Chapter 6 – Kernel smoothing methods in the book “The elements of statistical learning”

Wenjing Liao
School of Mathematics
Georgia Institute of Technology

Math 4803
Fall 2019
Outline

1. 6.1 – One-dimensional kernel smoothers

2. 6.2 – Selecting the width of the kernel

3. Local regression in $\mathbb{R}^p$
From kNN to kernel regression

\[ \hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x)) \]

where \( N_k(x) \) is the set of \( k \) training points nearest to \( x \).

**Does a weighted average work better?**
kNN and a weighted average

**FIGURE 6.1.** In each panel 100 pairs \( x_i, y_i \) are generated at random from the blue curve with Gaussian errors: \( Y = \sin(4X) + \varepsilon, X \sim U[0, 1], \varepsilon \sim N(0, 1/3) \). In the left panel the green curve is the result of a 30-nearest-neighbor running-mean smoother. The red point is the fitted constant \( \hat{f}(x_0) \), and the red circles indicate those observations contributing to the fit at \( x_0 \). The solid yellow region indicates the weights assigned to observations. In the right panel, the green curve is the kernel-weighted average, using an Epanechnikov kernel with (half) window width \( \lambda = 0.2 \).
Weighted average

Kernel weighted average

\[ \hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i)y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}, \]

\[ K_\lambda(x_0, x) = D \left( \frac{|x - x_0|}{\lambda} \right), \]

\[ D(t) = \begin{cases} 
\frac{3}{4} (1 - t^2) & \text{if } |t| \leq 1; \\
0 & \text{otherwise.} 
\end{cases} \]

More generally

\[ K_\lambda(x_0, x) = D \left( \frac{|x - x_0|}{h_\lambda(x_0)} \right). \]

- \( h_\lambda(x_0) = \lambda \) above
About kernel average

1. The smoothing parameter $\lambda$, which determines the width of the local neighborhood, has to be determined.

2. Boundary issue arises. The neighborhoods tend to contain less points on the boundaries.

3. Issues arise with nearest-neighbors when there are ties in the $x_i$. With most smoothing kernel techniques one can reduce the data set by averaging the $y_i$ at tied values of $X$.

4. There is a rich class of kernels.

   - tri-cube function
   
   $D(t) = \begin{cases} 
   (1 - |t|^3)^3 & \text{if } |t| \leq 1; \\
   0 & \text{otherwise}
   \end{cases}$
FIGURE 6.2. A comparison of three popular kernels for local smoothing. Each has been calibrated to integrate to 1. The tri-cube kernel is compact and has two continuous derivatives at the boundary of its support, while the Epanechnikov kernel has none. The Gaussian kernel is continuously differentiable, but has infinite support.
6.1.1 – Local linear regression

\[
\min_{\alpha(x_0), \beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) [y_i - \alpha(x_0) - \beta(x_0)x_i]^2.
\]

Estimator: \( \hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0 \)

**How to solve it?** Let \( b(x)^T = (1, x) \in \mathbb{R}^{1 \times 2}, \ B \in \mathbb{R}^{N \times 2} \) be the regression matrix with \( i \)-th row \( b(x_i)^T \), and \( W \in \mathbb{R}^{N \times N} \) be the diagonal matrix with \( i \)-th diagonal element \( K_\lambda(x_0, x_i) \). Then

\[
\hat{f}(x_0) = b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y
\]

\[
= \sum_{i=1}^{N} l_i(x_0) y_i.
\]
Example

FIGURE 6.3. The locally weighted average has bias problems at or near the boundaries of the domain. The true function is approximately linear here, but most of the observations in the neighborhood have a higher mean than the target point, so despite weighting, their mean will be biased upwards. By fitting a locally weighted linear regression (right panel), this bias is removed to first order.
**Weight** \( l_i(x_0) \)

**FIGURE 6.4.** The green points show the equivalent kernel \( l_i(x_0) \) for local regression. These are the weights in \( \hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0)y_i \), plotted against their corresponding \( x_i \). For display purposes, these have been rescaled, since in fact they sum to 1. Since the yellow shaded region is the (rescaled) equivalent kernel for the Nadaraya–Watson local average, we see how local regression automatically modifies the weighting kernel to correct for biases due to asymmetry in the smoothing window.
Why from linear to quadratic?

**FIGURE 6.5.** Local linear fits exhibit bias in regions of curvature of the true function. Local quadratic fits tend to eliminate this bias.
6.1.2 – Local polynomial regression

Estimator: \( \hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}_j(x_0)x_0^j \)

Moving to higher order polynomials

- Bias is reduced.
- Variance will increase.
Outline

1 6.1 – One-dimensional kernel smoothers

2 6.2 – Selecting the width of the kernel

3 Local regression in $\mathbb{R}^p$
Width of the kernel

- For the Epanechnikov or tri-cube kernel with metric width, $\lambda$ is the radius of the support region.
- For the Gaussian kernel, $\lambda$ is the standard deviation.
- $\lambda$ is the number $k$ of nearest neighbors in $k$-nearest neighborhoods, often expressed as a fraction or span $k/N$ of the total training sample.
- If the window is narrow, $\hat{f}(x_0)$ is an average of a small number of $y_i$ close to $x_0$, and its variance will be relatively large—close to that of an individual $y_i$. The bias will tend to be small, again because each of the $E(y_i) = f(x_i)$ should be close to $f(x_0)$.
- If the window is wide, the variance of $\hat{f}(x_0)$ will be small relative to the variance of any $y_i$, because of the effects of averaging. The bias will be higher, because we are now using observations $x_i$ further from $x_0$, and there is no guarantee that $f(x_i)$ will be close to $f(x_0)$.

**Solution:** cross-validation
Outline

1. 6.1 – One-dimensional kernel smoothers

2. 6.2 – Selecting the width of the kernel

3. Local regression in $\mathbb{R}^p$
Kernel regression in $\mathbb{R}^p$

Let $b(X)$ be a vector of polynomial terms in $X$ of maximum degree $d$.

\[ d = 0 \Rightarrow b(X) = (1) \]
\[ d = 1, p = 2 \Rightarrow b(X) = (1, X_1, X_2) \]
\[ d = 2, p = 2 \Rightarrow b(X) = (1, X_1, X_2, X_1^2, X_2^2, X_1X_2) \]

At each $x_0 \in \mathbb{R}^p$, one solves

\[
\min_{\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - b(x_i)^T \beta(x_0))^2
\]

to produce the fit $\hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0)$

**Typical kernel:**

\[
K_\lambda(x_0, x) = D \left( \frac{||x - x_0||}{\lambda} \right)
\]
Example

**FIGURE 6.8.** The left panel shows three-dimensional data, where the response is the velocity measurements on a galaxy, and the two predictors record positions on the celestial sphere. The unusual “star”-shaped design indicates the way the measurements were made, and results in an extremely irregular boundary. The right panel shows the results of local linear regression smoothing in $\mathbb{R}^2$, using a nearest-neighbor window with 15% of the data.
Reference
