Lec 8: KDE extensions

Weiping Zhang

2019.10.21

Outline

Density Derivatives

Kernel CDF estimation

Adaptive KDE

Boundary Correction

Higher-order kernels

Computation Aspect

Density Derivatives

- Consider the problem of estimating the rth derivative of the density

$$f^{(r)}(x) = \frac{d^r}{dx^r} f(x)$$

- Since the kernel density estimator is

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

- A natural estimator is found by taking derivatives of the kernel density estimator. This takes the form

$$\hat{f}_h^{(r)}(x) = \frac{1}{nh^{r+1}} \sum_{i=1}^{n} K^{(r)}(\frac{x - X_i}{h})$$

where $K^{(r)}$ is the rth order derivative of kernel K.

- This estimator only makes sense if $K^{(r)}(x)$ exists and is non-zero.
- Suppose the kernel K satisfy the previous assumptions, with additionally, $K^{(s)}(\infty) = 0, K^{(s)}(-\infty) = 0, s = 0, 1, 2, \ldots, r$.
- Notice that $K^{(r)}(\frac{X_i - x}{h}), i = 1, 2, \ldots, n$ are i.i.d variables,

$$
\begin{aligned}
Ef_h^{(r)}(x) &= \frac{1}{h^{r+1}} EK^{(r)}(\frac{x - X_1}{h}) \\
&= \frac{1}{h^r} \int K^{(r)}(u) f(x - uh) du = \frac{1}{h^r} \int f(x - uh) dK^{(r-1)}(u) \\
&= \frac{1}{h^r} K^{(r-1)}(u) f(x - uh)|_{-\infty}^{+\infty} - \frac{1}{h^r} \int K^{(r-1)}(u) df(x - uh) \\
&= \frac{1}{h^{r-1}} \int K^{(r-1)}(u) f'(x - uh) du \\
&= \frac{1}{h^{r-1}} \int K^{(r-1)}(\frac{x - z}{h}) f'(z) dz
\end{aligned}
$$

Repeating this a total of r times, we obtain

$$
\begin{aligned}
Ef_h^{(r)}(x) &= \frac{1}{h}\int K(\frac{x-z}{h})f^{(r)}(z)dz \\
&= \frac{1}{h}\int K(u)f^{(r)}(x-uh)du \\
&= \frac{1}{h}\int K(u)[f^{(r)}(x) - f^{(r+1)}(x)uh + \frac{1}{2}f^{(r+1)}(x)u^2h^2 + o(h^2)]du \\
&= f^{(r)}(x) + \frac{1}{2}f^{(r+2)}(x)\kappa_{21}h^2 + o(h^2)
\end{aligned}
$$

Thus, the bias of $\hat{f}_h^{(r)}(x)$ is

$$bias(\hat{f}_h^{(r)}(x)) = \frac{1}{2}f^{(r+2)}(x)\kappa_{21}h^2 + o(h^2)$$

For the variance, we find

$$\begin{aligned}
Var(f_h^{(r)}(x)) &= \frac{1}{n} Var(\frac{1}{h^{r+1}} K^{(r)}(\frac{x - X_1}{h})) \\
&= \frac{1}{n} E\Big[\frac{1}{h^{r+1}} K^{(r)}(\frac{x - X_1}{h})\Big]^2 - \frac{1}{n}\Big\{\frac{1}{h^{r+1}} E K^{(r)}(\frac{x - X_1}{h})\Big\}^2 \\
&= I_1 + I_2,
\end{aligned}$$

where the first term I_1,

$$\begin{aligned}
I_1 &= \frac{1}{n} E\Big[\frac{1}{h^{r+1}} K^{(r)}(\frac{x - X_1}{h})\Big]^2 = \frac{1}{nh^{2(r+1)}} \int [K^{(r)}(u)]^2 f(x - uh) du \\
&= \frac{f(x)}{nh^{2r+1}} \int [K^{(r)}(u)]^2 du + o(\frac{1}{nh^{2r+1}}).
\end{aligned}$$

Since

$$
\begin{aligned}
I_2 &= \frac{1}{n}\Big\{\frac{1}{h^{r+1}}EK^{(r)}(\frac{x-X_1}{h})\Big\}^2 \\
&= \frac{1}{n}[f^{(r)}(x)+\frac{1}{2}f^{(r+1)}(x)\kappa_{21}h^2+o(h^2)] \\
&= o(\frac{1}{nh^{2r+1}}).
\end{aligned}
$$

We have

$$
Var[(f_h^{(r)}(x)] = \frac{f(x)}{nh^{2r+1}}\int[K^{(r)}(u)]^2du + o(\frac{1}{nh^{2r+1}})
$$

and

$$
\begin{aligned}
MSE[(f_h^{(r)}(x)] = \frac{f(x)}{nh^{2r+1}}\int[K^{(r)}(u)]^2du + \frac{1}{4}[f^{(r+2)}(x)]^2\kappa_{21}^2h^4 \\
+ o(\frac{1}{nh^{2r+1}}) + o(h^4)
\end{aligned}
$$

Therefore, the MISE is

$$
\begin{aligned}
MISE[(f_h^{(r)}(x)] &= \int MSE[(f_h^{(r)}(x)]dx \\
&= \underbrace{\frac{1}{nh^{2r+1}} \int [K^{(r)}(u)]^2 du + \frac{1}{4} \int [f^{(r+2)}(x)]^2 dx \kappa_{21}^2 h^4}_{AMISE_r(h)} \\
&+ o(\frac{1}{nh^{2r+1}}) + o(h^4)
\end{aligned}
$$

The optimal bandwidth h_{opt} can be obtained by minimizing AMISE,

$$
\begin{aligned}
h_{opt} &= \arg\min AMISE_r(h) \\
&= \Big[\frac{(2r+1)\|K^{(r)}\|^2}{\|f^{(r+2)}\|^2 \kappa_{21}^2}\Big]^{1/(2r+5)} n^{-1/(2r+5)}
\end{aligned}
$$

With optimal h_{opt}, it is easily seen that

$$AMISE_r(h_{opt}) = O(n^{-4/(2r+5)})$$

- $r = 0$, $AMISE_0(h_{opt}) = O(n^{-4/5})$
- $r = 1$, $AMISE_1(h_{opt}) = O(n^{-4/7})$
- $r = 2$, $AMISE_2(h_{opt}) = O(n^{-4/9})$

To achieve a specific convergence rate for the AMISE, the sample size needs to be increase accordingly as the order r increases.

- We can also ask the question of which kernel function is optimal, and this is addressed by Muller (1984).
- His conclusion is that it is optimal to use a member of the Biweight class for a first derivative and a member of the Triweight for for a second derivative, while the Gaussian kernel is highly inefficient.
- The calculations suggest that when estimating density derivatives it is important to use the appropriate kernel.

Kernel CDF estimation

- Let $X \sim F$ with pdf f, since the empirical cumulative distribution function $\hat{F}_n$ is discontinuous, our aim is at finding a continuous estimator of F.
- From the KDE of f, a direct estimator of F is

$$\hat{F}_h(x) = \int_{-\infty}^{x} \hat{f}_h(u)du = \frac{1}{n}\sum_{i=1}^{n} G(\frac{x - X_i}{h})$$

where $G(x) = \int_{-\infty}^{x} K(z)dz$.

- Mean:

$$
\begin{aligned}
E[\hat{F}_h(x)] &= EG(\frac{x - X_1}{h}) \\
&= h \int G(u) f(x - uh) du = - \int G(u) dF(x - uh) \\
&= -G(u)F(x - uh)|_{-\infty}^{\infty} + \int F(x - uh) K(u) du \\
&= \int [F(x) - uhf(x) + \frac{1}{2} h^2 u^2 F^{(2)}(x)] K(u) du + o(h^2) \\
&= F(x) + \frac{1}{2} h^2 \kappa_{21} F^{(2)}(x) + o(h^2)
\end{aligned}
$$

Thus, the bias of $\hat{F}_h(x)$ is

$$
bias(\hat{F}_h(x)) = \frac{1}{2} h^2 \kappa_{21} F^{(2)}(x) + o(h^2)
$$

- Variance: Since

$$
\begin{aligned}
E[G(\frac{x-X_1}{h})]^2 &= h\int G^2(u)f(x-uh)du = -\int G^2(u)dF(x-uh) \\
&= -G^2(u)F(x-uh)|_{-\infty}^{\infty} + 2\int F(x-uh)G(u)K(u)du \\
&= 2\int [F(x)-uhf(x)]G(u)K(u)du + o(h) \\
&= F(x) - 2hf(x)D_1 + o(h)
\end{aligned}
$$

where the last step uses the fact that $\int G(u)K(u)du = 0.5$ and $D_1 = \int uG(u)K(u)du$.

we have

$$
\begin{aligned}
Var[\hat{F}_h(x)] &= \frac{1}{n} Var[G(\frac{x-X_1}{h})] \\
&= \frac{1}{n} E[G(\frac{x-X_1}{h})]^2 - \frac{1}{n}[EG(\frac{x-X_1}{h})]^2 \\
&= \frac{1}{n}[F(x) - 2hf(x)D_1] - \frac{1}{n}[F(x) + \frac{1}{2}h^2\kappa_{21}F^{(2)}(x)]^2 + o(\frac{h}{n}) \\
&= \frac{1}{n}F(x)(1-F(x)) - \frac{2h}{n}f(x)D_1 + o(\frac{h}{n}).
\end{aligned}
$$

Therefore,

$$
MSE[\hat{F}_h(x)] = \frac{1}{n}F(x)(1-F(x)) + h^4 C_1(x) + \frac{h}{n}C_2(x) + o(\frac{h}{n} + h^4)
$$

where $C_1(x) = \frac{1}{4}\kappa_{21}^2[F^{(2)}(x)]^2, C_2(x) = -2f(x)D_1$.

We then have the MISE:

$$MISE(h) = \frac{1}{n}\int F(x)(1 - F(x))dx + h^4 \int C_1(x)dx + \frac{h}{n}\int C_2(x)dx$$
$$+ o(h^4) + o(\frac{h}{n}).$$

The optimal bandwidth is

$$h_{opt} = \Big[\frac{\int C_2(x)dx}{4\int C_1(x)dx}\Big]^{1/3} n^{-1/3}$$

Since h_{opt} is not applicable in practice as the unknown integrants of C_1 and C_2, the optimal bandwidth is then obtained by cross-validation:

$$cv_F(h) = \frac{1}{n}\sum_{i=1}^{n}\int [I(X_i \leq x) - \hat{F}_h^{-i}(x)]^2 dx$$

where $\hat{F}_h^{-i}(x)$ is the CDF kernel estimator obtained after removing ith observation.

Adaptive KDE

- The basic definition of KDE assumes that the bandwidth h is constant for every individual kernel. A useful extension is to use a different h depending on the local density of the input data points.
- Adaptive KDE can be grouped into two categories: *balloon* estimators, and *sample point* estimators.
- The balloon estimator takes the form

$$\hat{f}_B(x;h) = \frac{1}{nh(x)} \sum_{i=1}^{n} K(\frac{x - X_i}{h(x)})$$

- Unfortunately, the balloon estimator suffers from a number of drawbacks the biggest one being that this estimator does not, in general, integrate to one over the entire domain.

- The MSE criterion means that the asymptotically optimal bandwidth is

$$h_{AMSE}(x) = \Big[\frac{f(x)\|K\|^2}{\kappa_{21}^2(f''(x))^2}\Big]^{1/5} n^{-1/5}$$

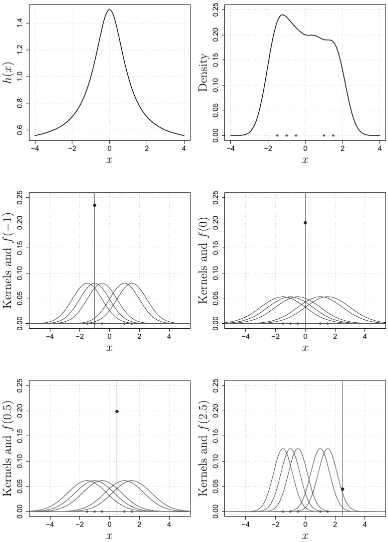
- The following Figure demonstrates how the balloon KDE works. The five data points are

$$X_1 = -1.5, X_2 = -1, X_3 = -0.5, X_4 = 1, X_5 = 1.5$$

  and an arbitrary chosen bandwidth function is

$$h(x) = 0.5 + 1/(x^2 + 1).$$

  The top left plot shows the $h(x)$ function. The top right plot shows the balloon KDE $\hat{f}_B(x; h(x))$. The last four plots show the kernels centered at each data point X_i and the KDE estimates at points $x = -1, x = 0, x = 0.5$ and $x = 2.5$. For every point x, a fixed bandwidth is chosen according to the $h(x)$ function.



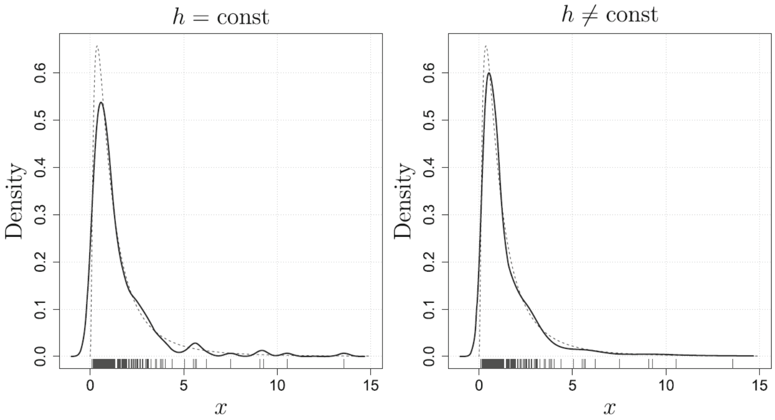
$h(x)$
Density
Kernels and $f(-1)$
Kernels and $f(0)$
Kernels and $f(0.5)$
Kernels and $f(2.5)$
x

- The sample point estimator uses a different bandwidth for each data point X_i. The estimate of f at every x is then an average of differently scaled kernels centered at each data point X_i. This estimator is described in the following way

$$\hat{f}_{SP}(x; h(X_i)) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h(X_i)} K(\frac{x - X_i}{h(X_i)})$$

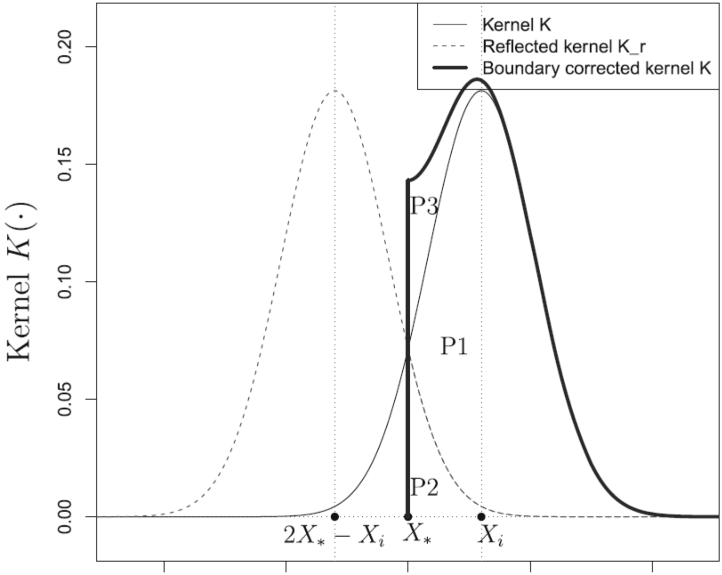
- Sample points estimators are 'true' densities but can suffer from another drawback, that is the estimate at a certain point can be strongly affected by data located far from the estimation point. However, this seems not to be a very serious problem in terms of practical applications and sample points estimators prove to be very useful.

A demonstration of the sample point KDE for the density $N(lnx; \mu = 0, \sigma = 1)$ with $n = 100$ and $h = 0.3$. The true density is plotted in the dashed line.



Kernel Density Estimation with Boundary Correction

- A general problem with KDE is that certain difficulties can arise at the boundaries and near them.
- In many practical situations the values of a random variable X are bounded. Even if a kernel with finite support is used, the consecutive KDE can usually go beyond the permissible domain.
- we present a smart procedure based on 'reflection' of same unnecessary KDE parts. See the following picture. Let the admissible domain be $X \in [X_*, \infty]$. The kernel K plotted in the thin solid line refers to a data point X_i.


Kernel K
Reflected kernel K_r
Boundary corrected kernel K
Kernel $K(\cdot)$
0.00
0.05
0.10
0.15
0.20
P1
P2
P3
$2X_* - X_i$
X_*
X_i

Obviously, the left-side boundary corrected kernel estimator is

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n}\Big[K(\frac{x-X_i}{h}) + K(\frac{x-(2X_*-X_i)}{h})\Big] I(x \in [X_*, \infty)).$$

and the right-side one is

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n}\Big[K(\frac{x-X_i}{h}) + K(\frac{x-(2X_*-X_i)}{h})\Big] I(x \in (-\infty, X_*]).$$

Boundary correction in general

- Assume, the support of f is $[0, \infty)$ and that f is two times continuous differentiable. K symmetric pdf wit support $[-1, 1]$.
- **Statistical properties in the interior of $f(x)$, $x \geq h$:**

$$E\hat{f}_h(x) \approx f(x) + \frac{1}{2}h^2\kappa_{21}f''(x)$$
$$Var(\hat{f}_h(x)) \approx \frac{1}{nh}\kappa_{02}f(x)$$

for $h = h(n) \to 0, n \to \infty$ and $nh \to \infty$.

- **Statistical properties at the boundary of $f(x)$, $x < h$:**
  Let $x = ph$ and $p < 1$(For $p \geq 1$ we are in the interior)

$$E\hat{f}_h(x) \approx a_0(p)f(x) - a_1(p)hf'(x) + \frac{1}{2}h^2 a_2(p)f''(x)$$
$$Var(\hat{f}_h(x)) \approx \frac{1}{nh}b(p)f(x)$$

where $a_l(p) = \int_{-1}^{p} u^l K(u)du$ and $b(p) = \int_{-1}^{p} K^2(u)du$.

Consistent: The kernel estimator is not consistent at the boundary, $E\hat{f}_h(0) \to \frac{f(0)}{2}$.

Simple $O(h)$ boundary corrections

Ensuring consistency at the boundary: Ensure the leading term in the expectation of the "boundary- corrected" kernel density estimate is $f(x)$.
Renormalization:
The multiplier of $f(x)$ is $\int_{-1}^{p} K(u)du$
Problem: The kernel mass "lost" beyond the boundary.
One solution: Renormalize each kernel to integrate to 1 ("local" renormalization)

$$\hat{f}_N(x) = \frac{\hat{f}_h(x)}{a_0(p)}$$

Notice: $a_0(p) = 1$ for $p \geq 1$, the formula works also in the interior.

- **Statistical properties of $\hat{f}_N(x)$:** For $\hat{f}_N(x) = \frac{\hat{f}_h(x)}{a_0(p)}$:

$$E\hat{f}_N(x) \approx f(x) - h\frac{a_1(p)}{a_0(p)}f'(x)$$

$$Var(\hat{f}_N(x)) \approx \frac{1}{nh}\frac{b(p)}{a_0^2(p)}f(x)$$

  Notice: $\hat{f}_N$ is consistent, but the bias is of order $O(h)$ near the boundary. Optimal MSE is of order $n^{-2/3}$ at the boundary, and of order $n^{-4/5}$ elsewhere.

- Another solution: (**Reflection**) Reinstate the "missing mass" by reflecting the estimate in the boundary

$$\hat{f}_R(x) = \hat{f}_h(x) + \hat{f}_h(-x)$$

  or equivalently replace $K_h(x - X_i)$ by $K_h(x - X_i) + K_h(-x - X_i)$.

- **Statistical properties of $\hat{f}_R(x)$:** For $\hat{f}_R(x) = \hat{f}_h(x) + \hat{f}_h(-x)$:

$$E\hat{f}_R(x) \approx f(x) - h2[a_1(p) + p(1 - a_0(p))]f'(x)$$
$$Var(\hat{f}_R(x)) \approx \frac{1}{nh}(\kappa_{02} + 2\int_{-1}^{p} K(u)K(u - 2p)du)f(x)$$

Notice: $\hat{f}_R$ is consistent, but the bias is of order $O(h)$ near the boundary. Optimal MSE is of order $n^{-2/3}$ at the boundary, and of order $n^{-4/5}$ elsewhere.

Comparison of renormalization $\hat{f}_N$ and reflection $\hat{f}_R$:

We compare the leading terms of bias and variance as function of p (ie. multipliers of $-hf'(x)$ and $\frac{1}{nh}f(x)$, respectively) for the biweight kernel, $K(t) = \frac{15}{16}(1-x^2)^2, x \in [-1,1]$.

- The leading terms of bias and variance of $\hat{f}_N$

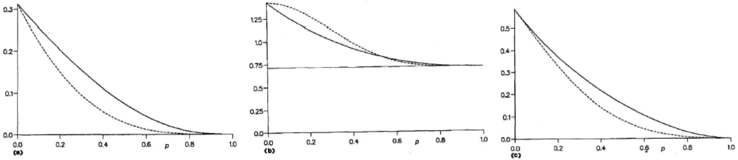
$$B(p) = \frac{a_1(p)}{a_0(p)}, V(p) = \frac{b(p)}{a_0^2(p)}$$

- The leading terms of bias and variance of $\hat{f}_R$

$$B(p) = 2[a_1(p) + p(1 - a_0(p))],$$
$$V(p) = (\kappa_{02} + 2\int_{-1}^{p} K(u)K(u-2p)du)$$

- Optimized mean squred error

$$[B(p)V(p)]^{2/3}$$



$\hat{f}_N$: Renormalization (solid line), and $\hat{f}_R$. Reflection (dashed line).

- Bias: Bias of $\hat{f}_R \leq$ Bias of $\hat{f}_N$ for $p \in [0, 1]$ (small difference).
- Variance: Variance of $\hat{f}_R \geq$ Variance of $\hat{f}_N$ for p less than about one half and opposite above one half (marginally).
- Combination of variance and bias: Reflection beats renormalization for all p (but small difference).

General conclusion: Very little difference between the two methods, and not as good as the following methods...

Generalized jackknifing

Goal: $O(h^2)$ bias near the boundary as well as in the interior.
Idea: Take a linear combination of K and L (closely related to K) in such a way that the resulting kernel has $a_0(p) = 1$ and $a_1(p) = 0$. The following linear combination has the desired $O(h^2)$ bias property

$$\frac{c_1(p)K(x) - a_1(p)L(x)}{c_1(p)a_0(p) - a_1(p)c_0(p)}$$

where $c_l(p) = \int_{-1}^{p} u^l L(u)du$.

For $L(x) = cK(cx)$, where $0 < c < 1$. Then the resulting "boundary kernel" is

$$K_c(x) = \frac{(a_1(pc) - a_1(c))K(x) - a_1(p)c^2K(cx)}{(a_1(pc) - a_1(c))a_0(p) - a_1(p)c(a_0(pc) + a_0(c) - 1)}$$

Choose $c = c(K)$ to optimize eg. some measure of effectiveness of the kernel, however there is very little to be gained.
Instead, let $c \to 1$,

$$K_{PD}(x) = \frac{a_2^{(1)}(p)K(x) - a_1(p)xK'(x)}{a_2^{(1)}(p)a_0(p) - a_1(p)a_1^{(1)}(p)}$$

where $a_l^{(1)} = \int_{-1}^{p} x^l K'(x)dx$
Notice: Alternative derivation would be to seek the appropriate linear combination of $K(x)$ and $xK'(x)$ to use as a boundary kernel.

A particularly useful boundary kernel comes from the linear combination of $K(x)$ and $xK(x)$

$$K_L(x) = \frac{a_2(p)K(x) - a_1(p)xK(x)}{a_0(p)a_2(p) - a_1^2(p)}$$

Another boundary kernel

$$K_D(x) = \frac{a_1^{(1)}(p)K(x) - a_1(p)K'(x)}{a_1^{(1)}(p)a_0(p) - a_1(p)a_0^{(1)}(p)}$$

which is a linear combination of ... $K(x)$ and $K'(x)$.
Notice: K_D not applicable to the uniform kernel, and K_D analogous to K_L for the normal kernel.

- Extension of reflection

$$K_{R1}(x) = \frac{(2p(1-a_0(p)) + a_1(p))K(x) - a_1(p)K(2p-x)}{(2p(1-a_0(p)) + a_1(p))a_0(p) - a_1(p)(1-a_0(p))}$$

Overview of boundary kernels

- General Jackknifing:

$$\frac{c_1(p)K(x) - a_1(p)L(x)}{c_1(p)a_0(p) - a_1(p)c_0(p)}$$

- Comb. of $K(x)$ and $cK(cx)$:

$$K_c(x) = \frac{(a_1(pc) - a_1(c))K(x) - a_1(p)c^2K(cx)}{(a_1(pc) - a_1(c))a_0(p) - a_1(p)c(a_0(pc) + a_0(c) - 1)}$$

- Comb. of $K(x)$ and $cK(cx)$ for $c \to 1$ (comb. of $K(x)$ and $xK'(x)$):

$$K_{PD}(x) = \frac{a_2^{(1)}(p)K(x) - a_1(p)xK'(x)}{a_2^{(1)}(p)a_0(p) - a_1(p)a_1^{(1)}(p)}$$

- Comb. of $K(x)$ and $xK(x)$:

$$K_L(x) = \frac{a_2(p)K(x) - a_1(p)xK(x)}{a_0(p)a_2(p) - a_1^2(p)}$$

- Comb. of $K(x)$ and $K'(x)$ (ext. of renormalization):

$$K_D(x) = \frac{a_1^{(1)}(p)K(x) - a_1(p)K'(x)}{a_1^{(1)}(p)a_0(p) - a_1(p)a_0^{(1)}(p)}$$

- Comb. of $K(x)$ and $K(2p - x)$ (ext. of reflection):

$$K_{R1}(x) = \frac{(2p(1 - a_0(p)) + a_1(p))K(x) - a_1(p)K(2p - x)}{(2p(1 - a_0(p)) + a_1(p))a_0(p) - a_1(p)(1 - a_0(p))}$$

Comparison of $O(h^2)$ boundary kernels:

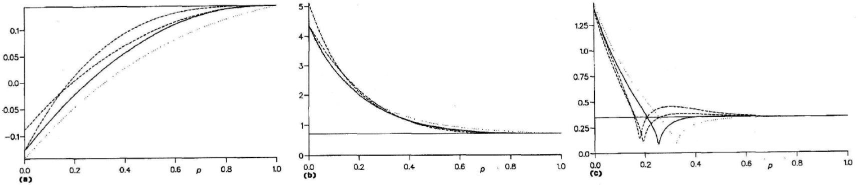
Compare the leading coefficients of bias and variance (ie. the multiplier of $\frac{1}{2}h^2 f''(x)$ and $\frac{1}{nh}f(x)$, respectively).

General formulae in terms of K and L for all generalized jackknife boundary kernels

$$B(p) = \frac{c_1(p)a_2(p) - a_1(p)c_2(p)}{c_1(p)a_0(p) - a_1(p)c_0(p)}$$

$$V(p) = \frac{c_1^2(p)b(p) - 2c_1(p)a_1(p)e(p) + a_1^2(p)g(p)}{(c_1(p)a_0(p) - a_1(p)c_0(p))^2}$$

where $e(p) = \int_{-1}^{p} K(x)L(x)dx$ and $g(p) = \int_{-1}^{p} L^2(x)dx$.



Bias, variance and optimized MSE for K_{PD} (dotted line), K_L (dashed line), K_D (solid line) and K_{R1} (dot-dashed line).

- Bias: Bias curves same shape and range of values. Each curve has a single point where it crosses zero.
- Variance: The variance is very similar.
- Optimized MSE: $\{B(p)V^2(p)\}^{2/5}$. Similar curves.

Note: The slightly increased variance of $\hat{f}_L$ close to $p = 0$ is balanced by the better bias there (dashed line).

general conclusion

- Almost equivalent results for all generalized jackknives.
- Major problem: The variance at and very close to $p = 0$.

For the biweight kernel

$$\frac{V(\hat{f}_L(0))}{V(\hat{f}_L(1))} \approx 7.16$$

whereas

$$\frac{V(\hat{f}_N(0))}{V(\hat{f}_N(1))} \approx 7.16$$

Hope for improved boundary corrections techniques. Local linear estimation has an attractive performance at the boundaries. (see reading paper for details)

Higher-order kernels

- We know that the best obtainable rate of convergence of the kernel estimator is of order $n^{-4/5}$. If we loose the condition that K must be a density, the convergence rate could be faster.
- We say an asymmetric function K is a kth order kernel if

$$\int K(u)du = 1, \int u^j K(u)du = 0 \ for \ j = 1, \dots, k-1$$

and

$$\int u^k K(u)du \neq 0$$

- Note that we do not require that $K(u) \geq 0$.
- One way to generate higher-order kernels is deductively from the lower-order kernels,

$$K_{[k+2]}(u) = \frac{3}{2}K_{[k]}(u) + \frac{1}{2}uK'_{[k]}(u)$$

  for example, set $K_{[2]}(u) = \phi(u)$, then
  $K_{[4]}(u) = \frac{1}{2}(3 - u^2)\phi(u)$.
- Another way is developed when f is a normal mixture density for a certain class of higher-order kernels

$$G_{[k]}(u) = \sum_{l=0}^{k/2-1} \frac{(-1)^l}{2^l l!}\phi^{(2l)}(u), l = 0, 2, 4, \ldots$$

For example, recall the asymptotic bias is given by

$$E\hat{f}_h(x) - f(x) = \frac{h^2}{2}\kappa_{21} f''(x) + o(h^2)$$

If we use 4th order kernel, then

$$\begin{aligned} E\hat{f}_h(x) &= \frac{1}{h}\int K(\frac{z-x}{h})f(z)dz = \int K(u)f(x+uh)du \\ &= \int K(u)[f(x) + f'(x)uh + \frac{1}{2}f''(x)u^2h^2 + \frac{1}{3!}f^{(3)}(x)u^3h^3 \\ &+ \frac{1}{4!}f^{(4)}(x)u^4h^4 + o(h^4)]du \\ &= f(x) + \frac{1}{4!}f^{(4)}(x)\kappa_{41}h^4 + o(h^4) \end{aligned}$$

The variance does not change, that is,

$$Var(\hat{f}_h(x)) = \frac{f(x)}{nh}\kappa_{02} + o(\frac{1}{nh})$$

Therefore,

$$AMISE(h) = \frac{\kappa_{02}}{nh} + \frac{1}{(4!)^2}\|f^{(4)}\|^2\kappa_{41}^2 h^8$$

Then the optimal bandwidth is

$$h_0 = \Big[\frac{72\kappa_{02}}{\|f^{(4)}\|^2\kappa_{41}^2}\Big]^{-1/9} n^{-1/9}$$

and $AMISE(h_0)$ thus has an optimal convergence rate of order $O_p(n^{-8/9})$.

- The convergence rate can be made arbitrarily close to the parametric n^{-1} as the order increases, which means it will eventually dominate second-order kernel estimators for large n. However, it does need a larger sample size ($K_{[4]}$ would require several thousand in order to reduce MISE compared to normal kernel).
- Another price that need to be paid for higher-order kernels is the negative contributions of the kernel may make the the estimated density not a density itself.

Computation Aspect

- CRAN packages **graphics::hist** and **ash** packages allows users to generate a histogram of the data x.
- CRAN packages **GenKern**, **kerdiest**, **KernSmooth**, **ks**, **np**, **plugdensity**, and **sm** all use the kernel density approach, as does **stats::density**. They differ primarily in their means of selecting bandwidth.
- CRAN packages **vemix** provides density, cumulative distribution function, quantile function and random number generation for boundary corrected kernel density estimators using a variety of approaches.

Package	Function cal	Max Dim.	Arbitrary Grid	Predicted Density	Approach
ASH	`ash1(bin1(x, ab = c(min(x), max(x)), nbin = 512))`	2	No	`d$y`	ASH
ftnonpar	`pmden(x)`	1	No	`d$y`	Taut Strings
GenKern	`KernSec(x, 512, range.x = c(min(x), max(x)))`	2	No	`d$yden/100`	Kernel
gss	`dssden(ssden(~x), seq(min(x), max(x), length = 512))`	2	Yes	`d`	Penalized
kerdiest	`kerdiest::kde(vec_data = x, y = xgrid)`	1	Yes	`d$Estimated_values`	Kernel
KernSmooth	`bkde(x = x, gridsize = 512L, range.x(min(x), max(x)))`	2d	No	`d$y`	Kernel
ks	`kde(x = x, hpi(x), eval.points = xgrid)`	6	Yes	`d$estimate`	Kernel
locfit	`density.lf(x, ev = xgrid)`	1	Yes	`d$y`	Local Likelihood
logspline	`dlogspline(xgrid, logspline(x))`	1	Yes	`d`	Penalized
MASS	`hist(x, 512)`	1	Yes	`d$density`	Histogram
np	`npudens(~ x, edat = xgrid)`	1	Yes	`d$dens`	Kernel
pendensity	`pendensity(x ~ 1)`	1	No	`d$results$fitted`	Penalized
plugdensity	`plugin.density(x, xout = xgrid)`	1	Yes	`d$y`	Kernel
stat	`density(x, n = 512)`	1	No	`d$y`	Kernel
sm	`sm.density(x, display = "none", eval.points = xgrid)`	3	Yes	`d$estimate`	Kernel

Table 1: Packages we investigated. We assume that the estimate output is `d`, the input data is `x`, and the desired evaluation grid is `xgrid`, which sequences x into 512 evaluation points.