Regression and $k$-NN

- Regression.

\[ \mathbb{E}(Y|X) = f(X) \]

- We want to learn $f(\cdot)$ from the training set $(x_1, y_1), \ldots, (x_N, y_N)$.

- The $k$-NN estimate is a direct estimate of the conditional expectation

\[ \hat{f}(x_0) = \text{Ave}\{y_i : x_i \in \mathcal{N}_k(x_0)\}, \]

where $\mathcal{N}_k(x_0)$ is the set of $k$ training points nearest to the query point $x_0$ in squared distance.
Nadaraya-Watson Estimate

- The classical **Nadaraya-Watson Estimate** is defined as a weighted average

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

where the weights are given by the kernel function

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

- \(\lambda\) is called **metric window size, bandwidth, window width** etc. More generally, let \(h_\lambda(x_0)\) be a width function that determines the width of the neighborhood at \(x_0\), we have the **adaptive kernel-weights**

\[
K_\lambda(x_0, x_i) = D \left( \frac{|x_i - x_0|}{h_\lambda(x_0)} \right).
\]
FIGURE 6.1. In each panel, 100 pairs $(x_i, y_i)$ are generated at random from the blue curve with Gaussian errors: $Y = \sin(4X) + \varepsilon$, $X \sim \mathcal{U}[0, 1]$, $\varepsilon \sim \mathcal{N}(0, \frac{1}{3})$.

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 $\lambda = 0.2$. 

$k$-NN and Nadaraya-Watson Estimates
Examples of Kernel Functions

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

Epanechnikov kernel.

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

Tri-cube kernel.
Local Linear Regression

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

Let

\[
B = \begin{pmatrix}
1 & x_1 - x_0 \\
1 & x_2 - x_0 \\
\vdots & \vdots \\
1 & x_N - x_0
\end{pmatrix}
\]

\[
W = \begin{pmatrix}
K_\lambda(x_0, x_1) & 0 & \cdots & 0 \\
0 & K_\lambda(x_0, x_2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & K_\lambda(x_0, x_N)
\end{pmatrix}
\]

Then

\[
\hat{f}(x_0) = (1, 0)^T (B^T W B)^{-1} B^T W y = \sum_{i=1}^{N} l_i(x_0) \cdot y_i.
\]

equivalent kernel
Bias Correction via Local Linear Regression near the Boundary

\[ \hat{f}(x_0) \]

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.
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.
Local Polynomial Regression

\[
\min_{\alpha(x_0), \beta_j(x_0), j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[ y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0)(x_i - x_0)^j \right]^2.
\]

Similarly we have

\[
\hat{f}(x_0) = \hat{\alpha}(x_0) \sum_{i=1}^{N} l_i(x_0) y_i.
\]
FIGURE 6.5. Local linear fits exhibit bias in regions of curvature of the true function. Local quadratic fits tend to eliminate this bias.
Variance Increases with the Order

FIGURE 6.6. The variances functions $||l(x)||^2$ for local constant, linear and quadratic regression, for a metric bandwidth ($\lambda = 0.2$) tri-cube kernel.
Selecting the Width of the Kernel

FIGURE 6.7. Equivalent kernels for a local linear regression smoother (tri-cube kernel; orange) and a smoothing spline (blue), with matching degrees of freedom. The vertical spikes indicate the target points.
FIGURE 5.8. The smoother matrix for a smoothing spline is nearly banded, indicating an equivalent kernel with local support. The left panel represents the elements of $S$ as an image. The right panel shows the equivalent kernel or weighting function in detail for the indicated rows.
Local Regression in $\mathbb{R}^p$

If the input $X$ is $p$-dimensional, let

$$b(x_i) = (1, x_{i1}, \ldots, x_{ip})^T$$

and

$$\beta(x_0) = (\beta_0(x_0), \beta_1(x_0), \ldots, \beta_p(x_0))^T.$$

Local linear regression in $\mathbb{R}^p$ solves the minimization problem

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

where typically the kernel is a radial function

$$K_\lambda(x_i, x_0) = D \left( \frac{\|x_i - x_0\|}{\lambda} \right).$$
The concept of local regression and varying coefficient models is extremely broad: any parametric model can be made local if the fitting method accommodates observation weights.

The log likelihood \( \ell(y_i; x_i, \theta) \) can be localized at the query point \( x_0 \). The estimate of \( \theta \) is the maximizer of the kernel-weighted log likelihood

\[
\sum_{i=1}^{N} K_\lambda(x_0, x_i) \ell(y_i; x_i, \theta(x_0)).
\]

Local linear discriminant analysis.
Figure 6.13. A kernel density estimate for systolic blood pressure (for the CHD group). The density estimate at each point is the average contribution from each of the kernels at that point. We have scaled the kernels down by a factor of 10 to make the graph readable.
FIGURE 6.14. The left panel shows the two separate density estimates for systolic blood pressure in the CHD versus no-CHD groups, using a Gaussian kernel density estimate in each. The right panel shows the estimated posterior probabilities for CHD, using (6.25).