simulation_week6

Tingyu Zhu

11/2/2020

```r
library(tidyverse)
library(ggplot2)
source("kde_est_func.R")
```

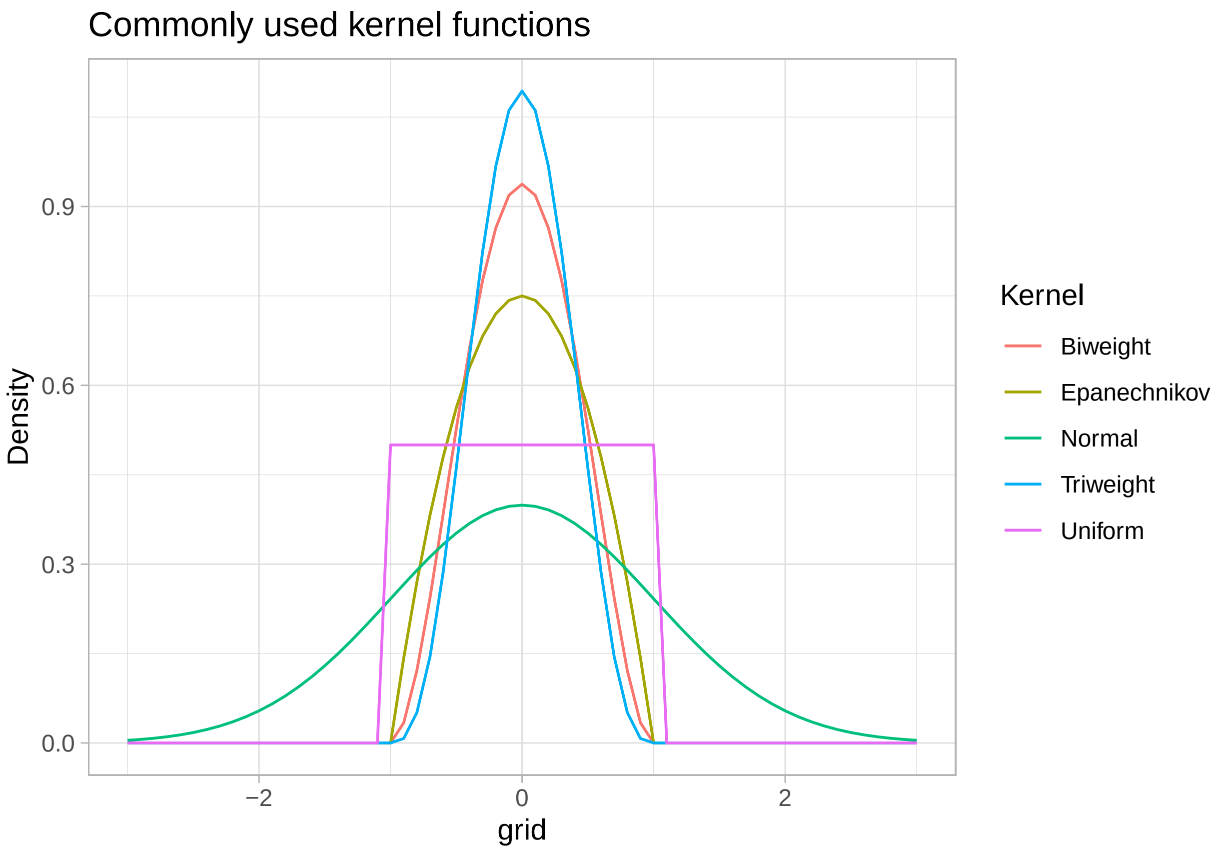
Commonly used kernel functions

In the beta family kernel density, lambda=0,1,2,3 correspond to the uniform, the Epanech- nikov, the biweight, and the triweight kernel functions, respectively.

```r
# Define kernel functions
kernel_norm <- function(x) dnorm(x)
Kernel_beta <- function(x,lambda){(1-x^2)^lambda*(abs(x)<=1)/beta(lambda+1,0.5)}
kernel_comp <- tibble(
  grid = seq(-3,3,by=0.1),
  Normal = dnorm(grid),
  Uniform = Kernel_beta(grid,0),
  Epanechnikov = Kernel_beta(grid,1),
  Biweight = Kernel_beta(grid,2),
  Triweight = Kernel_beta(grid,3)
  )

kernel_comp_long <- kernel_comp %>%
  pivot_longer(c(`Normal`, `Uniform`,`Epanechnikov`,
                 `Biweight`,`Triweight`), names_to = "Kernel", values_to = "Density")


ggplot(data=kernel_comp_long) +
  geom_line(aes(x = grid, y = Density, color = Kernel)) +
  labs(title = "Commonly used kernel functions") +
  theme_light()
```



Compute the MADE (Mean Absolute Deviation Errors)

The Mean Absolute Deviation Errors for $\hat{f}(\cdot)$is defined as

$$MADE = \frac{1}{n}\sum_{k=1}^{n}|(\hat{f}(u_k) - f(u_k))|$$

, where $\hat{f}(u_k)$ is the kernel estimte of $f(u_k)$), and $\{u_k\}$ are the frid points taken to be arbitrary within the range of data.

When the sample is generated from the known distribution like exp,beta, gamma, etc., we can use corresponding built-in functions (dexp(), dgamma, etc.) to calculated the true denstities of the grid points. If the sample is not from the familar distributions, we can use the built-in function "denstiy()" in R to get the value of $f(u_k)$)s.

```
made <- function(f_est,f_true) mean(abs(f_est-f_true))
```

sample from nrom(mu,s^2)

```
n <- 200
s <- 0.5
x <- rnorm(n, 1, s)
grid <- seq(-5,5,by=0.1)
### normal reference bandwidth selector
```

```r
h <- 1.06*s*n^{-0.2}
f1 <- KDE_est(x,grid,h,kernel_norm)
f1_true <- dnorm(grid, 1,s)

made(f1 , f1_true)
```

```
## [1] 0.01618103
```

```r
###
### plot
data <- cbind(f1_true, f1)
matplot(grid, data, lty=c(1:2),col=c(1:2), type="l")
legend(2,0.6,c("True","ker_est"),lty=c(1:2),col=c(1:2))
```