154 - Homework 2

2022-09-10

1. Choosing k via (leave-one-out) Cross-Validation (LOOCV)

In the first homework, you coded a k-NN regression estimator, and similar to the standard kernel estimator with bandwidth h, care needs to be taken in choosing k for the resulting estimate to be informative. Write a cross-validation routine for choosing the value of k. Generate data from wherever you prefer (lecture 3 has a function we considered), and see how the cross-validated estimate compares to the truth (so don't use XKCD since we don't have access to the truth).

Note: Lecture 4 has code for a LOO-CV routine for choosing h that you could adapt.

To remove an observation from the data set, consider the following

```
x <- runif(5)
x
```

```
## [1] 0.03818384 0.37601990 0.35373791 0.73719414 0.11878630
```

```
x.temp <- x[-5] # removes the 5th observation
x.temp
```

```
## [1] 0.03818384 0.37601990 0.35373791 0.73719414
```

```
x.temp <- x[-c(1,5)] # removes the 1st and 5th observation
x.temp
```

```
## [1] 0.3760199 0.3537379 0.7371941
```

Please see the attached Jupyter notebook.

2.

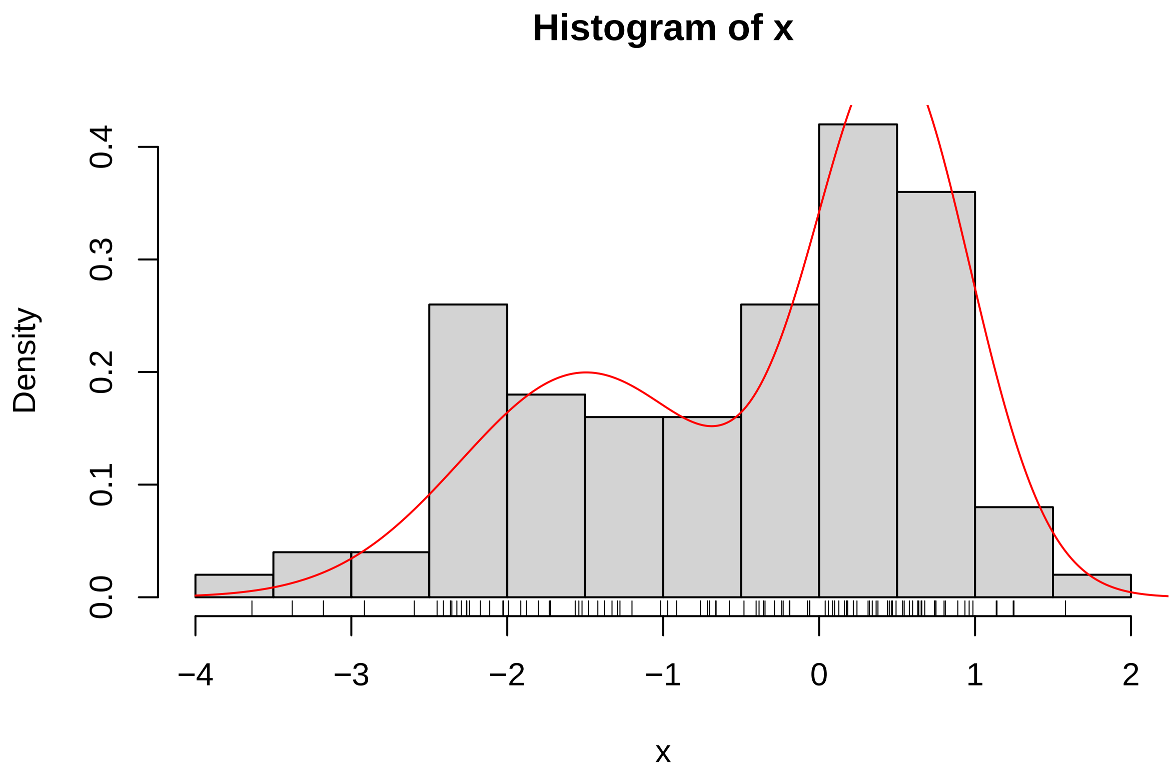
Choosing a bandwidth in a regression problem seems reasonably done via cross-validation.

Kernel estimators also show up in density estimation (the first kernel smoother that we actually ever saw). How would we do cross validation in this setting?

Generate data from a bimodal distribution (normal mixture - flip a coin and generate from a normal depending on the outcome of the flip)

```
grid <- seq(-4,3,.01)
f.x <- .4*dnorm(grid, -1.5,.8) + .6*dnorm(grid, .47, .5)

n <- 100
u <- runif(n)
x <- rnorm(100,-1.5, .8)*(u<=.4) + rnorm(100,.47, .5)*(u>.4)
hist(x, prob=TRUE)
lines(grid,f.x, col='red')
rug(x)
```

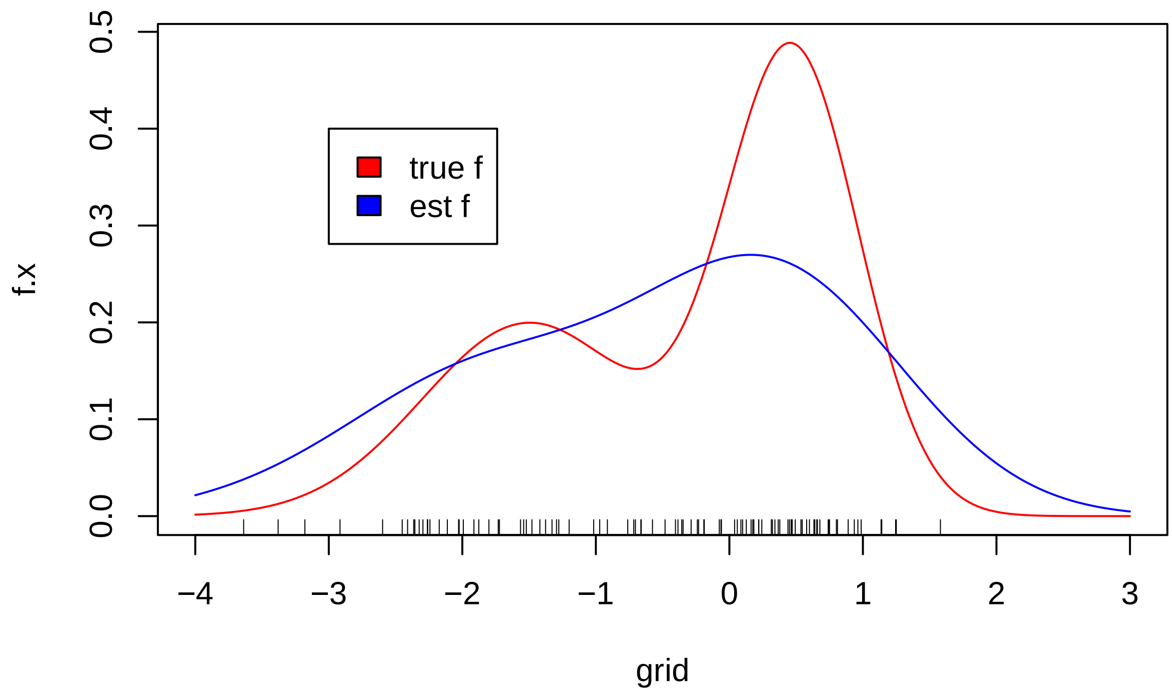


Here's a (normal) kernel smoother

```
dens.est <- function(pt,data, h)
  return(mean(dnorm((pt-data)/h))/h)
```

and it implemented

```
f.est <- c()
for (i in 1:length(grid)) {  #grid defined above
  f.est[i] <- dens.est(grid[i], x, .8)
}
plot(grid, f.x, col='red', t='l')
lines(grid, f.est, col='blue')
rug(x)
legend(-3,.4, c('true f', 'est f'), c('red', 'blue'))
```



Instead of looking at the squared error of our prediction of hidden points, we need a different idea. Come up with one (what would a bad estimate say about a future observation?) and implement it. Do you like your idea?

Instead of using mean absolute error, which does not work for 1-dimensional data, we should use the likelihood $\prod \hat{f}_h^{-i}(x_i)$ as suggested on Gradescope.

3.

Both kernel density estimation (KDE) and kernel regression rely on an approximation justified by assumed "niceness" of the function. That is that $P(a - h/2 < X < a + h/2) = \int_{a-h/2}^{a+h/2} f(x)dx \approx hf(a)$ and that $Y_i = f(x_i) + \epsilon_i \approx f(a) + \epsilon_i$ for $x_i \in (a - h/2, a + h/2)$ respectively.

Choose h too big, and this approximation gets bad and we end up with too much bias. Choose h too small, and there isn't enough data to make the estimator stable, and we end up with too much variance.

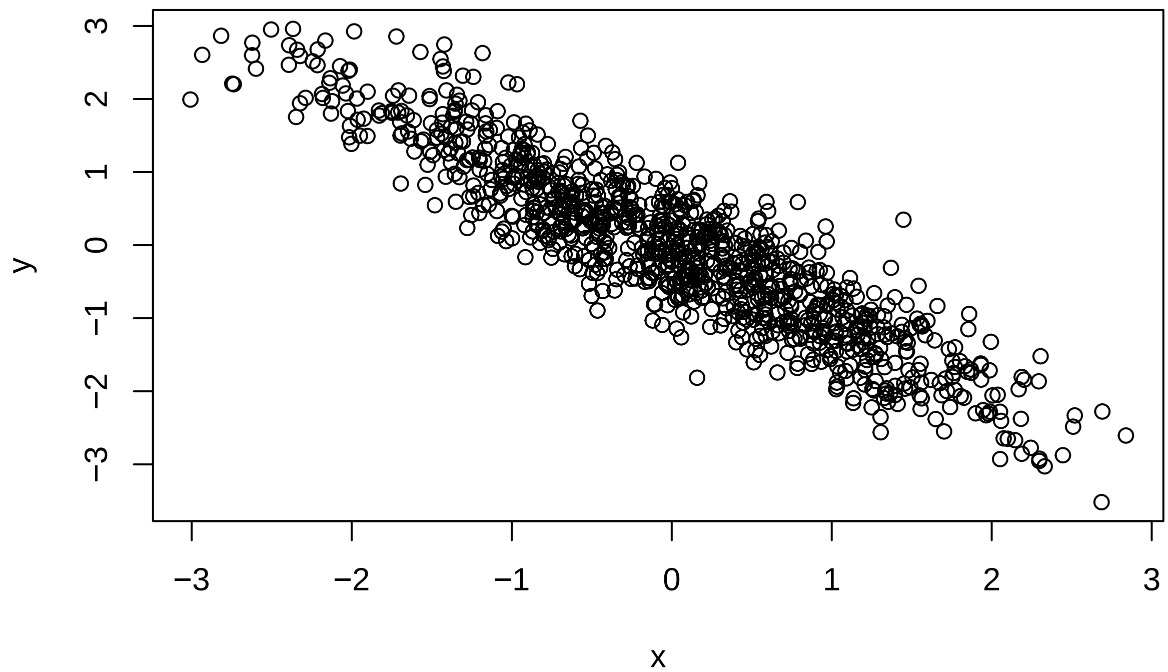
a. It looks like the bias is only going to be low if the target function f is constant (in either case), but it turns out that $f'(x)$ doesn't show up in the formula of the bias (which is in the section of the 4th notes that you aren't responsible for), rather what causes bias is large values of $f''(x)$. Justify why large values of $f'(x)$ don't cause bias, and state the property of the kernel function $K(\cdot)$ that is responsible for this.

Please see the attached proof in handwriting.

b. We can (and will) think about doing this in higher dimensions. Our neighborhoods then become 2-dimensional regions rather than intervals, and this allows some freedom in thinking what a neighborhood might look like. Two options are a square centered at a and a circle centered at a.

Suppose that the data from a bivariate density $f(x, y)$ looked like this:

```r
set.seed(47)
x <- rnorm(1000)
y <- -x + rnorm(1000,0,.5)
plot(x,y)
```



Given that KDE works well when we balance the size of the neighborhood with the quality of the approximation of the density as a constant locally, suggest a 3rd shape for the neighborhood that is well tailored to this setting.

I suggest a rhombus or an oval centered at a.