STAT 621 Lecture Notes
Nonparametric Density Estimation Part 2
Applictions, Extensions and Examples

Here we continue our discussion of nonparametric density estimation. We'll investigate the behavior of kernel density estimators using simulation, define some extensions to the basic estimator, and look at some real-data examples.

A Simulation Exercise: Recall the MISE of the density estimator $\widehat{f}(x)$,

$$\text{MISE} = E\int\{\widehat{f}(x) - f(x)\}^2 dx.$$

It can be shown that

$$\text{MISE}\{\widehat{f}(x)\} = \frac{1}{n\lambda}\int K^2(x)dx + \frac{\lambda^4}{4}\mu_2\int\{f^{"}(x)\}^2 dx$$

where $\mu_2 = \int x^2 K(x)dx$. Last time we drew some general conclusions about the asymptotic behavior of MISE as $n \to \infty$,

In some ways these arguments are impractical since we never have an infinite sample size. So let's use simulation to investigate this quantity for finite n. First consider this function to compute the ISE=$\int\{\widehat{f}(x) - f(x)\}^2 dx$.

```
ise=function(ftrue,fhat,xgrid)
{
  w=xgrid[2]-xgrid[1]
  d=fhat-ftrue
  y=w*t(d)%*%d
  y
}
```

The following function approximates MISE for the kernel density estimator on a sample of Chisquared random variables. Specifically,

- A total of `B` samples are generated
- Samples are generated from the Chisquare(`df`) distribution
- This is repeated for sample of given sizes, `n`=$(n_1, n_2, \ldots, n_k)$.

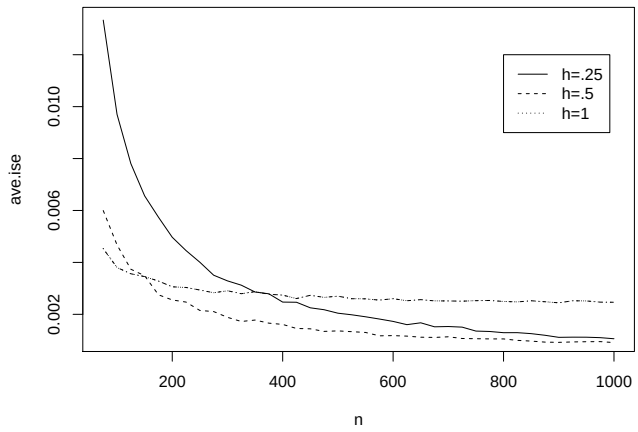
- The density function is estimated on a grid of x values (`xgrid`).
- The density function is estimated using bandwidth `lambda`.
- For each estimate $\widehat{f}(x)$, the ISE is computed with the function `ise` defined above.
- Finally, MISE is approximated as the average ISE over the B simulated data sets.

```
mise.chi=function(n,df,xgrid,B,lambda)
{
  out=rep(NA,length(n))
  f=dchisq(xgrid,df)
  for(i in 1:length(n))
  {
    temp=rep(NA,B)
    for(j in 1:B)
    {
      x=rchisq(n[i],df)
      fhat=density(x,from=min(xgrid),to=max(xgrid),n=length(xgrid),bw=lambda)
      temp[j]=ise(f,fhat$y,xgrid)
    }
    out[i]=mean(temp)
  }
  out
}
```

Now I'll run the function to estimate MISE for a range of sample sizes, with `df`=4. I repeat for a few different values of the bandwidth.

```
n=seq(75,1000,25)
xgrid=seq(0,10,.2)
ave.ise.25=mise.chi(n,4,xgrid,250,.25)
ave.ise.5=mise.chi(n,4,xgrid,250,.5)
ave.ise1=mise.chi(n,4,xgrid,250,1)

par(mfrow=c(1,1))
plot(n,ave.ise.25,type='l',ylab='ave.ise')
lines(n,ave.ise.5,lty=2)
lines(n,ave.ise1,lty=3)
legend(800,.012,c("h=.25","h=.5","h=1"),lty=1:3)
```



Comment. Is this what you would expect. Anything surprising?

Bivariate Density Estimation

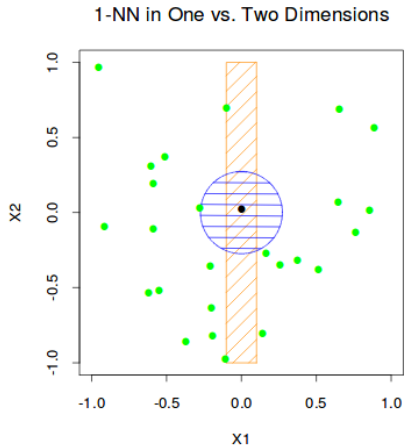
Here the objective is to estimate the joint density function $f(x_1, x_2)$ of two random variables. This is usually done using the straightforward extension of the one dimensional KDE.

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{h^2} \sum_{j=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{X}_j}{h}\right),$$

where $\boldsymbol{x} = (x_1, x_2)'$ and $\boldsymbol{X}_j = (X_{1j}, X_{2j})'$. For example, using a Gaussian product kernel and varying bandwidths yields

$$\widehat{f}(x_1, x_2) = \frac{1}{nh_1h_2} \sum_{j=1}^{n} \phi\left(\frac{x_1 - X_{1j}}{h_1}\right) \phi\left(\frac{x_2 - X_{2j}}{h_2}\right).$$

A note about sample sizes and the curse of dimensionality. This figure is from *The Elements of Statistical Learning*, Hastie et al.

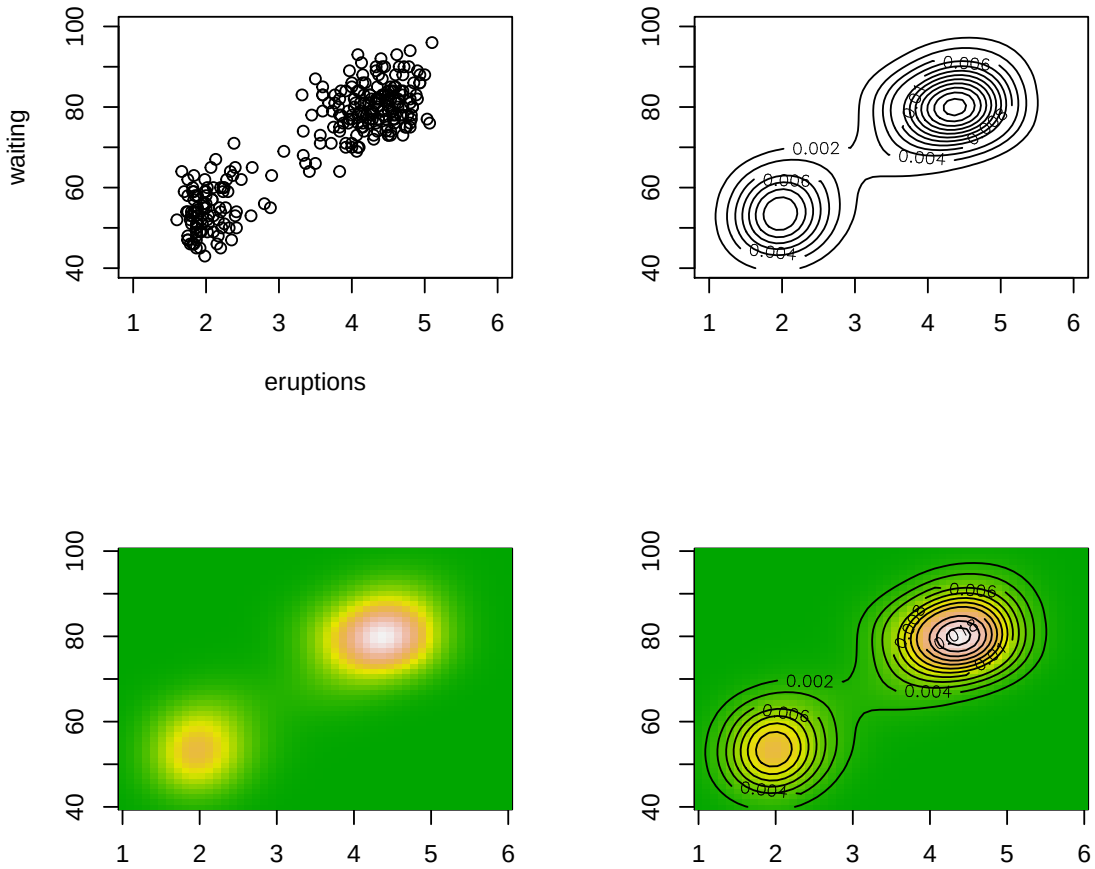


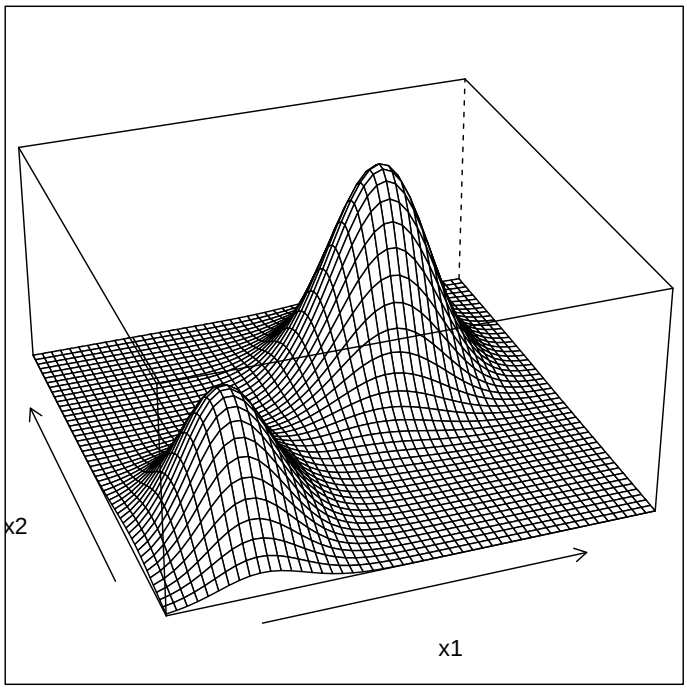
The function `kde2d` in the package `MASS` will compute this bivariate KDE. Here's an example using the Old Faithful data where waiting time between eruptions and eruption duration were measured.

```
attach(faithful)
library(MASS)
f1=kde2d(eruptions, waiting, n=50, lims=c(1,6,40,100))

par(mfrow=c(2,2))
plot(waiting~eruptions,xlim=c(1,6),ylim=c(40,100))
contour(f1,levels=seq(0,.1,by=.002))
image(f1, col=terrain.colors(1000))
image(f1, col=terrain.colors(1000))
contour(f1,levels=seq(0,.1,by=.002),add=T)

library(lattice)
wireframe(z~x*y, con2tr(f1),aspect=c(1,.5),screen=list(z=20,x=-60),zoom=1.1,xlab='x1',ylab='x2')
```

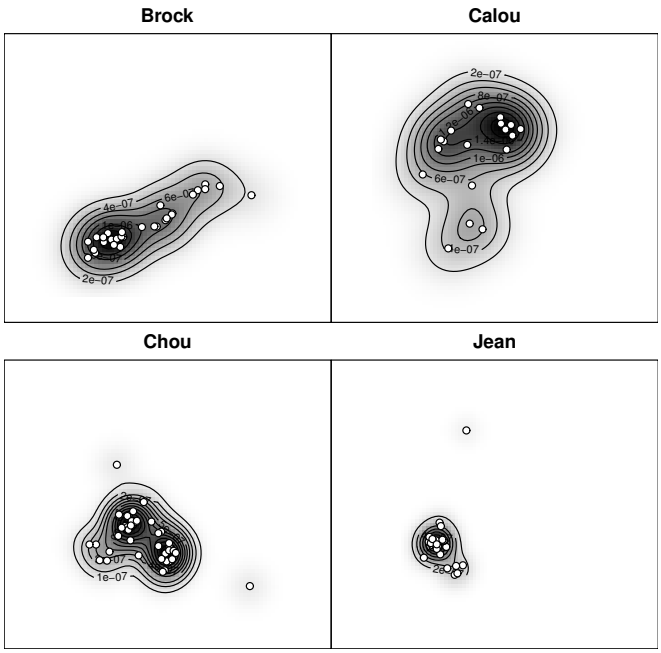




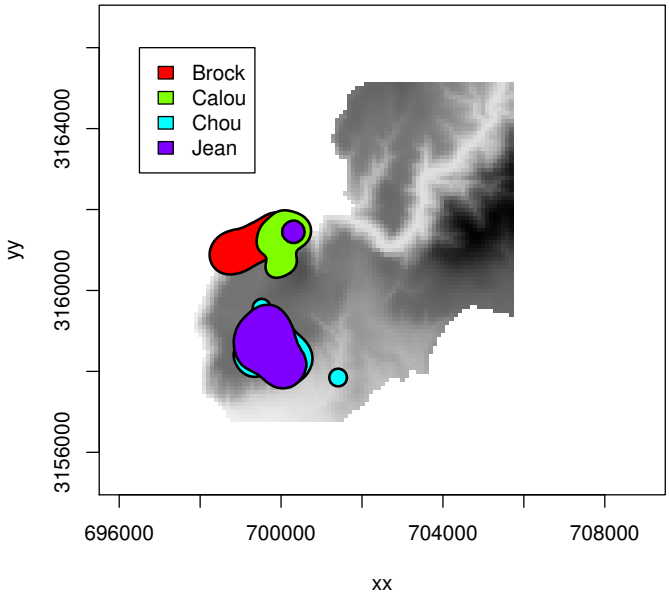
Application: Estimating animal home range.

Often a *utilization distribution* is used to describe animals' use of a spatial region. An individual animal's homerange is the subset of their utilization distribution that the animal uses most often (e.g., spends 95% of its time). Nonparametric density estimation has been use to estimate these functions. Describe the data and estimator:

An example of this is provided in the R package `adehabitat`. The data set `peuchabon` contains locations of four radio collared wild boar in France (Maillard 1996). Kernel density estimates of these individual animals' utilization distributions appear below.



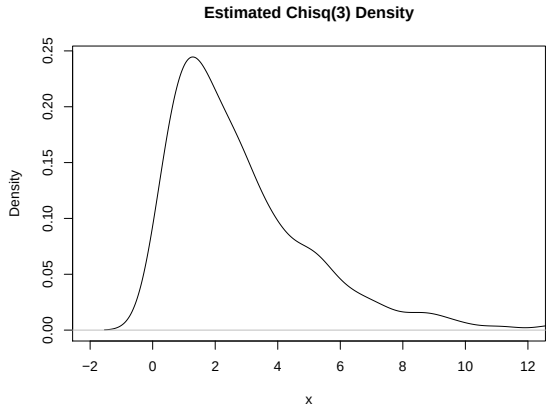
We might estimate home range for these animals as the region spanning the highest 95% of their utilization densty. Below the estimated home range for each animal is overlaid on the region where the study was conducted.



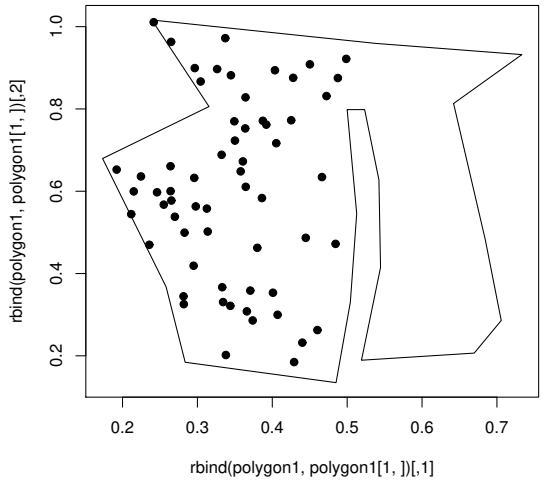
Additional Issues in Density Estimation

(1) **Boundary Bias:** Kernel density estimators have a hard time respecting boundaries. Here's an example:

```
x=rchisq(500,3)
plot(density(x),xlab='x',main="Estimated Chisq(3) Density",xlim=c(-2,12))
```



And in two dimensions, consider estimating the utilization distribution of fish located in the funny-shaped lake below. What are some possible problems?



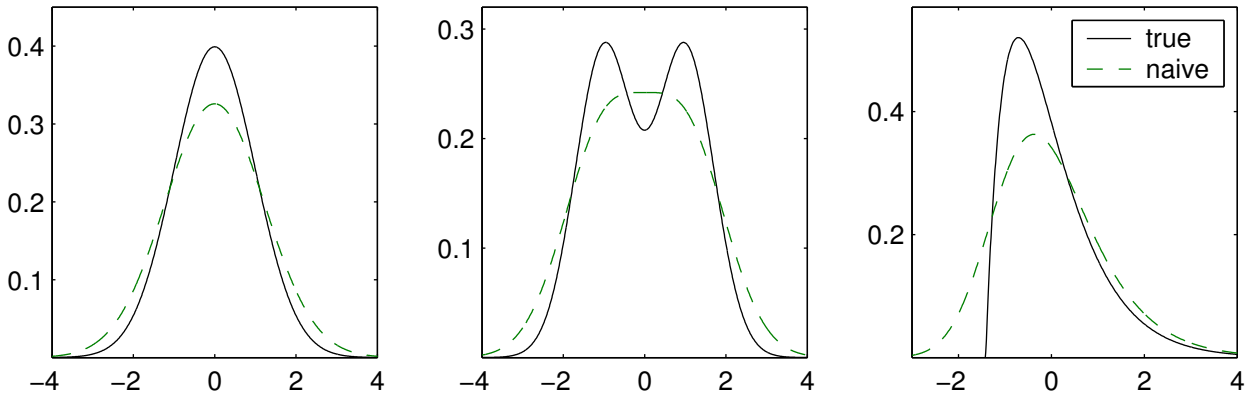
Solutions?

(2) **Measurement Error:** Suppose that the variable of interest X can't be measured precisely. Instead we observe data W_i, $i = 1, \ldots, n$ where

$$W_i = X_i + U_i.$$

The Us are random measurement errors typically assumed to have mean zero and are independent of the Xs. Now suppose we use these observed data in our kernel density estimator. That is we naively ignore the measurement error. What will be the problem?

A picture:



Solutions?