Time Series and Prediction
Computing Homework 1

The problem set is due on Monday 28th at 11:59pm.

Questions

1. Write a function in `R` called "`my.kernel`" to compute the value of the kernel function

$$k(u) = \begin{cases} \frac{35}{32}(1-u^2)^3 & \text{for } |u| \leq 1 \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

   The function "`my.kernel`"

   - requires as input a vector of real numbers
   - returns as output a vector (of the same size as the input vector). The i-th component of the output vector contains the value of the kernel function corresponding to the i-th element of the input vector

2. Write a function in `R` called "`my.kernel.density.estimator`" to compute the kernel density estimate using the kernel function in equation (1), that is

$$\hat{f}_h(y) = \frac{1}{Th} \sum_{t=1}^{T} k\left(\frac{y_t - y}{h}\right) , \tag{2}$$

   where y is a point in the support of Y_t, $y_1, \ldots, y_T$ is a realized sample of size T from $\{Y_t\}$ and h is the bandwidth parameter of the kernel.

   The function "`my.kernel.density.estimator`"

   - requires as inputs (in this order)
     (a) a point in the support of Y (i.e. y)
     (b) a vector containing the realized sample $y_1, \ldots, y_T$
     (c) the bandwidth parameter of the kernel h
   - returns as output the kernel estimate of the density $\hat{f}_h(y)$

3. Write a function in `R` called "`my.loglik.cv`" to compute the cross-validated log-likelihood relative to the bandwidth h, that is

$$\mathcal{L}_h = \sum_{t=1}^{T} \log \hat{f}_{h\,(-t)}(y_t) .$$

   where $\log$ is the natural log and $\hat{f}_{h\,(-t)}(\cdot)$ denotes the kernel density estimate in (2) computed on the basis of the realized sample $y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_T$. (that is on the basis of the realized sample without y_t). The cross-validated log-likelihood is used to evaluate different values of the bandwidth h: between two bandwidths h_1 and h_2 we choose the one associated with the higher-value of the cross-validated log-log likelihood.

   The function "`my.loglik.cv`"

- requires as inputs (in this order)
  (a) a vector containing the realized sample $y_1, \ldots, y_T$
  (b) the bandwidth parameter of the kernel h
- returns as output the cross-validated log-likelihood $\mathcal{L}_h$

Test Code

The following code can be used to test the functions above. This is also the code that is used to evaluate the accuracy of the submission.

```
# 1) evaluate the kernel function over a grid of points
kern <- my.kernel( seq(-3,3,0.01) )

plot( seq(-3,3,0.01) , kern , t='l')

# 2) simulate 1000 obs from a beta distribution and estimate the density
set.seed(12345)

data <- rbeta(1000,3,2)

x.range <- seq(0,1,0.1)
hat.f   <- rep(0,length(x.range))

# For now we set h=0.03. We shall optimize h by cross-validation below
h <- 0.03
for( t in 1:length(x.range) ){
  hat.f[t] <- my.kernel.density.estimator(x.range[t],data,h)
}

plot( x.range , hat.f , t='l' , lwd=2 , col='darkblue' )
lines( x.range , dbeta(x.range,3,2) , t='l' , lwd=2 , col='violet' )
legend( 'topleft' , c('kernel','true') , col=c('darkblue','violet') , lwd=3 )

# 3) evaluate which bandwidth h has smaller cross-validated loss
h.vec <- c( (2:10)/100 )
l.cv  <- rep(0,length(h.vec))

for( i in 1:length(h.vec) ){
  l.cv[i] <- my.loglik.cv(data,h.vec[i])
}

plot( h.vec , l.cv , t='b' , lwd=3 , col=sample(colors(),1) )

# construct fit for optimal h, compare with weird h
h.opt   <- h.vec[ which( l.cv==max(l.cv) ) ]
h.weird <- 0.002
```

```
37
38 hat.f.opt    <- rep(0,length(x.range))
39 hat.f.weird  <- rep(0,length(x.range))
40
41 for( t in 1:length(x.range) ){
42   hat.f.opt[t]   <- my.kernel.density.estimator(x.range[t],data,h.opt)
43   hat.f.weird[t] <- my.kernel.density.estimator(x.range[t],data,h.weird)
44 }
45
46 plot( x.range , hat.f.opt , t='l' , lwd=2 , col='red3' )
47 lines( x.range , hat.f.weird , t='l' , lwd=2 , col='blue' )
48 lines( x.range , dbeta(x.range,3,2) , t='l' , lwd=2 , col='violet' )
49 legend( 'topleft' , c('opt h','weird h','true') , col=c('red2','blue','violet') , lwd=3 )
```

Submission Instructions

Submit your homework by e-mail by sending it to the address `time.series.bot@gmail.com`
It is important that:

- you send your homework from the `barcelonagse.eu` e-mail account
- the subject of the e-mail must be "homework 1"
- you must copy and paste in the body of the e-mail the `R` code containing your function definitions within the lines "`# BEGIN CODE`" and "`# END CODE`". (Note that these two lines are not counted as comment lines in the problem set correction.)
- please note that the code you submit should only include the function definitions.
- you have three retries (in case you are not satisfied with the grade of your submission). Please note that the final grade of the homework is based on the last submission.