QATAR UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

REAL-TIME STATISTICAL LEARNING WITH APPLICATION TO FETAL WELL-BEING MONITORING AND REAL-ESTATE VALUE PREDICTION IN QATAR

BY

OMAMA AL HAMED

A Project Submitted to

the College of Arts and Sciences

in Partial Fulfillment of the Requirements for the Degree of

Bachelor in Statistics

June 2023

COMMITTEE PAGE

The members of the Committee approve the Project of

Omama Al Hamed defended on 06/06/2023.

Dr. Mohamed Chaouch
Project Supervisor

ABSTRACT

AL Hamed, Omama, Project : June : 2023, Bachelor in Statistics

Title: Real-time statistical learning with application to fetal well-being monitoring and real-estate value prediction in Qatar

Supervisor of Project: Dr. Mohamed Chaouch.

Predictive models, including supervised/unsupervised clustering and time series forecasting, represent powerful data science tools that help in taking decisions in different fields such as in medicine, finance and business. Nowadays with the progress in the technology of electronic devises, we have an easy access to structured and unstructured data (e.g. numeric, image, video, text, ...). Storing such massive amount of data and analyze them in real time is one of the most challenging topics in data science. In this project, we introduce a nonparametric predictive model that allows to classify objects or predict unknown values while either a big data set is at our disposal or data is received in streaming. The quality of the proposed predictor/classifier in terms of accuracy and computation time reduction is assessed through simulated data. Moreover, application to real-time monitoring of fetal well-being during pregnancy is discussed using cardiotocography data. Then the proposed methodology is also applied for real-time dates data classification which could be of great interest in food-quality monitoring. Finally, a third application is considered for an online prediction of real estate value in Qatar.

DEDICATION

To my Mother

ACKNOWLEDGMENTS

All praise is due to Allah, whom we ask for help, and who blesses us with everything we need.

I would like to express my special thanks of gratitude to my family who supported me during my studies, and to Dr. Mohamed Chaouch who gave me the precious opportunity to do this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Problem statement

Studying the relationship between variables is a common concern in the field of statistics. Exploring/understanding their associations is essential for making conclusions about the variables of interest in the presence of other variables. Hence, there is a wide range of statistical techniques that were developed for the aim of building predictive models. In general, these models fall into three main categories: Parametric, Non-Parametric and Semi-Parametric. They differ by the assumptions imposed on the data. The parametric methods, in general, make specific assumptions on the underlying distribution of the data, and they assume a specific shape of the link function. Although the parametric models are the most powerful in prediction, their results may be misleading or inaccurate if their assumptions are not met. On the other hand, the non-parametric models relax these assumptions and hence are more flexible and less restrictive. However, there are certain issues regarding the usage of the non-parametric models. For example, obtaining good results by these models requires a larger set of training data, and training the models takes a great deal of time. This project aims to discuss non-parametric estimators that provide reasonably accurate results with minimum time cost.

Project outline

In Chapter 2, we begin with a quick overview of the parametric regression models. Then, we move to the offline non-parametric regression estimation. Chapter 3 focuses on dealing with large datasets where reducing the computation time becomes necessary. The recursive online estimator is introduced, and then we compare its

performance with an offline estimator. Then, in Chapter 4, we explain the adaptation of these two estimators to the case of supervised learning. Finally, chapter 5 presents some applications with real data. The results include some comparisons between different parametric and non-parametric estimators in terms of accuracy and calculation time.

Contribution of the Project

This project introduces real-time predictive models when big data set is available or data are received in streaming. We adapt the Robbins-Monro estimator to the supervised statistical learning problem. A comparison between the offline Nadaraya-Watson estimator and the proposed estimator is considered to assess the consistency as well as the computation time efficiency. Several applications are considered in this project. The first one is medical which consists in real-time monitoring of fetal well-being during pregnancy. Given a certain number of medical measures, the proposed model allows to assist the doctor to decide whether the fetal state is normal, suspicious or pathological. The second application, aims to predict the class of dates in real-time given some dimension/size/shape, color and texture predictors. This application is of great interest in food-quality assessment. Finally, we apply the proposed Robbins-Monro estimator to predict the value of real estate in Qatar in real-time.

CHAPTER 2: OFFLINE REGRESSION ESTIMATION

Regression models aim to study the relationship between one variable of interest, say Y, and other explanatory variables called predictors. In practice, it is very important to understand how these variables are concomitant. Building such models allows us to understand/explain the existing dynamic in the relationship between the variables and allows to predict the unknown variable Y given specific values of the predictors. In the statistical literature, authors were interested in investigating several regression models. One finds three families of models: (1) parametric models, (2) nonparametric models, and (3) semi-parametric models.

Linear regression models belong to the family of parametric models and aim to explain the impact of the explanatory variables (predictors) on the variable of interest Y assuming a linear relationship. Moreover, in order to be able to estimate the parameters, on which depend the model, and study their properties, one needs to consider some specific probabilistic distribution on the variable Y. These two conditions are, in general, very difficult to be satisfied when we deal with real data. Nonparametric models came to relax the shape constraint assumed in linear models on the function linking Y to the predictors and do not necessarily assume that the data is generated according to a specific probabilistic distribution. Finally, semiparametric models can be seen as a combination of parametric and nonparametric approaches. They are designed to get the benefits from parametric approaches as well as those from the nonparametric ones. In the following sections, we will first give a brief review of linear regression models. Then, we discuss nonparametric kernel-type estimation of the regression function.

Linear regression model

Linear regression analysis is a statistical approach that seeks to fit a model that predicts a response variable. The aim is to find the smallest set of predictors that explain most of the variability in the response. To formulate mathematically the problem, let us denote $(X_1, \ldots, X_p) \in \mathbb{R}^p$ be a p-dimensional vector of predictors that are concomitant with a real-valued response variable $Y \in \mathbb{R}$. We consider the following linear relationship between $\mathbf{X}$ and Y:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon, \tag{2.1}$$

where $\boldsymbol{\beta} := (\beta_0, \beta_1, \ldots, \beta_p)^\top$ is a vector of unknown parameters defining the model in (2.1). The purpose here is to estimate the $(p+1)$ regression coefficients given an independent and identically distributed (i.i.d.) random sample, say $\{(X_{i1}, \ldots, X_{ip}, Y_i) : i = 1, \ldots, n\}$ generated according to the following linear model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i, \quad i = 1, \ldots, n, \tag{2.2}$$

where

Y_i is the i-th observation of the variable Y

X_{ij} is the i-th observation of the j-th predictor X_j

ϵ_i is the error term in the model. It includes the missing information that might explain linearly the values Y_i after considering $(X_{ij})_{j=1,\ldots,p}$ in the model.

Let us now formulate the problem using matrix notations:

Definition 1. *The linear model in* (2.2) *can be written as follows:*

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.3}$$

where

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \end{pmatrix} \equiv \begin{pmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \\ \vdots \\ \mathbf{X}_n^\top \end{pmatrix} \ \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \ \textit{and} \ \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

The estimation of the parameters in (2.3) are usually estimated based on the minimization of the errors. For this we use a loss function that considers the "global" errors of the model. There are several loss functions in the literature. Here we focus on the symmetric and quadratic loss function leading to the so-called *Least Squared* (LS) estimator of $\boldsymbol{\beta}$.

Let us now state the Gauss-Markov assumptions under which the LS estimator achieves some relevant properties, such as unbiasedness, BLUE, and allow to determine the asymptotic distribution of the estimator. The last result is of great interest since it represents the masterpiece in building confidence intervals as well as performing hypothesis testing procedure.

Assumptions:

$(A1)$ No perfect collinearity: the matrix $\mathbf{X}$ is of full rank.

$(A2)$ Zero mean: $\mathbb{E}(\boldsymbol{\epsilon}) = 0$.

$(A3)$ Homoscedasticity: $\text{var}(\boldsymbol{\epsilon}) = \sigma_{\boldsymbol{\epsilon}}^2 \mathbf{I}_n$.

$(A4)$ Normality: $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\epsilon}}^2 \mathbf{I}_n\right)$.

Definition 2. *(LS estimator)*

Under assumption (A1), the LS estimator, say $\widehat{\boldsymbol{\beta}}_n = \left(\widehat{\beta}_{0,n}, \widehat{\beta}_{1,n}, \ldots, \widehat{\beta}_{p,n}\right)^\top$*, of the true regression coefficients* $\widetilde{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \mathbb{E}\left[(Y - \beta_0 - \beta_1 X_1 - \cdots - \beta_p X_p)^2\right]$ *is defined as*

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_n &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip})\right)^2 \\ &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \end{aligned} \tag{2.4}$$

where $\|.\|_2$ *denotes the Euclidean norm.*

Remark 1. *Note that assumption (A1), which supposes that the matrix* $\mathbf{X}$ *is of full rank, guarantees the existence of the inverse of the matrix* $\mathbf{X}^\top\mathbf{X}$ *where* $\mathbf{X}^\top$ *denotes the transpose of the matrix* $\mathbf{X}$*. Therefore, the estimator* $\widehat{\boldsymbol{\beta}}_n$ *exists and is unique.*

By taking the first derivative of $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ with respect to (w.r.t.) $\boldsymbol{\beta}$, one can easily deduce that $\widehat{\boldsymbol{\beta}}_n$ is the zero of the following estimating equation

$$\mathbf{X}^\top\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right) = 0.$$

Consequently, one deduces that

$$\widehat{\boldsymbol{\beta}}_n = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}. \tag{2.5}$$

Proposition 1. *(Asymptotic properties of* $\widehat{\boldsymbol{\beta}}_n$*)*

Suppose that assumptions (A1)-(A4) hold true. Then, one gets

1. *The LS estimator $\widehat{\boldsymbol{\beta}}_n$ is unbiased. That is $\mathbb{E}(\widehat{\boldsymbol{\beta}}_n) = \widetilde{\boldsymbol{\beta}}$.*

2. *The variance-covariance matrix of $\widehat{\boldsymbol{\beta}}_n$ is var$(\widehat{\boldsymbol{\beta}}_n) = \sigma_{\boldsymbol{\epsilon}}^2 \, \mathbb{E}\left[\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}\right] =: \Sigma$.*

3. *$\widehat{\boldsymbol{\beta}}_n$ has the minimal variance among all linear unbiased estimators. Therefore, it is BLUE (Best Linear Unbiased Estimator) of $\widetilde{\boldsymbol{\beta}}$.*

4. $\widehat{\boldsymbol{\beta}}_n \sim \mathcal{N}\left(\widetilde{\boldsymbol{\beta}}, \Sigma\right)$.

Despite its simplicity, the Least Squared estimator suffers of several drawbacks among which one can cite, for instance, the sensitivity of the estimator to the existence of outliers. Alternatives to the LS approaches were considered in the literature. A class of robust regression models, which considers loss functions less sensitive to the presence of outliers in the data, were investigated in the literature. Among which one cites for instance the LAD (Least Absolute Deviation) (see the book of Huber ...)

In practice one may have several predictors ($p >> n$). The identification of the most relevant predictor that explains the variability in the response variable Y is of great importance. In such cases reducing the dimensionality of the space of predictors, using approaches such as stepwise selection, AIC or BIC criterion became one of the most important steps in linear regression analysis. Unfortunately, these techniques are in general not very efficient. However, approaches such as LASSO (least absolute shrinkage and selection operator) represent an interesting alternative to the LS approach.

Offline nonparametric estimation

Inferential statistics are generally classified into parametric and non-parametric techniques. Their aim is to use random samples to infer the properties of the populations, and to explore the relationships between variables. The parametric models rely on

making many assumptions about the underlying distributions of the data. Hence, the nature of the population parameters and their number are known in advance. For example, if it is assumed that the data are collected from a normal population, then we know that the only parameters that need to be estimated are μ and σ. However, the non-parametric approach is distribution-free and makes as few assumptions as possible. Let $\mathbf{X} := (X_1, \ldots, X_p)$ be a vector of independent random variables and consider Y as the dependent variable. In practice scientists are always interested in understanding how $\mathbf{X}$ and Y are concomitant. Modeling the relationship between the response variable and the predictor is one of the most studied topics in the literature. Let us consider the following regression model:

$$Y = m(\mathbf{X}) + \epsilon, \tag{2.6}$$

where ϵ represents the error term satisfying $\mathbb{E}(\epsilon|X) = 0$ and $m(\cdot)$ the unknown regression function. The purpose is to estimate the regression function based on $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ an i.i.d random sample distributed as $(\mathbf{X}, Y)$.

In a nonparameteric setting, we do not put any shape constraint on the form of m. The idea is to let the data talk about itself to estimate the form of the regression function. Note that the elimination of restrictions on the shape of m allows unlimited possibilities for the unknown functions. Moreover, observe that no prior probability distribution is assumed for the error terms.

Under the assumption that $\mathbb{E}(\epsilon|\mathbf{X}) = 0$, and taking the conditional expectation

in both sides of equation (2.6), one can easily show that, for $\mathbf{X} = \mathbf{x}$, one has

$$\begin{aligned} m(x) &= \mathbb{E}(Y \mid \mathbf{X} = \mathbf{x}) \\ &= \int y f_{Y|\mathbf{X}}(y|\mathbf{x}) dy, \end{aligned} \tag{2.7}$$

where $f_{Y|\mathbf{X}(y|\mathbf{x})}$ denotes the conditional probability density function of Y given $\mathbf{X} = \mathbf{x}$.

A plug-in estimator of the regression function could be obtained by replacing $f_{Y|\mathbf{X}}$ by its estimator. This leads us to first focus, in the following section, on the nonparametric estimation of the probability density function of a univariate random variable X. Note that the results discussed below for the univariate case could easily be extended to multivariate random variables.

Kernel density estimation

When studying random variables, it is more likely that their true distributions are unknown. However, there are several techniques devoted for the purpose of estimating these distributions. One way of understanding a random variable is by the identification of its density function. The graphical representation of the density function conveys an impression of the spread of a random variable and its features, which also gives a glimpse of the most and least likely situations. Additionally, making further exploration of a random variable requires knowing its density function. For example, some calculations cannot be carried out without the mathematical expression of the density, such as the expectation, variance, skewness, etc.

Density estimation refers to a set of statistical inference techniques used to draw conclusions about the population of the variable of interest. These techniques make use of random samples in different ways in order to draw insights about the population of

the data. In this section we focus on a non-parametric approach known as the kernel density estimator.

Let X be a random variable (r.v) with a cumulative distribution function (C.D.F.) $F : \mathbb{R} \longrightarrow (0, 1)$ defined by $x \mapsto F(x) = \mathbb{P}(X \leq x)$. Let f be the probability density function (p.d.f) of a random variable X. We know that $F'(x) = \dfrac{dF(x)}{dx} = f(x)$, for any $x \in \mathbb{R}$. More explicitly,

$$\begin{aligned} f(x) = \lim_{h \to 0} \frac{F(x+h) - F(x-h)}{2h} &= \lim_{h \to 0} \frac{1}{2h} \mathbb{P}(x - h \leq X \leq x + h) \\ &= \lim_{h \to 0} \frac{1}{2h} \mathbb{P}\left(-1 \leq \frac{X - x}{h} \leq 1\right) \end{aligned}$$

Let $X_1, \ldots, X_n$ be n independent and identically distributed (i.i.d.) copies of X with the same probability distribution F. Then, for $h := h_n(X_1, \ldots, X_n)$ small enough, a naive estimator of f, at a fixed point x, could be defined as:

$$\widehat{f}_{n,0}(x; h) := \frac{1}{2nh} \sum_{i=1}^{n} \mathbb{1}_{\left\{-1 \leq \frac{X_i - x}{h} \leq 1\right\}}. \tag{2.8}$$

Due to the indicator function in 2.8, there is a lack of smoothness in this estimator, and its derivative is undefined. From a practical point of view, most of the densities are smooth functions. Therefore, it is better to replace the indicator function in (2.8) with a smooth function, commonly in the literature by the kernel, and denoted K. Theoretically, considering a smooth version of the estimator (2.8) would lead to a better consistency rate, bias and variance reduction, without imposing heavy constraints on the kernel K. Additionally, note that only observations in the neighborhood of x will effectively contribute to the estimation of the density at the point x. Therefore, a

kernel-type estimator of the density, at a fixed point x, is defined as follows:

$$\widehat{f}_n(x;h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \tag{2.9}$$

where K is the kernel and $h := h_n$ is a sequence of positive real numbers tending to zero as n goes to infinity, known as the bandwidth. Note that, in contrast to $\widehat{f}_{n,0}$, the estimator in (2.9) uses all the data in the sample. Closer the observation X_i to the fixed point x, higher will be its contribution in calculating the value of $\widehat{f}_n(x;h)$.

Let us now compare the performance of $\widehat{f}_{n,0}$ and $\widehat{f}_n$ based on some simulated data. Let $X_1, \ldots, X_n$ be an i.i.d. random sample generated according to:

`model 1`: $X_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$.

`model 2`: $X_i \sim \text{Beta}(2, 3.5)$ for $i = 1, \ldots, n$.

One can see that $\widehat{f}_n$ depends on two tuning parameters: the kernel K and the bandwidth h_n. The quality of the estimation will depend on the choice of these two parameters. In the following we are interested in discussing the effect of the choice of K and h on the estimation of the density function under `model 1` and `model 2`. To assess the performance of each estimator, we consider the commonly used Integrated Square Error `ISE` defined as:

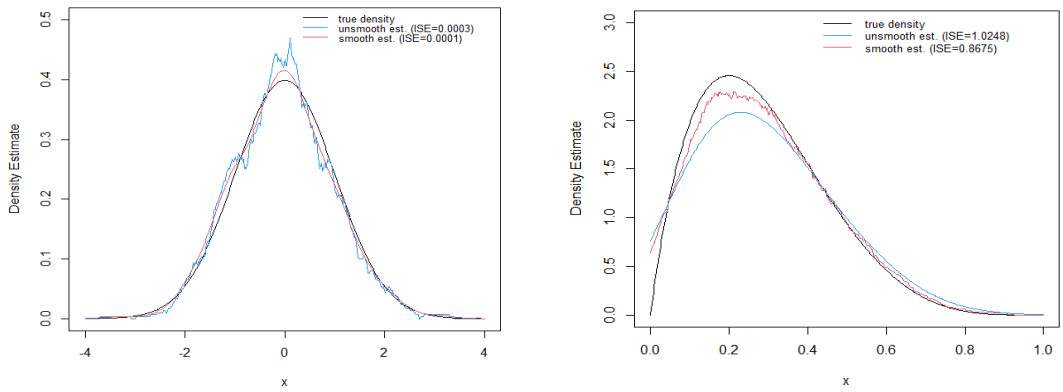
$$\texttt{ISE}(\tilde{f}) = \int \left(\tilde{f}(x) - f(x)\right)^2 dx, \tag{2.10}$$

where $\tilde{f}$ denotes one of the estimators of the density ($\widehat{f}_{n,0}$ and $\widehat{f}_n$) and f is the true density under `model 1` or `model 2`. In practice, we use the Riemann approximation to evaluate the integral in (2.10). For this, we consider a grid of values of x of length 500 taken from support of the density f.

(a) Comparison between Naive and Smooth estimator

We consider $K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$, $h = 0.3$ fixed, and $n = 500$.

Figure 2.1 displays the true density function under `model 1`, $\widehat{f}_{n,0}$ and $\widehat{f}_n$. It can be seen that the smooth estimator fits better the true density than the unsmooth one. In addition, by comparing the `ISE` values, $\widehat{f}_n$ has a better performance in terms of minimizing the estimation error. Indeed, for `model 1`, $\texttt{ISE}(\widehat{f}_{n,0}) = 0.0003$ (resp. 1.0248 for `model 2`) and $\texttt{ISE}(\widehat{f}_n) = 0.0001$ (resp. 0.8675 for `model 2`).



(a) (b)

Figure 2.1. (a) Estimation of the density under `model 1`. (b) Estimation of the density under `model 2`.

(b) Discussion on the choice of the kernel

Here we are interested in discussing the effect of the kernel on the estimation. Note that if X_i falls in the interval $x \pm h$, then it is clear that the distance $|X_i - x|$ does not exceed h. i.e. $|X_i - x| \leq h$ and hence $-1 \leq \dfrac{X_i - x}{h} \leq 1$. Therefore, K has to be defined on the interval $(-1, 1)$. Also, K needs to be a density (integrates to 1) and it usually has a symmetrical shape maximized at 0. In practice there are several examples of functions satisfying the above conditions:

- Gaussian Kernel: $K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right) \quad \forall u \in \mathbb{R}.$
- Epanechnikov: $K(u) = \frac{3}{4}(1-u^2)\mathbb{1}_{\{|u|\leq 1\}}.$
- Uniform: $K(u) = \frac{1}{2}\mathbb{1}_{\{|u|<=1\}}.$
- Triangular: $K(u) = (1-|u|)\mathbb{1}_{\{|u|\leq 1\}}.$
- Biweight: $K(u) = \frac{15}{16}(1-u^2)\mathbb{1}_{\{|u|\leq 1\}}.$
- Cosine: $K(u) = \frac{1}{2}(1+\cos(\pi u))\mathbb{1}_{\{|u|\leq 1\}}.$

Figure 2.2 displays the kernels graphically.

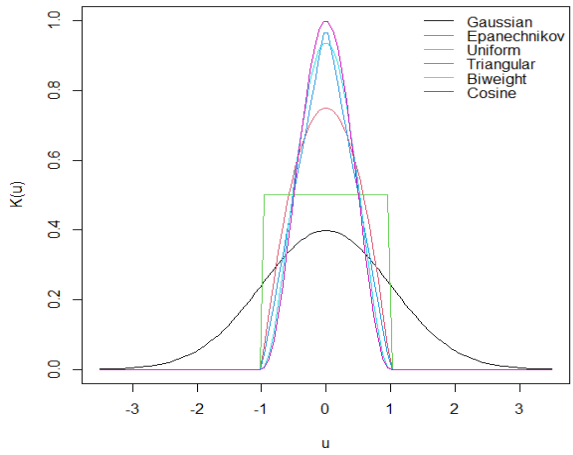
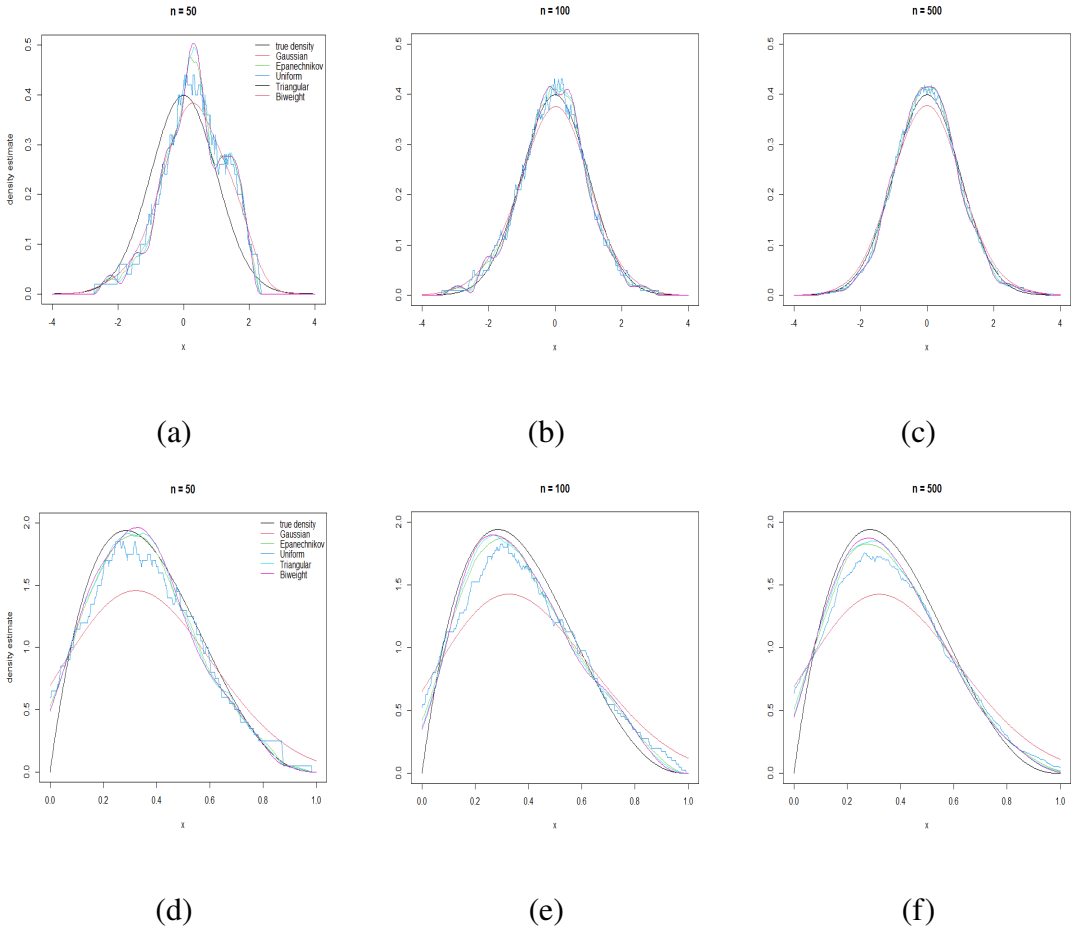


Figure 2.2. Examples of kernels.

In the literature, the choice of the kernel does not have a significant effect on the quality of estimation. To show this, we compare them by running a simulation that is based on random data generated from `model 1` and `model 2`. Figure 2.3 shows six plots of the two models for three different sample sizes. This time, our comparison is based on the MISE criterion, which is defined as:

$$\texttt{MISE}(\tilde{f}) = \mathbb{E}\int \left(f(x) - \tilde{f}(x)\right)^2 dx \tag{2.11}$$

The results are shown in Table 2.1 for both `model 1` and `model 2`. As n increases, the `MISE` values become closer and closer to each other, meaning that the effect of the kernel becomes negligible. Moreover, by comparing the results of `model 1` and `model 2`, it is notable that the kernel estimator acts more effectively with symmetric distributions in terms of minimizing errors.



(a) (b) (c)

(d) (e) (f)

Figure 2.3. Top 3 graphs: Estimation of the density under `model 1` with sample size $n = 50, 100$ and 500, respectively. Bottom 3 graphs: Estimation of the density under `model 2`.

Table 2.1. MISE and Median ISE values for each kernel under model 1 and model 2

		Model 1		Model 2	
		MISE	Median ISE	MISE	Median ISE
$n = 50$	Gaussian	0.00109	0.00087	0.10340	0.09874
	Epanechnikov	0.00114	0.00092	0.10369	0.09886
	Uniform	0.00105	0.00085	0.10271	0.09792
	Triangular	0.00111	0.00088	0 .10286	0.09853
	Biweight	0.00111	0.00036	0.09698	0.09799
$n = 100$	Gaussian	0.00072	0.00060	0.09967	0.09911
	Epanechnikov	0.00070	0.00057	0.09955	0.09769
	Uniform	0.00070	0.00058	0.09971	0.09748
	Triangular	0.00071	0.00058	0.09945	0.09756
	Biweight	0.00068	0.00055	0.10050	0.09852
$n = 500$	Gaussian	0.00038	0.00036	0.09710	0.09677
	Epanechnikov	0.00039	0.00038	0.09733	0.09722
	Uniform	0.00039	0.00037	0.09692	0.09648
	Triangular	0.00039	0.00036	0.09728	0.09680
	Biweight	0.00039	0.00036	0.09698	0.09696

(c) Discussion on the choice of the bandwidth: a Bias-Variance trade-off

In contrast to the kernel parameter, the bandwidth, h, has a significant effect on the quality of estimation. The choice of h is crucial since it controls the smoothness of the estimator. Large values of h reduce the variability of the estimator but increase its

bias (over-smoothing), while small values do the opposite (under-smoothing). Further, increasing h may lead to ignoring some important features of the true distribution, while decreasing it may lead to capturing unnecessary details (see, for example, Silverman 1952, p.15). Hence, it is important to build a criterion that considers a trade-off between the bias and the variance of the estimator. Note that the Mean Square Error (`MSE`) defined as follows is the most common criterion used in the literature:

$$\begin{aligned} \texttt{MSE}(\widehat{f}_n) &= \mathbb{E}\left((\widehat{f}_n - f)^2\right) \\ &= (\texttt{bias}(\widehat{f}_n))^2 + \texttt{var}(\widehat{f}_n). \end{aligned} \tag{2.12}$$

The identification of the optimal bandwidth minimizing the `MSE` requires the calculation of the bias and the variance of the estimator. In such a case, the optimal h that minimizes the (asymptotic) `MSE` will depend on several unknown quantities such as the density itself. Therefore, one can consider a numerical approach to identify the optimal bandwidth by minimizing the empirical version of the `MSE` defined in (2.12). In the following, we introduce the so-called cross-validation approach to find a numerical approximation of the asymptotic optimal bandwidth. That is:

$$h_{opt} = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \left(\widehat{f}_n(X_i, h) - f(X_i)\right)^2, \tag{2.13}$$

where H is a set of possible values of h. To illustrate the principle of data-driven selection of the smoothing parameter, we consider a simulation study.

We generate 500 observations according to `model 1`, and we choose the Gaussian kernel to calculate $\widehat{f}_n$. Our purpose is to find the optimal bandwidth, h_{opt}, that minimizes the cross-validation criterion, defined in (2.13). Figure 2.4 shows the objective function

(2.13) and the location of optimum h. Moreover, Figure 2.5 shows how the choice of the bandwidth affects the quality of estimation. The middle graph represents the estimator using h_{opt}. Also, note that small and large values of h result in either under-smoothed or over-smoothed estimators.

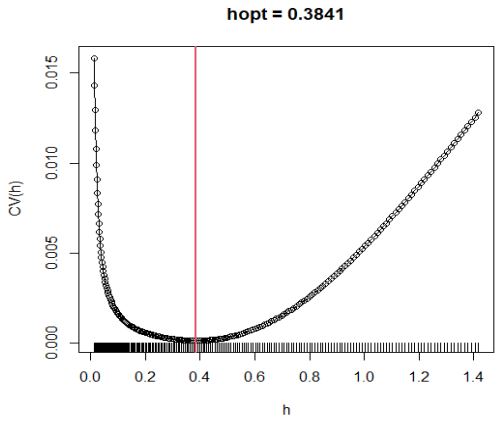


Figure 2.4. Optimal h according to cross-validation criterion.

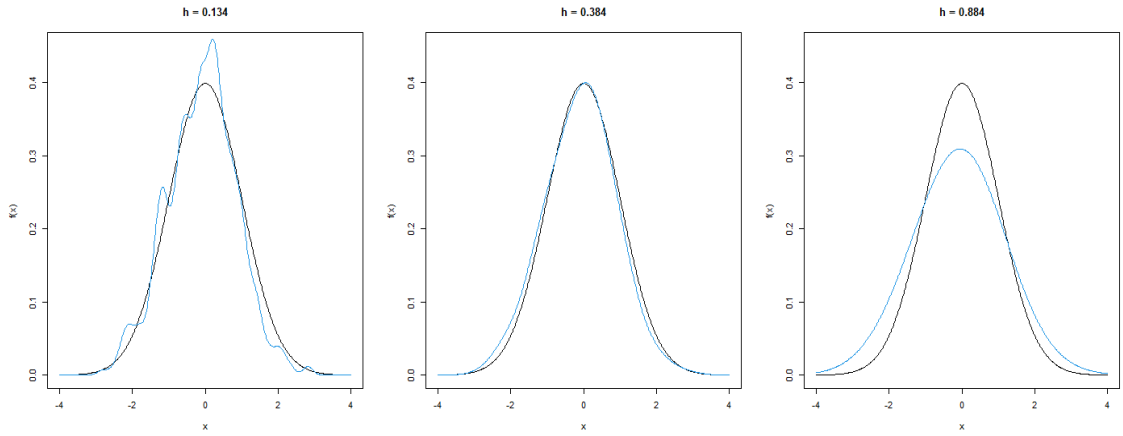


Figure 2.5. Effect of different choices of h.

Nonparametric regression estimation

In this section, we are interested in a nonlinear regression model of the form:

$$Y = m(\mathbf{X}) + \epsilon. \tag{2.14}$$

As we discussed above, under the assumption $\mathbb{E}(\epsilon|\mathbf{X}) = 0$, one gets

$$\begin{aligned} m(x) &= \mathbb{E}(Y \mid \mathbf{X} = x) \\ &= \int y f_{Y|\mathbf{X}}(y|x)dy \\ &= \int y \, \frac{f_{\mathbf{X}Y}(x,y)}{f_{\mathbf{X}}(x)}dy, \end{aligned}$$

where $f_{\mathbf{X}Y}$ is the joint probability density of $(\mathbf{X}, Y)$ and $f_{\mathbf{X}}$ is the marginal of $\mathbf{X}$. A nonparametric estimator of the f_X, at any fixed point x, is given in (2.9). Similarly, one can define a kernel-type estimator of $f_{\mathbf{X}Y}$. That is

$$\widehat{f}_{\mathbf{X}Y,n}(x, y; h) = \frac{1}{nh^2} \sum_{i=1}^{n} \prod_{j=1}^{p} K_j \left(\frac{X_j - x}{h} \right) H \left(\frac{Y_j - y}{h} \right), \tag{2.15}$$

where H is a kernel associated to the random variable Y. For simplicity, we consider here that $\mathbf{X}$ and Y have the same bandwidth h.

Finally, a plug-in estimator of m, at any point x, is obtained by replacing f_X and f_{XY} by their empirical version given in (2.9) and (2.15). Moreover, making use of some

calculus tools, one can show that

$$\widehat{m}_n(x) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)}, \tag{2.16}$$

where K and h are the kernel and the bandwidth, respectively. The technical details that lead to $\widehat{m}_n$ are given in the Appendix. The regression estimator given in (2.16) is called Nadaraya-Watson estimator (see [1] and [2])

One of the important properties that should be considered when studying any estimator is the convergence rate, which indicates how fast the estimator is reaching the true value, as the sample size increases. The rate of convergence depends on the sample size and the dimension of the covariate X. The effect of the dimesionality will be discussed in a later section.

Theorem 3 (Mean Square Convergence)**.**

$$n^{4/(d+4)}\ \mathbb{E}(\widehat{m}_n(x) - m(x))^2 \longrightarrow c$$

where d is the dimension of X and c is a constant (see Theorem 3.1 in page 70, [3] for more details about the expression of the constant).

Under some regularity conditions, and if we suppose that the optimal bandwidth is selected according to $h_n = c_n\ n^{-1/(d+4)}$, where $c_n \underset{n\longrightarrow\infty}{\longrightarrow} c$, then the above Theorem holds true. In words, one consequence of this Theorem is that asymptotically,

$$\mathbb{E}(\widehat{m}_n(x) - m(x))^2 = \frac{c}{n^{4/(d+4)}}.$$

This means that the convergence rate depends on the sample size n and the dimension of the data, d. Assume we fix n. Let $d = 1$, then the convergence rate is proportional to $\frac{c}{n^{4/5}}$. Now, if $d = 2$, then the convergence rate is proportional to $\frac{c}{n^{4/6}}$. It can be observed that the convergence rate decreases as the dimension d increases. This also applies to the bandwidth. Asymptotically, h_n is proportional to $n^{-1/(d+4)}$, and hence d causes the bandwidth to decrease as well.

Remark 2. *Since the constant c in Theorem 3 depends on several unknown parameters, such as the regression function m and the density f, it is not possible to find the optimal bandwidth based on the minimization of the asymptotic Mean Square Error.*

Now we use simulated data to illustrate the effect of

Data-driven choice of the bandwidth

The choice of optimal h is based on minimizing the cross-validation function. In other words,

$$h_{opt} = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{m}_{n,-i}(X_i))^2. \tag{2.17}$$

To illustrate how $\widehat{m}_n(x)$ is used in estimating non-linear models, we consider the quadratic function, which is defined as $m(x) = x^2$. Then, based on this function, we generate $n = 500$ observations through $(X_i, Y_i)_{i=1,\ldots,n}$ by

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

where $X \sim \text{Unif}(-3, 3)$ and $\epsilon \sim \mathcal{N}(0, 1)$. Figure 2.7 shows the random sample and the true curve.

In order to use (2.16) in estimation, we choose the Gaussian kernel for K, and we select the optimum h based on minimizing the cross-validation function, as shown in Figure 2.6. To illustrate the influence of h on the quality of estimation, we consider three values of the bandwidth: the optimal one and two values below and above. The three plots are shown in Figure 2.7. It can be observed that small values of h result in unbiased but inconsistent estimators (under-smoothing, left graph), and large values result in consistent but biased estimators (over-smoothed, right graph), while the optimal h makes a balance between these two features (middle graph).

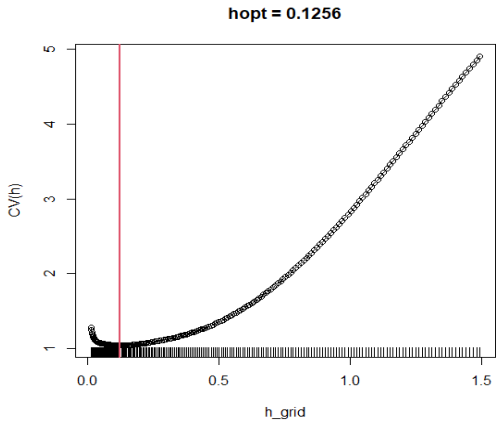


Figure 2.6. Optimal h that minimizes the CV function.

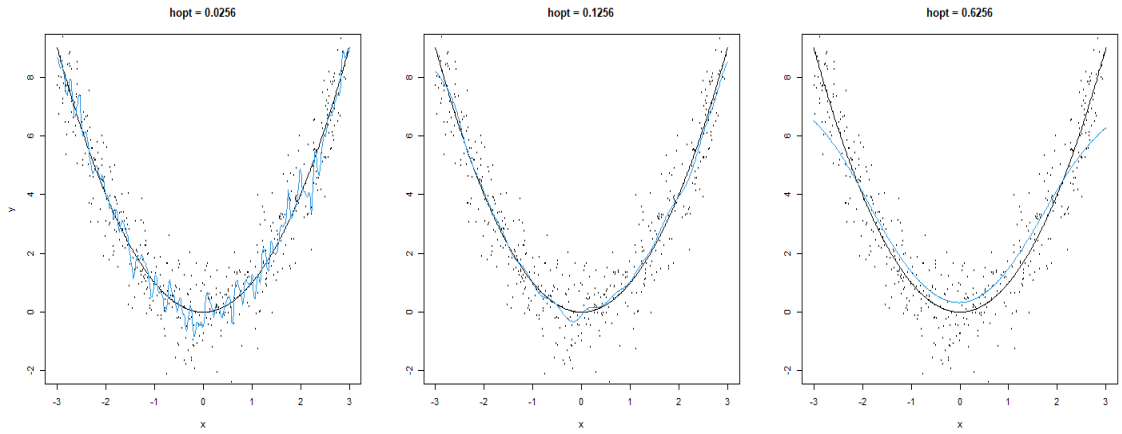


Figure 2.7. The effect of h on the estimated regression function.

Let us now consider the bivariate case where $\mathbf{X}$ is two-dimensional. In this

simulation, we generate X_1 and X_2 from the Uniform distribution with parameters -1.5 and 1.5. Then, Y is generated by

$$Y = m(x_1, x_2) + \epsilon.$$

where $\epsilon \sim N(0, 1)$. We define $m(\cdot)$ using the following two models:

`model 1`: $m(x_1, x_2) = \cos(2x_1) + \cos(2x_2) + 4$

`model 2`: $m(x_1, x_2) = x_1 + x_2 + 5$

To see the effect of the sample size, we generate two samples for each model, which are of sizes 100 and 300. The results are visualized in Figure 2.8 and Figure 2.9. The optimal bandwidth values for the first model are 0.2930 and 0.2346, while for the second model 0.3978 and 0.3324, for the two samples, respectively. It can be noticed from the figures that increasing the sample size enhanced the estimation for the two models.

The previous two examples illustrated the effect of the bandwidth and the sample size on the accuracy of estimation. Previously we introduced the difficulty of controlling model accuracy as it contains more predictors. The coming section aims to illustrate that with a simulation study.

Curse of dimensionality

As discussed above, the quality of estimation is affected by the choice of the bandwidth, the sample size, and the dimension of the covariate. This section focuses on the effect of the dimension. For a fixed sample size $n = 300$, three random samples $(X_i, Y_i)_{i=1,\ldots,n}$ are generated such that:

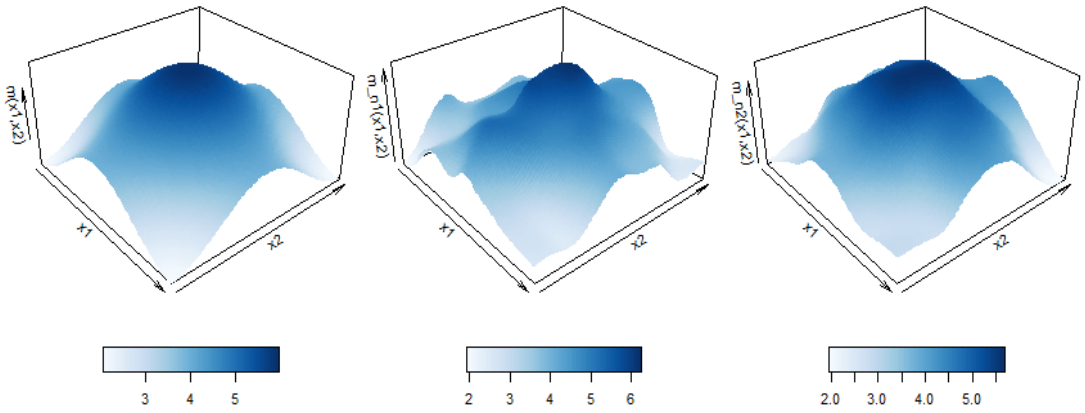


Figure 2.8. Graphical illustration of the effect of the sample size. Leftmost: The true regression function. Middle: An estimation of the regression function using a sample of size 100. Rightmost: An estimation of the regression function using a sample of size 300.

- For the first sample, $\mathbf{X} \sim N(0, 1)$
- For the second sample, $\mathbf{X}$ is generated from a Normal distribution with mean $\mu = \mathbf{0}$, $\sigma_1^2 = \sigma_2^2 = 1$ and $\rho_{12} = -0.7$
- For the third sample, $\mathbf{X}$ is generated from a Normal distribution with $\mu = \mathbf{0}$, $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 1$ and $\rho_{12} = \rho_{23} = -0.7$, $\rho_{13} = 0.5$

and Y_i is generated by

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

For the three models, we consider ϵ_i to be generated from the standard Normal distribution. Estimations are obtained by $\widehat{m}_n(x)$ and the MSE values are recorded. The values

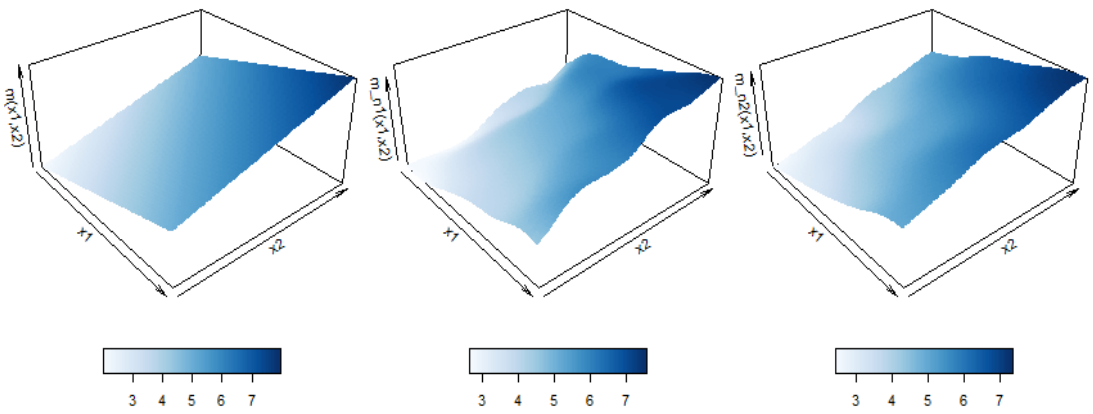


Figure 2.9. Graphical illustration of the effect of the sample size. Leftmost: The true regression function. Middle: An estimation of the regression function using a sample of size 100. Rightmost: An estimation of the regression function using a sample of size 300.

are 0.0266, 0.1083, 0.1769, for the three models, respectively. Note that increasing the dimensionality causes the estimated errors to increase. Moreover, the mean bandwidth values for each of the three models are: 0.17, 0.23, 0.26. Again, it can be noted that the optimal bandwidth increases as d is increased.

CHAPTER 3: REGRESSION ESTIMATION WITH MASSIVE DATA

In this chapter, we discuss the recursive estimation of the regression function. This estimator is designed to "properly" update the last value of the regression function, calculated at a fixed point x, as the sample size increases. The main advantage of this estimator is that its calculation does not necessarily requires any storage of the data which makes it adapted for the data streaming context. Moreover, it also can be used when we have to estimate the regression function when a massive database is available. In both cases, only a reasonably limited number of observations are needed to calculate the initial value, then we keep updating the estimate as the sample size grows.

Stochastic Approximation

The online regression estimation is inspired by the idea of stochastic approximation. Stochastic approximation is a class of recursive methods to find the zero root or the optimum of a function via noisy observations. In the last decades, stochastic approximation has been widely applied in many areas, such as signal processing, control theory and pattern recognition.

In practice, we often come across root-seeking problems such as in regression analysis where the purpose is to understand how a response variable is concomitant with some predictors. The regression function is often identified as the solution of a certain minimization problem. Thus a zero of a certain estimating equation (under some smoothness of the objective function to be minimized). The least square estimator and maximum likelihood estimator are examples of estimators that might be written as a zero of a certain estimating equation.

In general, if one desires to find the zero root x_0 of some known function $G(x)$

such that $G(x_0) = 0$, then the Newton-Raphson method is an alternative given by

$$x_{n+1} = x_n - \frac{G(x_n)}{G'(x_n)}. \tag{3.1}$$

However, in some situations the form of $G(x)$ is unknown and only noisy observations of $G(x)$ are available via

$$y_n = G(x_n) + \eta_n,$$

where η_n's are observational errors. Then the question is how to find the zero of $G(x)$ via the observations $\{y_n\}$. [4] proposed the recursive procedure (also known as the Robbins-Monro algorithm) to fulfill this work

$$x_{n+1} = x_n + a_n y_n, \tag{3.2}$$

where the step size $a_n > 0$ satisfies $\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$. For example, one can take $a_n = 1/n$ to satisfy both conditions. Comparing (3.1) and (3.2), it is apparent that the Robbins-Monro algorithm can be regarded as a kind of nonparametric method which does not leave the form of the function $G(x)$ to be specified. Inspired by [4] work, [5] proposed a recursive method to find the maximum of a function. Note that the continuous function $G(x)$ reaches the maximum at the point x_0 when $G'(x) = 0$. Therefore, the optimization problem is also equivalent to finding the root of $G'(x) = 0$. The authors proposed the recursive procedure to find the maximum of $G(x)$ in the case that there are only observations of $G'(x)$ available.

Recursive estimator

It can be shown that the problem of root seeking is closely related to nonparametric estimation. In order to estimate $m(x)$ in (2.6), let $r(y) = f(x)y - f(x)m(x)$ where $f(x)$ is the density function of X, then one can estimate $m(x)$ by finding the root y_0 such that $r(y_0) = 0$.

Given a random sample $(X_1, Y_1), \ldots, (X_n, Y_n)$, [6] proposed a recursive method to estimate $m(x)$ given by

$$\widehat{m}_n(x) = \widehat{m}_{n-1}(x) + \frac{1}{n} K\left(\frac{\|X_n - x\|}{h_n}\right)(Y_n - \widehat{m}_{n-1}(x)), \tag{3.3}$$

where one can consider $\widehat{m}_0(x) = 0$. Other choices for the initial value are also possible.

Remark 3. *Note that equation* (3.3) *could be called a totally recursive procedure since only one formula is used to estimate* $m(x)$.

A class of Robbins-Monro estimators

In this project, we consider a large class of Robbins-Monro estimators of the regression function. For this, we write (3.3) for any step-size, say θ_n. That is:

$$\widehat{m}_n(x) = \widehat{m}_{n-1}(x) + \theta_n (Y_n - \widehat{m}_{n-1}(x)), \tag{3.4}$$

where $(\theta_n)_n$ is a sequence of positive numbers tending to zero as n goes to infinity.

Several step size choices could be considered in practice. Some of them will make the calculation of $\widehat{m}_n(x)$ faster than others. Below, we give some examples of θ_n:

1. $\theta_n = 1/n$

2. $\theta_n = \Delta_n / \sum_{i=1}^{n} \Delta_i$, where $\Delta_n := K(\|X_n - x\|/h_n)$.

3. $\theta_n = \Delta_n h_n / \sum_{i=1}^{n} \Delta_i h_i$

4. $\theta_n = \Delta_n / n h_n$

5. $\theta_n = \dfrac{\gamma_n}{h_n} \cdot K\left(\dfrac{\|X_n - x\|}{h_n}\right)$

Note that some choices of the step size involve the selection of the optimal bandwidth as is the case in cases 2-5 above. Note that the bandwidth depends on the iteration number n. This means that we have to select an optimal bandwidth at each iteration which will make the total computation time slower. Since the purpose of using the recursive estimator is to reduce the computation time, we prefer to avoid these choices of the step size. The fifth form involves two quantities that depend on the sample size n. However, without losing the consistency of the estimator, Cardot(2012) mentioned that the following assumptions can be imposed:

$$\gamma_n = \frac{c_\gamma}{n^\gamma} \qquad\qquad h_n = \frac{c_h}{n^h}.$$

Hence, the step size becomes

$$\theta_n = c_\gamma \cdot K(n^{\frac{1}{d+4}} \cdot \|X_n - x\|)/n^{\frac{d+3}{d+4}}, \tag{3.5}$$

where c_γ is optimally selected by the cross-validation technique. The quantity c_γ does not depend on the sample size, and hence it is chosen only once. Moreover, note that choosing an h_n proportional to the sample size is also considered for simplicity reasons and is also justified theoretically as this is the order for an offline estimator.

On the consistency of the Robbins-Monro estimator

When comparing the quality of estimators, it is of interest to examine two important properties, the accuracy and the computation time. Intuitively, as the sample size increases, both of them increase as well. However, we prefer using estimators that is not time-consuming but provide reasonably accurate estimations. There is a trade-off between these two properties. Thus, we will use simulated data to study each of them. For a sequence of different sample sizes that ranges between 30 and 19500, we simulate $X_1, X_2 \sim \text{Unif}(-1.5, 1.5)$, and

$$Y = m(x_1, x_2) + \epsilon,$$

where $m(x_1, x_2) = \cos(2x_1) + \cos(2x_2) + 4$ and $\epsilon \sim \mathcal{N}(0, 1)$.

Then, we use the offline estimator and the online estimator to estimate the function m at the point $(0, 0)$. For the online estimator, we considered the following choices of the step size: $\theta_n = 1/n$ and $\theta_n = c_\gamma \cdot K(n^{\frac{1}{d+4}} \cdot \|X_n - x\|)/n^{\frac{d+3}{d+4}}$.

Also, for the second form of θ_n, we considered different choices of c_γ, which are 0.1 and 1. The results are shown in Figures 3.1 and 3.2.

It can be seen that the offline estimator converges to the true values as the sample size increases, while for the online estimator with $\theta_n = 1/n$, the estimator deviates away from the true value. Moreover, considering the second choice of the step size for the online estimator, the second two plots show that the online estimator is sensitive to the choice of c_γ.

It is worth mentioning that the optimal choice of c_γ depends on the data. This quantity has to be chosen by the cross-validation technique and by considering a sequence

of c_γ that depends on the sample size.

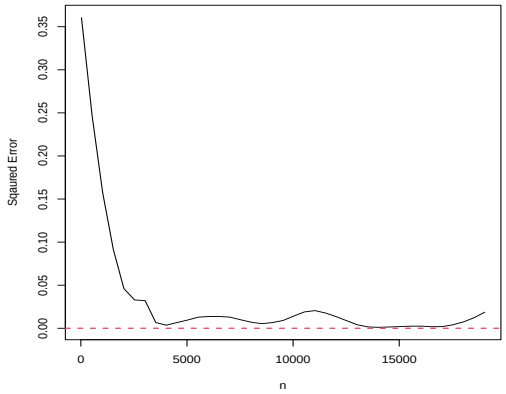
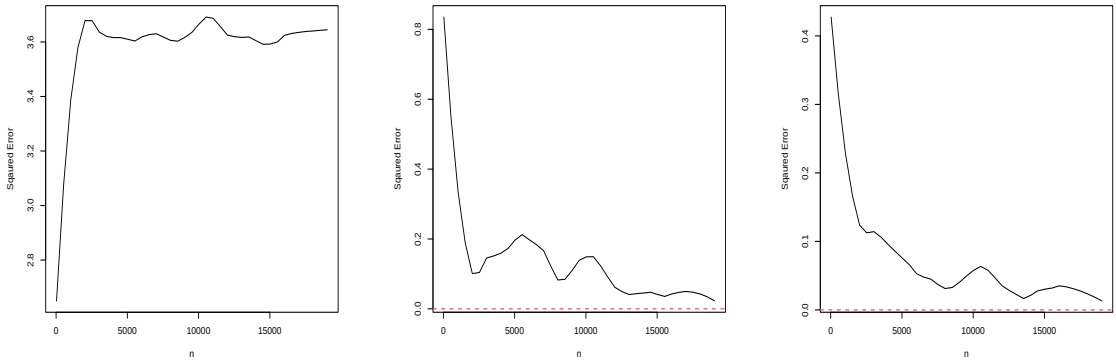


Figure 3.1. The convergence rate of the offline estimator.



(a) (b) (c)

Figure 3.2. (a) The convergence rate for the online estimator using the first choice of θ_n. (b) and (c) The convergence rates for the online estimator using the second choice of θ_n with $c_\gamma = 0.1$ and $c_\gamma = 1$, respectively.

Computation complexity and comparison with the offline estimator

In the preceding section, we compared the consistency of the two estimators using simulated data. In this section, we compare them in terms of the computation time. In the univariate setting, a random sample $(X_i, Y_i)_{i=1,\ldots,5000}$ was generated such that: $X \sim N(0,1)$ and $Y = m(X) + \epsilon$ where $\epsilon \sim N(0,1)$ and m is defined by

$m(x) = x^2$. Consider the case where we have online data that is accumulating over time. Whenever a new observation is received, a new point estimation is calculated using both estimators. The nonrecursive estimator re-evaluates $\hat{m}(x)$ using the whole available sample, including the new observation. However, the recursive estimator updates the previous estimation using only the new data point. The computation time taken by each estimator is recorded in seconds. Figure 3.3 shows the cumulative time of computation for both estimators. There are two notable differences between the plots, the range of values of the computation time, and the growth rate. By the time we reached the last observation, the total time consumed by the nonrecursive estimator exceeded 20 seconds, while it did not reach one second for the recursive one. Moreover, the exponential shape of the first plot indicates that the running time of the non-recursive estimator increases much faster than that of the recursive one.

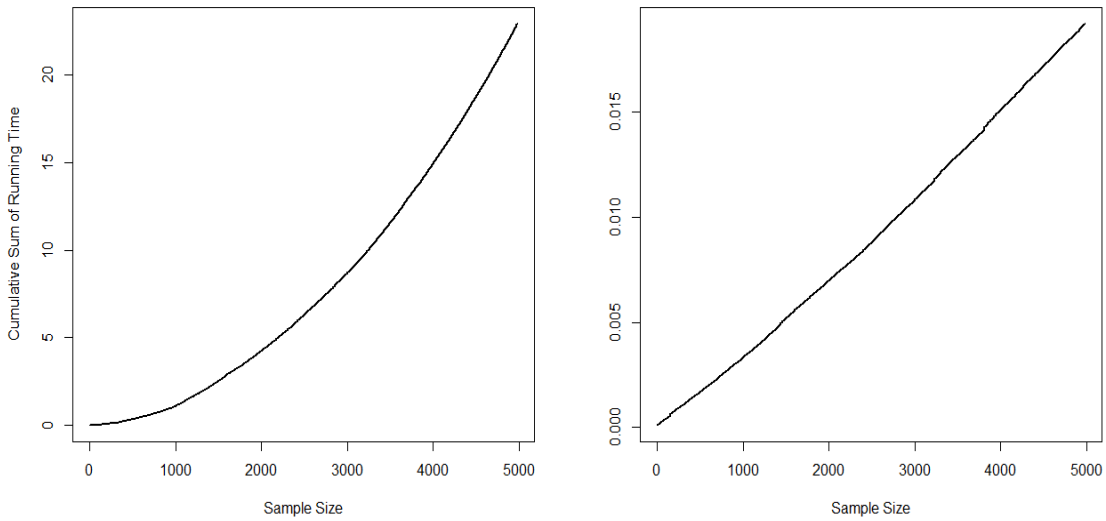


Figure 3.3. Cumulative computation time for the nonrecursive estimator (left). Cumulative computation time for the recursive estimator (right).

CHAPTER 4: OFFLINE AND ONLINE SUPERVISED LEARNING

In chapter 3 we discussed the estimation of the regression function when the response variable is continuous. The Robbins-Monro estimator can be used for example to predict unknown values of a response given a certain number of predictors. It can also be used for online time series forecasting. In some real-life problems, the response variable is not continuous but rather categorical. The problem then becomes a supervised classification problem. This chapter aims to discuss this case.

Offline supervised learning

In the previous chapter, we discussed the construction of the Nadaraya-Watson estimator, defined as

$$\hat{m}_n(x) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{\mathbf{X}_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\mathbf{X}_i - x}{h}\right)},$$

where K and h are the kernel and the bandwidth, respectively. This formula implies that an estimation of m at a fixed point x can be obtained by taking the weighted average of Y. Closer the distance between x and $\mathbf{X}_i$, higher the weight assigned to Y_i.

In this section, we discuss the adaptation of this estimator to the case of supervised learning. Consider Y to be a categorical variable that takes one of G possible classes, and $\mathbf{X}$ is a vector of d independent random variables. In order to predict the class of Y_{new}, we need to find whether $Y_{new} = g$ is true for each of the G possible classes. This means the response variable we need to predict is $\mathbb{1}_{\{Y_{new}=g\}}$. As discussed before, our regression model 2.14 predicts the response based on the expectation. Hence, our aim now is to estimate $\mathbb{E}(\mathbb{1}_{Y_{new}=g})$. Note that the indicator function $\mathbb{1}$ evaluates to 1 if the condition $Y_{new} = g$ is true, and takes 0 otherwise. Hence, the quantity $\mathbb{1}_{\{Y_{new}=g\}}$ can be

looked upon as a Bernoulli random variable with $\mathbb{P}(Y_{new} = g)$ being the probability of success, which is also the expectation. In other words,

$$\mathbb{P}(Y_{new} = g|X = X_{new}) = \mathbb{E}(\mathbb{1}_{\{Y_{new}=g\}}|X = X_{new}). \tag{4.1}$$

In the case of using the non-recursive estimator, this probability is estimated by

$$\widehat{\mathbb{P}}_{g,n}(\mathbf{X}_{new}) = \frac{\sum_{i=1}^{n} \mathbb{1}_{\{Y_i=g\}} K\left(\frac{\|X_i - \mathbf{X}_{new}\|}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{\|X_i - \mathbf{X}_{new}\|}{h}\right)}, \tag{4.2}$$

where K and h are the kernel and the bandwidth, respectively. Note that because of the indicator function, only the features of the specified class will contribute to the calculation of $\widehat{\mathbb{P}}_{g,n}(\mathbf{X}_{new})$. Assume that we are given $\mathbf{X}_{new}$, and the true class is j. To predict Y_{new}, 4.2 is calculated for each possible class. When it comes to calculating $\widehat{\mathbb{P}}_{j,n}(\mathbf{X}_{new})$, the distance $\|X_i - \mathbf{X}_{new}\|$ will decrease whenever Y_i satisfies $Y_i = j$. This is because items of the same class are expected to have similar features. As as a result, the estimated probability, $\widehat{\mathbb{P}}_{j,n}(\mathbf{X}_{new})$, will be higher. However, for all other classes, the distances $\|X_i - \mathbf{X}_{new}\|$ will increase, and hence the estimated probability will be lower.

Therefore, the predicted class of Y_{new} will be the one corresponding to

$$\max_{g\in\{1,\ldots,G\}} \widehat{\mathbb{P}}_{g,n}(\mathbf{X}_{new}).$$

Online supervised learning

In the case of using the recursive estimator for the purpose of classification, equation 3.4 is re-written as

$$\widehat{\mathbb{P}}_{g,n}(\mathbf{X}_{new}) = \widehat{\mathbb{P}}_{g,n-1}(\mathbf{X}_{new}) + \theta_n[\mathbb{1}_{\{Y_i=g\}} - \widehat{\mathbb{P}}_{g,n}(\mathbf{X}_{new})], \tag{4.3}$$

where θ_n is called the step size can take any of the forms mentioned in section 3.2. Note that the initial value is calculated based on the offline estimator in equation 4.2.

CHAPTER 5: SOME PREDICTIVE MODELS IN BIG DATA SETTING

In this chapter we provide three applications of the estimators introduced in chapter 3 and 4. The first application aims to predict the class of dates in real-time given some dimension/size/shape, color and texture predictors. This application is of great interest in food-quality assessment. The second one is medical which consists in real-time monitoring of fetal well-being during pregnancy. Given a certain number of medical measures, the proposed model allows to assist the doctor to decide whether the fetal state is normal, suspicious or pathological. Finally, we apply the proposed Robbins-Monro estimator to predict the value of a real estate in Qatar in real-time.

Application to supervised dates data classification

In the previous chapters, we discussed the recursive and non-recursive methods for estimating the non-linear regression model

$$Y = m(X) + \epsilon.$$

Also, we illustrated by simulation the trade-off between the accuracy and the computation time. In this chapter, we are going to apply these methods of estimation using real data. Again, we will use the results to compare the estimators based on their accuracy and time of computation.

The first application is concerned with fruit classification. Particularly, we are interested in training our model to detect different date varieties based on their characteristics. Our dataset consists of 150 observations on 18 variables, where one of the variables identifies the date variety. There are five classes of dates in this sample, which are: Khalas, Fardh, Lulu, Segai, and Majdoul. Each class has 30 observations. Table 5.1

shows a summary of the quantitative variables.

Data Description and Exploration

Table 5.1. Descriptive statistics about the dates variables

Variables	Summary statistics					
	Min.	Q25	Median	Mean	Q75	Max.
L	2.661	2.066	3.270	3.311	3.507	4.284
D1	1.024	1.506	1.657	1.656	1.780	2.217
D2	1.678	2.000	2.127	2.138	2.280	2.773
D3	0.797	1.097	1.256	1.266	1.395	1.858
Area	3.918	5.052	5.632	5.704	6.193	8.442
Perim.	7.519	8.804	9.182	9.264	9.703	11.477
Circ	0.6930	0.8090	0.8370	0.8316	0.8570	0.9300
Skew	-0.2700	0.7272	1.0335	1.0235	1.2760	1.9230
AR	1.215	1.456	1.593	1.585	1.700	2.002
Solidity	0.9480	0.9740	0.9800	0.9783	0.9840	0.9920
Red	62.34	79.11	99.68	100.76	118.96	164.35
Green	43.39	53.76	58.75	61.22	68.29	103.06
Blue	38.84	48.49	51.90	51.60	54.85	61.29
IDM	2.12E-4	2.922E-4	3.34E-4	3.451E-4	3.808E-4	6.89E-4
Entropy	0.4300	0.4833	0.5000	0.4975	0.5100	0.5400
SumofallGLCM	6.551	6.874	6.971	6.974	7.069	7.399

In this dataset, there is a total of 17 predictive continuous variables. As discussed previously, using a high number of features has a negative effect on the accuracy of prediction. For that reason, it is important to reduce the dimensionality by summarizing the information in a smaller set of variables. One way of achieving that is by performing the Principal Component Analysis (PCA). The PCA is a factor extraction technique that produces a new set of uncorrelated variables, called factors, that are linear combinations of the original variables. The PCA works well when the features are strongly correlated. Hence, it is reasonable to look at the correlation matrix at the first step.

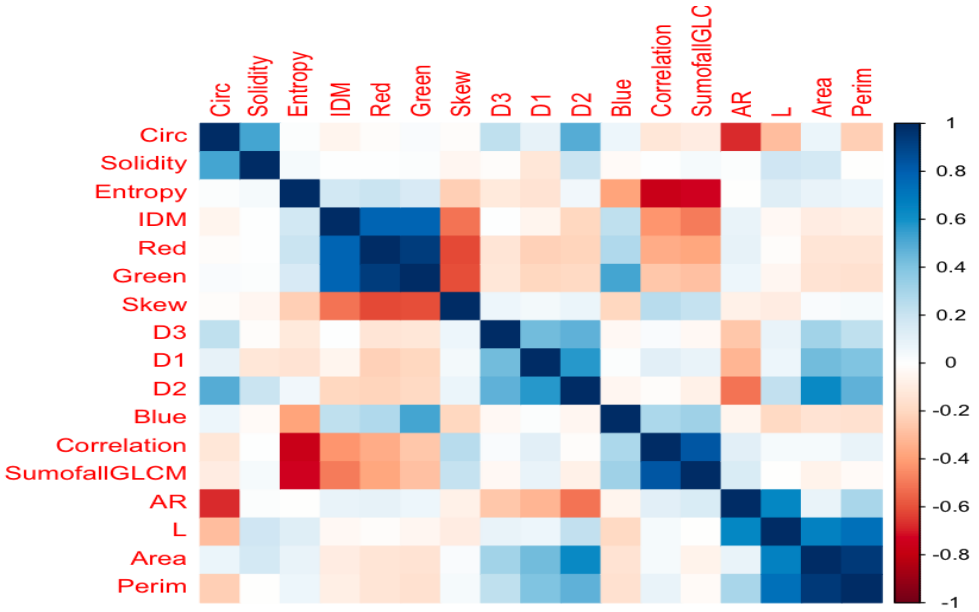


Figure 5.1. Correlation plot of the quantitative variables in the Dates dataset

Figure 5.1 visualizes the pairwise correlations of the numerical variables. Strong correlations are identified by the dark blue color if the correlation is positive, and by dark red if it is negative. It can be seen that some variables are strongly correlated with each other. For example, Area has a strong positive correlation with Perim, while AR has a strong negative correlation with Circ. The presence of highly correlated variables results in multicollinearity, which reduces the accuracy of prediction. This,

again, suggests using uncorrelated factors instead of the observed variables.

The Scree plot in Figure 5.4 shows that most of the variability can be explained by the first five principal components. More precisely, these components capture 70 to 75 percent of the total information.

Figure 5.2 shows the first two principal components on the two axes, along with the original variables represented by arrows. The angle between an arrow and a principal component indicates the direction of the corresponding variable with respect to that component, where the strength of its contribution is depicted by the length of that arrow.

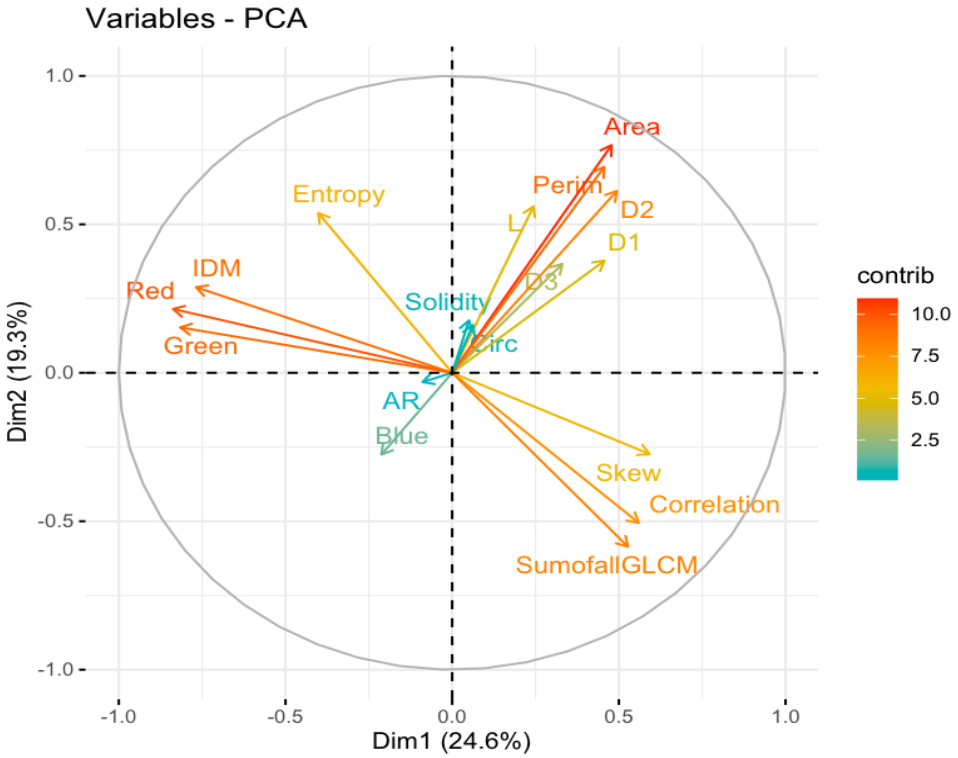


Figure 5.2. The contribution of each quantitative predictor to the first two dimensions.

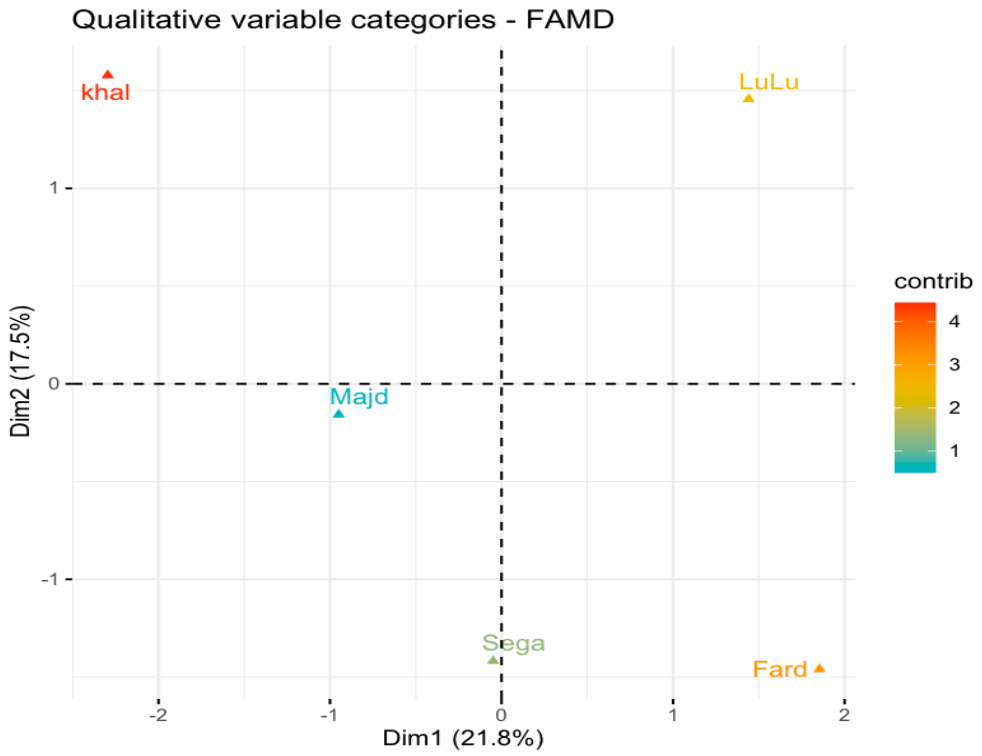


Figure 5.3

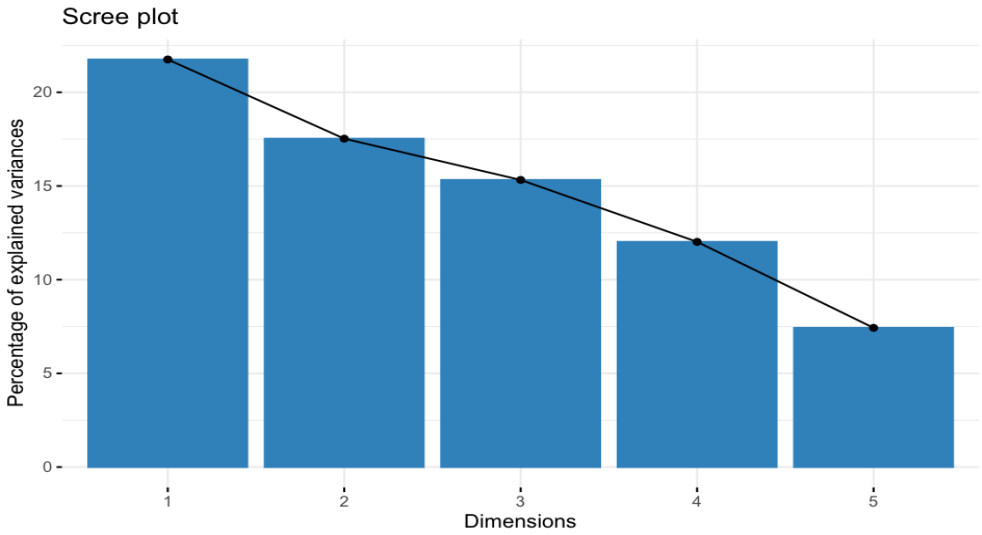


Figure 5.4. The Screeplot of the first five principal components.

In addition to using the recursive (RM) and non-recursive (NW) estimators, other classifiers will be used for comparison purposes. These classifiers are: Linear Discrimination Analysis (LDA), Quadratic Discrimination Analysis (QDA), and K-

Nearest Neighbors (KNN).

The LDA and QDA are two similar classification techniques that are based on maximizing the variance between classes. Both of these methods assume that the underlying distribution of each class is a multivariate Gaussian distribution. Additionally, the LDA is a special of the QDA since it assumes that the covariance matrices of these distributions are equal.

Given a vector of d features, $\mathbf{X}_{new}$, the predicted class of this vector is the one corresponding to:

$$\max_{g\in\{1,\dots,G\}} \widehat{\mathbb{P}}(Y=g|\mathbf{X}_{new}) = \max_{g\in\{1,\dots,G\}} \frac{\pi_g f_g(\mathbf{X}_{new})}{\sum_{k=1}^{G}\pi_k f_k(\mathbf{X}_{new})}, \tag{5.1}$$

where

- G is the number of classes.
- $\pi_g = \dfrac{\sum_{i=1}^{n} \mathbb{1}_{\{Y_i=g\}}}{n}$
- $f_g(\mathbf{X}_{new}) = \mathbb{P}(\mathbf{X}_{new}|Y=g)$

Note that $\widehat{\mathbb{P}}(Y=g|\mathbf{X}_{new})$ in equation 5.1 is computed using Bayes' Theorem.

The QDA relaxes the assumption of equal covariance matrices, and hence it is more flexible that the LDA. However, this also means that it requires estimating more parameters, which is problematic if the sample is not large enough.

For the recursive classifier, we consider the following form of the step size:

$$\theta_n = c_\gamma \cdot \frac{K(n^{\frac{1}{d+4}} \cdot \|X_n - x\|)}{n^{\frac{d+3}{d+4}}}.$$

As we have discussed, c_γ plays an important role in the quality of the recursive estimator.

In order to find the optimal value of c_γ, we considered a sequence of 100 values that has a range between 0.1 and 500. The optimal one is selected based on minimizing the misspecification rate (MSR). Figure 5.5 shows that the minimum MSR occurs when $c_\gamma = 157.38$. To illustrate the effect of c_γ on the MSR, we select two additional values below and above the optimal value.

In order to test the accuracy of each of the five classifiers, we split the data into two sets, a training sample and a test sample. Since each date variety has 30 observations, we built the training sample by randomly selecting 24 observations from each variety, while the remaining observations are assigned to the test sample. The MSR and the computation time are recorded for each classifier. Table 5.2 shows the computation time in seconds. Moreover, in order to compare the classifiers in terms of bias and variance, this process is done 100 times, and the boxplots of the MSR are shown in Figure 5.6. Note that for the recursive estimator, the optimal c_γ results in the least MSE value but with reasonable variance compared to the other choices. This shows the trade-off between bias and variance. Moreover, it can be noticed that it outperforms all other estimators in terms of accuracy and variance

Table 5.2 shows the time required by each classifier. For the Robins-Monro classifier, the computation time does not differ much for the different values of c_γ. So, we report the time for only one case of c_γ. Note that the first three classifiers are very quick. This is because they are based on fitting a model and using that model directly for classification. On the other hand, the offline estimator computes the optimal bandwidth for each increase in the sample size. That is why it is a time-consuming classifier.

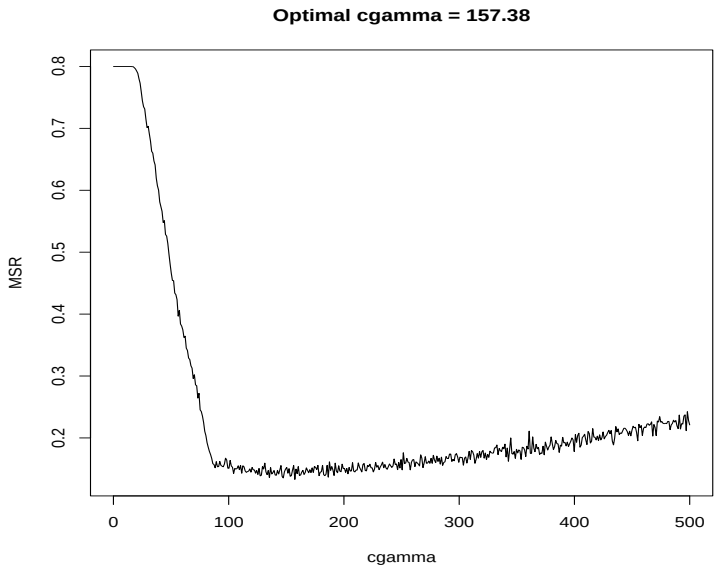


Figure 5.5. Optimal choice of c_γ.

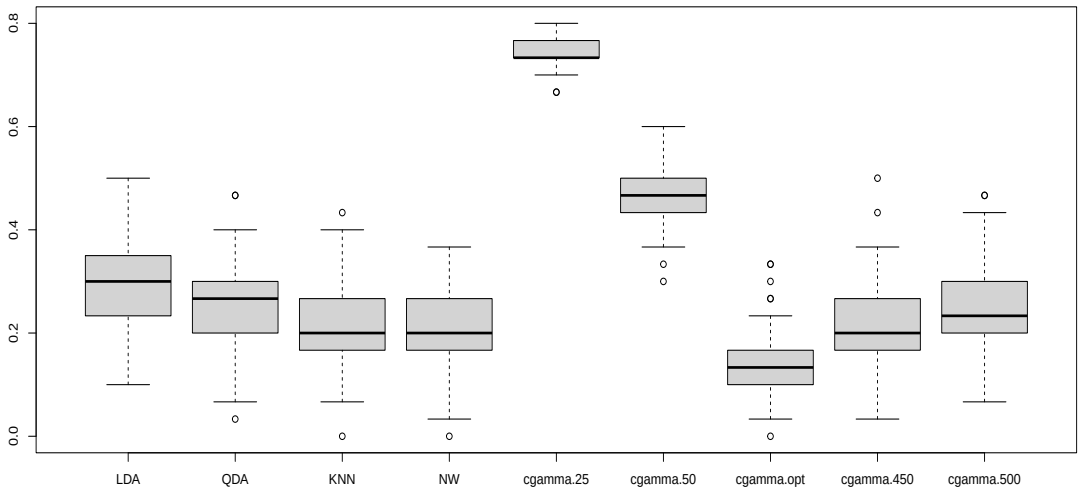


Figure 5.6. Boxplots of the (MisSpecification Rate) MSR of each classifier.

Table 5.2. The computation time in seconds of each classifier.

Classifier	LDA	QDA	KNN	RM, Optimal c_γ	NW
Computation Time	0.02	0.01	0	0.16	4274.83

Application to real-time monitoring of fetal well-being during pregnancy

Machine learning has made significant strides in obstetrics and improved the standard of care for expectant mothers and their babies. It is essential in the real-time monitoring of fetal health during pregnancy. Statistical learning algorithms can identify subtle patterns and deviations that may point to potential abnormalities or complications by analyzing extensive data collected from monitoring devices, such as fetal heart rate and maternal vital signs.

This early detection enables medical professionals to take immediate action, reducing risks and improving outcomes. Additionally, machine learning models incorporate various factors, such as demographic data, medical history, and real-time monitoring data, to provide personalized risk assessments, allowing for tailored insights and well-informed decision-making. These algorithms are also excellent at identifying preterm births in advance, using large datasets to identify contributing factors and provide early warning signs.

Additionally, machine learning is an effective tool for decision support, continuously analyzing monitoring data to offer healthcare providers insights and suggestions. In addition to improving prenatal care, this data-driven approach also stimulates obstetrics research and innovation, resulting in improved monitoring methods, fresh interventions, and a better understanding of fetal well-being.

Cardiotocography (CTG) is a technique that aims to collect a sequence of measurements of the fetal heartbeat along with uterine contractions. The purpose of this technique is to monitor fetal well-being during pregnancy and labor. Before or during birth, babies may suffer from Oxygen deprivation for multiple reasons. This lack of Oxygen affects the growth of the baby's organs and causes permanent damage to them.

Hence, it is necessary for medical doctors to collect such information in order to detect the issue earlier and take timely actions. The available dataset here consists of 2126 observations on 21 variables. One of the variables classifies the fetal state: Normal, Suspicious, or Pathological. Table 5.3 provide some information about the variables, and Table .5 (see Appendix) provides some descriptive statistics.

As we have discussed, dealing with high-dimensional data requires performing some dimensionality-reduction techniques to avoid the curse of dimensionality. Figure 5.7 shows that the variables exhibit some strong correlations, which suggests using the PCA to extract the latent factors. It can be seen from the Scree plot in Figure 5.8 that it suffices to use the first three principal components, which summarise 56.2% of the information in the data.

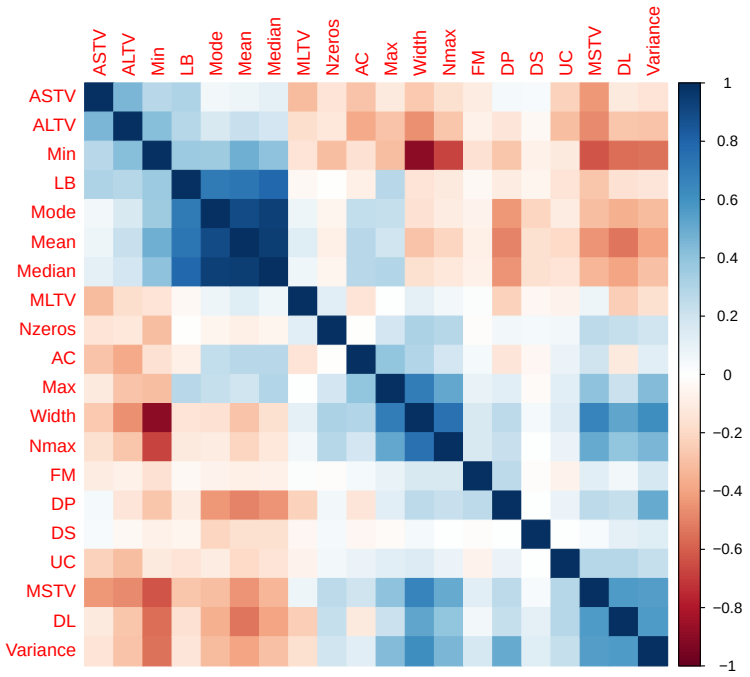


Figure 5.7. Correlation plot of the quantitative variables in the CTG data.

Table 5.3. CTG variables.

Features	Attribute Information
LB	FHR baseline (beats per minute)
AC	Number of accelerations per second
FM	Number of fetal movements per second
UC	Number of uterine contractions per second
DL	Number of light decelerations per second
DS	Number of severe decelerations per second
DP	Number of prolonged decelerations per second
ASTV	Percentage of time with abnormal short term variability
MSTV	Mean value of short term variability
ALTV	Percentage of time with abnormal long term variability
MLTV	Mean value of long term variability
Width	Width of FHR histogram
Min	Minimum of FHR histogram
Max	Maximum of FHR histogram
Nmax	Number of histogram peaks
Nzeros	Number of histogram zeros
Mode	Histogram mode
mean	Histogram mean
Median	Histogram median
Variance	Histogram variance
NSP	Fetal state class code (N=normal, S=suspect, P=pathologic)

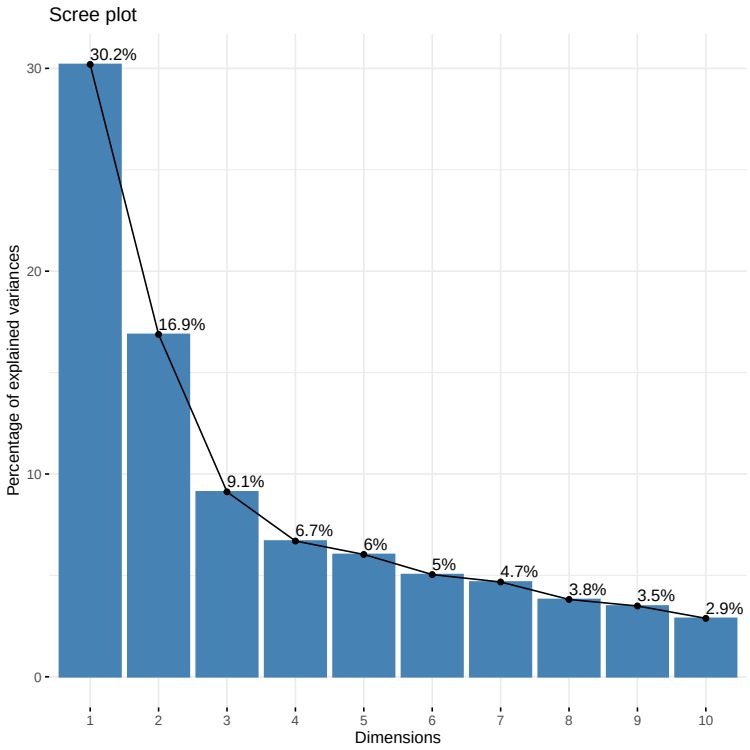


Figure 5.8. Scree Plot of the principal components of the CTG data.

We aim to assess the performance of the Robins-Monro estimator using two forms of the step size, which are: $\theta_n = 1/n$ and $\theta_n = c_\gamma \cdot K(n^{\frac{1}{d+4}} \cdot \|X_n - x\|)/n^{\frac{d+3}{d+4}}$, respectively. Note that these forms do not require the computation of the optimal bandwidth and hence are better in terms of reducing the computation time. Note that the second form requires the optimal choice of c_γ. For this, we defined a sequence of 50 values between 200 and 1000. Figure 5.9 shows that the optimal value is 673.47.

For this, the sample was partitioned into training and test samples according to the fetal state categories, which is illustrated in table 5.4.

Table 5.4. The partitioning of the sample by the NSP variable.

Fetal State	Normal	Suspicious	Pathological	Total
Training sample	1200	200	100	1500
Test sample	455	95	76	626
Total	1655	295	176	2126

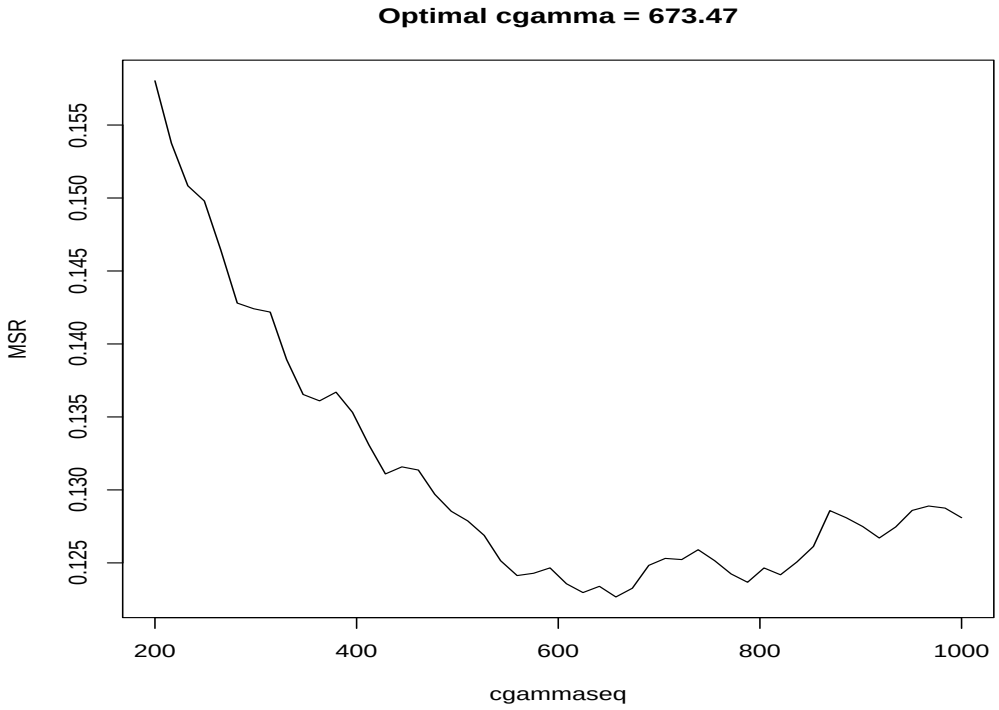


Figure 5.9. Optimal choice of c_γ for the CTG data.

Also, as we have done in the previous application, we aim to compare the performance Robins-Monro classifier with some parametric classifiers, which are the LDA and QDA, and the non-parametric KNN classifier. Figure 5.10 shows the boxplots of the misspecification rate of the classifiers. Robins-Monro classifier with the first choice of θ_n has the worst performance. However, with the second choice of θ_n, and by choosing the optimal c_γ, Robins-Monro performs better compared to the LDA and QDA.

However, unlike, the result of the previous application, its performance falls behind that of KNN.

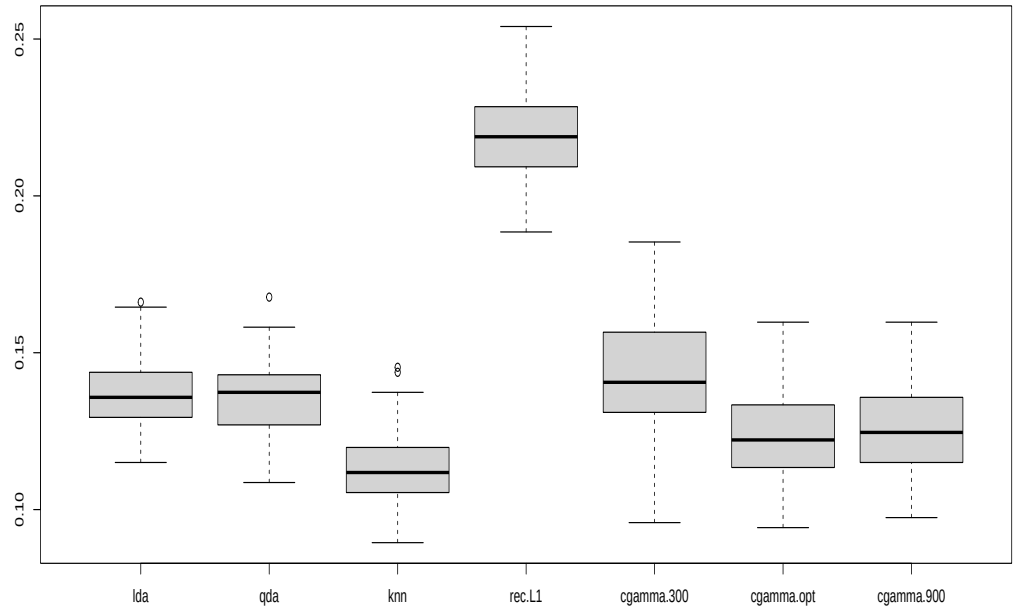


Figure 5.10. Boxplots of the misspecification rate (MSR) of each classifier.

Application to Real-Estate Value prediction in Qatar

Real estates are properties consisting of land and any permanent structures or improvements on the land, whether natural or man-made. There are multiple types of real estate, they could be residential, commercial, vacant lands, etc. Some people are interested in buying or investing in real estate. Therefore, it is important to assess whether a real estate is worth the money on the long term. The real estate value is defined to be the actual or estimated market value of any real property/building owned by the company/business applying for a certificate. There are several factors affecting the real estate value. These include the size of the real estate, its type, its location, etc. The real estate value is crucial since it helps the investor to take a decision on the investment.

In the dataset, we have two predictors, which are the area in square meters and the price per square foot. Their correlations with the real estate value are 0.45 and 0.62, respectively. However, the product of the area and the price is almost perfectly correlated with the real estate value, and hence will be used in the predictive model. Moreover, there are two categorical variables, which are the municipality and the real estate type. Figures 5.11 and 5.12 show the distribution of real estate values by municipality and type, respectively. By looking at some descriptive statistics, we noticed that the real estate value differs for different types in the same municipality. This also applies to the values of the same real estate type in different municipalities. These differences can be clearly seen in Figures 5.13, 5.14 and 5.15. Hence, the model will be trained after filtering the data by municipality and type. For example, if we are given the area and the price of a new real estate, along with the information regarding its location and type, we will extract from the data the observations that match with the location and type of the

new observation. Moreover, since the non-parametric techniques work better with large data sets, and since they are time saving when the sample size increases, we are going to use the Robins-Monro estimator if the extracted data contains more than 30 observations. For the subgroups that are of size 30 or below, we use the Nadaraya-Watson Estimator.

Figure 5.16 shows the eight municipalities of Qatar, where the darkness of the color is determined by the mean real estate value. The prediction results are visualized in Figure 5.17. It can be seen that the colors of the second graph are very close to those of the first graph. Moreover, Figure 5.18 shows that boxplots of the true and predicted values are close to each other, which reflects the quality of the predictions.

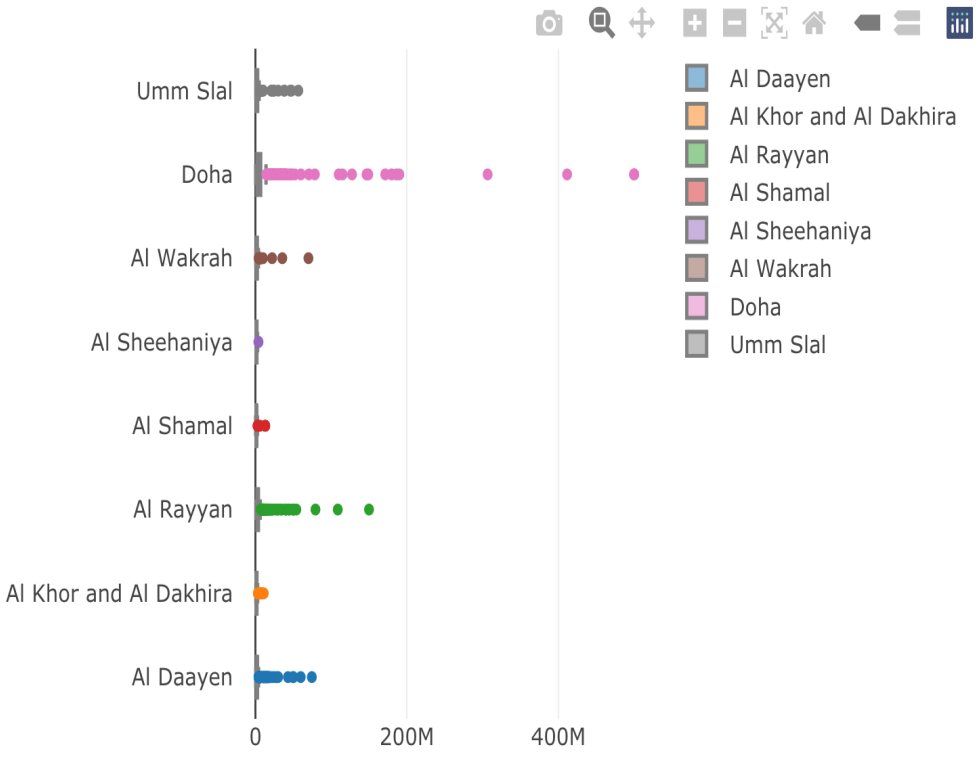


Figure 5.11. Distribution of real estate value by municipality.

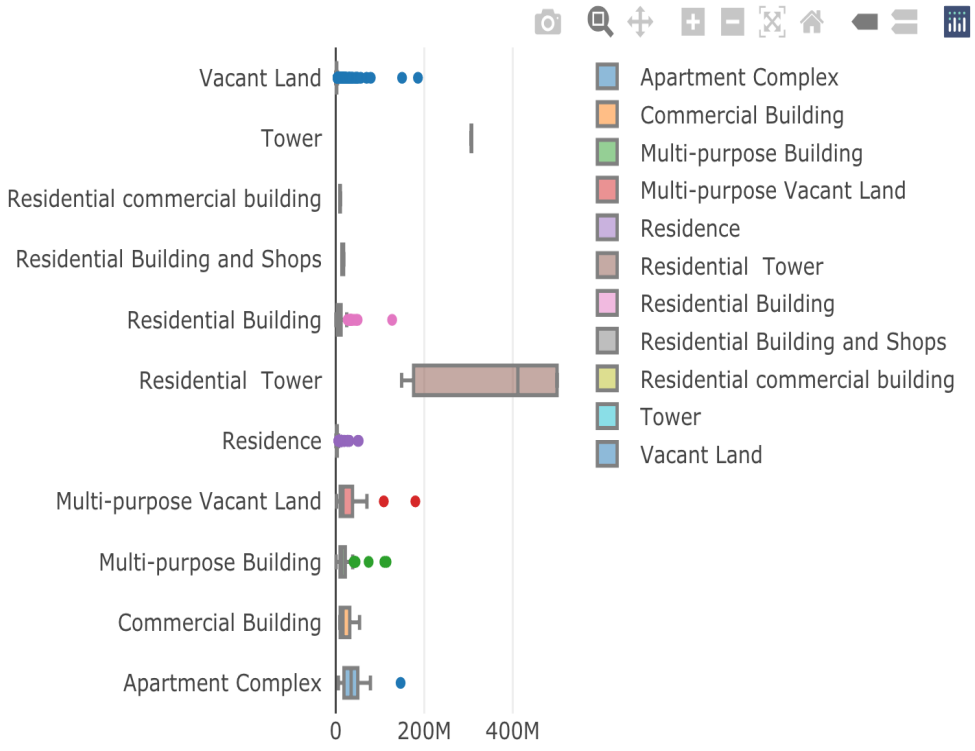


Figure 5.12. Distribution of real estate value by type.

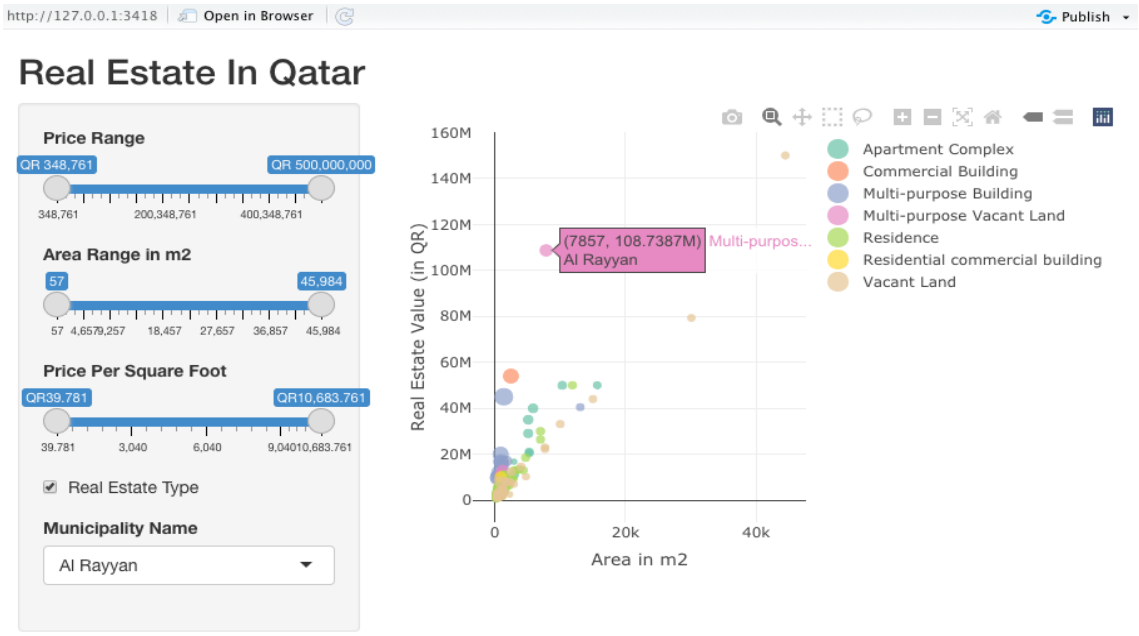


Figure 5.13. Initial Shiny Application to explore the real estate data in Qatar.

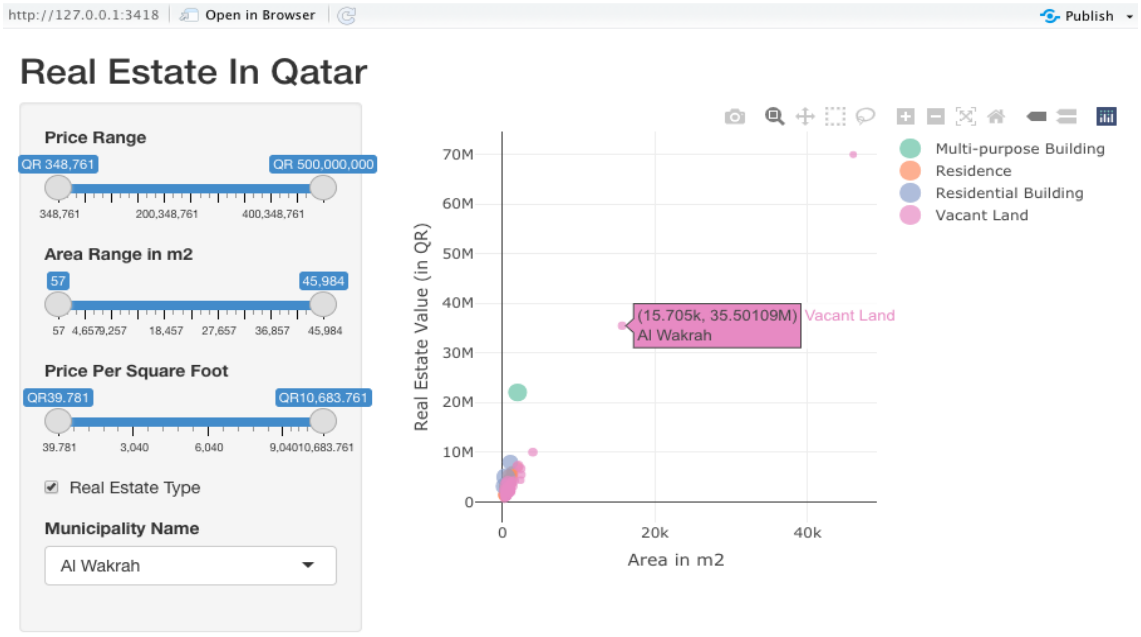


Figure 5.14. Initial Shiny Application to explore the real estate data in Qatar.

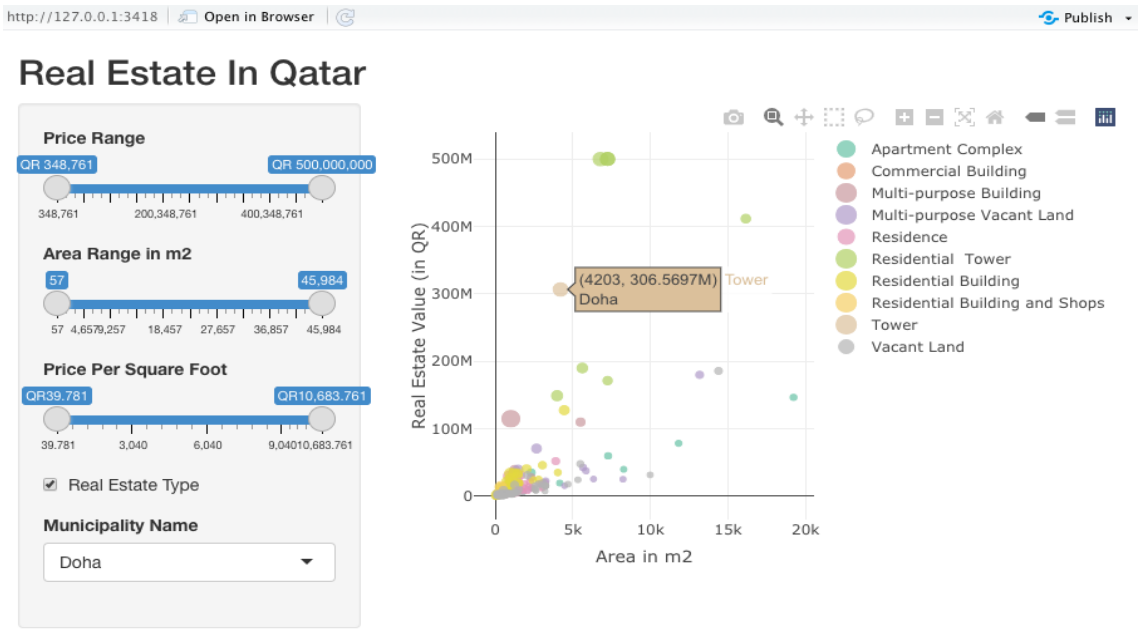


Figure 5.15. Initial Shiny Application to explore the real estate data in Qatar.

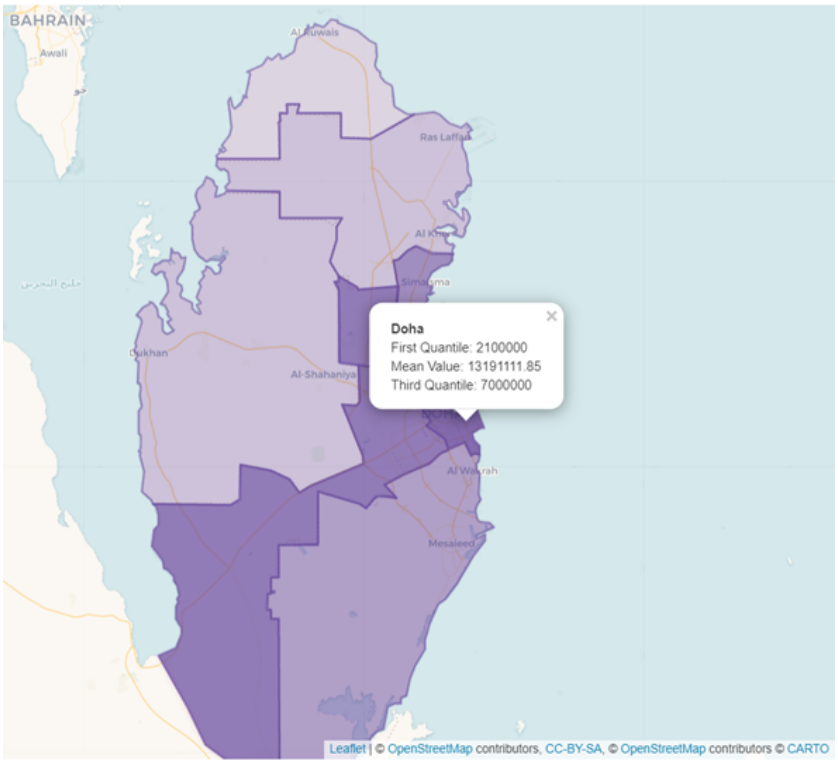


Figure 5.16. Observed average real estate value by municipality. Pop-up shows the mean, first and third quartile values.

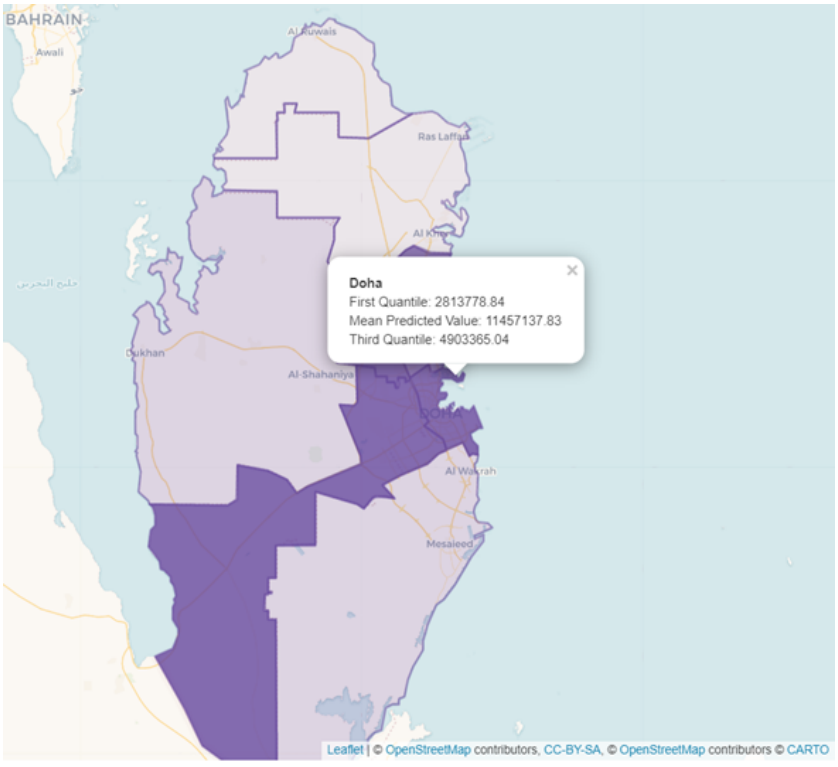


Figure 5.17. Estimated average real estate value by municipality. Pop-up shows the mean, first and third quartile values.

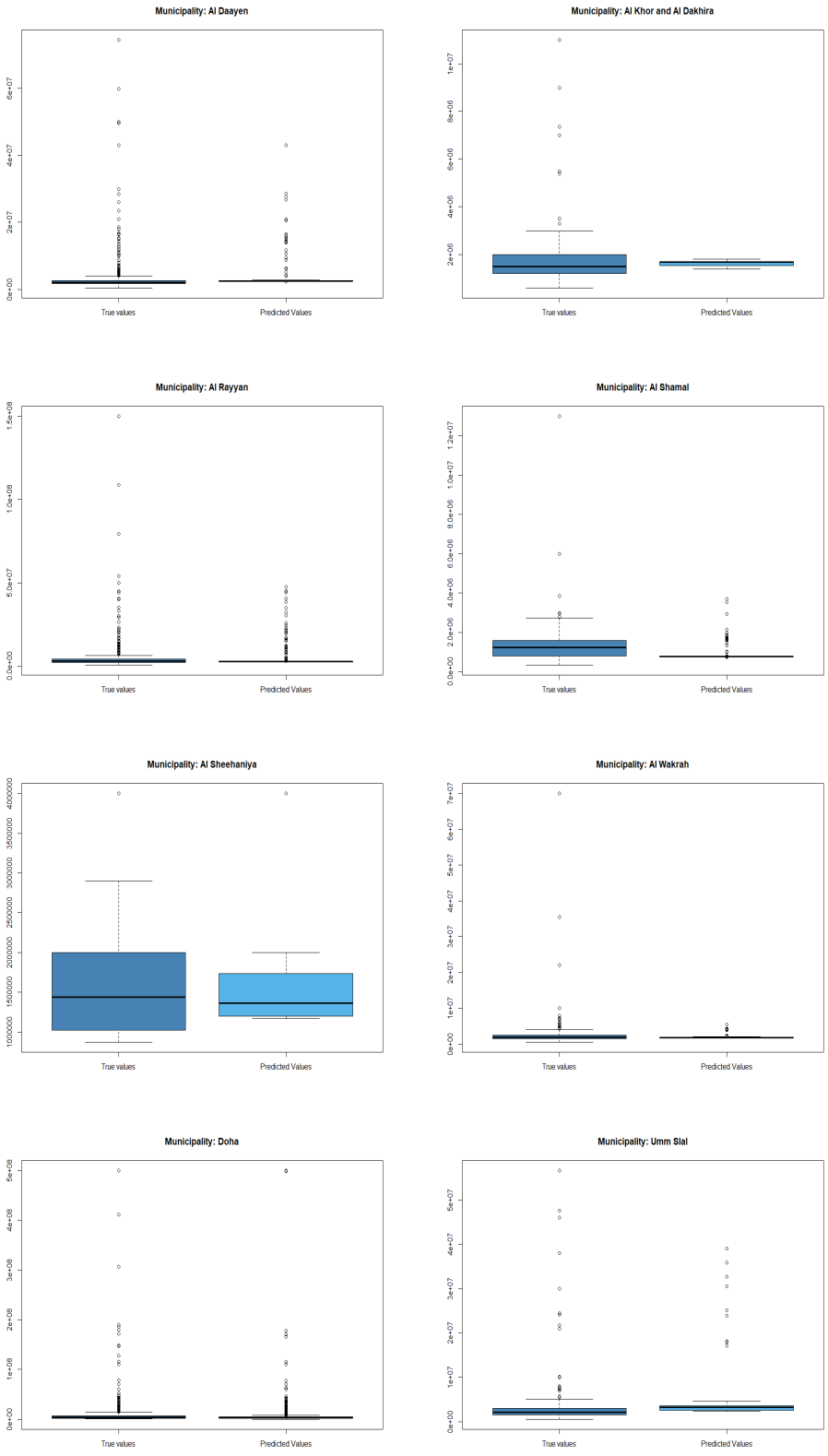


Figure 5.18. Boxplots of the actual and predicted real estates value for each municipality.

REFERENCES

[1] E. Nadaraya, "On the estimation regression," *Theory Probab. Appl*, vol. 9, pp. 141–142, 1964.

[2] G. Watson, "Smooth regression analysis," *Sankhya, Ser. A*, vol. 59, pp. 359–372, 1964.

[3] D. Bosq, *Nonparametric statistics for stochastic processes lecture. Estimation and prediction* (Lecture notes in statistics, 2nd ed.). Springer-Verlag, New York, 1998.

[4] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[5] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Statist.*, vol. 23, pp. 462–466, 1952.

[6] P. Revez, "How to apply the method of stochastic approximation in the non-parametric estimation of a regression function," *Math. Operationsforsch. Statist., Ser. Statistics*, vol. 8, pp. 119–126, 1977.

APPENDIX

Derivation of $\widehat{m}_n(x)$

As discussed in Section 2.2.2, $m(x)$ is identified, at any fixed point x, as:

$$\begin{aligned} m(x) &= \mathbb{E}(Y \mid X = x) \\ &= \int y f_{Y|\mathbf{X}}(y|x)dy \\ &= \int y \, \frac{f_{XY}(x,y)}{f_X(x)}dy, \end{aligned}$$

where f_X is the marginal density of X and f_{XY} is the joint density of X and Y. The kernel-type estimators of f_X and f_{XY} are defined as follows:

$$\begin{aligned} \widehat{f}_n(x;h) &= \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) \\ \widehat{f}_{XY,n}(x,y;h) &= \frac{1}{nh^2}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) H\left(\frac{Y_i - y}{h}\right). \end{aligned}$$

Therefore, a plug-in estimator of m can be defined after replacing f_X and f_{XY} by their empirical version. That is, for any fixed x, one gets

$$\begin{aligned} \widehat{m}_n(x) &= \int y \, \frac{n^{-1}h^{-2}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) H\left(\frac{y - Y_i}{h}\right)}{(nh)^{-1}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)} dy \\ &= \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}{h\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)} \int y \, H\left(\frac{y - Y_i}{h}\right) dy. \end{aligned}$$

Let $u = h^{-1}(y - Y_i)$. Then

$$\begin{aligned}
\widehat{m}_n(x) &= \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}{h\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)} \, h \int (uh + Y_i)\, H(u)du \\
&= h \int uH(u)du + \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)} \int H(u)du.
\end{aligned}$$

Note that if we assume that H is a symmetric density function then $\int H(u)du = 1$ and $\int uH(u)du = 0$. Hence, we finally obtain

$$\widehat{m}_n(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)}$$

Table .5. Descriptive statistics about CTG Variables

Variables	Quantity					
	Min.	$Q_{25\%}$	Median	Mean	$Q_{75\%}$	Max.
LB	106.0	126.0	133.0	133.3	140.0	160.0
AC	0.0	0.0	0.001630	0.003170	0.005631	0.019284
FM	0.0	0.0	0.0	0.009474	0.002512	0.480634
UC	0.0	0.001876	0.004482	0.004357	0.006525	0.014925
DL	0.0	0.0	0.0	0.001885	.003264	0.015385
DS	0.0	0.0	0.0	3.585E-06	0.0	1.353E-03
DP	0.0	0.0	0.0	.0001566	0.0	0.0053476
ASTV	12.0	32.00	49.00	46.00	61.00	87.00
MSTV	0.200	0.700	1.200	1.333	1.700	7.000
ALTV	0.0	0.0	0.0	9.847	11.0	91.0
MLTV	0.0	4.600	7.400	8.188	10.800	50.700
Width	3.00	37.00	67.50	70.45	100.00	180.00
Min	50.00	67.00	93.00	93.58	120.00	159.00
Max	122	152	162	164	174	238
Nmax	0.0	2.00	3.00	4.06	6.00	18.00
Nzeros	0.0	0.0	0.0	0.3236	0.0	10.00
Mode	60.0	129.0	139.0	137.5	148.0	187.0
Mean	73.0	125.0	136.0	134.6	145.0	182.0
Median	77.0	129.0	139.0	138.1	148.0	186.0
Variance	0.0	2.00	7.00	18.81	24.00	269.00