DATA-DRIVEN SMOOTHING PARAMETER SELECTION IN DENSITY ESTIMATION

by

SACHITHRA OPATHALAGE

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES OF
THE UNIVERSITY OF MANITOBA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF STATISTICS
UNIVERSITY OF MANITOBA
WINNIPEG, MANITOBA, CANADA

Abstract

Kernel density estimation (KDE) is a seasoned concept in nonparametric density estimation problems. KDE accuracy depends on the shape of the kernel as well as the bandwidth of the kernel. However, the shape of the kernel has only a minor influence on the estimation, whereas selecting proper smoothing parameter (bandwidth) is critical. If the bandwidth is too small, then spurious features become visible, whereas when the selected bandwidth is too large, important features disappear. Many bandwidth selection methods have been developed over the years, where each has its own characteristics. Few bandwidth selection methods are selected systematically from the recent research literature and verified using simulations in R for a sample dataset. Strengths and limitations of each method is identified and discussed.

Similarly, there exists Bernstein density estimation (BDE) methods for nonparametric density estimation, which are gaining much interest recently. BDEs have an advantage over KDEs when underling density is supported in an unit interval. BDEs are inherently stable in boundaries and have very low boundary bias, but they also introduce considerable variance when compared to KDEs. Like bandwidth selection in KDE, accurate order selection is critical in BDE. Order selection criteria of existing BDEs are then discussed. Based on the limitations identified from KDEs and existing BDEs, few data driven order selection methods are introduced for Bernstein polynomial estimators of density functions on the unit interval. These methods are also verified with a simulation in R, and respective error criteria are compared to verify the effectiveness of the new order selection methods. Finally, bootstrapped order selection method is identified as a potential candidate for further investigation, whereas it's desirable features are clearly identified.

Keywords: Kernel density estimation, Nonparametric density estimation, Bandwidth selection, Bernstein density estimation, Order selection, Bootstrapped order selector.

Acknowledgements

I would like to express sincere thanks to my advisor Prof. Alexandre Leblanc for the guidance and support given to me throughout the M.Sc. program. His advice and suggestions were extremely helpful not only for the research but also to decide my future career path as well. I consider it a privilege to work with a highly knowledgeable person as you.

Many thanks to my thesis committee members, Prof. Saumen Mandal and Prof. Po Yang, for the comments and suggestions to improve my research.

Last but not least, I would like to thank my family and friends. Many thanks to my loving husband and my son for their kindness and patience and the support given while I was pursuing my studies.

Dedication

To my loving family.

TABLE OF CONTENTS

Chapter 1

Introduction

Apart from histograms, the kernel density estimator is the most commonly used density estimator in academia (see Silverman, 2018). But to effectively use this estimation method, a proper bandwidth should be selected for the kernel. In simple cases one might be able to use visual inspection for adjusting the smoothing parameter, but there should be a systematic way of obtaining an appropriate smoothing parameter relative to a given data set. One reason for such automatic selection of smoothing parameter is the necessity to set a default in data analysis software (see M. C. Jones et al., 1996). Another reason is for use in applications where many densities have to be estimated simultaneously from many data sets, for instance when monitoring processes for quality control.

1.1 Density Estimation

Loosely speaking, the density, or the probability density, is the relationship between the outcomes of a random variable and its probability. Some outcomes of a random variable may yield low probability, whereas another outcome may result in a higher probability. Probability calculations can be done using the probability density function (PDF) if the variable is continuous. The shape of the PDF is called the probability distribution and common probability distributions are uniform, normal, exponential, etc.

For any random variable X with a probability density function f, the probabilities associated with X can be expressed as

$$P(a < X < b) = \int_a^b f(x)\,dx\,,$$

for all $a < b$. If we know the density of a random variable, we can calculate its moments, such as mean and variance. We can also use it for deciding whether an observation is unlikely or very unlikely and might be an outlier or anomaly. One common issue is that we may not know the probability distribution for a specific random variable because we don't have access to every single outcome for that random variable. However, we may have access to a random sample from that distribution and thus, we can estimate this probability density function.

One way to deal with density estimation is to use a parametric approach. Consider data drawn from a known distribution, such as the normal distribution with mean μ and variance σ^2. Once we know the reference family of distributions, we can estimate the density of the random variable by estimating the parameters of the distribution from a random sample of data. After we have estimated the density, we can find if it is a good fit. Easily, we can plot the density function and compare it with the histogram as shown in Figure 1.1. Another approach is to repeatedly sample from this estimated density function and compare those simulated samples to the real sample that was actually observed.

In some instances, however, the PDF can't be easily identified from the sampled data. For this instance, the data may not seem closely related to a known parametric family. Parametric density estimation is then not useful and a nonparametric method can instead be used. The distribution will still have parameters but these will not be easily estimated as with a simple PDF. For this instance, all observations in a random sample will be used to estimate the PDF, which makes all of them "parameters" of the PDF in some sense.

The focal point of this thesis will be nonparametric methods to estimate a univariate probability function. In particular, we will discuss Histograms and Kernel density estimates

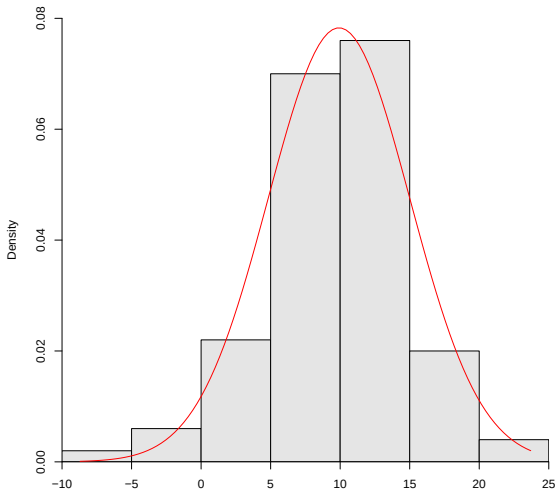


Figure 1.1. *Histogram of a random sample of size* $n = 100$ *generated from a normal distribution with* $\mu = 10$ *and* $\sigma = 5$*. The true density is given in red.*

(KDE) but our main interest will be Bernstein polynomial estimates.

1.2 Histograms

Histograms are a powerful tool when it comes to identifying the overall shape of univariate distributions. The histogram is the oldest and most widely used density estimation tool. Specifically, let $X_1, X_2, ..., X_n$ denote a sample of n independent univariate observations with density f. Now consider an origin x_0 and a bin width h. The bins of the histogram are defined as intervals of the form $[x_0 + mh, x_0 + (m + 1)h)$ for positive and negative integers m. Here, the left boundary is chosen to be closed and right boundary is chosen to be open although the opposite would work as well. Silverman (2018) defines a histogram as

$$\hat{f}(x) = \frac{1}{nh}(\text{no of } X_i \text{ in same bin as x}).$$

Clearly, bin width plays an important role in the above expression and plays the role of a parameter in density estimation using a histogram. Coarser bins produce flatter density

estimates whereas finer bins produce fluctuating density estimates.

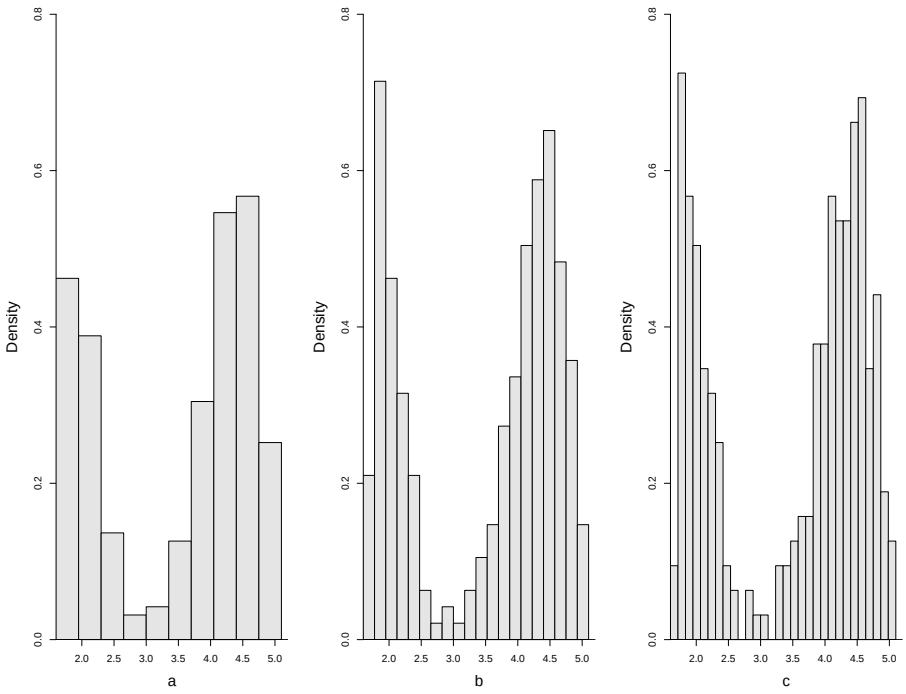


Figure 1.2. *Histogram of the old faithful geyser data where number of eruptions* $n = 272$ *with bin width a) h=0.4; b) h=0.2; c) h=0.1.*

This can be seen in Figure 1.2 where we present a histogram of the famous old faithful geyser data. These data represent the durations of 272 eruptions of the old faithful geyser in Yellow Stone National Park, USA, and were originally used by Silverman (1986). Another important parameter is the starting position of the first bin, which will affect density estimation for all bins and this effect is shown in Figure 1.3. The estimate defined above can also be written as

$$\hat{f}(x) = \frac{(\text{proportion of } X_i \text{ in same bin as x})}{(\text{width of bin containing } x)}.$$

Bin width and origin should be selected appropriately to get a satisfactory density estimate as a bad choice of these parameters will result in an unsatisfactory density estimate. Furthermore, being step functions, histograms provide density estimates that can change suddenly.

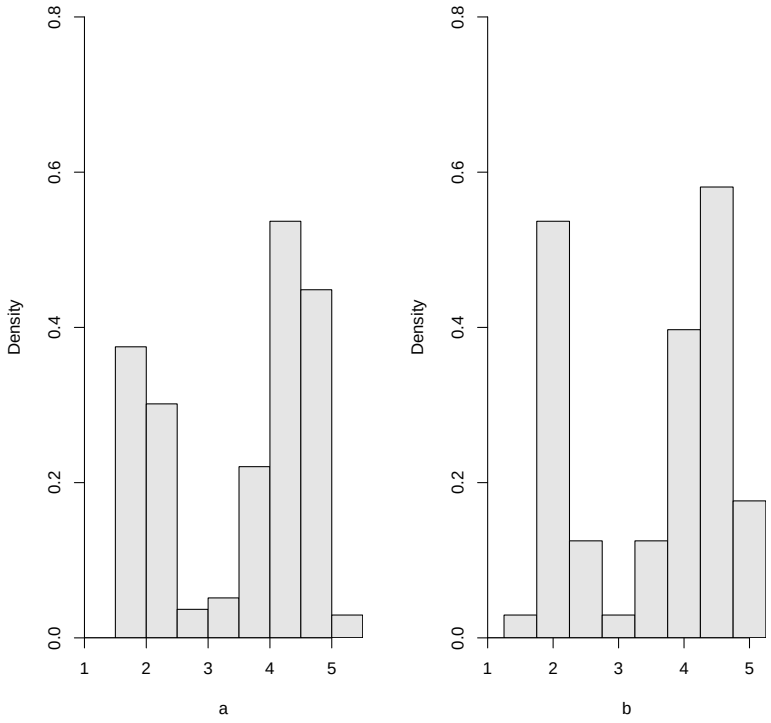


Figure 1.3. *Histogram of the Old Faithful Geyser data with origin a)* $x_0 = 1.5$*; b)* $x_0 = 1.25$

Histograms can be flat and then change suddenly for a few outliers. Finally, when the observations are multi-dimensional, the number of bins increases exponentially thus making the number of observations required to populate the bins also increase too quickly.

Although the histogram is a simple, intuitive and powerful tool in density estimation, the above drawbacks motivate one to consider alternate density estimation techniques. For instance, a generalization of the histogram can be obtained by allowing bin width to vary, but by fixing the number of observations falling into each bin instead. This popular variant of the histogram is known as the nearest neighbour estimator. The approach we take in what follows will be different however.

1.3 The Kernel Density Estimator

Kernel density estimation (KDE) is the most common nonparametric approach to PDF estimation. As stated earlier, histograms are not smooth, because they are basically step functions. The overall shape of the histogram depends on the bin width and the starting point of the bins. Kernel estimators on the other hand, place a kernel function at each data point to eliminate the dependence on the end points of the bins. A kernel is a mathematical function which satisfies the conditions

$$\int_{-\infty}^{\infty} K(x)dx = 1,$$

and $K(x) \geq 0$ for all $x \in R$, making K a probability density function. Although it is not required, it is also generally assumed that K is symmetric. Following (see Silverman, 2018), we define the kernel estimator with kernel K as

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - X_i}{h}\Big). \tag{1.1}$$

By introducing the notation $K_h(u) = h^{-1}K(u/h)$ we can further express (1.1) as,

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i) \tag{1.2}$$

This estimator places an identical "bump" at each observation, $X_1, X_2, ..., X_n$. The shape of these bumps is determined by the kernel function K, whereas the width is determined by the smoothing parameter h, also called *bandwidth*. As can be seen in (1.1) and illustrated in Figure 1.4, the estimator $\hat{f}_h$ is constructed by adding together all these bumps located at the observations. For this example, we use Gaussian kernel and the observed sample is X_i = 0, 1, 1.1, 1.5, 1.9, 2.8, 2.9 and 3.5.

The change in bandwidth and its effect on the kernel estimate is illustrated in Figure 1.5. When h is small, spikes tend to appear in the density estimate, whereas the estimate flattens when h becomes large. In other words, a large bandwidth may produce coarse density

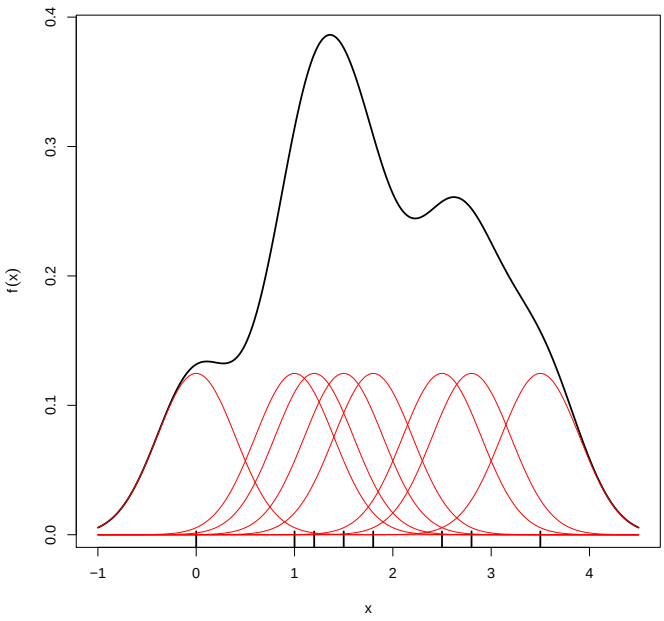


Figure 1.4. *Kernel estimator with individual kernels of bandwidth h=0.4 for a sample size n=8 with observations on X_i = 0, 1, 1.1, 1.5, 1.9, 2.8, 2.9 and 3.5.*

estimates with little details and a small bandwidth may show too much details. Although a sample of size 8 was used in Figures 1.4 and 1.5, the same effect can be seen with large datasets. Figure 1.6 illustrates this by displaying two kernel estimators constructed from the Old Faithful Geyser data considered earlier. In contrast to the histograms of Figures 1.2 and 1.3, individual windows are centered at each data point. Hence the KDE doesn't depend on a change of origin or starting point. We have seen that the bandwidth h has a crucial impact on the KDE; its shape can vary significantly when changing the value of h. Apart from the bandwidth h, the KDE also depends on the kernel function used for estimation. The Gaussian, Uniform and Epanechnikov kernels are some of the commonly used kernel functions. Interestingly, the kernel function has a minor influence on the resulting kernel estimate as illustrated in Figure 1.7.

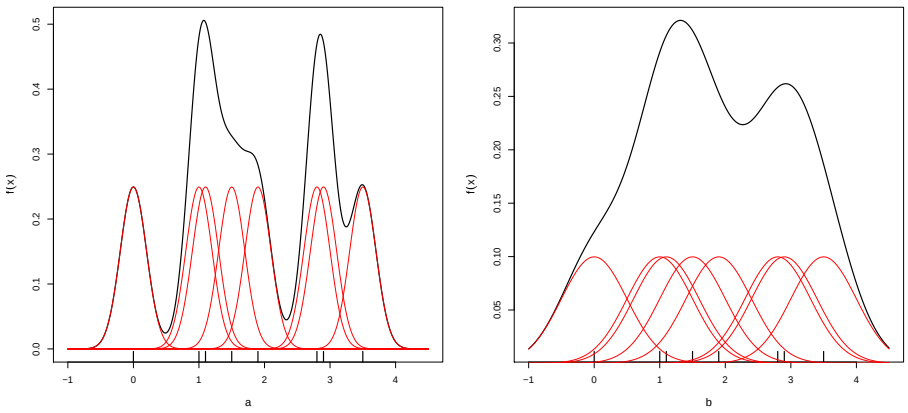


Figure 1.5. *Kernel estimator with individual Gaussian kernels, (a) h=0.2 and (b) h=0.5, for the same example as in Figure 1.4*

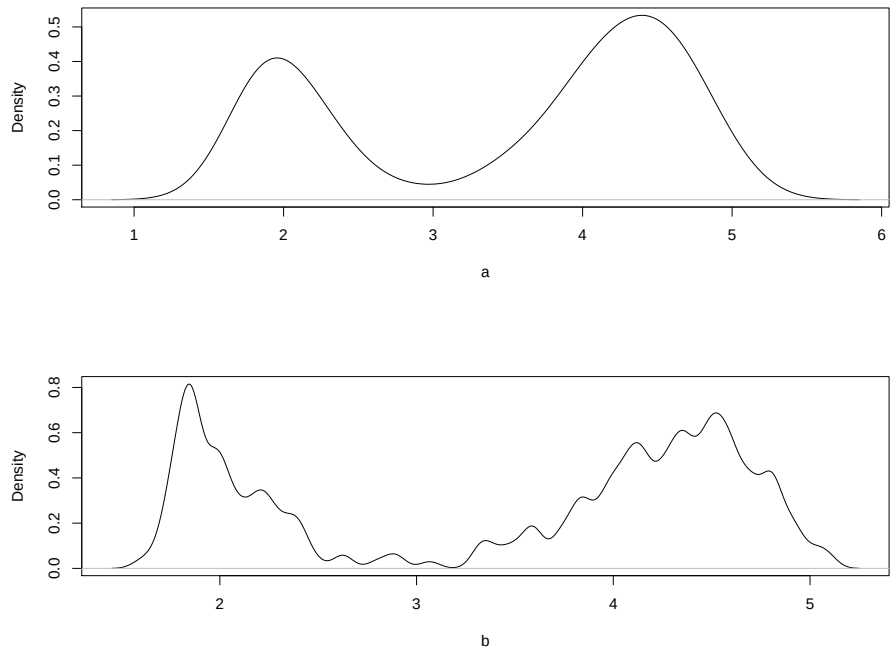


Figure 1.6. *Kernel estimates for the Geyser data obtained using a Gaussian kernel with a) h=0.25 b) h=0.05*

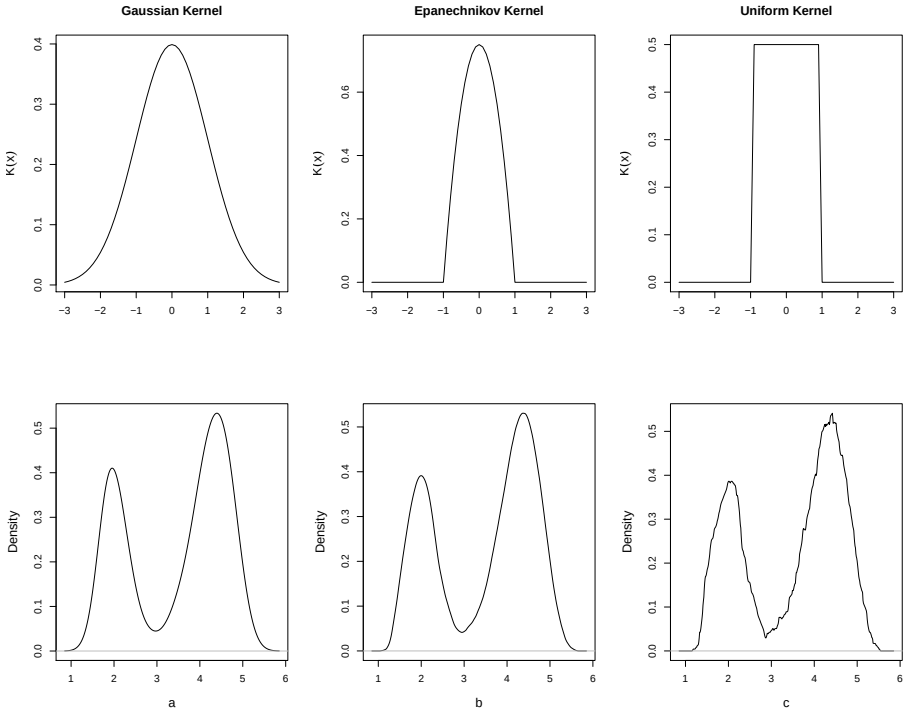


Figure 1.7. *Three kernel functions and corresponding density estimators for the Old Faithful Geyser data with h=0.25.*

1.3.1 Evaluation of Kernel Estimator

From an inferential perspective, it is important that an appropriate error criteria be defined to evaluate the performance of the kernel density estimator. One such common measure in Statistics is the mean squared error (MSE), defined as

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta)^2,$$

where $\hat{\theta}$ generally denotes an estimator of a parameter θ, for which inference is desired. This can be decomposed into variance and squared bias as

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + \Big[\mathbb{E}(\hat{\theta}) - \theta\Big]^2.$$

This decomposition makes it easy to interpret the performance of the kernel density estimator. In the current context, MSE can be used to asses the performance of $\hat{f}_h$ at a point $x \in \mathbb{R}$. For this, the mean and variance of $\hat{f}_h(x)$ can be derived from (1.2) where X is a random variable of density f. Specifically, we have

$$\mathbb{E}\Big[\hat{f}_h(x)\Big] = \mathbb{E}\Big[K_h(x - X)\Big] = \int_{\mathbb{R}} K_h(x - y) f(y)\, dy,$$

and, introducing the convolution notation,

$$(f * g)(x) = \int_{\mathbb{R}} f(x - y) g(y)\, dy,$$

the bias of $\hat{f}_h(x)$ can be finally expressed as,

$$\mathbb{E}\Big[\hat{f}_h(x)\Big] - f(x) = (K_h * f)(x) - f(x). \tag{1.3}$$

Similarly, the variance of $\hat{f}_h(x)$ can be derived as,

$$\mathrm{Var}\Big[\{\hat{f}_h(x)\}\Big] = \frac{1}{n}\Big[(K_h^2 * f)(x) - (K_h * f)^2(x)\Big]. \tag{1.4}$$

Expressions (1.3) and (1.4) can be combined to obtain

$$\text{MSE}\{\hat{f}_h(x)\} \quad = \quad \frac{1}{n}\{(K_h^2 \ * \ f)(x) \ - \ (K_h \ * \ f)^2(x)\} \ + \ \{(K_h^2 \ * \ f)(x) \ - \ f(x)\}^2$$

Thus, as above, we again have that

$$\text{MSE}\{\hat{f}_h(x)\} = VARIANCE + BIAS^2.$$

Instead of considering estimating the density f at a unique fixed point x, it is usually designed to estimate f over its entire support, especially to visualize its global features. Hence it is required to asses the estimation error over the global domain rather than in a local region or at a given point. This is achieved by integrating the mean squared error. The resulting integrated mean squared error is given by

$$\text{IMSE}(\hat{f}_h) = \int_{\mathbb{R}} MSE(\hat{f}_h(x))dx.$$

This measure of performance averages the performance of $\hat{f}_h$ over all possible samples. Another approach is to instead assess how well f has been estimated for the given data set (see Wand & Jones, 1994). This is done using the integrated squared error

$$\text{ISE}\{\hat{f}_h\} = \int_{\mathbb{R}} \left\{\hat{f}_h(x) - f(x)\right\}^2 dx. \tag{1.5}$$

Note that $\text{ISE}(\hat{f}_h)$ is itself a random variable which depends on h and it evaluates the error for the given data set only, it does not account for other possible data sets that can be drawn from the density. We note that it is normally the case that

$$\mathbb{E}[\text{ISE}\{\hat{f}_h\}] = \mathbb{E}\int_{\mathbb{R}} \left\{\hat{f}_h(x) - f(x)\right\}^2 dx = \int_{\mathbb{R}} \text{MSE}\{\hat{f}_h(x)\}dx = \text{IMSE}(\hat{f}_h) \tag{1.6}$$

An interesting discussion on these discrepancy measures can be found in M. C. Jones et al. (1996). Squared error is often preferred to the absolute error for instance, as it reduces

computational complexity. This being said, some authors are strong advocates of the use of integrated absolute errors (see Devroye et al., 1997).

1.3.2 Bandwidth selection

When MISE is considered, one common approach to minimize the error associated with the density estimator is to obtain the optimum bandwidth h_{opt} defined as

$$h_{opt} = \underset{h>0}{\operatorname{argmin}} \operatorname{MISE}(h).$$

In order to do this, however, we need an explicit expression of the MISE that we can try to minimize. When f is sufficiently smooth (twice continuously differentiable) such an expression that is valid asymptotically is,

$$\operatorname{MISE}(\hat{f}_h) = \frac{1}{4}\mu_2^2(K)R(f'')h^4 + \frac{1}{nh}R(K) + o(h^4 + \frac{1}{nh}),$$

where, $\mu_2(K) = \int_{\mathbb{R}} x^2 K(x)dx$ and $R(z) = \int_{\mathbb{R}} z(x)^2 dx$ (see Wand & Jones, 1994), sect.2.5. The remainder form of the above expression is negligible when $h \to 0, \frac{1}{nh} \to 0$ as $n \to \infty$. The governing part of the MISE, denoted as the asymptotic MISE (AMISE), is then given by

$$\operatorname{AMISE}(\hat{f}_h) = \frac{1}{4}\mu_2^2(K)R(f'')h^4 + \frac{1}{nh}R(K). \tag{1.7}$$

Looking closely at the above equation, we can see that the first term (integrated squared bias) is asymptotically comparable to h^4. This means if we want to decrease bias, we should take h to be small. But by decreasing h the second term (integrated variance) is increased due to its inverse relationship. As n increases, h should behave in such a way that both terms of AMISE become smaller thus leading to the conditions given above. This is often referred to as the variance-bias trade off, and it is a critical aspect of proper bandwidth selection.

The expression for AMISE in (1.7) is much simpler than the expression for the MISE.

Due to its simple form it is possible to derive the bandwidth that minimizes the AMISE as,

$$h_{\text{AMISE}} = \left[\frac{R(K)}{\mu_2(K)^2 R(f'')}\right]^{1/5} n^{-1/5}. \tag{1.8}$$

Apart from the dependence to K and n, the h_{AMISE} is inversely proportional to $R(f'')^{1/5}$, where $R(f'') = \int_{\mathbb{R}} (f''(x))^2 dx$ is a measure of the total curvature of the density function to be estimated. Thus, the optimal bandwidth should be smaller for a function with higher curvature and vice versa. Unfortunately, using (1.8) directly is usually not possible since $R(f'')$ is not known. One solution is to estimate $R(f'')$ based on the available data and then derive the h_{AMISE}, but many other approaches are possible. Some of these will be discussed in detail in Chapter 2.

1.4 Bernstein Polynomials

Polynomial functions are extensively used to model scientific problems due to their finite properties under algebraic operations. In other words, they show closure under mathematical operations such as addition, multiplication, differentiation, integration and composition. Generally, a closed form expression for a scientific problem is not easily obtained. Often a solution is instead approximated using one or more closed form polynomial approximations. The inception of the Bernstein basis was to address this approximation requirement (see Bernstein, 1912; Farouki, 2012).

The Bernstein polynomial basis of degree m is defined as

$$P_{k,m}(x) = \binom{m}{k} x^k (1-x)^{m-k}, \tag{1.9}$$

for $k = 0, ..., m$. A linear combination of Bernstein polynomials can be defined by

$$B_n(x) = \sum_{k=0}^{m} \beta_k P_{k,m}(x),$$

and is known as a Bernstein polynomial of order m, where β_k are referred to as Bernstein

coefficients (Kakizawa, 2004).

Density estimation can be achieved with the aid of Bernstein polynomials since they are a simple and convenient form of polynomial approximation and have a nice probabilistic interpretation. In the case of a density defined on the unit interval, kernel estimators generally tend to result in unsatisfactory behaviour near the boundaries, whereas Bernstein polynomial estimators naturally lead to acceptable behaviour near the boundaries (see Leblanc, 2010). More details on Bernstein polynomial estimators, on the selection of the order m of an estimator and the application of Bernstein polynomial estimators to density estimation will be presented in Chapter 3.

1.5 Organization of Thesis

This thesis is organized as follows. In Chapter 2, we discuss existing bandwidth selection methods in kernel density estimation. A brief description of some selected methods is given. The pros and cons of each method are discussed. In Chapter 3, we discuss on Bernstein density estimators (BDE) and order selection for BDE, where as cross-validation based methods and bootstrap based methods are compared. Simulation results and data analysis are presented in Chapter 4. We propose Bernstein polynomial order selection methods and examine and validate them through simulation. Chapter 5, the final chapter, summarizes our findings and presents conclusions and opportunities for improvement.

Chapter 2

Bandwidth selection methods in Kernel Density Estimation

Even though bandwidth selection is a critical aspect in kernel density estimation, a perfect procedure for selecting an optimal bandwidth is yet unknown in the statistical literature. However, selection of a reasonable bandwidth selector can be achieved from existing bandwidth selection methods, at least for specific problems (see Heidenreich et al., 2013). Hence it is helpful to review these methods and their performance. More than 30 bandwidth selectors can be identified in the recent statistical literature, of which a few methods will be discussed in this chapter.

2.1 Plug-in methods

2.1.1 Normal scale rules

This method is based on using the bandwidth that is AMISE-optimal for the normal distribution scaled to the estimate of the underlying distribution. Using (1.8) the bandwidth that minimizes MISE($\hat{f}_h$) can be shown as

$$h_{\text{AMISE}} = \left[\frac{R(K)}{\mu_2(K)^2 R(f'')}\right]^{1/5} n^{-1/5},$$

where

$$\mu_j(K) = \int_{\mathbb{R}} x^j K(x)dx$$

is the jth moment of the kernel K. Then kernel K is called a k^{th} *order* kernel, if $\mu_0(K) = 1, \mu_j(K) = 0$ for $j = 1,..,k-1$ and $\mu_k(K) \neq 0$. In what follows, it is required that K is symmetric and that k is even. It can be shown that, for a normal distribution f with variance σ^2,

$$h_{AMISE} = \left[\frac{8\pi^{1/2}R(K)}{3\mu_2(K)^2}\right]^{1/5} \sigma\, n^{-1/5}.$$

Now by replacing σ with an estimate $\hat{\sigma}$ as described in Silverman (1986), pp.45-47, we can obtain a normal scale bandwidth selector

$$\hat{h}_{NS} = \left[\frac{8\pi^{1/2}R(K)}{3\mu_2(K)^2}\right]^{1/5} \hat{\sigma} n^{-1/5}.$$

In this case the bandwidth to use then depends on the samples data through $\hat{\sigma}$ and on the kernel to be used to construct the density estimator.

2.1.2 Direct plug-in rules

The basic idea in plug-in methods is to "plug in" an estimate for $R(f'')$ in the asymptotically optimal bandwidth. Define

$$R(f^{(s)}) = \int_{\mathbb{R}} f^{(s)}(x)^2 dx,$$

the integral of the squared s^{th} derivative of f. Under sufficient smoothness assumptions on f, it can be shown that

$$R(f^{(s)}) = (-1)^s \int_{\mathbb{R}} f^{(2s)}(x) f(x) dx.$$

Therefore, we can study estimations of functions of the form

$$\psi_r = \int_{\mathbb{R}} f^{(r)}(x) f(x) dx,$$

for even r. Then, as previously discussed in (1.8) the AMISE optimal bandwidth can be defined as

$$h_{\text{AMISE}} = \left[\frac{R(K)}{\mu_2(K)^2 \psi_4} \right]^{1/5} n^{-1/5}.$$

Direct plug-in (DPI) rules simply replace ψ_4 by an estimated $\hat{\psi}_4$ leading to

$$\hat{h}_{\text{DPI}} = \left[\frac{R(K)}{\mu_2(K)^2 \hat{\psi}_4} \right]^{1/5} n^{-1/5}.$$

As outlined in Wand and Jones (1994)-sect.3.5 a common approach is to use a kernel estimator

$$\hat{\psi}_r(g) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}^{(r)}(X_i; g) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} L_g^{(r)}(X_i - X_j),$$

where L is another kernel function potentially different from K and that depends on a pilot bandwidth g. Even though it is simple, this method is not fully automatic due to the dependence of $\hat{h}_{DPI}$ on the choice of this pilot bandwidth. A simple way of choosing g is to use the equation for the AMSE-optimal bandwidth for $\hat{\psi}_4(g)$. Considering the use of the same second order kernel, the AMSE-optimal bandwidth to estimate ψ_4 can be stated as

$$g_{\text{AMSE}} = \left[\frac{2K^{(4)}(0)}{-\mu_2(K)\psi_6} \right]^{1/7} n^{-1/7}. \tag{2.1}$$

As can be seen from the above equation, the optimal choice for g depends on the unknown ψ_6. If we try to estimate ψ_6 from another kernel estimate, as was done above with ψ_4, then there will be a dependence to ψ_8. Generally, the optimal bandwidth for estimating the functional ψ_r depends on ψ_{r+2}. So, there is a family of plug-in bandwidth selectors which depend on the number of stages l, rather than one simple selector for a given problem. This rule is called the *l stage direct plug-in* bandwidth selector and the resulting bandwidth is denoted as $\hat{h}_{DPI,l}$. The normal scale rule is a special instance of the method and corresponds to the zero-stage direct plug-in bandwidth.

The functional ψ_r associated to the normal density with variance σ^2 can be shown to

be

$$\psi_r = \frac{(-1)^{r/2}\, r!}{(2\sigma)^{r+1}(r/2)!\pi^{1/2}}, \tag{2.2}$$

when r is even, and $\psi_r = 0$ for r odd. Making use of this, the following 4-stage direct plug-in was proposed by Sheather and Jones (1991) using a second order kernel K. First estimate ψ_8 using the normal scale rule by using (2.2), leading to

$$\hat{\psi}_8^{NS} = 105/(32\pi^{1/2}\hat{\sigma}^9), \tag{2.3}$$

where $\hat{\sigma}$ is an estimate of scale. Next ψ_6 can be estimated using the kernel estimator $\hat{\psi}_6(g_1)$ where,

$$g_1 = \Big[\frac{-2K^{(6)}(0)}{\mu_2(K)\hat{\psi}_8^{NS}}\Big]^{1/9} n^{-1/9}. \tag{2.4}$$

Then, ψ_4 is estimated using the kernel estimator $\hat{\psi}_4(g_2)$ where,

$$g_2 = \Big[\frac{-2K^{(4)}(0)}{\mu_2(K)\hat{\psi}_6(g_1)}\Big]^{1/7} n^{-1/7}. \tag{2.5}$$

Finally, the selected plug-in bandwidth can be obtained as,

$$\hat{h}_{\mathrm{DPI},4} = \Big[\frac{R(K)}{\mu_2(K)^2\hat{\psi}_4(g_2)}\Big]^{1/5} n^{-1/5}. \tag{2.6}$$

Despite its simplicity, this approach leads to another difficulty, when it comes to choosing the number of stages to be used for the plug-in approach: can one determine the required number of stages? When the number of stages l increases, the bias of the bandwidth selector is reduced. On the other hand, when the number of stages increases, the selector becomes more variable. Determining the optimal number of stages is still an open problem.

2.1.3 Solve-the-equation rules

The main idea behind the solve-the-equation rule is to choose h to satisfy (2.7) below, and related to the expression for AMISE-optimal bandwidths (see Park & Marron, 1990; Scott

et al., 1980; Sheather, 1986). As was described above, the optimal AMISE value of h satisfies

$$h = \left[\frac{R(K)}{\mu_2(K)^2 \hat{\psi}_4(\gamma(h))} \right]^{1/5} n^{-1/5}, \tag{2.7}$$

where $\gamma(h)$ is a bandwidth to be determined. On the other hand, a pilot bandwidth to estimate ψ_4 can be found by noting that the optimal bandwidth g to estimate ψ_4 satisifes

$$g_{\text{AMSE}} = \left[\frac{2L^{(4)}(0)\mu_2(K)^2}{R(K)\mu_2(L)} \right]^{1/7} (-\psi_4/\psi_6)^{1/7} h_{\text{AMISE}}^{5/7}.$$

Hence, $\gamma(h)$ can be defined as,

$$\gamma(h) = \left[\frac{2L^{(4)}(0)\mu_2(K)^2}{R(K)\mu_2(L)} \right]^{1/7} \{-\hat{\psi}_4(g_1)/\hat{\psi}_6(g_2)\}^{1/7} h^{5/7}, \tag{2.8}$$

by making use of the kernel estimates of ψ_4 and ψ_6 that we previously introduced. The pilot bandwidths g_1 and g_2 can be calculated from (2.4) and (2.5), (see Sheather & Jones, 1991). Finally, h_{Solve} is found by numerically solving for the h value satisfying (2.7), using $\gamma(h)$ as defined in (2.8).

A two-stage solve-the-equation bandwidth selection can also be defined in the following way (see Wand & Jones, 1994)-sect.3.6.2. We proceed similarly to DPI using (2.3). First, assume that K is a symmetric kernel of order k, $k = 2, 4, ...$, having r derivatives, such that $(-1)^{(r+k)/2+1} K^{(r)}(0)\mu_k(K) > 0$ and estimate ψ_6 and ψ_8 with,

$$\hat{\psi}_6^{NS} = -15/(16\pi^{1/2}\hat{\sigma}^7) \quad \text{and} \quad \hat{\psi}_8^{NS} = 105/(32\pi^{1/2}\hat{\sigma}^9).$$

Next ψ_4 and ψ_6 can be estimated using $\hat{\psi}_4(g_1)$ and $\hat{\psi}_6(g_2)$ where,

$$g_1 = \left[\frac{-2K^{(4)}(0)}{mu_2(K)\hat{\psi}_6^{NS}} \right]^{1/7} n^{-1/7} \quad \text{and} \quad g_2 = \left[\frac{-2K^{(6)}(0)}{\mu_2(K)\hat{\psi}_8^{NS}} \right]^{1/9} n^{-1/9}.$$

Finally, as before, estimate ψ_4 using there kernel estimator $\hat{\psi}_4(\gamma_2(h))$ where,

$$\gamma_2(h) = \left[\frac{2K^{(4)}(0)\mu_2(K)\hat{\psi}_4(g_1)}{-\hat{\psi}_6(g_2)R(K)} \right]^{1/7} h^{5/7}.$$

Then, the selected bandwidth $h_{Solve,2}$ is found by solving for the value of h satisfying,

$$h = \left[\frac{R(K)}{\mu_2(K)^2 \hat{\psi}_4(\gamma_2(h))} \right]^{1/5} n^{-1/5},$$

which is the same as (2.7), but using different pilot bandwidths.

2.1.4 Further plug-in methods

Looking at other plug-in methods in recent literature, we can see that the main difference of these methods is in the selection of pilot bandwidth g. Apart form the methods discussed above, an interesting plug-in approach can be found in Härdle et al. (2012). This refined plug-in method starts with the Silverman (1986) rule-of-thumb bandwidth for Gaussian kernels given by

$$h_s = 1.06 \min\{1.34^{-1} \text{ IQR}, s_n\} n^{-1/5},$$

where IQR denotes the Interquartile Range of the sample and s_n denotes the sample standard deviation. Then, h_s is adjusted for Quartic kernels based on the idea of canonical kernels and equivalence bandwidths. This adjustment is done considering the slower optimal rate for second derivative estimation, from which the pilot bandwidth is derived as (see Heidenreich et al., 2013)

$$g = h_s 2.6226 n^{4/45}.$$

However, we are not including this method in our simulation study, and focus on the direct plug-in and solve-the-equation plug-in methods discussed earlier.

2.2 Cross-validation methods

By far, the most researched method for bandwidth selection is cross-validation (CV) (see Chiu, 1996) and all of its variants. The origin of this methodology dates back to the 1980's (see Bowman et al., 1984; Rudemo, 1982) but pseudo-likelihood cross-validation methods date back even further to 1974 in the statistical literature (see Habbema et al., 1974). Ever since the idea has been introduced, numerous modifications and improvements have been proposed mainly to overcome the lack of stability of these methods in some contexts (Chiu, 1991). Some of the improvements in cross-validation methods are modified cross-validation (MCV) (see Stute, 1992), one sided cross-validation (OSCV) (see Martinez-Miranda et al., 2006) and indirect cross-validation (ICV) (see Savchuk et al., 2010).

In principle, cross-validation methods try to estimate the optimal bandwidth $\hat{h}_{opt}$ by minimizing the integrated squared error of the considered estimator. Expanding the terms in (1.5), the integrated squared error (ISE) can be written as

$$\mathrm{ISE}(h) = \int_{\mathbb{R}} \hat{f}_h^2(x)dx - 2\mathbb{E}\big\{\hat{f}_h(X)\big\} + R(f). \tag{2.9}$$

Looking closely at the above equation, we can see that the first and second terms are dependent on the bandwidth and can be calculated or estimated from data. The third term is not dependent on bandwidth and can be ignored when interest lies in minimizing (2.9) with respect to h. Approaches to estimating the second term on the right hand side of (2.9) vary and lead to different versions of cross-validation.

2.2.1 Least squares cross-validation

Least squares cross-validation (LSCV) is one of the most popular CV methods and can be explained by three steps in general. First, a subset of the data are selected from the sample to estimate a model. Then, the remaining observations are used to asses the fitted model. This is repeated over many possible subsets of the initial data and the model with the best average fit or, often, minimum average "error" is selected. In the current setup, each of

these models correspond to a different choice of the bandwidth h, and estimated densities are constructed using a leave one out method (see Olga, 2009). We can rearrange (2.9) to give,

$$\text{ISE}(h) - R(f) = \int_{\mathbb{R}} \hat{f}_h^2(x)dx - 2\mathbb{E}\big\{\hat{f}_h(X)\big\}$$

Note that $R(f) = \int_{\mathbb{R}} f^2(x)dx$ doesn't depend on h. Taking the expected value of the above expression,

$$\text{MISE}(h) - R(f) = \mathbb{E}(\text{ISE}(h)) - R(f) = \mathbb{E}\bigg\{ \int_{\mathbb{R}} \hat{f}_h^2(x)dx - 2\mathbb{E}\big\{\hat{f}_h(X)\big\}\bigg\}.$$

We can see that the right side of the above is unknown because of its dependence on f. However, we can write an unbiased estimator for this expression as (see Heidenreich et al., 2013),

$$\text{LSCV}(h) = \int_{\mathbb{R}} \hat{f}_h^2(x)dx - 2\frac{1}{n}\sum_{i=1}^{n} \hat{f}_{-i}(X_i), \tag{2.10}$$

where

$$\hat{f}_{(-i)}(x) = \frac{1}{(n-1)}\sum_{j\neq i}^{n} K_h(x - X_j)$$

is the density estimate excluding X_i, hence called the leave-one-out density estimator. We can further simplify (2.10) as (see Appendix A.1),

$$\text{LSCV}(h) = \int_{\mathbb{R}} \hat{f}_h^2(x)dx - \frac{2}{n-1}\sum_{i=1}^{n} \hat{f}_h(X_i) - \frac{2K_h(0)}{(n-1)}.$$

This alternate expression is important for its computational convenience. The goal is then to choose h to minimize $\text{LSCV}(h)$. This selected bandwidth will be denoted as $\hat{h}_{\text{LSCV}}$. Note that in some instances, $\text{LSCV}(h)$ has more than one local minimum (see Hall & Marron, 1991).

Even though this method is simple to use, it is known to lack stability in some cases even when the sample size increases. Specifically, $\hat{h}_{\text{LSCV}}$ is highly variable. But due to its simplicity, this method has been used extensively and many researchers have proposed

modifications in an attempt to improve stability.

2.2.2 Indirect cross-validation

This method can be used select the bandwidth of any second order kernel estimator as described in Section 2.1.1. Indirect cross-validation (ICV) can be described in three steps. First, the bandwidth of a KDE using kernel L is investigated using LSCV. Denote the resulting minimizing bandwidth as $\hat{b}_{\text{LSCV}}$. Then, based on the assumption that the density function to be estimated is twice differentiable, the optimal bandwidth of a KDE using kernel K, denoted h_n, and optimal bandwidth of the KDE using kernel L, denoted b_n, can be shown to satisfy,

$$h_n = \left(\frac{R(K)\mu_2^2(L)}{R(L)\mu_2^2(K)}\right)^{1/5} b_n \equiv C b_n. \tag{2.11}$$

The ICV bandwidth to use with the KDE with kernel K is then defined as $\hat{h}_{\text{ICV}} = C\hat{b}_{\text{LSCV}}$. The key is that C can be calculated as it does not depend on any unknown quantities. Now in order to select L, consider the family of kernels $£ = \{L(.;\alpha,\sigma) : \alpha > 0, \sigma > 0\}$, such that for all u,

$$L(u;\alpha,\sigma) = (1+\alpha)\phi(u) - \frac{\alpha}{\sigma}\phi\Big(\frac{u}{\sigma}\Big), \tag{2.12}$$

where ϕ denotes the Gaussian kernel. Note that this reduces to the Gaussian kernel when $\sigma = 1$. Also, each member of £ is symmetric around 0 and satisfies

$$\mu_{2L} = \int_{\mathbb{R}} u^2 L(u)du = 1 + \alpha - \alpha\sigma^2 = 1 + \alpha(1-\sigma^2),$$

and is a second order kernel, except when $\sigma = \sqrt{(1+\alpha)/\alpha}$. Practically, α and σ values that have been recommended are

$$\alpha = 10^{3.390 - 1.093\log_{10}(n) + 0.025\log_{10}(n)^{3 - 0.00004\log-10(n)^6}}$$

and

$$\sigma = 10^{-0.58 + 0.386\ \log_{10}(n) - 0.012\ \log_{10}(n)^2}$$

when $100 \leq n \leq 500000$ (see Savchuk et al., 2010). Based on practical considerations, the simple choices of $\alpha = 2.42$ and $\sigma = \max(5.06, 0.149\, n^{3/8})$ have also been recommended. Note that, $\max(5.06, 0.149\, n^{3/8}) = 5.06$ for $n \leq 12094$.

2.2.3 Biased cross-validation

Biased cross-validation, introduced by Scott and Terrell (1987), has better asymptotic stability compared to least squares cross-validation. Here, the asymptotic MISE (AMISE) given in (1.7) is used instead of MISE as the basis for bandwidth selection. Specifically, an estimate $\widetilde{R(f'')}$ is used for the unknown $R(f'')$ in the BCV objective function,

$$\widetilde{R(f'')} = R(\hat{f}_h'') - (nh^5)^{-1}R(K'') = n^{-2}\sum\sum_{i\neq j}(K_h'' * K_h'')(X_i - X_j). \tag{2.13}$$

Since $\widetilde{R(f'')}$ is a crossvalidatory estimator (also based on a leave-one-out argument), this method has a strong relation to least squares cross-validation. Then,

$$\mathrm{BCV}(h) = (nh)^{-1}R(K) + \frac{1}{4}h^4\mu_2(K)^2\widetilde{R(f")} \tag{2.14}$$

is clearly an estimator of AMISE. Also, since the unknown $R(f'')$ is replaced by $\widetilde{R(f'')}$, this selector can be considered as a hybrid of cross-validation and plug-in bandwidth selection, as discussed in Section 2.1. As an estimator of h_{opt}, the minimizer of $\mathrm{BCV}(h)$, is denoted by $\hat{h}_{BCV}$. This bandwidth selector has a lower asymptotic variance but increased bias compared to $\hat{h}_{LSCV}$.

2.2.4 One-sided cross-validation

This is another method proposed to obtain a more stable smoothing parameter compared to ordinary cross-validation. The method was initially introduced for local linear smoothers, and then was extended to kernel smoothers. For a kernel density estimator $\hat{f}_{h,M}$ based on kernel M, the associated local linear estimator can be denoted as $\hat{f}_{h,M^*}$ with kernel M^*

given by (see Hart & Yi, 1998)

$$M^*(u) = \frac{\mu_2(M) - \mu_1(M)u}{\mu_0(M)\mu_2(M) - \mu_1^2(M)} M(u).$$

Applying the OSCV method, the kernel $M(u)$ is taken as $2K(u)1_{(-\infty,0)}$ in left-sided cross-validation and as $2K(u)1_{(0,\infty)}$ in right-sided cross-validation. This produces the left and right side kernels,

$$K_L(u) = \frac{\mu_2(K) + u\mu_1^*(K)}{\mu_2(K) - (\mu_1^*(K))^2} 2K(u)1_{(-\infty,0)},$$

$$K_R(u) = \frac{\mu_2(K) - u\mu_1^*(K)}{\mu_2(K) - (\mu_1^*(K))^2} 2K(u)1_{(0,\infty)},$$

where $\mu_1^*(K) = 2\int_0^\infty uK(u)du$ and $1_{(-\infty,0)}$, $1_{(0,\infty)}$ are indicator functions, based on the assumption that the kernel K is symmetric. Now, the left-sided cross-validation criterion OSCV_L is defined as

$$\text{OSCV}_L(h) = \int_{\mathbb{R}} \hat{f}^2_{h,K_L}(x)dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{h,K_L}(X_i). \tag{2.15}$$

Let $\hat{h}_L$ be the minimizer of the above. Then, it can be shown asymptotically that (see Mammen et al., 2011)

$$\hat{h}_{L,\text{OSCV}} = C_L \hat{h}_L, \tag{2.16}$$

where

$$C_L = \left(\frac{R(K)\mu_2^2(K_L)}{\mu_2^2(K)R(K_L)}\right)^{1/5}. \tag{2.17}$$

Similarly, the right-sided cross-validation criterion can be defined as

$$\text{OSCV}_R(h) = \int_{\mathbb{R}} \hat{f}^2_{h,K_R}(x)dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{h,K_R}(X_i), \tag{2.18}$$

and the right-OSCV bandwidth satisfies

$$\hat{h}_{R,\text{OSCV}} = C_R \hat{h}_R, \tag{2.19}$$

with C_R defined as in (2.17) with the obvious modification to use K_R. Note that (2.15) and (2.18) are not the leave-one-out criterion, but the authors explain that the impact of this is asymptotically negligible. If one wants to use the leave-one-out method for $\mathrm{OSCV}_L(h)$ the term $\hat{f}_{h,K_L}(x)$ should be replaced by $n(n-1)^{-1}\hat{f}_{h,K_L}(x)$. A similar change can be adopted for $\mathrm{OSCV}_R(h)$.

One-sided cross-validation doesn't yield good results in a local constant version, that is, when $K_L(u) = 2K(u)1_{(-\infty,0)}$ or $K_R = 2K(u)1_{(0,\infty)}$. This is due to the inferior rate of convergence of the one-sided local constant kernel density estimator. This is the reason to use local linear density estimation as the basis for this approach to bandwidth selection in Martinez-Miranda et al. (2006)

.

2.2.5 Do-validation

Do-validation is based on the idea of merging left and right-sided cross-validation (see Mammen et al., 2011). Consider the expressions for $\hat{h}_{L,\mathrm{OSCV}}$ and $\hat{h}_{R,\mathrm{OSCV}}$ derived in one-sided cross-validation. Then the do-validation selector $\hat{h}_{\mathrm{DO}}$ is given by

$$\hat{h}_{\mathrm{DO}} = \frac{1}{2}\left(\hat{h}_{L,\mathrm{OSCV}} + \hat{h}_{R,\mathrm{OSCV}}\right)$$

Note that OSCV_L and OSCV_R in (2.16) and (2.19) are not identical in general, in particular due to the difference in boundary conditions for data or an interval. Asymptotically, however they are equivalent. Do-validation seems to be a good and stable compromise, with similar asymptotic properties, but with better overall finite sample performance than many cross-validation methods.

2.2.6 Smoothed cross-validation

There is a similarity between smoothed cross-validation (SCV) and plug-in bandwidth selection methods, see Section 2.3. In this method, the integrated squared-bias component of

the MISE$\{\hat{f}_h\}$ is estimated from a kernel estimator with pilot bandwidth g. Unlike with direct plug-in methods, SCV uses the exact integrated squared-bias instead of its asymptotic approximation. Recall the expression for the MISE of a density estimator $\hat{f}_h$, given by

$$\text{MISE}[\hat{f}_h] = \mathbb{E}\int_{\mathbb{R}} \left\{\hat{f}_h(x) - f(x)\right\}^2 dx.$$

By changing the order of integration we get,

$$\text{MISE}[\hat{f}_h] = \int_{\mathbb{R}} \mathbb{E}\{\hat{f}_h(x) - f(x)\}^2 dx = \int_{\mathbb{R}} \text{MSE}(\hat{f}_h(x))dx. \tag{2.20}$$

Now, the mean squared error of $\hat{f}_h(x)$ (note this is for a fixed x) can be expressed as,

$$\text{MSE}[\hat{f}_h(x)] = n^{-1}\{(K_h^2 * f)(x) - (K_h * f)^2(x)\} + \{(K_h * f)(x) - f(x)\}^2. \tag{2.21}$$

Substituting expression (2.21) in (2.20), we get

$$\text{MISE}[\hat{f}_h] = n^{-1}\int_{\mathbb{R}} \{(K_h^2 * f)(x) - (K_h * f)^2(x)\}dx + \int_{\mathbb{R}} \{(K_h * f)(x) - f(x)\}^2 dx,$$

which can be further simplified to

$$\begin{aligned}\text{MISE}[\hat{f}_h] = (nh)^{-1}R(K) + (1 - n^{-1})\int_{\mathbb{R}} (K_h * f)^2(x)dx \\ -2\int_{\mathbb{R}} (K_h * f)(x)f(x)dx + \int_{\mathbb{R}} f(x)^2 dx.\end{aligned}$$

Ignoring the asymptotically negligible n^{-1} term, we can simplify the above to

$$\text{MISE}[\hat{f}_h] \approx (nh)^{-1}R(K) + \int_{\mathbb{R}} (K_h * f - f)(x)^2 dx. \tag{2.22}$$

Here, the integrated variance is approximated by the first term, whereas the second term is the integrated squared-bias (ISB) of $\hat{f}_h$. Now, by replacing f with a pilot estimator, we can

obtain the SCV objective function. For this, we introduce

$$\hat{f}_{g,L}(x) = n^{-1} \sum_{i=1}^{n} \frac{1}{g} L\left(\frac{x - X_i}{g}\right),$$

where g is a pilot bandwidth. Then the SCV function is defined as,

$$\mathrm{SCV}(h) = (nh^{-1})R(K) + \widehat{ISB}(h),$$

where,

$$\widehat{ISB}(h) = \int_{\mathbb{R}} \{K_h * \hat{f}_{g,L}(x) - \hat{f}_{g,L}(x)\}^2(x)dx.$$

There are several specific approaches to determine the SCV bandwidth selector (see Hall et al., 1992), the choice of the pilot bandwidth being how these methods differ (see M. Jones et al., 1991). In all cases, minimizing the above expression in terms of h leads to $\hat{h}_{\mathrm{SCV}}$. The dependence of h on the pilot bandwidth can be shown to be,

$$g = Cn^p h^m.$$

This relationship is suggested by Wand and Jones (1994), where C, p and m are constants defined to enhance the asymptotic performance of $\hat{h}_{\mathrm{SCV}}$. Let us now briefly consider the case where $K = L = \phi$, the Gaussian kernel. As a first step, compute the kernel estimates $\hat{\psi}_6(g_1)$ and $\hat{\psi}_{10}(g_2)$ where,

$$g_1 = \{2/(7n)\}^{1/9}\, 2^{1/2}\, \hat{\sigma} \quad \text{and} \quad g_2 = \{2/(11n)\}^{1/13}\, 2^{1/2}\, \hat{\sigma}.$$

Here g_1 and g_2 are normal scale estimates based on ψ_8 and ψ_{12}. The next step is to compute the kernel estimates $\hat{\psi}_4(g_3)$ and $\hat{\psi}_8(g_4)$, where

$$g_3 = [-6/\{(2\pi)^{1/2}\, \hat{\psi}_6(g_1)n\}]^{1/7} \quad \text{and} \quad g_4 = [-210/\{(2\pi)^{1/2}\, \hat{\psi}_{10}(g_2)n\}]^{1/11}$$

Finally, using the fact that for normal kernels the convolution operation can be simplified to

$$(\phi_\sigma * \phi_{\sigma'})(x) = \phi_{(\sigma^2+\sigma'^2)^{1/2}}(x),$$

hence, we can write

$$\mathrm{SCV}(h) = (nh)^{-1}(2\pi^{1/2})^{-1} + \sum_{i=1}^{n}\sum_{j=1}^{n}\left\{\phi_{(2h^2+2g^2)^{1/2}} - 2\phi_{(h^2+2g^2)^{1/2}} + \phi_{(2g^2)^{1/2}}\right\}\left(X_i - X_j\right), \quad (2.23)$$

where

$$g = \hat{C}n^{-23/45}h^{-2} \;\;,\;\; \hat{C} = \left\{441/(64\pi)\right\}^{1/18}(4\pi)^{-1/5}\,\hat{\psi}_4\,(g_3)^{-2/5}\,\hat{\psi}_8(g_4)^{-1/9}.$$

Minimizing (2.23) with respect to h leads to the $\hat{h}_{SCV}$ in the fully Gaussian case.

2.2.7 Further cross-validation methods

We finally mention a few cross-validation methods that we consider to be interesting but that are not considered later in any of our simulations. Many modified cross-validation concepts are found throughout the research literature, one such interesting method is discussed in Stute (1992), where the second term of the LSCV function (2.10) is approximated through its Hajek projection. Another approach, named modified cross-validation, has been introduced by Feluch and Koronacki (1992) and is better suited for time series data.

The partitioned cross-validation (PCV) concept has been introduced by Marron (1987), where the CV criterion is modified by splitting the sample into m sub-samples. Then a set of score functions (CV-score) for all sub-samples are defined and PCV is calculated by minimizing the average of these score functions. This method depends on the number m of sub-samples, on the choice of pilot bandwidth, and requires a relatively large sample size to create sub-samples with considerable size.

Another interesting cross-validation method is the original pseudo-likelihood cross-validation approach mentioned earlier, which has been proposed by Habbema et al. (1974) and Robert

(1976). This methods is based on the concept of using the maximum likelihood procedure to estimate the bandwidth h. Unfortunately, this method was criticized by many authors and deemed not suitable for density estimation.

2.3 Bootstrap based procedures

Bootstrap based inferential procedures are investigated extensively in the recent literature. The idea behind bootstrap bandwidth selection methods is to select the bandwidth using bootstrap estimates of MISE. In some cases, no re-sampling is required to implement the bootstrap bandwidth. The general idea of bootstrap based bandwidth selection is simple and straightforward, and relies on a bootstrap version of MISE(h) given by,

$$\text{MISE}^*(h) = \mathbb{E}_* \int_{\mathbb{R}} \left(\hat{f}_h^*(t) - \hat{f}_g(t)\right)^2 dt, \tag{2.24}$$

where E_* is an expectation taken over bootstrap samples $X_1^*, ..., X_n^*$, $\hat{f}_g(t)$ is taken as a density estimate which relies on the original sample $X_1, ..., X_n$ and a pilot bandwidth g and $\hat{f}_h^*(t)$ is an estimate based on bootstrap sample $X_1^*, ..., X_n^*$. A bandwidth h is then selected by minimizing MISE$^*(h)$. Bootstrap methods differ from each other through the choice of pilot bandwidth g and through the method used for generating the re-sampled bootstrap data $X_1^*, ..., X_n^*$ (see Cao et al., 1994).

2.3.1 Smoothed bootstrap with pilot bandwidth

Faraway and Jhun (1990) have considered a smoothed bootstrap procedure, which starts by calculating g from least-squares cross-validation. Then, MISE$^*(h)$ is approximated by sampling from $\hat{f}_g$. Specifically, based on B bootstrap samples, we define

$$\text{BMISE}(h) = \frac{1}{B} \sum_{j=1}^{B} \int_{\mathbb{R}} \left(\hat{f}_{h(j)}^*(t) - \hat{f}_g(t)\right)^2 dt, \tag{2.25}$$

where $\hat{f}^*_{h(j)}$ refers to the estimator constructed from the j-th bootstrap sample. Here, the bootstrap sample is taken from $\hat{f}_g$, where least-squares cross-validation is used to choose g from $X_1, ..., X_n$. Instead of using the exact expression for $\text{MISE}_*(h)$ an approximation is used by these authors. The authors do not provide asymptotic results for the resulting bandwidth selector, but report that the resulting bandwidth $\hat{h}_{\text{BF}}$ performs better than $\hat{h}_{\text{LSCV}}$.

Another approach was introduced by Cao-Abad and Gonzales-Manteiga (1990). Once again the bootstrap samples are drawn from a pilot estimate $\hat{f}_g$ and the pilot bandwidth g is selected via cross-validation. However, the authors use the exact expression of $\text{MISE}_*(h)$ stated below instead of the approximation given above;

$$\text{MISE}_*(h) = \frac{1}{nh}c_k - \frac{1}{n}\int_{\mathbb{R}}\left(\int_{\mathbb{R}} K(u)\hat{f}_g(x-hu)du\right)^2 dx + \frac{1}{n^2 g^6} \times \sum_{i,j}\int_{\mathbb{R}}\left(\int_{\mathbb{R}} K(u)(K_g(x-hu-X_i) - Kg(x-Xi))du\right)dx. \quad (2.26)$$

The main advantage is there actually is no need for re-sampling in some cases where $\text{MISE}_*(h)$ is explicitly known, for instance when using Gaussian kernels. The minimizer of the (2.26) is defined as the resulting bandwidth $\hat{h}_{B_C}$.

2.3.2 Smoothed bootstrap without pilot bandwidth

Here the pilot bandwidth is taken as $g = h$ in (2.24) and the exact value of $\text{MISE}_*(h)$ is computed. Consider a bootstrap sample with the density $\hat{f}_h$. Then $\text{MISE}_*(h)$ for the Gaussian kernel can be computed as (see Taylor, 1989)

$$\text{MISE}_*(h) = \frac{1}{2n^2 h(2\pi)^{1/2}}\left[\sum_{i,j} exp\left\{-\frac{(X_i - Xj)^2}{8h^2}\right\} - \frac{4}{3^{1/2}}\sum_{i,j} exp\left\{-\frac{(X_j - X_i)^2}{6h^2}\right\} + 2^{1/2}\sum_{i,j} exp\left\{-\frac{(X_j - Xi)^2}{4h^2}\right\} + n2^{1/2}\right]. \quad (2.27)$$

The bandwidth $\hat{h}_{SB}$ obtained by minimizing (2.27) was found to behave well in repeated simulation studies (see Taylor, 1989). Once again, we see from (2.27) that re-sampling is not required for the estimation of $\text{MISE}_*(h)$. However, as $h \to \infty$, $\text{MISE}_*(h) \to 0$. Since $\text{MISE}_*(h)$ doesn't have a finite minimum, it is not a fully suitable estimator of $\text{MISE}(h)$ when considering bandwidth selection. In practice, this issue can be circumvented since a finite interval is considered in the search for an optimal bandwidth. When the value of h_{BT} was investigated through numerical optimization, it was found that the asymptotic efficiency of this bandwidth selector is depending on the interval of search. The smaller the interval of search, the higher the efficiency of the obtained h_{BT}. When the interval is chosen large, the efficiency was seen to reduce rapidly in some cases (see Cao-Abad, 1990).

2.3.3 Further bootstrap methods

The main difference between bootstrap methods is how the pilot bandwidth g is chosen to generate bootstrap samples. Apart from the methods discussed above, Hall (1990) has proposed to use the empirical distribution to draw bootstrap samples of size $m < n$, where $m \simeq n^{1/2}$ and $h = g(m/n)^{1/5}$. Then MISE^* has been minimized with respect to g. But this approach has been proven to be unstable by Cao et al. (1994) for mixtures of normal distributions. Hence, this method is no considered in our simulation study, which will focus on the smoothed bootstrap with pilot bandwidth.

Chapter 3

Order selection for Bernstein Density Estimation (BDE)

3.1 The Bernstein density estimator

When it comes to nonparametric density estimation, Bernstein polynomials and Bernstein density estimation play an increasing role, especially when the function is supported on $[0, 1]$. Consider a series of independent and identically distributed random variables $(X_1, X_2, ...)$ with common distribution function F and density f supported on the unit interval. Then, the Bernstein estimator of order $m > 0$ of the distribution F is given by (see Leblanc, 2012)

$$\hat{F}_{m,n}(x) = \sum_{k=0}^{m} F_n(k/m) P_{k,m}(x). \tag{3.1}$$

Similar to (1.9), the polynomials $P_{k,m} = \binom{m}{k} x^k (1-x)^{m-k}$ are binomial probabilities and the sample distribution F_n is constructed from the first n observations $(X_1, X_2, ..., X_n)$. Taking the derivative of the Bernstein estimator $\hat{F}_{m,n}$ with respect to x, we get

$$\frac{d}{dx}\hat{F}_{m,n}(x) = m \sum_{k=0}^{m-1} [F_n([k+1]/m) - F_n(k/m)] P_{k,m-1}(x) = \hat{f}_{m,n}(x), \tag{3.2}$$

the Bernstein density estimation of order m. Note $\hat{F}_{m,n}$ and $\hat{f}_{m,n}$ are both polynomials, respectively of degree m and $m-1$. The above equation can be expressed in an alternate form as

$$\hat{f}_{m,n}(x) = \sum_{k=0}^{m-1} [F_n([k+1]/m) - F_n(k/m)]\beta_{k+1,m-k}(x), \tag{3.3}$$

where $\beta_{a,b}(x)$ is the Beta density with parameters a and b given by

$$\beta_{a,b}(x) = \begin{cases} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} & \text{for } x \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$$

From the above representation, we can see that the estimator $\hat{f}_{m,n}$ can be represented as a mixture of Beta densities with data driven weights.

An interesting property of $\hat{f}_{m,n}$ is that, for any value of m, it provides estimates that are genuine density functions on the interval $[0,1]$. This Bernstein estimator has been introduced by Vitale (1975) and further studied by Babu et al. (2002) and Leblanc (2010), amoung others. It has been shown that $\hat{f}_{m,n}$ is an asymptotically normal and consistent estimator which achieves an optimal convergence rate of $n^{-4/5}$ in the mean squared error (MSE) when the underlying density f is twice differentiable and m is chosen proportional to $n^{2/5}$.

An expression for the mean integrated squared error (MISE) of $\hat{f}_{m,n}$ can be derived under the assumption that f admits two continuous and bounded derivatives on $[0,1]$(see Leblanc, 2010). That is, as $n \to \infty$,

$$\text{MISE}[\hat{f}_{m,n}] = \frac{m^{1/2}C_1}{n} + \frac{C_2}{m^2} + o\left(\frac{m^{1/2}}{n}\right) + o(m^{-2}), \tag{3.4}$$

where

$$C_1 = \int_0^1 f(x)\psi(x)dx \quad \text{and} \quad C_2 = \int_0^1 \Delta_1^2(x)dx.$$

Here $\psi(x) = (4\pi x(1-x))^{-1/2}$ and Δ_1 is defined as

$$\Delta_1(x) = \frac{1}{2}[(1-2x)f^{'}(x) + x(1-x)f^{''}(x)]. \tag{3.5}$$

As f is the unknown density to be estimated, the first and second derivatives $f^{'}$ and $f^{''}$ are also unknown. Nevertheless, considering the asymptotically optimal choice of m, which can be shown to be $m_{opt} = [4C_2/C1]^{2/5}n^{2/5}$, we get that the smallest achievable MISE for the Bernstein density estimator is

$$\text{MISE}[\hat{f}_{m_{opt},n}] = \frac{5(4C_1^4C_2)^{1/5}}{4n^{4/5}} + o(n^{-4/5}).$$

We note that $h = 1/m$ can be considered as the bandwidth of the Bernstein estimator in a natural way. From (3.2) above, we can conclude that the Bernstein density estimator $\hat{f}_{m,n}$ has integrated bias of $O(h)^2$, which is much larger compared to that of kernel estimators. However, the integrated variance of $\hat{f}_{m,n}$ is found to be $O(1/nh^{1/2})$, which is much smaller than standard kernel estimators. An interesting approach to correct for bias is proposed in Leblanc (2010), where the following bias-corrected Bernstein estimator is introduced

$$\hat{f}_{m,M,n} = \frac{m}{m-M}\hat{f}_{m,n}(x) - \frac{M}{m-M}\hat{f}_{M,n(x)}, \tag{3.6}$$

where $m > M$ and $\hat{f}_{M,n}$ and $\hat{f}_{m,n}$ are Bernstien estimators of f of different orders. Also note that $\hat{f}_{m,M,n}$ is constructed from the weighted sum of these two estimators, where the two weights $\frac{m}{m-M}$ and $\frac{-M}{m-M}$ are summing to unity, although the first weight is larger than one and the other is negative.

Kakizawa (2004) proposed a very different approach for density estimation also relying on Bernstein polynomials. His method is also proposed for estimating densities with compact support $[0,1]$. Kakizawa introduced three estimators developed using an approach also

based on boundary kernels. First, let

$$max_{t\in[0,1]}f(t) = C_0 \quad , \quad max_{t\in[0,1]}f'(t) = C_1 \quad \text{and} \quad max_{t\in[0,1]}f''(t) = C_2,$$

where C_0, C_1 and C_2 are finite positive constants. Also, the indicator of a set A is denoted by $\varkappa_A(.)$. Now, consider a class $\kappa[a_1, a_2]$ of bounded functions on the interval $[a_1, a_2]$, specifically, $K \in \kappa[a_1, a_2]$ are continuous, except possibly at a finite number of points, and satisfy

$$\int_{a1}^{a2} K(u)du = 1 \quad , \quad \int_{a1}^{a2} uK(u)du = 0 \quad \text{and} \quad \int_{a1}^{a2} u^2K(u)du \neq 0.$$

Then the boundary kernel κ_x is defined by (see Kakizawa, 2004; Müller, 1991)

$$\kappa_x(.) \equiv \begin{cases} K(.) \in \kappa[-1, 1], & x \in [h, 1-h] \\ K^+_{x/h}(.) \in \kappa[-1, x/h], & x \in [0, h) \\ K^-_{(x-1)/h}(.) \in \kappa[(x-1)/h, 1], & x \in (1-h, 1], \end{cases} \tag{3.7}$$

where $h \geq 0$ is a bandwidth. It is further assumed that the continuity conditions $K^+_1(.) = K^-_{-1}(.) = K(.)$ are satisfied and the kernel is uniformly bounded as per

$$\sup_{q\in[0,1],u\in R}|K^+_q(u)| \leq C \quad and \quad \sup_{q\in[0,1],u\in R}|K^-_q(u)| \leq C.$$

Using this boundary kernel, the following modified boundary kernel estimator is defined

$$\hat{f}^*_{K,h}(x) = \frac{1}{nh}\sum_{i=1}^{n} \kappa_x\left(\frac{x - X_i}{h}\right), \quad x \in [0, 1]. \tag{3.8}$$

The asymptotic MSE of this estimator satisfies, as $n \to \infty$ and $h \to 0$,

$$\text{MSE}[\hat{f}^*_{K,h}(x)] = \frac{1}{nh}R(\kappa_x)f(x) + \frac{h^4}{4}\mu_2(\kappa_x)f''(x)^2 + o((nh)^{-1} + h^4).$$

One simple family of boundary kernels κ_x can be formed as a linear combination of the

original kernel K and another function denoted L, in such a way that the resulting kernel is satisfying all conditions defined in (3.7). If we select $L(u) = uK(u)$, then the left boundary kernel can be given as,

$$K_q^+(u) = \frac{\mu_{2,q}^+(K) - u\mu_{1,q}^+(K)}{\mu_{0,q}^+(K)\mu_{2,q}^+(K) - \mu_{1,q}^+(K)^2} K(u)\mathscr{K}_{[-1,q]}(u),$$

and the right boundary can be given as

$$K_{-q}^-(u) = \frac{\mu_{2,-q}^-(K) - u\mu_{1,-q}^-(K)}{\mu_{0,-q}^-(K)\mu_{2,-q}^-(K) - \mu_{1,-q}^-(K)^2} K(u)\mathscr{K}_{[-q,1]}(u),$$

for $q \in [0,1]$, where $\mathscr{K}_{[-1,q]}(u)$ and $\mathscr{K}_{[-q,1]}(u)$ are indicator functions as introduced above. A main disadvantage of these two kernels is they can take negative values. Specifically when the original kernel $K \in \kappa_2[-1,1]$ is symmetric and non-negative, then the left boundary kernel will take negative values for $-1 \le u < \int_{-1}^{q} u^2K(u)du / \int_{-1}^{-q} uK(u)du < 0, q \in [0,1]$. Another method to normalize $\hat{f}_{K,h}^*(x)$ can be found in Kakizawa (2004) referred to as the renormalized boundary kernel, but we do not discuss this approach here.

Based on the boundary kernel estimators defined in (3.8), Kakizawa (2004) defines three generalized Bernstein-based boundary kernel estimators. They are

$$\hat{f}_{B_1,h,m,\gamma}(x) = \sum_{j=0}^{m-1} \hat{f}_{K,h}^*\left(\frac{j+\gamma}{m}\right) P_{j,m-1(x)}, \tag{3.9}$$

$$\hat{f}_{B_2,h,m}(x) = \sum_{j=0}^{m} \hat{f}_{K,h}^*\left(\frac{j}{m}\right) P_{j,m(x)}, \tag{3.10}$$

$$\hat{f}_{B_3,h,m}(x) = \sum_{j=0}^{m-1} \hat{f}_{K,h}^*\left(\frac{j+x}{m}\right) P_{j,m-1}(x), \tag{3.11}$$

where γ is a fixed constant satisfying $0 \le \gamma \le 1$. The Vitale estimator introduced in (3.2) falls under the class $\hat{f}_{B_1,h,m,\gamma}(x)$ as the special case where $h = 1/(2m)$ and $\gamma = 1/2$ and where K is the uniform kernel. How to apply cross-validation principles to the Data-driven selection of smoothing parameters for these estimators will be discussed in the next section.

3.2 Cross-validation and related methods

In this section, we discuss how to select the order of Bernstein estimators using cross-validation principles. Mainly, the Vitale (1975) estimator and the boundary-kernel based estimators introduced by Kakizawa (2004) will be investigated. Specifically, we consider the application of cross-validation principles to existing BDEs. For notational convenience, in what follows we denote the original Vitale estimator given in (3.3) as $\hat{f}_V$, and the three Kakizawa estimators given in (3.9), (3.10) and (3.11) as $\hat{f}_{B_1}, \hat{f}_{B_2}$ and $\hat{f}_{B_3}$, respectively. We will later consider using the order selectors in a scheme that is similar to Do-validation to introduce different order selectors for the original Vitale BDE.

3.2.1 Applying cross-validation principles to Bernstein density estimators

First, recall that the basic Bernstein density estimator can be written as,

$$\hat{f}_{V,m}(x) = m \sum_{k=0}^{m-1} [F_n([k+1]/m) - F_n(k/m)] P_{k,m-1}(x),$$

where F_n is the empirical distribution function. Consider the vector

$$\mathbf{F}_m = \begin{pmatrix} F_n(1/m) \\ F_n(2/m) - F(1/m) \\ \vdots \\ 1 - F_n(1-1/m), \end{pmatrix}, \tag{3.12}$$

and the $m \times M$ matrix $\mathbf{A}_{m,M}$ with entries

$$a_{kl} = \frac{\binom{m-1}{k}\binom{M-1}{l}}{\binom{m+M-2}{k+l}}, \tag{3.13}$$

for $k = 0, 1, ..., m-1$ and $l = 0, 1, ..., M-1$. Let also $\lfloor x \rfloor$ be the largest integer smaller than x and define the integer sequence $k_i = \lfloor mX_i \rfloor$. Note that this implies $X_i \in (k_i/m, (k_i+1)/m]$.

Table 3.1. *Matrix of cross-validated Bernstein estimator over many samples in a simulation study*

Sample i	1	2	$\cdots$	M
1	$\mathrm{CV}_1(\hat{f}_{V,1})$	$\mathrm{CV}_1(\hat{f}_{V,2})$		$\mathrm{CV}_1(\hat{f}_{V,M})$
2	$\mathrm{CV}_2(\hat{f}_{V,1})$	$\mathrm{CV}_2(\hat{f}_{V,2})$		$\mathrm{CV}_2(\hat{f}_{V,M})$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
i	$\mathrm{CV}_i(\hat{f}_{V,1})$	$\mathrm{CV}_i(\hat{f}_{V,2})$		$\mathrm{CV}_i(\hat{f}_{V,M})$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
N	$\mathrm{CV}_N(\hat{f}_{V,1})$	$\mathrm{CV}_N(\hat{f}_{V,2})$		$\mathrm{CV}_N(\hat{f}_{V,M})$

Then, the cross-validation function of $\hat{f}_V$ can be written as (see Appendix A),

$$CV_V(m) = \frac{m^2}{2m-1}\mathbf{F}^t{}_m\mathbf{A}_{m,m}\mathbf{F}_m - \frac{2}{(n-1)}\left[\sum_{i=1}^{n}\hat{f}_{m,n}(X_i) - \frac{1}{n}\sum_{i=1}^{n}\beta_{k_i+1,m-k_i}(X_i)\right], \qquad (3.14)$$

and the order to be selected is $m_V = \mathrm{argmin}_m CV_V(m)$ (see Leblanc, 2010). In Chapter 4, we present the results of a simulation study where we cross-validate this estimator over many samples $i = 1, ..., N$ and for many orders $m = 1, ..., M$. For this, we construct a matrix as given in Table 3.1. In that Table, each row corresponds to a specific sample, the CV criterian being calculated for many orders, and each column corresponds to a specific order. Further if we minimize along a row we can find the minimum cross-validated value for a specific sample. Then, we can select the corresponding order m related to the identified minimum value. So by taking the row minimizer for each row $i = 1, 2, ..., N$ and by selecting the corresponding order we can construct a vector $\left(\hat{m}_{V,i}\right)$ of orders for Vitale's estimator. We note that m being an integer, typical numerical minimization technique can not be used directly. Similarly, we can obtain orders matrices by cross-validating Kakizawa's estimators.

To get cross-validated expressions for Kakizawa's first, second and third estimators, we followed the steps outlined below. First, consider a generic kernel estimator as $\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)$. Then as was mentioned previously, a leave-one-out version of this estima-

tor is given by

$$\hat{f}_{(-i)}(x) = \frac{1}{(n-1)h}\sum_{j\neq i} K\left(\frac{x-X_j}{h}\right).$$

This estimator can be further written as (refer to the proof in appendix A),

$$\hat{f}_{(-i)}(x) = \frac{n}{(n-1)}\hat{f}(x) - \frac{1}{(n-1)h}K\left(\frac{x-X_i}{h}\right).$$

This idea allows us to define the leave-one-out version of the boundary kernel estimator defined earlier in (3.8) as

$$\hat{f}_{K,h(-i)}(x) = \frac{1}{(n-1)h}\sum_{i\neq i} K_x\left(\frac{x-X_j}{h}\right).$$

Then, using (3.9, 3.10, 3.11), we can define the leave-one-out Bernstein-based kernel estimators as

$$\hat{f}_{(-i)B_1}(x) = \sum_{k=0}^{m-1} \hat{f}_{(-i)}\left(\frac{k+\gamma}{m}\right)P_{k,m-1}(x_i), \tag{3.15}$$

$$\hat{f}_{(-i)B_2}(x) = \sum_{k=0}^{m} \hat{f}_{(-i)}\left(\frac{k}{m}\right)P_{k,m}(x), \tag{3.16}$$

$$\hat{f}_{(-i)B_3}(x) = \sum_{k=0}^{m-1} \hat{f}_{(-i)}\left(\frac{k+x}{m}\right)P_{k,m-1}(x), \tag{3.17}$$

where $P_{k,m-1}(x)$ is defined as in (3.1). Now, consider (3.15). We can rearrange the equation to give (refer to the proof in Appendix A)

$$\hat{f}_{(-i)B_1}(x) = \frac{n}{n-1}\hat{f}_{B_1}(x) - \frac{1}{(n-1)h}\kappa_x\left(\frac{\frac{k_i^*+\gamma}{m} - X_i}{h}\right)P_{k_i^*,m-1}(X_i), \tag{3.18}$$

where $k_i^* = [mX_i - \gamma + \frac{1}{2}]$ for $i = 1, 2, ..., n$. We can then derive the expression for LSCV_{B_1} as

$$\text{LSCV}_{B_1} = \int \hat{f}_{B_1}^2(x)dx - \frac{2}{n}\sum_{i=1}^{n}\hat{f}_{(-i)B_1}(X_i) \tag{3.19}$$

$$= \int \hat{f}_{B_1}^2(x)dx - \frac{2}{n-1}\sum_{i=1}^{n}\hat{f}_{B_1}(X_i) + \frac{4m}{n(n-1)}\sum_{i=1}^{n}K\left(\frac{\frac{k_i^*+\gamma}{m} - X_i}{h}\right)P_{k_i^*,m-1}(X_i). \tag{3.20}$$

We can finally simplify expression (3.20) to

$$\text{LSCV}_{B_1} = \frac{1}{(2m-1)}\mathbf{F}_{1,m}^t\mathbf{A}_{m,m}\mathbf{F}_{1,m} - \frac{2}{n-1}\sum_{i=1}^{n}\hat{f}_{B_1}(X_i) + \frac{4m}{n(n-1)}\sum_{i=1}^{n}K\left(\frac{\frac{k_i^*+\gamma}{m} - X_i}{h}\right)P_{k_i^*,m-1}(X_i), \tag{3.21}$$

where $\mathbf{A}_{m,m}$ is defined in (3.13) and

$$\mathbf{F}_{1,m} = \begin{pmatrix} \hat{f}_{B_1}(\frac{1+\gamma}{m}) - \hat{f}_{B_1}(\frac{\gamma}{m}) \\ \vdots \\ \hat{f}_{B_1}(1) - \hat{f}_{B_1}(\frac{m-1+\gamma}{m}) \end{pmatrix}.$$

Note that expression (3.21) can be adopted easily to cross-validate Kakizawa's second estimator. Similarly, as above, from (3.16), we can show that

$$\hat{f}_{(-i)B_2}(x) = \frac{n}{n-1}\hat{f}_{B_2}(x) - \frac{1}{(n-1)h}\kappa_x\left(\frac{\frac{k_i^*}{m} - X_i}{h}\right)P_{k_i^*,m-1}(X_i),$$

where $k_i^* = [mX_i + \frac{1}{2}]$ for $i = 1, 2, ..., n$. Now, we can dervive the expression for LSCV_{B_2}

$$\text{LSCV}_{B_2} = \frac{1}{(2m-1)}\mathbf{F}_{2,m}^t\mathbf{A}_{m,m}\mathbf{F}_{2,m} - \frac{2}{n-1}\sum_{i=1}^{n}\hat{f}_{B_2}(X_i) + \frac{4m}{n(n-1)}\sum_{i=1}^{n}K\left(\frac{\frac{k_i^*}{m} - X_i}{h}\right)P_{k_i^*,m}(X_i), \tag{3.22}$$

once again $\mathbf{A}_{m,m}$ is defined in (3.13) and

$$\mathbf{F}_{2,m} = \begin{pmatrix} \hat{f}_{B_2}(\frac{1}{m}) - \hat{f}_{B_2}(0) \\ \vdots \\ \hat{f}_{B_2}(1) - \hat{f}_{B_2}(\frac{m-1}{m}) \end{pmatrix},$$

Similarly, for Kakizawa's third estimator we use

$$\text{LSCV}_{B_3} = \int \hat{f}_{B_3}(x)dx - \frac{2}{n-1}\sum_{i=1}^{n}\hat{f}_{B_3}(X_i) + \frac{4m}{n(n-1)}\sum_{i=1}^{n}K\left(\frac{\frac{k^*_{i1}}{m} - (\frac{m-1}{m})X_i}{h}\right)P_{k^*_{i2},m-1}(X_i),$$

where $k^*_{i1} = [mX_i - \gamma + \frac{1}{2}]$ and $k^*_{i2} = [(m-1)X_i + \frac{1}{2}]$ for $i = 1, 2, ..., n$. For all three estimators we use $h = 1/2m$.

In our simulation study, we cross-validate these three estimators over many samples, as we did for the basic Bernstein estimator $\hat{f}_V$. To do this, we obtained a matrix of cross-validated values similar to Table 3.1 for each of the three estimators. Then, we minimize along a row so that we can find the cross-validated value of each sample. Following this, we can obtain three vectors $\left(\hat{m}_{B_1,i}\right)$, $\left(\hat{m}_{B_2,i}\right)$, $\left(\hat{m}_{B_3,i}\right)$ with $\hat{m}_i$ of length N. Now, we can evaluate the performance of these order selector, which will be discussed in the next Section.

3.2.2 Approximating MISE for Bernstein density estimators

As discussed in Chapter 2, MISE can be used to compare the performance of these estimators. It is aso useful in simulation to identify the optimal order m of Bernstein estimator for a given problem. Now, recall that the ISE of $\hat{f}_{V,m}$ is given by

$$\text{ISE}(\hat{f}_{V,m}) = \int_0^1 (\hat{f}_{V,m}(x) - f(x))^2 dx,$$

and its MISE is defined as

$$\text{MISE}(\hat{f}_{V,m}) = \mathbb{E}(\text{ISE}(\hat{f}_{V,m})),$$

Table 3.2. *Matrix of ISE values of the Bernstein estimator over many samples in a simulation study*

Sample i	1	2	$\cdots$	M
1	$\text{ISE}_1(\hat{f}_{V,1})$	$\text{ISE}_1(\hat{f}_{V,2})$		$\text{ISE}_1(\hat{f}_{V,M})$
2	$\text{ISE}_2(\hat{f}_{V,1})$	$\text{ISE}_2(\hat{f}_{V,2})$		$\text{ISE}_2(\hat{f}_{V,M})$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
i	$\text{ISE}_i(\hat{f}_{V,1})$	$\text{ISE}_i(\hat{f}_{V,2})$		$\text{ISE}_i(\hat{f}_{V,M})$
$\vdots$		$\vdots$		$\vdots$
N	$\text{ISE}_N(\hat{f}_{V,1})$	$\text{ISE}_N(\hat{f}_{V,2})$		$\text{ISE}_N(\hat{f}_{V,M})$

the expectation being taken over the sample $X_1, ..., X_n$. As a result, MISE can be approximated using a simple Monte Carlo simulation scheme. Specifically, we compute ISE values for samples $i = 1, 2, ..., N$ and repeat this for different values of $m = 1, 2, ..., M$, generating a matrix of ISE values as per Table 3.2. The MISE of $\hat{f}_{V,m}$ can be approximated, for different values of m by considering the average of the columns of Table 3.2, that is

$$\frac{1}{N}\sum_{i=1}^{N}\text{ISE}_i(\hat{f}_{V,m}) \simeq \text{MISE}(\hat{f}_{V,m}). \tag{3.23}$$

Minimizing (3.23) with respect to m, we can obtain an approximation of the best MISE value and associated m_{MISE}, the optimal order for the Bernstein estimator according to MISE. In order to assess any order selection method, we can compare their order matrices with this optimum value, which needs to be found for each estimator. We know that the ISE matrix given in Table 3.2 displays on its first row the values of $ISE_1(m)$ for m=1,2,...,M for the first sample. Now, let us consider Vitale's order selector and its order vector $\left(\hat{m}_{V,i}\right)$ for $i = 1, 2, ..., N$, where the order for the first sample is $\hat{m}_{V,1}$. This value can be found from the orders $1, ..., M$ given in the ISE matrix and the corresponding ISE value for the first sample can be identified, ie. $\text{ISE}(\hat{m}_{V,1})$. Similarly, the corresponding ISE values for all entries of the order vector $\left(\hat{m}_{V,i}\right)$ can found for samples $i = 1, 2, ..., N$. This will result in an ISE vector $(\text{ISE}(\hat{m}_{V,i}))$ for Vitale's order selector. By taking the average of these values, we can approximate the MISE of the Vitale's order selector,ie. $\text{MISE}\hat{f}_{\hat{m}_V} \simeq \frac{1}{N}\sum_{i=1}^{N}\text{ISE}_i(\hat{f}_{V,\hat{m}_{V,i}})$. This

process can be repeated for other order selectors and we can find MISE values MISE($\hat{m}_{B_1,m}$), MISE($\hat{m}_{B_2,m}$), MISE($\hat{m}_{B_3,m}$) for each order selector. Having these MISE values we can compare the performance of the Vitale estimator $\hat{f}_V$ based on using each order selection strategy with the optimum MISE. This is the methodology we follow in our simulation.

In the next section, we follow a similar process as that used above and turn to using bootstrapping principles, to tackle the order selection problem for Bernstein estimators.

3.3 Bootstrap method

Bootstrapping is a statistical procedure that generally requires one to resample from a single dataset to generate many simulated samples. In this section, we discuss how to use Bootstrapping principles for selecting the optimal order $\hat{m}_{opt}$ of a Bernstein density estimator. First, let $\hat{f}_{m_0}$ denote the Bernstein density estimator constructed from the sample $X_1, ..., X_n$ with a pilot order m_0. Now, we denote by $X^*_{i,1}, ..., X^*_{i,n}$ is the i^{th} bootstrap sample and $\hat{f}^*_{i,m}$ is the BDE of order m constructed from this bootstrap sample. Details of the choice of m_0 and on the generalization of bootstrap samples are given below. We define the Bootstrap estimate of MISE(m) for the Bernstein estimator with order m as

$$\text{Boot}(m) = \frac{1}{B}\sum_{i=1}^{B}\int_0^1 (\hat{f}^*_{i,m}(x) - \hat{f}_{m_0}(x))^2 dx \tag{3.24}$$

$$= \frac{1}{B}\sum_{i=1}^{B}\text{ISE}^*_i(m), \tag{3.25}$$

where B denotes the number of bootstrap samples. Now, the terms from the above sum can be expanded to get

$$\text{ISE}^*_i(m) = \int_0^1 {\hat{f}^*_{i,m}}^2(x)dx + \int_0^1 \hat{f}^2_{m_0}(x)dx - 2\int_0^1 \hat{f}^*_{i,m}(x)\hat{f}_{m_0}(x)dx.$$

Table 3.3. *Matrix of ISE values for the bootstrapped Bernstein estimator over many samples in a simulation study*

Bootstrap sample j	1	2	$\cdots$	M	
1	$\text{ISE}_1^*(\hat{f}_{V,1})$	$\text{ISE}_1^*(\hat{f}_{V,2})$		$\text{ISE}_1^*(\hat{f}_{V,M})$	
2	$\text{ISE}_2^*(\hat{f}_{V,1})$	$\text{ISE}_2^*(\hat{f}_{V,2})$		$\text{ISE}_2^*(\hat{f}_{V,M})$	
$\vdots$	$\vdots$	$\vdots$			$\vdots$
j	$\text{ISE}_j^*(\hat{f}_{V,1})$	$\text{ISE}_j^*(\hat{f}_{V,2})$		$\text{ISE}_j^*(\hat{f}_{V,M})$	
$\vdots$	$\vdots$	$\vdots$		$\vdots$	
B	$\text{ISE}_B^*(\hat{f}_{V,1})$	$\text{ISE}_B^*(\hat{f}_{V,2})$		$\text{ISE}_B^*(\hat{f}_{V,M})$	

Using $\mathbf{F}_{1,m}$ and $\mathbf{A}_{m,M}$ as defined in (3.12) and (3.13) and following the approach outlined in Appendix A.3, we can further express the above equality as

$$\text{ISE}_i^*(m) = \frac{m^2}{(2m-1)}\mathbf{F}_{i,m}^{*t}\mathbf{A}_{m,m}F_{i,m}^* + \frac{m_0^2}{(2m_0-1)}\mathbf{F}_{m_0}^t\mathbf{A}_{m_0,m_0}\mathbf{F}_{m_0} - 2\frac{mm_0}{(m+m_0-1)}\mathbf{F}_{i,m}^{*t}\mathbf{A}_{m,m_0}\mathbf{F}_{m_0}. \tag{3.26}$$

Minimizing (3.24) with respect to m leads to $\hat{m}_{\text{Boot}}$, the bootstrapped estimate of $\hat{m}_{\text{opt}}$. Our goal is to obtain an optimal order by bootstrapping the Bernstein estimator. In our simulation we assume that $m_{0,i}$ are the pilot orders where $m_{0,i} = \left(\hat{m}_{V,i}\right)$ for $i = 1, 2, ..., N$, that is by using the cross-validated value of m mentioned earlier. For each sample, we can generate B bootstrap samples form $\hat{f}_{m_0}$ using the fact that it can be viewed as a mixture of Beta distributions (acting as if this was the true density of the sample). For each bootstrap sample, we calculate the squared distance between $\hat{f}_m^*$ constructed from the bootstrap sample and $\hat{f}_{m_0}$ constructed from the original data. This is an approximation for the ISE. We calculate this distance for each $m_{0,i}$. Then we move on to the next bootstrap samples and repeat this calculation. Likewise we repeat this process over all bootstrap samples to obtain a matrix of bootstrapped ISE vlaued given in Table 3.3.

By taking the column average of Table 3.3 we can calculate Boot(m), thus approximating the MISE based on bootstrapping for one observed sample. By minimizing these approximate MISE values, we can find the corresponding order m for minimum MISE. This

is the optimal order for sample 1, whereas we can repeat this for $i = 1, 2, ..., N$. Note that N is the size of our Monte Carlo simulation, while B is the number of bootstrap samples in each case. So by taking the column average and minimizing the averages we can find the approximation for minimum MISE and corresponding optimum order m for one sample and should be repeated for each sample, leading to the bootstrapped orderes $\left(\hat{m}_{Boot,i}\right)$ for $i = 1, 2, ..., N$.

After obtaining the order vector $\left(\hat{m}_{\text{Boot},i}\right)$ from bootstrapping, we try to evaluate the performance of these orders. To do this, follow the evaluation criteria introduced in Section 3.2.2. Recall the ISE matrix given in Table 3.2. By selecting the corresponding ISE for each sample, we get ISE($\hat{f}_{V,\hat{m}_{Boot}}$). Then, by averaging these ISE values we get the $\text{MISE}_{\text{Boot}}$. In the next Chapter, we further discuss cross-validation and bootstrapping and introduce other order selection strategies. We also briefly study these methods through simulation.

Chapter 4

Smoothing parameter selection ; Simulation results and Discussion

In this chapter, we first study some existing bandwidth selection methods through conducting a simulation study. We compare the performance of these methods by means of the performance of these selectors in repeated sampling. We also identify the most satisfactory bandwidth selection methods through an analysis of their strengths and weaknesses. We then move on to study the performance of these approaches to the selection of smoothing parameters in the specific context of Bernstein polynomial density estimation (BDE). Finally, we introduce new approaches to the selection of the order of BDEs by applying and adapting principles identified in existing KDE methods. We also compare the performance of these new approaches with the commonly used ones.

4.1 Bandwidth selection in KDE

We discussed many bandwidth selection methods in Chapter 2, whereas some were chosen for simulation study based on their performance. Specifically, we implemented the following methods in our study: LSCV (lease-squares cross-validation), ICV (indirect cross-validation), BCV (biased cross-validation), DO (do-validation), BOOT (bootstrap method),

DPI (direct pulg-in), SCV (smoothed cross-validation) and STE (solve-the-equation rules).

First, we compare the MISE of the KDE associated to each considered bandwidth selection method. Secondly, the bias and MSE of these bandwidth selectors are compared. Finally, we summarize the overall performance of the bandwidth selectors in a small scale simulation study. All results are based on 100 simulation runs for sample sizes 25, 50 ,100, 200 and data are generated from the standard normal distribution. Only 100 simulation runs were considered since some methods resulted in very long computing times.

As discussed in Section 1.3, the kernel density estimator depends on the selected bandwidth as well as on the kernel that is being used. It is a known fact that the choice of the kernel function has a minor influence on the resulting kernel estimate, when compared to the bandwidth of the kernel (recall Figure 1.7 and surrounding discussion). We carried out a simulation to check the effect of the selected kernel on the achieved MISE of the KDE. From our findings, displayed in Figure 4.1, we can see that Gaussian kernel leads to slightly lower MISE values than the other kernels. This was to be expected given the true underlying density to be estimated is itself Gaussian.

4.1.1 Comparison of MISE of the KDE using different bandwidth selectors

As discussed in Section 1.3.1, MISE provides a more global evaluation of density estimators compared to other error criteria. In this section, we present the approximate MISE of the KDE using each bandwidth selection method. Our simulation study is based on data generated from $N(0,1)$ and 100 simulation runs. First, we calculate optimal bandwidths for each bandwidth selection method as discussed in Chapter 2. Then, MISE is approximated by averaging ISE over many samples for each of these methods. Results are compared in Table 4.1. The optimal MISE value MISE($\hat{f}_{h_{\text{opt}}}$) is also given. As expected, we can see that for all bandwidth selectors, MISE values tend to decrease when the sample size is increased. Biased Cross-Validation(BCV) shows lower MISE values for small sample sizes, whereas DO-Validation(DO) outperforms BCV when the sample size reaches n = 100. We

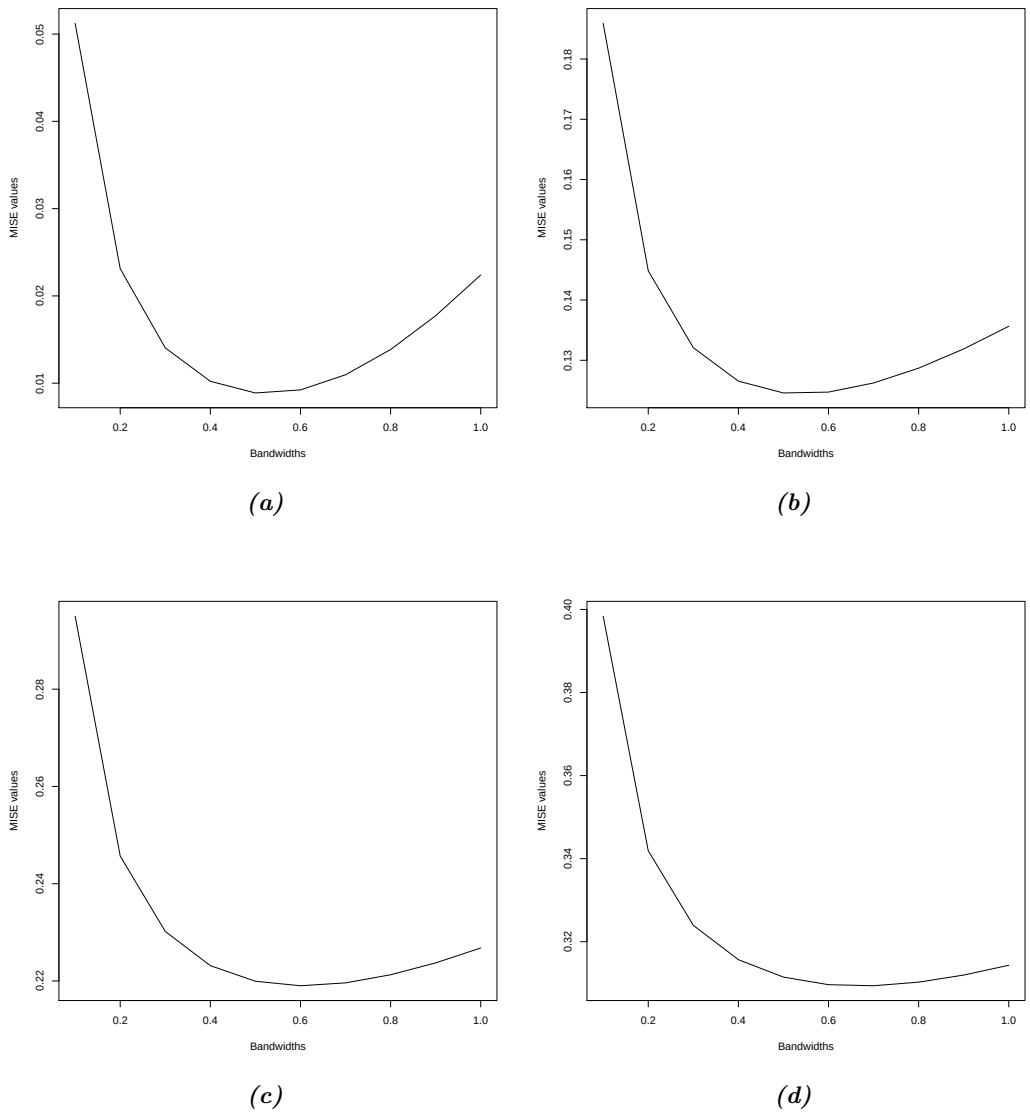


Figure 4.1. *Variation of MISE for the KDE using different kernels for samples of size n=100 generated from N(0,1), (a) Gaussian (b) Epanechnikov (c) Biweight (d) Triweight.*

Table 4.1. *MISE of the KDE using each bandwidth selection method with sample sizes, n=25, 50, 100, 200 generated from N(0,1), we used 100 simulation runs in each case.*

Method	Sample size			
	25	50	100	200
LSCV	0.0200	0.0137	0.0086	0.0042
ICV	0.0172	0.0116	0.0079	0.0042
BCV	0.0144	0.0105	0.0073	0.0039
DO	0.0154	0.0106	0.0071	0.0037
SCV	0.0222	0.0152	0.0104	0.0058
BOOT	0.0182	0.0125	0.0081	0.0039
DPI	0.0159	0.0106	0.0071	0.0037
STE	0.0202	0.0109	0.0069	0.0035
MISE($\hat{f}_{h_{opt}}$)	0.0115	0.0088	0.0061	0.0034
$\hat{h}_{opt}$	0.61	0.53	0.45	0.38

can also see that the Solve-the-equation(STE) selector outperforms DO when the sample size increases, but in-contrast STE shows poor performance for smaller samples.

4.1.2 Behaviour of different bandwidth selectors relative to optimum bandwidth.

In Figure 4.2, the MISE optimum bandwidth is marked with a vertical line. The MISE optimum bandwidths for n = 25, 50, 100, 200 were found to be $\hat{h}_{opt}$= 0.61, 0.53, 0.45, 0.38 respectively. This is in line with the concepts discussed in Chapter 2, where h_{opt} decreases as n increase. Ideally, a good bandwidth selector should lead to a distribution of values centered on the optimum bandwidth line and tightly distributed around that line. We can see that SCV has tight distributions located away from the optimum bandwidth. This bias is still there even for large sample sizes. LSCV and ICV bandwidths are appropriately centered around the optimum bandwidth but they display much larger variability than other methods. We can see that DO bandwidths are reasonably close to the optimum bandwidth and their distribution is more concentrated than LSCV and ICV. This further improves when the sample size increases. Similarly, BOOT bandwidths are also closer to the optimum

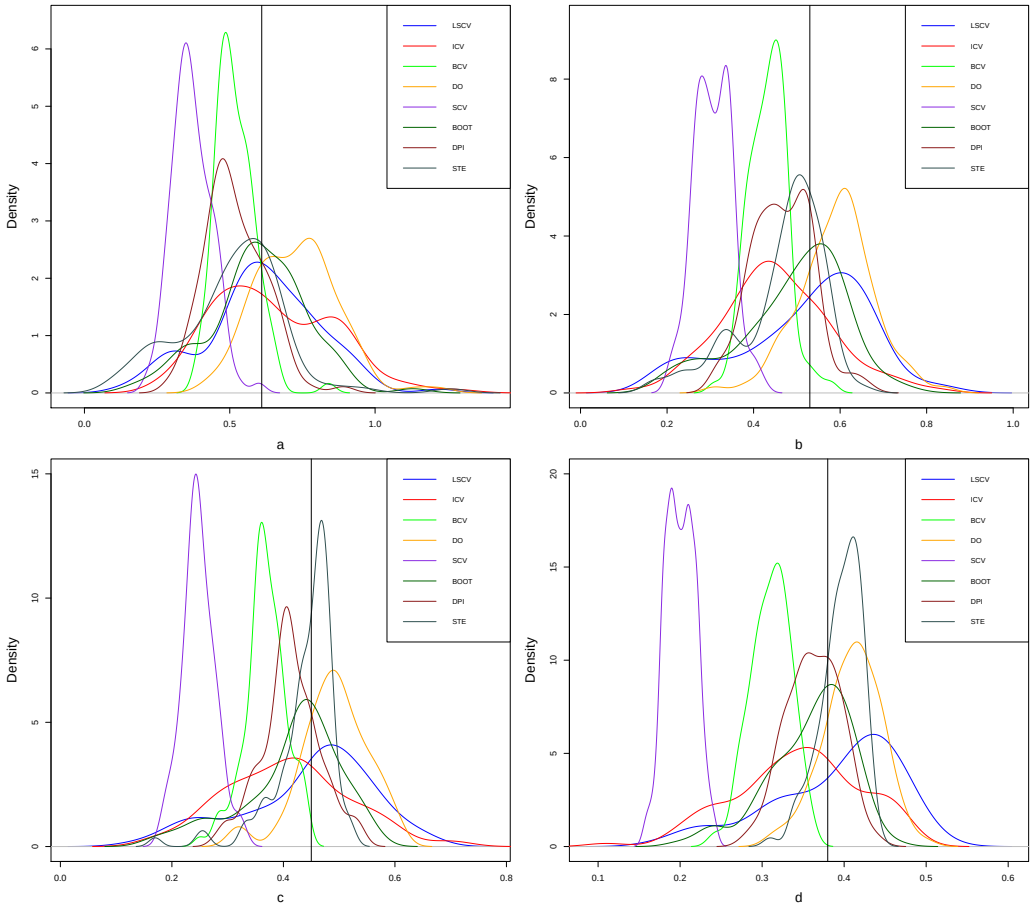


Figure 4.2. *Simulated distributions of selected bandwidth selectors through repeated sampling with sample sizes n=25, 50, 100, 200. The underlying density to be estimated was* $N(0,1)$ *and 100 simulation runs were used. In each case, the vertical line represents the MISE optimal bandwidth for the given sample size.*

bandwidth and more tightly distributed than LSCV and ICV. Interestingly, however, the BOOT selector tends to under-smooth (i.e. use bandwidths that are too small) and the DO selector tends to over-smooth (i.e. use bandwidths that are too large) in the current context. Finally we note that, DPI and STE bandwidth selectors have a very good performance for larger sample sizes. This make sense as they are derived from asymptotic theory, which holds approximately for larger sample sizes. We can clearly see that they outperform DO for sample sizes above $n = 100$.

Table 4.2. *Squared bias(SB) and variance(Var) of different bandwidth selectors for sample sizes, n=25, 50, 100, 200. The underlying density to be estimated was $N(0,1)$ and 100 simulation runs were used.*

Method	Sample size							
	25		50		100		200	
	SB	Var	SB	Var	SB	Var	SB	Var
LSCV	0.00001	0.03836	0.00015	0.02393	0.00004	0.01401	0.00009	0.00616
ICV	0.00162	0.04186	0.00598	0.01580	0.00186	0.01228	0.00132	0.00577
BCV	0.00841	0.00472	0.00929	0.00197	0.00681	0.00135	0.00482	0.00060
DO	0.01414	0.01970	0.00376	0.00763	0.00166	0.00376	0.00089	0.00127
SCV	0.05794	0.00445	0.05022	0.00179	0.04064	0.00086	0.03229	0.00031
BOOT	0.00014	0.02765	0.00152	0.01463	0.00119	0.00779	0.00044	0.00254
DPI	0.00850	0.01062	0.00403	0.00480	0.00135	0.00273	0.00030	0.00107
STE	0.01049	0.03341	0.00426	0.00978	0.00011	0.00304	0.00027	0.00062

4.1.3 Comparison of MSE for different bandwidth selectors.

Before we compare the MSE for different bandwidth selectors, let us compare their squared bias and variance. Generally, the squared error criteria is often preferred to the absolute or actual error for mathematical simplicity (see Wand & Jones, 1994). Hence, we consider the squared bias instead of the actual bias, which can yield negative values. Data are again generated from the standard normal distribution $N(0,1)$ for sample sizes n = 25, 50, 100 and 200. The squared bias and variance of each bandwidth selector are given in Table 4.2 and their MSE are given in Table 4.3.

From Table 4.2 we can see that LSCV and ICV have the lowest bias, but they also have the largest variance. BCV, DO, BOOT, DPI and STE all show a decreasing trend for both bias and variance when the sample size increases. SCV has the highest bias for all sample sizes, but it also shows significantly lower variance for higher sample sizes when compared to other methods.

Similarly to Table 4.1, where MISE($\hat{f}_{\hat{h}}$) is presented for all considered bandwidth selectors, we see in Table 4.3 that MSE($\hat{h}$) also decreases when the sample size is increased. The DO selector shows the most promising results, considering all sample sizes. However, when

Table 4.3. *MSE of different bandwidth selectors for sample sizes, n=25, 50, 100, 200, generated from N(0,1) for 100 simulation runs.*

Method	Sample size			
	25	50	100	200
LSCV	0.0384	0.0241	0.0141	0.0063
ICV	0.0435	0.0218	0.0141	0.0071
BCV	0.0131	0.0113	0.0082	0.0054
DO	0.0338	0.0114	0.0054	0.0022
SCV	0.0624	0.0520	0.0415	0.0326
BOOT	0.0278	0.0162	0.0090	0.0030
DPI	0.0191	0.0088	0.0041	0.0014
STE	0.0439	0.0140	0.0032	0.0009

focusing on larger sample sizes, we can see that DPI and STE outperform DO. Once again, we see the SCV selector yields poor performance. LSCV, ICV and BCV have moderate performance when compared to the other bandwidth selectors.

From our simple analysis, it is clear that the DO selector seems to be quite stable and exhibit favourable performance for all sample sizes in the current context. Recall that do-validation (DO) is a combination of two one-sided cross-validation techniques specifically, right and left sided cross-validation. It is suggested that DO is more robust for the normal data that we used for our simulation. However, it is also robust even when there is asymmetry present in the distribution (see Mammen et al., 2011). Therefore we only considered the DO selector for our simulation analysis, rather than studying one sided cross validation (OSCVs) separately. Many authors have mentioned the stability issues associated with CV methods and their tendency to under-smooth. This is because CV related methods tend to select smaller bandwidths than other selectors (see Heidenreich et al., 2013) as they try to capture features of the sample at hand.

As mentioned above, the plug-in methods DPI and STE outperformed the DO for larger sample sizes in our simulated experiment. A potential explanation for this is that a large sample size is preferred to estimate the pilot bandwidth of these estimators. When the pilot bandwidth is closer to the optimum bandwidth, then the final plug-in estimator is expected

to be more accurate. Another interesting observation is that the relative performance of the bootstrap selector (BOOT) improves as sample size increases. However, BOOT selector is outperformed by the plug-in selectors DPI and STE for large sample sizes.

Considering all the bandwidth selectors included in our small scale simulation study SCV shows by far the poorest performance even for large sample sizes. All-in-all, do-validation (DO) performs well for all sample sizes, whereas the solve-the-equation (STE) method is preferred for large samples. These results are obtained using Gaussian kernel and data are generated form standard normal distribution for different sample sizes as mentioned above.

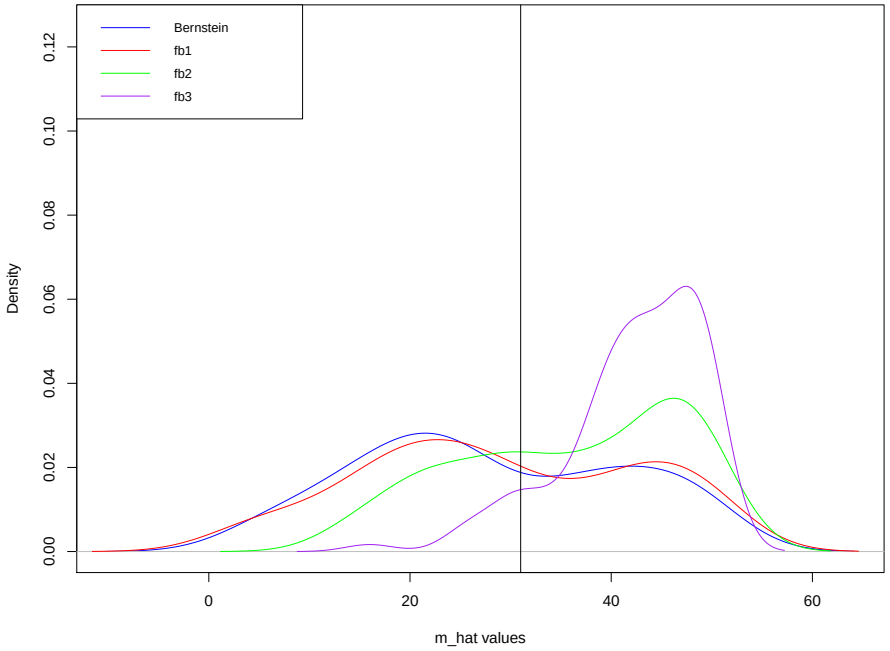
4.2 Order selection for BDE

In this section, we report on simulations that were performed to compare order selection strategies for Bernstein density estimation. It should be noted that, although the Bernstein basis was discovered more than 100 years ago, Bernstein polynomial based estimation methods are relatively immature compared to kernel estimation methods. Hence, only a handful of Bernstein estimators are available in the literature. With the availability of high performance computing, Bernstein based methods are gaining interest at present. In our simulation study, we concentrate on a few order selection methods and compare the performance of these methods for BDEs using MISE, squared bias, variance and MSE for different sample sizes.

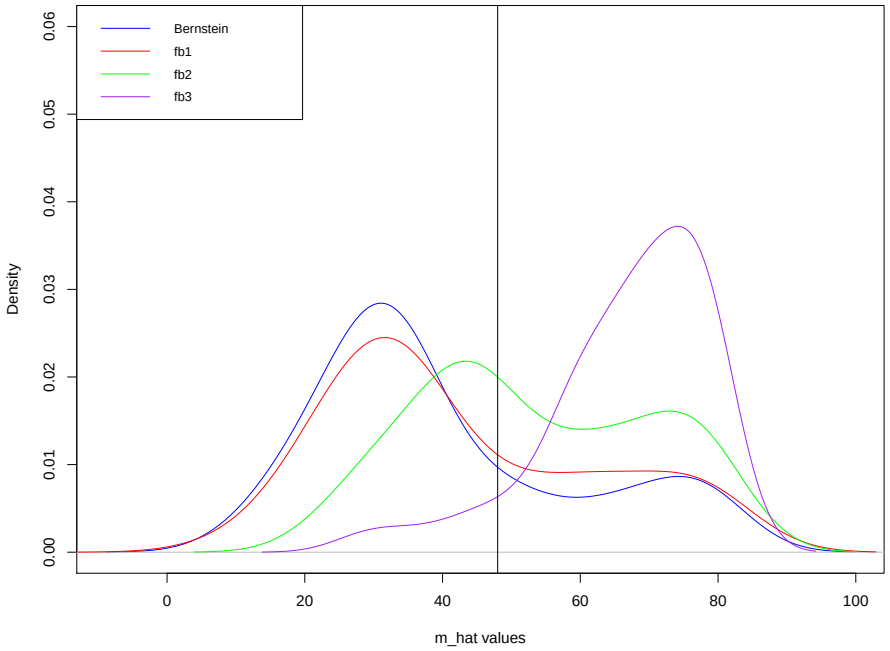
4.2.1 Density of Cross-validated orders for four estimators

Here, we illustrate the density of the cross-validated orders related to four Bernstein estimators $\hat{f}_V$, $\hat{f}_{B_1}$, $\hat{f}_{B_2}$ and $\hat{f}_{B_3}$ as specified in Table 4.4. Figure 4.3 is generated considering sample sizes of $n = 25$ and 50 with 100 simulation runs. The vertical lines represent the optimum MISE order, $m_{V,opt} = 30$ and 48 respectively considering the Vitale (1975) estimator $\hat{f}_V$.

Accordingly, it is understood that a good order selection procedure should lead to $\hat{m}$



(a)



(b)

Figure 4.3. *Simulated distribution of $\hat{m}$ for different Bernstein estimators with sample size (a) n=25, (b) n=50.*

Table 4.4. *Existing order selectors in BDE, considered for the simulation study*

Order selector	Description	Equation
$\hat{m}_V$	Order selector introduced by Vitale (1975)	(3.3)
$\hat{m}_{B_1}$	First boundary-kernel based order selector introduced by Kakizawa (2004)	(3.9)
$\hat{m}_{B_2}$	Second boundary-kernel based order selector introduced by Kakizawa (2004)	(3.10)
$\hat{m}_{B_3}$	Third boundary-kernel based order selector introduced by Kakizawa (2004)	(3.11)

having a density that is centered and tightly distributed around the optimum order. We can see that $\hat{m}_{B_2}$ is very close to this criteria, followed by $\hat{m}_{B_3}$, $\hat{m}_V$ and $\hat{m}_{B_1}$. We then explored the same relationship on large sample sizes, whereas it was found that $\hat{m}_{B_3}$ can not be evaluated due to very high computing times. However, it is necessary to consider large sample sizes for our simulation study to match the real world scenarios. Therefore, we did not consider $\hat{m}_{B_3}$ for larger sample sizes in our comparisons in our study.

4.2.2 Effect of distribution

Three distributions are selected as illustrated in Figure 4.4 to check the effect of the underlying distribution on the performance of order selectors. We considered the sample size n = 500 with 1000 simulation runs for the study presented here. As Bernstein estimators are appropriate for data on an interval, and to account for different shapes of distributions, we use the following distributions in our simulation study:

- D_1 = The mixture $\frac{2}{5}Beta(2,5) + \frac{1}{2}Beta(3,20) + \frac{1}{10}Beta(7,2)$.
- D_2 =The mixture $\frac{1}{5}Beta(2,5) + \frac{1}{5}Beta(3,20) + \frac{3}{5}Beta(7,2)$.
- D_3 =The symmetric $Beta(4,4)$.

The simulated densities of the resulting cross-validated order selectors for the different underlying distributions are plotted in Figure 4.5. Note that the vertical lines represent the optimum order $m_{V,opt}$ with respect to MISE for the classical Bernstein estimator $\hat{f}_V$. We found that distribution given in Figure 4.4a and 4.4c result in the best performance of the different order selectors, since the simulated densities of these order selectors are very close to the optimum order. Nevertheless, we consider all three distributions illustrated in Figure 4.4 for our simulation study.

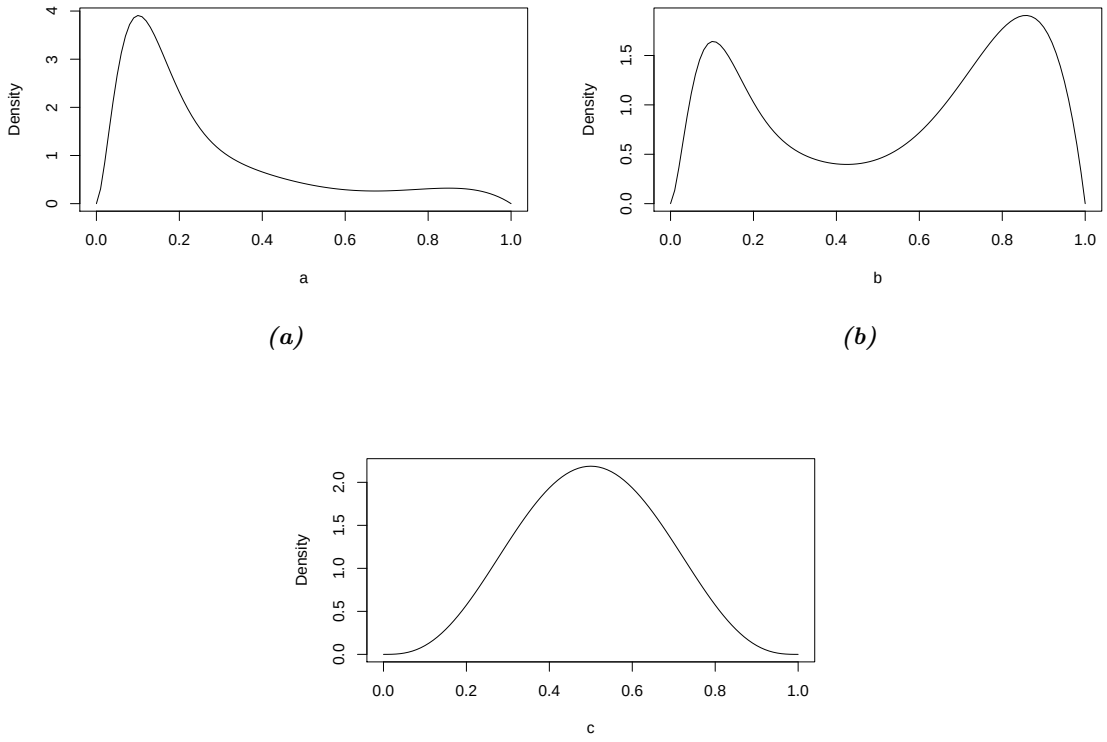
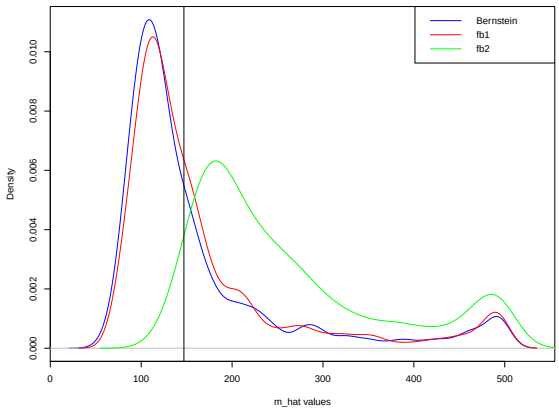
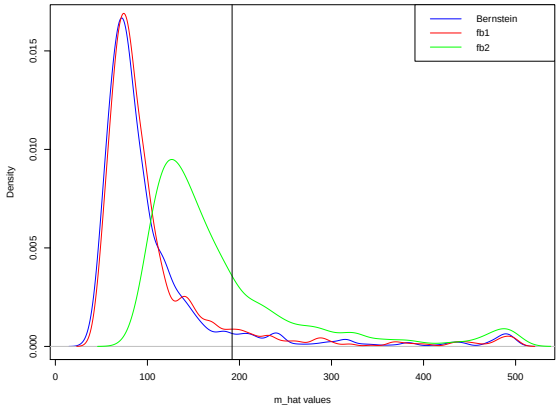


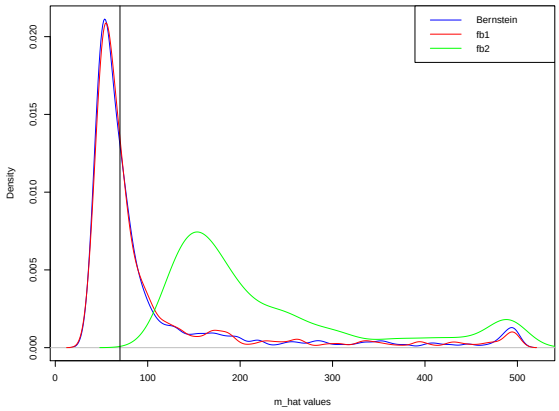
Figure 4.4. *True density plots of the three distributions used in simulation, (a) D_1, (b) D_2 and (c)D_3.*



(a)



(b)



(c)

Figure 4.5. *Simulated distributions of different order selectors under each scenario for the true density to be estimated (a) D_1 (b) D_2 (c) D_3. Vertical lines represent optimum order $m_{V,opt}$ in each case.*

4.2.3 Effect of γ

Recall the expression for boundary kernel based BDE's (3.9, 3.10 and 3.11). The estimator $\hat{f}_{B_1}$ depends on the selected value of the parameter γ which has not been addressed before. We have considered $\gamma = 0.3, 0.6, 0.8$ and studied the impact of this choice on order selection for $\hat{f}_{B1}$. Again, we considered a sample size of $n = 500$ with 1000 simulation runs. As illustrated in Figure 4.6, the density of the resulting order selectors $\hat{m}$ moves towards the optimal value when γ is increased, but they also have long right tails and short left tails. So, in order to avoid any ambiguities, we consider all three γ values in our simulation study to better compare.

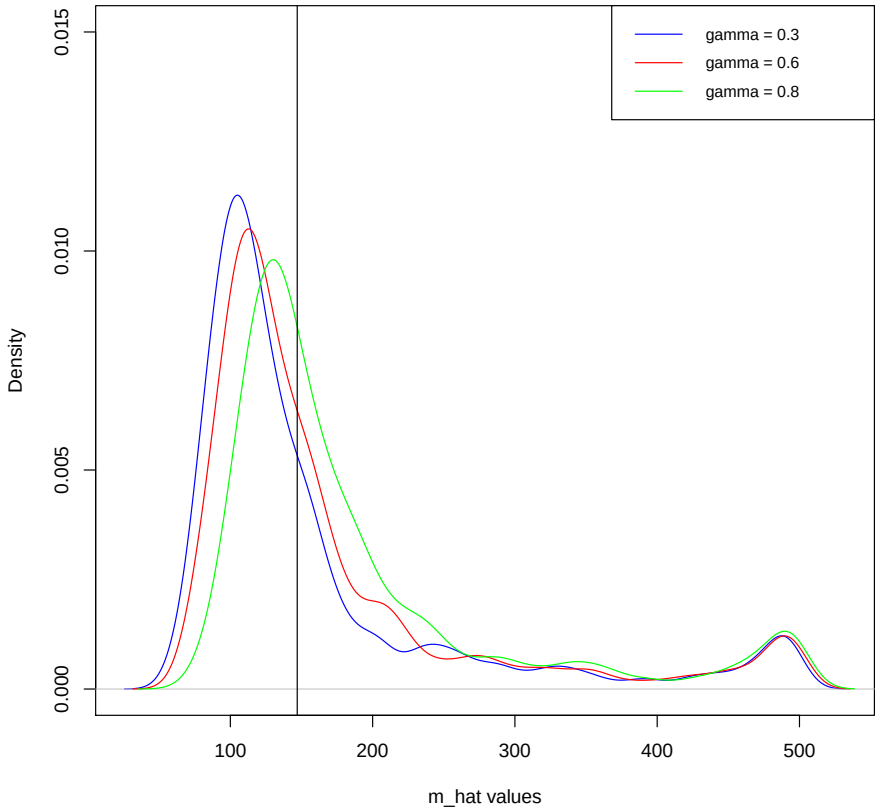


Figure 4.6. *Simulated distribution of the cross-validated order selector of $\hat{m}_{B_1}$, using $\gamma = 0.3, 0.6, 0.8$; the vertical line represent the optimal order $m_{V,opt}$.*

Table 4.5. *MISE of BDE using different cross-validated order selectors $\hat{m}_V, \hat{m}_{B_1}$ and $\hat{m}_{B_2}$ for sample sizes n = 20, 50, 100, 250, 500, 800.*

Selectors	Sample size					
	20	50	100	250	500	800
$\hat{m}_V$	0.22131	0.12203	0.09944	0.04695	0.02512	0.01728
$\hat{m}_{B_1}$	0.21974	0.12289	0.09962	0.04702	0.02507	0.01723
$\hat{m}_{B_2}$	0.21737	0.12337	0.09992	0.04716	0.02607	0.01800
MISE_{opt}	0.21514	0.11164	0.06843	0.03508	0.02123	0.01495

4.2.4 Comparison of MISE for selected estimators

We considered sample sizes n = 20, 50, 100, 250, 500, 800 and evaluated the MISE of the classical Bernstein density estimator (BDE) using the different order selection strategies. We can see that in Table 4.5, as the sample size increases, MISE values of all the estimators decrease. The order selector $\hat{m}_{B_1}$ results in the lowest MISE for large sample sizes, specifically when the sample size increases above 250. However, considering all sample sizes, the MISE of the BDE obtained from all three order selectors are in a close range.

4.2.5 Effect of the sample size on order selection

In Figure 4.7, we analyse the behaviour of the different order selectors for different sample sizes. We can see that for smaller samples, specifically n = 25 and 50 there is a difference in the shape of the densities of the order selectors. However, when the sample size increases, we can see that the shape of these densities do not show a significant variation. This is due to the fact that order selectors are more stable for large sample sizes.

We also can see bumps in the right hand tails of these densities (note these bumps are also visible in Figure 4.5). This artifact is generated based on how the simulation is implemented. Specifically, in each case we calculate the MISE and order selection criterion for m between 1 and a preselected maximum, denoted m^*. In some simulation contexts, the selected orders can be close to m^* for some samples and would in fact exceed that value if larger values of m had been considered. However, since the criterion value was not computed

for m values above m^*, we see a form of censoring in the selected values, which creates these bumps. The mass associated to these bumps would be spread out across larger m values if m^* was increased. We did not go through this process because of the expected limited impact of this phenomena when the bump is small and because of the required simulation time in these cases. We note that computing times can become prohibitive when large m values are considered with large samples.

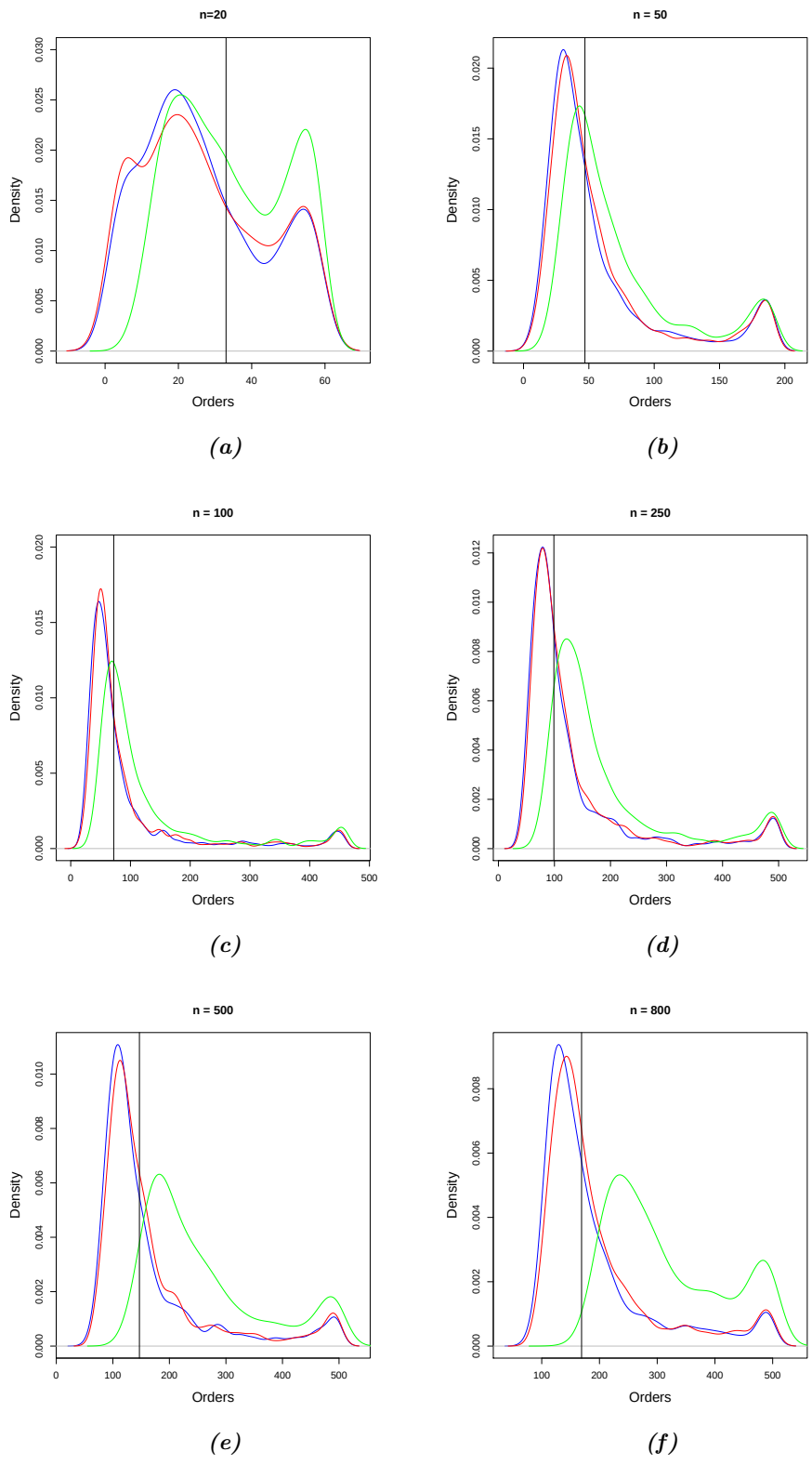


Figure 4.7. *Simulated distributions of different order selectors $\hat{m}_V$ (blue), $\hat{m}_{B_1}$ (red) and $\hat{m}_{B_2}$ (green) with different sample sizes (a) n=20 (b) n=50 (c) n=100 (d) n=250 (e) n=500 (f) n=800; the vertical line represent the optimal order $m_{V,opt}$.*

4.2.6 Introducing new order selectors for BDE

Having analysed the behaviour of the above order selectors, we now introduce new order selectors, which are linear combinations of the former. The concept of averaging the optimal orders obtained with different strategies to obtain an improved selector is adapted from the indirect cross-validation and do-validation methods. We have seen that these methods can provide an improvement in the context of KDE. We have to explore a similar strategy, but to BDE. For this, we formulate a few different order selection strategies in Table 4.6. Then, we evaluate these new selectors based on MISE, squared bias, variance and MSE, similar to our previous discussions.

Table 4.6. *Order selection strategies for BDEs considered for the simulation study*

Name	Description
$\hat{m}_V$	Cross-validated order of $\hat{f}_V$ using (3.14)
$\hat{m}_{B_1}$	Cross-validated order of $\hat{f}_{B_1}$ using (3.21)
$\hat{m}_{B_2}$	Cross-validated order of $\hat{f}_{B_2}$ using (3.22)
$\hat{m}_{B_3}$	$(\hat{m}_{B_1} + \hat{m}_{B_2})/2$
$\hat{m}_{B_4}$	$(\hat{m}_V + \hat{m}_{B_1})/2$
$\hat{m}_{B_5}$	$(\hat{m}_V + \hat{m}_{B_2})/2$
$\hat{m}_{B_6}$	$(\hat{m}_V + \hat{m}_{B_1} + \hat{m}_{B_2})/3$
$\hat{m}_{B_7}$	$(\hat{m}_{B_{1\gamma=0.6}} + \hat{m}_{B_{1\gamma=0.3}})/2$
$\hat{m}_{B_8}$	$(\hat{m}_{B_{1\gamma=0.8}} + \hat{m}_{B_{1\gamma=0.3}})/2$
$\hat{m}_{B_9}$	$(\hat{m}_{B_{1\gamma=0.6}} + \hat{m}_{B_{1\gamma=0.8}})/2$
$\hat{m}_{B_{10}}$	$(\hat{m}_{B_{1\gamma=0.6}} + \hat{m}_{B_{1\gamma=0.8}} + \hat{m}_{B_{1\gamma=0.3}})/3$

Table 4.7 was generated for the order selectors introduced based on a sample size of $n = 500$ for 1000 simulation runs. In fact, throughout upcoming discussions we consider the same sample size and the same number of simulation runs. Here MISE of the BDE generated using different order selection strategies are investigated for all three distribu-

Table 4.7. *Comparison of MISE for BDE using each order selector for sample size* $n = 500$ *and 1000 simulation runs. Distributions are defined in Section 4.2.2.*

Selectors	Underlying Distribution		
	D_1	D_2	D_3
$\hat{m}_V$	0.02512	0.01841	0.01301
$\hat{m}_{B_1}$	0.02507	0.01842	0.01282
$\hat{m}_{B_2}$	0.02607	0.02063	0.01590
$\hat{m}_{B_3}$	0.02513	0.01915	0.01381
$\hat{m}_{B_4}$	0.02512	0.01835	0.01281
$\hat{m}_{B_5}$	0.02508	0.01914	0.01382
$\hat{m}_{B_6}$	0.02489	0.01871	0.01316
$\hat{m}_{B_7}$	0.02515	0.01837	0.01269
$\hat{m}_{B_8}$	0.02499	0.01837	0.01265
$\hat{m}_{B_9}$	0.02493	0.01836	0.01269
$\hat{m}_{B_{10}}$	0.02494	0.01834	0.01268

tions specified in Figure 4.4. From the table, we can see that the BDE using the order selectors $\hat{m}_{B_6}, \hat{m}_{B_7}, \hat{m}_{B_8}, \hat{m}_{B_9}$ and $\hat{m}_{B_{10}}$ have slightly lower MISE values for all three distributions. Hence, we focus on the comparison of these order selectors for the other properties, specifically squared bias, variance and MSE.

As illustrated in Table 4.8, considering all three distributions, the order selectors $\hat{m}_{B_2}$ and $\hat{m}_{B_6}$ result in higher squared bias and variance. Order selectors $\hat{m}_V$, $\hat{m}_{B_1}$ and $\hat{m}_{B_7}$ shows the lowest squared bias, but they also have higher variance. Order selectors $\hat{m}_{B_8}$, $\hat{m}_{B_9}$ and $\hat{m}_{B_{10}}$ have moderately lower squared bias values and lower variance values.

The MSE of the Bernstein density estimators generated from the chosen order selectors are given in Table 4.9. Once again, the underlying distributions used for the simulations are as per Figure 4.4. Considering all three distributions, the selector $\hat{m}_{B_7}$ results in the lowest MSE, which is even lower than $\hat{m}_V$, the previously used selector in the literature. Order selectors $\hat{m}_{B_2}$ and $\hat{m}_{B_6}$ shows the highest MSE values, also note these two order selectors show higher squared bias and variance, therefore yield poor performance. Other order selectors, $\hat{m}_{B_8}$, $\hat{m}_{B_9}$ and $\hat{m}_{B_{10}}$ are comparable with $\hat{m}_B$, $\hat{m}_{B_1}$ and $\hat{m}_{B_7}$, whereas the percentage difference of the resulting MSE is less than 5%.

Table 4.8. *Comparison of squared bias and variance for selected order selectors for distributions considered in Section 4.2.2.*

Selectors	Squared bias			Variance		
	D_1	D_2	D_3	D_1	D_2	D_3
$\hat{m}_V$	239.6	435.2	1100.4	10145.9	6062.9	10507.2
$\hat{m}_{B_1}$	460.8	536.1	1020.5	10235.1	5941.2	9728.1
$\hat{m}_{B_2}$	12444.7	9117.4	25021.9	11118.5	8101.2	12608.8
$\hat{m}_{B_6}$	2450.1	2161.6	5540.9	9401.9	5840.2	9150.2
$\hat{m}_{B_7}$	278.3	526.5	1067.7	10064.4	5790.7	9610.6
$\hat{m}_{B_8}$	667.8	766.3	1222.1	9901.1	5757.5	9670.7
$\hat{m}_{B_9}$	936.8	777.7	1171.5	9948.7	5825.3	9682.8
$\hat{m}_{B_{10}}$	594.6	685.1	1153.6	9872.3	5747.2	9614.6

Table 4.9. *Variation of MSE for selected order selectors related to distributions D_1, D_2 and D_3.*

Selectors	MSE		
	D_1	D_2	D_3
$\hat{m}_B$	10385.4	6498.2	11607.7
$\hat{m}_{B_1}$	10696.0	6477.2	10748.6
$\hat{m}_{B_2}$	23563.2	17218.6	37630.7
$\hat{m}_{B_6}$	11852.0	8001.8	14691.1
$\hat{m}_{B_7}$	10342.7	6317.3	10678.3
$\hat{m}_{B_8}$	10568.9	6523.8	10892.7
$\hat{m}_{B_9}$	10885.6	6603.1	10854.3
$\hat{m}_{B_{10}}$	10466.8	6432.4	10768.1

4.3 The bootstrap order selector

In this section, we discuss order selection for BDE based on the bootstrap, following principles discussed in Section 3.3.

4.3.1 Behaviour of the bootstrap selector

First, we compare MISE of the BDE using bootstrap method for distributions discussed in Section 4.2.2. As given in Table 4.10, we can see that MISE values decrease when the

Table 4.10. *MISE of the BDE using bootstrap method related to distributions D_1, D_2 and D_3 for sample sizes n = 20, 50, 100 and 500.*

Sample size	D_1		D_2		D_3	
	$\text{MISE}_{\text{Boot}}$	MISE_{opt}	$\text{MISE}_{\text{Boot}}$	MISE_{opt}	$\text{MISE}_{\text{Boot}}$	MISE_{opt}
20	0.286	0.215	0.239	0.184	0.121	0.091
50	0.139	0.112	0.121	0.094	0.064	0.049
100	0.081	0.068	0.066	0.055	0.039	0.030
500	0.023	0.021	0.017	0.016	0.011	0.010

sample size increase for all three distributions. Also, the $\text{MISE}_{\text{Boot}}$ values are getting closer to MISE_{opt} for large sample sizes.

Next, we illustrate simulated densities of order selectors $\hat{m}_V$, $\hat{m}_{B_1}$, $\hat{m}_{B_2}$ and $\hat{m}_{\text{Boot}}$ in Figure 4.8. The density of the bootstrapped order selector ($\hat{m}_{\text{Boot}}$) is closer to the optimum order ($m_{V,opt}$) and tightly distributed than other cross-validated order selectors. Also, the bootstrapped order selector doesn't show any undesirable bumps on the right tail as cross-validated order selectors. These are two important advantages over cross-validated order selectors.

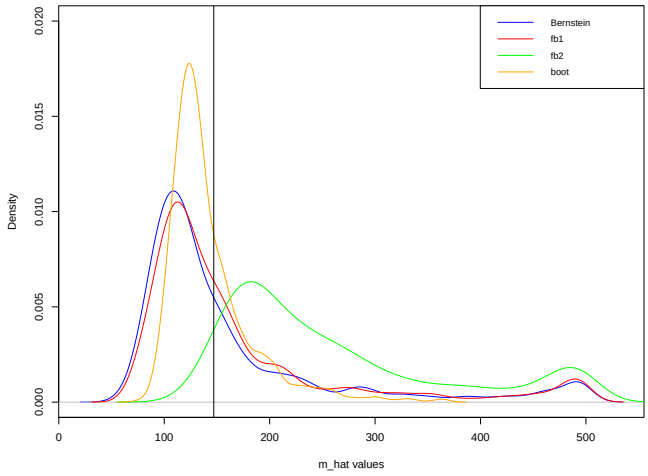
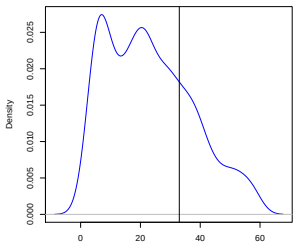
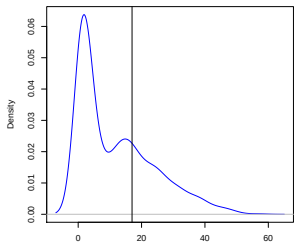


Figure 4.8. *Simulated densities of order selectors $\hat{m}_V$ (blue), $\hat{m}_{B_1}$ (red), $\hat{m}_{B_2}$ (green), $\hat{m}_{Boot}$ (orange), for sample size n = 500; the vertical line represent the optimal order $m_{V,opt}$.*

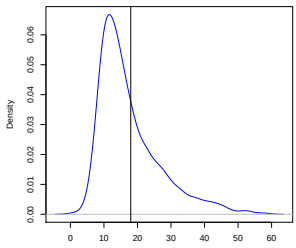
Then, we plot the simulated densities of the bootstrap order selector over repeated sampling from distributions D_1, D_2 and D_3 in Figure 4.9. Simulations are based on sample sizes $n = 20, 50, 100, 500$ for 1000 simulation runs. Note that the vertical line represent the optimum order, $m_{V,opt}$, according to MISE. From the plots, we can see that the shapes of the density curves are tightly distributed around the $m_{V,\text{opt}}$ for larger sample sizes considering all three distribution. Also, for distribution D_3, we can see the same even for smaller sample sizes. As expected, all density curves again exhibit a short left hand tail and a long right hand tail. One noteworthy feature of these graphs is the fact that the bumps (and related censoring phenomenon) mentioned in Section 4.2.5 associated to cross-validation disappear here. This is an advantage over the CV based methods, as such artifacts are undesirable. Next, properties such as squared bias, variance and MSE of the Bootstrap order selector will be investigated for all three distributions for all mentioned sample sizes.



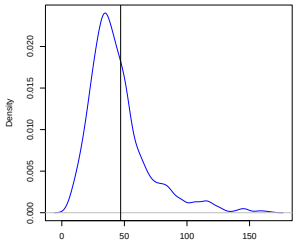
(a) *n=20, D_1*



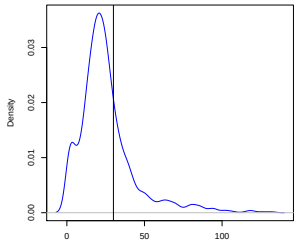
(b) *n=20, D_2*



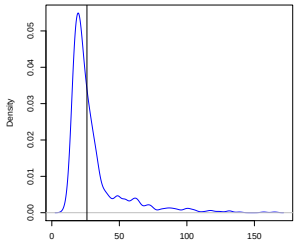
(c) *n=20, D_3*



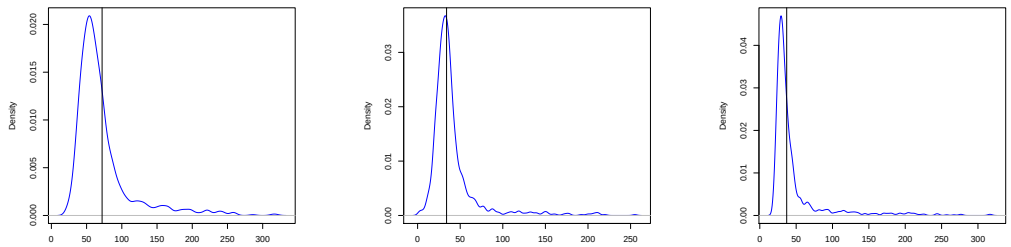
(d) *n=50, D_1*



(e) *n=50, D_2*



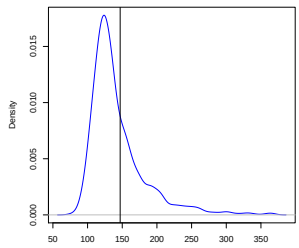
(f) *n=50, D_3*



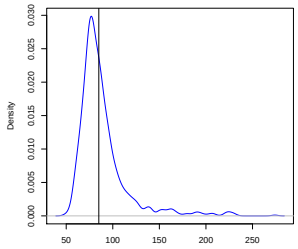
(g) *n=100, D_1*

(h) *n=100, D_2*

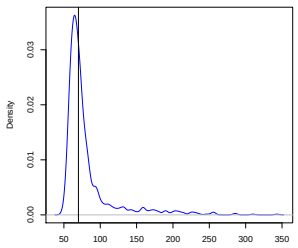
(i) *n=100, D_3*



(j) *n=500, D_1*



(k) *n=500, D_2*



(l) *n=500, D_3*

Figure 4.9. *Simulated densities of bootstrap order selector with different distributions and different sample sizes; the vertical lines represent the optimal order $m_{V,opt}$.*

4.3.2 Squared bias, variance and MSE comparison with CV estimators

Having analysed the squared bias, variance and MSE of BDEs generated from chosen cross-validation (CV) based order selectors, we compare them with the bootstrap estimator, specifically for sample size n = 500 with 1000 simulation runs. We can see that the bootstrap order selector shows significantly lower values for squared bias, variance and MSE compared to other estimators. This important finding, coupled with the advantages outlined in Section 4.3.1 suggest that the bootstrapped order selector compares very favourable to all cross-validated based selectors. Unfortunately, the computational burden of using the bootstrap selector is still prohibitive and we could not examine cases of sample sizes above 500.

Table 4.11. *Variation of MSE for chosen order selectors and bootstrap order selector for sample size* $n = 500$ *and 1000 simulation runs.*

Selectors	MSE		
	Dist (a)	Dist (b)	Dist (c)
$\hat{m}_B$	10385.4	6498.2	11607.7
$\hat{m}_{B_1}$	10696.0	6477.2	10748.6
$\hat{m}_{B_2}$	23563.2	17218.6	37630.7
$\hat{m}_{B_6}$	11852.0	8001.8	14691.1
$\hat{m}_{B_7}$	10342.7	6317.3	10678.3
$\hat{m}_{B_8}$	10568.9	6523.8	10892.7
$\hat{m}_{B_9}$	10885.6	6603.1	10854.3
$\hat{m}_{B_{10}}$	10466.8	6432.4	10768.1
$\hat{m}_{\text{Boot}}$	1646.6	727.3	1444.1

Table 4.12. *Comparison of squared bias and variance for cross-validated order selectors and the bootstrap order selector, related to distributions* D_1, D_2 *and* D_3 *for sample size* $n = 500$ *and 1000 simulation runs.*

Selectors	Squared bias			Variance		
	Dist (a)	Dist (b)	Dist (c)	Dist (a)	Dist (b)	Dist (c)
$\hat{m}_B$	239.6	435.2	1100.4	10145.9	6062.9	10507.2
$\hat{m}_{B_1}$	460.8	536.1	1020.5	10235.1	5941.2	9728.1
$\hat{m}_{B_2}$	12444.7	9117.4	25021.9	11118.5	8101.2	12608.8
$\hat{m}_{B_6}$	2450.1	2161.6	5540.9	9401.9	5840.2	9150.2
$\hat{m}_{B_7}$	278.3	526.5	1067.7	10064.4	5790.7	9610.6
$\hat{m}_{B_8}$	667.8	766.3	1222.1	9901.1	5757.5	9670.7
$\hat{m}_{B_9}$	936.8	777.7	1171.5	9948.7	5825.3	9682.8
$\hat{m}_{B_{10}}$	594.6	685.1	1153.6	9872.3	5747.2	9614.6
$\hat{m}_{\text{Boot}}$	15.51	12.52	122.17	1631.08	714.78	1321.94

Chapter 5

Conclusion

This thesis provides a systematical approach to study order selectors for Bernstein density estimation (BDE). BDE is an emerging topic in the field of density estimation, whereas only a handful of Bernstein polynomial based estimators are available in recent literature. Hence, we started by reviewing Kernel density estimators (KDEs), where an extensive research on the topic is already available. Specifically, we studied plug-in methods, cross-validation methods and bootstrap based methods. Next, we performed simulation studies on each of these methods to compare properties such as MISE, MSE, bias and variance. Simulation results revealed some interesting findings on these methods. We found the do-validation method to have achieved the best overall performance. However, the solve-the-equation plug-in methods outperformed do-validation for large sample sizes. Having understood the strengths and weakness of these KDEs, we then moved on to analysis of order selection for BDEs.

Even though the Bernstein polynomial basis was introduced more than a century ago, this line of density estimation methods have been subservient until the 21st century. One reason behind this is the heavy computational burden of these estimators. With the development of computers and availability of high computational power, these methods are gaining their rightful place in the field of density estimation. As selecting the smoothing parameter is of paramount importance in kernel density estimation, selecting the order is

crucial in Bernstein density estimation. The main difference of these methods over KDEs is that, KDEs tend to under-perform in boundary regions, whereas BDEs naturally behave adequately around boundaries, which is crucial when the density function to be estimated is supported on a compact interval $[a, b]$. In particular, BDEs do not suffer from increased boundary bias. However, they have slightly higher variance when compared to KDEs. It should be noted that boundary kernels do exist (and in fact were used to construct the Kakizawa (2004) estimators), but they are cumbersome to use. Our goal, in this thesis, was to study automatic order selection methods in BDEs and propose simple and easy to implement improvements. Hence, we tried to identify these limitations and proposed a few new order selection methods.

We introduced the Vitale (1975) estimator followed by three estimators introduced by Kakizawa (2004). We derived expressions for these estimators and the corresponding properties such as MISE, MSE, squared bias and variance, which are important for our simulation study. Next, we explained the simulation methodology and two methods used in the simulation to perform cross-validation and bootstrapping. Then, a few new order selectors suggested based on using linear combinations of the existing order selectors. Two principles were investigated in the context of order selection, specifically, cross-validation including the indirect and do-validation variants and bootstrapping.

These new order selectors are compared with existing order selectors to identify selectors which would outperform existing selectors. Properties such as variance, squared bias and MSE were used for the comparison as illustrated in Tables 4.11 and 4.12. It was found that the performance of the new order selectors based on CV are dependent on the underlying distribution and are in the same range as existing order selectors. No significant improvement was observed over ordinary least-squares cross-validation. The bootstrapping order selector, on the other hand, outperformed all other selectors for all three distributions considering a sample size up to $n = 500$. However, this should be further studied for larger sample sizes and different distributions. We leave this as a future work due to the high simulation time and high computational burden experienced when the sample size is larger than $n = 500$. More

theoretical and computational developments are necessary to efficiently use the Bootstrap based approach for large samples. Another favourable feature of the bootstrapping order selector is that it eliminates the bumps observed in the right tail in CV based order selectors. This is due the fact that bootstrapping inherently averages over many samples and crucially makes order selection less focused on the one sample at hand.

Nevertheless, our main finding here is that the bootstrapping approach behaves very advantageously compared to all CV-based selectors and is a promising candidate for further improvement.

5.1 Numerical limitations

We now briefly summarize the numerical and computational issues we encountered during this work. The first important numerical limitation we encountered in the simulation study was in the evaluation of Kakizawa's third estimator for sample sizes higher than $n \geq 100$. The simulation halted in many attempts due to extreme integration behaviour. We also found that this estimator doesn't perform well, even for lower sample sizes. So, we decided to exclude this estimator from our analysis.

The second issue that we faced is that for large sample sizes (above 500), the computational burden and simulation times for BDEs are considerably higher compared to simulations done with KDEs. Working from simplified equations involving quadratic forms instead of integrals helped reduce computing times significantly, but more work is still required to further improve the situation, especially on the side of efficient computing and algorithms.

Finally, for the simulation results on bootstrap order selector, we encountered some major problems with large samples. In other words, it is expected that for large samples bootstrap method should perform well. Whereas, the squared bias, variance and MSE should reduce, instead we have seen surprising increases and fluctuations in some cases.These are expected to be due to computational issues that need to be better understood. Unfortunately, the high computational time for simulations over large sample sizes was prohibitive and, we

had to stop our investigation at this point. Hence, our investigations of the performance of the bootstrap selector, although extremely promising, not fully conclusive.

5.2 Summary of main findings

- Considering BDEs, the new cross-validation based order selectors that were introduced are performing similarly to existing order selectors. The performance also depends on the underlying distribution.
- The Bootstrapped order selector in the context of BDEs shows significant improvement in performance compared to cross-validated order selectors for small and moderate sample sizes.
- Undesirable bumps were observed in the right of the simulated distribution of cross-validated order selectors. Such a bump is related to how numerical minimization is performed during cross-validation and to the flatness of the CV function for some non-negligible fraction of simulated samples. The Bootstrapped order selector, on the other hand, did not have this artifact in its simulated distribution. Interestingly, the bootstrap based approach avoids these problems, even if it is using a CV-based pilot bandwidth, by averaging over the bootstrap samples. This is an important finding in the context of BDE.

5.3 Future works

Finally, a future extensions of this research, we plan the following investigations:

- Perform simulations for the bootstrap Bernstein order selector for large sample sizes and evaluate its performance. This will require the study of efficient computational techniques.
- Implement the Beta kernel estimator of Chen (1999) and study its use towards implementing a different version of indirect cross-validation for BDEs.

- Study the Bernstein density estimation method proposed by Guan (2016), which is based on likelihood related arguments and the EM algorithm, and consider how the order selection method suggested in that work can be compared to the method discussed in this thesis.

Appendix A

A.1 Derivation of least-squares cross-validation for KDE

From (2.10), we can write LSCV function;

$$LSCV(h) = \int \hat{f}_h^2(x)dx - \underbrace{\frac{2}{n}\sum_{i=1}^{n}\hat{f}_{h,-i}(X_i)}_{\text{a}}. \tag{A.1}$$

Then, the second term of the above equality can be further expressed as

$$\begin{aligned}
\text{a} = \sum_{i=1}^{n}\hat{f}_{h,-i}(X_i) &= \sum_{i=1}^{n}\left[\frac{1}{(n-1)}\sum_{j\neq i}K_h(X_i - X_j)\right] \\
&= \sum_{i=1}^{n}\left[\frac{1}{(n-1)}\left(\sum_{j=1}^{n}K_h(X_i - X_j) - K_h(0)\right)\right] \\
&= \sum_{i=1}^{n}\left[\frac{n}{(n-1)}\hat{f}(X_i) - \frac{1}{(n-1)})K_h(0)\right] \\
&= \frac{n}{n-1}\sum_{i=1}^{n}\hat{f}(X_i) - \frac{n}{(n-1)}K_h(0).
\end{aligned}$$

Finally, replacing this into (A.1) leads to the LSCV expression

$$\text{LSCV}(h) = \int \hat{f}_h^2(x)dx - \frac{2}{n-1}\sum_{i=1}^{n}\hat{f}_h(X_i) - \frac{2}{(n-1)}K_h(0).$$

A.2 Derivation of cross-validation function of Vitale's estimator

Recall expression (2.10) for LSCV. The LSCV for the estimator of order m is given by

$$LSCV(m) = \int_0^1 \hat{f}_m^2(x)dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{m,-i}(X_i).$$

Here, the leave-one-out estimator can be re-expressed as

$$\begin{aligned}
\hat{f}_{m,-i}(x) &= \frac{1}{(n-1)}\sum_{j\neq i}\beta_{k_j+1,m-k_j}(x) \\
&= \frac{1}{(n-1)}\left[\sum_{j=1}^{n}\beta_{k_j+1,m-k_j}(x) - \beta_{k_i+1,m-k_i}(x)\right] \\
&= \frac{n}{(n-1)}\left[\frac{1}{n}\sum_{j=1}^{n}\beta_{k_j+1,m-k_j}(x) - \beta_{k_i+1,m-k_i}(x)\right] \\
&= \frac{n}{(n-1)}\hat{f}_m(x) - \frac{1}{(n-1)}\beta_{k_i+1,m-k_i}(x).
\end{aligned}$$

Hence,

$$\begin{aligned}
\text{LSCV}_{(m)} &= \int_0^1 \hat{f}_m^2(x)dx - \frac{2}{n}\sum_{i=1}^{n}\left[\frac{n}{(n-1)}\hat{f}_m(X_i) - \frac{1}{(n-1)}\beta_{k_i+1,m-k_i}(X_i)\right] \\
&= \int_0^1 \hat{f}_m^2(x)dx - \frac{2}{(n-1)}\sum_{i=1}^{n}\hat{f}_m(X_i) + \frac{2}{n(n-1)}\sum_{i=1}^{n}\beta_{k_i+1,m-k_i}(X_i).
\end{aligned} \tag{A.2}$$

The first term of the previous equality can actually be calculated and obtained in closed-form. For this, note that

$$\begin{aligned}
\int_0^1 \hat{f}_m^2(x)dx &= \int_0^1 m^2 \sum_k \sum_l F_k F_l P_k(x) P_l(x)dx \\
&= m^2 \sum_k \sum_l F_k F_l \int_0^1 P_k(x)P_l(x)dx.
\end{aligned}$$

Then, using the expression of the binomial probabilities $P_{k,m-1}$, we have that the above further simplifies to

$$\begin{aligned}\int_0^1 \hat{f}_m^2(x)dx &= m^2 \sum_k \sum_l F_k F_l \int_0^1 \binom{m-1}{k} x^k (1-x)^{m-1-k} . \binom{m-1}{l} x^l (1-x)^{m-1-l} dx \\ &= m^2 \sum_k \sum_l F_k F_l \binom{m-1}{k} \binom{m-1}{l} \ \beta(k+l+1, 2m-(k+l)-1)\end{aligned}$$

Now, we note that

$$\begin{aligned}&= \beta(k+l+1, 2m-(k+l)-1) \\ &= \frac{\Gamma(k+l+1)\Gamma(2m-(k+l)-1)}{\Gamma(2m)} \\ &= \frac{1}{(2m-1)\binom{2m-2}{k+l}}.\end{aligned}$$

So that, using (3.13),

$$\begin{aligned}\int_0^1 \hat{f}_m^2(x)dx &= m^2 \sum_k \sum_l F_k F_l \frac{\binom{m-1}{k}\binom{m-1}{l}}{(2m-1)\binom{2m-2}{k+l}} \\ &= \frac{m^2}{(2m-1)} \sum_k \sum_l F_k F_l a_{kl} \\ &= \frac{m^2}{(2m-1)} \mathbf{F}_m^t \mathbf{A}_{m,m} \mathbf{F}_m.\end{aligned} \tag{A.3}$$

Finally, (A.2) and (A.3) together imply that, for Vitale's estimator,

$$\mathrm{LSCV}_{(m)} = \frac{m^2}{(2m-1)} \mathbf{F}_m^t \mathbf{A}_{m,m} \mathbf{F}_m - \frac{2}{(n-1)} \sum_{i=1}^n \hat{f}_m(X_i) + \frac{2}{n(n-1)} \sum_{i=1}^n \beta_{k_i+1, m-k_i}(X_i)$$

A.3 Derivation of Bootstrapping function

$$\begin{aligned}\mathrm{ISE}_i^*(m) &= \int_0^1 (\hat{f}_{i,m}^*(x) - \hat{f}_{m_0}(x))^2 dx \\ &= \int_0^1 \hat{f}_{i,m}^{*\,2}(x)dx + \int_0^1 \hat{f}_{m_0}^2(x)dx - 2\int_0^1 \hat{f}_{i,m}^*(x)\hat{f}_{m_0}(x)dx.\end{aligned} \tag{A.4}$$

Following a similar approach as in (A.3), the First and second terms of (A.4) can be written as

$$\int_0^1 {\hat{f}^*_{i,m}}^2(x)dx = \frac{m^2}{(2m-1)}\mathbf{F}^{*t}_{i,m}\mathbf{A}_{m,m}\mathbf{F}^*_{i,m} \tag{A.5}$$

and

$$\int_0^1 \hat{f}^2_{m_0}(x)dx = \frac{m^2}{(2m-1)}\mathbf{F}^t_{m_0}\mathbf{A}_{m_0,m_0}\mathbf{F}_{m_0}. \tag{A.6}$$

The third term of (A.4) can be calculated and obtained in closed form as

$$\begin{aligned} 2\int_0^1 \hat{f}^*_{i,m}(x)\hat{f}_{m_0}(x)dx &= \int_0^1 2mm_0\Big[\sum_k F^*_k P_{k,m-1}(x)\Big]\Big[\sum_l F_l P_{l,m_0-1}(x)\Big]dx \\ &= 2mm_0\sum_k\sum_l F^*_k F_l \int_0^1 P_{k,m-1}(x)P_{l,m_0-1}(x)dx. \end{aligned}$$

Then, using the expression of the binomial probabilities above expression can be further simplified to

$$\begin{aligned} 2\int_0^1 \hat{f}^*_{i,m}(x)\hat{f}_{m_0}(x)dx &= 2mm_0\sum_k\sum_l F^*_k F_l \int_0^1 x^{k+l}(1-x)^{m+m_0-(k+l)-2}dx \\ &= 2mm_0\sum_k\sum_l F^*_k F_l \tbinom{m-1}{k}\tbinom{m_0-1}{l}\ \beta(k+l+1, m+m_0-(k+l)-1) \\ &= \frac{2mm_0}{(m+m_0-1)}\sum_k\sum_l F^*_k F_l a_{kl} \\ &= \frac{2mm_0}{(m+m_0-1)}\mathbf{F}^{*t}_{i,m}\mathbf{A}_{m,m_0}\mathbf{F}_{m_0}. \end{aligned} \tag{A.7}$$

Finally, using (A.5), (A.6) and (A.7)

$$\begin{aligned} \mathrm{ISE}^*_i(m) = \frac{m^2}{(2m-1)}\mathbf{F}^{*^t}_{i,m}\mathbf{A}_{m,m}F^*_{i,m} + \frac{m_0^2}{(2m_0-1)}\mathbf{F}^t_{m_0}\mathbf{A}_{m_0,m_0}\mathbf{F}_{m_0} \\ -2\frac{mm_0}{(m+m_0-1)}\mathbf{F}^{*^t}_{i,m}\mathbf{A}_{m,m_0}\mathbf{F}_{m_0}. \end{aligned}$$

Bibliography

Babu, G. J., Canty, A. J., & Chaubey, Y. P. (2002). Application of bernstein polynomials for smooth estimation of a distribution and density function. *Journal of Statistical Planning and Inference*, *105*(2), 377–392.

Bernstein, S. (1912). Demonstration of the weierstrass theorem based on the calculus of probabilitiess. *Communications of the Kharkov Mathematical Society*, *13*(1), 1–2.

Bowman, A. W., Hall, P., & Titterington, D. (1984). Cross-validation in nonparametric estimation of probabilities and probability densities. *Biometrika*, *71*(2), 341–351.

Cao, R., Cuevas, A., & Manteiga, W. G. (1994). A comparative study of several smoothing methods in density estimation. *Computational Statistics & Data Analysis*, *17*(2), 153–176.

Cao-Abad, R. (1990). *Applications and new results of the bootstrap method in the nonparametric estimation of curves* (Doctoral dissertation). Ph. D. Dissertation, University of Santiago de Compostela, Spain.

Cao-Abad, R., & Gonzales-Manteiga, W. (1990). Bootstrap methods in regression smoothing: An alternative procedure to the wild resampling plan. *Preprint*.

Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, *31*(2), 131–145.

Chiu, S.-T. (1991). Bandwidth selection for kernel density estimation. *The Annals of Statistics*, 1883–1905.

Chiu, S.-T. (1996). A comparative review of bandwidth selection for kernel density estimation. *Statistica Sinica*, 129–145.

Devroye, D., Beirlant, J., Cao, R., Fraiman, R., Hall, P., Jones, M., Lugosi, G., Mammen, E., Marron, J., Sánchez-Sellero, C., Et al. (1997). Universal smoothing factor selection in density estimation: Theory and practice. *Test*, *6*(2), 223–320.

Faraway, J. J., & Jhun, M. (1990). Bootstrap choice of bandwidth for density estimation. *Journal of the American Statistical Association*, *85*(412), 1119–1122.

Farouki, R. T. (2012). The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, *29*(6), 379–419.

Feluch, W., & Koronacki, J. (1992). A note on modified cross-validation in density estimation. *Computational statistics & data analysis*, *13*(2), 143–151.

Guan, Z. (2016). Efficient and robust density estimation using bernstein type polynomials. *Journal of Nonparametric Statistics*, *28*(2), 250–271.

Habbema, J., Hermans, J., & Van Der Broek, K. (1974). A stepwise discrimination program using density estimation, In *Compstat*. Physica Verlag Vienna.

Hall, P. (1990). Using the bootstrap to estimate mean squared error and select smoothing parameter in nonparametric problems. *Journal of multivariate analysis*, *32*(2), 177–203.

Hall, P., & Marron, J. S. (1991). Local minima in cross-validation functions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *53*(1), 245–252.

Hall, P., Marron, J., & Park, B. U. (1992). Smoothed cross-validation. *Probability theory and related fields*, *92*(1), 1–20.

Härdle, W. K., Müller, M., Sperlich, S., & Werwatz, A. (2012). *Nonparametric and semiparametric models*. Springer Science & Business Media.

Hart, J. D., & Yi, S. (1998). One-sided cross-validation. *Journal of the American Statistical Association*, *93*(442), 620–631.

Heidenreich, N.-B., Schindler, A., & Sperlich, S. (2013). Bandwidth selection for kernel density estimation: A review of fully automatic selectors. *AStA Advances in Statistical Analysis*, *97*(4), 403–433.

Jones, M. C., Marron, J. S., & Sheather, S. J. (1996). A brief survey of bandwidth selection for density estimation. *Journal of the American statistical association*, *91*(433), 401–407.

Jones, M., Marron, J. S., & Park, B. U. (1991). A simple root n bandwidth selector. *The Annals of Statistics*, 1919–1932.

Kakizawa, Y. (2004). Bernstein polynomial probability density estimation. *Journal of Nonparametric Statistics*, *16*(5), 709–729.

Leblanc, A. (2010). A bias-reduced approach to density estimation using bernstein polynomials. *Journal of Nonparametric Statistics*, *22*(4), 459–475.

Leblanc, A. (2012). On the boundary properties of bernstein polynomial estimators of density and distribution functions. *Journal of Statistical Planning and Inference*, *142*(10), 2762–2778.

Mammen, E., Martinez Miranda, M. D., Nielsen, J. P., & Sperlich, S. (2011). Do-validation for kernel density estimation. *Journal of the American Statistical Association*, *106*, 651–660.

Marron, J. S. (1987). Partitioned cross-validation. *Econometric Reviews*, *6*(2), 271–283.

Martinez-Miranda, M. D., Nielsen, J. P., & Sperlich, S. (2006). One sided cross validation for density estimation with an application to operational risk. *Available at SSRN 952450*.

Müller, H.-G. (1991). Smooth optimum kernel estimators near endpoints. *Biometrika*, *78*(3), 521–530.

Olga, S. (2009). *Choosing a kernel for cross-validation*. Texas A&M University.

Park, B. U., & Marron, J. S. (1990). Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, *85*(409), 66–72.

Robert, P. (1976). On the choice of smoothing parameters for parzen estimators of probability density functions. *IEEE Transactions on Computers*, *25*(11), 1175–1179.

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 65–78.

Savchuk, O. Y., Hart, J. D., & Sheather, S. J. (2010). Indirect cross-validation for density estimation. *Journal of the American Statistical Association*, *105*(489), 415–423.

Scott, D. W., Tapia, R. A., & Thompson, J. R. (1980). Nonparametric probability density estimation by discrete maximum penalized-likelihood criteria. *The Annals of Statistics*, 820–832.

Scott, D. W., & Terrell, G. R. (1987). Biased and unbiased cross-validation in density estimation. *Journal of the american Statistical association*, *82*(400), 1131–1146.

Sheather, S. J. (1986). An improved data-based algorithm for choosing the window width when estimating the density at a point. *Computational Statistics & Data Analysis*, *4*(1), 61–65.

Sheather, S. J., & Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, *53*(3), 683–690.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). CRC press.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.

Stute, W. (1992). Modified cross-validation in density estimation. *Journal of Statistical Planning and Inference*, *30*(3), 293–305.

Taylor, C. C. (1989). Bootstrap choice of the smoothing parameter in kernel density estimation. *Biometrika*, *76*(4), 705–712.

Vitale, R. A. (1975). A bernstein polynomial approach to density function estimation, In *Statistical inference and related topics*. Elsevier.

Wand, M. P., & Jones, M. C. (1994). *Kernel smoothing*. Crc Press.