Local Orthogonal Polynomial Expansion and Empirical Saddlepoint Approximation for Density Estimation

by

Placida D.A. Dassanayake, M.Sc.

Department of Mathematics and Statistics

A Dissertation in

Statistics

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

Dr. Alex Trindade
Co-Chair

Dr. Igor Volobouev
Co-Chair

Dr. Leif Ellingson
Committee member

Mark Sheridan
Dean of the Graduate School

December 2014

ACKNOWLEDGMENTS

First and foremost I would like to express my sincerest gratitude to my advisors, Dr. Alex Trindade and Dr. Igor Volobouev who have enormously supported me throughout my thesis with their patience and knowledge whilst giving advice during my work. Also I would like to thank Dr. Leif Ellignson for his support and encouragement. During my time at Texas Tech University I had the opportunity to learn and work with many great teachers in the Department of Mathematics and Statistics and I indebted to them.

I am grateful to all those who helped me to complete this thesis. Especially I would like to thank my husband, Achintha Maddumabandara, for his love, continuous support and encouragement and to all my friends in Texas Tech for their help in numerous ways.

Last but not all the least, I am deeply indebted to my family for their dedication, guidance, never-ending advice, and blessings.

TABLE OF CONTENTS

ABSTRACT

This thesis proposes a new non-parametric density estimation method and studies further the so-called Empirical Saddlepoint Approximation (ESPA). We introduce Local Orthogonal Polynomial Expansion (LOrPE), a new method to estimate the density function of a uni-variate continuous random variable. LOrPE is related to several existing methods. In a manner of construction, it is similar to the Local Likelihood Density Estimation (LLDE). LLDE matches localized sample moments to localized population moments using the log-polynomial density approximation while LOrPE matches localized expectation values of orthogonal polynomials to their sample values using polynomial density approximation. We demonstrate that, in the limit of large bandwidth, LOrPE is equivalent to Orthogonal Series Density Estimation (OSDE). In the limit of small bandwidth, LOrPE essentially functions as Kernel Density Estimation (KDE) without boundary bias. We compare the performance of LOrPE to KDE, LLDE and OSDE in a number of tests. In terms of Mean Integrated Squared Error (MISE), our results show that with a proper balance of the two tuning parameters, bandwidth(h) and degree(M), LOrPE performs better than the other methods when reconstructing densities with sharp boundaries.

The Empirical Saddle-point Approximation (ESPA) is a potentially attractive tool for density estimation since it is entirely non-parametric and free of tuning parameters. it consists of inverting the empirical moment generating function via the saddlepoint method. We derive the first order asymptotic regime of the ESPA using both influence functions and M-estimation methodology. These results are used in simulated data sets to construct confidence bands for the resulting densities.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Density Estimation Approaches

In Statistics, the density estimation is considered as a fundamental concept and it has been studies well in past as well as in present. density estimates can provide numerous information about the data at hand. There are several approaches in literature to obtain density estimates such as parametric approach, semi-parametric approach and non-parametric approach. In all these approaches the main goal is to find the probability density function of the data described as follows:
For any random variable X, the probability density function f of X allows us to find any probability associated with X. For example the probability that X belongs to a set A, which is any set of real numbers, can be written as

$$P(X \in A) = \int_A f(x)dx. \tag{1.1}$$

The goal in density estimation is to find an estimate $\hat{f}(x)$ of f using n realizations of the variable X.
Parametric density estimations assume the data are drown from a known family of parametric distribution and then use techniques such as maximum likelihood estimation to estimate the parameters. For example if the data is assumed to be normally distributed. Parametric approach is to estimate the mean μ and variance σ^2 using data.

Sometimes assuming a particular parametric family might not good enough. In that case one can use semi-parametric density estimation methods. For example, if a well established distribution like normal distribution is assumed, only the mean and the co-variance is to be estimated. Then the whole process of density estimation depends on one selected family of distributions. But sometimes the underlying distribution of the data can be a mixture of distributions such as a Gaussian mixture. The basic idea of semi-parametric density estimation methods is to assume that the density estimate is a linear combination of m parametric density models such that

1.2 Classical Non-Parametric Density Estimation Methods

1.2.1 Histograms

Histogram [19] is the oldest density estimator in non-parametric density estimation. Also it is widely used because of it's simplicity and the speed of computation. When an origin x_0 and a binwidth h is given, the histogram estimator of the density is given by

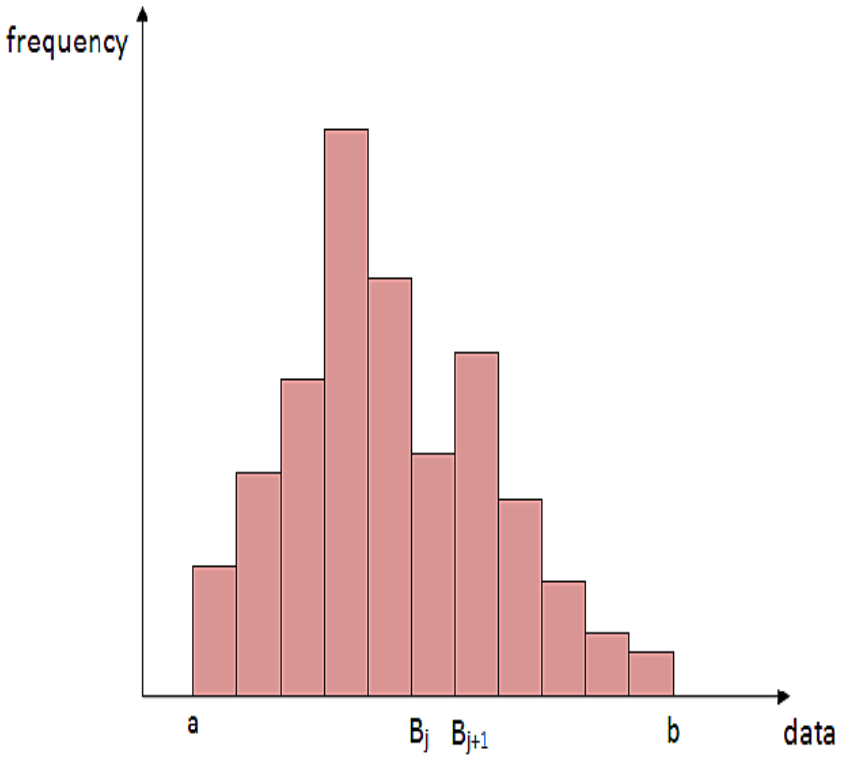


Figure 1.1. Histogram Estimation

$$\hat{f}_N(x) = \frac{1}{N} \sum_{j=0}^{N} \frac{I(x_i \in B_j)}{h}, \tag{1.2}$$

where x_i are sample points and $B_j = [x_0 + jh, x_0 + (j+1)h)$. The sub interval B_j's are chosen closed from the right and open from the left for definiteness.

There are several method of selecting the optimal binwidth given in Scott(1979), Freedman-Diaconis (1981)[7] and Sturges (1926)[20].
A method explained in Wasserman(2006)[24] is as follows:
Under the assumptions that f' is absolutely continuous and that $\int (f')^2 du < \infty$ the optimal binwidth for histogram can be obtained by the following formula [25] :

$$h^* = \frac{1}{n^{1/3}} \left(\frac{6}{\int (f')^2 du} \right)^{1/3} \tag{1.3}$$

The amount of smoothing is determined by the binwidth h.
The bias of the histogram at a point x is given by

$$hf^{'}(x)/2 - f^{'}(x)(x - t_x) + O(h^2)$$

where t_x is the left end point of the bin which contains the point x.
Histograms are useful in exploring the data and identifying the basic pattern or behavior of the data but it has several drawbacks such as the discontinuity of the histogram makes it very difficult if the derivatives is required. Also the estimate depends on the origin of the first bin and binwidth which makes it difficult to use in high dimensional data. Another main disadvantage is that the histogram is extremely sensitive to the binwidth h. Therefore with out the proper selection of binwidth, histogram can be easily over smoothed or under smoothed.

1.2.2 Kernel Density Estimation Method

KDE overcome the above mentioned drawbacks of histograms, specially the discreteness, by placing a smooth kernel function at each data point where the center of the kernel is at the data point and then summing to find the density estimate. KDE is the most studied density estimation method in non-parametric density estimation literature. Also selecting the optimal tuning parameter(h) is well studied in KDE. Given the data, the kernel density estimate is given by

$$\hat{f}_N(x) = \frac{1}{n}\sum_{i=1}^{N}\frac{1}{h}K\left(\frac{x - X_i}{h}\right). \tag{1.4}$$

Where h is the smoothing parameter which is called the bandwidth and K is the kernel function which is a non-negative function(in most practical cases) symmetric around 0 and integrates to 1. If the kernel function takes negative values the estimate it self might have negative values which needs to be adjusted.
Some commonly used kernel functions in kernel density estimation:

- Gaussian kernel : $K(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$
- Epanechnikov kernel : $K(x) = \frac{3}{4}(1 - x^2)I(x)$
- Biweight : $\frac{15}{16}(1 - t^2)^2$; $|t| < 1$
  0 ; otherwise.
- Triangular: $1 - |t|$; $|t| < 1$
  0 ; otherwise.

The optimal bandwidth for KDE is based on minimizing the Mean Integrated Squared Error(MISE) defined as follows:

$$MISE = E\int[\hat{f}_N(x) - f(x)]^2 dx \tag{1.5}$$

For a kernel K, it is asymptotically equals to

$$\frac{\int K^2}{nh} + \frac{\sigma_K^4 h^4 \int (f'')^2}{4} \tag{1.6}$$

Where $\sigma_K^2 = \int x^2 K(x)dx$.
Minimizing Equation 1.6 with an assumption that the underlying distribution is Normal, a common bandwidth can be derived as follows:

$$h = (1.06n^{-1/5})min\left\{\hat{\sigma}, IQR/1.34\right\} \tag{1.7}$$

However there are several other methods has been derived to get the optimal h with out the assumption of Normality. In Scott & Terrell (1987) bandwidth has derived minimizing MISE using cross-validation. In Sheather & Jones (1991), the term $\int (f'')^2$ in Equation 1.6 has been replaced by an estimate using data.
KDE is arguably the "gold-standard" among non-parametric estimation methods. Reasons for this are that, under mild assumptions on f, it can be shown that KDE is consistent, with an AMISE-optimal choice of bandwidth (h_*) which depends on (computable) kernel moments and the (uncomputable) integrated squared curvature of f. Using h_*, it can be shown that the MISE of KDE converges to zero at the rate $O(n^{-4/5})$, which is faster than the rate for histograms.
One main challenge to KDE is the changes in data density. When there is a high data density we could have small bandwidths, but areas with sparse data need large bandwidths. One approach to overcome this problem is to let bandwidth changes in the expression of KDE as follows[12]:

In KDE expression, let bandwidth h is replaced with h_i as follows:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i}K\left(\frac{x - X_i}{h_i}\right) \tag{1.8}$$

Here the bandwidths are proportional to a fixed bandwidth h used to build a primary density estimate $\hat{f}_P$. Hence $h_i = \lambda_i h$, where

$$\lambda_i = \left(\frac{\hat{f}_P}{\left(\prod_{j=1}^{n}\hat{f}_P(X_j)\right)^{\frac{1}{n}}}\right)^{-\alpha} \tag{1.9}$$

This way the bandwidth h_i will tend to be small if the number of observed values

near X_i is large and h_i will be large in order to get an accurate density estimate when the number of observed values near X_i is considerably smaller value.
Also in [19] and in [21] we have similar approaches of variable kernel density estimation.

1.2.2.1 KDE Boundary Correction

Since KDE places a fixed kernel centered at each observation and summing them it can be problematic if the underlying true density has a sharp boundary. To overcome this problem at the boundary there exists a vast literature about KDE boundary correction methods under the standard symmetric fixed KDE. For example the reflection method (see [4]), the cut and normalized method(see [8]) and also some superior methods such as boundary kernel method [13] and generalized reflection method (see [14]) are in use.
A recent study on boundary correction for KDE by Malec and Schienle suggests choosing asymmetric gamma kernels instead of the symmetric fixed kernels to improve the density estimation at the boundary. They also highlighted that the asymptotic performances of gamma kernel estimates are remarkably better than any fixed kernel boundary adjustment methods.

1.2.3 Orthogonal Series Density Estimation Method

OSDE is less commonly used compared to KDE. OSDE is representing the density f in terms of an orthogonal series expansion under a specified basis. The scalar coefficients in the expansion are determined by the data. In a polynomial expansion the number of coefficients is infinite but a finite number has to be selected for practical use. The following is the basic idea of constructing a non-parametric density estimate using OSDE. Suppose that the density function $f(x)$ is the true density function of a continuous random variable X. It can be approximated by a series $f_N(x)$,

$$f_N(x) = \sum_{i=1}^{N} \theta_i \, \varphi_i(x); \quad \text{as} \quad N \quad \text{becomes larger.} \tag{1.10}$$

Where $\{\varphi_i(x), i = 0, 1, ...\}$ an orthonormal basis. N is the cut off and $\varphi_i(x)$'s are fixed and known functions,

$$\theta_j = \int f(x)\,\varphi_j(x)\,dx = E[\varphi_j(x)] \tag{1.11}$$

the expectation of $\varphi_j(x)$ is θ_j.

Therefore we can estimate $\hat{\theta_j}$ by sample mean of $\varphi_i(x)$'s,

$$\hat{\theta_j} = \frac{1}{n}\sum_{l=1}^{n}\varphi_j(x_l) \tag{1.12}$$

Therefore the series estimate with cut off N is

$$\hat{f}_N(x) = \sum_{j=0}^{N}\hat{\theta}_j\varphi_j(x) \tag{1.13}$$

Choosing the orthogonal basis is usually depends on the support of the density. For infinite support, Hermite and Laguerre polynomials are used and for density support [-1,1] Legendre or Chebyshev series are recommended. Also if the density decays quickly or it defines on a circle, then Trigonometric or Fourier Series are recommended. The choice of Cosine basis is convenient because it nicely approximates aperiodic functions.

Also selecting an appropriate number of coefficients (cutoff N) so that the estimate represents the underlying data is one main challenge in OSDE. Various studies has been done selecting the optimal cut off by minimizing MISE. Efromovich [5] mentions three different approaches of reaching optimal cutoff J given in Tarter & Kronmal [15], Hart [9] and Diggle & Hall [3]

1.2.4 Local Likelihood Density Estimation Method

The LLDE matches localized sample moments to population moments using the log-polynomial density approximation.The most general formulation is by Hjort & Jones [11], where we have data $X_1, X_2, ..., X_n$ with an unknown density $f(x)$. The generalize the global log-likelihood to accommodate non-negative f's that don't necessarily integrate to 1,

$$\mathcal{L}(f) = \frac{1}{N}\sum_{i=1}^{N} \log f(x_i) - \left(\int f(u)du - 1 \right), \tag{1.14}$$

If f is a density Equation 1.14 is equivalent to the usual log-likelihood. For a kernel K, the local log-likelihood at x is defined to be a locally smoothed version of the global log-likelihood

$$\mathcal{L}_x(\theta) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x}{h}\right) \log f(x_i;\theta) - \frac{1}{h}\int K\left(\frac{u-x}{h}\right) f(u;\theta)du. \tag{1.15}$$

By maximizing Equation 1.15 over $\theta \in \mathbb{R}^{p+1}$ to obtain $\hat{\theta}(x)$, and the respective estimates of $f(x)$:
Loader [16] uses the local polynomial approximation to $log(f(u))$ such that $log(f(u))$ can be approximated by a low-degree polynomial in the neighborhood of the fitting point x as follows:

$$log(f(u)) \approx P(u-x) \tag{1.16}$$

Where $P(u-x) = a_0 + a_1(u-x) + ... + a_p(u-x)^p$ Using the above equation we can re-write Equation 1.15:

$$\mathcal{L}_x(\theta) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x}{h}\right) \log P(X_i - x) - \frac{1}{h}\int K\left(\frac{u-x}{h}\right) exp(P(X_i - x))du. \tag{1.17}$$

For a fixed point x on the grid with $(\hat{a}_0, \hat{a}_1, ..., \hat{a}_p)$ the maximizer of Equation 1.17the local likelihood density can be written as follows:

$$\hat{f}(x) = exp(\hat{a}_0) \tag{1.18}$$

If Equation 1.17 has no maximizer, then $\hat{f}(x) = 0$

1.3 Motivation to LOrPE Method

One major drawback of KDE is the boundary bias, specially when estimating densities with sharp edges. In that case KDE doesn't achieve it's expected convergence rate(Jones, 1993). One solution to overcome this situation is to use variable-bandwidth approach as given in Equation (1.8). Another way is to use a local polynomial or likelihood based method, in particular LLDE. LLDE lessen the boundary bias but involves solving non-linear equations at each x which makes the method computationally slow. Variable-bandwidth method also faces equivalent obstacles dealing with selecting multiple bandwidths(e.g., Zhu *et al.*, 2006). Therefore there a need for a non-parametric density estimation method which overcomes the boundary bias issue.

LOrPE method was initially formed based on a problem in high energy physics where there is a need to estimate the total energy in jets using calorimeter(Volobouev, 2011). Starting with the particle with given momentum and direction, the interactions of the particle with the detector media can be simulated and those interactions are random processes. Therefore the measured value y from the measuring device is also a random quantity.

The initial idea is to estimate the distribution using a flexible parametric model (Johnsons curves, Elderton & Johnson, 1969)[6] and to describe the deviations non-parametrically with the knowledge of 'composite distributions' as follows:

Let x_i be the observations from an unknown density *f*.

1. Approximate f with parametric Johnsons curves and suppose the parametric estimate of f is g.

2. Then the approximated cumulative distribution function(CDF) *G(x)* can be used as a transformation which maps x_i to the [0,1] interval $s.t.$ $G(x_i) = y_i$.

3. Then the distribution of y_i can be approximated non-parametrically and let the non-parametric estimate be ψ.

4. Finally the initial distribution f can be written as follows: $f(x) = g(x)\psi(G(x))$.

Step 3 is where LOrPE method comes in since it works well estimating densities with sharp boundaries.

In next chapter presents LOrPE method in details and how it handles the boundary bias.

CHAPTER II
LOCAL ORTHOGONAL POLYNOMIAL EXPANSIONS(LORPE) FOR DENSITY ESTIMATION

2.1 LOrPE Method

The LOrPE method (By Dr. Igor Volobouev)for non-parametric density estimation will be introduced in this chapter. Also how LOrPE handles estimating densities with sharp boundaries and the connection of LOrPE method to some other non-parametric density estimation methods will be discussed.

Suppose, we have an i.i.d. sample of measurements x_i, $i = 1, 2, ..., N$ from a continuous pdf $f(x)$ defined on the support $[a, b]$. First consider a simple estimator $\hat{f}_N(x)$ of $f(x)$, which we take to be the empirical density function (EPDF)

$$\hat{f}_N^{\mathrm{EMP}}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x_i), \tag{2.1}$$

where $\delta(x)$ is the Dirac delta function. LOrPE amounts to constructing a truncated expansion of the EPDF into orthogonal polynomial series near each point $x_{\,\mathrm{fit}}$ where we want to build the density estimate. For a chosen bandwidth h, this expansion looks as follows

$$\hat{f}_N^{\mathrm{LOrPE}}(x) = \sum_{k=0}^{M} c_k(x_{\,\mathrm{fit}}, h) P_k\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right). \tag{2.2}$$

The polynomials $P_k(x)$ are built to satisfy the normalization condition

$$\frac{1}{h}\int_a^b P_j\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) P_k\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) K\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) dx = \delta_{jk}, \tag{2.3}$$

which is equivalent to

$$\int_{(a - x_{\,\mathrm{fit}})/h}^{(b - x_{\,\mathrm{fit}})/h} P_j(y) P_k(y) K(y) dy = \delta_{jk}, \tag{2.4}$$

where δ_{jk} is the Kronecker delta, and $K(x)$ is a suitably chosen kernel function.

Remark 2.1. *For commonly used kernels from the beta family (Epanechnikov, Biweight, Triweight, etc.), condition (2.4) generates the normalized Gegenbauer polynomials (up to a common multiplicative constant) at grid points x_{fit} sufficiently deep inside the support interval, provided h is small enough to guarantee that $(a-x_{fit})/h \leq -1$ and $(b-x_{fit})/h \geq 1$.*

Proof. By definition, the normalized Gegenbauer polynomials, $P_j^{(\alpha)}(x)$, $j = 0, 1, \ldots$, are orthonormal on $[-1, 1]$ with respect to the weight function $w(x) = (1-x^2)^{\alpha-1/2}$, for some $\alpha \geq -1/2$. This means that

$$\int_{-1}^{1} P_j^{(\alpha)}(x) P_k^{(\alpha)}(x) w(x)\, dx = \delta_{jk}. \tag{2.5}$$

Noting that $w(x) = c_\alpha K(x)$, where $K(x) = c_\alpha^{-1}(1-x^2)^{\alpha-1/2} I_{[-1,1]}(x)$ is a beta kernel with associated normalizing constant $c_\alpha = \Gamma(\alpha+1)/[\sqrt{\pi}\Gamma(\alpha+1/2)]$, equation (2.5) becomes

$$\delta_{jk} \;=\; \int_{-1}^{1} P_j^{(\alpha)}(x) P_k^{(\alpha)}(x) c_\alpha K(x)\, dx = c_\alpha \int_{(a-x_{\,\mathrm{fit}})/h}^{(b-x_{\,\mathrm{fit}})/h} P_j^{(\alpha)}(x) P_k^{(\alpha)}(x) K(x)\, dx,$$

since $K(x) = 0$ outside of $[-1, 1]$ and $(a - x_{\,\mathrm{fit}})/h \leq -1$ and $(b - x_{\,\mathrm{fit}})/h \geq 1$. This requires extending the polynomials so that they are also defined on $[(a - x_{\,\mathrm{fit}})/h, (b - x_{\,\mathrm{fit}})/h]$. While this extension is not unique, any reasonable definition will do, e.g. by using the same coefficients as on the $[-1, 1]$ interval. Values of $\alpha = 3/2, 5/2, 7/2, 9/2$ define respectively the Epanechnikov, Biweight, Triweight, and Quadweight kernels. □

If $x_{\,\mathrm{fit}}$ is sufficiently close to the ends of the support $[a, b]$ relative to the kernel support, then, since the kernel is used as the weight function in generating them, the polynomials will vary depending on $x_{\,\mathrm{fit}}$, and the notation $P_k(\cdot, x_{\,\mathrm{fit}})$ would be more appropriate. However, we will usually suppress this dependence on $x_{\,\mathrm{fit}}$ in order to simplify the notation.

The expansion coefficients $c_k(x_{\text{fit}}, h)$ are determined by

$$c_k(x_{\text{fit}}, h) = \frac{1}{h} \int \hat{f}_N(x) P_k((x - x_{\text{fit}})/h) K((x - x_{\text{fit}})/h) dx. \tag{2.6}$$

For the empirical EPDF $\hat{f}_N(x) = \hat{f}_N^{\text{EMP}}(x)$, this is equivalent to

$$c_k(x_{\text{fit}}, h) = \frac{1}{Nh} \sum_{i=1}^{N} P_k((x_i - x_{\text{fit}})/h) K((x_i - x_{\text{fit}})/h). \tag{2.7}$$

Here equation (2.6) can be interpreted as the expected values of $P_k((x - x_{\text{fit}})/h)$ w.r.t. the empirical density (EPDF(x)) and the kernels $\frac{1}{h} K((x_i - x_{\text{fit}})/h)$ can be considered as the localization weights.

When calculated for all possible values of k and h, the system of expansion coefficients $c_k(x_{fit}, h)$ provides an enormous amount of information about $p_{emp}(x)$. In fact, the coefficients $c_k(x, h)$ can be build for all kinds of functions, not just empirical densities. Just like the windowed Fourier transform or wavelet transform, this expansion can be turned into a very useful general tool for signal analysis, filtering, compression, etc. It would be interesting to understand whether a fast algorithm for calculating $c_k(x, h)$ can be developed. One can perhaps find a specific kernel K(x) and a grid of points x_α such that the coefficients $c_k(x, h_m)$ found for a scale h_m can be reused for calculating coefficients at a smaller scale hn and a denser grid of points x_β. Existence of such computational techniques as the fast wavelet transform, fast Gauss transform, and Gauss-Kronrod quadrature rules provide a tantalizing hint that a fast algorithm for calculating $c_k(x, h)$ might exist.

The density estimate at x_{fit} is $\max\{0, \hat{f}_N^{\text{LOrPE}}(x_{\text{fit}})\}$.
In general, LOrPE does not produce a bona fide density (in this respect it is similar to the orthogonal series estimator), and after calculating the density for all x_{fit} values one has to perform the overall renormalization.

Equation 2.2 can be usefully generalized as follows:

$$\hat{f}_N^{\mathrm{LOrPE}}(x) = \sum_{k=0}^{\infty} g(k) c_k(x_{\,\mathrm{fit}}, h) P_k((x - x_{\,\mathrm{fit}})/h). \tag{2.8}$$

$g(k)$ is a *taper function*. Normally, $g(0) = 1$ and there is an integer M such that $g(k) = 0$ for any $k > M$. The taper function suppresses high order terms gradually instead of using a sharp cutoff at M.

2.2 Connection Between LOrPE and KDE

Theorem 1. *When evaluated at points x_{fit}, (2.8) is equivalent to kernel density estimator (KDE) with the effective kernel*

$$K_{eff}(x) = \sum_{k=0}^{\infty} g(k) P_k(0) P_k(-x) K(-x). \tag{2.9}$$

Under the following additional assumptions:

(1) $K(x)$ is an even kernel supported on some interval $(-a_K, a_K)$ that is symmetric about 0;

(2) x_{fit} is sufficiently far away from the density support boundaries $[a, b]$ so that the $P_k(\cdot)$'s can be generated on an interval interval of orthogonality that is symmetric about 0 and subsequently extended to $[\tilde{a}_{fit}, \tilde{b}_{fit}]$ by keeping the same coefficients, where $\tilde{a}_{fit} \equiv (a - x_{fit})/h$ and $\tilde{b}_{fit} \equiv (b - x_{fit})/h$, as in the proof of Remark 2.1; and

(3) we have $\tilde{a}_{fit} \leq -a_K < a_K \leq \tilde{b}_{fit}$;

then the effective kernel (2.9):

(i) is an even function supported on $(-a_K, a_K)$;

(ii) is normalized provided $g(0) = 1$; and

(iii) is a high-order kernel if $g(k)$ is a step function, i.e. $g(k) = 1$ for all $k \leq M$ and $g(k) = 0$ for all $k > M$, in which case the kernel order is $M + 1$ if M is odd and $M + 2$ if M is even.

Proof. Let $\hat{f}_N^{\mathrm{KDE}}$ denote the KDE of $f(x)$ for kernel K and bandwidth h,

$$\hat{f}_N^{\mathrm{KDE}}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right).$$

Let $\hat{f}_N^{\mathrm{LOrPE}}$ denote the LOrPE of $f(x)$ for polynomials $P_k(\cdot)$, coefficients $c_k(\cdot)$, bandwidth h, and taper function $g(\cdot)$, on a grid of points whose generic element is $x_{\,\mathrm{fit}}$

$$\hat{f}_N^{\mathrm{LOrPE}}(x) = \sum_{k=0}^{\infty} g(k) c_k(x_{\,\mathrm{fit}}, h) P_k((x - x_{\,\mathrm{fit}})/h).$$

Plugging in the expression for $c_k(\cdot)$ from (2.7) gives

$$\begin{aligned}
\hat{f}_N^{\mathrm{LOrPE}}(x) &= \sum_{k=0}^{\infty} g(k) c_k(x_{\,\mathrm{fit}}, h) P_k\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) \\
&= \sum_{k=0}^{\infty} g(k) \left\{ \frac{1}{Nh} \sum_{i=1}^{N} P_k\left(\frac{x_i - x_{\,\mathrm{fit}}}{h}\right) K\left(\frac{x_i - x_{\,\mathrm{fit}}}{h}\right) \right\} P_k\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) \\
&= \frac{1}{Nh} \sum_{i=1}^{N} \underbrace{\left\{ \sum_{k=0}^{\infty} g(k) P_k\left(\frac{x - x_{\,\mathrm{fit}}}{h}\right) P_k\left(\frac{x_i - x_{\,\mathrm{fit}}}{h}\right) K\left(\frac{x_i - x_{\,\mathrm{fit}}}{h}\right) \right\}}_{K_{\,\mathrm{eff}}\left(\frac{x - x_i}{h}\right)} \\
&= \frac{1}{Nh} \sum_{i=1}^{N} K_{\,\mathrm{eff}}\left(\frac{x - x_i}{h}\right)
\end{aligned}$$

Defining $y = (x_{\,\mathrm{fit}} - x_i)/h$, evaluate $K_{\,\mathrm{eff}}$ at grid point $x_{\,\mathrm{fit}}$ to see that

$$K_{\,\mathrm{eff}}\left(\frac{x_{\,\mathrm{fit}} - x_i}{h}\right) \equiv K_{\,\mathrm{eff}}(y) = \sum_{k=0}^{\infty} g(k) P_k(0)\, P_k(-y)\, K(-y).$$

To establish (i)-(iii), note that Assumptions 1 & 2 imply that $P_k(x)$ is an even (odd) function for any even (odd) integer k

This means $P_k(-x) = P_k(x)$ for k even, and $P_k(0) = 0$ for k odd, so that the effective kernel becomes

$$K_{\,\mathrm{eff}}(x) = \sum_{\{k\,:\,k \geq 0,\ k \text{ even}\}} g(k) P_k(0)\, P_k(x)\, K(x), \tag{2.10}$$

and $K_{\,\mathrm{eff}}(-x) = K_{\,\mathrm{eff}}(x)$ is an even function supported also on $(-a_K, a_K)$, thus establishing (i).

Now, multiplying both sides of the above equation by $P_0(x) \equiv 1$ and integrating

gives

$$\begin{aligned}
\int_{\mathbb{R}} K_{\text{eff}}(x)dx &= \int_{-a_K}^{a_K} K_{\text{eff}}(x)P_0(x)dx \\
&= \int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} \sum_{\{k\,:\,k\geq 0,\ k \text{ even}\}} g(k)P_k(0)P_0(x)P_k(x)K(x)dx, \qquad \text{Assumption (3)} \\
&= \sum_{\{k\,:\,k\geq 0,\ k \text{ even}\}} g(k)P_k(0)\int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} P_0(x)P_k(x)K(x)dx, \text{swap integral and sum} \\
&= \sum_{\{k\,:\,k\geq 0,\ k \text{ even}\}} g(k)P_k(0)\delta_{0\,k}, \qquad \text{from (2.4)} \\
&= g(0)P_0(0) = g(0),
\end{aligned}$$

which establishes (ii).

To prove (iii), first define the j-th kernel moment as

$$\mu_j(K_{\text{eff}}) \equiv \int_{\mathbb{R}} x^j K_{\text{eff}}(x)dx.$$

Now, since the effective kernel is an even function, it is clear $\mu_j(K_{\text{eff}}) = 0$ for j odd. Hence, it suffices to consider the case when j is even, whence

$$\mu_j(K_{\text{eff}}) = \int_{\mathbb{R}} x^j K_{\text{eff}}(x)dx = \sum P_k(0)\int_{-a_K}^{a_K} x^j P_k(x)K(x)dx = \sum \alpha_{jk}P_k(0),$$

Where Summation is taken over $\{k : 0 \leq k \leq M, k \; is \; even\}$.

if we define

$$\alpha_{jk} = \int_{-a_K}^{a_K} x^j P_k(x)K(x)dx = \int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} x^j P_k(x)K(x)dx.$$

Now, from the theory of orthogonal polynomials, we know that

$$x^j = \sum_{k=0}^{j} a_{jk}P_k(x), \qquad \text{where} \qquad a_{jk} = \int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} x^j P_k(x)K(x)dx = \alpha_{jk}.$$

Since α_{jk} is the coefficient of the $P_k(x)$ contribution (a polynomial of order k) to the

series expansion of x^j, it is obvious that $\alpha_{jk} = 0$ for $k > j$, and $\alpha_{jk} = 0$ when k and j have opposite parity (only even k terms contribute when j is even, and vice-versa). With these observations, it is clear that for $j \leq M$

$$\mu_j(K_{\text{eff}}) = \sum_{k=0}^{M} \alpha_{jk} P_k(0), \qquad \text{and} \qquad x^j = \sum_{k=0}^{M} \alpha_{jk} P_k(x),$$

whence we see that

$$\mu_j(K_{\text{eff}}) \;\; = \;\; x^j\big|_{x=0} = \begin{cases} 1, & j = 0, \\ 0, & j = 1, \ldots, M. \end{cases}$$

If M is even, then since $M + 1$ is odd and $K_{\text{eff}}(x)$ is an even function, we have additionally $\mu_{M+1}(K_{\text{eff}}) = 0$. Thus the effective kernel order is $M + 1$ if M is odd, and $M + 2$ if M is even. □

Remark 2.2. *Unlike KDE, however, LOrPE does not suffer from the boundary bias because (2.4) automatically adjusts the shape of orthogonal polynomials near the boundary.*

One-dimensional KDE with fixed kernel K(x) has only one important parameter which regulates the amount of smoothing: bandwidth h. LOrPE has two such parameters: bandwidth h and the highest polynomial order M (or, in general, the shape of the taper function). It is intuitively obvious that polynomial modeling should result in a smaller bias than KDE for densities with several (at least M) continuous derivatives, and that a proper balance of h and M should result in a better estimator overall (e.g., in terms of mean integrated squared error). In Chapter 3 we present various approaches we used to find such h and M which minimizes MISE.

2.3 Connection Between LOrPE and OSDE

In this section we consider how LOrPE connects with OSDE. There is a strong connection between LOrPE and the *orthogonalseries density estimator* (OSDE)[5] as follows:
If $\{\phi_k\}$ is an orthonormal basis and the pdf of X is square integrable, then the OSDE of $f(x)$ is

$$\hat{f}_N^{\text{OSDE}}(x) = \frac{1}{N}\sum_{j=0}^{J}\sum_{i=1}^{N}\phi_j(x_i)\phi_j(x).$$

The tuning parameters here consist of the choice of basis functions and the number of them, J, to carry in the summation. LOrPE can be thought of as a localized version of OSDE.

Theorem 2. *In the limit as $h \to \infty$, the LOrPE estimate (2.2) for pdf $f(x)$ with finite support $[a, b]$, reduces to OSDE in terms of the basis functions*

$$\phi_j(x) = \sqrt{\frac{2}{b-a}}L_j\left(\frac{2x-a-b}{b-a}\right),$$

where the $\{L_j\}$ are orthonormal Legendre polynomials on $[-1, 1]$.

Proof. As $h \to \infty$ the value of the kernel $K(\cdot)$ becomes less and less dependent on the grid point x_{fit} inside $[a, b]$. In fact, starting from (2.3), note that for very large h, $K((x_i - x_{\text{fit}})/h)$ essentially becomes constant on $[a, b]$. Equation (4) then gives rise to Legendre polynomials since these are generated when integrating with respect to a constant weight function, in a manner similar to Remark 2.1. To see this, start with the orthonormal Legendre polynomials $L_k(x)$ on $[-1, 1]$, satisfying

$$\delta_{jk} = \int_{-1}^{1} L_j(x)L_k(x)dx. \tag{2.11}$$

To construct the corresponding orthonormal system on $[a, b]$, we make the transformation, $y = (2x - a - b)/(b - a)$, so that (2.11) becomes

$$\delta_{jk} = \int_a^b \frac{2}{b-a}L_j\left(\frac{2x-a-b}{b-a}\right)L_k\left(\frac{2x-a-b}{b-a}\right)dx = \int_a^b P_j(x)\,P_k(x)\,dx,$$

where

$$P_k(x) \equiv \sqrt{\frac{2}{b-a}} L_k\left(\frac{2x-a-b}{b-a}\right). \tag{2.12}$$

Now construct an orthonormal system on the interval $[\tilde{a}_{\text{fit}}, \tilde{b}_{\text{fit}}]$, with $K(0)$ used as the weight function instead of 1.

Making the transformation $y = (x - x_{\text{fit}})/h$ and using $K(0)$ as the weight function instead of 1, (2.12) becomes

$$\begin{aligned}
\delta_{jk} &= \int_a^b \frac{1}{\sqrt{K(0)}} P_j(x) \frac{1}{\sqrt{K(0)}} P_k(x) K(0) dx \\
&= \int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} \sqrt{\frac{h}{K(0)}} P_j(yh + x_{\text{fit}}) \sqrt{\frac{h}{K(0)}} P_k(yh + x_{\text{fit}}) K(0) dy \\
&= \int_{\tilde{a}_{\text{fit}}}^{\tilde{b}_{\text{fit}}} \tilde{P}_j(y) \tilde{P}_k(y) K(0) dy
\end{aligned}$$

where

$$\tilde{P}_k(y) \equiv \sqrt{\frac{2h}{(b-a)K(0)}} L_k\left(\frac{2yh + 2x_{\text{fit}} - a - b}{b-a}\right), \tag{2.13}$$

which follows from (2.12).

Now, from the proof of Theorem 1 we have the following expression for LOrPE:

$$\hat{f}_N^{\text{LOrPE}}(x) = \frac{1}{Nh} \sum_{i=1}^{N} \sum_{k=0}^{M} P_k\left(\frac{x - x_{\text{fit}}}{h}\right) P_k\left(\frac{x_i - x_{\text{fit}}}{h}\right) K\left(\frac{x_i - x_{\text{fit}}}{h}\right).$$

Substituting $\tilde{P}_k(\cdot)$ for $P_k(\cdot)$ in the above equation, gives

$$\begin{aligned}
\hat{f}_N^{\mathrm{LOrPE}}(x) &= \frac{1}{Nh}\sum_{i=1}^{N}\sum_{k=0}^{M}\tilde{P}_k\left(\frac{x-x_{\,\mathrm{fit}}}{h}\right)\tilde{P}_k\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right)K\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right)\\
&= \frac{1}{Nh}\sum_{i=1}^{N}\sum_{k=0}^{M}\frac{2h}{(b-a)K(0)}L_k\left(\frac{2\left(\frac{x-x_{\,\mathrm{fit}}}{h}\right)h+2x_{\,\mathrm{fit}}-a-b}{b-a}\right)\\
&\quad L_k\left(\frac{2\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right)h+2x_{\,\mathrm{fit}}-a-b}{b-a}\right)K\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right)\\
&= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{M}\frac{2}{(b-a)K(0)}L_k\left(\frac{2x-a-b}{b-a}\right)L_k\left(\frac{2x_i-a-b}{b-a}\right)K\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right).
\end{aligned} \tag{2.14}$$

Since

$$\lim_{h\to\infty}K\left(\frac{x_i-x_{\,\mathrm{fit}}}{h}\right)=K(0),$$

we obtain

$$\hat{f}_N^{\mathrm{LOrPE}}(x)=\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{M}\sqrt{\frac{2}{b-a}}L_k\left(\frac{2x-a-b}{b-a}\right)\sqrt{\frac{2}{b-a}}L_k\left(\frac{2x_i-a-b}{b-a}\right), \tag{2.15}$$

which is classical OSDE in terms of the orthogonal polynomials

$$\phi_k(x)=\sqrt{\frac{2}{b-a}}L_k\left(\frac{2x-a-b}{b-a}\right).$$

□

CHAPTER III
METHODS OF SELECTING OPTIMAL BANDWIDTH(H) AND OPTIMAL DEGREE(M) FOR LORPE METHOD

3.1 Approaching Minimum MISE by a Bandwidth-Degree Scan

In this approach we used the fact that LOrPE is similar to high-order KDE Theom 1(the order, r, is a function of M). We used the AMISE-optimal value of h [23] for KDE as a starting point for a bandwidth scan and on a grid of h and M we compute LOrPE and hence the MISE(more details on the calculations will be on chapter 5). Then obtained the corresponding h and M on the grid which minimizes the MISE. We considered this method as an oracle method is worth trying [2] because it will provide enough evidence to show that LOrPE wins KDE estimating densities with sharp edges.

The following computations illustrates how to obtain $\hat{h}_{AMISE}$ using AMISE-optimal value of h [23] for KDE:

The AMISE-optimal value of h is given by

$$h_*(r) = \left[\frac{(r!)^2 R_{\mathrm{K}_{\mathrm{eff}}}(r)}{2rN\mu_{\mathrm{K}_{\mathrm{eff}}}(r)^2 R_{f^{(r)}}(r)}\right]^{1/(2r+1)}, \tag{3.1}$$

where

$$\mu_{\mathrm{K}_{\mathrm{eff}}}(r) = \int x^r K_{\mathrm{eff}}(x)dx, \qquad R_{\mathrm{K}_{\mathrm{eff}}}(r) = \int K_{\mathrm{eff}}(x)^2 dx, \qquad R_{f^{(r)}}(r) = \int f^{(r)}(x)^2 dx.$$

Substituting this back into the expression for AMISE, gives

$$\mathrm{AMISE}_{h_*}(r) = \frac{2r+1}{2r}\left[2r(r!)^{-2} R_{\mathrm{K}_{\mathrm{eff}}}(r)^{2r} \mu_{\mathrm{K}_{\mathrm{eff}}}(r)^2 R_{f^{(r)}}(r) N^{-2r}\right]^{1/(2r+1)}. \tag{3.2}$$

The unknown moments $\mu_{\mathrm{K}_{\mathrm{eff}}}(r)$ and $R_{\mathrm{K}_{\mathrm{eff}}}(r)$ can be computed once the underlying kernel $K(\cdot)$ is selected. E.g. for Gaussian kernels we have Hermite polynomials; for beta kernels Gegenbauer polynomials; etc.

For typical values of N (from a few hundred to a few tens of thousands), optimal

values of kernel order r are not likely to be very high. For kernel functions such as beta family or Gaussian the corresponding moments could be tabulated with Mathematica or other similar computer algebra system and then included in the relevant programs. For example, in Appendix C is a code snippet which will do this for the Gaussian kernel with Mathematica. What does look more complicated is the calculation of the boundary contribution into the AMISE and analysis of the bias introduced by the truncation of the reconstructed density when it is forced to be non-negative (with subsequent renormalization).

Thus the only real difficulty is estimation of $R_{f^{(r)}}(r)$.

- From Wand & Jones (1995, sec 2.12) [23], a natural estimator of the rth derivative of f , which generalizes KDE, is

$$\hat{f}_N^{(r)}(x) = \frac{\nu!}{Nh^{r+1}} \sum_{i=1}^{N} K^{(r)}\left(\frac{x - x_i}{h}\right), \qquad \nu = 1,$$

  but this doesn't seem to be much help. This is similar to the estimator proposed by Zhang & Fan (2000) [26], where $\nu = r$ and $K^{(r)}$ is replaced by a higher-order kernel. By a simple transformation, it can be shown that

$$R_{f^{(r)}}(r) = (-1)^r \mathbb{E}[f^{(2r)}(X)],$$

  and thus it suffices to study estimation of functional

$$\psi_s \equiv \mathbb{E}[f^{(s)}(X)], \qquad \text{for } s \text{ even.}$$

- Wand & Jones (sec 3.5) [23] discuss multi-stage direct plug-in algorithms. These all involve iterating a KDE-type estimator of ψ_r with optimal bandwidth that depends on ψ_s, $s > r$. Starting with a rough estimate of ψ_s at some stage, which can be based on that value corresponding to a $N(\mu, \sigma^2)$,

$$\psi_s = \frac{(-1)^{s/2} s!}{(2\sigma)^{s+1}(s/2)!\sqrt{\pi}}, \tag{3.3}$$

  (Wand & Jones,formula 3.7) [23] this is iterated to arrive at some estimate $\hat{\psi}_s$.

However, this is computationally intensive and involves the choice of yet another kernel.

Nevertheless, (3.3) gives us a naive estimate of ψ_{2r}, by using an estimate $\hat{\sigma}$ (e.g. sample standard deviation)

$$\hat{R}_{f^{(r)}}(r) = (-1)^r \psi_{2r} = \frac{(2r)!}{(2\hat{\sigma})^{2r+1} r! \sqrt{\pi}}.$$

(For the case $r = 2$ in the context of KDE for f this is known as *Silverman's Rule*.) Plugging this in for the unknown $R_{f^{(r)}}(r)$ in (3.2) gives

$$\text{AMISE}_{h_*}(r) \approx \frac{2r+1}{4r\hat{\sigma}} \left[\frac{2r(2r!)}{(r!)^3 \sqrt{\pi}} \, \mu_{\text{K}_{\text{eff}}}(r)^2 \left(\frac{R_{\text{K}_{\text{eff}}}(r)}{N} \right)^{2r} \right]^{1/(2r+1)} \tag{3.4}$$

Now minimize this in $r \in \{1, 2, \ldots, 10\}$ to get $\hat{r}$. Finally, plug $\hat{r}$ into (3.1) to obtain

$$\hat{h}_{AMISE} = 2\hat{\sigma} \left[\frac{(\hat{r}!)^3 \sqrt{\pi}}{2\hat{r}(2\hat{r}!)N} \, \frac{R_{\text{K}_{\text{eff}}}(\hat{r})}{\mu_{\text{K}_{\text{eff}}}(\hat{r})^2} \right]^{1/(2\hat{r}+1)} \tag{3.5}$$

As mentioned at the beginning of this section, this $\hat{h}_{AMISE}$ may not the optimal h for LOrPE method. In chapter 5 under simulations, $\hat{h}_{AMISE}$ has been used as a starting point for a bandwidth scan.

3.2 Cross-validation for selecting Optimal h and M

There are two types of cross-validation methods considered in order to find the optimal h and M for LOrPE method. The main reason of considering the cross-validation methods because it doesn't assume anything about the true density as in the situation of the previous section. The two cross-validation methods used to get the optimal h and M are least squares cross-validation and pseudo likelihood cross-validation. .

3.2.1 Least Squares Cross-validation(LSCV) Method

Under LSCV,first consider the integrated squared error(ISE) of the LOrPE density estimation $\hat{f}_{h,m}^{LOrPE}$.

$$ISE_{h,M} = \int (\hat{f}_{h,m}^{LOrPE}(x) - f(x))^2 dx \tag{3.6}$$

Here the goal is to select h and M which minimizes ISE as much as possible. There are three main terms taken in to account by expanding the above expression.

$$ISE_{h,M} = \int (\hat{f}_{h,m}^{LOrPE}(x))^2 dx - 2\int f(x)\hat{f}_{h,m}^{LOrPE}(x)dx + \int (f(x))^2 dx \tag{3.7}$$

- $\int (\hat{f}_{h,m}^{LOrPE}(x))^2 dx$ can be derived using the data set.
- $\int f(x)\hat{f}_{h,m}^{LOrPE}(x)dx$ is the expected value of $\hat{f}_{h,m}^{LOrPE}(x)$ w.r.t. an independent random variable X.
- $\int (f(x))^2 dx$ doesn't depend on h and M.

We can estimate $E\left[\hat{f}_{h,m}^{LOrPE}(x)\right]$ by taking the average:

$$E[\hat{f}_{h,m}^{LOrPE}(x)] = \frac{1}{n}\sum_{i=1}^{n} \hat{f}_{h,m,-i}^{LOrPE}(x_i) \tag{3.8}$$

Where

$$\hat{f}_{h,m,-i}^{LOrPE}(x_i) = \frac{1}{Nh}\sum_{k=0}^{M}\sum_{j=0,j\neq i}^{N-1} P_k((x_j - x_{\text{fit}})/h)K((x_j - x_{\text{fit}})/h)P_k\left(\frac{x - x_{\text{fit}}}{h}\right) \tag{3.9}$$

here $\hat{f}^{LOrPE}_{h,m,-i}(x)$ means leave i^{th} observation in estimating $\hat{f}^{LOrPE}_{h,m,-i}(x)$.
Therefore we can reconsider the equation (3.7)

$$ISE_{h,M} = \int (\hat{f}^{LOrPE}_{h,m}(x))^2 dx - 2E[\hat{f}^{LOrPE}_{h,m}(x)] + \int (f(x))^2 dx \tag{3.10}$$

Since the term $\int (f(x))^2 dx$ doesn't depend on h and M we can arrange equation (3.10) as follows:

$$ISE_{h,M} - \int (f(x))^2 dx = \int (\hat{f}^{LOrPE}_{h,m}(x))^2 dx - 2E[\hat{f}^{LOrPE}_{h,m}(x)] \tag{3.11}$$

This gives the so-called Least squares cross-validation criterion as follows:

$$LSCV(h,M) = \int (\hat{f}^{LOrPE}_{h,m}(x))^2 dx - -2E[\hat{f}^{LOrPE}_{h,m}(x)] \tag{3.12}$$

Minimization of LSCV(h,M) can leads to reasonable h and M estimates. Also Minimization of LSCV(h,M) can be leads to multiple minimums. Therefore it could be done in a grid of h and M values and find the h and M with minimum MISE.The main draw back of this process if that the convergence of these h and M estimates towards the MISE optimal h and M might be very slow.

3.2.2 Pseudo Likelihood Cross-validation Method

The psuedo likelihood cross-validation is based on taking the likelihood function of the density estimate $\hat{f}^{LOrPE}_{h,m,-i}(x)$, which is the leave-one-out density estimate used in least-squares cross-validation.
Psuedo cross-validation(PLCV) function is given by the following formula:

$$PLCV(h,M) = \prod_{i=0}^{N-1} \hat{f}^{LOrPE}_{h,m,-i}(x_i) \tag{3.13}$$

The above criterion doesn't work in some situations such as

- when $\hat{f}^{LOrPE}_{h,m,-i}(x_i) = 0$ for any x_i
- when applying PLCV for a density with infinite support and has strong influence fluctuations near tails of the distribution.

A regularization condition has been introduced to avoid these situations(NPStat package, Volobouev, 2013) which is called RPLCV given below:

$$RPLCV(h,M) = \prod_{i=0}^{N-1} max\left(\hat{f}^{LOrPE}_{h,m,-i}(x_i), \frac{\hat{f}^{LOrPE}_{h,m,i}(x_i)}{N^{\alpha}}\right) \tag{3.14}$$

Where α is the regularization parameter. PLCV criterion can be used by choosing a reasonable value for α. The effect of selecting α on cross-validation will be discussed under simulations in chapter 5.

3.3 Some Other Ideas Of selecting Optimal h and M

There are couple of ideas under consideration to build a criterion to choose optimal h and M for LOrPE estimator. Those ideas are based on the Aiken Information Criterion (AIC).

AIC is defined as $AIC = -2loglik + 2(NDOF)$;where loglik is the maximized value of the likelihood function for the estimated model and NDOF is the number of free parameters in the model.

Since LOrPE at some x_fit values can be 0, we can use the same empirical regularization condition (3.14) used in PLCV (NPStat Package, Volobouev, 2013). As mentioned in the previous section. Here also the regularization parameter has to be optimized to get the minimum AIC. Now the question is how to select NDOF for the model. There are two suggested methods(by Dr. Igor Volobouev) to compute NDOF for LOrPE method as follows:

Since LOrPE smoother is a linear operator which transforms the (empirical distribution function (EDF) into the smoothed density, we can write $\hat{f}_L = Af$. Here the matrix A is the smoothing matrix.

For LOrPE method, A is a generalized row stochastic matrix and AA^t is positive-semi-definite since A is a real matrix. Therefore all the eigenvalues of AA^t are positive.

1. The first suggestion for NDOF is called $NDOF_1$ written as follows:

$$NDOF_1 = (S/L_{max}) - 1 \tag{3.15}$$

Where S is the sum of the eigenvalues of AA^t and L_{max} is the largest eigenvalue

of AA^t. Here we subtract 1 because of the constrain that the density integration must be equal to 1. This definition is based on the definition on Elements of Stat. Learning, May 2013[10].

2. The second definition for NDOF called $NDOF_2$ which is

$$NDOF_2 = (The\ exponent\ of\ the\ entropy\ of\ the\ AA^t\ eigenspectrum\ -\ 1) \tag{3.16}$$

These two definitions of NDOF are properties of the matrix A and those doesn't depend on the density which the reconstruction is to be done. Both $NDOF_2$ and $NDOF_2$ depend on the width of the density support. When reconstructing a density with an infinite support these values might tend to infinity. Therefore it would be more effective to consider NDOF per unit length.

3. Also another method of dealing with NDOF The next definition is to use the co-variance expression (Elements of Stat. Learning, May 2013) as the NDOF (say $NDOF_3$).

Suppose, we draw samples of N points from distribution with density f(x) and cumulative distribution function F(x). For sample number k, one can define the quantity

$$SL(N, p, k) = \overline{N} \sum_{i=1}^{N} LN_p(F(x_i))$$

Here, LN_p are the Legendre polynomials orthonormal on the interval $[0, 1]$. These polynomials are just $\sqrt{2p+1}L_p(2x+1)$, where L_p are the standard Legendre polynomials of degree p. Obviously, $SL(N, p, k)$ will have mean 0 (for all $p > 0$) and variance $\frac{1}{N}$. For different values of p, $SL(N, p, k)$ will be uncorrelated.

For any LOrPE estimate . $f_n(x)$, one can build the "comparison distribution" $H(x)$ with density $h(x)$ supported on $[0, 1]$. This distribution is defined by the equation . $f_n(x)/f(x) = h(F(x))$ (or just $h(x) = f_n(Q(x))/f(Q(x))$, where $Q(x)$ is the quantile function for distribution $F(x) : F(Q(y)) = y$). Naturally, if . $f_n(x) = f(x)$ then $H(x)$ is just the uniform distribution.

Now, for each, $f_n(x)$ obtained in the sampling process, one can calculate the expectation value of LN_k with respect to the comparison density $h(x) : S.L(N, p, k)$. Then we can define the number of effective degrees of freedom by

$$NDOF_3 = N \sum_{p=1}^{N} Cov(SL(N, p, k), \hat{SL}(N, p, k)) \tag{3.17}$$

See Appendix B for more details on the calculations that has been done under $NDOF_3$

CHAPTER IV
EMPIRICAL SADDLEPOINT APPROXIMATION(SPA) FOR DENSITY ESTIMATION

4.1 SPA Method

This section describes the basic saddlepoint density estimation method. The most fundamental SPA was introduced by Daniels(1945) and his formula approximating a density from its associated moment generating function or the cumulative generating function. The main importance of this method is that it doesn't involve any unknown parameters such as bandwidth or degree which we have to optimize. The following is how saddlepoint density estimation is defined [1]:
For a continuous variable X with unknown density function f, the saddlepoint approximation of f at x is given as

$$\hat{f}(x) = \frac{1}{\sqrt{2\pi \hat{K}''(\hat{s}_x)}} exp\left\{K(\hat{s}_x) - x\hat{s}_x\right\} \tag{4.1}$$

Where $M(s)$ is the moment generating function of density f and $K(s)$ is the cumulative generating function of density f such that $K(s) = log\left(\int e^{sy} dF(y)\right)$. The $\hat{s}$ denotes the unique solution to the equation $K'(\hat{s}) = x; \hat{s} \in (a, b)$. Where the equation $K'(\hat{s}) = x$ is called the saddlepoint equation and the unique solution of the saddlepoint equation $\hat{s}$ is called the saddlepoint at x. The saddlepoint density function $\hat{f}(x)$ has to be normalized using a normalization as follows:

$$c = \int \hat{f}(x)dx \neq 1$$

the normalized saddlepoint density can be written as

$$\overline{f}(x) = c^{-1}\hat{f}(x)$$

The followings are the empirical equations of the functions used in the saddle-point density function:
Let $x_1, x_2, ..., x_n \sim f$ and the empirical mgf $\hat{M}(s) = \frac{1}{n}\sum_{i=1}^{n} e^{sx_i}$.
Also

$$\begin{aligned}
\hat{K}(s) &= log(\sum_{i=1}^{n} e^{sx_i} - log\, n) \\
\hat{K}'(s) &= \frac{\sum_{i=1}^{n} x_i e^{sx_i}}{\sum_{i=1}^{n} e^{sx_i}} \\
&= \frac{\hat{M}'(s)}{\hat{M}(s)} \\
\hat{K}''(s) &= \frac{\hat{M}(s)\hat{M}''(s) - \hat{M}'(s)^2}{\hat{M}(s)^2}
\end{aligned}$$

and $\hat{s}_x$ solves :

$$\hat{K}'(\hat{s}_x) = \frac{\sum_{i=1}^{n} x_i e^{\hat{s}_x x_i}}{\sum_{i=1}^{n} e^{\hat{s}_x x_i}} = x$$

4.2 Influence Functions Method

The estimates of the parameters or functions of parameters are not provided in non-parametric statistics. Instead it estimates statistical functionals such that a functional $\theta = T(F)$ is any function of F; F is the distribution function.The followings are some examples of statistical functionals of some descriptive statistics:
Statistical functionals : mean : $T(F) = \int x dF(x)$
variance : $T(F) = \int (x-\mu)^2 dF(x)$
quantile : $T(F) = F^{-1}(p)$
To estimating the above quantities can be based on calculating $\hat{F}$(the empirical distribution function) using non-parametric maximum likelihood estimator of F.
Is $T(\hat{F}) \overset{a.s.}{\to} T(F)$?
In general the above condition doesn't hold. Therefore it is required to consider certain conditions on the smoothness of T(F) to find when $T(\hat{f})$ is a consistent estimator. Now consider the Gateaux derivative of T at F in the direction of G as follows:

$$L_F(T;G) \quad = \quad lim_{\epsilon \to 0} \left[\frac{T(1-\epsilon)F + \epsilon G - T(F)}{\epsilon} \right] \tag{4.2}$$

The Gateaux derivative from a statistical perspective describes the rate of change in a statistical functional T(F) upon a small amount of contamination by another distribution G. But even if Gateaux derivative exists for a statistical functional it's not necessarily ensures $T(\hat{F}) \overset{a.s.}{\to} T(F)$. The Hadamard differentiability is the key to make it happened.
Any functional T is said to be Hadamard differentiable if, for any sequence ϵ_n such that $\epsilon_n \to 0$ and any sequence D_n such that $sup_x |D_n(x) - D(x)| \to 0$, we have

$$\frac{T(F + \epsilon_n D_n) - T(F)}{\epsilon_n} \to L_F(T;D) \tag{4.3}$$

Therefore for any statistical functional which is Hadamard differentiable,$T(\hat{F}) \overset{P}{\to} T(F)$ A special case of Gateaux derivative called influence function is used in statistics:

$$L(x) \quad = \quad lim_{\epsilon \to 0} \left[\frac{T(1-\epsilon)F + \epsilon \delta_x - T(F)}{\epsilon} \right] \tag{4.4}$$

Where

$$\delta_x(y) = \begin{cases} 0, & \text{if } y < x \\ 1, & \text{if } y \geq x \end{cases} \tag{4.5}$$

The followings are some examples of influence functions of statistical functionals used in the saddlepoint density function:
The influence function of $M(s)$ is

$$\begin{aligned} L_M(y,F) &= lim_{\epsilon\to 0}\frac{T\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-T(F)}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{\int e^{sy}d\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-\int e^{sy}dF(y)}{\epsilon} \\ &= \frac{(1-\epsilon)\int e^{sy}dF(y)+\epsilon\int dI_{[y,\infty]}-\int e^{sy}dF(y)}{\epsilon} \\ &= y-M(s) \end{aligned}$$

The influence function of $K(s)$ is

$$\begin{aligned} L_K(y,F) &= lim_{\epsilon\to 0}\frac{T\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-T(F)}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{log\left[\int e^{sy}d\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}\right]-log\left[\int e^{sy}dF(y)\right]}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{log\left[(1-\epsilon)M(s)+\epsilon e^{sy}\right]-logM(s)}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{e^{sy}-M(s)}{(1-\epsilon)M(s)+\epsilon e^{sy}} \\ &= \frac{e^{sy}-M(s)}{M(s)} \end{aligned}$$

The influence function of $M'(s)$ is

$$M'(s) = \int xe^{sx}dF(x) = T(F)$$

$$\begin{aligned} L'_M(y,F) &= lim_{\epsilon\to 0}\frac{T\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-T(F)}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{\int ye^{sy}d\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-\int ye^{sy}dF(y)}{\epsilon} \\ &= ye^{ys}-M'(s) \end{aligned}$$

The influence function of $M''(s)$ is

$$M''(s)=\int x^2e^{sx}dF(x)=T(F)$$

$$\begin{aligned} L'_M(y,F) &= lim_{\epsilon\to 0}\frac{T\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-T(F)}{\epsilon} \\ &= lim_{\epsilon\to 0}\frac{\int y^2e^{sy}d\left\{(1-\epsilon)F+\epsilon I_{[y,\infty]}(x)\right\}-\int y^2e^{sy}dF(y)}{\epsilon} \\ &= y^2e^{ys}-M''(s) \end{aligned}$$

The whole purpose of this chapter is to find the asymptotic distribution of the saddlepoint density estimate $\hat{f}$ The asymptotic distribution of $\hat{f}$ can be obtained using the influence function method. The following formula gives the asymptotic variance of any statistical functional T:

$$\sigma_T^2=\frac{1}{n}\int L_F^2(T,G)dF(y) \tag{4.6}$$

The saddlepoint density estimate is a function of $\hat{s}_x$. Therefore the asymptotic distribution of $\hat{f}$ can be obtained by the delta method as follows:

$$\sqrt{n}\left(\ \hat{f}(x)-f(x)\ \right)\xrightarrow{d}\ N(0,\tfrac{\partial g}{\partial s_x}\sigma^2_{s_x})$$

The asymptotic variance of $\hat{s}_x$,$\sigma^2_{s_x}$, can be obtained using the equation (4.6) The next section contains the method of calculating the influence function of s_x using the estimating function in m-estimators method.

4.3 M-Estimators Method

Theorem 3. *Define* $\underline{\theta} = (\theta_1, \theta_2)$ *and let the characteristic function is* $\psi_\theta(x) = [\psi_{\theta,1}(x)\ \psi_{\theta,2}(x)]'$ *such that* $E[\psi_\theta(x)] = 0$

$$\sqrt{n}\left(\ \underline{\theta} - \underline{\theta}_0\ \right) \xrightarrow{d}\ N\left(0, V'_{\theta_0} E[\psi_{\theta_0}\psi'_{\theta_0}] V_{\theta_0}\right)$$

where

$$V_{\theta_0}\ = \begin{bmatrix} \frac{\partial E\,\psi_{\theta,1}}{\partial\theta_1} & \frac{\partial E\,\psi_{\theta,1}}{\partial\theta_2} \\ \frac{\partial E\,\psi_{\theta,2}}{\partial\theta_1} & \frac{\partial E\,\psi_{\theta,2}}{\partial\theta_2} \end{bmatrix}$$

Then by the Delta method we can write

$$\sqrt{n}\left(\ g(\underline{\theta}) - g(\underline{\theta}_0)\ \right) \xrightarrow{d}\ N\left(0, [\tfrac{\partial g}{\partial\underline{\theta}}]' V'_{\theta_0} E[\psi_{\theta_0}\psi'_{\theta_0}] V_{\theta_0} [\tfrac{\partial g}{\partial\underline{\theta}}]\right)$$

Theorem 4. *Suppose that a parameter* $\theta = t(F)$ *is determined implicitly through the estimating equation*

$$\int u(y;\theta) dF(y) = 0$$

. Then the influence function for $t(.)$ *is*

$$L_t(y;F) = \frac{u(y;\theta)}{-\int \dot{u}(x;\theta) dF(x)}$$

Proof

replace F by $(1-\epsilon)F + \epsilon H$, where $H(u-y) = I_{[y,\infty)}(u)$, and differentiate the estimating equation w.r.t. ϵ:

$$F_\epsilon(x) = (1-\epsilon)F + \epsilon I_{[y,\infty)}(x)$$

derivative w.r.t. ϵ:

$$\begin{aligned} dF_\epsilon(x) &= (1-\epsilon)dF + \epsilon I_{[y,\infty)}(x) \\ \frac{dF_\epsilon(x)}{d\epsilon} &= -dF(x) + dI_{[y,\infty)}(x) \end{aligned}$$

The derivative of the estimating equation:

$$\begin{aligned} 0 &= \int \frac{\partial}{\partial \epsilon} u(y; t(F_\epsilon)) dF_\epsilon(x) \\ 0 &= \int \dot{u}(x; t(F_\epsilon)) \frac{\partial t(F_\epsilon)}{\partial \epsilon} + \int u(y; t(F_\epsilon)) \frac{\partial}{\partial \epsilon} dF_\epsilon(x) \end{aligned}$$

Evaluating both sides the equation at $\epsilon = 0$

$$\begin{aligned} \frac{\partial t(F_\epsilon)}{\partial \epsilon} \int \dot{u}(x; t(F_\epsilon)) dF(x) &= -\int u(y; t(F_\epsilon))[-dF(x) + dI_{[y,\infty)}(x)] \\ &= \int u(y; t(F_\epsilon)) dF(x) - \int u(y; t(F_\epsilon)) dI_{[y,\infty)}(x) \\ &= 0 + u(y, t(F)) \end{aligned}$$

Therefore

$$\begin{aligned} L_t(y, F) &= \frac{\partial t(F_\epsilon)}{\partial \epsilon} \\ &= -\frac{u(y, t(F))}{\int \dot{u}(x; t(F_\epsilon)) dF(x)} \end{aligned}$$

4.4 Asymptotic Distribution of Saddlepoint Density Function using Influence functions method

Now in order to get the influence function of $\hat{s}_x$ we need the estimating function of $\hat{s}_x$.

Estimating function of $\hat{s}_x$ is $u(y; \theta)) = ye^{\theta y} - xe^{\theta y}$

$$L_t(y, F) = \frac{ye^{\theta y} - xe^{\theta y}}{M''(\theta) - xM'(\theta)} \tag{4.7}$$

then the variance of $\hat{s}_x$

$$\begin{aligned} n\sigma^2_{\hat{s}_x} &= \int L_t^2(y,\hat{s}_x)dF(y) \\ &= \int \left(\frac{ye^{\hat{s}_x y} - xe^{\hat{s}_x y}}{M''(\hat{s}_x) - xM'(\hat{s}_x)}\right)^2 dF(y) \\ &= \left(\frac{M''(2\hat{s}_x) - xM'(\hat{s}_x)}{M''(\hat{s}_x) - xM'(\hat{s}_x)}\right)^2 \end{aligned}$$

Therefore to get the asymptotic distribution of the saddlepoint density estimation we can apply the Delta method because the saddlepoint density, $\hat{f}(x)$ is a function of $\hat{s}_x$:

$$\begin{aligned} \hat{f}(x) &= \frac{1}{\sqrt{2\pi\hat{K}''(\hat{s}_x)}} exp\left\{\hat{K}(\hat{s}_x) - x\hat{s}_x\right\} \\ &= g(s_x) \end{aligned}$$

Therefore by the Delta method the asymptotic distribution of $\hat{f}(x)$ is :

$$\sqrt{n}\left(\ \hat{f}(x) - f(x)\ \right) \xrightarrow{d} N(0, \left(\frac{\partial g}{\partial s_x}\frac{M''(2\hat{s}_x) - xM'(\hat{s}_x)}{M''(\hat{s}_x) - xM'(\hat{s}_x)}\right)^2)$$

Where $\frac{\partial g}{\partial s_x} = -\hat{f}(s_x)\left[K''(s_x)s_x + \frac{K'''(s_x)}{2K''(s_x)}\right]$
The 95% confidence interval for $\hat{f}(\hat{s}_x)$ is

$$\left(\hat{f} \pm 1.96 \times \frac{\sigma}{\sqrt{n}}\right)$$

where $\sigma = \left(\frac{\partial g}{\partial s_x}\frac{M''(2\hat{s}_x) - xM'(\hat{s}_x)}{M''(\hat{s}_x) - xM'(\hat{s}_x)}\right)$

CHAPTER V
SIMULATIONS AND CONCLUSIONS

5.1 Simulations and Conclusions under LOrPE Method

5.1.1 Comparing LOrPE to KDE using Bandwidth-Degree Scan

The main goal of this section is to compare LOrPE and KDE where KDE is the main competitor for LOrPE. Also to check whether LOrPE is better than KDE in estimating densities with sharp boundaries. Another goal is to understand the optimal h and M of the LOrPE method. For that, Some known distributions were selected, including ones which truncate sharply at the boundary, to be reconstructed by LOrPE and KDE. To come up with a standard set of distributions, the distribution list of Wand & Jones(Table 2.2) [23] which includes increasingly difficult pdf's for KDE to be estimated (according to the $D(f^*)/D(f)$ criterion) has been used.

$$\frac{D(f^*)}{D(f)} = \frac{35}{243}\left[\sigma(f)^5 \int f''(x)^2 dx\right]^{-1/4}, \qquad \sigma(f)^2 = \mathrm{Var}(f),$$

In the list below, of distributions, let $\phi(z)$ and $\Phi(z)$ denote the pdf and cdf of a $\mathrm{N}(0,1)$. There are seven different distributions to be reconstructed by LOrPE and KDE. The first obvious choice is the standard Gaussian distribution. Then there are two Gaussian mixtures named Normal mix 1 and Normal mix 2. Also Beta(4,4) which is another symmetric distribution. Then there are three distributions with sharp boundaries, N(0,1) truncated at 0, N(0,1) truncated at -1 and Exponential(1). It is expected that KDE would handle N(0,1) truncated at 0 using data reflection method(data mirroring). It would be interesting to see performance of LOrPE compares with KDE in this ideal setting for KDE. It is expected to see that LOrPE will outperform KDE when estimating N(0,1) truncated at -1 as data reflecting method doesn't work in that case(due to the first derivative discontinuity).

In the table below, let $\phi(z)$ and $\Phi(z)$ denote the pdf and cdf of a $\mathrm{N}(0,1)$.

Name	Distribution of X and Simulation Notes	$D(f^*)/D(f)$
N(0,1)	$f(x) = \phi(x)$	1.47
Normal mix 1 (bimodal)	$X \sim \frac{3}{4}Z_1 + \frac{1}{4}Z_2$, $B \sim \mathrm{Bern}(3/4)$ $Z_1 \sim \mathrm{N}(0,1)$, $Z_2 \sim \mathrm{N}(3/2, 1/9)$ $\Rightarrow\ X = BZ_1 + (1-B)Z_2$	0.57
Normal mix 2 (sharp peak at 0)	$X \sim \frac{2}{3}Z_1 + \frac{1}{3}Z_2$, $B \sim \mathrm{Bern}(2/3)$ $Z_1 \sim \mathrm{N}(0,1)$, $Z_2 \sim \mathrm{N}(0, 1/100)$ $\Rightarrow\ X = BZ_1 + (1-B)Z_2$	0.11
Beta(4,4) (optimal by KDE)	$f^*(x) = \frac{35}{32}(1-x^2)^3 I(\lvert x\rvert < 1)$ $Z \sim$ Beta(4,4) on (0,1) $\Rightarrow\ X = 2Z - 1$	1
Truncated N(0,1) (truncated at 0)	$f(x) = 2\phi(x)I(x > 0)$ $Z \sim \mathrm{N}(0,1) \quad \Rightarrow\ X = \lvert Z\rvert$	0.16
Truncated N(0,1) (truncated at -1)	$f(x) = \frac{1}{\Phi(1)}\phi(x)I(x > -1)$ $Z \sim \mathrm{N}(0,1) \quad \Rightarrow\ X = Z$, if $Z > -1$	0.14
Exponential(1) (sharp boundary at 0)	$f(x) = e^{-x}I(x > 0)$	0.17

Table 5.1. Distribution list to compare LOrPE and KDE using oracle method

Throughout section 3.1, all the comparisons of LOrPE and KDE were based on the above seven distributions. MISE is used to compare LOrPE and KDE which is given as follows:

$$MISE(f) = E\left[\int (f - \hat{f})^2 dx\right].$$

The programs *lorpe_study.tcl* and *kde_study.tcl* in NPStat package[22] were used for the simulations. The programs work as follows:

Input : sample size, number of tries(replicates), number of bins, choice of kernel function and degree of the estimate.

- Under each distribution to be estimated, the program generates 1000 samples(number of tries) of random points with sample sizes 10^n where $n = 1, 2, 3, 4, 5.$ are used in simulations. The program sets the support of the sampled distribution. We can truncate it more or less arbitrarily, as long as the density is not expected to be 0 inside the interval. Then while running the simulation we change the number of intervals for discretization(number of bins)and this information is to be used for KDE as the software uses a fast method to implement KDE using Fast Fourier Transformations. This method reduces the computational burden of KDE. Then we use the choice of the kernel to be the symmetric Beta kernel where the software supports more than one kernel function. Then we set the degree(M) of the LOrPE/KDE polynomial to be $0, 1, 2, ..., 19$.

Process:

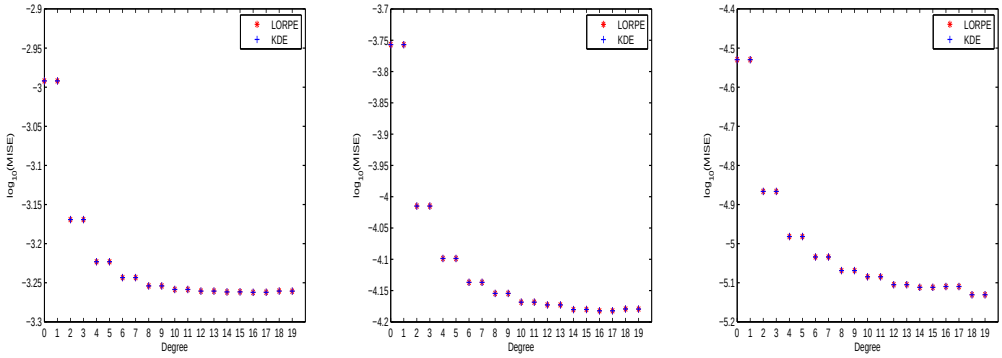
- Using all the parameters that set above, first the program calculates the AMISE optimal h described in Equation 3.1. The program uses it as a starting point and obtain more bandwidths around it. It generates 30 different bandwidth values(h) around (by factor 1.1)the AMISE optimal h in Equation 3.1.
- Then for each pair of h and M on the grid, it calculates MISE. This process has been repeated for both LOrPE and KDE under each distribution listed in Table 1.

Output: MISE values on the grid of h and M.
From the above matrix of MISE, one can pick the minimum MISE and the corresponding h and M. But we further fine tuned the results as follows:

On the grid of h and M, for each M, the minimum MISE is obtained. If this minimum MISE is inside the grid then a third degree polynomial is fitted considering five points around the corresponding h_{min_MISE} and the minimum is obtained from

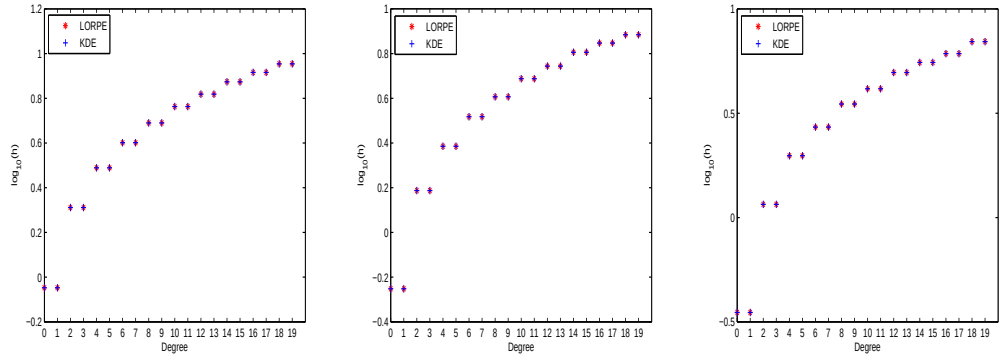
that fit. if the minimum MISE is at the grid boundary then the fit is not performed and the point at which the minimum is found used directly. This is more accurate than the MISE on the original grid of h and M. Finally the minimization over M can be done. The following graphs summarize the results taken for LOrPE and KDE using the above described procedure. For example Figure 1 displays $MISE_{min}$ vs. h_{opt} and Figure 2 displays the connection between M and h_{opt} (which can be used to fine M_{opt}) for reconstructing $N(0,1)$ using LOrPE and KDE. The graphs are only for four distributions in Table 5.1.1. See appendix A for all graphs.

N(0,1)



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

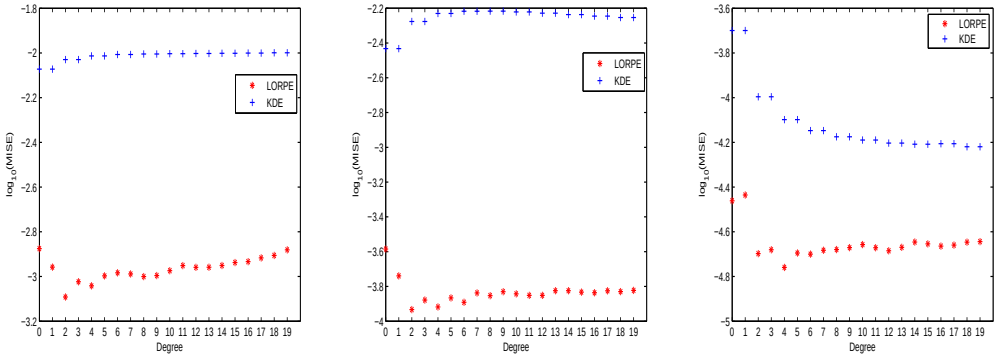
Figure 5.1. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

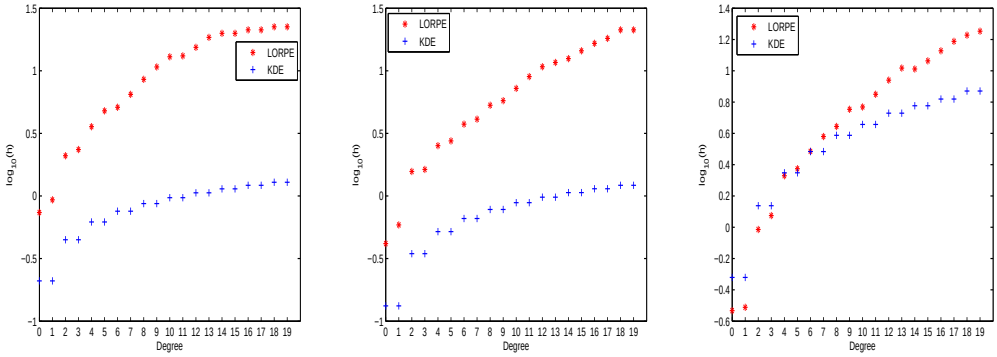
Figure 5.2. $log_{10}(h)$ vs. M

N(0,1) truncated at 0 (KDE without data mirroring)



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

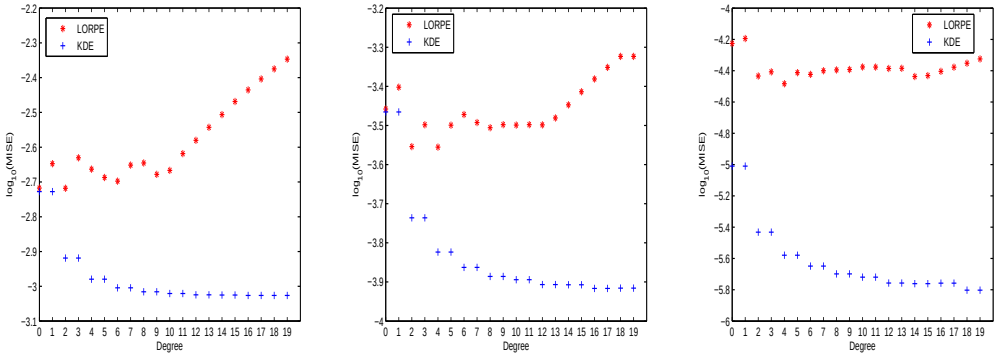
Figure 5.3. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

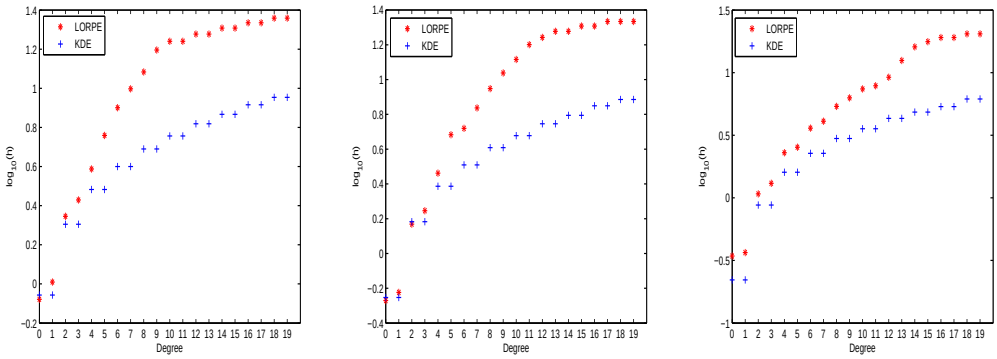
Figure 5.4. $log_{10}(h)$ vs. M

N(0,1) truncated at 0(KDE with data mirroring)



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

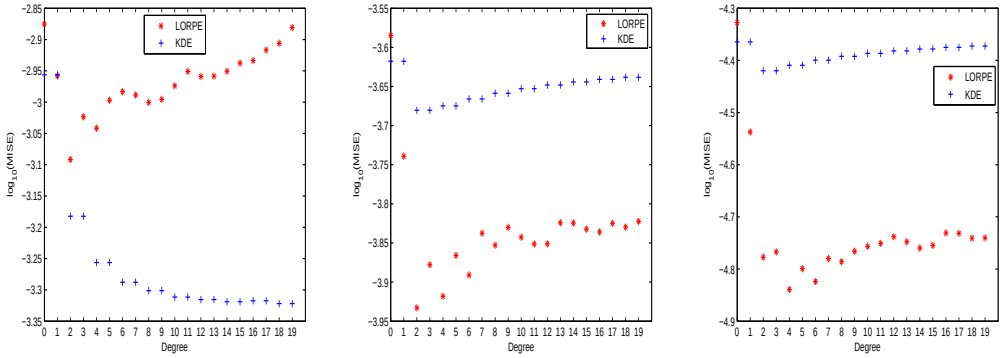
Figure 5.5. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

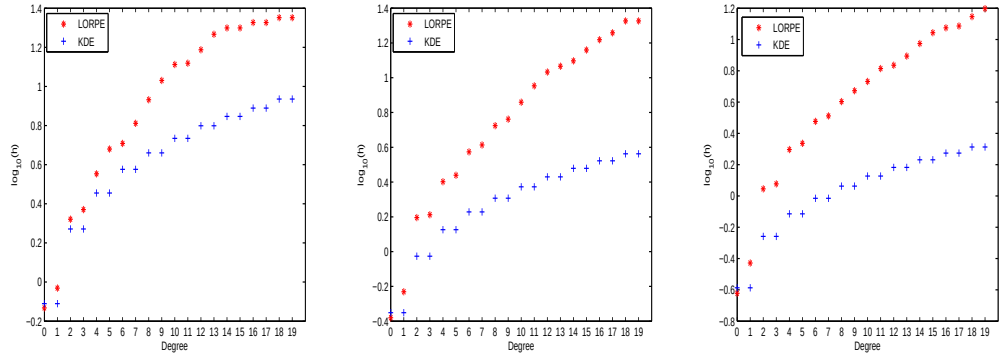
Figure 5.6. $log_{10}(h)$ vs. M

N(0,1) truncated at -1 (KDE with data mirroring)



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

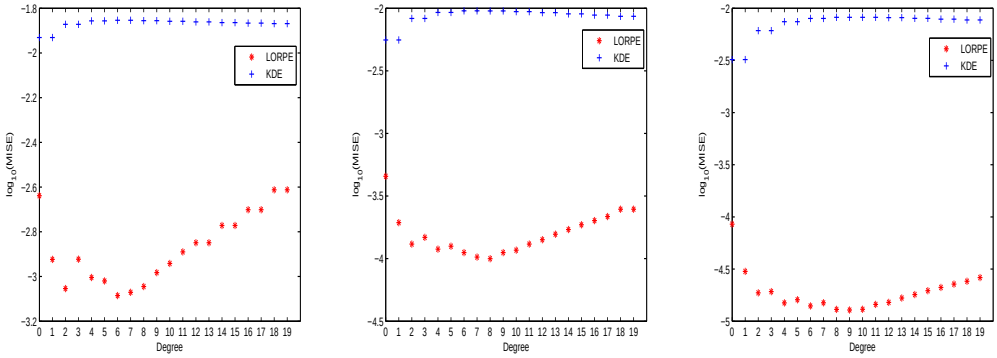
Figure 5.7. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

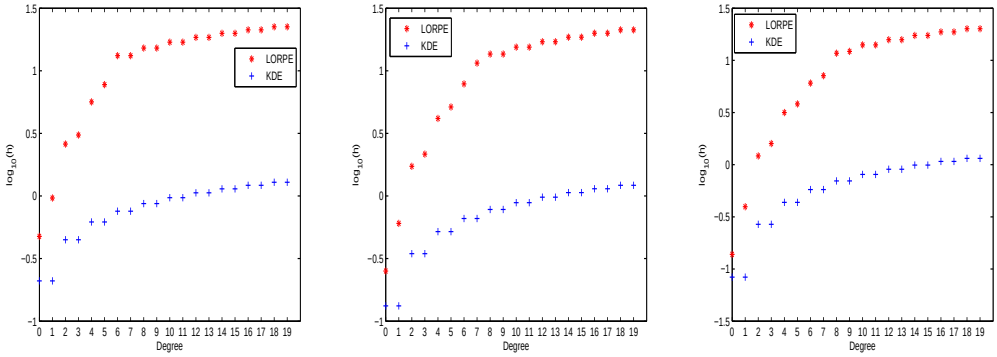
Figure 5.8. $log_{10}(h)$ vs. M

Exponential(1)



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 5.9. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 5.10. $log_{10}(h)$ vs. M

According to the graphical summaries it's clear that LOrPE works exact the same way as KDE when reconstructing Distributions with long tails such as $N(0,1)$. Similar results occurred for Beta$(4,4)$,Normal Mix 1 and Normal Mix 2(see appendix A). When reconstructing distributions with sharp edges LOrPE has lower MISE than KDE. For $N(0,1)$ truncated at 0, KDE works better than LOrPE since KDE has the data mirroring technique which is an ideal method to apply for that particular distribution. But For $N(0,1)$ truncated at -1, LOrPE is better than KDE, with or with out data mirroring for large sample sizes. Reconstructing Exponential(1) is an ideal situation to see how LOrPE wins KDE. Using the graphs it's easy to capture the min_{MISE} and its' corresponding h(h_{opt}) and M(M_{opt}). The following tables contain all h_{opt} , M_{opt},$MISE_{min}$ results summarized for all reconstructed distributions in Table 5.1.1 :

(a) N(0,1)

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00362 (19,11.416)	0.00363 (13,8.317)
1000	0.00055 (16,8.246)	0.00055 (17,8.238)
10000	0.00007 (17,7.030)	0.00007 (17,7.030)
100000	0.00001 (18,6.969)	0.00001 (18,6.969)

(b) Normal Mix 1

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00968 (0,1.011)	0.00968 (0,1.011)
1000	0.00168 (4,1.453)	0.00168 (4,1.453)
10000	0.00022 (11, 2.001)	0.00022 (11, 2.001)
100000	0.00003 (18,2.686)	0.00003 (18,2.686)

Table 5.2.

(a) Normal Mix 2

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.04189 (0,0.251)	0.04263 (2,0.427)
1000	0.00620 (6,0.509)	0.00620 (6,0.509)
10000	0.00078 (14,0.738)	0.00078 (14,0.738)
100000	0.00009 (18,0.778)	0.00009 (18,0.778)

(b) Beta(4,4)

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00896 (15,3.475)	0.00896 (15,3.475)
1000	0.00143 (19,3.667)	0.00143 (19,3.667)
10000	0.00022 (9,1.425)	0.00022 (8,1.425)
100000	0.00003 (6,0.904)	0.00003 (6,0.904)

Table 5.3.

(a) N(0,1) truncated at 0

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00571 (0,1.194)	0.00336 (17,9.654)
1000	0.00101 (2,1.924)	0.00048 (19,8.609)
10000	0.00013 (4,2.655)	0.00006 (18,7.473)
100000	0.00003 (4,2.294)	0.000002 (18,6.156)

(b) N(0,1) truncated at -1

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00598 (2,2.964)	0.00574 (4,3.883)
1000	0.00081 (2,2.099)	0.00246 (1,0.789)
10000	0.00012 (2,1.565)	0.00098 (0,0.185)
100000	0.000010 (4, 1.978)	0.00031 (0,0.039)

Table 5.4.

(a) Exponential(1)

n	LOrPE MISE (M,h)	KDE MISE (M,h)
100	0.00543 (2,4.149)	0.03453 (0,0.478)
1000	0.00082 (6,13.243)	0.01113 (0,0.160)
10000	0.00009 (8,13.654)	0.00406 (0,0.082)
100000	0.00001 (9,12.244)	0.00147 (0,0.034)

Table 5.5.

Based on the above tables we could clearly se that in comparison of oracle methods, LOrPE wins over KDE when reconstructing densities with sharp boundaries. Both methods perform the same way when reconstructing densities with long tails.

5.1.2 Simulations and Comparisons of LOrPE with Other Methods

This section compares LOrPE to KDE, LLDE and OSDE. LOrPE MISE values were calculated using the plugin method using the program *run_lorpe* from NPStat package [22]. For KDE results, the R package *ks* and for LLDE results the R package *locfit* were used. To get the OSDE *run_osde* from NPStat package [22] is used. The support of the density estimate for OSDE is calculated internally. The data point with the smallest coordinate will be mapped into 1/(2 N), where N is the number of points in the sample. The data point with the largest coordinate is mapped into 1 - 1/(2 N). All other points mapped linearly using the mapping determined by these two points. The support of the density estimate is defined by inversely mapping the [0, 1] interval.

once again several distributions taken from table 5.1 and some truncated Student t (on $[-1, 2]$) distributions were reconstructed using the above mentioned methods. The following tables summarizes the MISE values for three different sample sizes. The MISE values in each row has to be multiplied by the multiplier in the corresponding row. The MISE in bold is the minimum value under that particular sample size. According to the tables below, 50% of the times LOrPE has minimum MISE.

Table 5.6. N(0,1) truncated at -1

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	**0.823**	0.949	1.456	1.503	10^{-2}
1000	**0.989**	2.729	6.481	1.587	10^{-3}
10000	**1.336**	10.47	56.54	1.675	10^{-4}

Table 5.7. N(0,1) truncated at 0

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	**0.665**	2.657	3.744	2.160	10^{-2}
1000	**1.194**	9.772	25.48	2.280	10^{-3}
10000	**1.699**	40.58	243.8	2.324	10^{-4}

Table 5.8. N(0,1)

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	0.728	**0.634**	0.656	2.324	10^{-2}
1000	0.817	1.063	**0.662**	22.37	10^{-3}
10000	0.901	1.815	**0.695**	22.28	10^{-4}

Table 5.9. Normal mix 1

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	1.809	**1.294**	1.499	7.120	10^{-2}
1000	7.794	**5.987**	9.373	70.99	10^{-3}
10000	**33.34**	52.73	87.06	709.6	10^{-4}

Table 5.10. Exponential(1)

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	**0.577**	4.231	5.025	21.04	10^{-2}
1000	**1.216**	16.50	41.12	58.50	10^{-3}
10000	**1.818**	69.59	404.9	2096	10^{-4}

Table 5.11. Truncated student t, $df = 1$

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	1.284	0.482	**0.357**	4.494	10^{-2}
1000	1.940	2.021	**0.999**	4.605	10^{-3}
10000	**2.183**	7.613	7.448	4.607	10^{-4}

Table 5.12. Truncated student t, $df = 2$

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	1.048	0.451	**0.431**	3.909	10^{-2}
1000	1.450	0.861	**0.644**	3.941	10^{-3}
10000	**1.889**	2.845	2.567	3.981	10^{-4}

Table 5.13. Truncated student t, $df = 3$

n	LOrPE_plugin	KDE	LLDE	OSDE	Multiplier
100	0.914	**0.470**	0.514	3.658	10^{-2}
1000	1.320	0.868	**0.621**	3.739	10^{-3}
10000	1.701	**1.392**	1.724	3.836	10^{-4}

The above simulations leads to following conclusions:

- When reconstructing truncated Normal distributions and the Exponential distribution, LOrPE plugin method has the smallest MISE values.
- We could see mixed results for Normal mixtures and for truncated t distributions and LOrPE seems to have minimum MISE for larger sample sizes.

5.1.3 Simulations of using NDOF definitions to Select h and M

For LOrPE method it is important to understad how to chose h and M so that the MISE gets smaller than the other commonly used methods, specially when reconstructing densities with sharp boundaries. In the previous chapter we discussed two approaches of doing that. One if to find an AIC like criterion and the other is to use cross-validation. In this section we present the simulations we performed under the first approach which is to use AIC. In AIC formula we are focusing on the effective degrees of freedom(NDOF) part since we need to find if there is a connection between that and h and M.

Following are the simulation work done under each definition of NDOF.$NDOF_1$ is calculated inside the rectangle $(0 < \frac{width}{h} < 10, 0 < M < 10)$ with 50 points along each direction and plot the result. This was to see how NDOF changes when both $\frac{w}{h}$ and M changes and see there exists any visible relation between NDOF, h and M. The following plot is the mesh plot of $NDOF_1$ vs $\frac{w}{h}$ vs M.

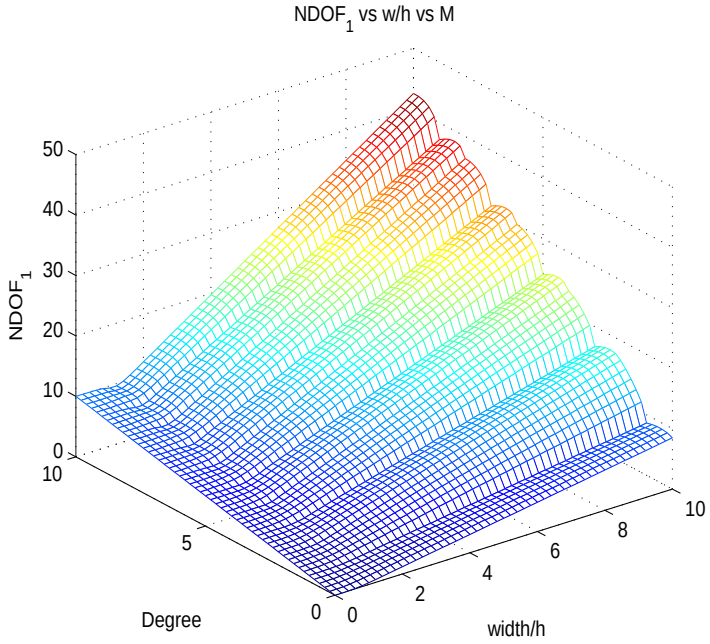


Figure 5.11.

From the plot above we can observe that, when $\frac{width}{h}$ is close to 0, $NDOF_1$ is close to M. On the other hand as M increases the NDOF increases but there are sudden drops when M is an integer.

Similar simulations were performed under $NDOF_2$ as in $NDOF_1$. The following is the resulting plot:

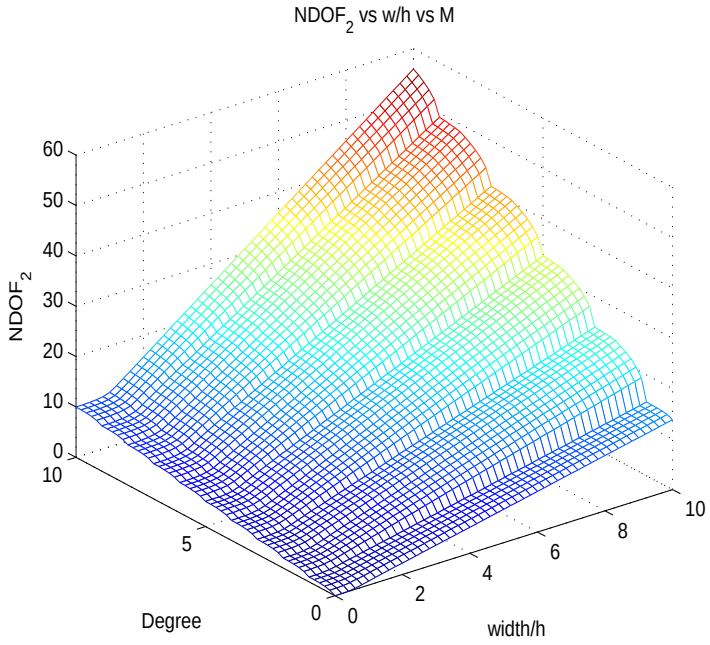


Figure 5.12.

In this case $NDOF_2$ is close to M when M is an integer but, when M is not integer, $NDOF_2$ is larger than M.

The main problems in $NDOF_1$ definition is NDOF drops when M is an integer. The reason for that is the maximum eigen value becomes larger than one when M is an integer. Therefore the denominator of the $NDOF_1$ increases. Therefore $NDOF_1$ decreases. Similar problem occurs with $NDOF_2$ definition
The suggested future work to get rid of this problem is to calculate

$$A^t A = \sum_{i=1}^{n} \lambda_i a_i a_i^t$$

and truncate the expression using $\eta_i = max(1, \lambda_i)$ and get the expression $\sum_{i=1}^{n} \eta_i a_i a_i^t$. Then find a new soothing matrix B which follows the properties of the matrix A $s.t.$ $B^t B = \sum_{i=1}^{n} \eta_i a_i a_i^t$.

5.1.4 LOrPE using Cross-Validation

Using cross-validation techniques described in chapter 2 (LS cross-validation and PL cross-validation). In PL cross-validation, the regularization parameter set set to 0.5. The program *lorpe_study_cv.tcl* from NPStat package [22] is been used for LOrPE cross-validation.

In the following figures we try to compare cross-validation results with LOrPE and KDE oracle results and see if cross-validation can be a method which we can use as a non-oracle method for LOrPE.

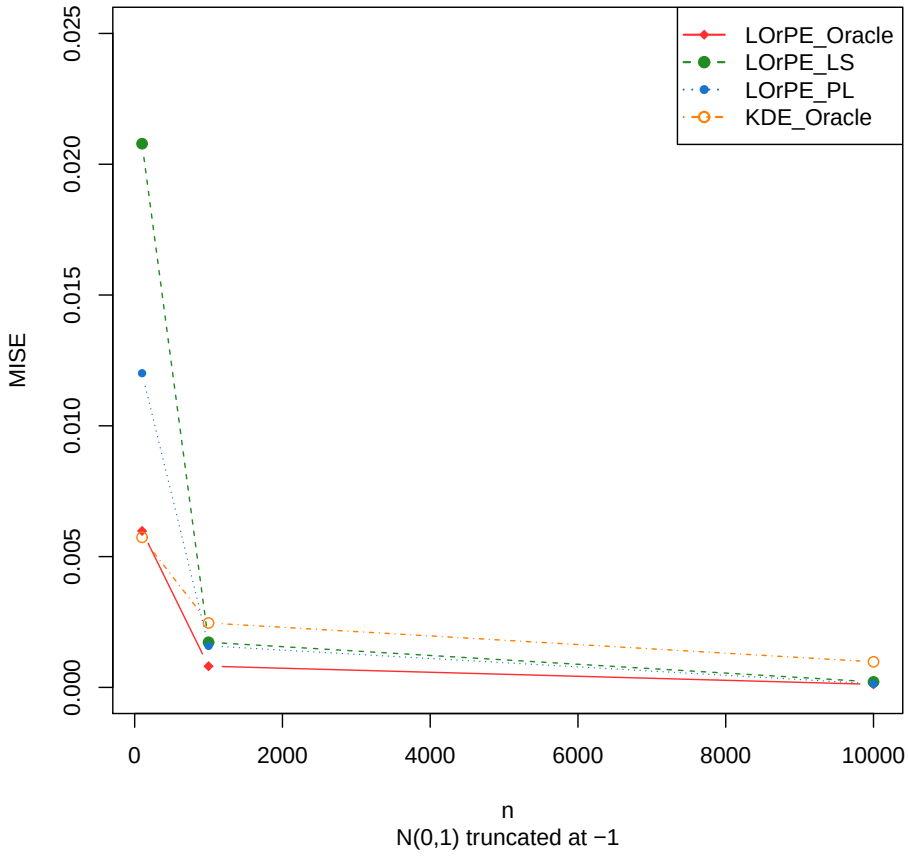


Figure 5.13.

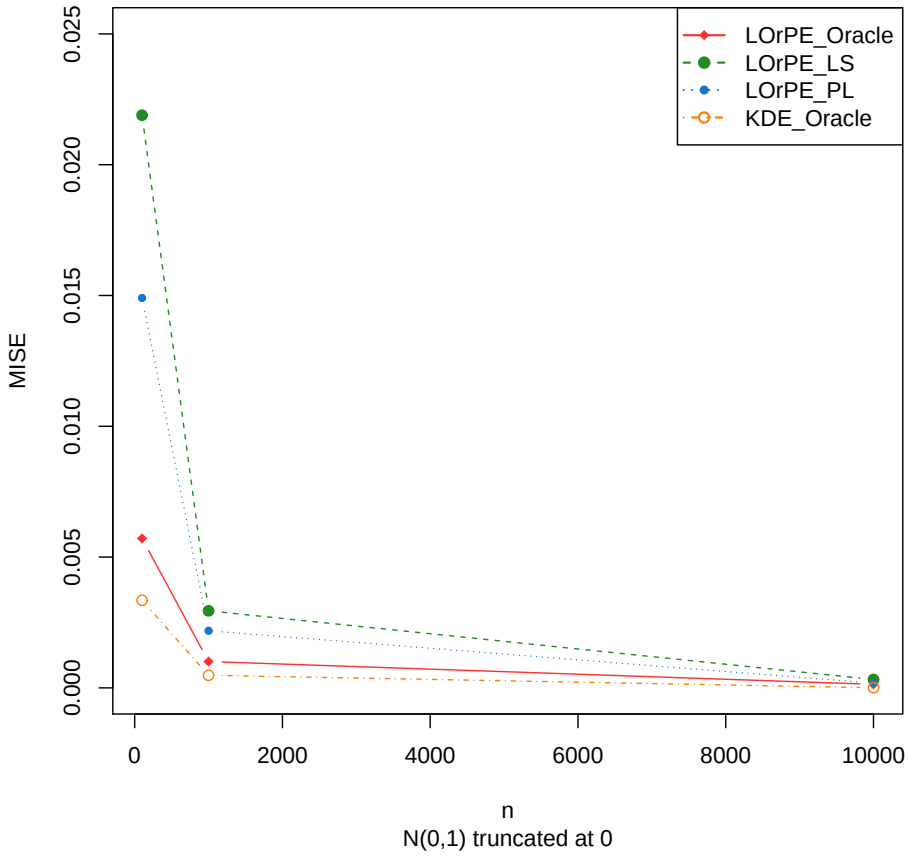

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.000
0.005
0.010
0.015
0.020
0.025
0
2000
4000
6000
8000
10000
n
N(0,1) truncated at 0

Figure 5.14.

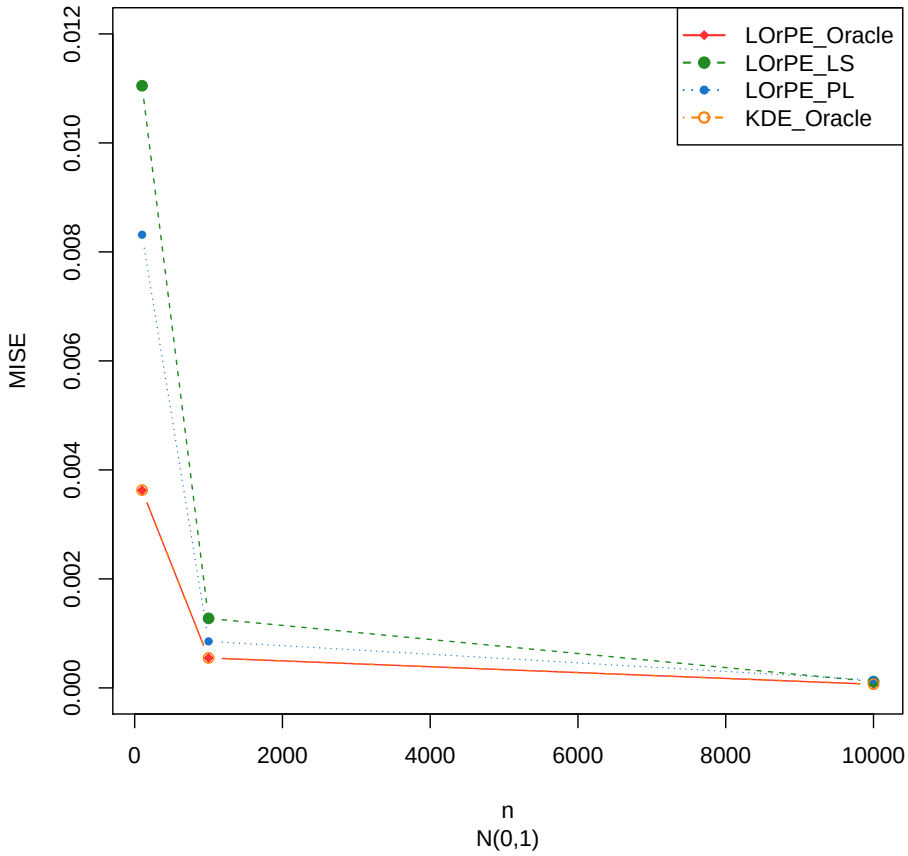

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0
2000
4000
6000
8000
10000
n
N(0,1)

Figure 5.15.

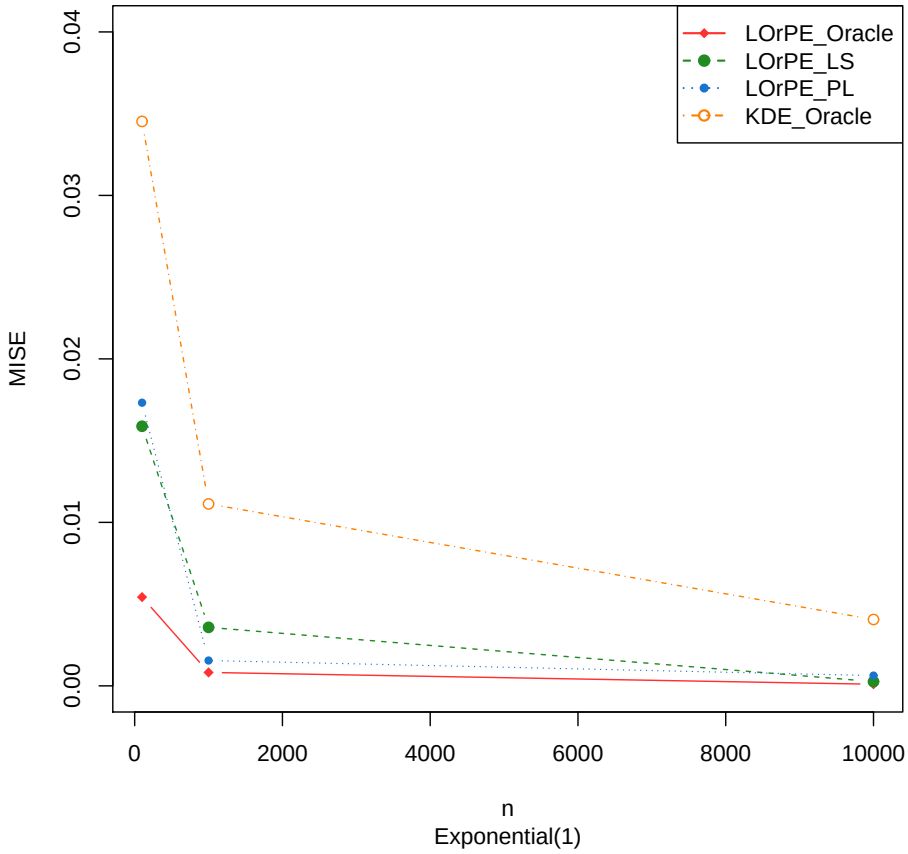

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.00
0.01
0.02
0.03
0.04
0
2000
4000
6000
8000
10000
n
Exponential(1)

Figure 5.16.

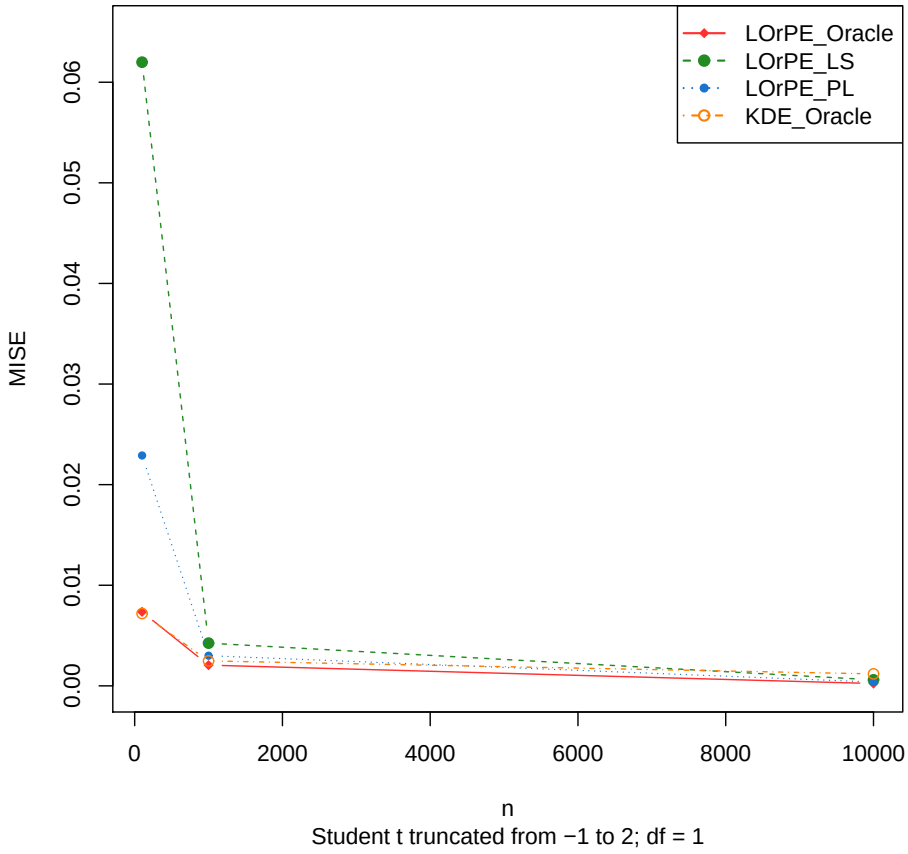

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
2000
4000
6000
8000
10000
n
Student t truncated from −1 to 2; df = 1

Figure 5.17.

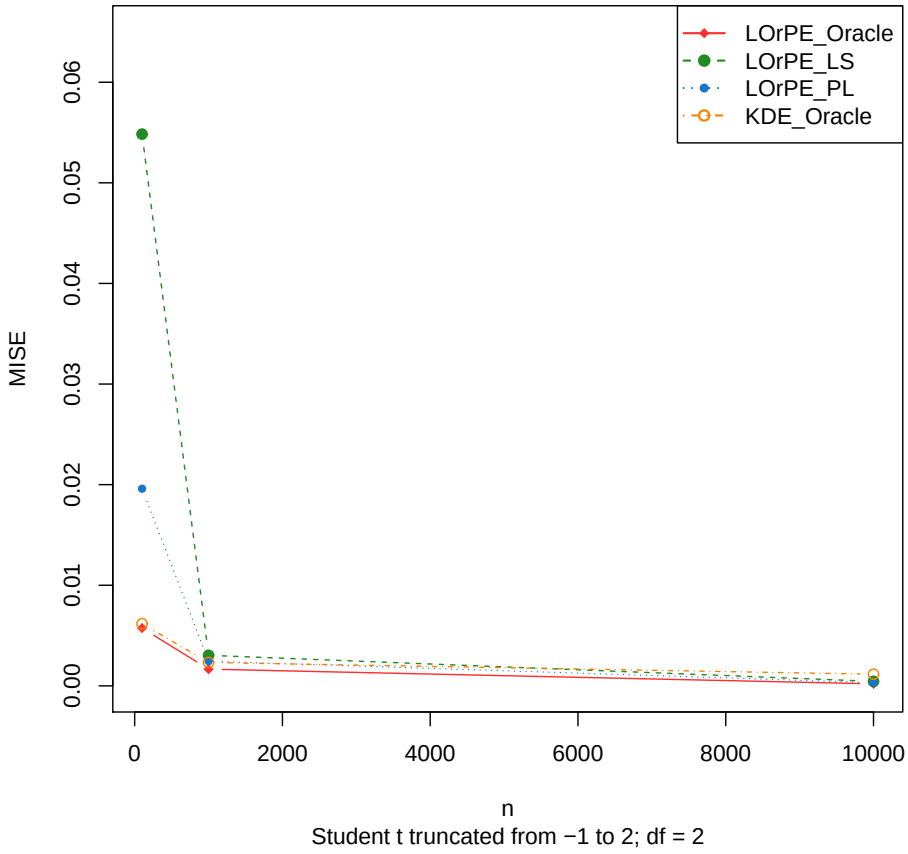

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
2000
4000
6000
8000
10000
n
Student t truncated from −1 to 2; df = 2

Figure 5.18.

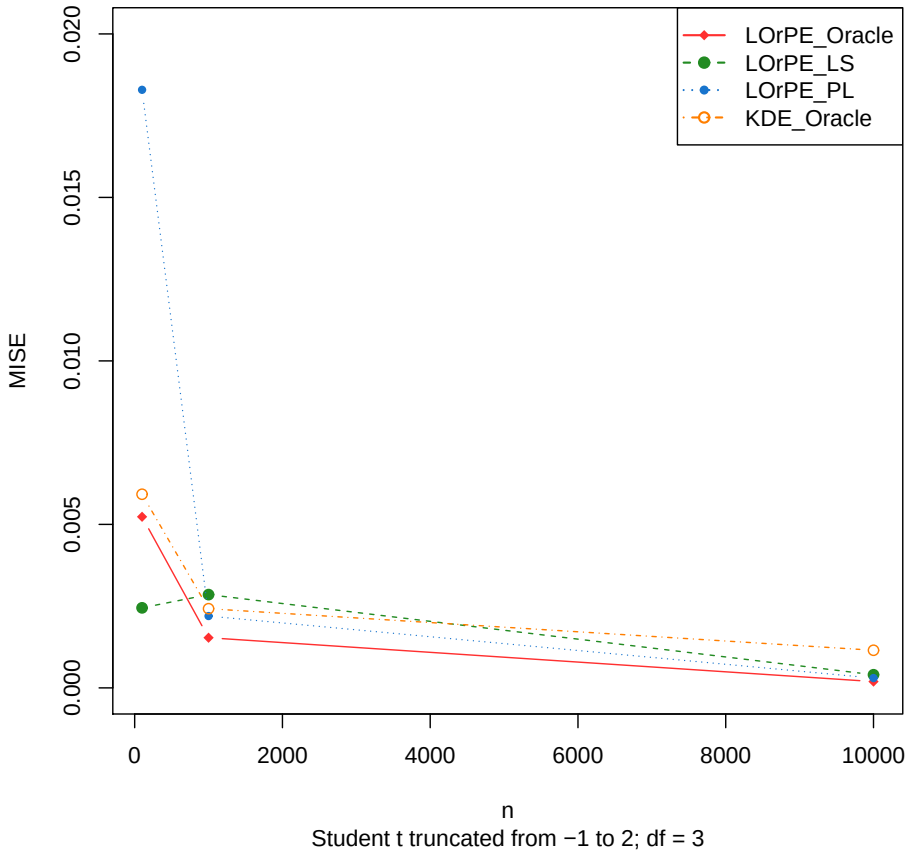

LOrPE_Oracle
LOrPE_LS
LOrPE_PL
KDE_Oracle
MISE
0.000
0.005
0.010
0.015
0.020
0
2000
4000
6000
8000
10000
n
Student t truncated from −1 to 2; df = 3

Figure 5.19.

LOrPE with cross-validation clearly achieves the minimum MISE than KDE oracle for Exponential(1), N(0,1) truncated at -1 and Student t with $df = 1, 2, 3$. Also from the graphs we could see that when the sample size is large cross-validation is getting closer the LOrPE oracle MISE values. In some cases, specially for small sample sizes LOrPE method is not achieving the minimum MISE. One reason for this might be the regularization. $r = 0.5$ might not be the optimal value for regularization. Therefore once again the same calculations were done in the same manner for different r values from 0 to 1 by 0.1. The following panel has the graphical summaries under several distributions:

The plots below contains the MISE at each r . The error bands around each MISE is the difference of 84.13^{th} percentile and 15.87^{th} percentile divide by two times the square root of the sample size which is a robust version of calculating the standard deviation.

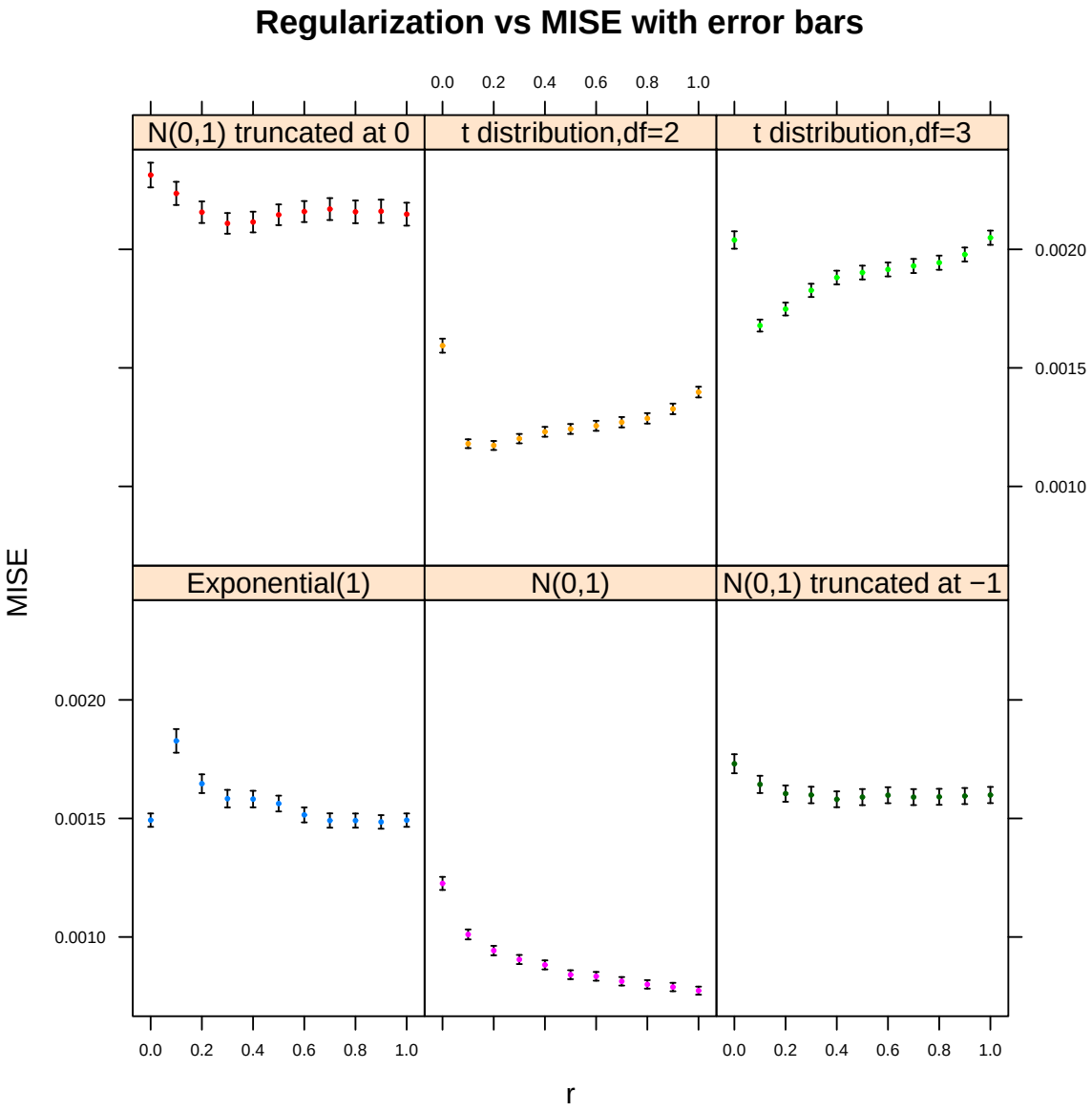


Figure 5.20. n=1000

By the behaviors in the above panel of plots it's clear in some situations choosing $r = 0.5$ might not be the optimal. This is one possible reason in some cases the cross validation results aren't reaching the minimum MISE. In that case it's necessary to search for minimum MISE by optimizing r. With the proper regularization parameter one could use cross-validation for LOrPE method and get better results than KDE when reconstructing densities with sharp edges.

At this point it's possible to make some overall conclusions based on the simulations of LOrPE method as follows: LOrPE is a useful extension of KDE which allows doing the following important things in a systematic way:

- Eliminate boundary bias.
- Build high-order kernels.

Since LOrPE doesn't suffer from boundary bias, it beats KDE when estimating distributions with sharp boundaries. Also LOrPE allows for the taper function and this feature takes LOrPE beyond KDE with high-order kernels. These reasons makes LOrPE applicable in a wider set of problems than KDE. When estimating distributions which decay rapidly at infinity, LOrPE results are exact same as KDE. Also the polynomial modeling could result in a smaller bias than KDE for densities with several (at least M) continuous derivatives, and that a proper balance of h and M can result in a better estimator overall. Cross-validation seems to be a promising way of selecting optimal h and M. For large n, MISE is reaching the oracle MISE. Specially the PL cross-validation results can be improved more with a suitable regularization parameter. LOrPE calculations remain essentially unchanged in multivariate settings: the only difference is switching to multivariate orthogonal polynomial systems.

5.2 Comparison of Several Density Estimates with Real Data

This section focused on comparison of several density estimations using a real data set. The data used in the following figures is the adoption visa data used in [17]. The goal was to compare all density estimation methods discussed in this thesis but due to several problems of using the technology within a time frame the only results included here are for KDE, LLDE and SPA. KDE is estimated using the R package *ks* with Sheather and Jones plugin bandwidth [18], R package *locfit* is used for LLDE and R package *boot* is used for SPA. The following figures illustrates the comparison between the above mentioned methods for adoption data(Note that the x axis is the number of visas):

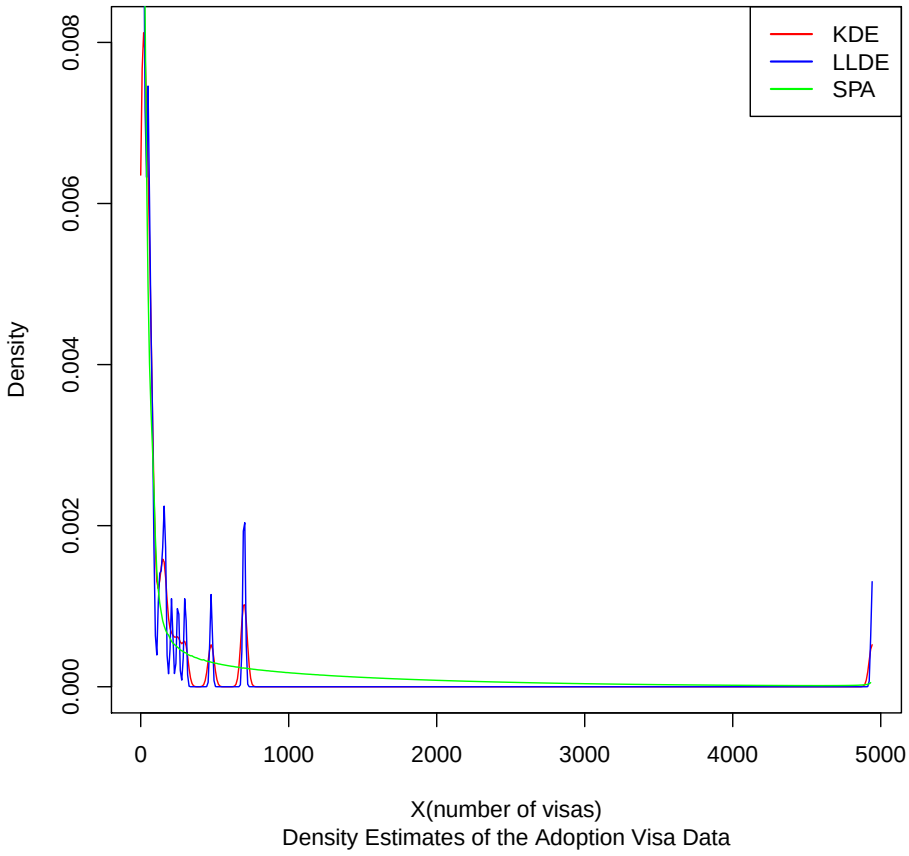


Figure 5.21. Sample 1

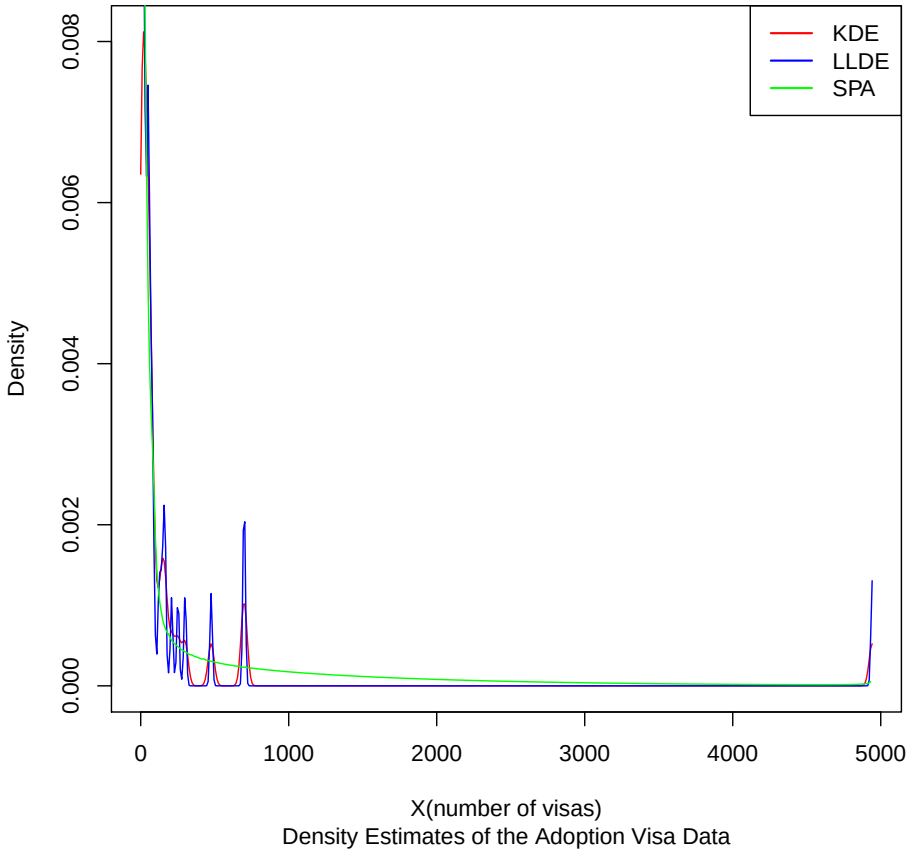


Figure 5.22. Sample 2

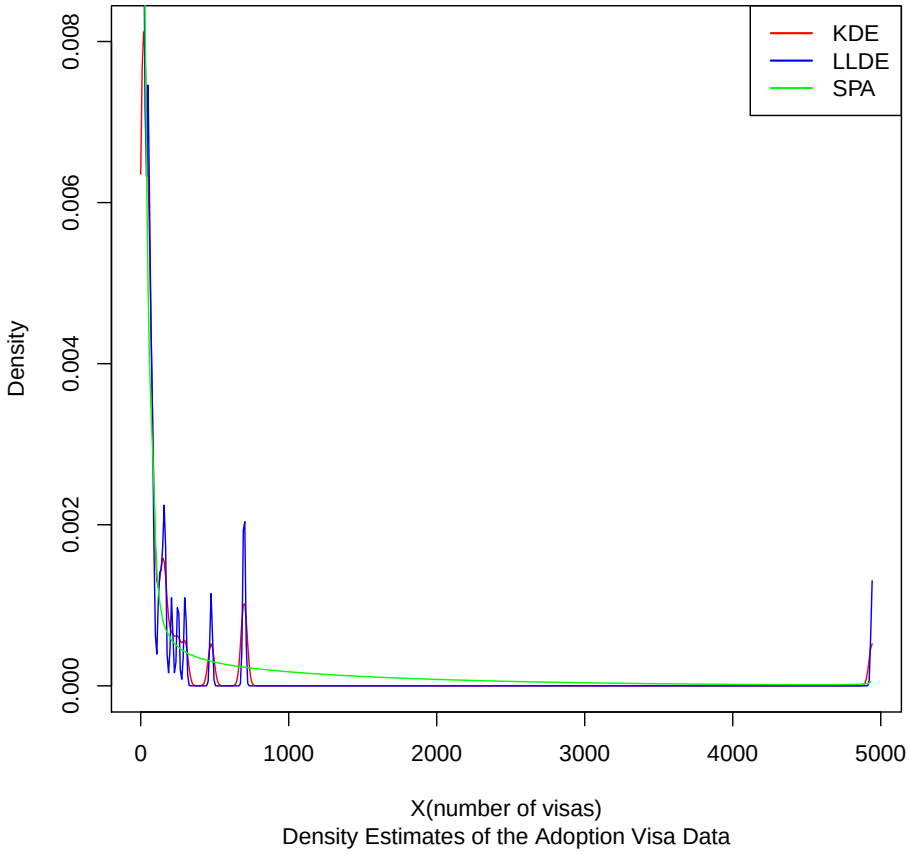


Figure 5.23. Sample 3

When using SPA for the estimation we could see that it over smoothed the density and there is a visible boundary bias near the boundaries. KDE is reasonable inside the density support but it also shows some boundary bias specially near the sharp boundary. LLDE has less bias compared to KDE and SPA but it seemed to over smoothed in side the density support.

5.3 Simulations and Conclusions Under SPA : 95% Confidence Interval for Saddlepoint Density Estimation

The following graphs illustrate the several distributions reconstructed using Saddlepoint density approximation(SPA) and the 95 % confidence bands around SPA:

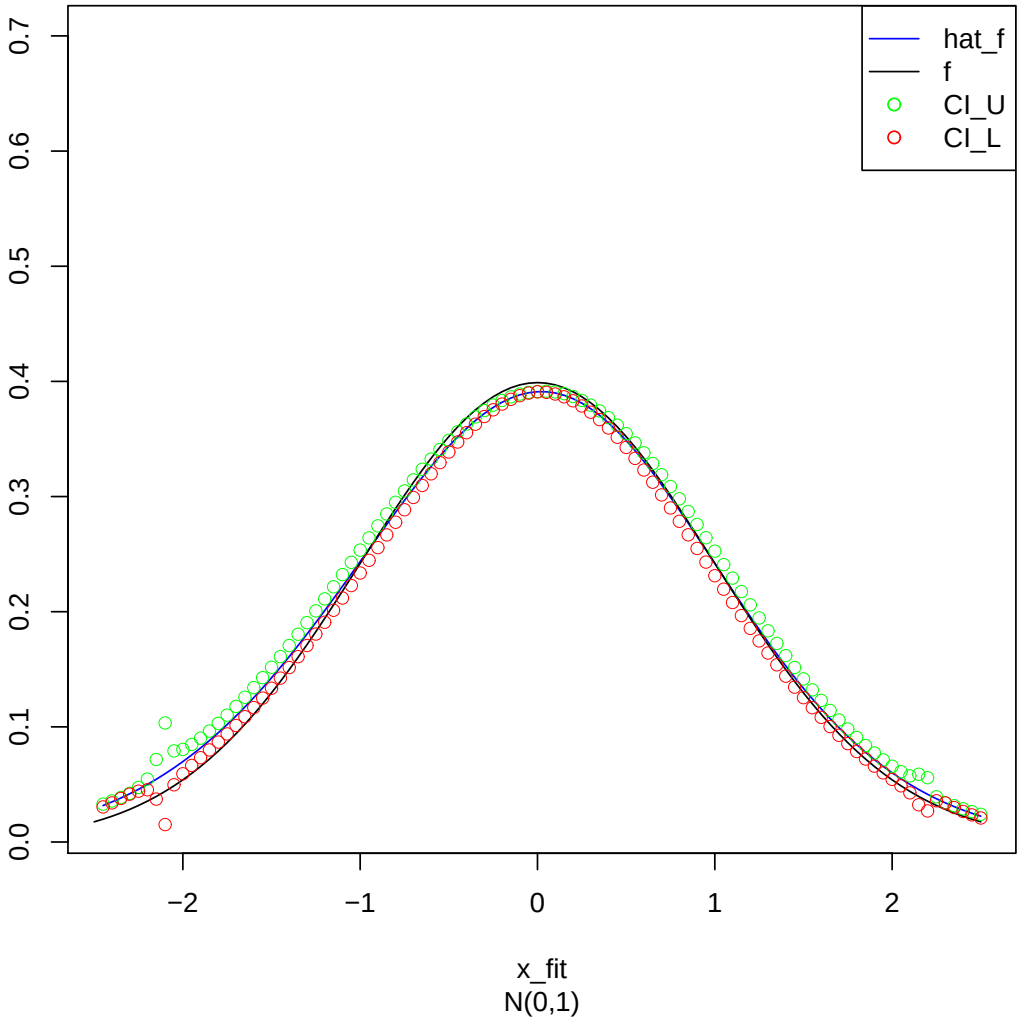


Figure 5.24. n = 1000

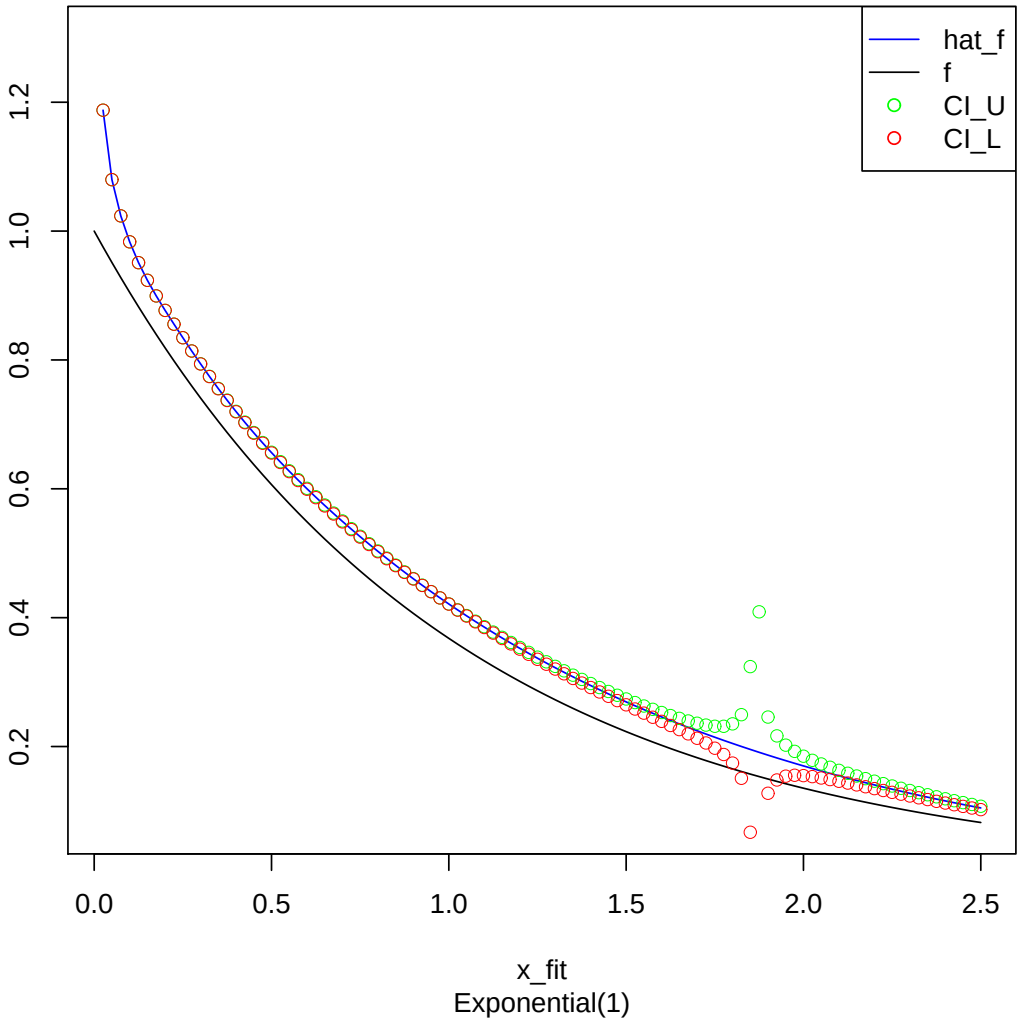


Figure 5.25. n = 1000

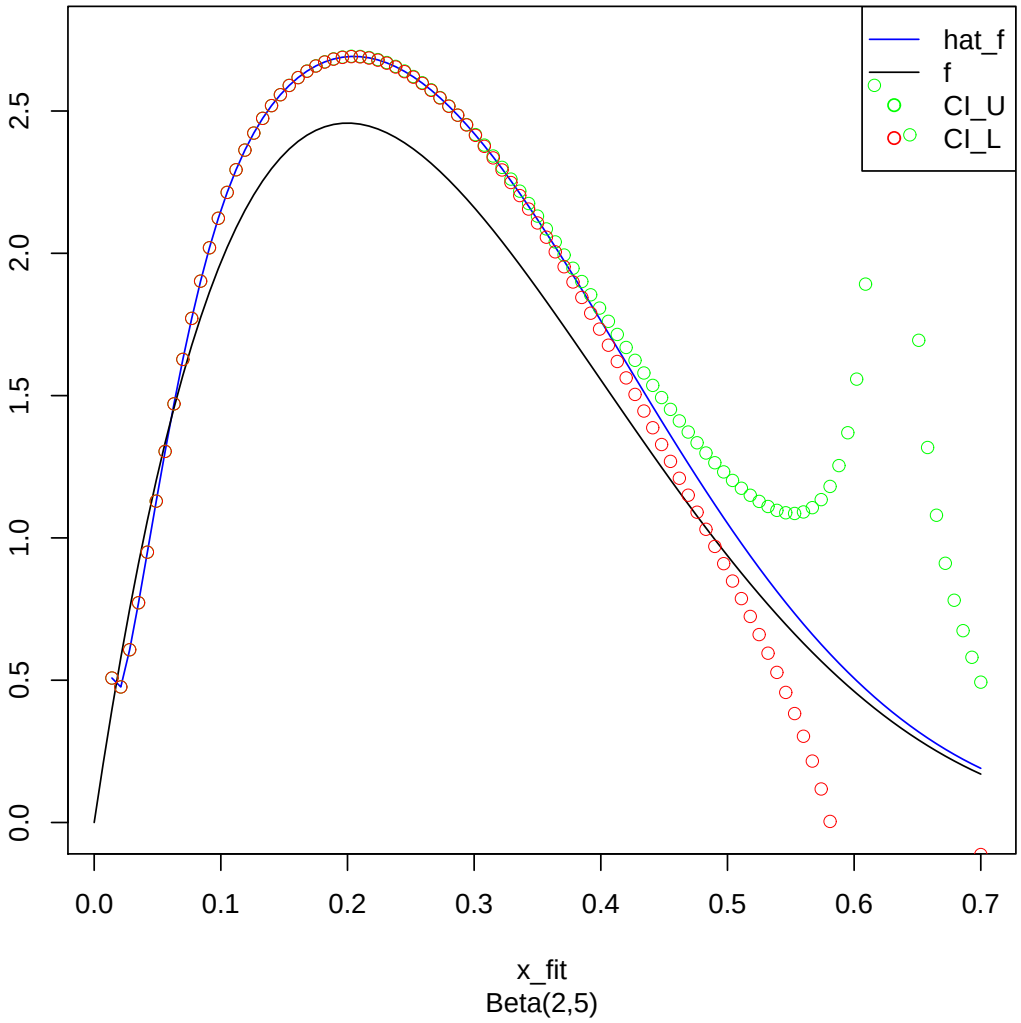


Figure 5.26. n = 1000

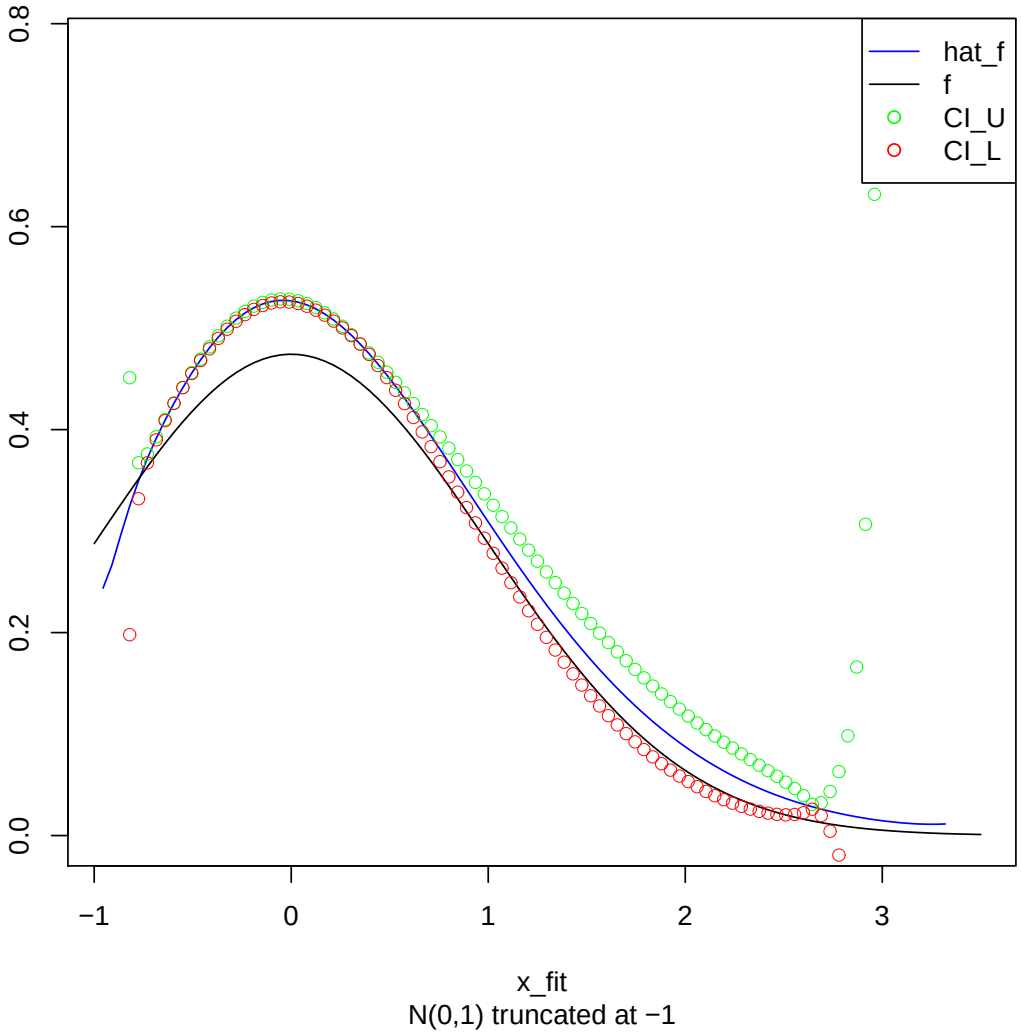


Figure 5.27. n = 1000

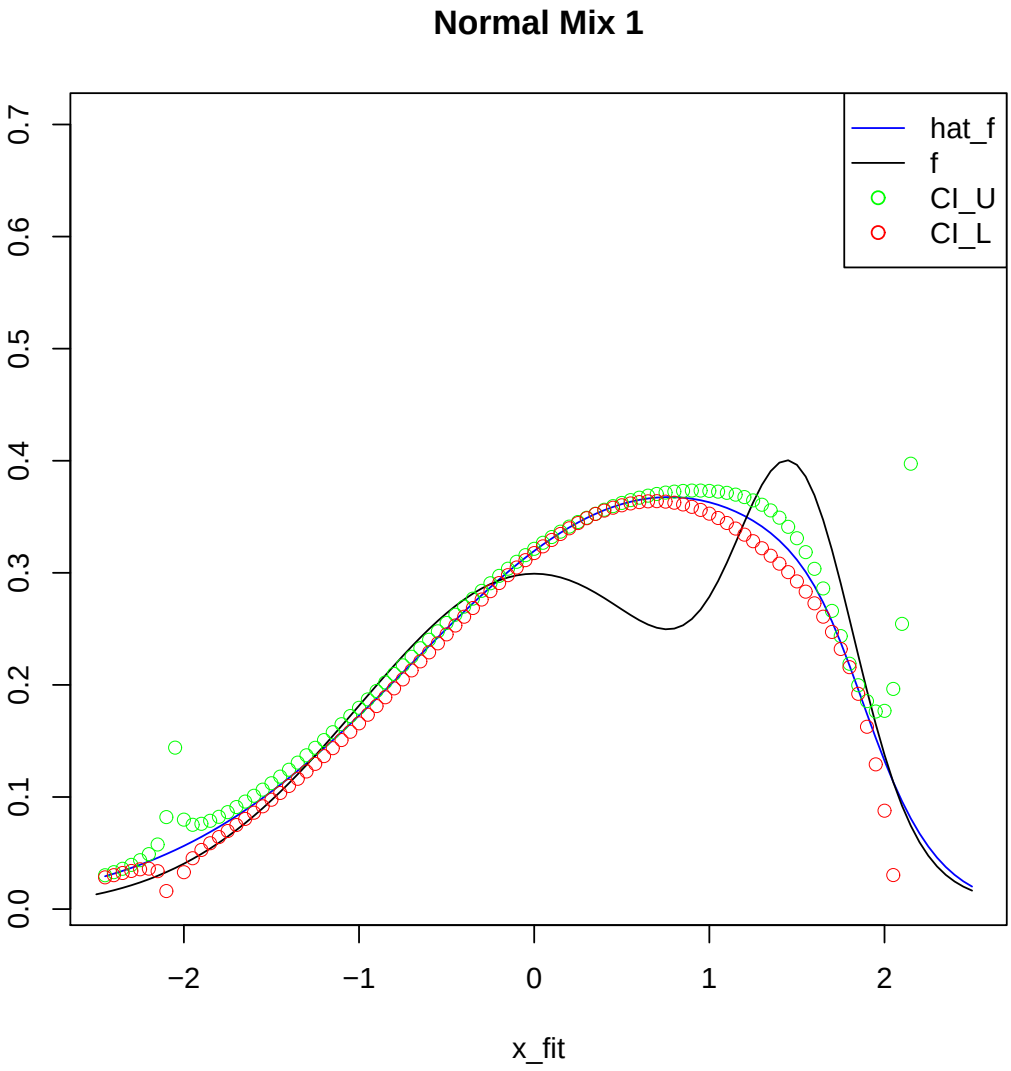


Figure 5.28.

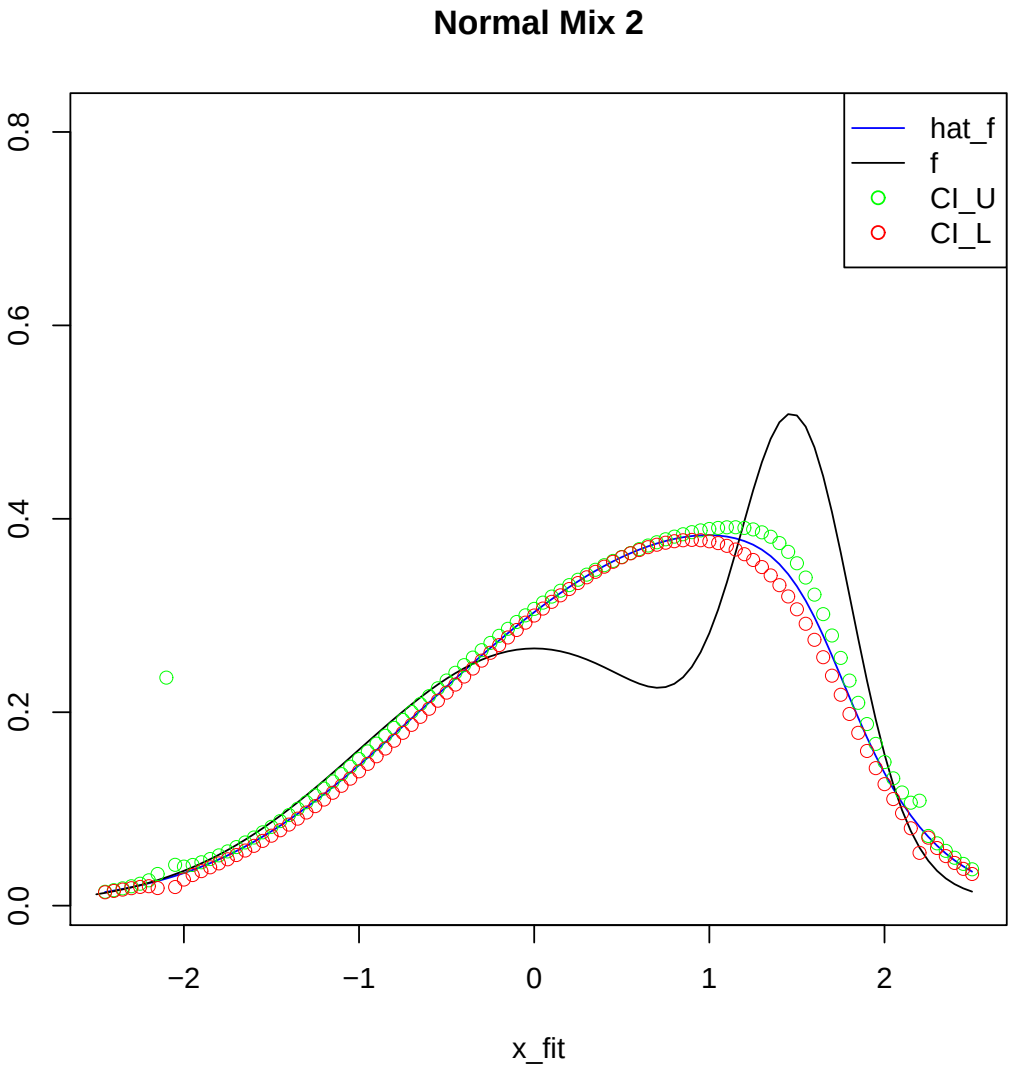


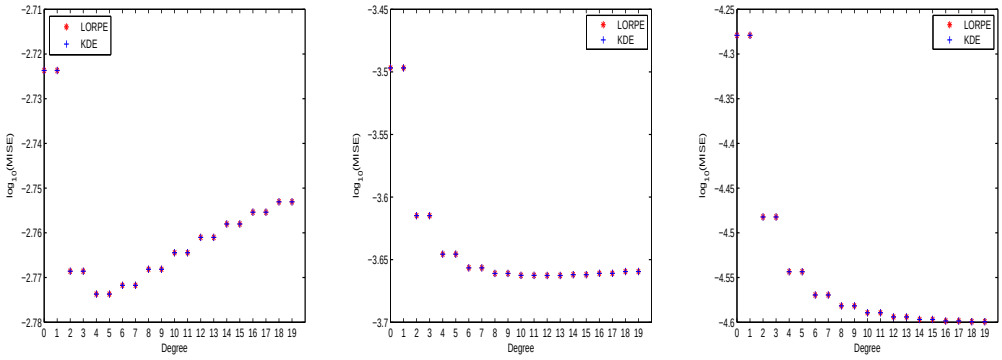
Figure 5.29.

In all the figures we could see that the confidence bands are very tight around the density estimate except at couple of values that it goes to infinity. The reason for this is that the denominator of the variance expression becomes zero for some x values. To overcome this problem we have to consider a second order M-estimator method which involves solving second order equations which could be pretty complicated situation and in this thesis we didn't consider establishing it.

APPENDIX A
GRAPHICAL SUMMERIES OF LORPE VS. KDE

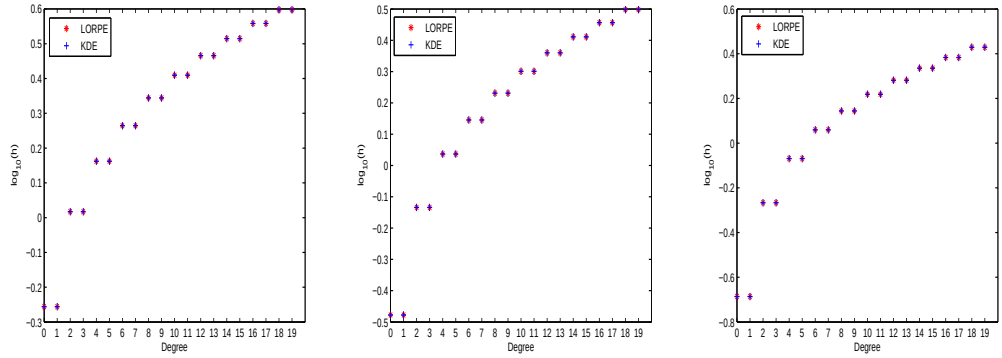
Normal Mix 1

Figure 1.1. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 1.2. $log_{10}(h)$ vs. M

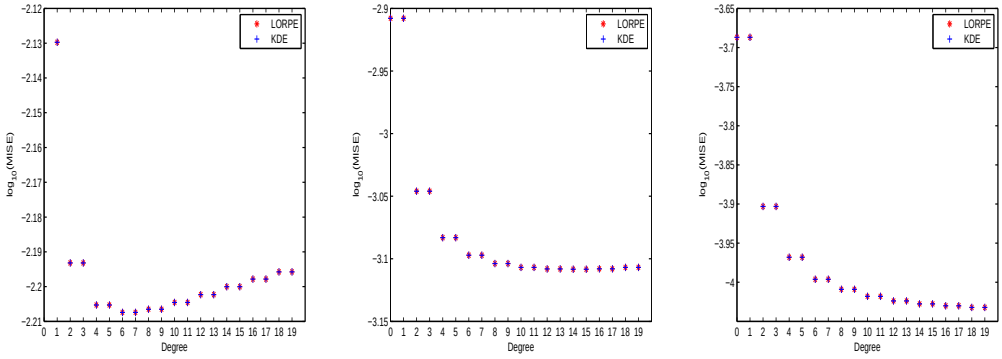


(a) n = 10^3 (b) n = 10^4 (c) n =10^5

- For large n, LOrPE gives exact same results as KDE.
  Minimum MISE would occur : Large M (8 - 12) and smaller h (1.50 - 2.50)

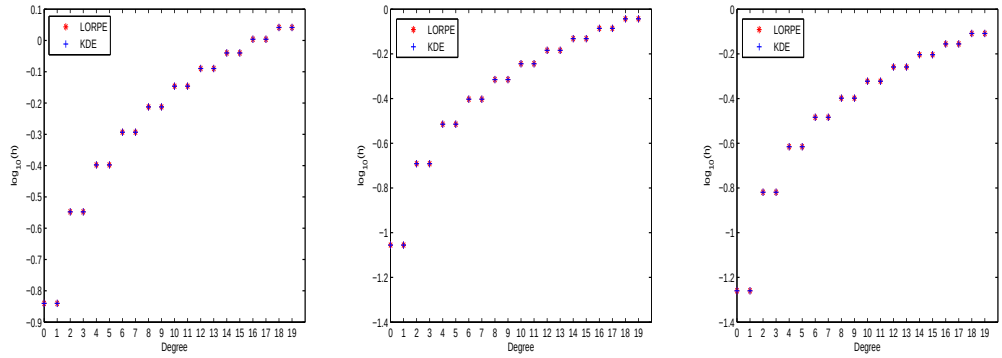
Normal Mix 2

Figure 1.3. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 1.4. $log_{10}(h)$ vs. M

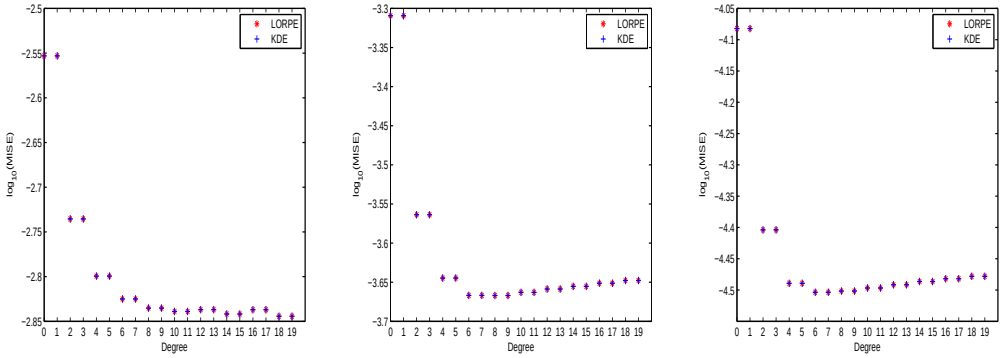


(a) n = 10^3 (b) n = 10^4 (c) n =10^5

- For large n, LOrPE gives exact same results as KDE.
  Minimum MISE would occur : Large M (14 - 18) and smaller h (0.50 - 0.80)

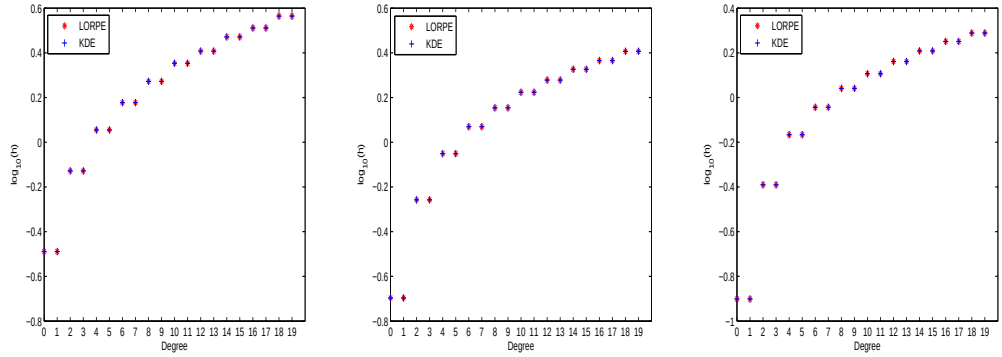
Beta(4,4)

Figure 1.5. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 1.6. $log_{10}(h)$ vs. M

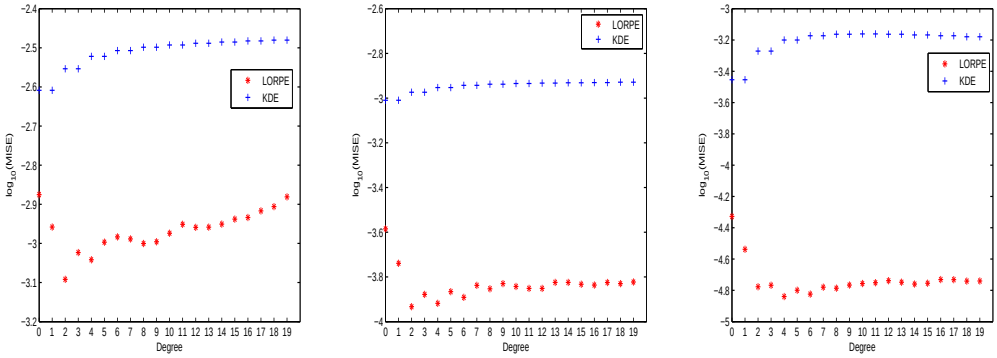


(a) n = 10^3 (b) n = 10^4 (c) n =10^5

- For large n, LOrPE gives exact same results as KDE.
  Minimum MISE would occur : M (5 - 10) and smaller h (0.90 - 1.50)

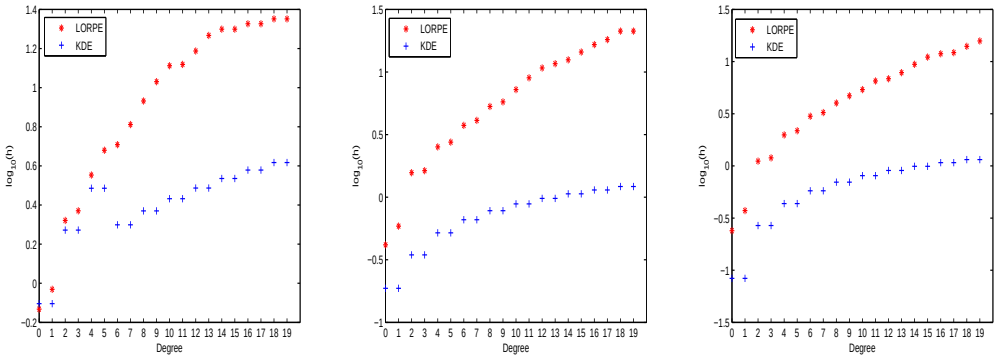
N(0,1) truncated at -1 (KDE without data mirroring)

Figure 1.7. $log_{10}(MISE)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

Figure 1.8. $log_{10}(h)$ vs. M



(a) n = 10^3 (b) n = 10^4 (c) n =10^5

- For large n, LOrPE works better than KDE.

Minimum MISE would occur : Lower M (0 - 2) and smaller h (0.20 - 2.90)

APPENDIX B
CALCULATIONS MADE UNDER CO-VARIANCE AND NDOF3

First I studied about the Legendre polynomial basics basics and then applied the co-variance formula defined above for empirical density estimation to get an idea of how it works. I derived an analytical formulas for $Cov(S_n(j), S_n(k))$, $Cov(\hat{S}_n(j), \hat{S}_n(k))$ and $Cov(S_n(j), \hat{S}_n(k))$ and the definitions of these co-variance are given below as the process goes by.

Legendre Polynomial Basics

- Defined w.r.t. weight function $w(x) = I(-1 < x < 1)$: $P_n(x)$.

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_1(x) = \frac{3}{2}x^2 - \frac{1}{2}, \quad P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x, \quad \dots$$

- Normalize to get: $p_n(x)$.

$$p_0(x) = \frac{1}{\sqrt{2}}, \quad p_1(x) = \sqrt{\frac{3}{2}}x, \quad p_2(x) = \sqrt{\frac{5}{8}}(3x^2 - 1), \quad p_3(x) = \sqrt{\frac{7}{8}}(5x^3 - 3x)$$

- Normalize w.r.t. weight function $w(x) = I(0 < x < 1)$ to get: $L_n(x)$.

$$L_0(x) = 1, \quad L_1(x) = \sqrt{3}(2x - 1), \quad L_2(x) = \sqrt{5}(6x^2 - 6x + 1)$$

$$L_3(x) = \sqrt{7}(20x^3 - 30x^2 + 12x - 1)$$

Gives:

$$\int_0^1 L_j(x)L_k(x)dx = \delta_{jk}.$$

- Recurrence relations:

$$\frac{n+1}{\sqrt{2n+3}}L_{n+1}(x) = \frac{(2n+1)(2x-1)}{\sqrt{2n+1}}L_n(x) - \frac{n}{\sqrt{2n-1}}L_{n-1}(x), \qquad n = 1, 2, \dots$$

- Basic moments: If $U \sim$ iid, then:

$$\mathbb{E}L_j(U) = \int_0^1 L_j(u)L_0(u)du = I(j = 0), \qquad \mathrm{Var}L_j(U) = I(j \neq 0)$$

$$Cov(L_j(U), L_k(U)) = \delta_{jk}$$

Derivation of $Cov(S_n(j), S_n(k))$

If $U_1, \ldots, U_n \sim$ iid , define:

$$S_n(k) = \frac{1}{n}\sum_{i=1}^{n} L_k(U_i).$$

Then we can compute following:

$$\begin{aligned} \mathbb{E}S_n(k) &= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} L_k(u_i)] \\ &= I(k=0) \end{aligned}$$

$$\begin{aligned} \mathrm{Var}S_n(k) &= \frac{1}{n^2}\mathrm{Var}(\sum_{i=1}^{n} L_k(u_i)) \\ &= \frac{1}{n} I(k \neq 0) \end{aligned}$$

$$\begin{aligned} Cov(S_n(j), S_n(k)) &= Cov(\frac{1}{n}\sum_{i=1}^{n} L_j(u_i), \frac{1}{n}\sum_{l=1}^{n} L_k(u_l)) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n} Cov(L_j(u_i), L_k(u_l)) \\ &= \frac{1}{n} I(j=k\neq 0) \;\; ;for\ i=l \end{aligned}$$

By the co-variance $S_n(j), S_n(k)$ are uncorrelated for $j \neq k$ but dependent since both depend on u_i's.

Derivation of $Cov(\hat{S}_n(j), \hat{S}_n(k))$

Assume $X_1, \ldots, X_n \sim$ iid with cdf $F(x)$, pdf $f(x)$, and consider the EDF:

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x) \equiv \hat{F}_x.$$

(a) :

$$\begin{aligned}\mathbb{E}\hat{F}_x &= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x)] \\ &= F(x)\end{aligned}$$

$$\begin{aligned}\mathrm{Var}\hat{F}_x &= \frac{1}{n^2}\mathrm{Var}(\sum_{i=1}^{n} I(X_i \leq x)) \\ &= \frac{1}{n}\mathrm{Var}(I(X \leq x)) \\ &= \frac{1}{n}F(x)(1-F(x))\end{aligned}$$

$$\begin{aligned}Cov(\hat{F}_x, \hat{F}_y) &= \mathbb{E}[\hat{F}_x\hat{F}_y] - F(x)F(y) \\ &= \frac{1}{n^2}[nF(min(x,y)) - n(n-1)F(x)F(y)] - F(x)F(y) \\ &= \frac{1}{n}[F(min(x,y)) - F(x)F(y)]\end{aligned}$$

(b) Define the vectors: $\hat{L}_{jk}(x,y) = [L_j(\hat{F}_x), L_k(\hat{F}_y)]'$
and $L_{jk}(x,y) = [L_j(F(x)), L_k((F(y))]'$ Delta Method :
The recurrence formula for the first derivative of $L_n(x)$ w.r.t. x is

$$L_{n+1}^{'}(x) = \sqrt{\frac{2n+3}{2n+1}}\,[(2n+1)L_n(x) + (2x-1)L_n'(x)] \tag{2.1}$$

Let J be the Jacobian's matrix such that

$$J = \begin{bmatrix} \frac{\partial L_j(F(x))}{\partial F(x)} & \frac{\partial L_j(F(x))}{\partial F(y)} \\ \frac{\partial L_k(F(y))}{\partial F(x)} & \frac{\partial L_k(F(y))}{\partial F(y)} \end{bmatrix}$$

Then

$$J = \begin{bmatrix} L_j^{'}(F(x)) & 0 \\ 0 & L_k^{'}(F(y)) \end{bmatrix}$$

If

$$\begin{bmatrix} \hat{F}_x - F(x) \\ \hat{F}_y - F(y) \end{bmatrix} \xrightarrow{d} N(\underline{0}, \tfrac{\Sigma}{n})$$

Where

$$\Sigma = \begin{bmatrix} F(x)(1-F(x)) & F(min(x,y)) - F(x)F(y) \\ F(min(x,y)) - F(x)F(y) & F(y)(1-F(y)) \end{bmatrix}$$

then

$$\begin{bmatrix} \hat{L}_{jk}(x,y) - L_{jk}(x,y) \end{bmatrix} \xrightarrow{d} N(\underline{0}, J\tfrac{\Sigma}{n}J')$$

Therefore

$$\mathbb{E}\hat{L}_{jk}(x,y) \approx L_{jk}(x,y)$$

and

$$\text{Var}\hat{L}_{jk}(x,y) = \Big[(L_j^{'}(F(x)))^2 F(x)(1-F(x)), (L_k^{'}(F(y)))^2 F(y)(1-F(y))\Big]$$

and

$$Cov(L_j(\hat{F}_x), L_k(\hat{F}_y)) = \frac{1}{n} L_j^{'}(F(x)) L_k^{'}(F(y))[F(min(x,y)) - F(x)F(y)]$$

(c) Define the empirical version of $S_n(k)$, which makes sense since by PIT, $F(X) \sim$ Unif$(0,1)$:

$$\hat{S}_n(k) = \frac{1}{n}\sum_{i=1}^{n} L_k(\hat{F}_x).$$

Now let's compute $Cov(\hat{S}_n(j), \hat{S}_n(k))$

$$\begin{aligned} Cov(\hat{S}_n(j), \hat{S}_n(k)) &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n}(Cov(L_j(\hat{F}_{x_i}), L_k(\hat{F}_{x_l})) \\ &= \frac{1}{n^3}\sum_{i=1}^{n}\sum_{l=1}^{n} L_j^{'}(F(x_i)) L_k^{'}(F(x_l))[F(min(x_i,x_l)) - F(x_i)F(x_l)] \end{aligned}$$

Derivation of $Cov(\hat{S}_n(j), S_n(k))$

Now we need to consider $Cov(\hat{S}_n(j), S_n(k))$ with the assumption $F(X) \sim \text{Unif}(0,1)$

$$Cov(\hat{S}_n(j), S_n(k)) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n} Cov(L_j(\hat{F}_{x_i}), L_k(F(X_l)))$$

where

$$E[\hat{F}_{x_i}] = F(x_i)\ , E[F(X_l)] = \frac{1}{2},$$

$$Var(\hat{F}_{x_i}) = \frac{1}{n}F(x_i)(1-F(x_i))\ , Var(F(X_l)) = \frac{1}{12}$$

$$\begin{aligned} Cov(\hat{F}_{x_i}, F(X_l)) &= E\left[\frac{1}{n}\sum_{t=1}^{n} I(X_t \le x_i)F(X_l)\right] - \frac{1}{n}\sum_{t=1}^{n} E\left(I(X_t \le x_i)\right)E(F(X_l)) \\ &= \frac{1}{n}\sum_{t=1}^{n} E\left[I(X_t \le x_i)F(X_l)\right] - \frac{1}{2n}\sum_{t=1}^{n} F_{X_t}(x_i) \end{aligned}$$

$$Cov(\hat{F}_{x_i}, F(X_l)) = \begin{cases} 0 & \text{if } t \neq l \\ \frac{1}{n}\sum_{t=1}^{n}\left[\int F(X_l)f(x)dx - F_{X_t}(x_i)\right] & \end{cases}$$

Let $Cov(\hat{F}_{x_i}, F(X_l)) = A$

Then by the Delta method
If

$$\sqrt{n}\begin{bmatrix} \hat{F}_{x_i} - F(x_i) \\ F(X_l) - \frac{1}{2} \end{bmatrix} \xrightarrow{d} N(\underline{0}, \Sigma)$$

Where

$$\Sigma = \begin{bmatrix} \frac{F(x_i)(1-F(x_i))}{n} & A \\ A & \frac{1}{12} \end{bmatrix}$$

then

$$\sqrt{n}\begin{bmatrix} L_j(\hat{F}_{x_i}) - L_j(F_{x_i}) \\ L_k(F(X_l)) - L_k(\frac{1}{12}) \end{bmatrix} \xrightarrow{d} N(\underline{0}, J\Sigma J')$$

Here J is the Jacobian matrix where

$$J = \begin{bmatrix} \frac{\partial L_j(\hat{F}_{x_i})}{\partial \hat{F}_{x_i}} & 0 \\ 0 & \frac{\partial L_k(F(X_l))}{\partial F(X_l)} \end{bmatrix}$$

Therefore

$$Cov(\hat{S}_n(j), S_n(k)) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n} L_j'(\hat{F}_{x_i})AL_k'(F(X_l))$$

APPENDIX C
R CODE OF SPA PLOTS

```
This program generates plots of Saddle-point density estimation
and its 95% confidence intervals
library(boot)
library(truncnorm)
library(nor1mix)
dist:
0- expo(1)
1-N(0,1)
2-truncated N(0,1)
3-beta(4,4)
4-Normalmix1
5-Normalmix2

dist = 0
N = 500  number of intervals
n = 1000  sample size

generate data based on the given distribution:
if(dist == 0)
{
x_min =0
x_max =2.5
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
x = rexp(n,1)
ttt = dexp(tt)
}
if(dist == 1)
{
x_min =-2.5
x_max =2.5
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
x = rnorm(n,0,1)
ttt = dnorm(tt)
}
```

```
if(dist == 2)
{
x_min =0
x_max =2.2
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
x = rtruncnorm(n, a=x_min, b=x_max, mean = 0, sd = 1)
ttt = dtruncnorm(tt, a=x_min, b=x_max, mean = 0, sd = 1)
}
if(dist == 3)
{
x_min =0
x_max =0.7
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
x = rbeta(n, 2, 5)
ttt = dbeta(tt,2,5)
}
if(dist == 4)
{
x_min =-2.5
x_max =2.5
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
r <- norMix(mu=c(0,3/2), sig2=c(1,1/9), w=c(.75,.25))
x = rnorMix(n,r)
ttt = dnorMix(tt,obj=r)
}
if(dist == 5)
{
x_min =-2.5
x_max =2.5
h = (x_max - x_min)/N
tt = seq(x_min,x_max,by = h)
r <- norMix(mu=c(0,1.5), sig2=c(1,1/10), w=c(0.6666667,0.3333333))
x = rnorMix(n,r)
ttt = dnorMix(tt,obj=r)
}
```

```
x_fit = mat.or.vec(N,1)
hat_f = mat.or.vec(N,1)
n_hat_f = mat.or.vec(N,1)
hat_s = mat.or.vec(N,1)

Grid
for(i in 1:N)
{
x_fit[i] = x_min + i*h
}



Saddle-point density estimation using the function 'saddle':
for(i in 1:N)
{
K.adj = function(s) {
log((sum(exp(x*s)))/n)- t*s
}
K2 = function(s)
{
p = exp(x*s)
p1 = x*p
p2 = (x^2)*p
(sum(p)*sum(p2)- (sum(p1))^2 )/(sum(p))^2
}

t = x_fit[i]
SS = saddle(K.adj=K.adj,K2=K2)
hat_s[i] = SS$zeta.hat[1]
hat_f[i] = SS$spa[1]
}

for(i in 1:N)
{
n_hat_f[i] = hat_f[i]
}


Calculation of the asymptotic variance of the saddlepoint density function:
```

```
K = mat.or.vec(1,N)
K_dprime = mat.or.vec(1,N)
K_ddprime = mat.or.vec(1,N)
K_prime = mat.or.vec(1,N)
sq_deriv = mat.or.vec(1,N)
M = mat.or.vec(1,N)
M_prime= mat.or.vec(1,N)
M_dprime= mat.or.vec(1,N)
M_ddprime= mat.or.vec(1,N)
M_2s_prime= mat.or.vec(1,N)
M_2s_dprime= mat.or.vec(1,N)
sigma_hat_s =mat.or.vec(1,N)
#W =mat.or.vec(1,N)
sigma_f= mat.or.vec(1,N)
CI_L= mat.or.vec(1,N)
CI_U= mat.or.vec(1,N)

for(j in 1:N)
{
M[j] = (1/n)*sum(exp(x_fit*hat_s[j]))
M_prime[j] = (1/n)*sum(x_fit*exp(x_fit*hat_s[j]))
M_dprime[j] = (1/n)*sum(((x_fit)^2) *exp(x_fit*hat_s[j]))
M_ddprime[j] = (1/n)*sum(((x_fit)^3) *exp(x_fit*hat_s[j]))
M_2s_prime[j] =(1/n)*sum(2*x_fit*exp(x_fit*2*hat_s[j]))
M_2s_dprime[j] = (1/n)*sum(((2*x_fit)^2)*exp(x_fit*2*hat_s[j]))
}

K=log(M)
K_prime= (M_prime/M)
K_dprime=(M*M_dprime -(M_prime^2))/(M^2)
K_ddprime= (1/(M^3))*((M^2*M_ddprime)
                -(3*M_dprime*M_prime*M)+2*(M_prime^3))
sigma_hat_s = (1/n)*((M_2s_dprime-x_fit*M_2s_prime)
                          /(M_dprime-x_fit*M_prime))^2
sq_deriv = (-n_hat_f*(hat_s*K_dprime
                  + ((0.5*K_ddprime)/K_dprime)))^2
sigma_f = sqrt(sigma_hat_s*sq_deriv)

95% confedence intervals of the saddlepoint density
function:
```

```
CI_L = n_hat_f-(1.96*sigma_f/sqrt(N))
CI_U = n_hat_f+(1.96*sigma_f/sqrt(N))
CI_L =  CI_L[!is.na(CI_L)]
CI_U =CI_U[!is.na(CI_U)]


Plots:

colors = c('blue','black','green','red')
plot(x_fit,hat_f,col=colors[1],type='l',ylim=c(min(ttt)
,(max(ttt)+0.3)),ylab="")
lines(tt,ttt,col=colors[2])
if(length(x_fit[-c(which(n_hat_f %in% NA))]) == 0)
{
lines(x_fit,CI_U,col=colors[3],lwd=0.5)
lines(x_fit,CI_L,col=colors[4],lwd=0.5)
}
if(length(x_fit[-c(which(n_hat_f %in% NA))]) > 0)
{
lines(x_fit[-c(which(n_hat_f %in% NA))],CI_U,col=colors[3],lwd=0.5)
lines(x_fit[-c(which(n_hat_f %in% NA))],CI_L,col=colors[4],lwd=0.5)
}
legend(0.1,0.5,
c("hat_f","f","CI_U","CI_L"),
lty=c(1,1),
lwd=c(1.2,2),col=c("blue","black","green","red"),cex=1)
title("Beta(2,5)")
```

APPENDIX D
R CODE FOR KDE AND LLDE MISE CALCULATIONS USING PLUG-IN METHODS

```
library(nor1mix)
library(betareg)
library(AICcmodavg)
library(Formula)
library(locfit)
library(truncnorm)
library(caTools)
library(ks)

#density support :
set.seed(12)
x_min <- 0
x_max <- 1
n <- 1000 # number of points in a sample
j <- 2048 # number of replicates
d = 0 # degree for LLDE
noint <- 1000 # number of sub intervals (x_min, x_max)
is divided in to this number of intervals
dist <- 5
df = 1
dist:
0 - N(0,1)
1 - N(0,1) truncated at 0
2 - N(0,1) truncated at -1
3 - Normalmix 1
4 - Normalmix2
5 - Beta(4,4)
6 - Exponential(1)
7 - Student t with given df.
x <- mat.or.vec(noint,1)
#generating random data
random_data <- array(0,dim=c(n,j))
k <- 1
binwidth = (x_max - x_min)/noint
for(i in 1:(noint-1))
```

```
{
x[[i+1]]= x_min + i*binwidth
}
x[1]<- x_min

f <- array(0,dim=c(length(x),1))

if(dist == 0)
{
f = dnorm(x)
for(k in 1:j)
{
random_data[,k] <- rnorm(n, mean = 0, sd = 1)
}
}
if(dist == 1)
{

f = dtruncnorm(x,a=x_min,b=x_max,mean=0,sd=1)
for(k in 1:j)
{
random_data[,k] <- rtruncnorm(n, a=x_min, b=x_max, mean = 0, sd = 1)
}
}
if(dist == 2)
{

f = dtruncnorm(x,a=x_min,b=x_max,mean=0,sd=1)
for(k in 1:j)
{
random_data[,k] <- rtruncnorm(n, a=x_min, b=x_max, mean = 0, sd = 1)
}
}

if(dist == 3)
{
h <- array(0,dim=c(length(x),1))
g <- array(0,dim=c(length(x),1))
h <- (1/sqrt(2*pi))*exp((-x^2)/2)
g <- ((1/sqrt(2*pi*(1/9)))*exp(-((1.5-x)/(2*(1/3)))^2))
```

```
f <- 0.75*h+0.25*g
for(k in 1:j)
{
r <- norMix(mu=c(0,3/2), sig2=c(1,1/9), w=c(.75,.25))
random_data[,k] <- rnorMix(n,r)
}
}

if(dist == 4 )
{
h <- array(0,dim=c(length(x),1))
g <- array(0,dim=c(length(x),1))
h <- (1/sqrt(2*pi))*exp((-x^2)/2)
g <- ((1/sqrt(2*pi*(1/100)))*exp(-0.5*((1.5-x)/((1/10)))^2))
f <- (2/3)*h + (1/3)*g
for(k in 1:j)
{
r <- norMix(mu=c(0,1.5), sig2=c(1,1/10), w=c(0.6666667,0.3333333))
random_data[,k] <- rnorMix(n,r)
}
}
if(dist == 5)
{
f <- dbeta(x,4,4)
for(k in 1:j)
{
random_data[,k] <- rbeta(n, 4, 4)
}
}
if(dist == 6)
{

f = dexp(x)
for(k in 1:j)
{
random_data[,k] <- rexp(n,1)
}
}

if(dist == 7)
```

```
{

f = dt(x,df)
for(k in 1:j)
{
random_data[,k] <- rt(n,df)
}
}

TR_KDE = array(0,dim=c(1,j))
h_mid = array(0,dim=c(1,j))
for(k in 1:j)
{
#h_mid = hpi(random_data,deriv.order = 0)
h_mid[k] = dpik(random_data[,k],scalest = "stdev",kernel = "normal")
fit= kde(random_data[,k],h=h_mid[k],xmin=x_min,xmax=x_max
,eval.points=x)
sq_f = (fit$estimate - f)^2
TR_KDE[k] = trapz(x, sq_f)
}
MISE_KDE = sum(TR_KDE)/j
# apply locfit and get f_hat_LLDE
f_hat_LLDE <- array(0,dim=c(1,length(x)))
TR_LLDE = array(0,dim=c(1,j))
for(i in 1:j)
{
fit <- locfit(~lp(random_data[,i],deg = d),ev=lfgrid(mg=1000,ll = x_min, ur = x_ma
f_hat_LLDE <- predict(fit,x)
sq_f = (f_hat_LLDE - f)^2
TR_LLDE[i] = trapz(x, sq_f)
}
MISE_LLDE = sum(TR_LLDE)/j
print(MISE_LLDE)
print(MISE_KDE)
print(dist)
```

BIBLIOGRAPHY

[1] Ronald W Butler. *Saddlepoint approximations with applications*. Cambridge University Press Cambridge, 2007.

[2] D.P.A. Dassanayake. Local orthogonal polynomial expansions for density estimation. Master's thesis, Texas Tech University, 2012.

[3] Peter J Diggle and Peter Hall. The selection of terms in an orthogonal series density estimator. *Journal of the American Statistical Association*, 81(393):230–233, 1986.

[4] Schuster E. *Communication In Statistics*. 1958.

[5] Sam Efromovich. *Nonparametric curve estimation: methods, theory and applications*. Springer, 1999.

[6] William Palin Elderton and Norman Lloyd Johnson. Systems of frequency curves. 1969.

[7] David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. *Probability theory and related fields*, 57(4):453–476, 1981.

[8] Theo Gasser and Hans-Georg Müller. *Kernel estimation of regression functions*. Springer, 1979.

[9] Jeffrey D Hart. On the choice of a truncation point in fourier series density estimation. *Journal of Statistical Computation and Simulation*, 21(2):95–116, 1985.

[10] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[11] NL Hjort and MC Jones. Locally parametric nonparametric density estimation. *The Annals of Statistics*, pages 1619–1647, 1996.

[12] Myles Hollander, Douglas A Wolfe, and Eric Chicken. *Nonparametric statistical methods*, volume 751. John Wiley & Sons, 2013.

[13] M Chris Jones. Simple boundary correction for kernel density estimation. *Statistics and Computing*, 3(3):135–146, 1993.

[14] Rhoana J Karunamuni and Tom Alberts. On boundary correction in kernel density estimation. *Statistical Methodology*, 2(3):191–212, 2005.

[15] R Kronmal and Michael Tarter. The estimation of probability densities and cumulatives by fourier series methods. *Journal of the American Statistical Association*, pages 925–952, 1968.

[16] Clive R Loader et al. Local likelihood density estimation. *The Annals of Statistics*, 24(4):1602–1618, 1996.

[17] Peter Malec and Melanie Schienle. Nonparametric kernel density estimation near the boundary. *Computational Statistics & Data Analysis*, 72:57–76, 2014.

[18] Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

[19] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[20] Herbert A Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 1926.

[21] George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[22] I. Volobouev. Statistical methods in the npstat package, 2012.

[23] Matt P Wand and M Chris Jones. *Kernel smoothing*, volume 60. Crc Press, 1994.

[24] Larry Wasserman. *All of nonparametric statistics*. Springer, 2006.

[25] Larry Wasserman. *All of nonparametric statistics*. Springer, 2006.

[26] Jinting Zhang and Jianqing Fan. Minimax kernels for nonparametric curve estimation. *International journal of computer mathematics*, 12(3):417–445, 2000.