Computational Statistics Presentation

1.1 Density Estimation

Saifullah Khan Babrak, Magnus Raabo Andersen, Martin Hoshi Vognsen

Non-parametric density estimate with Epanechnikov kernel

Kernel estimator:

$$\hat{f}_h(x) = \frac{1}{hn}\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right) \quad (1)$$

Epanechnikov kernel:

$$K(x) = \frac{3}{4}(1-x^2)\mathbb{1}_{[-1,1]}(x) \quad (2)$$

Squared 2-norm of 2nd derivative of pilot density using Gaussian kernel

$$\begin{aligned}
\|\widetilde{f}''\|_2^2 &= \int_{-\infty}^{\infty} \widetilde{f}_r''(x)^2 dx \\
&= \frac{1}{n^2 r^6}\sum_{i=1}^{n}\sum_{j=1}^{n}\int_{-\infty}^{\infty} H''\left(\frac{x-x_i}{r}\right) H''\left(\frac{x-x_j}{r}\right) dx \\
&\stackrel{*}{=} \frac{1}{n^2(\sqrt{2}r)^5}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi^{(4)}\left(\frac{x_i-x_j}{\sqrt{2}r}\right) \\
&= \frac{1}{n^2(\sqrt{2}r)^5}\sum_{i=1}^{n}\sum_{j=1}^{n} e^{-\frac{1}{2}w^2}\left(\frac{w^4-6w^2+3}{\sqrt{2\pi}}\right),
\end{aligned} \quad (3)$$

*Clive R. Loader: "Bandwidth Selection: Classical or Plug-In?"

where $w = \left(\frac{x_i-x_j}{\sqrt{2}r}\right)$

$$= \frac{1}{8n^2r^5\sqrt{\pi}}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[e^{c_1 z_{ij}^2}\left((c_2 z_{ij})^4 - 6(c_2 z_{ij})^2 + 3\right)\right] \quad (4)$$

where $z = x_i - x_j$, $c_1 = -1/(4r^2)$, $c_2 = 1/(\sqrt{2}r)$.

$$\begin{aligned}
\mathsf{IQR}_{\text{theoretical}} &= \Phi^{-1}(0.75) - \Phi^{-1}(0.25) \\
\mathsf{IQR}_{\text{empirical}} &= \texttt{quantile(x, 0.75) - quantile(x, 0.25)} \\
\widetilde{\sigma} &= \min(\widehat{\sigma}, \mathsf{IQR}_{\text{empirical}}/\mathsf{IQR}_{\text{theoretical}}) \\
\widehat{r} &= \left(\frac{4}{3}\right)^{\frac{1}{5}} \widetilde{\sigma} n^{-\frac{1}{5}} \approx 1.059224 \widetilde{\sigma} n^{-\frac{1}{5}}
\end{aligned} \quad (5)$$

Optimal $\widehat{h}$ by AMISE:

$$\widehat{h}_n = \left(\frac{\|K\|_2^2}{\|\widetilde{f}_0''\|_2^2 \sigma_K^4}\right)^{\frac{1}{5}} n^{-\frac{1}{5}} \quad (6)$$

where $\|K\|_2^2 = \left(\frac{3}{4}\right)^2 \int_{-1}^{1}\left(1-x^2\right)^2 dx = 0.6$ and $\sigma_K^2 = 2\int_0^1 x^2 K[x]\, dx = 2\int_0^1 \frac{1}{4}3x^2\left(1-x^2\right) dx = \frac{1}{5}$

Implementation

Kernel density estimator:

$$\hat{f}_h(x) = \frac{1}{hn} \sum_{j=1}^{n} K\left(\frac{x - x_j}{h}\right) \tag{7}$$

```
kern_dens <- function(x, h, m = 512, kernel =
     epan) {
   rg <- range(x)
   n <- length(x)
   grid_points <- seq(rg[1] - 3 * h, rg[2] + 3 *
     h, length.out = m)
   y <- numeric(m)
   for (i in seq_along(grid_points)) {
      y[i] <- sum(kernel((grid_points[i] -
     x[j])/h))
   }
   y <- y / (n*h)
   list(x = grid_points, y = y)
}
epan <- function(x){
   val <- 0.75*(1 - x^2)
   val * (abs(x) < 1)
}

n = 10000
x = rnorm(n)
q = seq(-5, 5, length.out = n)
norm_dens = sapply(q, function(q) {1/(sqrt(2*pi))
     * exp(-0.5*q^2)}
```

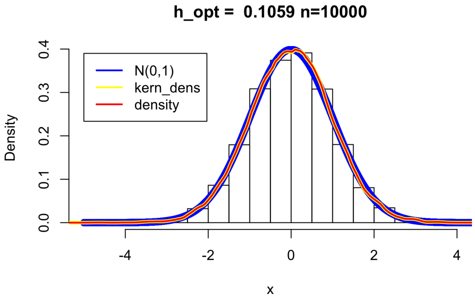

Figure 1: Correctness of density estimate

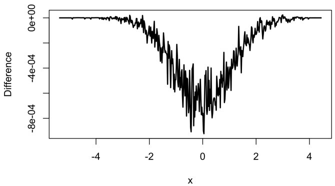

Figure 2: f_hat - f_hat_dens

Bandwidth selection by AMISE

Estimate optimal oracle bandwidth h_n with $\widehat{h}_n$:

$$\widehat{h}_n = \left(\frac{\|K\|_2^2}{\|\widetilde{f}_0''\|_2^2 \sigma_K^4} \right)^{\frac{1}{5}} n^{-\frac{1}{5}} \tag{8}$$

where $\|K\|_2^2 = \left(\frac{3}{4}\right)^2 \int_{-1}^{1} \left(1 - x^2\right)^2 \, dx = 0.6$ and
$\sigma_K^2 = 2 \int_0^1 x^2 K[x] \, dx = 2 \int_0^1 \frac{1}{4} 3x^2 \left(1 - x^2\right) \, dx = \frac{1}{5}$

Estimate pilot bandwith r with $\widehat{r}$:

$$\widehat{r} = \left(\frac{4}{3}\right)^{\frac{1}{5}} \widetilde{\sigma} n^{-\frac{1}{5}} \approx 1.059224 \widetilde{\sigma} n^{-\frac{1}{5}} \tag{9}$$

$$\begin{aligned} \text{IQR}_{\text{theoretical}} &= \Phi^{-1}(0.75) - \Phi^{-1}(0.25) \\ \text{IQR}_{\text{empirical}} &= \texttt{quantile(x, 0.75) - quantile(x, 0.25)} \\ \widetilde{\sigma} &= \min(\widehat{\sigma}, \text{IQR}_{\text{empirical}} / \text{IQR}_{\text{theoretical}}) \end{aligned} \tag{10}$$

```r
hn_hat <- function(wiggle, n) {
   K_2norm <-0.6
   (25 * K_2norm / wiggle)^(0.2) * n^(-0.2)
}
```

```r
r_hat <- function(x, n) {
   sigma_hat <- sd(x)
   IQR <- quantile(x, 0.75) - quantile(x, 0.25)
   IQR_theoretical <- qnorm(0.75) - qnorm(0.25)
   sigma_tilde <- min(sigma_hat,
     IQR/IQR_theoretical)
   ##  Silverman's rule: 0.9 * sigma_tilde *
     n^(-0.2)
   (4/3)^(1/5) * sigma_tilde * n^(-0.2) ##
     (4/3)^(1/5) = 1.059224
}
```

Optimization

Squared norm of 2nd derivative of pilot density

$$\|\widetilde{f}''\|_2^2 = \frac{1}{8n^2 r^5 \sqrt{\pi}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left[e^{c_1 z_{ij}^2} \left((c_2 z_{ij})^4 - 6(c_2 z_{ij})^2 + 3 \right) \right] \tag{11}$$

Nested

```
wiggle_gauss <- function(x, r) {
wiggle = 0
c1 = -1/(4*r^2)
c2 = 1/(sqrt(2)*r)
for(i in seq_along(x)){
   for(j in seq_along(x)) {
      z <- (x[i] - x[j])/c2
      wiggle <- wiggle + exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3)
   }
}
wiggle <- wiggle / (8 * n^2 * r^5 * sqrt(pi))
wiggle
}
```

Optimization (cont)

Sum

```
wiggle_gauss_vec <- function(x, r) {
wiggle = 0
c1 = -1/(4*r^2)
c2 = 1/(sqrt(2)*r)
for(i in seq_along(x)){
   z <- (x[i] - x)/c2
   wiggle <- wiggle + sum(exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3))
}
wiggle <- wiggle / (8 * n^2 * r^5 * sqrt(pi))
wiggle
}
```

Outer

```
wiggle_gauss_outer <- function(x, r) {
c1 = -1/(4*r^2)
c2 = 1/(sqrt(2)*r)
wiggle <- outer(x, x, function(ww, w){
   z <- (ww - w)/c2
   exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3)
})
sum(wiggle) / (8 * n^2 * r^5 * sqrt(pi))
}
```

AMISE plug-in profiling

Runtime in ms, n=1000

```r
wiggle_gauss <- function(x, r) {
  wiggle = 0
  c1 = -1/(4*r^2)
  c2 = 1/(sqrt(2)*r)
  for(i in seq_along(x)){
    for(j in seq_along(x)) {
      z <- (x[i] - x[j])/c2
      wiggle <- wiggle + exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3)
    }
  }
  wiggle <- wiggle / (8 * n^2 * r^5 * sqrt(pi))
  wiggle
}
```

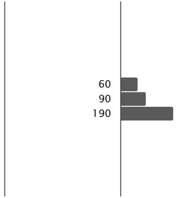

Figure 3: $||\widetilde{f}''||_2^2$ with nested loop implementation

```r
wiggle_gauss_vec <- function(x, r) {
  wiggle = 0
  c1 = -1/(4*r^2)
  c2 = 1/(sqrt(2)*r)
  for(i in seq_along(x)){
      z <- (x[i] - x)/c2
      wiggle <- wiggle + sum(exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3))
  }
  wiggle <- wiggle / (8 * n^2 * r^5 * sqrt(pi))
  wiggle
}
```

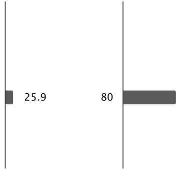

Figure 4: $||\widetilde{f}''||_2^2$ with `sum` vectorization implementation

```r
wiggle_gauss_outer <- function(x, r) {
  c1 = -1/(4*r^2)
  c2 = 1/(sqrt(2)*r)
    wiggle <- outer(x, x, function(ww, w){
      z <- (ww - w)/c2
      exp(c1 * z^2) * ((c2*z)^4 - 6 * (c2*z)^2 + 3)
    })
  sum(wiggle) / (8 * n^2 * r^5 * sqrt(pi))
}
```

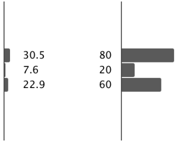

Figure 5: $||\widetilde{f}''||_2^2$ with `outer` implementation

AMISE plug-in benchmarking

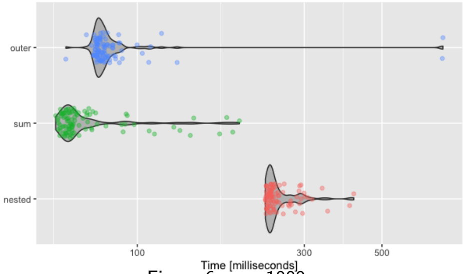

Figure 6: n = 1000

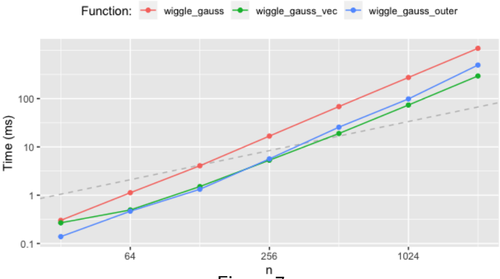

Figure 7

Gray dashed line: $o(n)$, `wiggle_gauss`: $o(n^2)$

Runtimes for `kern_dens` (`my_density`) and `density`

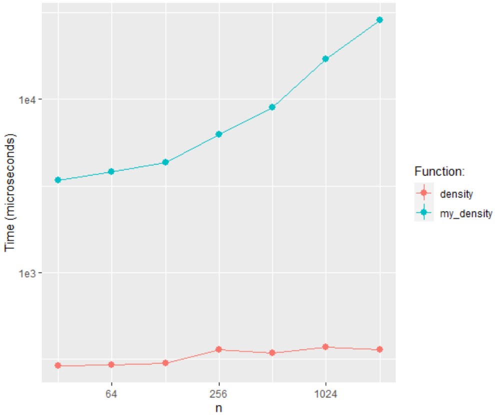

Figure 8

Future tests

- RCPP for speed
- OOP for generality
- LOOCV
- binning