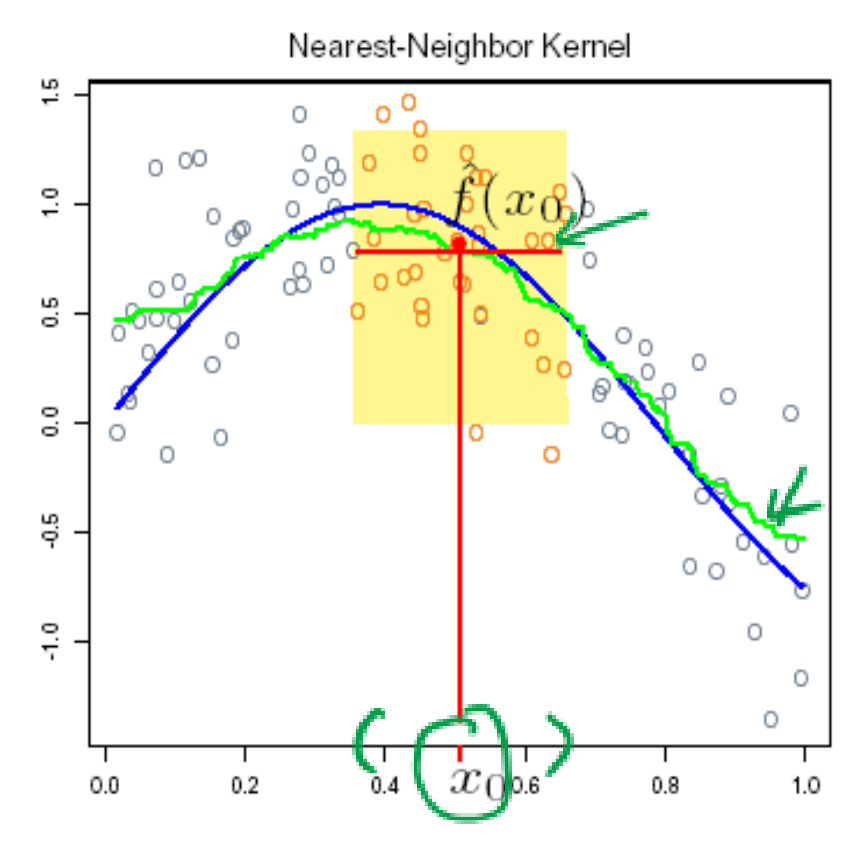
Lecture 6. Kernel Smoothing Methods

Outline

- Description of Kernel
- One-dimensional Kernel smoothing
- Selecting the Width of the Kernel
- Local Regression in R^p
- Kernel Density Estimation and Classification

Nearest Neighbor Smoothing



$$\hat{f}(x) = \frac{\sum_{x_i \in N_k(x)} f(x_i)}{k}$$

Average according to uniform weights on all k neighbors

$$Y = \sin(4X) + \varepsilon,\ X \sim U[0,1],\ \varepsilon \sim N(0, 1/3)$$

Nearest Neighbor Smoothing

- Properties:
  - Approximates E(Y | X)
  - Not continuous
- To overcome discontinuity: assign weights that die off smoothly with distance from the target point.
- Nadaraya-Watson kernel-weighted average

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

Kernels - Definition

- A *Reproducing (Mercer's) Kernel* $K(\cdot, \cdot)$, function of two variables, is an inner product of two vectors that are the image of the two variables under a feature mapping
  - Positive-definiteness: $K(x,y) = \langle \Phi(x), \Phi(y) \rangle$
  - Inner product is related to a distance metric, e.g.,
    - d(x,y) = <x-y,x-y> = $||x||^2 + ||y||^2 - 2\langle x,y \rangle$
    - $$d(x,y) = \sqrt{\sum_{j=1}^{p} (\phi_j(x) - \phi_j(y))^2}$$
- A *(Nadaraya-Watson) kernel* can be represented as a decreasing function of a distance between the two objects
  - a measure of similarity between two objects

Kernels with One-dimensional Features

- D: a decreasing (often with a compact support) function on $\boldsymbol{R}^+$
- $h_\lambda(.)$:
  - a window with some specified width, e.g. $h_\lambda(x_0)=\lambda$
  - often adaptive to data points
  - a scaling function on $\boldsymbol{R}$

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

Kernel

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

- Similarity to the target ``x0''
  - Larger λ implies lower variance but high bias

K-Nearest Neighbor Kernel

- In conjunction with a D(.) that has compact support, and use,

$$\lambda = k$$

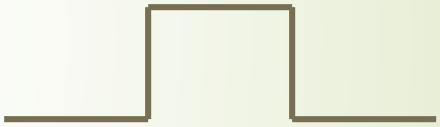
$$h_k(x_0) = | x_0 - x_{[k]} |$$

- $X_{[k]}$ denotes the k^{th} nearest neighbor to x_0

不光滑

Uniform kernel

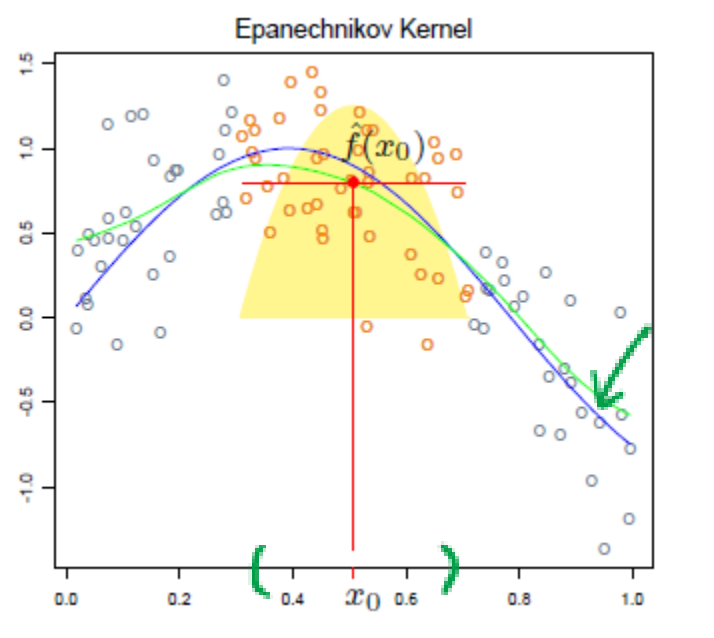
$$D(t) = I(|t| \leq 1)$$

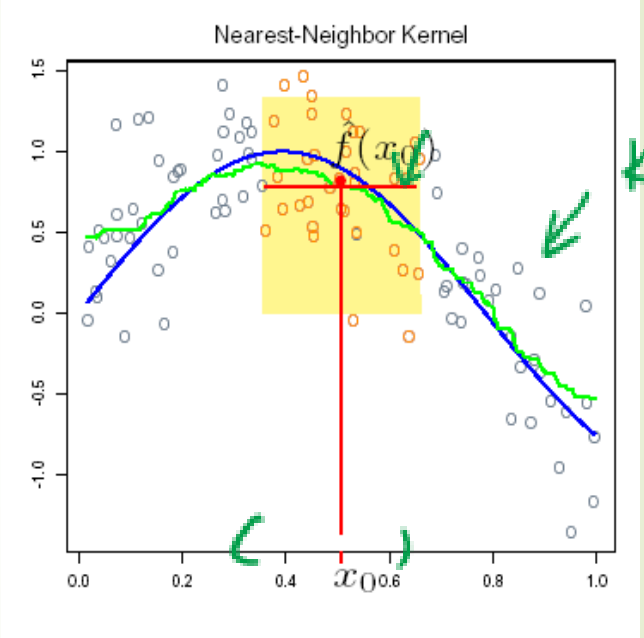


$$h_\lambda = \lambda$$

- Local smoothing with this kernel yields moving averages
- It is a compact support kernel.

Epanechnikov Quadratic Kernel





KNN

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

with

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

Compact support, non-continuous derivative

Tri-Cube Kernel

$$D(t) = (1 - |t|^3)^3 I(|t| \leq 1)$$

$$h_\lambda = \lambda$$

Compact support kernel, two continuous derivatives

Gaussian Kernel

$$D(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

$$h_\lambda = \lambda$$

Noncompact support Kernel, infinitely differentiable

Shape of Kernels: Examples

- Example: Epanechnikov Kernel – $h_\lambda(x_0) = \lambda$ is constant
- K-NN: neighborhood size k replaces λ $h_k(x_0) = |x_0 - x_{[k]}|$
- Tri-cube:

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

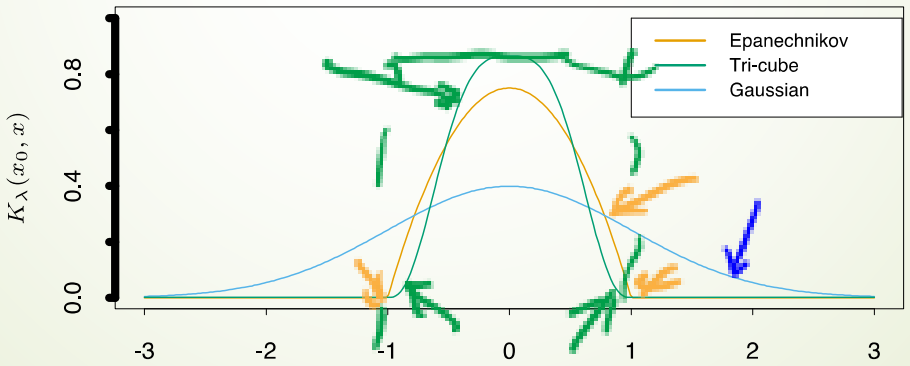


FIGURE 6.2. *A comparison of three popular kernels for local smoothing. Each has been calibrated to integrate to* 1. *The tri-cube kernel is compact and has two continuous derivatives at the boundary of its support, while the Epanechnikov kernel has none. The Gaussian kernel is continuously differentiable, but has infinite support.*

Local Linear Regression

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

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0.$$

Define the vector-valued function $b(x)^T = (1, x)$. Let $\mathbf{B}$ be the $N \times 2$ regression matrix with ith row $b(x_i)^T$, and $\mathbf{W}(x_0)$ the $N \times N$ diagonal matrix with ith diagonal element $K_\lambda(x_0, x_i)$. Then

$$\begin{aligned} \hat{f}(x_0) &= b(x_0)^T(\mathbf{B}^T\mathbf{W}(x_0)\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}(x_0)\mathbf{y} \\ &= \sum_{i=1}^{N} l_i(x_0)y_i. \end{aligned}$$

equivalent kernel.

Local linear regression vs. N-W average at Boundary

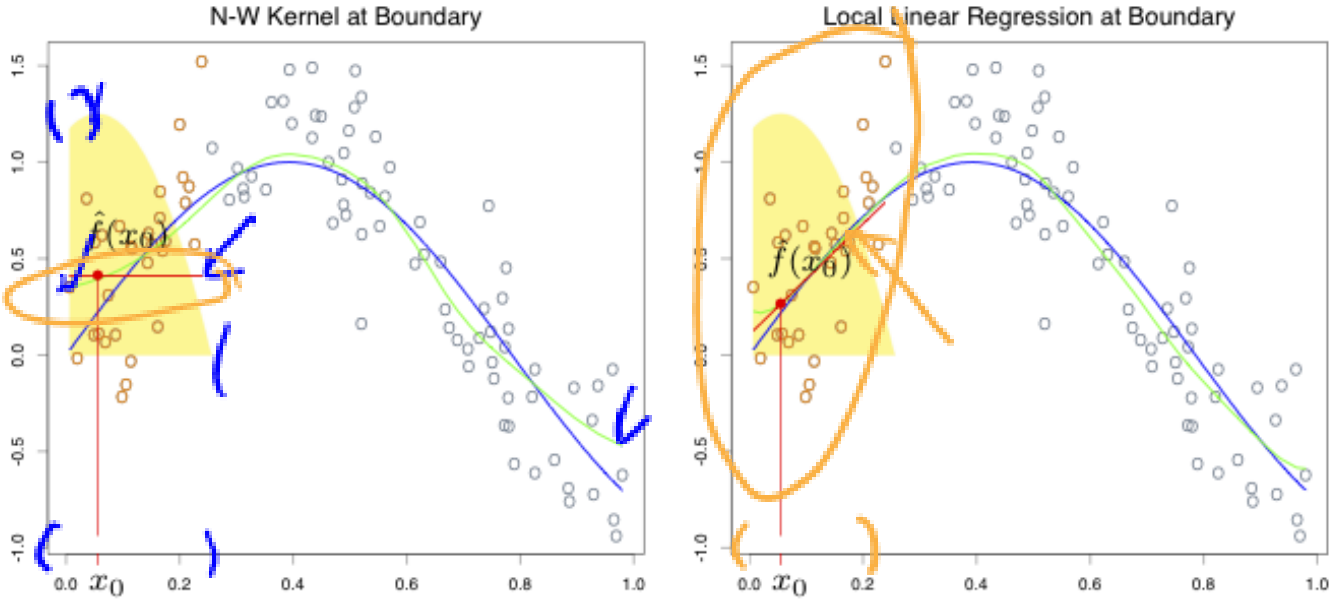


FIGURE 6.3. *The locally weighted average has bias problems at or near the boundaries of the domain. The true function is approximately linear here, but most of the observations in the neighborhood have a higher mean than the target point, so despite weighting, their mean will be biased upwards. By fitting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

Asymmetry of kernel may lead to bias near boundary

Local linear regression

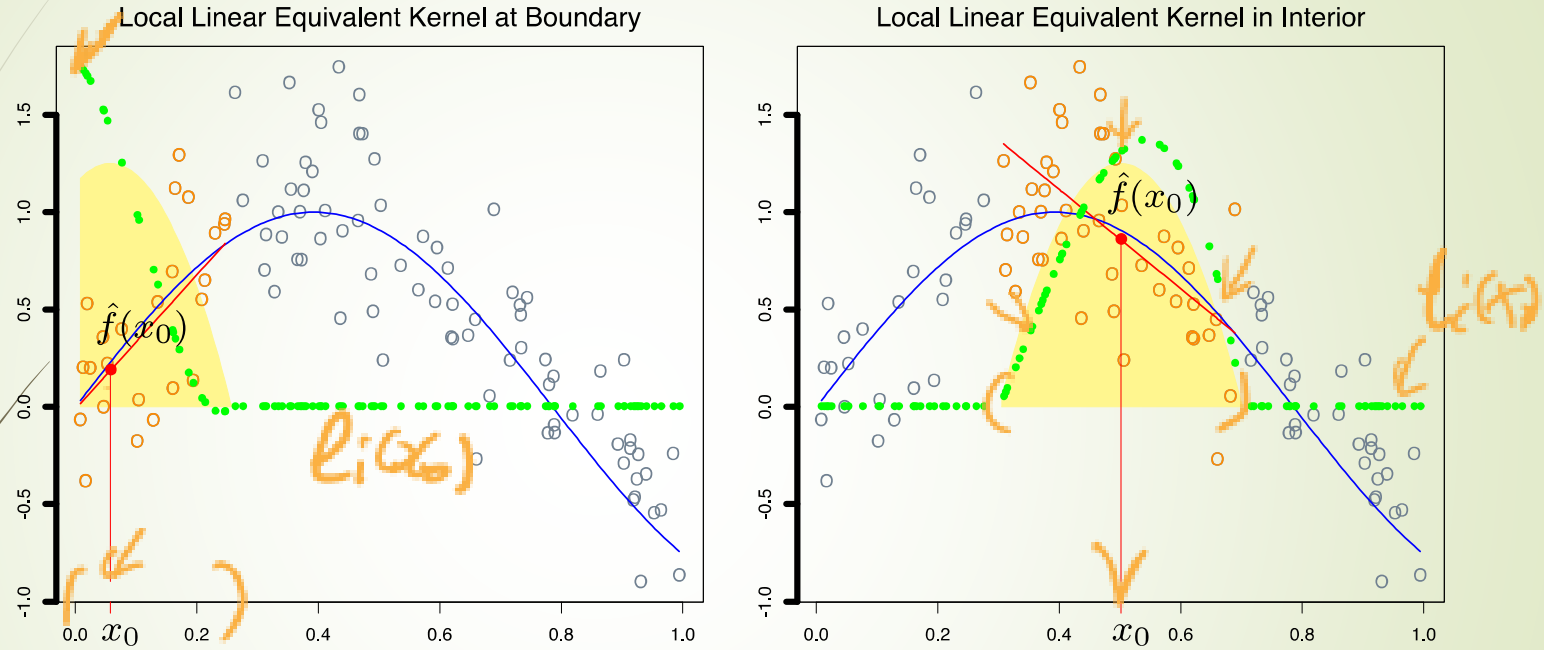


FIGURE 6.4. *The green points show the equivalent kernel $l_i(x_0)$ for local regression. These are the weights in $\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) y_i$, plotted against their corresponding x_i. For display purposes, these have been rescaled, since in fact they sum to 1. Since the yellow shaded 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 linear regression: Asymptotic Order of Bias

$$\begin{aligned}
\mathrm{E}\hat{f}(x_0) &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\
&= f(x_0) \sum_{i=1}^{N} l_i(x_0) + f'(x_0) \sum_{i=1}^{N} (x_i - x_0) l_i(x_0) \\
&\quad + \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + R,
\end{aligned}$$

$\sum_{i=1}^{N} l_i(x_0) = 1$ *

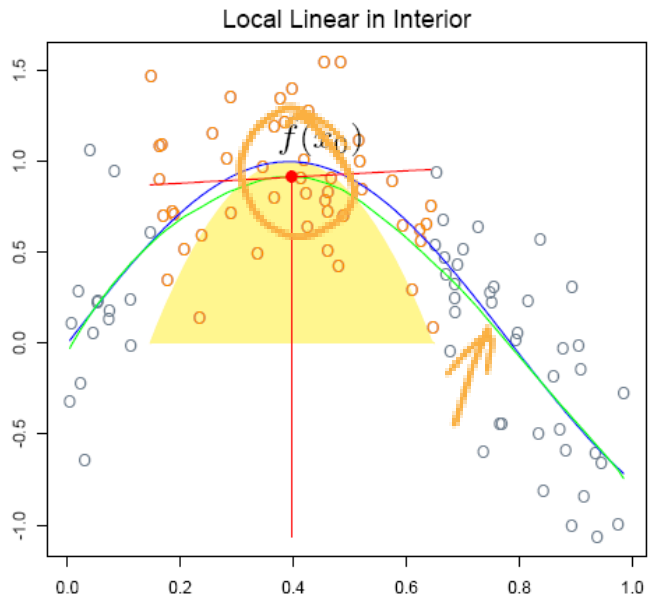
$\sum_{i=1}^{N} (x_i - x_0) l_i(x_0) = 0.$ *

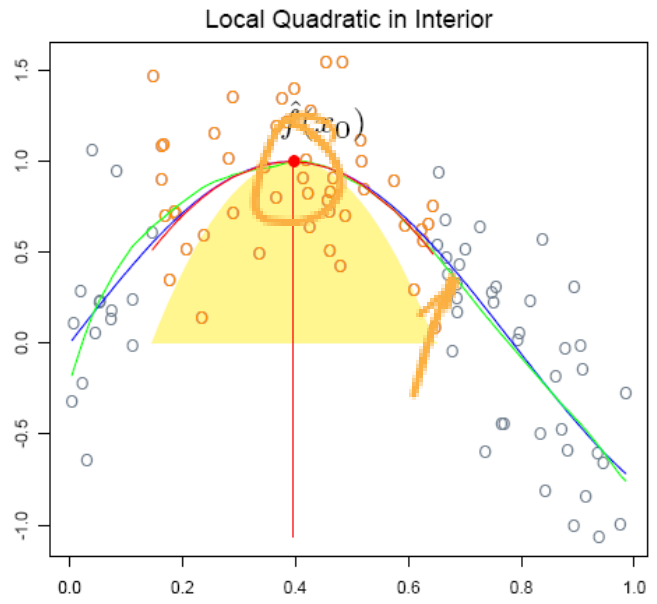
Homework?

the bias is $\mathrm{E}\hat{f}(x_0) - f(x_0)$, we see that it depends only on quadratic and higher–order terms in the expansion of f.

Local Polynomial Regression

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

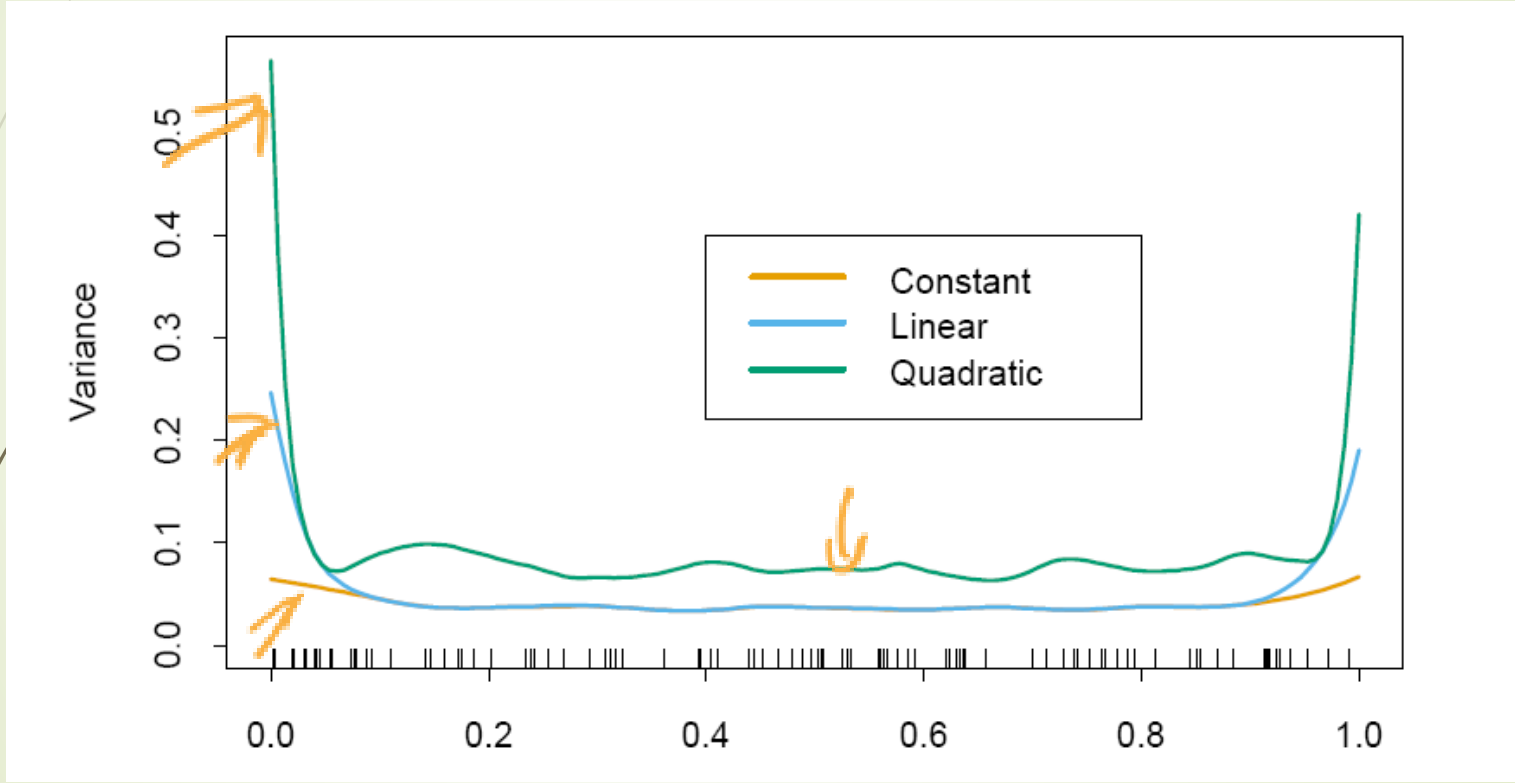




Reduce the bias to (d+1)-order asymptotically

Local Polynomial Regression

- The price of bias reduction? Variance!



interpolation
extrapolation

$$y_i = f(x_i) + \varepsilon_i,$$

$$\varepsilon_i \sim N(0, \sigma^2)$$

$$\mathrm{Var}(\hat{f}(x_0)) = \sigma^2 ||l(x_0)||^2$$

Selecting the Width of the Kernel

- λ controls the width of the local region

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

with

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

- Epanechnikov or tri-cube: λ is the radius of the support region
- KNN: λ is the number of neighbors
- Gaussian: λ is the standard deviation

Smoother Matrix (Sensitivity) S_λ

$$\hat{f}(x_0) = \sum_{i=1}^{n} l_i(x_0) y_i$$

$$\{S_\lambda\}_{ij} = l_i(x_j)$$

Degrees of freedom = $\mathrm{tr}(S_\lambda)$

Selecting the Width of the Kernel

- A natural bias-variance tradeoff as we change the width of the averaging window (take local average as an example)
  - If the window is narrow, variance will be bigger and bias is smaller
  - If the window is wide, variance will be smaller and bias is big, because some x will be far away from x0
- So CV, AIC, BIC can all be used to select a good λ

$$\hat{f} = S_\lambda y, \qquad df = trace(S_\lambda)$$

Local Regression in R^p

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

$$\hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0)$$

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

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

For example, with $d = 1$ and $p = 2$ we get $b(X) = (1, X_1, X_2)$

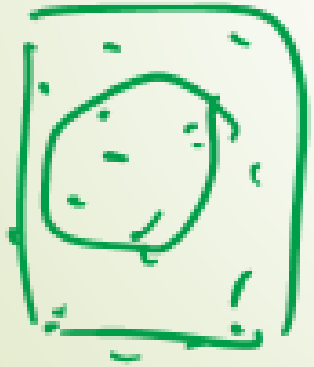
$d = 2$

$(1, X_1, X_2, X_1^2, X_2^2, X_1X_2)$

Local Regression in R^p

- Boundary effects: boundary points increase with dimension
- Less useful in high dimensions: localness (low bias) vs. insufficient samples (high variance)
- Difficult for visualization

Manifolds



p=10

$\frac{1}{2}^2$

Structured Regression

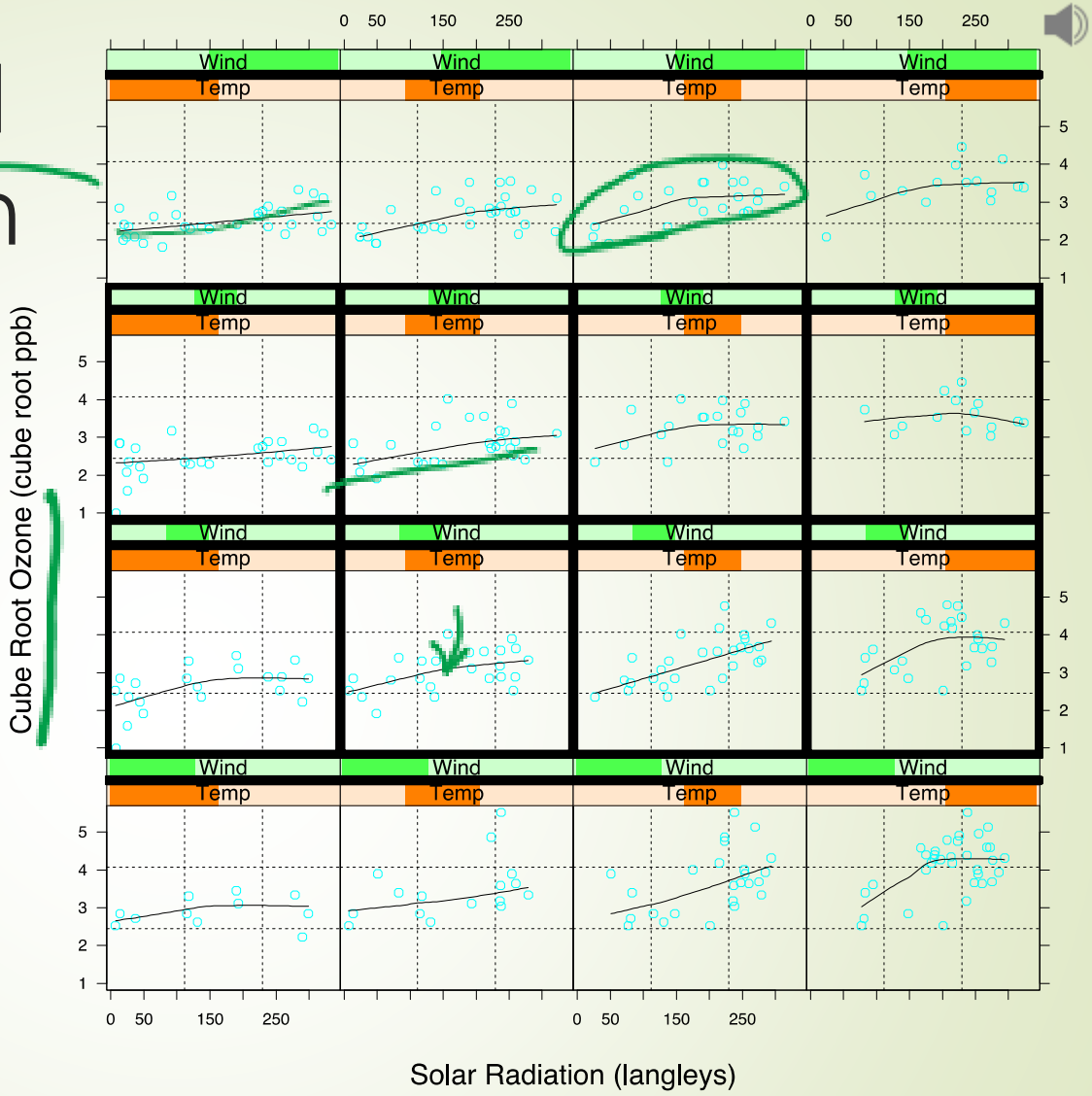


FIGURE 6.9. *Three-dimensional smoothing example. The response is (cube-root of) ozone concentration, and the three predictors are temperature, wind speed and radiation. The* trellis *display shows ozone as a function of radiation, conditioned on intervals of temperature and wind speed (indicated by darker green or orange shaded bars). Each panel contains about 40% of the range of each of the conditioned variables. The curve in each panel is a univariate local linear regression, fit to the data in the panel.*

Structured Local Regression in $\mathbb{R}^p$: Mahalanobis-distance

- When p/n is big, local regression is not helpful, unless we have some structural information.
- Mahalanobis weighted distance

$$K_{\lambda,A}(x_0, x) = D\left(\frac{(x - x_0)^T \mathbf{A}(x - x_0)}{\lambda}\right).$$

A : sparse variable sel.

low rank dim red.

Structured Local Regression in $\mathbb{R}^p$: ANOVA

- Structured regression functions: high order interactions
- E.g. Additive models only assume first order interactions

$$f(X_1, X_2, \ldots, X_p) = \alpha + \sum_j g_j(X_j) + \sum_{k<\ell} g_{k\ell}(X_k, X_\ell) + \cdots$$

- One-dimensional local regression for each stage

GAM

Varying Coefficient Models

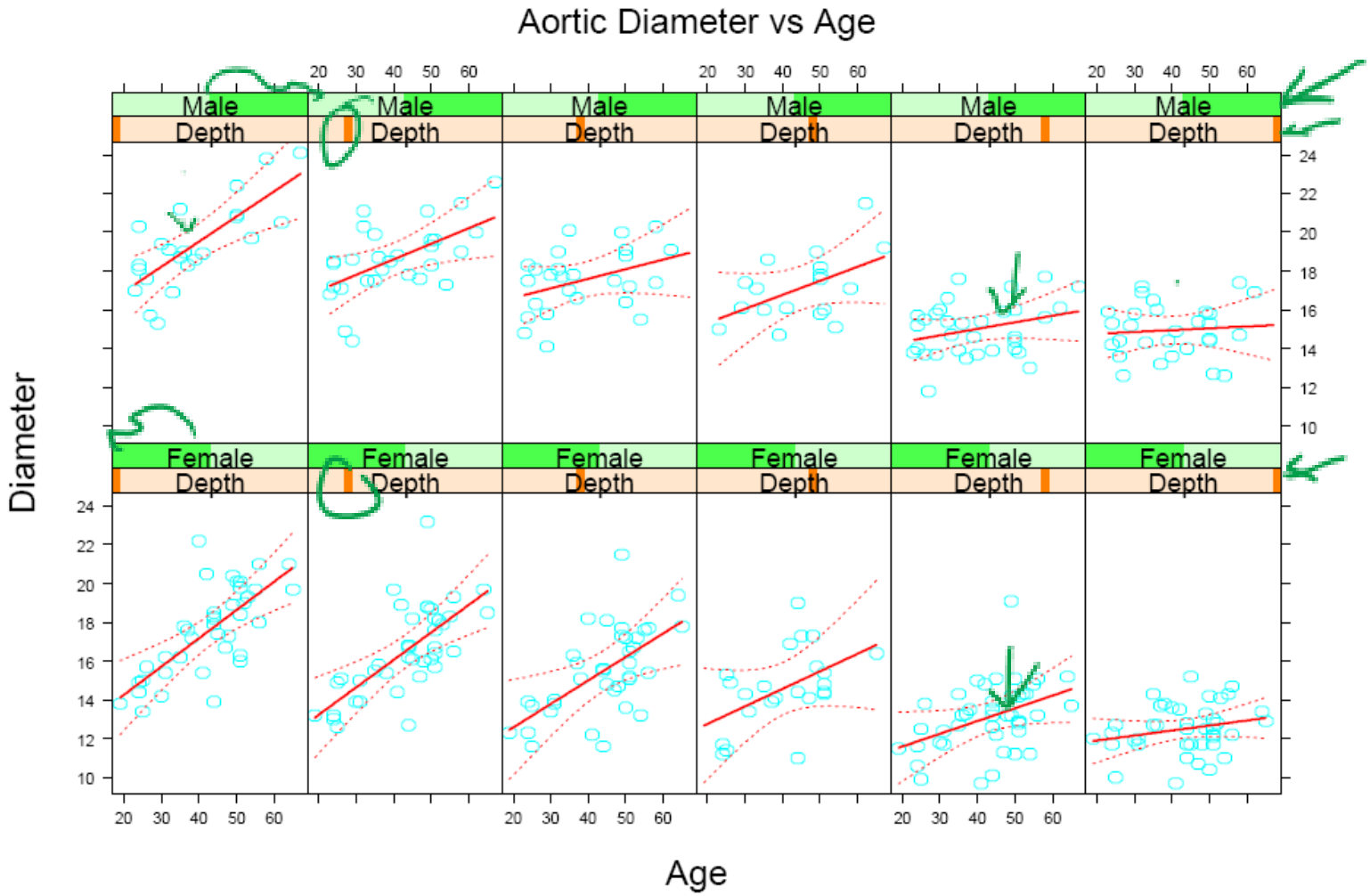
- $X \in R^p$
- Divide the p predictors into two sets:
  - $X_1, X_2, \ldots, X_q$ and Z
- We assume the conditional linear model:

$$f(X) = \alpha(Z) + \beta_1(Z)X_1 + \cdots + \beta_q(Z)X_q.$$

$$\min_{\alpha(z_0),\beta(z_0)} \sum_{i=1}^{N} K_\lambda(z_0, z_i) \left(y_i - \alpha(z_0) - x_{1i}\beta_1(z_0) - \cdots - x_{qi}\beta_q(z_0)\right)^2 .$$

Extended reading: Gelman and Hill, Multilevel Models, 2007

Varying Coefficient Models: an example



Local Likelihood and Other Models

- From global to local:

$$l(\beta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) l(y_i, x_i^T \beta(x_0)).$$

- Example: local logistic regression model.

Locally logistic regression

$$\Pr(G = j|X = x) = \frac{e^{\beta_{j0}+\beta_j^T x}}{1 + \sum_{k=1}^{J-1} e^{\beta_{k0}+\beta_k^T x}}. \tag{6.18}$$

The local log-likelihood for this J class model can be written

$$\max \sum_{i=1}^{N} K_\lambda(x_0, x_i) \Big\{ \beta_{g_i 0}(x_0) + \beta_{g_i}(x_0)^T (x_i - x_0) - \log \Big[1 + \sum_{k=1}^{J-1} \exp\left(\beta_{k0}(x_0) + \beta_k(x_0)^T (x_i - x_0)\right) \Big] \Big\}. \tag{6.19}$$

$$\log \frac{\Pr(G = j|X = x)}{\Pr(G = j|X = x_0)}$$

Example: Heart Disease

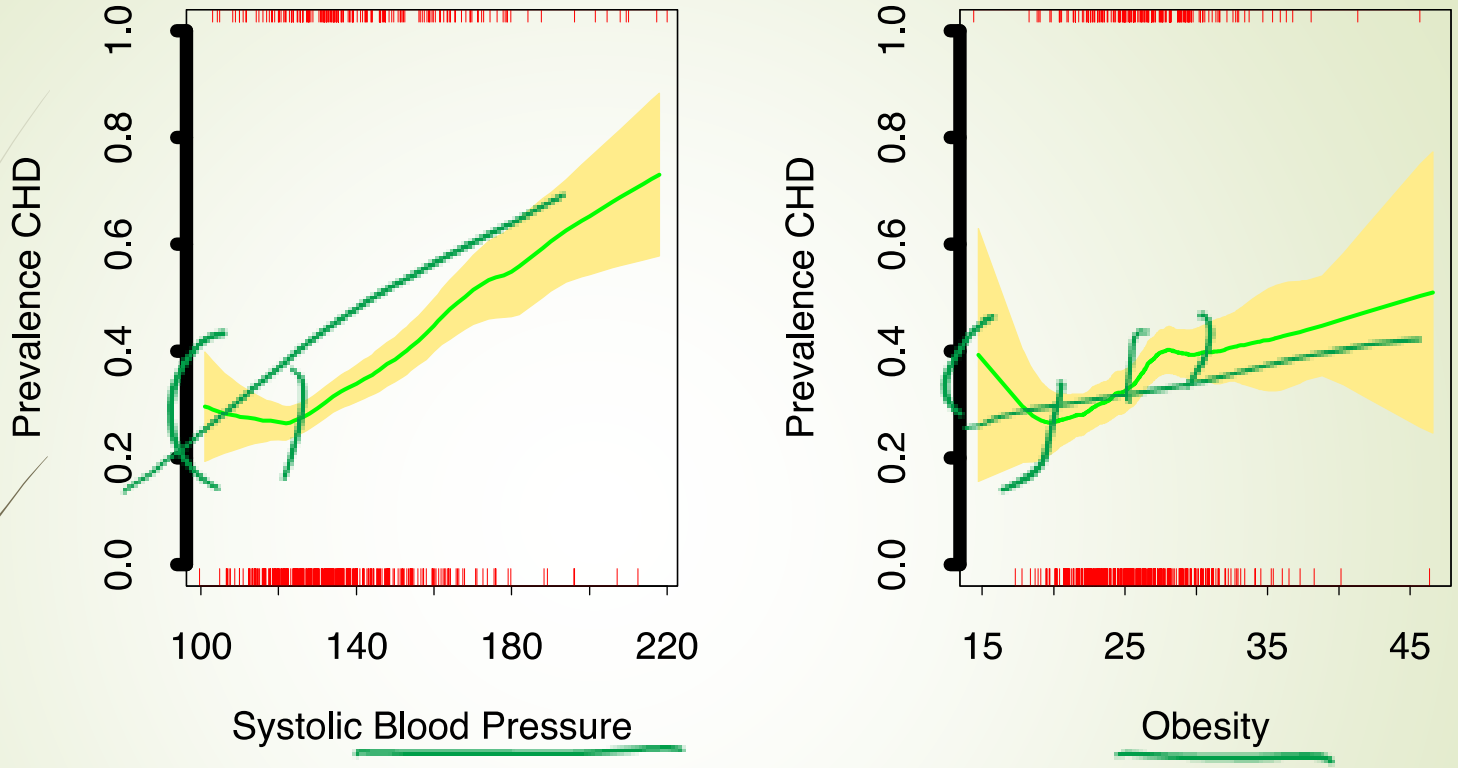
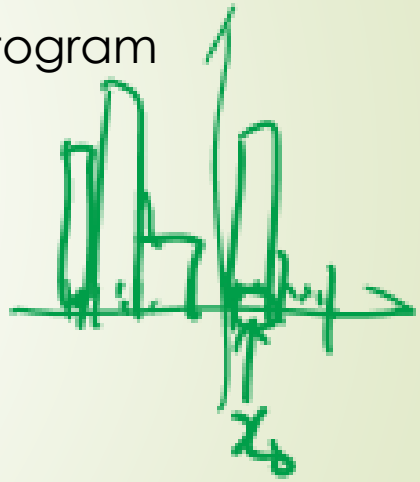


FIGURE 6.12. *Each plot shows the binary response CHD (coronary heart disease) as a function of a risk factor for the South African heart disease data. For each plot we have computed the fitted prevalence of CHD using a local linear logistic regression model. The unexpected increase in the prevalence of CHD at the lower ends of the ranges is because these are retrospective data, and some of the subjects had already undergone treatment to reduce their blood pressure and weight. The shaded region in the plot indicates an estimated pointwise standard error band.*

Kernel Density Estimation and Classification

- Kernel Density Estimation: histogram

$$\hat{f}_X(x_0) = \frac{\#x_i \in \mathcal{N}(x_0)}{N\lambda},$$



- Parzen's Smooth version:

$$\hat{f}_X(x_0) = \frac{1}{N\lambda}\sum_{i=1}^{N} K_\lambda(x_0, x_i),$$

- One choice of the kernel:

$$K_\lambda(x_0, x) = \phi(|x - x_0|/\lambda)$$

Kernel Density Classification

One can use nonparametric density estimates for classification in a straightforward fashion using Bayes' theorem. Suppose for a J class problem we fit nonparametric density estimates $\hat{f}_j(X)$, $j = 1, \ldots, J$ separately in each of the classes, and we also have estimates of the class priors $\hat{\pi}_j$ (usually the sample proportions). Then

$$\hat{\Pr}(G = j | X = x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \hat{\pi}_k \hat{f}_k(x_0)}.$$

Naïve Bayes Classifier

- Assumption: given class G=j, features are independent

$$f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k).$$

- Generalized Additive Model vs. Naïve Bayes

$$\begin{aligned}
\log \frac{\Pr(G=\ell|X)}{\Pr(G=J|X)} &= \log \frac{\pi_\ell f_\ell(X)}{\pi_J f_J(X)} \\
&= \log \frac{\pi_\ell \prod_{k=1}^{p} f_{\ell k}(X_k)}{\pi_J \prod_{k=1}^{p} f_{Jk}(X_k)} \\
&= \log \frac{\pi_\ell}{\pi_J} + \sum_{k=1}^{p} \log \frac{f_{\ell k}(X_k)}{f_{Jk}(X_k)} \\
&= \alpha_\ell + \sum_{k=1}^{p} g_{\ell k}(X_k).
\end{aligned} \tag{6.27}$$

GAM

Mixture Models for Density Estimation and Classification

- Gaussian mixture model: Any smooth function can be approximated well by mixture Gaussian models.

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \Sigma_m)$$

Hidden param

- Maximum likelihood: EM algorithm is used for the parameter estimation

Computation Consideration

- Kernel and local regression (density estimation) are memory-based methods.
- The method is based on the entire training data set.
- For real-time applications, those methods can be infeasible.
  - Weibiao Wu et al. @Chicago online method

Homework

- Due Nov 9.
- ESLII_print 10, Exercise: 6.2, 6.4, 6.9, 6.10, 6.12