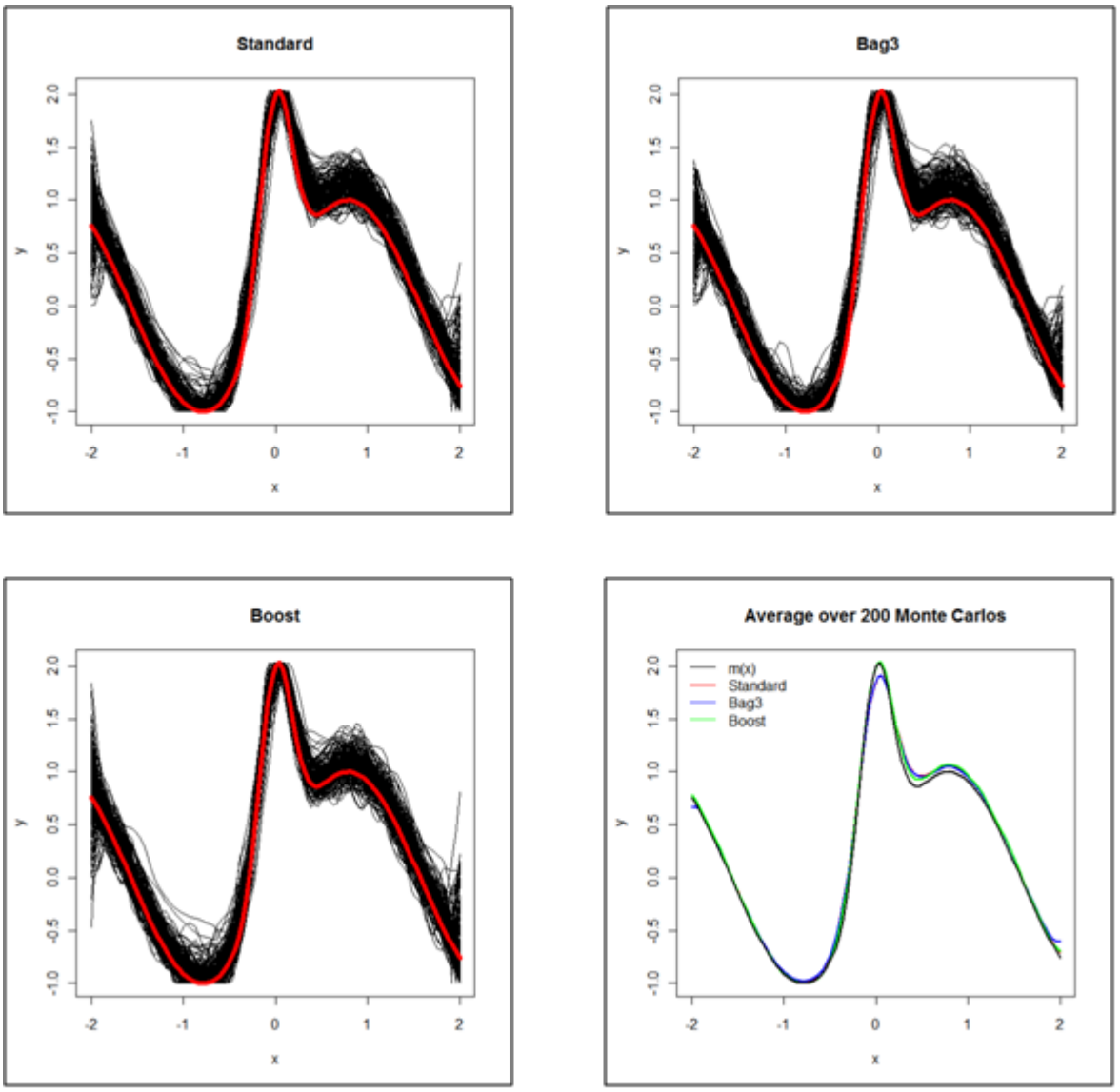

Figure H.6: Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.

Top left: The standard local linear estimator with plug-in bandwidth selection. *Top right:* The bagged local linear estimator with plug-in bandwidth selection. *Bottom left:* The boosted local linear estimator with plug-in bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

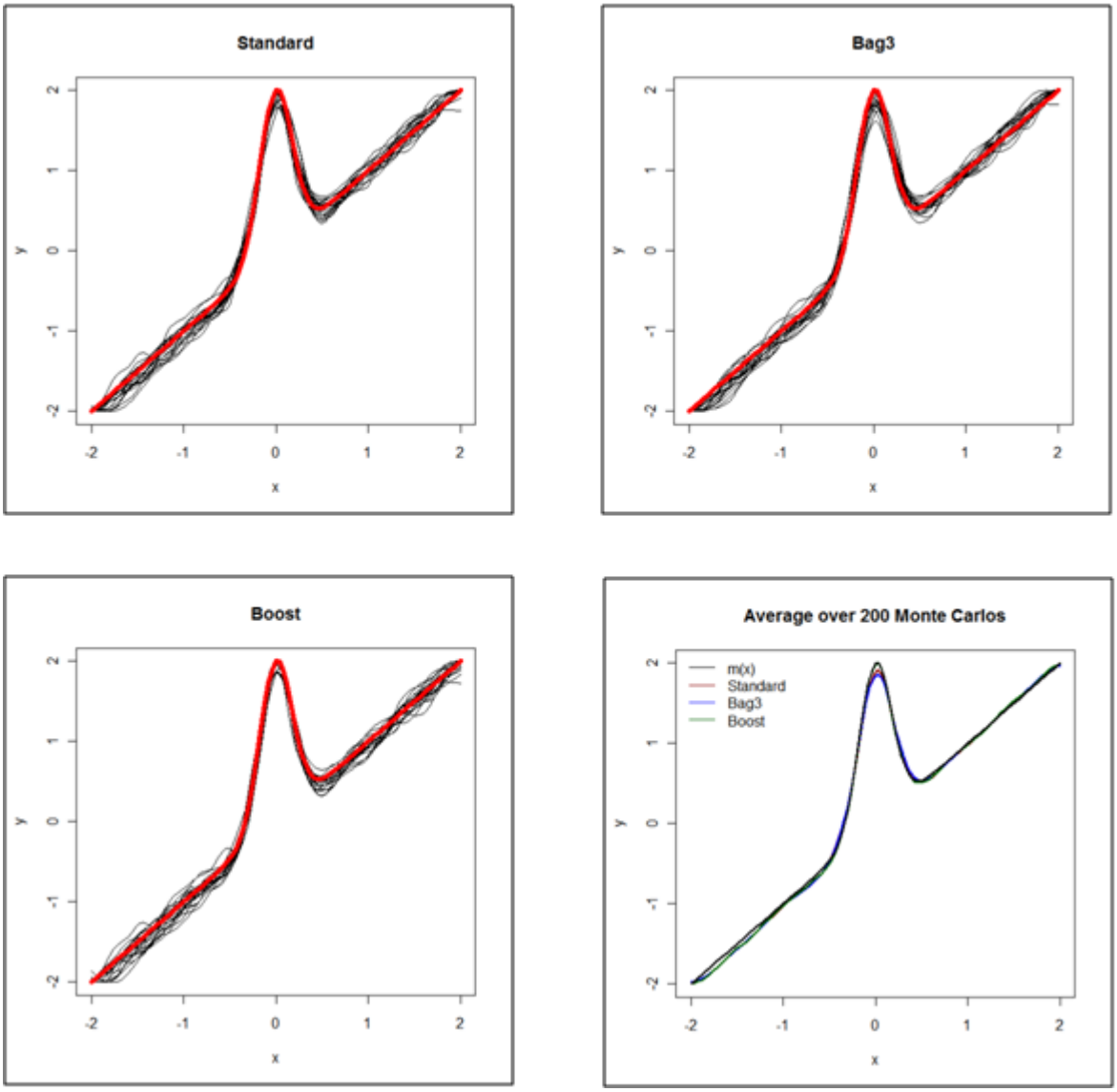

Figure H.7: Model 1, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 200$.

Top left: The standard BRNP estimator with cross-validation bandwidth selection. *Top right:* The bagged BRNP estimator with cross-validation bandwidth selection. *Bottom left:* The boosted BRNP estimator with cross-validation bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.

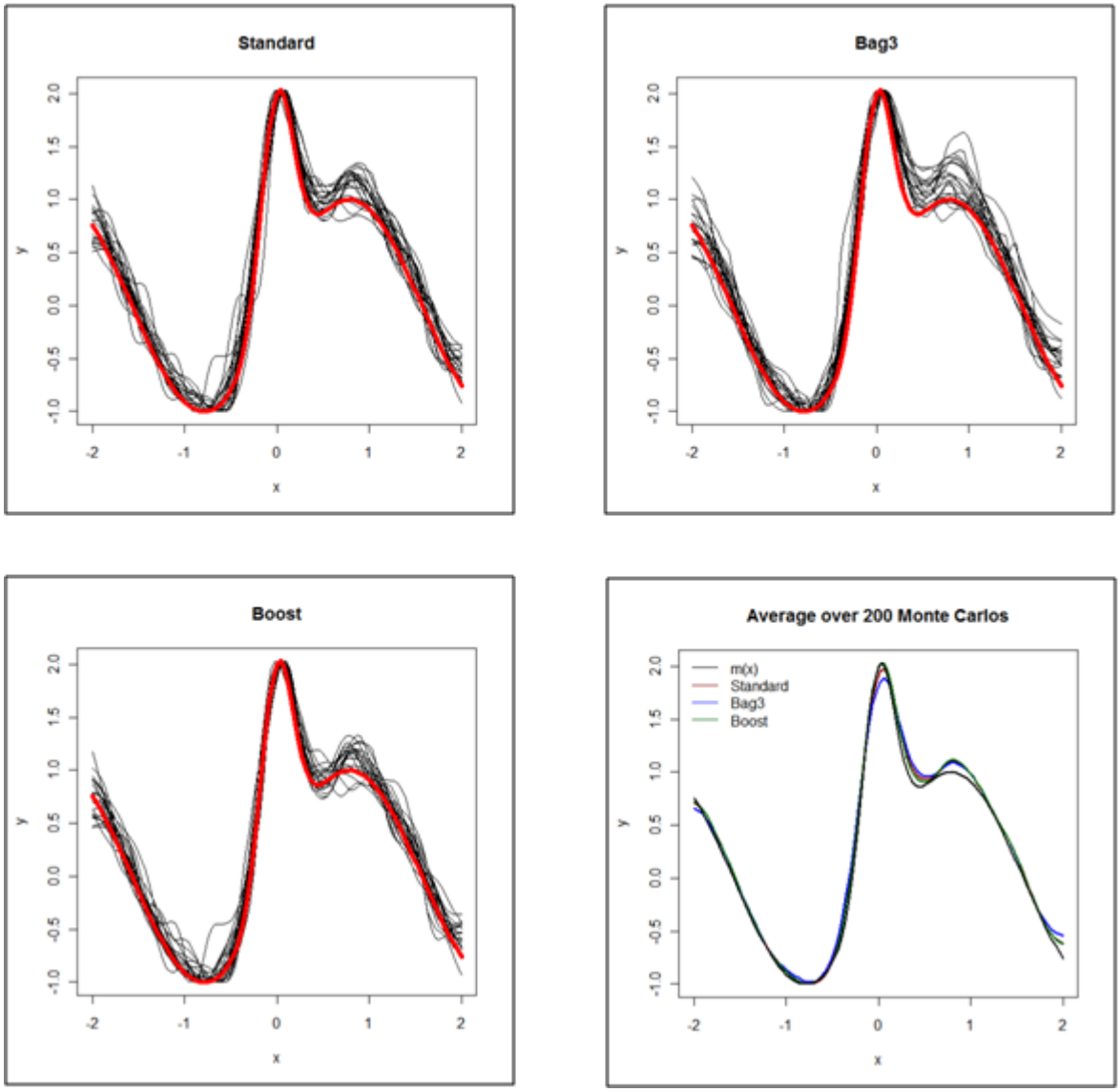

Figure H.8: Model 2, $X \sim U(-2, 2)$, $N(0, 1)$ kernel, $\sigma = 0.2$, $n = 100$.

Top left: The standard BRNP estimator with cross-validation bandwidth selection. *Top right:* The bagged BRNP estimator with cross-validation bandwidth selection. *Bottom left:* The boosted BRNP estimator with cross-validation bandwidth selection. The black lines represent the estimates for each of the 200 Monte Carlo simulations. The red line indicates $m(x)$. *Bottom right:* The average estimates over the 200 Monte Carlo simulations for Standard, Boost, Bag3, as indicated in the legend.