Very fast optimal bandwidth selection for univariate kernel density estimation

VIKAS CHANDRAKANT RAYKAR and RAMANI DURAISWAMI

Perceptual Interfaces and Reality Laboratory
Department of Computer Science and Institute for Advanced Computer Studies
University of Maryland, CollegePark, MD 20783
{vikas,ramani}@cs.umd.edu

Most automatic bandwidth selection procedures for kernel density estimates require estimation of quantities involving the density derivatives. Estimation of modes and inflexion points of densities also require derivative estimates. The computational complexity of evaluating the density derivative at M evaluation points given N sample points from the density is $O(MN)$. In this paper we propose a computationally efficient $\epsilon - exact$ approximation algorithm for the univariate Gaussian kernel based density derivative estimation that reduces the computational complexity from $O(MN)$ to linear $O(N+M)$. The constant depends on the desired arbitrary accuracy, ϵ. We apply the density derivative evaluation procedure to estimate the optimal bandwidth for kernel density estimation, a process that is often intractable for large data sets. For example for $N = M = 409,600$ points while the direct evaluation of the density derivative takes around 12.76 hours the fast evaluation requires only 65 seconds with an error of around 10^{-12}. Algorithm details, error bounds, procedure to choose the parameters and numerical experiments are presented. We demonstrate the speedup achieved on the bandwidth selection using the solve-the-equation plug-in method. We also demonstrate that the proposed procedure can be extremely useful for speeding up exploratory projection pursuit techniques.

[**CS-TR-4774/UMIACS-TR-2005-73**]: December 20, 2005

1. INTRODUCTION

Kernel density estimation/regression techniques [Wand and Jones 1995] are widely used in various inference procedures in machine learning, data mining, pattern recognition, and computer vision. Efficient use of these methods require the optimal selection of the smoothing parameter called the *bandwidth* of the kernel. A plethora of techniques have been proposed for data-driven bandwidth selection [Jones et al. 1996]. The most successful state of the art methods rely on the estimation of general integrated squared *density derivative functionals.* This is the most computationally intensive task, the computational cost being $O(N^2)$, in addition to the $O(N^2)$ cost of computing the kernel density estimate. The core task is to *efficiently compute an estimate of the density derivative.* The current most practically successful approach, *solve-the-equation plug-in* method [Sheather and Jones 1991] involves the numerical solution of a non-linear equation. Iterative methods to solve this equation will involve repeated use of the density functional estimator for different bandwidths which adds much further to the computational burden. We also point out that estimation of the density derivatives also comes up in various other applications like estimation of modes and inflexion points of densities [Fukunaga and Hostetler 1975] and estimation of the derivatives of the projection index in projection pursuit algorithms [Huber 1985; Jones and Sibson 1987]. A good list of applications which require the estimation of density derivatives can be found in [Singh 1977a].

The computational complexity of evaluating the density derivative at M evaluation points given N sample points from the density is $O(MN)$. In this paper we propose a computationally efficient $\epsilon - exact$ approximation algorithm for the univariate Gaussian kernel based density derivative estimation that reduces the computational complexity from $O(MN)$ to linear $O(N + M)$. The algorithm is $\epsilon - exact$ in the sense that the constant hidden in $O(N + M)$, depends on the desired *accuracy*, which can be *arbitrary.* In fact for machine precision accuracy there is no difference between the direct and the fast methods. The proposed method can be viewed as an extension of the improved fast Gauss transform [Yang et al. 2003] proposed to accelerate the kernel density estimate.

The rest of the paper is organized as follows. In § 2 we introduce the kernel density estimate and discuss the performance of the estimator. The kernel density derivative estimate is introduced in § 3. § 4 discusses the density functionals which are used by most of the automatic bandwidth selection strategies. § 5 briefly describes the different strategies for automatic optimal bandwidth selection. The solve-the-equation plug-in method is described in detail. Our proposed fast method is described in detail in § 6. Algorithm details, error bounds, procedure to choose the parameters, and numerical experiments are presented. In § 7 we show the speedup achieved for bandwidth estimation both on simulated and real data. In § 8 we also show how the proposed procedure can be used for speeding up projection pursuit techniques. § 9 finally concludes with a brief discussion on further extensions.

2. KERNEL DENSITY ESTIMATION

A univariate random variable X on $\mathbf{R}$ has a density p if, for all Borel sets A of $\mathbf{R}$, $\int_A p(x)dx = \Pr[x \in A]$. The task of density estimation is to estimate p from an i.i.d. sample $x_1, \ldots, x_N$ drawn from p. The estimate $\widehat{p} : \mathbf{R} \times (\mathbf{R})^N \rightarrow \mathbf{R}$ is called the *density estimate.* The *parametric approach* to density estimation assumes a functional form for the density, and then estimates the unknown parameters using techniques like the maximum likelihood estimation. However unless the form of the density is known a priori, assuming a functional form for a density very often leads to erroneous inference. On the other hand *nonparametric methods* do not make any assumption on the form of the underlying density. This is sometimes referred to as *'letting the data speak for themselves'* [Wand and Jones 1995]. The price to be paid is a rate of convergence slower than $1/N$, which is typical of parametric methods. Some of the commonly used non-parametric estimators include histograms, kernel density estimators, and orthogonal series estimators [Izenman 1991]. The histogram is very sensitive to the placement of the bin edges and the asymptotic convergence is much slower than kernel density estimators [1].

The most popular non-parametric method for density estimation is the *kernel density estimator* (KDE) (also known as the *Parzen window estimator* [Parzen 1962]) given by

$$\widehat{p}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right), \tag{1}$$

where $K(u)$ is called *kernel function* and $h = h(N)$ is called the *bandwidth.* The bandwidth h is a scaling factor which goes to zero as $N \rightarrow 0$. In order that $\widehat{p}(x)$ is a bona fide density, $K(u)$ is required to satisfy the following two conditions:

$$K(u) \geq 0, \quad \int_{\mathbf{R}} K(u)du = 1. \tag{2}$$

The kernel function is essentially spreading a probability mass of $1/N$ associated with each point about its neighborhood. The most widely used kernel is the Gaussian of zero mean and unit variance [2].

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{3}$$

In this case the kernel density estimate can be written as

$$\widehat{p}(x) = \frac{1}{N\sqrt{2\pi h^2}} \sum_{i=1}^{N} e^{-(x-x_i)^2/2h^2}. \tag{4}$$

2.1 Computation complexity

The computational cost of evaluating Eq. 4 at N points is $O(N^2)$, making it prohibitively expensive. Different methods have been proposed to accelerate this sum.

If the source points are on an evenly spaced grid then we can evaluate the sum at an evenly spaced grid exactly in $O(N \log N)$ using the *fast Fourier transform*

[1] The best rate of convergence of the MISE of kernel density estimate is of order $N^{-4/5}$ while that of the histogram is of the order $N^{-2/3}$.

[2] The KDE is not very sensitive to the shape of the kernel. While the Epanechnikov kernel is the optimal kernel, in the sense that it minimizes the MISE, other kernels are not that suboptimal [Wand and Jones 1995]. The Epanechnikov kernel is not used here because it gives an estimate having a discontinuous first derivative, because of its finite support.

(FFT). One of the earliest methods, especially proposed for univariate fast kernel density estimation was based on this idea [Silverman 1982]. For irregularly spaced data, the space is divided into boxes, and the data is assigned to the closest neighboring grid points to obtain grid counts. The KDE is also evaluated at regular grid points. For target points not lying on the the grid the value is obtained by doing some sort of interpolation based on the values at the neighboring grid points. As a result there is no guaranteed error bound for such kind of methods.

The *Fast Gauss Transform*(FGT) [Greengard and Strain 1991] is an approximation algorithm that reduces the computational complexity to $O(N)$, at the expense of reduced precision. The constant depends on the desired precision, dimensionality of the problem, and the bandwidth. Yang et al. [Yang et al. 2003; Yang et al. 2005] presented an extension of the fast Gauss transform (the *improved fast Gauss transform* or IFGT) that was suitable for higher dimensional problems and provides comparable performance in lower dimensions. The main contribution of the current paper is the extension of the improved fast Gauss transform to accelerate the kernel *density derivative* estimate, and solve the optimal bandwidth problem.

Another class of methods for such problems are *dual-tree methods* [Gray and Moore 2001; 2003] which are based on space partitioning trees for both the source and target points. Using the tree data structure distance bounds between nodes can be computed. An advantage of the dual-tree methods is that they work for all common kernel choices, not necessarily Gaussian.

2.2 Performance

In order to understand the performance of the KDE we need a measure of distance between two densities. The commonly used criteria, which can be easily manipulated is the L_2 norm, also called as the *integrated square error* (ISE) [3]. The ISE between the estimate $\widehat{p}(x)$ and the actual density $p(x)$ is given by

$$\text{ISE}(\widehat{p}, p) = L_2(\widehat{p}, p) = \int_{\mathbf{R}} [\widehat{p}(x) - p(x)]^2 dx. \tag{5}$$

The ISE depends on a particular realization of N points. The ISE can be averaged over these realizations to get the *mean integrated squared error* (MISE) defined as

$$\begin{aligned} \text{MISE}(\widehat{p}, p) &= E[\text{ISE}(\widehat{p}, p)] = E\left[\int_{\mathbf{R}} [\widehat{p}(x) - p(x)]^2 dx\right] \\ &= \int_{\mathbf{R}} E[\{\widehat{p}(x) - p(x)\}^2] dx = \text{IMSE}(\widehat{p}, p), \end{aligned} \tag{6}$$

where IMSE is *integrated mean squared error.* The MISE or IMSE doesn't depend on the actual data-set as we take expectation. So this is a measure of the 'average' performance of the kernel density estimator, averaged over the support of the density and different realization of the points. The MISE for the KDE can be shown

[3] Other distance measures like mean integrated absolute error (based on the L_1 distance [Devroye and Lugosi 2000]), Kullback-Liebler divergence, and Hellinger distance are used. In this paper we use only the L_2 criterion.

to be (see § 10 for a derivation)

$$\text{MISE}(\widehat{p}, p) \; = \; \frac{1}{N}\int_{\mathbf{R}} \left[(K_h^2 * p)(x) - (K_h * p)^2(x)\right] dx + \int_{\mathbf{R}} \left[(K_h * p)(x) - p(x)\right]^2 dx, \tag{7}$$

where $*$ is the convolution operator and $K_h(x) = (1/h)K(x/h)$. The dependence of the MISE on the bandwidth h is not very explicit in the above expression. This makes it difficult to interpret the influence of the bandwidth on the performance of the estimator. An asymptotic large sample approximation for this expression is usually derived via the Taylor's series called as the AMISE, the A is for asymptotic. Based on a certain assumptions[4], the AMISE between the actual density and the estimate can be shown to be

$$\text{AMISE}(\widehat{p}, p) = \frac{1}{Nh}R(K) + \frac{1}{4}h^4\mu_2(K)^2R(p^{''}), \tag{8}$$

where

$$R(g) = \int_{\mathbf{R}} g(x)^2dx, \;\; , \; \mu_2(g) = \int_{\mathbf{R}} x^2g(x)dx, \tag{9}$$

and $p^{''}$ is the second derivative of the density p (See § 11 for a complete derivation.). The first term in the expression 8 is the integrated variance and the second term is the integrated squared bias. The bias is proportional to h^4 whereas the variance is proportional to $1/Nh$, which leads to the well known *bias-variance tradeoff.*

Based on the AMISE expression the optimal bandwidth h_{AMISE} can be obtained by differentiating Eq. 8 w.r.t. bandwidth h and setting it to zero.

$$h_{AMISE} = \left[\frac{R(K)}{\mu_2(K)^2R(p'')N}\right]^{1/5}. \tag{10}$$

However this expression cannot be used directly since $R(p^{''})$ depends on the second derivative of the density p, which we are trying to estimate in the first place. We need to use an estimate of $R(p^{''})$.

Substituting Eq. 10 in Eq. 8 the minimum AMISE that can be attained is

$$\inf_h \text{AMISE}(\widehat{p}, p) = \frac{5}{4}\left[\mu_2(K)^2R(K)^4R(p^{''})\right]^{1/5} N^{-4/5}. \tag{11}$$

This expression shows that the best rate of convergence of the MISE of KDE is of order $N^{-4/5}$.

3. KERNEL DENSITY DERIVATIVE ESTIMATION

In order to estimate $R(p^{''})$ we will need an estimate of the density derivative. A simple estimator for the density derivative can be obtained by taking the derivative of the kernel density estimate $\widehat{p}(x)$ defined earlier [Bhattacharya 1967; Schuster

[4] The second derivative $p^{''}(x)$ is continuous, square integrable and ultimately monotone. $\lim_{N\to\infty} h = 0$ and $\lim_{N\to\infty} Nh = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N$.The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2K(z)dz < \infty$).

1969] [5]. If the kernel K is differentiable r times then the r^{th} density derivative estimate $\widehat{p}^{(r)}(x)$ can be written as

$$\widehat{p}^{(r)}(x) = \frac{1}{Nh^{r+1}} \sum_{i=1}^{N} K^{(r)} \left(\frac{x - x_i}{h} \right), \tag{12}$$

where $K^{(r)}$ is the r^{th} derivative of the kernel K. The r^{th} derivative of the Gaussian kernel $k(u)$ is given by

$$K^{(r)}(u) = (-1)^r H_r(u) K(u) \tag{13}$$

where $H_r(u)$ is the r^{th} Hermite polynomial. The Hermite polynomials are set of orthogonal polynomials [Abramowitz and Stegun 1972] . The first few Hermite polynomials are

$$H_0(u) \ = \ 1, \ H_1(u) = u, \text{ and } H_2(u) = u^2 - 1.$$

Hence the density derivative estimate with the Gaussian kernel can be written as

$$\widehat{p}^{(r)}(x) = \frac{(-1)^r}{\sqrt{2\pi} N h^{r+1}} \sum_{i=1}^{N} H_r \left(\frac{x - x_i}{h} \right) e^{-(x-x_i)^2/2h^2}. \tag{14}$$

3.1 Computational complexity

The computational complexity of evaluating the r^{th} derivative of the density estimate due to N points at M target locations is $O(rNM)$.

3.2 Performance

Similar to the analysis done for KDE the AMISE for the kernel density derivative estimate, under certain assumptions [6], can be shown to be (See § 12 for a complete derivation)

$$\text{AMISE}(\widehat{p}^{(r)}, p^{(r)}) = \frac{R(K^{(r)})}{Nh^{2r+1}} + \frac{h^4}{4} \mu_2(K)^2 R(p^{(r+2)}). \tag{15}$$

It can be observed that the AMISE for estimating the r^{th} derivative depends upon the the $(r+2)^{th}$ derivative of the true density. Differentiating Eq. 15 w.r.t. bandwidth h and setting it to zero we obtain the optimal bandwidth h^r_{AMISE} to estimate the r^{th} density derivative.

$$h^r_{AMISE} = \left[\frac{R(K^{(r)})(2r+1)}{\mu_2(K)^2 R(p^{(r+2)}) N} \right]^{1/2r+5}. \tag{16}$$

Substituting Eq. 16 in the equation for AMISE, the minimum AMISE that can be attained is

$$\inf_h \text{AMISE}(\widehat{p}^{(r)}, p^{(r)}) = C \left[\mu_2(K)^2 R(K)^{4(2r+1)} R(p^{''}) \right]^{2r+1/2r+5} N^{-4/2r+5}.$$

[5] Some better estimators which are not necessarily the p^{th} order derivatives of the KDE have been proposed [Singh 1977b].

[6] The $(r+2)^{th}$ derivative $p^{(r+2)}(x)$ is continuous, square integrable and ultimately monotone. $\lim_{N\rightarrow\infty} h = 0$ and $\lim_{N\rightarrow\infty} Nh^{2r+1} = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N^{2r+1}$. The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2 K(z)dz < \infty$) .

where C is a constant depending on r. This expression shows that the best rate of convergence of the MISE of KDE of the derivative is of order $N^{-4/2r+5}$. The rate becomes slower for higher values of r, which says that estimating the derivative is more difficult than estimating the density.

4. ESTIMATION OF DENSITY FUNCTIONALS

Rather than the actual density derivative methods for automatic bandwidth selection require the estimation of what are known as *density functionals.* The general integrated squared density derivative functional is defined as

$$R(p^{(s)}) = \int_{\mathbf{R}} \left[p^{(s)}(x)\right]^2 dx. \tag{17}$$

Using integration by parts, this can be written in the following form,

$$R(p^{(s)}) = (-1)^s \int_{\mathbf{R}} p^{(2s)}(x)p(x)dx. \tag{18}$$

More specifically for even s we are interested in estimating density functionals of the form,

$$\Phi_r = \int_{\mathbf{R}} p^{(r)}(x)p(x)dx = E\left[p^{(r)}(X)\right]. \tag{19}$$

An estimator for Φ_r is,

$$\widehat{\Phi}_r = \frac{1}{N}\sum_{i=1}^{N} \widehat{p}^{(r)}(x_i). \tag{20}$$

where $\widehat{p}^{(r)}(x_i)$ is the estimate of the r^{th} derivative of the density $p(x)$ at $x = x_i$. Using a kernel density derivative estimate for $\widehat{p}^{(r)}(x_i)$ (Eq. 12) we have

$$\widehat{\Phi}_r = \frac{1}{N^2h^{r+1}}\sum_{i=1}^{N}\sum_{j=1}^{N} K^{(r)}(\frac{x_i - x_j}{h}). \tag{21}$$

It should be noted that computation of $\widehat{\Phi}_r$ is $O(rN^2)$ and hence can be very expensive if a direct algorithm is used.

4.1 Performance

The asymptotic MSE for the density functional estimator under certain assumptions [7] is as follows. (See § 13 for a complete derivation.)

$$\begin{aligned}\text{AMSE}(\widehat{\Phi}_r, \Phi_r) \;=\; & \left[\frac{1}{Nh^{r+1}}K^{(r)}(0) + \frac{1}{2}h^2\mu_2(K)\Phi_{r+2}\right]^2 + \frac{2}{N^2h^{2r+1}}\Phi_0 R(K^{(r)}) \\ & + \frac{4}{N}\left[\int p^{(r)}(y)^2p(y)dy - \Phi_r^2\right]\end{aligned} \tag{22}$$

[7] The density p had $k > 2$ continuous derivatives which are ultimately monotone. The $(r+2)^{th}$ derivative $p^{(r+2)}(x)$ is continuous, square integrable and ultimately monotone. $\lim_{N\to\infty} h = 0$ and $\lim_{N\to\infty} Nh^{2r+1} = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N^{2r+1}$. The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2K(z)dz < \infty$) .

The optimal bandwidth for estimating the density functional is chosen the make the bias term zero. The optimal bandwidth is given by [Wand and Jones 1995]

$$g_{\rm MSE} = \left[\frac{-2K^{(r)}(0)}{\mu_2(K)\Phi_{r+2}N} \right]^{1/r+3}. \tag{23}$$

5. AMISE OPTIMAL BANDWIDTH SELECTION

For a practical implementation of KDE the choice of the bandwidth h is very important. Small h leads to an estimator with small bias and large variance. Large h leads to a small variance at the expense of increase in bias. The bandwidth h has to be chosen optimally. Various techniques have been proposed for optimal bandwidth selection. A brief survey can be found in [Jones et al. 1996] and [Wand and Jones 1995]. The best known of these include rules of thumb, oversmoothing, least squares cross-validation, biased cross-validation, direct plug-in methods, solve-the-equation plug-in method, and the smoothed bootstrap.

5.1 Brief review of different methods

Based on the AMISE expression the optimal bandwidth h_{AMISE} has the following form,

$$h_{\rm AMISE} = \left[\frac{R(K)}{\mu_2(K)^2 R(p'')N} \right]^{1/5}. \tag{24}$$

However this expression cannot be used directly since $R(p^{''})$ depends on the second derivative of the density p, which we are trying to estimate in the first place.

The *rules of thumb* use an estimate of $R(p^{''})$ assuming that the data is generated by some parametric form of the density (typically a normal distribution).

The *oversmoothing* methods rely on the fact that there is a simple upper bound for the AMISE-optimal bandwidth for estimation of densities with a fixed value of a particular scale measure. The *least squares cross-validation* directly minimize the MISE based on a "leave-one-out" kernel density estimator. The problem is that the function to be minimized has fairly large number of local minima and also the practical performance of this method is somewhat disappointing.

The *biased cross-validation* uses the AMISE instead of using the exact MISE formula. This is more stable than the least squares cross-validation but has a large bias.

The *plug-in methods* use an estimate of the density functional $R(p^{''})$ in Eq. 24. However this is not completely automatic since estimation of $R(p^{''})$ requires the specification of another *pilot bandwidth g*. This bandwidth for estimation of the density functional is quite different from the the bandwidth h used for the kernel density estimate. As discussed in Section 4 we can find an expression for the AMISE-optimal bandwidth for the estimation of $R(p^{''})$. However this bandwidth will depend on an unknown density functional $R(p^{'''})$. This problem will continue since the optimal bandwidth for estimating $R(p^{(s)})$ will depend on $R(p^{(s+1)})$. The usual strategy used by the *direct plug-in* methods is to estimate $R(p^{(l)})$ for some l, with bandwidth chosen with reference to a parametric family, usually a normal density. This method is usually referred to as the *l-stage direct plug-in* method. As the the number of stages l increases the bias of the bandwidth decreases, since the dependence on the assumption of some parametric family decreases. However this comes at the price of the estimate being more variable. There is no good method for the choice of l, the most common choice being $l = 2$.

5.2 Solve-the-equation plug-in method

The most successful among all the current methods, both empirically and theoretically, is the *solve-the-equation plug-in* method [Jones et al. 1996]. This method differs from the direct plug-in approach in that the pilot bandwidth used to estimate $R(p^{''})$ is written as a function of the kernel bandwidth h. We use the following version as described in [Sheather and Jones 1991]. The AMISE optimal bandwidth is the solution to the equation

$$h = \left[\frac{R(K)}{\mu_2(K)^2\widehat{\Phi}_4[\gamma(h)]N}\right]^{1/5}, \tag{25}$$

where $\widehat{\Phi}_4[\gamma(h)]$ is an estimate of $\Phi_4 = R(p^{''})$ using the pilot bandwidth $\gamma(h)$, which depends on the kernel bandwidth h. The bandwidth is chosen such that it minimizes the asymptotic MSE for the estimation of Φ_4 and is given by

$$g_{\text{MSE}} = \left[\frac{-2K^{(4)}(0)}{\mu_2(K)\Phi_6 N}\right]^{1/7}. \tag{26}$$

Substituting for N from Eq. 24 g_{MSE} can be written as a function of h as follows

$$g_{\text{MSE}} = \left[\frac{-2K^{(4)}(0)\mu_2(K)\Phi_4}{R(K)\Phi_6}\right]^{1/7} h_{\text{AMISE}}^{5/7}. \tag{27}$$

This suggest that we set

$$\gamma(h) = \left[\frac{-2K^{(4)}(0)\mu_2(K)\widehat{\Phi}_4(g_1)}{R(K)\widehat{\Phi}_6(g_2)}\right]^{1/7} h^{5/7}, \tag{28}$$

where $\widehat{\Phi}_4(g_1)$ and $\widehat{\Phi}_6(g_2)$ are estimates of Φ_4 and Φ_6 using bandwidths g_1 and g_2 respectively.

$$\widehat{\Phi}_4(g_1) = \frac{1}{N(N-1)g_1^5}\sum_{i=1}^{N}\sum_{j=1}^{N} K^{(4)}(\frac{x_i - x_j}{g_1}). \tag{29}$$

$$\widehat{\Phi}_6(g_2) = \frac{1}{N(N-1)g_2^7}\sum_{i=1}^{N}\sum_{j=1}^{N} K^{(6)}(\frac{x_i - x_j}{g_2}). \tag{30}$$

The bandwidths g_1 and g_2 are chosen such that it minimizes the asymptotic MSE.

$$g_1 = \left[\frac{-2K^{(4)}(0)}{\mu_2(K)\widehat{\Phi}_6 N}\right]^{1/7} \quad g_2 = \left[\frac{-2K^{(6)}(0)}{\mu_2(K)\widehat{\Phi}_8 N}\right]^{1/9}, \tag{31}$$

where $\widehat{\Phi}_6$ and $\widehat{\Phi}_8$ are estimators for Φ_6 and Φ_8 respectively. We can use a similar strategy for estimation of Φ_6 and Φ_8. However this problem will continue since the optimal bandwidth for estimating Φ_r will depend on Φ_{r+2}. The usual strategy is to estimate a Φ_r at some stage, using a quick and simple estimate of bandwidth chosen with reference to a parametric family, usually a normal density. It has been observed that as the the number of stages increases the variance of the bandwidth increases. The most common choice is to use only two stages.

If p is a normal density with variance σ^2 then for even r we can compute Φ_r exactly [Wand and Jones 1995].

$$\Phi_r = \frac{(-1)^{r/2} r!}{(2\sigma)^{r+1}(r/2)!\pi^{1/2}}. \tag{32}$$

An estimator of Φ_r will use an estimate $\widehat{\sigma}^2$ of the variance. Based on this we can write an estimator for Φ_6 and Φ_8 as follows.

$$\widehat{\Phi}_6 = \frac{-15}{16\sqrt{\pi}}\widehat{\sigma}^{-7}, \ \widehat{\Phi}_8 = \frac{105}{32\sqrt{\pi}}\widehat{\sigma}^{-9}. \tag{33}$$

The two stage solve-the-equation method using the Gaussian kernel can be summarized as follows.

(1) Compute an estimate $\widehat{\sigma}$ of the standard deviation σ.

(2) Estimate the density functionals Φ_6 and Φ_8 using the normal scale rule.

$$\widehat{\Phi}_6 = \frac{-15}{16\sqrt{\pi}}\widehat{\sigma}^{-7}, \ \widehat{\Phi}_8 = \frac{105}{32\sqrt{\pi}}\widehat{\sigma}^{-9}.$$

(3) Estimate the density functionals Φ_4 and Φ_6 using the kernel density estimators with the optimal bandwidth based on the asymptotic MSE.

$$g_1 = \left[\frac{-6}{\sqrt{2\pi}\widehat{\Phi}_6 N}\right]^{1/7} \quad g_2 = \left[\frac{30}{\sqrt{2\pi}\widehat{\Phi}_8 N}\right]^{1/9}$$

$$\widehat{\Phi}_4(g_1) \ = \ \frac{1}{N(N-1)\sqrt{2\pi}g_1^5}\sum_{i=1}^{N}\sum_{j=1}^{N} H_4\left(\frac{x_i - x_j}{g_1}\right) e^{-(x_i-x_j)^2/2g_1^2}.$$

$$\widehat{\Phi}_6(g_2) \ = \ \frac{1}{N(N-1)\sqrt{2\pi}g_2^7}\sum_{i=1}^{N}\sum_{j=1}^{N} H_6\left(\frac{x_i - x_j}{g_2}\right) e^{-(x_i-x_j)^2/2g_2^2}.$$

(4) The bandwidth is the solution to the equation

$$h - \left[\frac{1}{2\sqrt{\pi}\widehat{\Phi}_4[\gamma(h)]N}\right]^{1/5} = 0,$$

where

$$\widehat{\Phi}_4[\gamma(h)] \ = \ \frac{1}{N(N-1)\sqrt{2\pi}\gamma(h)^5}\sum_{i=1}^{N}\sum_{j=1}^{N} H_4\left(\frac{x_i - x_j}{\gamma(h)}\right) e^{-(x_i-x_j)^2/2\gamma(h)^2},$$

and

$$\gamma(h) = \left[\frac{-6\sqrt{2}\widehat{\Phi}_4(g_1)}{\widehat{\Phi}_6(g_2)}\right]^{1/7} h^{5/7}.$$

This equation can be solved using any numerical routine like the Newton-Raphson method.

The main computational bottleneck is the estimation of Φ which is of $O(N^2)$.

6. FAST DENSITY DERIVATIVE ESTIMATION

The r^{th} kernel density derivative estimate using the Gaussian kernel of bandwidth h is given by

$$\widehat{p}^{(r)}(x) = \frac{(-1)^r}{\sqrt{2\pi}Nh^{r+1}} \sum_{i=1}^{N} H_r\left(\frac{x - x_i}{h}\right) e^{-(x-x_i)^2/2h^2}. \tag{34}$$

Let us say we have to estimate the density derivative at M *target points*, $\{y_j \in \mathbf{R}\}_{j=1}^{M}$. More generally we need to evaluate the following sum,

$$G_r(y_j) = \sum_{i=1}^{N} q_i H_r\left(\frac{y_j - x_i}{h_1}\right) e^{-(y_j-x_i)^2/h_2^2} \quad j = 1, \ldots, M, \tag{35}$$

where $\{q_i \in \mathbf{R}\}_{i=1}^{N}$ will be referred to as the *source weights*, $h_1 \in \mathbf{R}^+$ is the bandwidth of the Gaussian and $h_2 \in \mathbf{R}^+$ is the bandwidth of the Hermite. The computational complexity of evaluating Eq. 35 is $O(rNM)$. The fast algorithm is based on separating the x_i and y_j in the Gaussian via the factorization of the Gaussian by Taylor series and retaining only the first few terms so that the error due to truncation is less than the desired error. The Hermite function is factorized via the binomial theorem. For any given $\epsilon > 0$ the algorithm computes an approximation $\widehat{G}_r(y_j)$ such that

$$\left|\frac{\hat{G}_r(y_j) - G_r(y_j)}{Q}\right| \leq \epsilon, \tag{36}$$

where $Q = \sum_{i=1}^{N} |q_i|$. We call $\widehat{G}_r(y_j)$ an $\epsilon -$ *exact* approximation to $G_r(y_j)$.

6.1 Factorization of the Gaussian

For any point $x_* \in \mathbf{R}$ the Gaussian can be written as,

$$\begin{aligned} e^{-\|y_j-x_i\|^2/h_2^2} &= e^{-\|(y_j-x_*)-(x_i-x_*)\|^2/h_2^2} \\ &= e^{-\|x_i-x_*\|^2/h_2^2} e^{-\|y_j-x_*\|^2/h_2^2} e^{2(x_i-x_*)(y_j-x_*)/h_2^2}. \end{aligned} \tag{37}$$

In Eq. 37 the first exponential $e^{-\|x_i-x_*\|^2/h^2}$ depends only on the source coordinates x_i. The second exponential $e^{-\|y_j-x_*\|^2/h^2}$ depends only on the target coordinates y_j. However for the third exponential $e^{2(y_j-x_*)(x_i-x_*)/h^2}$ the source and target are entangled. This entanglement is separated using the Taylor's series expansion.

The factorization of the Gaussian and the evaluation of the error bounds are based on the Taylor's series and Lagrange's evaluation of the remainder which we state here without the proof.

THEOREM 6.1. [Taylor's Series] *For any point $x_* \in \mathbf{R}$, let $I \subset \mathbf{R}$ be an open set containing the point x_*. Let $f : I \rightarrow \mathbf{R}$ be a function which is n times differentiable* *on I. Then for any $x \in I$, there is a $\theta \in \mathbf{R}$ with $0 < \theta < 1$ such that*

$$f(x) \;=\; \sum_{k=0}^{n-1} \frac{1}{k!}(x-x_*)^k f^{(k)}(x_*) + \frac{1}{n!}(x-x_*)^n f^{(n)}(x_* + \theta(x-x_*)), \qquad (38)$$

where $f^{(k)}$ is the k^{th} derivative of the function f.

Based on the above theorem we have the following corollary.

Corollary 6.1. *Let $B_{r_x}(x_*)$ be a open interval of radius r_x with center $x_* \in \mathbf{R}$, i.e., $B_{r_x}(x_*) = \{x : \|x - x_*\| < r_x\}$. Let $h \in \mathbf{R}^+$ be a positive constant and $y \in \mathbf{R}$ be a fixed point such that $\|y - x_*\| < r_y$. For any $x \in B_{r_x}(x_*)$ and any non-negative integer p the function $f(x) = e^{2(x-x_*)(y-x_*)/h^2}$ can be written as*

$$f(x) \;=\; e^{2(x-x_*)(y-x_*)/h^2} = \sum_{k=0}^{p-1} \frac{2^k}{k!}\left(\frac{x-x_*}{h}\right)^k \left(\frac{y-x_*}{h}\right)^k + R_p(x), \qquad (39)$$

and the residual

$$\begin{aligned} R_p(x) \;&\le\; \frac{2^p}{p!}\left(\frac{\|x-x_*\|}{h}\right)^p \left(\frac{\|y-x_*\|}{h}\right)^p e^{2\|x-x_*\|\|y-x_*\|/h^2}. \\ &<\; \frac{2^p}{p!}\left(\frac{r_x r_y}{h^2}\right)^p e^{2 r_x r_y/h^2}. \end{aligned} \qquad (40)$$

Proof. Let us define a new function $g(x) = e^{2[x(y-x_*)]/h^2}$. Using the result

$$g^{(k)}(x_*) = \frac{2^k}{h^k} e^{2[x_*(y-x_*)]/h^2} \left(\frac{y-x_*}{h}\right)^k \qquad (41)$$

and Theorem 6.1, we have for any $x \in B_{r_x}(x_*)$ there is a $\theta \in \mathbf{R}$ with $0 < \theta < 1$ such that

$$\begin{aligned} g(x) \;=\; e^{2[x_*(y-x_*)]/h^2} & \left\{ \sum_{k=0}^{p-1} \frac{2^k}{k!}\left(\frac{x-x_*}{h}\right)^k \left(\frac{y-x_*}{h}\right)^k \right. \\ & \left. + \frac{2^p}{p!}\left(\frac{x-x_*}{h}\right)^p \left(\frac{y-x_*}{h}\right)^p e^{2\theta[(x-x_*).(y-x_*)]/h^2} \right\}. \end{aligned}$$

Hence

$$f(x) \;=\; e^{2(x-x_*)(y-x_*)/h^2} = \sum_{k=0}^{p-1} \frac{2^k}{k!}\left(\frac{x-x_*}{h}\right)^k \left(\frac{y-x_*}{h}\right)^k + R_p(x),$$

where,

$$R_p(x) = \frac{2^p}{p!}\left(\frac{x-x_*}{h}\right)^p \left(\frac{y-x_*}{h}\right)^p e^{2\theta[(x-x_*)(y-x_*)]/h^2}.$$

The remainder is bounded as follows.

$$\begin{aligned} R_p(x) &\leq \frac{2^p}{p!}\left(\frac{\|x-x_*\|}{h}\right)^p\left(\frac{\|y-x_*\|}{h}\right)^p e^{2\theta\|x-x_*\|\|y-x_*\|/h^2}, \\ &\leq \frac{2^p}{p!}\left(\frac{\|x-x_*\|}{h}\right)^p\left(\frac{\|y-x_*\|}{h}\right)^p e^{2\|x-x_*\|\|y-x_*\|/h^2} \text{ [Since } 0<\theta<1], \\ &< \frac{2^p}{p!}\left(\frac{r_x r_y}{h^2}\right)^p e^{2r_x r_y/h^2} \text{ [Since } \|x-x_*\|<r_x \text{ and } \|y-x_*\|<r_y]. \end{aligned}$$

□

Using Corollary 6.1 the Gaussian can now be factorized as

$$e^{-\|y_j-x_i\|^2/h_2^2} = \sum_{k=0}^{p-1}\frac{2^k}{k!}\left[e^{-\|x_i-x_*\|^2/h_2^2}\left(\frac{x_i-x_*}{h_2}\right)^k\right]\left[e^{-\|y_j-x_*\|^2/h_2^2}\left(\frac{y_j-x_*}{h_2}\right)^k\right]+error_p. \tag{42}$$

where,

$$error_p \leq \frac{2^p}{p!}\left(\frac{\|x_i-x_*\|}{h_2}\right)^p\left(\frac{\|y_j-x_*\|}{h_2}\right)^p e^{-(\|x_i-x_*\|-\|y_j-x_*\|)^2/h_2^2}. \tag{43}$$

6.2 Factorization of the Hermite polynomial

The r^{th} Hermite polynomial can be written as [Wand and Jones 1995]

$$H_r(x) = \sum_{l=0}^{\lfloor r/2\rfloor} a_l x^{r-2l}, \text{where } a_l = \frac{(-1)^l r!}{2^l l!(r-2l)!}.$$

Hence,

$$H_r\left(\frac{y_j-x_i}{h_1}\right) = \sum_{l=0}^{\lfloor r/2\rfloor} a_l\left(\frac{y_j-x_*}{h_1}-\frac{x_i-x_*}{h_1}\right)^{r-2l}.$$

Using the binomial theorem $(a+b)^n = \sum_{m=0}^{n}\binom{n}{m}a^m b^{n-m}$, the x_i and y_j can be separated as follows.

$$\left(\frac{y_j-x_*}{h_1}-\frac{x_i-x_*}{h_1}\right)^{r-2l} = \sum_{m=0}^{r-2l}(-1)^m\binom{r-2l}{m}\left(\frac{x_i-x_*}{h_1}\right)^m\left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m}.$$

Substituting in the previous equation we have

$$H_r\left(\frac{y_j-x_i}{h_1}\right) = \sum_{l=0}^{\lfloor r/2\rfloor}\sum_{m=0}^{r-2l} a_{lm}\left(\frac{x_i-x_*}{h_1}\right)^m\left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m} \tag{44}$$

where,

$$a_{lm} = \frac{(-1)^{l+m} r!}{2^l l! m!(r-2l-m)!}. \tag{45}$$

6.3 Regrouping of the terms

Using Eq. 42 and 44, $G_r(y_j)$ after ignoring the error terms can be approximated as

$$\begin{aligned}\widehat{G}_r(y_j) &= \sum_{k=0}^{p-1}\sum_{l=0}^{\lfloor r/2\rfloor}\sum_{m=0}^{r-2l} a_{lm}\left[\frac{2^k}{k!}\sum_{i=1}^{N} q_i e^{-\|x_i-x_*\|^2/h_2^2}\left(\frac{x_i-x_*}{h_2}\right)^k\left(\frac{x_i-x_*}{h_1}\right)^m\right] \\ &\quad \left[e^{-\|y_j-x_*\|^2/h_2^2}\left(\frac{y_j-x_*}{h_2}\right)^k\left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m}\right] \\ &= \sum_{k=0}^{p-1}\sum_{l=0}^{\lfloor r/2\rfloor}\sum_{m=0}^{r-2l} a_{lm}B_{km}e^{-\|y_j-x_*\|^2/h_2^2}\left(\frac{y_j-x_*}{h_2}\right)^k\left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m}\end{aligned}$$

where

$$B_{km} = \frac{2^k}{k!}\sum_{i=1}^{N} q_i e^{-\|x_i-x_*\|^2/h_2^2}\left(\frac{x_i-x_*}{h_2}\right)^k\left(\frac{x_i-x_*}{h_1}\right)^m .$$

The coefficients B_{km} can be evaluated separately in $O(prN)$. Evaluation of $\widehat{G}_r(y_j)$ at M points is $O(pr^2M)$. Hence the computational complexity has reduced from the quadratic $O(rNM)$ to the linear $O(prN + pr^2M)$.

6.4 Space subdivision

Thus far, we have used the Taylor's series expansion about a certain point x_*. However if we use the same x_* for all the points we typically would require very high truncation number p since the Taylor's series gives good approximation only in a small open interval around x_*. We uniformly sub-divide the space into K intervals of length $2r_x$. The N source points are assigned into K clusters, S_n for $n = 1, \dots, K$ with c_n being the center of each cluster. The aggregated coefficients are now computed for each cluster and the total contribution from all the clusters is summed up.

$$\widehat{G}_r(y_j) = \sum_{n=1}^{K}\sum_{k=0}^{p-1}\sum_{l=0}^{\lfloor r/2\rfloor}\sum_{m=0}^{r-2l} a_{lm}B^n_{km}e^{-\|y_j-c_n\|^2/h_2^2}\left(\frac{y_j-c_n}{h_2}\right)^k\left(\frac{y_j-c_n}{h_1}\right)^{r-2l-m} \tag{46}$$

where,

$$B^n_{km} = \frac{2^k}{k!}\sum_{x_i\in S_n} q_i e^{-\|x_i-x_*\|^2/h_2^2}\left(\frac{x_i-x_*}{h_2}\right)^k\left(\frac{x_i-x_*}{h_1}\right)^m . \tag{47}$$

6.5 Decay of the Gaussian

Since the Gaussian decays very rapidly a further speedup is achieved if we ignore all the sources belonging to a cluster if the cluster is greater than a certain distance from the target point, i.e., $\|y_j - c_n\| > r_y$. The cluster cutoff radius r_y depends on the desired error ϵ. Substituting $h_1 = h$ and $h_2 = \sqrt{2}h$ we have

$$\widehat{G}_r(y_j) = \sum_{\|y_j-c_n\|\le r_y}\sum_{k=0}^{p-1}\sum_{l=0}^{\lfloor r/2\rfloor}\sum_{m=0}^{r-2l} a_{lm}B^n_{km}e^{-\|y_j-c_n\|^2/2h^2}\left(\frac{y_j-c_n}{h}\right)^{k+r-2l-m} \tag{48}$$

where,

$$B_{km}^n = \frac{1}{k!} \sum_{x_i \in S_n} q_i e^{-\|x_i - x_*\|^2/2h^2} \left(\frac{x_i - x_*}{h} \right)^{k+m}. \tag{49}$$

6.6 Computational and space complexity

Computing the coefficients B_{km}^n for all the clusters is $O(prN)$. Evaluation of $\widehat{G}_r(y_j)$ at M points is $O(npr^2M)$, where n if the maximum number of neighbor clusters which influence y_j. Hence the total computational complexity is $O(prN + npr^2M)$. Assuming $N = M$ the total computational complexity is $O(cN)$ where the constant $c = pr + npr^2$ depends on the desired error, the bandwidth, and r. For each cluster we need to store all the pr coefficients. Hence the storage needed is of $O(prK + N + M)$.

6.7 Error bounds and choosing the parameters

Given any $\epsilon > 0$, we want to choose the following parameters, K (the number of intervals), r_y (the cut off radius for each cluster), and p (the truncation number) such that for any target point y_j

$$\left| \frac{\hat{G}_r(y_j) - G_r(y_j)}{Q} \right| \leq \epsilon, \tag{50}$$

where $Q = \sum_{i=1}^N |q_i|$. Let us define Δ_{ij} to be the point wise error in $\widehat{G}_r(y_j)$ contributed by the i^{th} source x_i. We now require that

$$|\hat{G}_r(y_j) - G_r(y_j)| = \left| \sum_{i=1}^N \Delta_{ij} \right| \leq \sum_{i=1}^N |\Delta_{ij}| \leq \sum_{i=1}^N |q_i| \epsilon. \tag{51}$$

One way to achieve this is to let

$$|\Delta_{ij}| \leq |q_i| \epsilon \ \forall i = 1, \ldots, N.$$

We choose this strategy because it helps us to get tighter bounds. Let c_n be the center of the cluster to which x_i belongs. There are two different ways in which a source can contribute to the error. The first is due to ignoring the cluster S_n if it is outside a given radius r_y from the target point y_j. In this case,

$$\Delta_{ij} = q_i H_r \left(\frac{y_j - x_i}{h} \right) e^{-\|y_j - x_i\|^2/2h^2}. \tag{52}$$

For all clusters which are within a distance r_y from the target point the error is due to the truncation of the Taylor's series after order $p - 1$. From Eqs. 43 and using the fact that $h_1 = h$ and $h_2 = \sqrt{2}h$ we have,

$$\Delta_{ij} \leq \frac{q_i}{p!} H_r \left(\frac{y_j - x_i}{h} \right) \left(\frac{\|x_i - c_n\|}{h} \right)^p \left(\frac{\|y_j - c_n\|}{h} \right)^p e^{-(\|x_i - c_n\| - \|y_j - c_n\|)^2/2h^2}. \tag{53}$$

6.7.1 *Choosing the cut off radius.* From Eq. 52 we have

$$\left| H_r \left(\frac{y_j - x_i}{h} \right) \right| e^{-\|y_j - x_i\|^2/2h^2} \leq \epsilon \tag{54}$$

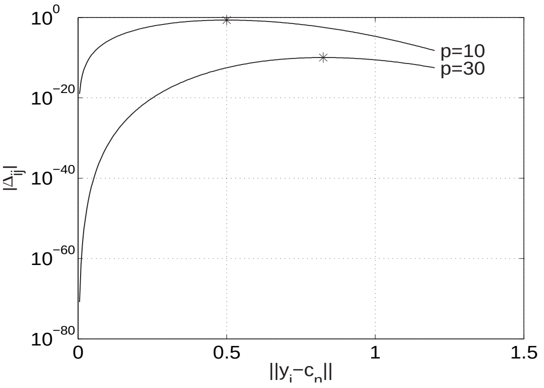


Fig. 1. The error at y_j due to source x_i, i.e., Δ_{ij} [Eq. 62] as a function of $\|y_j - c_n\|$ for different values of p and for $h = 0.1$ and $r = 4$. The error increases as a function of $\|y_j - c_n\|$, reaches a maximum and then starts decreasing. The maximum is marked as '*'. $q_i = 1$ and $\|x_i - c_n\| = 0.1$.

We use the following inequality to bound the Hermite polynomial [Baxter and Roussos 2002].

$$\left|H_r\left(\frac{y_j - x_i}{h}\right)\right| \leq \sqrt{r!}e^{\|y_j - x_i\|^2/4h^2}. \tag{55}$$

Substituting this bound in Eq. 54 we have

$$e^{-\|y_j - x_i\|^2/4h^2} \leq \epsilon/\sqrt{r!}. \tag{56}$$

This implies that $\|y_j - x_i\| > 2h\sqrt{\ln{(\sqrt{r!}/\epsilon)}}$. Using the reverse triangle inequality, $\|a - b\| \geq \big|\|a\| - \|b\|\big|$, and the fact that $\|y_j - c_n\| > r_y$ and $\|x_i - c_n\| \leq r_x$, we have

$$\begin{aligned} \|y_j - x_i\| &= \|(y_j - c_n) - (x_i - c_n)\| \\ &\geq \big|\|(y_j - c_n)\| - \|(x_i - c_n)\|\big| \\ &> \big|r_y - r_x\big| \end{aligned} \tag{57}$$

So in order that the error due to ignoring the faraway clusters is less than $|q_i|\epsilon$ we have to choose r_y such that

$$\big|r_y - r_x\big| > 2h\sqrt{\ln{(\sqrt{r!}/\epsilon)}}. \tag{58}$$

If we choose $r_y > r_x$ then,

$$r_y > r_x + 2h\sqrt{\ln{(\sqrt{r!}/\epsilon)}}. \tag{59}$$

Let R be the maximum distance between any source and target point. The we choose the cutoff radius as

$$r_y > r_x + \min\left(R, 2h\sqrt{\ln{(\sqrt{r!}/\epsilon)}}\right). \tag{60}$$

6.7.2 *Choosing the truncation number.* For all sources for which $\|y_j - c_k\| \leq r_y$ we have

$$\Delta_{ij} \leq \frac{q_i}{p!} H_r \left(\frac{y_j - x_i}{h}\right) \left(\frac{\|x_i - c_n\|}{h}\right)^p \left(\frac{\|y_j - c_n\|}{h}\right)^p e^{-(\|x_i-c_n\|-\|y_j-c_n\|)^2/2h^2}. \tag{61}$$

Using the bound on the Hermite polynomial (Eq. 55) this can be written as

$$|\Delta_{ij}| \leq \frac{|q_i|\sqrt{r!}}{p!} \left(\frac{\|x_i - c_n\|}{h}\right)^p \left(\frac{\|y_j - c_n\|}{h}\right)^p e^{-(\|x_i-c_n\|-\|y_j-c_n\|)^2/4h^2}. \tag{62}$$

For a given source x_i we have to choose p such that $|\Delta_{ij}| \leq |q_i|\epsilon$. Δ_{ij} depends both on distance between the source and the cluster center, i.e., $\|x_i - c_n\|$ and the distance between the target and the cluster center, i.e., $\|y_j - c_n\|$. The speedup is achieved because at each cluster S_n we sum up the effect of all the sources. As a result we do not have a knowledge of $\|y_j - c_n\|$. So we will have to bound the right hand side of Eq. 62, such that it is independent of $\|y_j - c_n\|$. Fig. 1 shows the error at y_j due to source x_i, i.e., $|\Delta_{ij}|$ [Eq. 62] as a function of $\|y_j - c_n\|$ for different values of p and for $h = 0.1$ and $r = 4$. The error increases as a function of $\|y_j - c_n\|$, reaches a maximum and then starts decreasing. The maximum is attained at (obtained by taking the first derivative of the R.H.S. of Eq. 62 and setting it to zero),

$$\|y_j - c_n\|_* = \frac{\|x_i - c_n\| + \sqrt{\|x_i - c_n\|^2 + 8ph^2}}{2} \tag{63}$$

Hence we choose p such that,

$$|\Delta_{ij}| \left|_{[\|y_j - c_n\| = \|y_j - c_n\|_*]}\right. \leq |q_i|\epsilon. \tag{64}$$

In case $\|y_j - c_n\|_* > r_y$ we need to choose p based on r_y, since Δ_{ij} will be much lower there. Hence out strategy for choosing p is (we choose $r_x = h/2$.),

$$|\Delta_{ij}| \left|_{[\|y_j - c_n\| = \min(\|y_j - c_n\|_*, r_y),\ \|x_i - c_n\| = h/2]}\right. \leq |q_i|\epsilon, \tag{65}$$

6.8 Numerical experiments

In this section we present some numerical studies of the speedup and error as a function of the number of data points, the bandwidth h, the order r, and the desired error ϵ. The algorithms were programmed in C++ and was run on a 1.6 GHz Pentium M processor with 512Mb of RAM.

Figure 2 shows the running time and the maximum absolute error relative to Q for both the direct and the fast methods as a function of $N = M$. The bandwidth was $h = 0.1$ and the order of the derivative was $r = 4$. The source and the target points were uniformly distributed in the unit interval. We see that the running time of the fast method grows linearly as the number of sources and targets increases, while that of the direct evaluation grows quadratically. We also observe that the error is way below the desired error thus validating our bound. However the bound is not very tight. Figure 3 shows the tradeoff between precision and speedup. An increase in speedup is obtained at the cost of reduced accuracy. Figure 4 shows the results

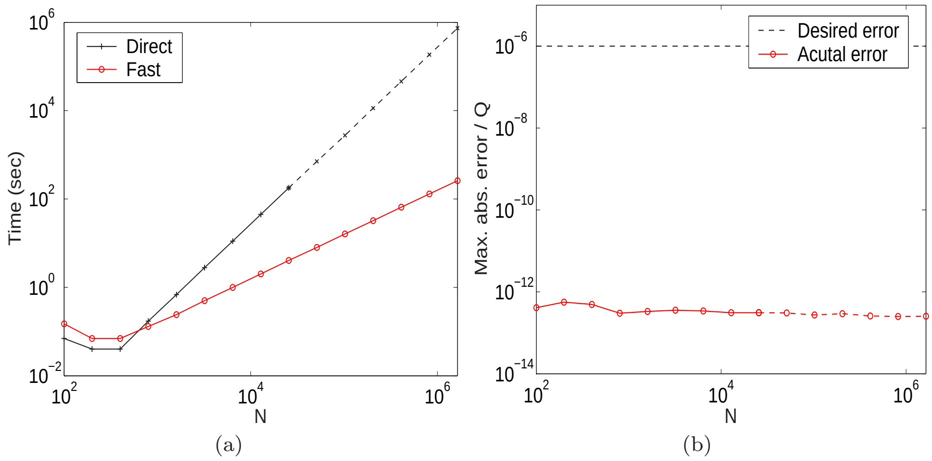


Fig. 2. (a) The running time in seconds and (b) maximum absolute error relative to Q for the direct and the fast methods as a function of N. $N = M$ source and the target points were uniformly distributed in the unit interval. For $N > 25600$ the timing results for the direct evaluation were obtained by evaluating the result at $M = 100$ points and then extrapolating. [$h = 0.1$, $r = 4$, and $\epsilon = 10^{-6}$.]

as a function of bandwidth h. Better speedup is obtained at larger bandwidths. Figure 5 shows the results for different orders of the density derivatives.

7. SPEEDUP ACHIEVED FOR BANDWIDTH ESTIMATION

The solve-the-equation plug-in method of [Jones et al. 1996] was implemented in MATLAB with the core computational task of computing the density derivative written in C++.

7.1 Synthetic data

We demonstrate the speedup achieved on the mixture of normal densities used by Marron and Wand [Marron and Wand 1992]. The family of normal mixture densities is extremely rich and, in fact any density can be approximated arbitrarily well by a member of this family. Fig. 6 shows the fifteen densities which were used by the authors in [Marron and Wand 1992] as a typical representative of the densities likely to be encountered in real data situations. We sampled $N = 50,000$ points from each density. The AMISE optimal bandwidth was estimated both using the direct methods and the proposed fast method. Table I shows the speedup achieved and the absolute relative error. Fig. 6 shows the actual density and the estimated density using the optimal bandwidth estimated using the fast method.

7.2 Real data

We used the Adult database from the UCI machine learning repository [Newman et al. 1998]. The database extracted from the census bureau database contains 32,561 training instances with 14 attributes per instance. Of the 14 attributes 6 are continuous and 8 nominal. Table II shows the speedup achieved and the absolute relative error for two of the continuous attributes.

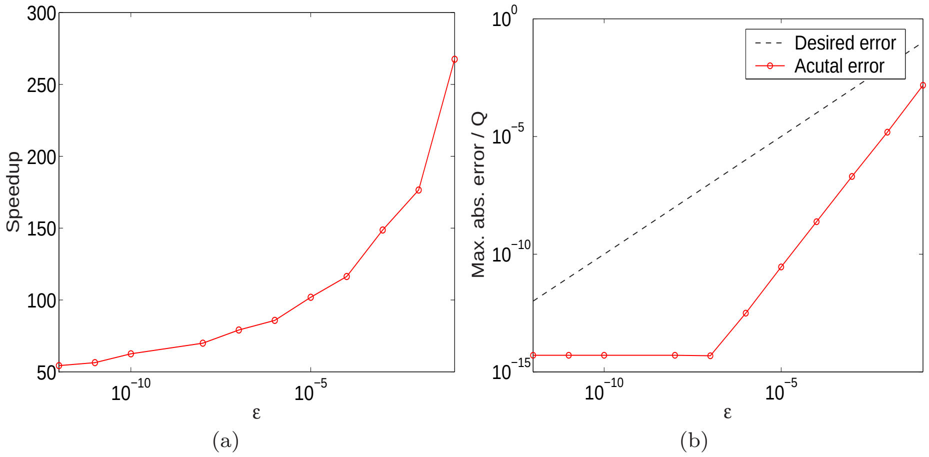


Fig. 3. (a) The speedup achieved and (b) maximum absolute error relative to Q for the direct and the fast methods as a function of ϵ. $N = M = 50,000$ source and the target points were uniformly distributed in the unit interval. $[h = 0.1$ and $r = 4]$

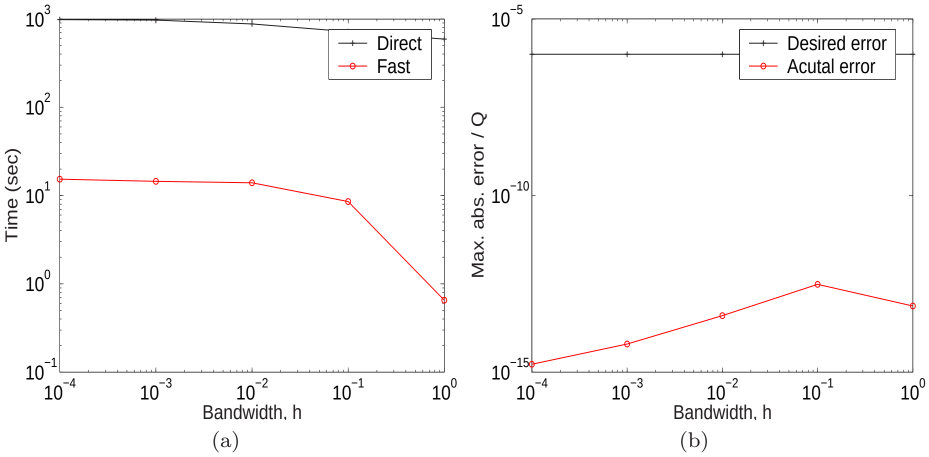
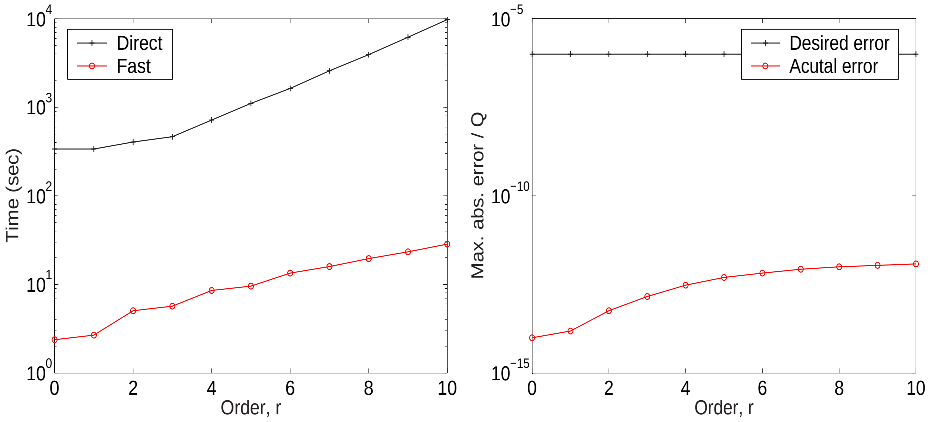


Fig. 4. (a) The running time in seconds and (b) maximum absolute error relative to Q for the direct and the fast methods as a function of h. $N = M = 50,000$ source and the target points were uniformly distributed in the unit interval. $[\epsilon = 10^{-6}$ and $r = 4]$

8. PROJECTION PURSUIT

Projection Pursuit (PP) is an exploratory technique for visualizing and analyzing large multivariate data-sets [Friedman and Tukey 1974; Huber 1985; Jones and Sibson 1987]. The idea of projection pursuit is to search for projections from high- to low-dimensional space that are most *interesting*. These projections can then be used for other nonparametric fitting and other data-analytic purposes The conventional dimension reduction techniques like principal component analysis looks for a projection that maximizes the variance. The idea of PP is to look for projections



(a) (b)

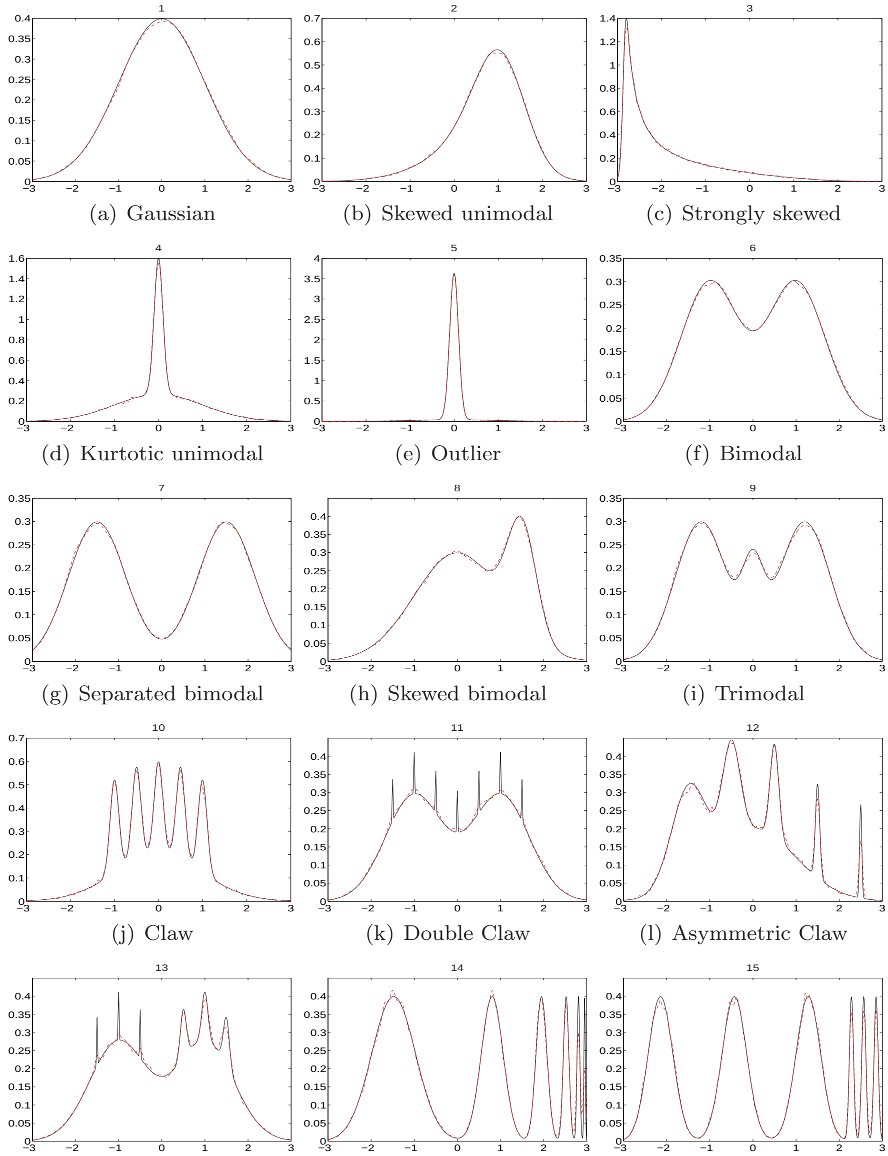
Fig. 5. (a) The running time in seconds and (b) maximum absolute error relative to Q for the direct and the fast methods as a function of r. $N = M = 50,000$ source and the target points were uniformly distributed in the unit interval. [$\epsilon = 10^{-6}$ and $h = 0.1$]

Table I. The bandwidth estimated using the solve-the-equation plug-in method for the fifteen normal mixture densities of Marron and Wand. h_{direct} and h_{fast} are the bandwidths estimated using the direct and the fast methods respectively. The running time in seconds for the direct and the fast methods are shown.The absolute relative error is defined as $|h_{direct} - h_{fast}/h_{direct}|$. In the study $N = 10,000$ points were sampled from the corresponding densities. For the fast method we used $\epsilon = 10^{-3}$.

Density	h_{direct}	h_{fast}	T_{direct} (sec)	T_{fast} (sec)	Speedup	Abs. Relative Error
1	0.122213	0.122215	4182.29	64.28	65.06	1.37e-005
2	0.082591	0.082592	5061.42	77.30	65.48	1.38e-005
3	0.020543	0.020543	8523.26	101.62	83.87	1.53e-006
4	0.020621	0.020621	7825.72	105.88	73.91	1.81e-006
5	0.012881	0.012881	6543.52	91.11	71.82	5.34e-006
6	0.098301	0.098303	5023.06	76.18	65.93	1.62e-005
7	0.092240	0.092240	5918.19	88.61	66.79	6.34e-006
8	0.074698	0.074699	5912.97	90.74	65.16	1.40e-005
9	0.081301	0.081302	6440.66	89.91	71.63	1.17e-005
10	0.024326	0.024326	7186.07	106.17	67.69	1.84e-006
11	0.086831	0.086832	5912.23	90.45	65.36	1.71e-005
12	0.032492	0.032493	8310.90	119.02	69.83	3.83e-006
13	0.045797	0.045797	6824.59	104.79	65.13	4.41e-006
14	0.027573	0.027573	10485.48	111.54	94.01	1.18e-006
15	0.023096	0.023096	11797.34	112.57	104.80	7.05e-007

that maximize other measures of interestingness, like non-normality, entropy etc. The PP algorithm for finding the most interesting one-dimensional subspace is as follows.

(1) Given N data points in a d dimensional space (centered and scaled), $\{x_i \in \mathbf{R}^d\}_{i=1}^N$, project each data point onto the direction vector $a \in \mathbf{R}^d$, i.e., $z_i = a^T x_i$.



(a) Gaussian (b) Skewed unimodal (c) Strongly skewed

(d) Kurtotic unimodal (e) Outlier (f) Bimodal

(g) Separated bimodal (h) Skewed bimodal (i) Trimodal

(j) Claw (k) Double Claw (l) Asymmetric Claw

(m) Asym. Double Claw (n) Smooth Comb (o) Discrete Comb

Fig. 6. The fifteen normal mixture densities of Marron and Wand. The solid line corresponds to the actual density while the dotted line is the estimated density using the optimal bandwidth estimated using the fast method.

Table II. Optimal bandwidth estimation for five continuous attributes for the Adult database from the UCI machine learning repository. The database contains 32561 training instances. The bandwidth was estimated using the solve-the-equation plug-in method. h_{direct} and h_{fast} are the bandwidths estimated using the direct and the fast methods respectively. The running time in seconds for the direct and the fast methods are shown. The absolute relative error is defined as $|h_{direct} - h_{fast}/h_{direct}|$. For the fast method we used $\epsilon = 10^{-3}$.

Attribute	h_{direct}	h_{fast}	T_{direct} (sec)	T_{fast} (sec)	Speedup	Error
Age	0.860846	0.860856	4679.03	66.42	70.45	1.17e-005
fnlwgt	4099.564359	4099.581141	4637.09	68.83	67.37	4.09e-006

(2) Compute the univariate nonparametric kernel density estimate, $\widehat{p}$, of the projected points z_i.

(3) Compute the projection index $I(a)$ based on the density estimate.

(4) Locally optimize over the the choice of a, to get the *most interesting* projection of the data.

(5) Repeat from a new initial projection to get a different view.

The projection index is designed to reveal specific structure in the data, like clusters, outliers, or smooth manifolds. Some of the commonly used projection indices are the Friedman-Tukey index [Friedman and Tukey 1974], the entropy index [Jones and Sibson 1987], and the moment index. The entropy index based on Rényi's order-1 entropy is given by

$$I(a) = \int p(z) \log p(z) dz. \tag{66}$$

The density of zero mean and unit variance which uniquely minimizes this is the standard normal density. Thus the projection index finds the direction which is most non-normal.In practice we need to use an estimate $\widehat{p}$ of the the true density p, for example the kernel density estimate using the Gaussian kernel. Thus we have an estimate of the entropy index as follows.

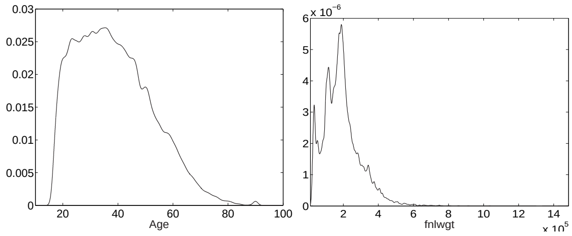
$$\begin{aligned} \widehat{I}(a) &= \int \log \widehat{p}(z) p(z) dz = E\left[\log \widehat{p}(z)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N} \log \widehat{p}(z_i) = \frac{1}{N}\sum_{i=1}^{N} \log \widehat{p}(a^T x_i). \end{aligned} \tag{67}$$

The entropy index $\widehat{I}(a)$ has to be optimized over the d-dimensional vector a subject to the constraint that $\|a\| = 1$. The optimization function will require the gradient of the objective function. For the index defined above the gradient can be written as

$$\frac{d}{da}[\widehat{I}(a)] = \frac{1}{N}\sum_{i=1}^{N} \frac{\widehat{p'}(a^T x_i)}{\widehat{p}(a^T x_i)} x_i. \tag{68}$$

For the PP the computational burden is greatly reduced if we use the proposed fast method. The computational burden is reduced in the following three instances.

(1) Computation of the kernel density estimate.



(a) Age (b) fnlwgt

Fig. 7. The estimated density using the optimal bandwidth estimated using the fast method, for two of the continuous attributes in the Adult database from the UCI machine learning repository.

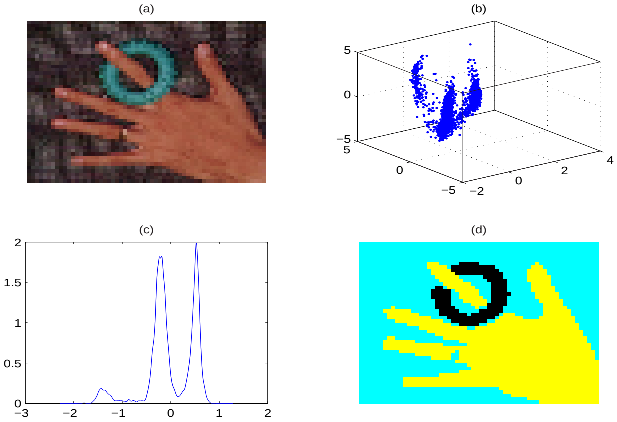


Fig. 8. (a) The original image. (b) The centered and scaled RGB space. Each pixel in the image is a point in the RGB space. (c) KDE of the projection of the pixels on the most interesting direction found by projection pursuit. (d) The assignment of the pixels to the three modes in the KDE.

(2) Estimation of the optimal bandwidth.

(3) Computation of the first derivative of the kernel density estimate, which is required in the optimization procedure.

Fig. 8 shows an example of the PP algorithm on a image. Fig. 8(a) shows the original image of the hand with a ring against a background. Perceptually the image has three distinct regions, the hand, the ring, and the background. Each pixel is represented as a point in a three dimensional RGB space. Fig. 8(b) shows the the presence of three clusters in the RGB space. We ran the PP algorithm on this space. Fig. 8(c) shows the KDE of the points projected on the most interesting direction. This direction is clearly able to distinguish the three clusters. Fig. 8(d) shows the segmentation where each pixel is assigned to the mode nearest to it.

9. CONCLUSIONS

We proposed an fast $\epsilon - exact$ algorithm for kernel density derivative estimation which reduced the computational complexity from $O(N^2)$ to $O(N)$. We demonstrated the speedup achieved for optimal bandwidth estimation both on simulated as well as real data. As an example we demonstrated how to potentially speedup the projection pursuit algorithm. We focussed on the univariate case in the current paper since the bandwidth selection procedures for the univariate case are pretty mature. Bandwidth selection for the multivariate case is a field of very active research [Wand and Jones 1994]. Our future work would include the relatively straightforward but more involved extension of the current procedure to handle higher dimensions. As pointed out earlier many applications other than bandwidth estimation require derivative estimates. We hope that our fast computation scheme should benefit all the related applications. The C++ code is available for academic use by contacting the first author.

10. APPENDIX 1 : MISE FOR KERNEL DENSITY ESTIMATORS

First note that MISE=IMSE.

$$\text{MISE}(\widehat{p}, p) = E\left[\int_{\mathbf{R}} [\widehat{p}(x) - p(x)]^2 dx\right] = \int_{\mathbf{R}} E[\widehat{p}(x) - p(x)]^2 dx = \text{IMSE}(\widehat{p}, p). \tag{69}$$

The mean square error (MSE) can be decomposed into variance and squared bias of the estimator.

$$\text{MSE}(\widehat{p}, p, x) = E[\widehat{p}(x) - p(x)]^2 = Var[\widehat{p}(x)] + (E[\widehat{p}(x)] - p(x))^2. \tag{70}$$

The kernel density estimate $\widehat{p}(x)$ is given by

$$\widehat{p}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x - x_i}{h}) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i),$$

where $K_h(x) = (1/h)K(x/h)$.

10.1 Bias

The mean of the estimator can be written as

$$E[\widehat{p}(x)] = \frac{1}{N}\sum_{i=1}^{N} E[K_h(x - x_i)] = E[K_h(x - X)] = \int_{\mathbf{R}} K_h(x - y)p(y)dy. \tag{71}$$

Using the convolution operator $*$ we have

$$E[\widehat{p}(x)] - p(x) = (K_h * p)(x) - p(x). \tag{72}$$

The bias is the difference between the smoothed version (using the kernel) of the density and the actual density.

10.2 Variance

The variance of the estimator can be written as

$$Var[\widehat{p}(x)] = \frac{1}{N} Var[K_h(x - X)] = \frac{1}{N}(E[K_h^2(x - X)] - E[K_h(x - X)]^2). \tag{73}$$

Using Eq. 71 we have the following expression for the variance.

$$Var[\widehat{p}(x)] = \frac{1}{N}[(K_h^2 * p)(x) - (K_h * p)^2(x)]. \tag{74}$$

10.3 MSE

Using Eq. 72 and Eq. 74 the MSE at a point x can be written as,

$$\mathrm{MSE}(\widehat{p}, p, x) \; = \; \frac{1}{N}\left[(K_h^2 * p)(x) - (K_h * p)^2(x)\right] + \left[(K_h * p)(x) - p(x)\right]^2 . \quad (75)$$

10.4 MISE

Since MISE=IMSE we have,

$$\mathrm{MISE}(\widehat{p}, p) \; = \; \frac{1}{N}\int_{\mathbf{R}}\left[(K_h^2 * p)(x) - (K_h * p)^2(x)\right] dx + \int_{\mathbf{R}}\left[(K_h * p)(x) - p(x)\right]^2 dx. \quad (76)$$

The dependence of the MISE on the bandwidth h is not very explicit in the above expression. This makes it difficult to interpret the influence of the bandwidth on the performance of the estimator. An asymptotic approximation for this expression is usually derived called as the AMISE.

11. APPENDIX 2 : ASYMPTOTIC MISE FOR KERNEL DENSITY ESTIMATORS

In order to derive an large sample approximation to MISE we make the following assumptions on the density p, the bandwidth h, and the kernel K.

(1) The second derivative $p^{''}(x)$ is continuous, square integrable and ultimately monotone [8].

(2) $\lim_{N\to\infty} h = 0$ and $\lim_{N\to\infty} Nh = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N$.

(3) In order that $\widehat{p}(x)$ is a valid density we assume $K(z) \geq 0$ and $\int_{\mathbf{R}} K(z)dz = 1$. The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2K(z)dz < \infty$) .

11.1 Bias

From Eq. 71 and a change of variables we have

$$E[\widehat{p}(x)] \; = \; (K_h * p)(x) = \int_{\mathbf{R}} K_h(x-y)p(y)dy = \int_{\mathbf{R}} K(z)p(x-hz)dz. \quad (77)$$

Using Taylor's series $p(x-hz)$ can be expanded as

$$p(x-hz) = p(x) - hzp^{'}(x) + \frac{1}{2}h^2z^2p^{''}(x) + o(h^2). \quad (78)$$

Hence

$$E[\widehat{p}(x)] \; = \; p(x)\int_{\mathbf{R}} K(z)dz - hp^{'}(x)\int_{\mathbf{R}} zK(z)dz + \frac{1}{2}h^2p^{''}(x)\int_{\mathbf{R}} z^2K(z)dz + o(h^2). \quad (79)$$

[8] An ultimately monotone function is one that is monotone over both $(-\infty, -M)$ and (M, ∞) for some $M > 0$.

From Assumption 3 we have,

$$\begin{aligned} \int_{\mathbf{R}} K(z)dz &= 1 \\ \int_{\mathbf{R}} zK(z)dz &= 0 \\ \mu_2(K) &= \int_{\mathbf{R}} z^2 K(z)dz < \infty \end{aligned} \quad (80)$$

Hence

$$E[\widehat{p}(x)] - p(x) = \frac{1}{2}h^2\mu_2(K)p^{''}(x) + o(h^2). \quad (81)$$

The KDE is asymptotically unbiased. The bias is directly proportional to the value of the second derivative of the density function, i.e., the curvature of the density function.

11.2 Variance

From Eq. 74 and a change of variables we have

$$\begin{aligned} Var[\widehat{p}(x)] &= \frac{1}{N}[(K_h^2 * p)(x) - (K_h * p)^2(x)] \\ &= \frac{1}{N}\left[\int_{\mathbf{R}} K_h^2(x-y)p(y)dy\right] - \frac{1}{N}\left[\int_{\mathbf{R}} K_h(x-y)p(y)dy\right]^2 \\ &= \frac{1}{Nh}\left[\int_{\mathbf{R}} K^2(z)p(x-hz)dz\right] - \frac{1}{N}\left[\int_{\mathbf{R}} K(z)p(x-hz)dz\right]^2 \end{aligned} \quad (82)$$

Using Taylor's series $p(x - hz)$ can be expanded as

$$p(x-hz) = p(x) + o(1). \quad (83)$$

We need only the first term because of the factor $1/N$. Hence

$$\begin{aligned} Var[\widehat{p}(x)] &= \frac{1}{Nh}\left[p(x) + o(1)\right]\int_{\mathbf{R}} K^2(z)dz - \frac{1}{N}\left[p(x) + o(1)\right]^2 \\ &= \frac{1}{Nh}p(x)\int_{\mathbf{R}} K^2(z)dz + o(1/Nh) \end{aligned} \quad (84)$$

Based on Assumption 2 $\lim_{N\to\infty} Nh = \infty$, the variable asymptotically converges to zero.

11.3 MSE

The MSE at a point x can be written as (using Eqs. 81 and 84),

$$\mathrm{MSE}(\widehat{p}, p, x) = \frac{1}{Nh}p(x)R(K) + \frac{1}{4}h^4\mu_2(K)^2 p^{''}(x)^2 + o(h^4 + 1/Nh). \quad (85)$$

where $R(K) = \int_{\mathbf{R}} K^2(z)dz$.

11.4 MISE

Since MISE=IMSE we have,

$$\begin{aligned}\text{MISE}(\widehat{p}, p) &= \frac{1}{Nh} R(K) \int_{\mathbf{R}} p(x)dx + \frac{1}{4} h^4 \mu_2(K)^2 \int_{\mathbf{R}} p^{''}(x)^2 dx + o(h^4 + 1/Nh) \\ &= \text{AMISE}(\widehat{p}, p) + o(h^4 + 1/Nh), \end{aligned} \tag{86}$$

where

$$\text{AMISE}(\widehat{p}, p) = \frac{1}{Nh} R(K) + \frac{1}{4} h^4 \mu_2(K)^2 R(p^{''}). \tag{87}$$

12. APPENDIX 3 : AMISE FOR KERNEL DENSITY DERIVATIVE ESTIMATORS

First note that MISE=IMSE.

$$\begin{aligned}\text{MISE}(\widehat{p}^{(r)}, p^{(r)}) &= E\left[\int_{\mathbf{R}} [\widehat{p}^{(r)}(x) - p^{(r)}(x)]^2 dx\right] \\ &= \int_{\mathbf{R}} E[\widehat{p}^{(r)}(x) - p^{(r)}(x)]^2 dx \\ &= \text{IMSE}(\widehat{p}^{(r)}, p^{(r)}). \end{aligned} \tag{88}$$

The mean square error (MSE) can be decomposed into variance and squared bias of the estimator.

$$\begin{aligned}\text{MSE}(\widehat{p}^{(r)}, p^{(r)}, x) &= E[\widehat{p}^{(r)}(x) - p^{(r)}(x)]^2 \\ &= \text{Var}[\widehat{p}^{(r)}(x)] + (E[\widehat{p}^{(r)}(x)] - p^{(r)}(x))^2. \end{aligned} \tag{89}$$

An simple estimator for the density derivative can be obtained by taking the derivative of the kernel density estimate $\widehat{p}(x)$ [Bhattacharya 1967; Schuster 1969]. If the kernel K is differentiable r times then the r^{th} density derivative estimate $\widehat{p}^{(r)}(x)$ can be written as

$$\begin{aligned}\widehat{p}^{(r)}(x) &= \frac{1}{Nh^{r+1}} \sum_{i=1}^{N} K^{(r)}\left(\frac{x - x_i}{h}\right) \\ &= \frac{1}{N} \sum_{i=1}^{N} K_h^{(r)}(x - x_i) \end{aligned} \tag{90}$$

where $K^{(r)}$ is the r^{th} derivative of the kernel K and $K_h^{(r)}(x) = (1/h^{r+1}) K^{(r)}(x/h)$.

In order to derive an large sample approximation to MISE we make the following assumptions on the density p, the bandwidth h, and the kernel K.

(1) The $(r+2)^{th}$ derivative $p^{(r+2)}(x)$ is continuous, square integrable and ultimately monotone [9].

(2) $\lim_{N\to\infty} h = 0$ and $\lim_{N\to\infty} Nh^{2r+1} = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N^{2r+1}$.

[9] An ultimately monotone function is one that is monotone over both $(-\infty, -M)$ and (M, ∞) for some $M > 0$.

(3) In order that $\widehat{p}(x)$ is a valid density we assume $K(z) \geq 0$ and $\int_{\mathbf{R}} K(z)dz = 1$. The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2K(z)dz < \infty$) .

12.1 Bias

The mean of the estimator can be written as

$$\begin{aligned} E[\widehat{p}^{(r)}(x)] &= \frac{1}{N}\sum_{i=1}^{N} E[K_h^{(r)}(x-x_i)] \\ &= E[K_h^{(r)}(x-X)] \\ &= \int_{\mathbf{R}} K_h^{(r)}(x-y)p(y)dy. \end{aligned} \tag{91}$$

Using the convolution operator $*$ we have

$$E[\widehat{p}^{(r)}(x)] = (K_h^{(r)} * p)(x) = (K_h * p^{(r)})(x). \tag{92}$$

where we have used the relation $K_h^{(r)} * p = K_h * p^{(r)}$. We now derive a large sample approximation to the mean. Using a change of variables the mean can be written as follows.

$$\begin{aligned} E[\widehat{p}^{(r)}(x)] &= (K_h * p^{(r)})(x) = \int_{\mathbf{R}} K_h(x-y)p^{(r)}(y)dy \\ &= \int_{\mathbf{R}} K(z)p^{(r)}(x-hz)dz. \end{aligned} \tag{93}$$

Using Taylor's series $p^{(r)}(x-hz)$ can be expanded as

$$p^{(r)}(x-hz) = p^{(r)}(x) - hzp^{(r+1)}(x) + \frac{1}{2}h^2z^2p^{(r+2)}(x) + o(h^2). \tag{94}$$

Hence

$$\begin{aligned} E[\widehat{p}^{(r)}(x)] &= p^{(r)}(x)\left[\int_{\mathbf{R}} K(z)dz\right] - hp^{(r+1)}(x)\left[\int_{\mathbf{R}} zK(z)dz\right] \\ &+\frac{1}{2}h^2p^{(r+2)}(x)\left[\int_{\mathbf{R}} z^2K(z)dz\right] + o(h^2). \end{aligned} \tag{95}$$

From Assumption 3 we have,

$$\begin{aligned} \int_{\mathbf{R}} K(z)dz &= 1 \\ \int_{\mathbf{R}} zK(z)dz &= 0 \\ \mu_2(K) &= \int_{\mathbf{R}} z^2K(z)dz < \infty \end{aligned} \tag{96}$$

Hence the bias can be written as

$$E[\widehat{p}^{(r)}(x)] - p^{(r)}(x) = \frac{1}{2}h^2\mu_2(K)p^{(r+2)}(x) + o(h^2). \tag{97}$$

The estimate is asymptotically unbiased. The bias is estimating the r^{th} derivative is directly proportional to the value of the $(r+2)^{th}$ derivative of the density function.

12.2 Variance

The variance of the estimator can be written as

$$\begin{aligned} Var[\widehat{p}^{(r)}(x)] &= \frac{1}{N} Var[K_h^{(r)}(x-X)] \\ &= \frac{1}{N}(E[K_h^{(r)}(x-X)^2] - E[K_h^{(r)}(x-X)]^2). \end{aligned} \tag{98}$$

Using Eq. 91 we have the following expression for the variance.

$$Var[\widehat{p}^{(r)}(x)] = \frac{1}{N}\left[\left(K_h^{(r)} * p\right)(x)^2 - \left(K_h^{(r)} * p\right)^2(x)\right]. \tag{99}$$

Using a change of variables we have

$$\begin{aligned} Var[\widehat{p}^{(r)}(x)] &= \frac{1}{N}\left[\int_{\mathbf{R}} K_h^{(r)}(x-y)^2 p(y)dy\right] - \frac{1}{N}\left[\int_{\mathbf{R}} K_h^{(r)}(x-y)p(y)dy\right]^2 \\ &= \frac{1}{Nh^{2r+1}}\left[\int_{\mathbf{R}} K^{(r)}(z)^2 p(x-hz)dz\right] - \frac{1}{Nh^{2r}}\left[\int_{\mathbf{R}} K^{(r)}(z)p(x-hz)dz\right]^2. \end{aligned} \tag{100}$$

Using Taylor's series $p(x-hz)$ can be expanded as

$$p(x-hz) = p(x) + o(1). \tag{101}$$

We need only the first term because of the factor $1/N$. Hence

$$\begin{aligned} Var[\widehat{p}^{(r)}(x)] &= \frac{1}{Nh^{2r+1}}\left[p(x)+o(1)\right]\int_{\mathbf{R}} K^{(r)}(z)^2 dz - \frac{1}{Nh^{2r}}\left[p(x)+o(1)\right]^2\left[\int_{\mathbf{R}} K^{(r)}(z)\right]^2 dz \\ &= \frac{1}{Nh^{2r+1}} p(x)\int_{\mathbf{R}} K^{(r)}(z)^2 dz + o(1/Nh^{2r+1}). \end{aligned} \tag{102}$$

Based on Assumption 2 $\lim_{N\to\infty} Nh^{2r+1} = \infty$, the variable asymptotically converges to zero.

12.3 MSE

The MSE at a point x can be written as (using Eqs. 97 and 102),

$$\begin{aligned} \text{MSE}(\widehat{p}^{(r)}, p^{(r)}, x) &= \frac{1}{Nh^{2r+1}} p(x)R(K^{(r)}) + \frac{1}{4}h^4\mu_2(K)^2 p^{(r+2)}(x)^2 \\ &+ o(h^4 + 1/Nh^{2r+1}). \end{aligned} \tag{103}$$

where $R(K^{(r)}) = \int_{\mathbf{R}} K^{(r)}(z)^2 dz$.

12.4 MISE

Since MISE=IMSE we have,

$$\begin{aligned} \text{MISE}(\widehat{p}^{(r)}, p^{(r)}) &= \frac{1}{Nh^{2r+1}} R(K^{(r)})\int_{\mathbf{R}} p(x)dx + \frac{1}{4}h^4\mu_2(K)^2\int_{\mathbf{R}} p^{(r+2)}(x)^2 dx \\ &+ o(h^4 + 1/Nh^{2r+1}) \\ &= \text{AMISE}(\widehat{p}^{(r)}, p^{(r)}) + o(h^4 + 1/Nh^{2r+1}) \end{aligned} \tag{104}$$

where

$$\text{AMISE}(\widehat{p}^{(r)}, p^{(r)}) \;=\; \frac{1}{Nh^{2r+1}} R(K^{(r)}) + \frac{1}{4} h^4 \mu_2(K)^2 R(p^{(r+2)}). \qquad (105)$$

13. APPENDIX 4 : ASYMPTOTIC MSE FOR DENSITY FUNCTIONAL ESTIMATORS

We want to estimate the density functional Φ_r.

$$\Phi_r = \int_{\mathbf{R}} p^{(r)}(x)p(x)dx = E\left[p^{(r)}(X)\right]. \qquad (106)$$

An estimator for Φ_r is,

$$\widehat{\Phi}_r = \frac{1}{N}\sum_{i=1}^{N} \widehat{p}^{(r)}(x_i). \qquad (107)$$

where $\widehat{p}^{(r)}(x_i)$ is the estimate of the r^{th} derivative of the density $p(x)$ as $x = x_i$. Using a kernel density derivative estimate for $\widehat{p}^{(r)}(x_i)$ (Eq. 12) we have

$$\begin{aligned}\widehat{\Phi}_r &= \frac{1}{N^2h^{r+1}} \sum_{i=1}^{N}\sum_{j=1}^{N} K^{(r)}(\frac{x_i - x_j}{h}) \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_h^{(r)}(x_i - x_j). \qquad (108)\end{aligned}$$

The mean square error (MSE) can be decomposed into variance and squared bias of the estimator.

$$\text{MSE}(\widehat{\Phi}_r, \Phi_r) = E[\widehat{\Phi}_r - \Phi_r]^2 = Var[\widehat{\Phi}_r] + (E[\widehat{\Phi}_r] - \Phi_r)^2. \qquad (109)$$

In order to derive an large sample approximation to MSE we make the following assumptions on the density p, the bandwidth h, and the kernel K.

(1) The density p had $k > 2$ continuous derivatives which are ultimately monotone. The $(r{+}2)^{th}$ derivative $p^{(r+2)}(x)$ is continuous, square integrable and ultimately monotone [10].

(2) $\lim_{N\to\infty} h = 0$ and $\lim_{N\to\infty} Nh^{2r+1} = \infty$, i.e., as the number of samples N is increased h approaches zero at a rate slower than $1/N^{2r+1}$.

(3) In order that $\widehat{p}(x)$ is a valid density we assume $K(z) \geq 0$ and $\int_{\mathbf{R}} K(z)dz = 1$. The kernel function is assumed to be symmetric about the origin ($\int_{\mathbf{R}} zK(z)dz = 0$) and has finite second moment ($\int_{\mathbf{R}} z^2K(z)dz < \infty$) .

We write $\widehat{\Phi}_r$ as follows

$$\begin{aligned}\widehat{\Phi}_r &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} K_h^{(r)}(x_i - x_j) \\ &= \frac{1}{N}K_h^{(r)}(0) + \frac{1}{N^2}\sum\sum_{i\neq j} K_h^{(r)}(x_i - x_j). \qquad (110)\end{aligned}$$

[10] An ultimately monotone function is one that is monotone over both $(-\infty, -M)$ and (M, ∞) for some $M > 0$.

The first term is a constant independent of the data.

13.1 Bias

The expected value of the estimator can be written as

$$E[\widehat{\Phi}_r] = \frac{1}{N}K_h^{(r)}(0) + \left(1 - \frac{1}{N}\right) E[K_h^{(r)}(X_1 - X_2)]. \tag{111}$$

The term $E[K_h^{(r)}(X_1 - X_2)]$ can be simplified as follows

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \int\int K_h^{(r)}(x-y)p(x)p(y)dxdy$$

Using the relation $K_h^{(r)} * p = K_h * p^{(r)}$ we have

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \int\int K_h(x-y)p(x)p^{(r)}(y)dxdy \tag{112}$$

By a change of variables we have

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \int\int K(u)p(y+hu)p^{(r)}(y)dudy \tag{113}$$

Using Taylor's series $p(y+hu)$ can be expanded as

$$p(y+hu) = p(y) + hup^{'}(y) + \frac{1}{2}h^2u^2p^{''}(y) + O(h^3). \tag{114}$$

Hence

$$\begin{aligned} E[K_h^{(r)}(X_1 - X_2)] \;=&\; \left(\int K(u)du\right)\left(\int p^{(r)}(y)p(y)dy\right) \\ &+\; h\left(\int uK(u)du\right)\left(\int p^{(r)}(y)p^{'}(y)dy\right) \\ &+\; \frac{1}{2}h^2\left(\int u^2K(u)du\right)\left(\int p^{(r)}(y)p^{''}(y)dy\right) + O(h^3). \end{aligned}$$

From Assumption 3 we have,

$$\begin{aligned} \int K(u)du \;&=\; 1 \\ \int uK(u)du \;&=\; 0 \\ \mu_2(K) \;&=\; \int u^2K(u)du < \infty \end{aligned} \tag{115}$$

Substituting we have

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \Phi_r + \frac{1}{2}h^2\mu_2(K)\left(\int p^{(r)}(y)p^{''}(y)dy\right) + O(h^3). \tag{116}$$

Using the assumption that the density derivatives are ultimately monotone this can be simplifies using integration by parts as follows.

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \Phi_r + \frac{1}{2}h^2\mu_2(K)\Phi_{r+2} + O(h^3).$$

Hence substituting in Eq. 111 the bias of the estimator can be written as

$$E[\widehat{\Phi}_r] - \Phi_r \;=\; \frac{1}{Nh^{r+1}}K^{(r)}(0) + \frac{1}{2}h^2\mu_2(K)\Phi_{r+2} + O(h^3) - \frac{1}{N}\Phi_r - \frac{1}{2N}h^2\mu_2(K)\Phi_{r+2}. \tag{117}$$

The bias after ignoring the $1/N$ terms can be written as

$$E[\widehat{\Phi}_r] - \Phi_r \;=\; \frac{1}{Nh^{r+1}}K^{(r)}(0) + \frac{1}{2}h^2\mu_2(K)\Phi_{r+2} + O(h^3).$$

13.2 Variance

If r is even then the variance can be shown to be

$$\begin{aligned} Var[\widehat{\Phi}_r] \;&=\; \frac{2(N-1)}{N^3}Var[K_h^{(r)}(X_1 - X_2)] \\ &\quad + \frac{4(N-1)(N-2)}{N^3}Cov[K_h^{(r)}(X_1 - X_2), K_h^{(r)}(X_2 - X_3)]. \end{aligned} \tag{118}$$

First we will compute

$$E[K_h^{(r)}(X_1 - X_2)^2] \;=\; \int\int K_h^{(r)}(x-y)^2 p(x)p(y)dxdy$$

Using a change of variables we have

$$E[K_h^{(r)}(X_1 - X_2)^2] \;=\; \frac{1}{h^{2r+1}}\int\int K^{(r)}(u)^2 p(y+hu)p(y)dudy$$

Using Taylor's series $p(y+hu)$ can be expanded as

$$p(y+hu) = p(y) + o(1). \tag{119}$$

Hence

$$E[K_h^{(r)}(X_1 - X_2)^2] \;=\; \frac{1}{h^{2r+1}}\Phi_0 R(K^{(r)}) + o(1/h^{2r+1}).$$

Also we have

$$E[K_h^{(r)}(X_1 - X_2)] \;=\; \Phi_r + o(1). \tag{120}$$

From the above two equations the variance can be written as

$$\begin{aligned} Var[K_h^{(r)}(X_1 - X_2)] \;&=\; E[K_h^{(r)}(X_1 - X_2)^2] - E[K_h^{(r)}(X_1 - X_2)]^2 \\ &=\; \frac{1}{h^{2r+1}}\Phi_0 R(K^{(r)}) - \Phi_r^2 + o(1/h^{2r+1}). \end{aligned}$$

The covariance term can be written as

$$\begin{aligned} Cov[K_h^{(r)}(X_1 - X_2), K_h^{(r)}(X_2 - X_3)] \;&=\; E[K_h^{(r)}(X_1 - X_2)K_h^{(r)}(X_2 - X_3)] \\ &\quad - E[K_h^{(r)}(X_1 - X_2)]E[K_h^{(r)}(X_2 - X_3)] \end{aligned}$$

The first term can be simplified as follows

$$\begin{aligned} & E[K_h^{(r)}(X_1 - X_2)K_h^{(r)}(X_2 - X_3)] \\ = & \int\int\int K_h^{(r)}(x-y)K_h^{(r)}(y-z)p(x)p(y)p(z)\,dx\,dy\,dz \\ = & \int\int\int K_h(x-y)K_h(y-z)p^{(r)}(x)p(y)p^{(r)}(z)\,dx\,dy\,dz \\ = & \int\int\int K(u)K(v)p^{(r)}(y+hu)p(y)p^{(r)}(y-hv)\,du\,dv\,dy \\ = & \int p^{(r)}(y)^2 p(y)dy + o(1). \end{aligned}$$

Hence

$$Cov[K_h^{(r)}(X_1 - X_2), K_h^{(r)}(X_2 - X_3)] \;=\; \int p^{(r)}(y)^2 p(y)dy - \Phi_r^2 + o(1).$$

Using these approximations the variance can be written as

$$\begin{aligned} Var[\widehat{\Phi}_r] \;=\; & \frac{2}{N^2h^{2r+1}}\Phi_0 R(K^{(r)}) + \frac{4}{N}\left[\int p^{(r)}(y)^2 p(y)dy - \Phi_r^2\right] \\ & + o(1/N^2h^{2r+1} + 1/N). \end{aligned} \tag{121}$$

13.3 MSE

The asymptotic MSE can be written as

$$\begin{aligned} \text{MSE}(\widehat{\Phi}_r, \Phi_r) \;=\; & E[\widehat{\Phi}_r - \Phi_r]^2 \\ =\; & Var[\widehat{\Phi}_r] + (E[\widehat{\Phi}_r] - \Phi_r)^2 \\ =\; & \left[\frac{1}{Nh^{r+1}}K^{(r)}(0) + \frac{1}{2}h^2\mu_2(K)\Phi_{r+2}\right]^2 \\ & + \frac{2}{N^2h^{2r+1}}\Phi_0 R(K^{(r)}) + \frac{4}{N}\left[\int p^{(r)}(y)^2 p(y)dy - \Phi_r^2\right] \\ & + O(h^6) + o(1/N^2h^{2r+1} + 1/N). \end{aligned} \tag{122}$$

REFERENCES

Abramowitz, M. and Stegun, I. A. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, Chapter Orthogonal Polynomials, 771–802.

Baxter, B. J. C. and Roussos, G. 2002. A new error estimate of the fast gauss transform. *SIAM J. Sci. Stat. Comput. 24,* 1, 257–259.

Bhattacharya, P. K. 1967. Estimation of a probability density function and its derivatives. *Sankhya, Series A 29*, 373–382.

Devroye, L. and Lugosi, G. 2000. *Combinatorial Methods in Density Estimation.* Springer-Verlag.

Friedman, J, H. and Tukey, J. W. 1974. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput. 23*, 881–889.

Fukunaga, K. and Hostetler, L. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Info. Theory 21,* 1, 32–40.

Gray, A. and Moore, A. 2001. N-body problems in statistical learning. In *Advances in Neural Information Processing Systems.* 521–527.

GRAY, A. G. AND MOORE, A. W. 2003. Nonparametric density estimation: Toward computational tractability. In *SIAM International conference on Data Mining.*

GREENGARD, L. AND STRAIN, J. 1991. The fast Gauss transform. *SIAM J. Sci. Stat. Comput. 12,* 1, 79–94.

HUBER, P. J. 1985. Projection pursuit. *The Annals of Statistics 13*, 435–475.

IZENMAN, A. J. 1991. Recent developments in nonparametric density estimation. *J. Amer. Stat. Assoc. 86,* 413, 205–224.

JONES, M. C., MARRON, J. S., AND SHEATHER, S. J. 1996. A brief survey of bandwidth selection for density estimation. *J. Amer. Stat. Assoc. 91,* 433 (March), 401–407.

JONES, M. C. AND SIBSON, R. 1987. What is projection pursuit? *J. R. Statist. Soc. A 150*, 1–36.

MARRON, J. S. AND WAND, M. P. 1992. Exact mean integrated squared error. *The Ann. of Stat. 20,* 2, 712–736.

NEWMAN, D. J., HETTICH, S., BLAKE, C. L., AND MERZ, C. J. 1998. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

PARZEN, E. 1962. On estimation of a probability density function and mode. *Ann. Math. Statist. 33,* 3, 1065–1076.

SCHUSTER, E. F. 1969. Estimation of a probability density function and its derivatives. *The Annals of Mathematical Statistics 40,* 4 (August), 1187–1195.

SHEATHER, S. AND JONES, M. 1991. A reliable data-based bandwidth selection method for kernel density estimation. *J. Royal Statist. Soc. B 53*, 683–690.

SILVERMAN, B. W. 1982. Algorithm AS 176: Kernel density estimation using the fast Fourier transform. *Journal of Royal Statistical society Series C: Applied statistics 31,* 1, 93–99.

SINGH, R. S. 1977a. Applications of estimators of a density and its derivatives to certain statistical problems. *Journal of the Royal Statistical Society. Series B (Methodological) 39,* 3, 357–363.

SINGH, R. S. 1977b. Improvement on some known nonparametric uniformly consistent estimators of derivatives of a density. *The Annals of Statistics 5,* 2 (March), 394–399.

WAND, M. P. AND JONES, M. C. 1994. Multivariate plug-in bandwidth selection. *Computational Statistics 9*, 97–117.

WAND, M. P. AND JONES, M. C. 1995. *Kernel Smoothing.* Chapman and Hall.

YANG, C., DURAISWAMI, R., AND DAVIS, L. 2005. Efficient kernel machines using the improved fast Gauss transform. In *Advances in Neural Information Processing Systems.* 1561–1568.

YANG, C., DURAISWAMI, R., AND GUMEROV, N. 2003. Improved fast Gauss transform. Tech. Rep. CS-TR-4495, Dept. of Computer Science, University of Maryland, College Park.

YANG, C., DURAISWAMI, R., GUMEROV, N., AND DAVIS, L. 2003. Improved fast Gauss transform and efficient kernel density estimation. In *IEEE Int. Conf. on Computer Vision.* 464–471.