Kernel density estimation

Patrick Breheny

October 18

Introduction

- We've looked at one method for estimating density: histograms
- Histograms are based on estimating a *local* density; in their case, points are local to each other if they fall into the same bin
- However, bins are a rather inflexible way of implementing local density estimation

Drawbacks of histograms

- This leads to some undesirable properties; for example
  - The resulting estimate is not smooth
  - An observation may be closer to an observation in the neighboring bin than it is to points in its own bin
- Today we will introduce and discuss methods for taking weighted local density estimates at each observation x_i and then aggregating them to yield an overall density

Local neighborhood density

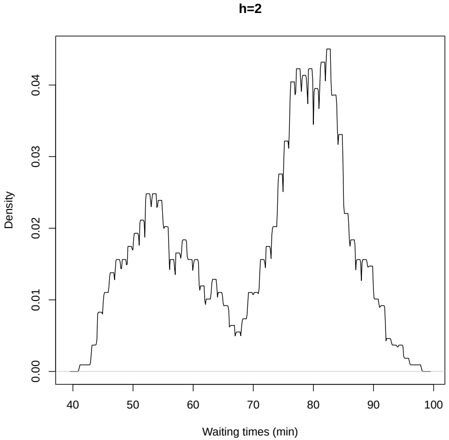
- For example, consider estimating the density at a point x_0 by taking the local density of the points within distance h of x_0:

$$\hat{f}(x_0) = \frac{n^{-1}\sum_i I(|x_i - x_0| \leq h)}{2h}$$

- This solves one problem of the histogram – namely, it ensures that no point further away from x_0 than x_i will contribute more than x_i does to the density estimate

Local neighborhood density: Old Faithful data

However, the resulting density estimate is still bumpy:



Kernel density estimate

- Let's fix the bumpiness: instead of giving every point in the neighborhood equal weight, let's assign a weight which dies off toward zero in a continuous fashion as we get further away from the target point x_0
- Specifically, consider estimators of the following form:

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

  where h, which controls the size of the neighborhood around x_0, is the smoothing parameter
- The function K is called the *kernel*, and it controls the weight given to the observations $\{x_i\}$ at each point x_0 based on their proximity

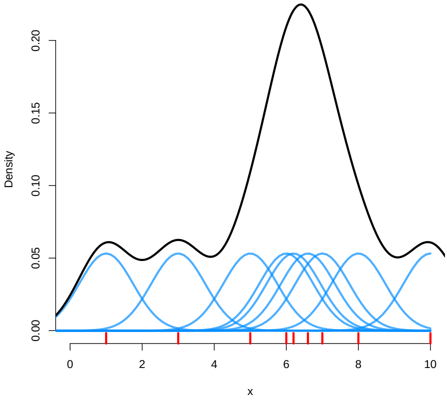
Kernel properties

To yield meaningful estimates, a kernel function must satisfy four properties:

- $K(u) \geq 0$
- Symmetric about 0
- $\int K(u)du = 1$
- $\int u^2 K(u)du > 0$

Gaussian kernel: density estimate

An example of a kernel function is the Gaussian density



Other kernels

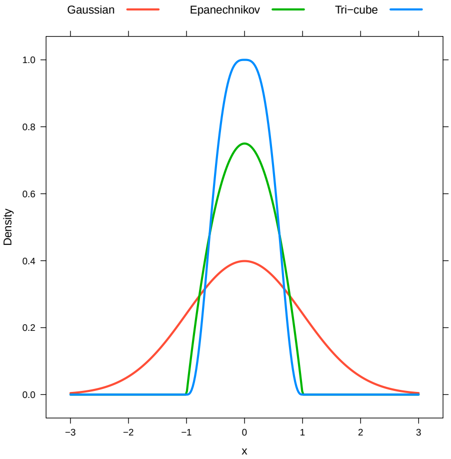
- One possible drawback of the Gaussian kernel is that its support runs over the entire real line; occasionally it is desirable that a kernel have compact support
- Two popular compact kernels are the Epanechnikov kernel:

$$K(u) = \begin{cases} \frac{3}{4}(1-u^2) & if\ |u| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

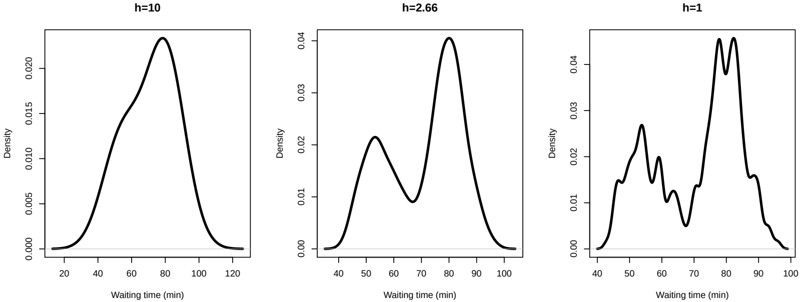
and the tri-cube kernel:

$$K(u) = \begin{cases} (1-|u|^3)^3 & if\ |u| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Kernels: illustration



Effect of changing bandwidth



Consistency

- For a fixed h, kernel density estimates are not consistent
- However, if the bandwidth decreases with sample size at an appropriate rate, then they are, regardless of which kernel is used
- **Theorem:** Suppose that f is continuous at x, that $h_n \to 0$, and that $nh_n \to \infty$ as $n \to \infty$. Then $\hat{f}(x) \xrightarrow{\mathsf{P}} f(x)$.

Rate of convergence

Theorem: Suppose f'' is absolutely continuous and that $\int f'''(x)^2 dx < \infty$. Then

$$\mathbb{E}L(h) = \frac{1}{4}\sigma_K^4 h^4 \int f''(x)^2 dx + \frac{\int K^2(x)dx}{nh} + O(n^{-1}) + O(h^6),$$

where $\sigma_K^2 = \int x^2 K(x)dx$. Furthermore, the optimal bandwidth h is given by

$$h_* = \left(\frac{\int K^2(x)dx}{n\sigma_K^4 \int f''(x)^2 dx}\right)^{1/5}.$$

For this choice of bandwidth, $\mathbb{E}L = O(n^{-4/5})$.

Cross-validation

- There are two common approaches to choosing an appropriate bandwidth
- The first is cross-validation, which works exactly as it did for histograms:

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

Normal reference rule

- The second approach is to use the formula

$$h_* = \left(\frac{\int K^2(x)dx}{n\sigma_K^4 \int f''(x)^2 dx} \right)^{1/5}$$

  and calculate $\int f''(x)^2 dx$ from the normal distribution
- 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

Cross-validation and optimal bandwidths

On the other hand, cross-validation has very attractive theoretical properties regardless of f:

Theorem: Suppose that f is bounded and let $\hat{h}$ denote the optimal bandwidth as estimated by cross-validation. Then

$$\frac{\int \{f(x) - \hat{f}_{\hat{h}}(x)\}^2 dx}{\inf_h \int \{f(x) - \hat{f}_h(x)\}^2 dx} \xrightarrow{\text{a.s.}} 1$$

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

Bandwidth specification

- By default, `density` selects the bandwidth based on the normal reference rule
- However, you can manually choose the bandwidth by specifying, for example, `bw=4`
- You can also obtain automatic selection by cross-validation by specifying `bw=''ucv''`
- As an example of the oversmoothing that was alluded to earlier, for the Old Faithful data the normal reference rule chooses a bandwidth of 4.70, compared to a bandwidth of 2.66 chosen by cross-validation

Computational considerations

- Kernel density estimation can be implemented very efficiently using the fast Fourier transform
- This imposes a computational burden on the order of n operations
- If cross-validation is used, then the cost increases to $O(n^2)$ operations
- If the normal reference rule is used, the cost remains $O(n)$

Homework

Homework: 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).

Homework

Homework: Try to obtain a kernel density estimate for the nerve pulse data on the course website, with bandwidth chosen by cross-validation. You will receive a warning message, and your estimate will appear clearly incorrect.

(a) What's going on? What is causing this problem?

(b) Fix the problem and obtain a reasonable-looking estimate of the density of waiting times between nerve pulses.

Multivariate densities

- It is straightforward to extend the idea of kernel density estimation to obtain multidimensional densities:

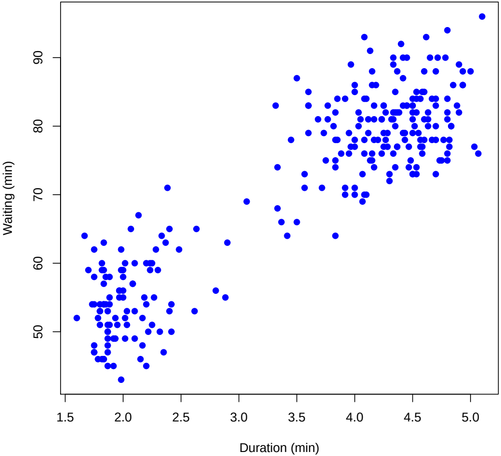
$$\hat{f}(\mathbf{x}_0) = \frac{1}{nh^p} \sum_i K\left(\frac{\|\mathbf{x}_i - \mathbf{x}_0\|}{h}\right),$$

  where p is the dimension of $\mathbf{x}$

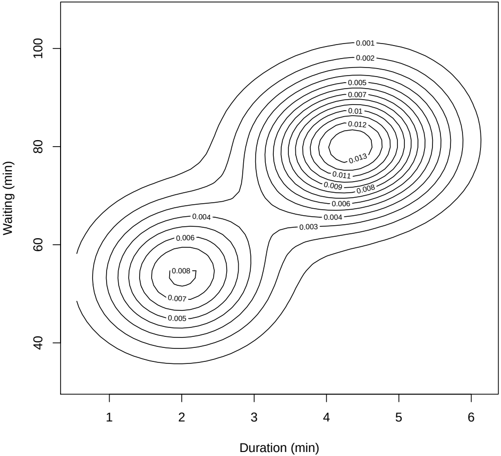
- This can be further generalized by allowing different bandwidths in each dimension:

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

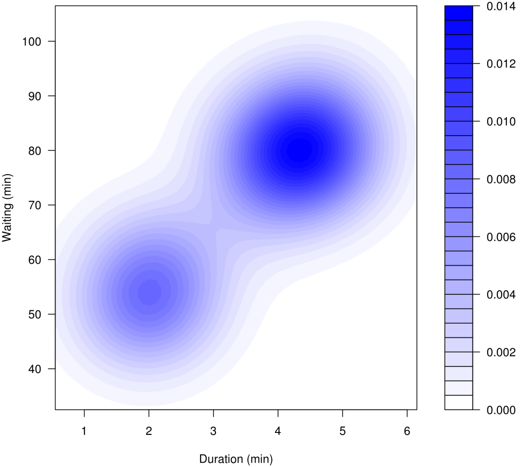
Old faithful data



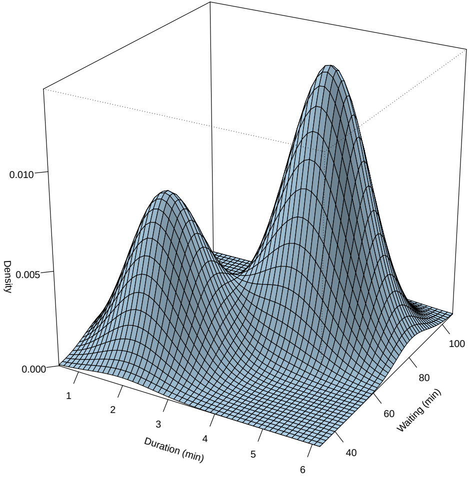
Old faithful 2D density estimate: contour plot



Old faithful 2D density estimate: filled contour plot



Old faithful 2D density estimate: perspective plot



Limitations

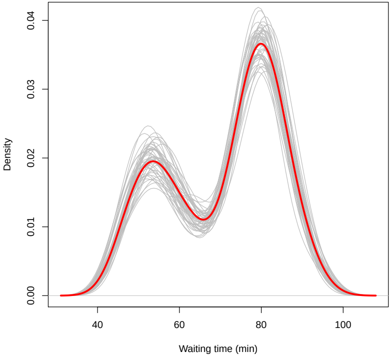
- The `density` function is exclusively for one-dimensional kernel density estimation, but 2D density estimates like the ones just presented are available via the `KernSmooth` package
- The package is limited, however, in that it does not provide automatic methods for choosing bandwidths and it only extends to the 2D case
- 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

The curse of dimensionality

- For example, suppose that x follows a uniform distribution on the unit p-cube, and we choose $h = 0.05$
- When $p = 1$ and $n = 100$, we can expect 10 points in the neighborhood of x_0
- When $p = 2$, we need 1000 observations to get 10 points worth of data in a neighborhood of the same size
- When $p = 5$, we need 1,000,000 observations
- This phenomenon is commonly referred to as the *curse of dimensionality*, and we will return to it again in the course

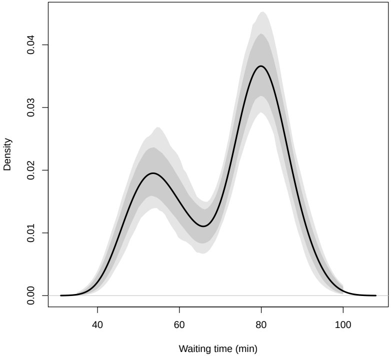
Assessing variability with the bootstrap

Closed-form confidence intervals for kernel density estimates are not trivial, but the bootstrap can be used to assess variability:



Assessing variability with the bootstrap (cont'd)

And, with the same caveats as histograms, we can get pointwise 95% confidence intervals and bands:



Adaptive kernels

- We have discussed kernel density estimates with fixed bandwidth
- Potentially, this is suboptimal, as an appropriate bandwidth in an area of high density is not necessarily an appropriate choice in a low-density region
- Another possibility is *adaptive bandwidth* kernel estimators, in which the bandwidth changes as a function of x_0
- The idea is to have local estimates in regions where there is ample data, but to expand the neighborhood in regions where data is more scarce
- In terms of the bias-variance tradeoff, these estimators introduce added bias in region with little data in order to reduce variance there

Homework

Homework: In proving the theorem on slide 13, we derived the variance of $\hat{f}(x)$ (approximately; for the sake of this problem you may ignore the remainder term).

(a) How does the variance of $\hat{f}(x)$ change as a function of x?

(b) How does the variance of $\hat{f}(x)/f(x)$ change as a function of x?

(c) It is sometimes claimed that methods using an adaptive bandwidth (in which h changes as function of x) correct for the tendency of fixed-bandwidth estimators to have high variance in regions with little data. Are such claims referring to the variance of the density itself or the relative accuracy of the density?

Boundary issues

- Another problem with kernel density estimation occurs at the boundaries of the range of the data
- For example, the course website contains a data set with measurements of the rainfall from 26 clouds (for the purposes of this assignment, ignore the "Seeded" column)
- Most clouds gave off very little precipitation, so the density near 0 is high
- Standard kernel approaches produce an estimate of the density $\hat{f}$ that is high near zero and – this is the problem – below zero

Homework

- Obviously, rainfall cannot be negative, so this estimate is unappealing
- One solution is to take a log transformation of the data, fit the density on this scale, then transform this density back to the original scale
- **Homework:** Estimate the probability density of rainfall amount based on the cloud data in three ways: (a) directly, (b) based on the log transformation approach just described, and (c) by reflecting the estimated density that lies to the left of 0 about 0. Plot these estimates.