Kernel density estimation and classification

Patrick Breheny

November 1

Introduction

- The idea of using kernels to combine local estimates in a smooth way has applications beyond that of regression and GAMs
- Another application for which they are widely used is in density estimation: given a set of iid observations $\{y_i\}$, can we estimate the density of the underlying distribution, $f(y)$?
- Similar to the case for regression, the parametric approach is to assume a specific form (*e.g.*, normal) for f, whereby the problem reduces to one of estimate a small number of unknown parameters (*e.g.*, μ and σ^2)

Kernel density estimates

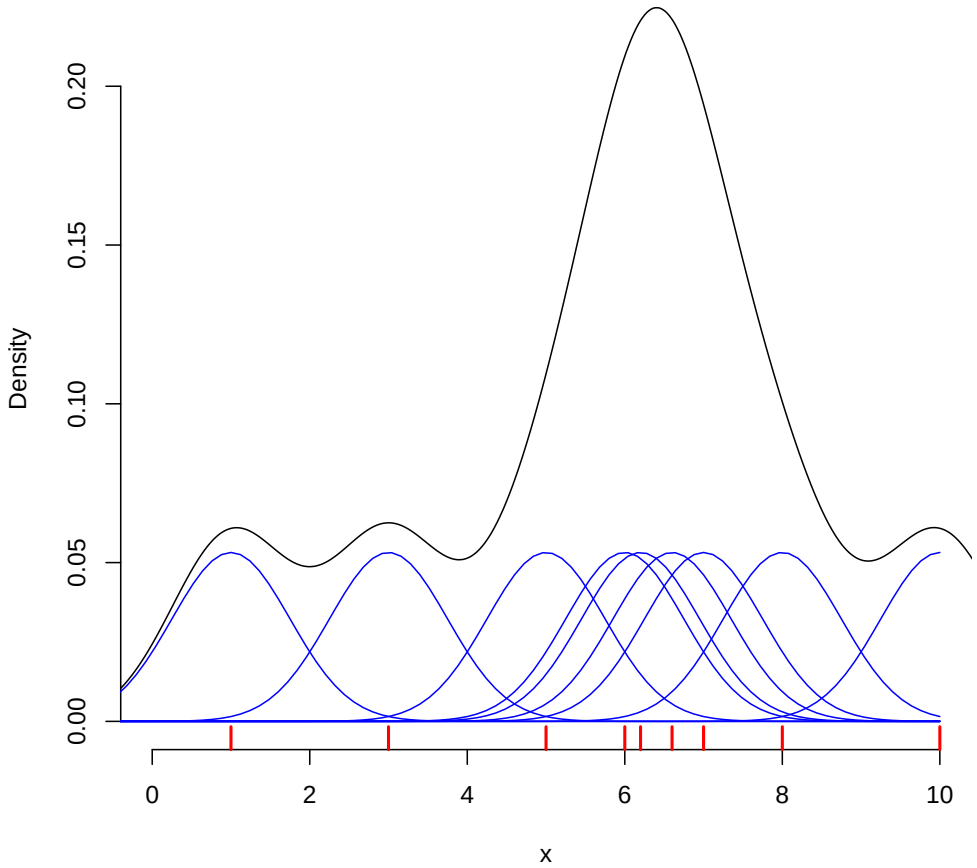
- Kernels can be used to construct nonparametric estimates for f that make no such assumptions
- Specifically, consider estimators of the following form:

$$\hat{f}(x_0) = \frac{1}{n\lambda} \sum_i K\left(\frac{x_i - x_0}{\lambda}\right),$$

  where the kernel K integrates to 1
- NOTE: For the definitions given in the 10-20 notes, the tri-cube kernel does not integrate to 1, although it can be made to do so by including a normalizing constant; this constant is already included for the Epanechnikov kernel

Gaussian kernel: density estimate

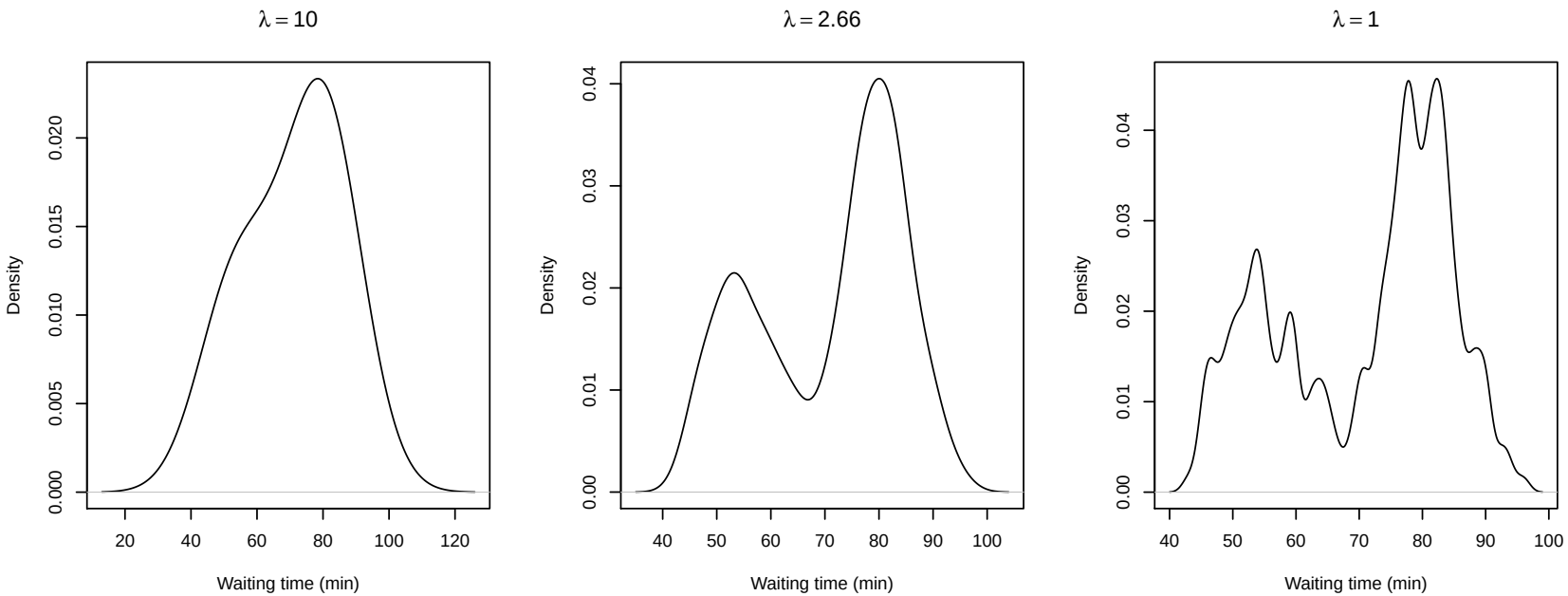
An example using the Gaussian density as the kernel function



Example: Old Faithful eruptions

- As an example of a density estimation problem, consider data collected on the waiting times between eruptions of the Old Faithful geyser in Yellowstone National Park
- The data were collected from August 1 to August 15, 1985, by the park's geologist

Density estimates at different bandwidths



Automatic selection of bandwidth

- A surprisingly large number of methods for choosing an optimal bandwidth have been developed
- The approaches are based on minimizing the mean integrated squared error:

$$\int \mathrm{E}\{\hat{f}(x) - f(x)\}^2 dx,$$

  or rather, an asymptotic approximation of it
- The problem, as we have encountered many times already, is that this expression contains a bias term which depends on $\int f''(x)^2 dx$, and is therefore impossible to evaluate without knowing the true f

Normal reference rule

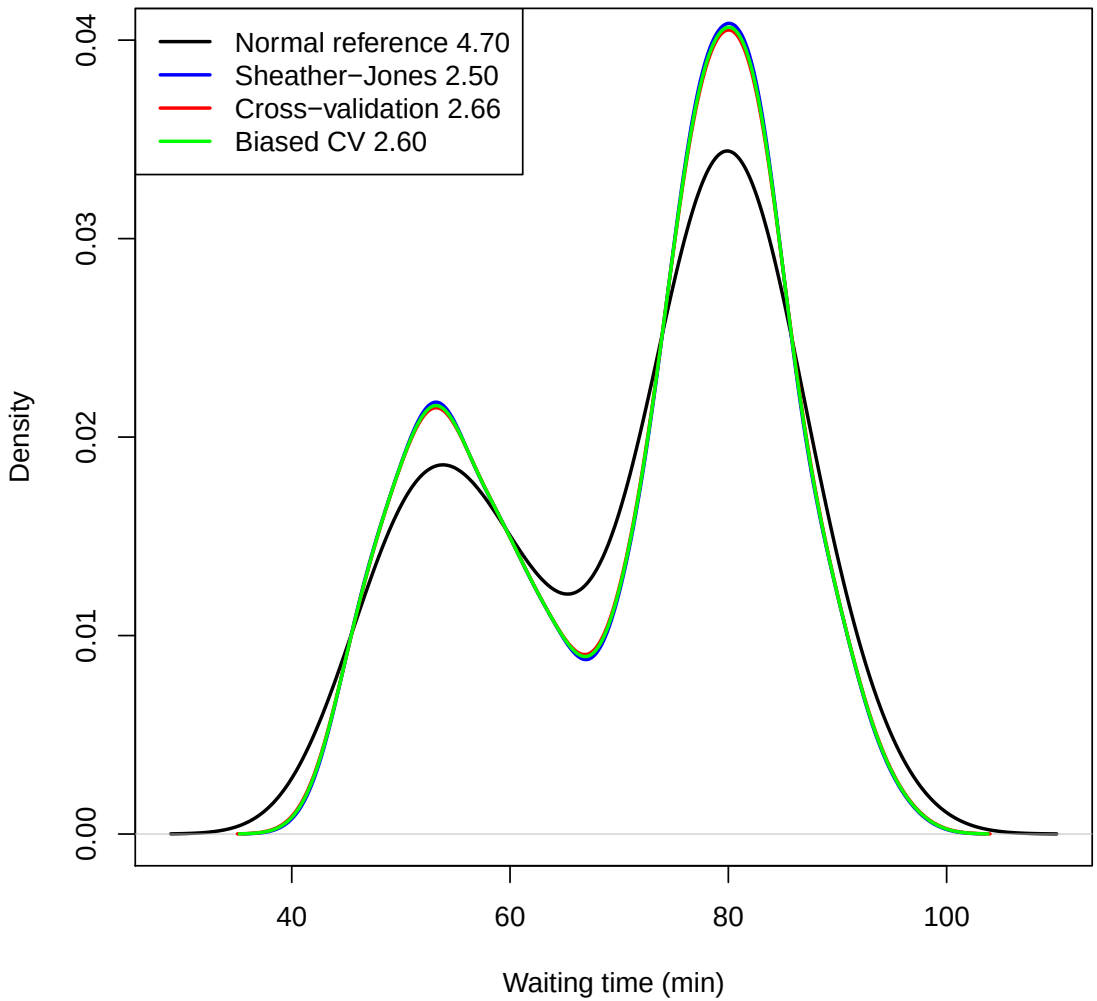
- One simple solution is to use the value of $\int f''(x)^2 dx$ for the normal distribution
- This approach is called the *normal reference rule*
- This is a convenient rule of thumb, but if the true f is very different from the normal, this can result in an oversmoothed density

Other approaches

- An alternative approach comes from Sheather and Jones (1991), who use a so-called "pilot bandwidth" λ_0 to estimate $\int f''(x)^2 dx$
- Another method called *biased cross-validation* is based on setting up a grid of λ values and then calculating $\int f''_\lambda(x)^2 dx$ for each value (the name is a misnomer, cross-validation is not actually involved)
- Finally, you can actually use cross-validation and minimize the quantity

$$\int \hat{f}^2(x)dx - 2\sum_i \frac{1}{n}\hat{f}_{(-i)}(x_i)$$

Bandwidth selection methods: Example



Remarks

- Note that the normal reference approach produces the smoothest estimate; this is true in general
- In this particular example, there is no meaningful difference between cross-validation/Sheather-Jones/BCV, although this is not always the case
- Indeed, many authors find cross-validation unsatisfactory, in that it often produces an estimate quite a bit "rougher" than a person would obtain from visually smoothing the histogram
- However, it should be noted that unlike the others, cross-validation is an unbiased estimate of the optimal bandwidth, regardless of how smooth the underlying density is (provided, of course, that it is still continuously differentiable)

PROC KDE

- Kernel density estimates are available in SAS via `PROC KDE`:

```
ods graphics on;
proc kde data=faithful;
   univar waiting / plots=(density histogram histdensity);
run;
ods graphics off;
```

- By default, `PROC KDE` uses a Gaussian kernel and selects λ using the Sheather-Jones approach
- You can change how λ is selected via the `/ METHOD=` option, but the Gaussian is the only available kernel
- You can also manually adjust λ by specifying `/ BWM=2` to select a bandwidth twice as large as the original

The density function

- Kernel density estimates are available in R via the density function:

```
d <- density(faithful$waiting)
plot(d)
```

- By default, density uses a Gaussian kernel, but a large variety of other kernels are available by specifying the kernel option
- By default, density selects λ using normal reference, but again, other options are available: bw=''SJ'' for Sheather-Jones, bw=''bcv'' for biased cross-validation, and bw=''ucv'' for regular (unbiased) cross-validation
- In addition, you can directly specify bandwidth (bw=3) or adjust the original via adjust, which works exactly like SAS's BWM

Exercise

Exercise: The course website contains a data set from the National Health and Nutrition Examination Survey (NHANES) that lists the triglyceride levels of 3,026 adult women.

(a) Obtain a kernel density estimate for the distribution of triglyceride levels in adult women and plot it. You are free to decide on whatever kernel and bandwidth you like, but describe which ones you used.

(b) Obtain a parametric density estimate assuming that triglyceride levels follow a normal distribution and overlay this density estimate with your estimate from (a).

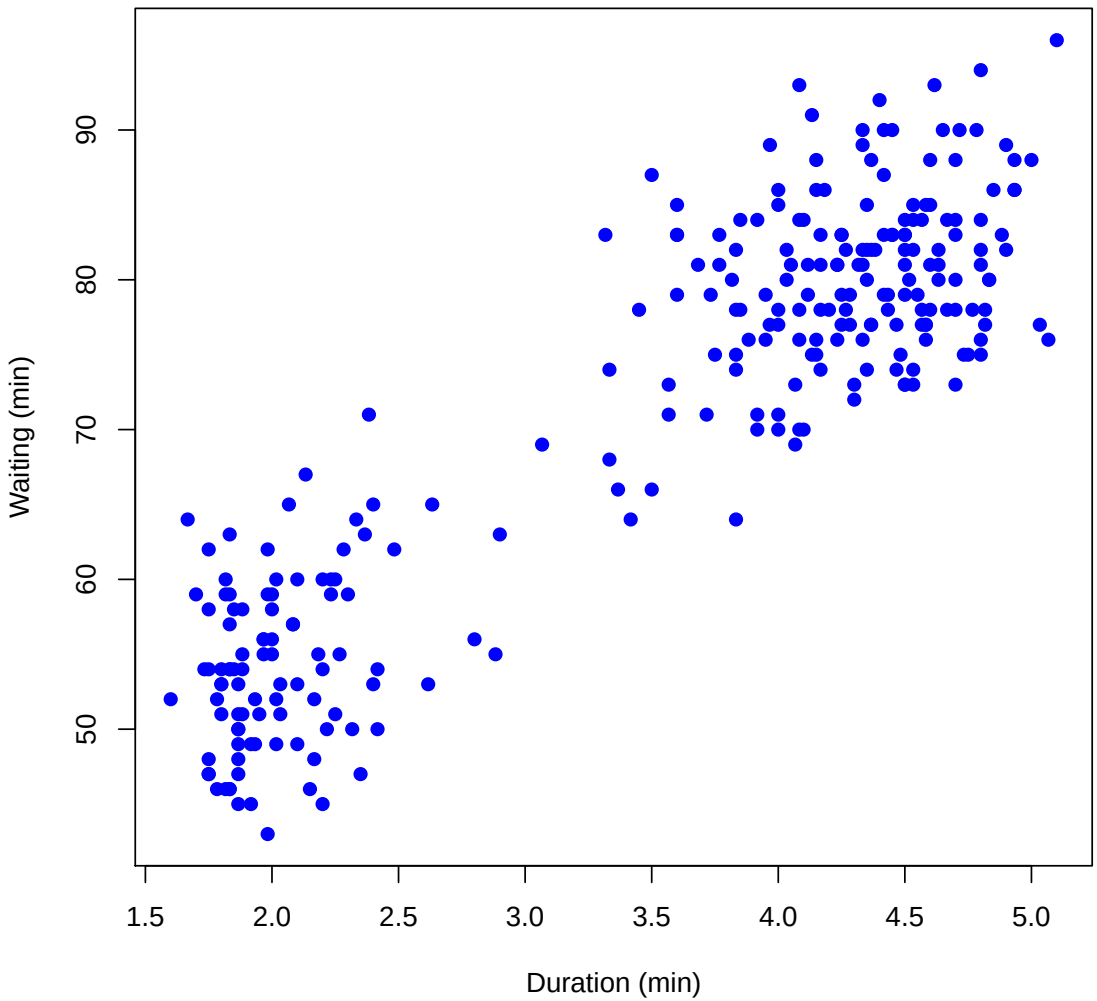
Multivariate densities

- As we discussed last week, it is straightforward to extend the idea of kernels to multiple dimensions:

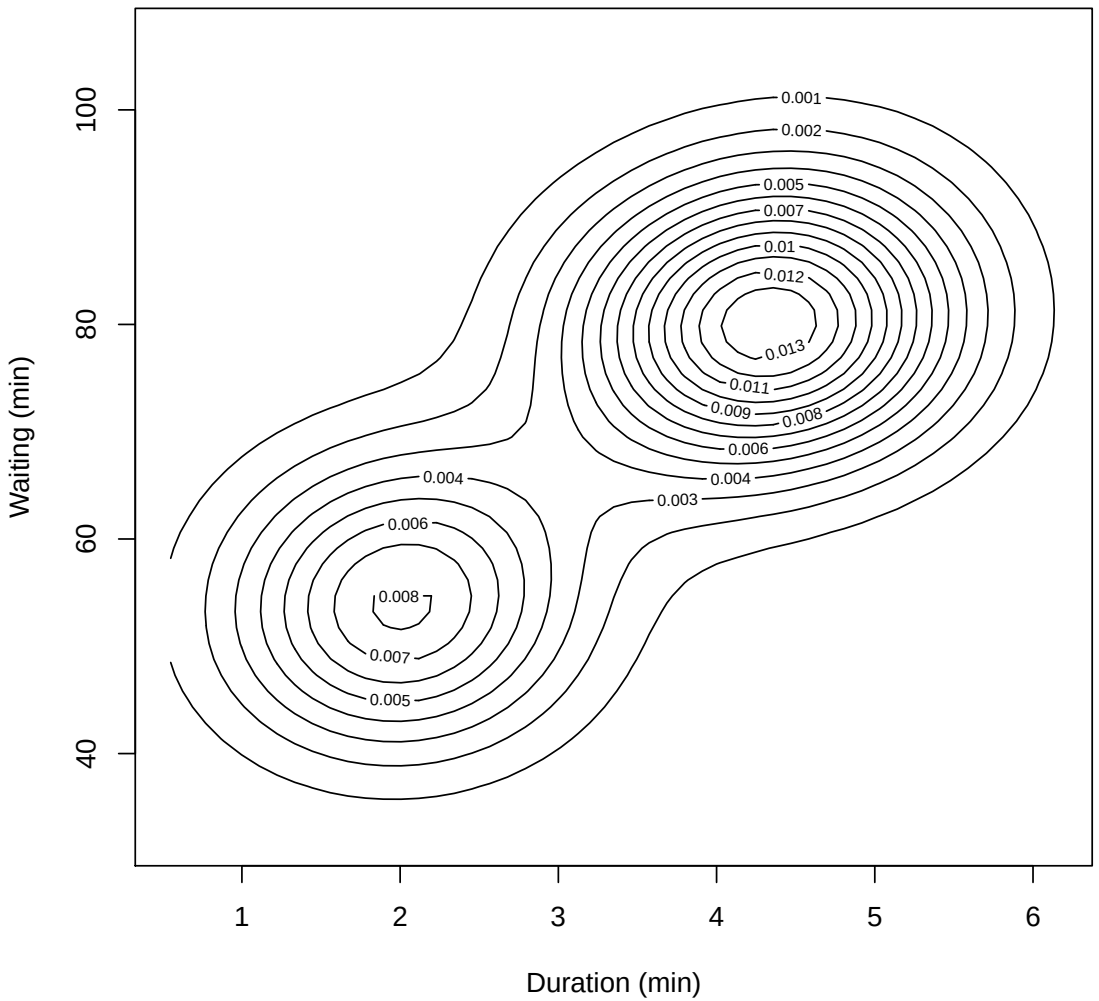
$$\hat{f}(\mathbf{x}_0) = \frac{1}{n} \sum_i \prod_{j=1}^{p} \frac{1}{h_j} K\left(\frac{x_{ij} - x_{0j}}{h_j}\right)$$

  where p is the dimension of $\mathbf{x}$
- In the Old Faithful data set, two variables are recorded: duration of the eruption and the waiting time until the next eruption

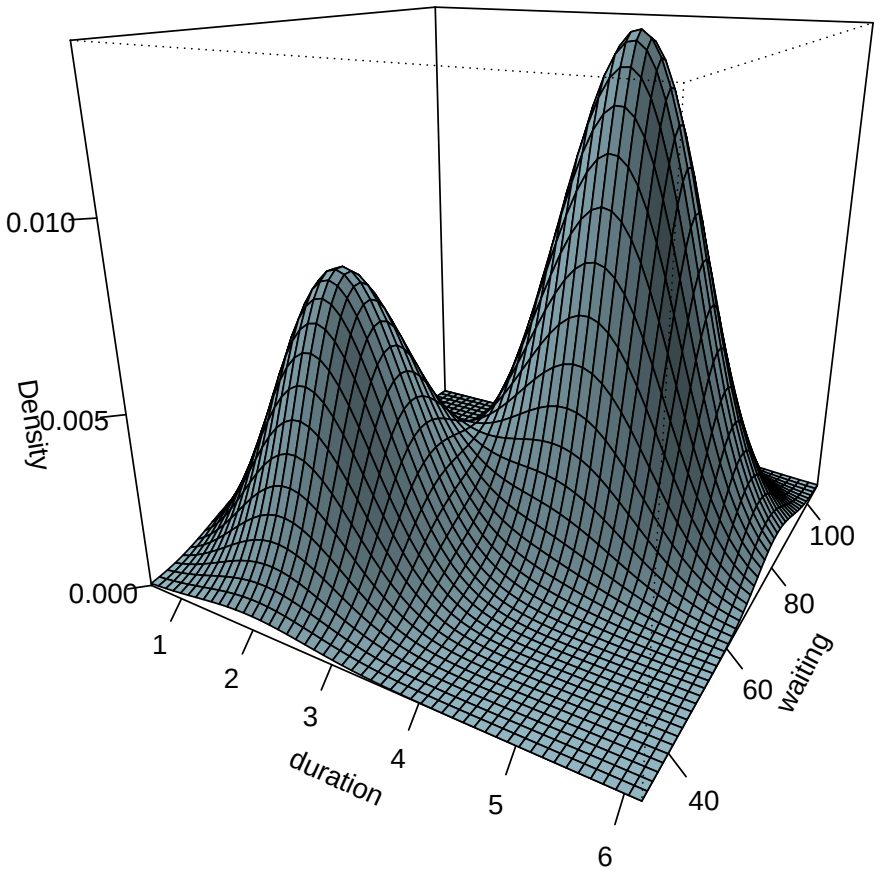
Old faithful data: Duration vs. waiting time



Old faithful 2D density estimate: contour plot



Old faithful 2D density estimate: perspective plot



Implementation and limitations

- The `density` function is exclusively for one-dimensional kernel density estimation, but 2D density estimates like the ones just presented are available via the `KernSmooth` package in `R`; in SAS, simply replace the `UNIVAR` statement with a `BIVAR` statement
- As we have discussed, although one can easily write down an expression for the kernel density estimate in higher dimensions, the statistical properties of the estimator worsen rapidly as p grows

Classification

- Density estimation is interesting in its own right, but also a means for classification
- Recall that in LDA, we carried out classification by Bayes' rule:

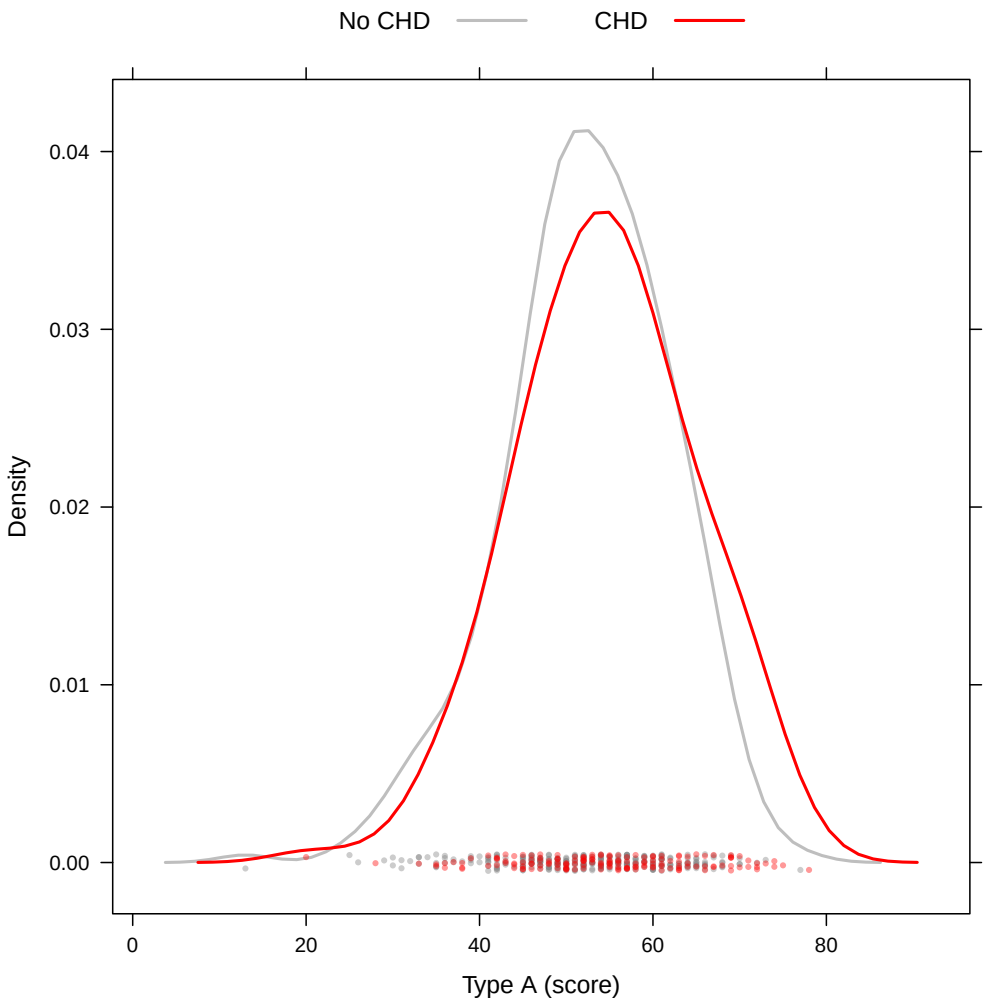
$$\Pr(G = k|\mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_l f_l(\mathbf{x})\pi_l},$$

  where the densities $\{f_k\}$ were estimated based on assuming multivariate normality
- Alternatively, we could estimate them using kernel density estimates

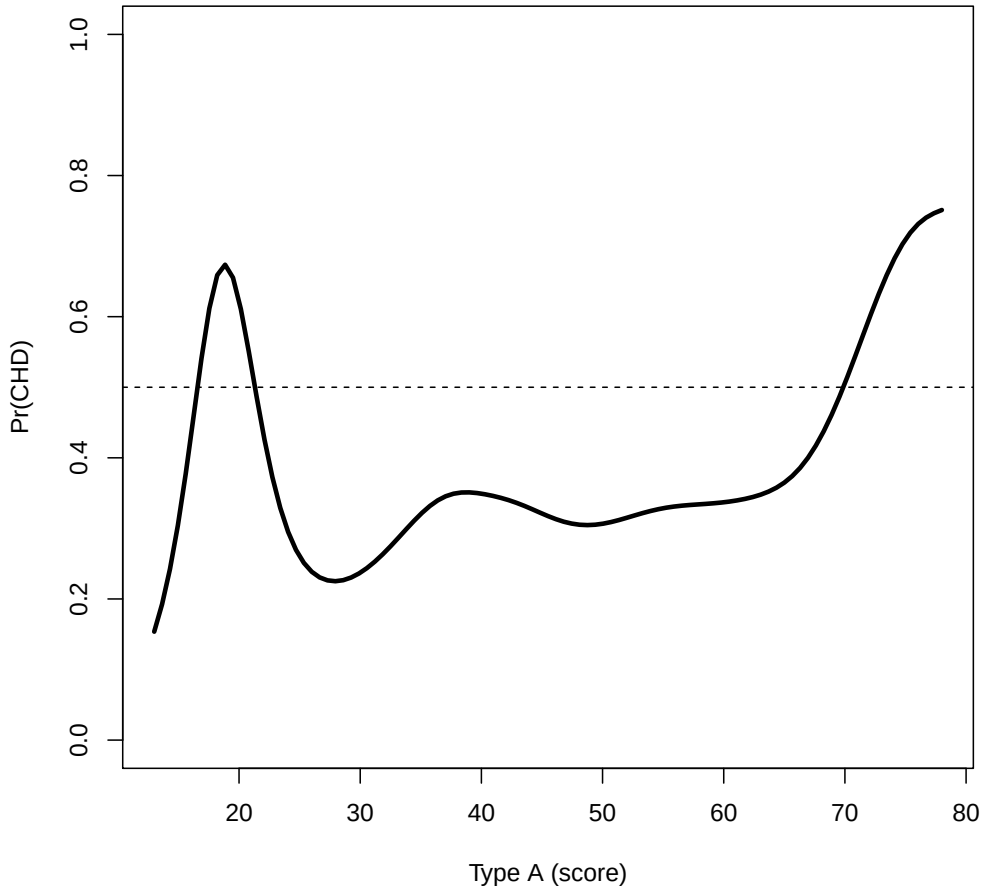
Coronary heart disease study

- For example, recall our study of coronary heart disease
- Let's focus on the relationship between CHD and stress (type A score)

Kernel density estimates



Estimate of posterior probability



Evaluation

- As we can see, unlike LDA, the kernel density classifier is not restricted to a linear function, although it seems rather unstable in regions where there is little data
- Furthermore, as we have seen, there will be many regions with little data when we move to higher dimensions

The independence assumption

- Thus, the simplifying assumption of independence is often made:

$$\hat{f}_k(\mathbf{x}) = \prod_{j=1}^{p} \hat{f}_{kj}(x_j),$$

  where $\hat{f}_{jk}$ is an estimate of the density of the jth variable for the kth class
- This assumption is, generally speaking, not true, and the above estimate goes by the not particularly inspiring name of the *naive Bayes classifier*
- However, it drastically reduces variance and alleviates the curse of dimensionality and often performs well as a classifier

Connection to additive models

- Finally, note that for the naive Bayes classifier,

$$\text{logit}(y = 1|\mathbf{x}) = \beta_0 + \sum_{k=1}^{K} g_j(x_j)$$

- Thus, the naive Bayes classifier is a way of constructing an additive logistic regression model, with flexible functions g_k determining the impact of x_k on the log-odds that $y = 1$