Statistical Machine Learning

Lecture 5: Kernel Smoothing Methods

W.Q.Cui Research Group

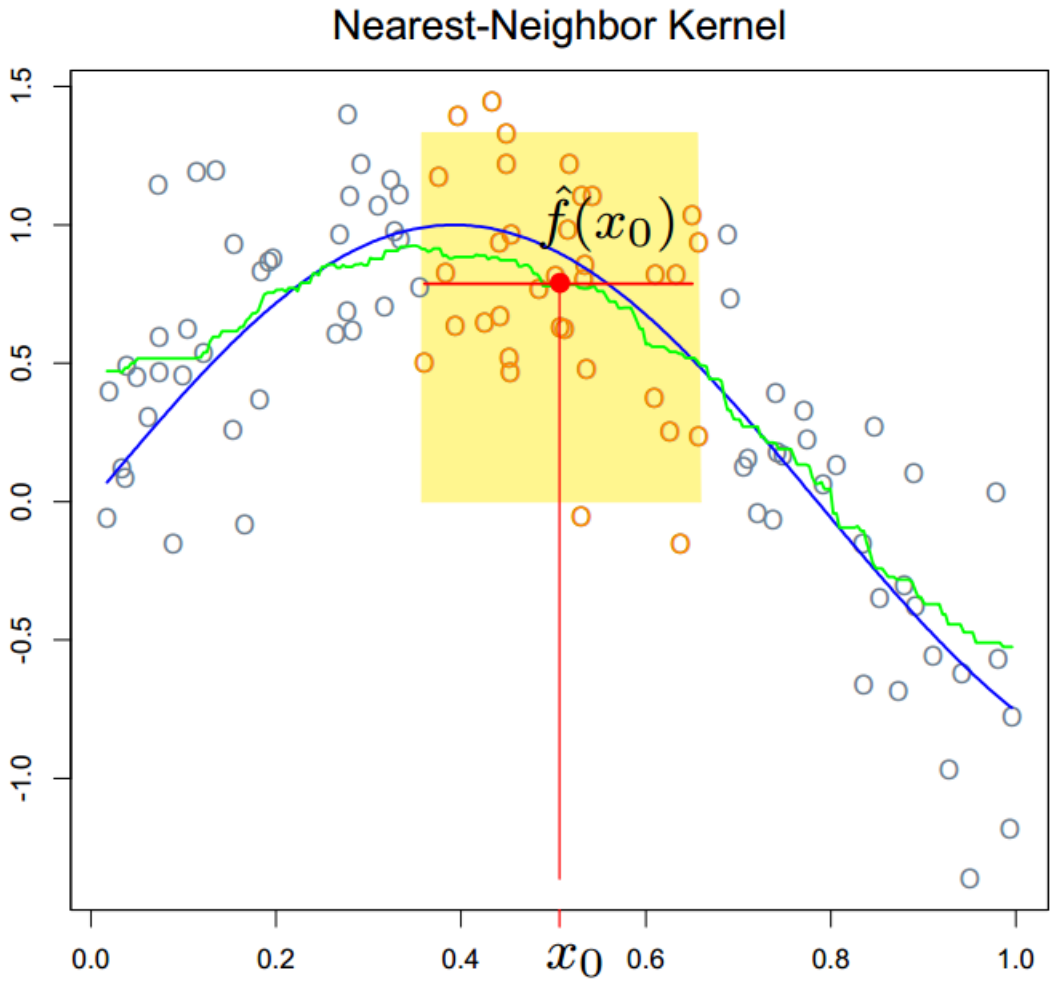
Department of Statistics and Finance
University of Science and Technology of China

2018 Autumn

Contents

One-Dimensional Kernel Smoothers

$\hat{f}(x) = Ave(y_i|x_i \in N_k(x))$ use the 30-nearest neighborhood



An example:Nadaraya-Watson kernel-weighted average

Nadaraya-Watson kernel-weighted average:

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

with the Epanechnikov quadratic kernel

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

with

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

An example:Nadaraya-Watson kernel-weighted average

Continuous and quite smooth.

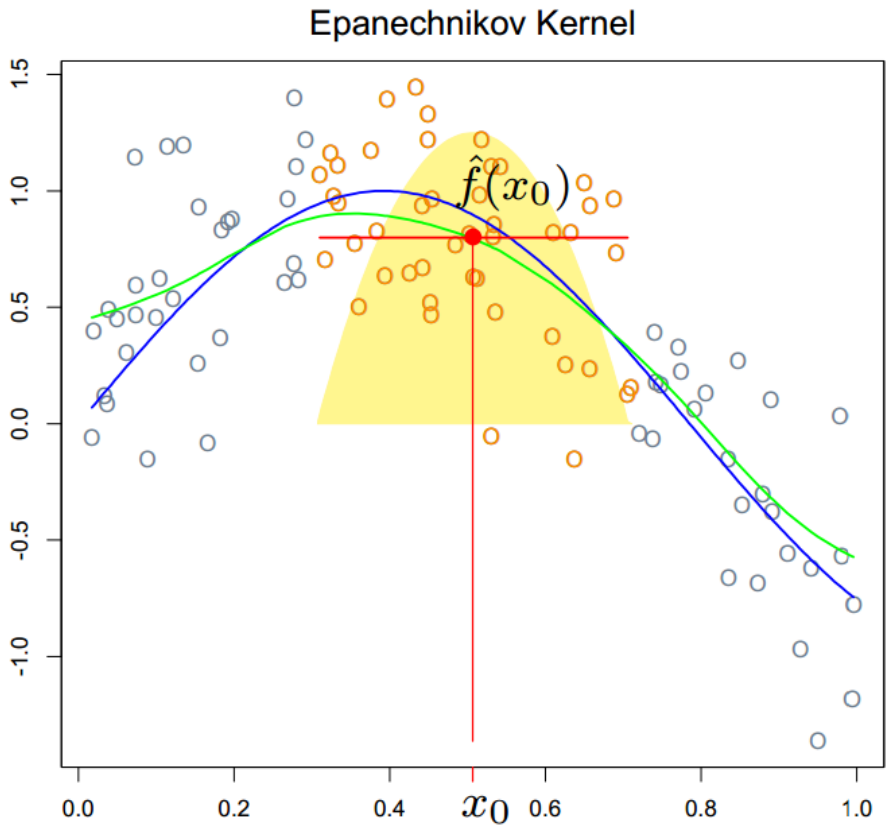


Figure: An Epanechnikov kernel with (half) window width $\lambda = 0.2$

Kernel - Definition

- A 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
  —Inner product is related to a norm (metric)

- A 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

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

- D:a decreasing function on R^+
- $h_\lambda(\cdot)$:
  - a window with some specified width
  - a scaling function on R

Some kinds of kernel

Name	D(t)	hλ	Compact or not
Uniform kernel	$D(t) = I(\|t\| \leq 1)$	$h_\lambda = \lambda$	Yes
Epanecnikov Quadratic Kernel	$D(t) = \frac{3}{4}(1-t^2)I(\|t\| \leq 1)$	$h_\lambda = \lambda$	Yes
Tri-Cube Kernel	$D(t) = (1-\|t\|^3)^3 I(\|t\| \leq 1)$	$h_\lambda = \lambda$	Yes
Gaussian Kernel	$D(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$	$h_\lambda = \lambda$	Not

Details

There are a number of details that one has to attend to in practice:

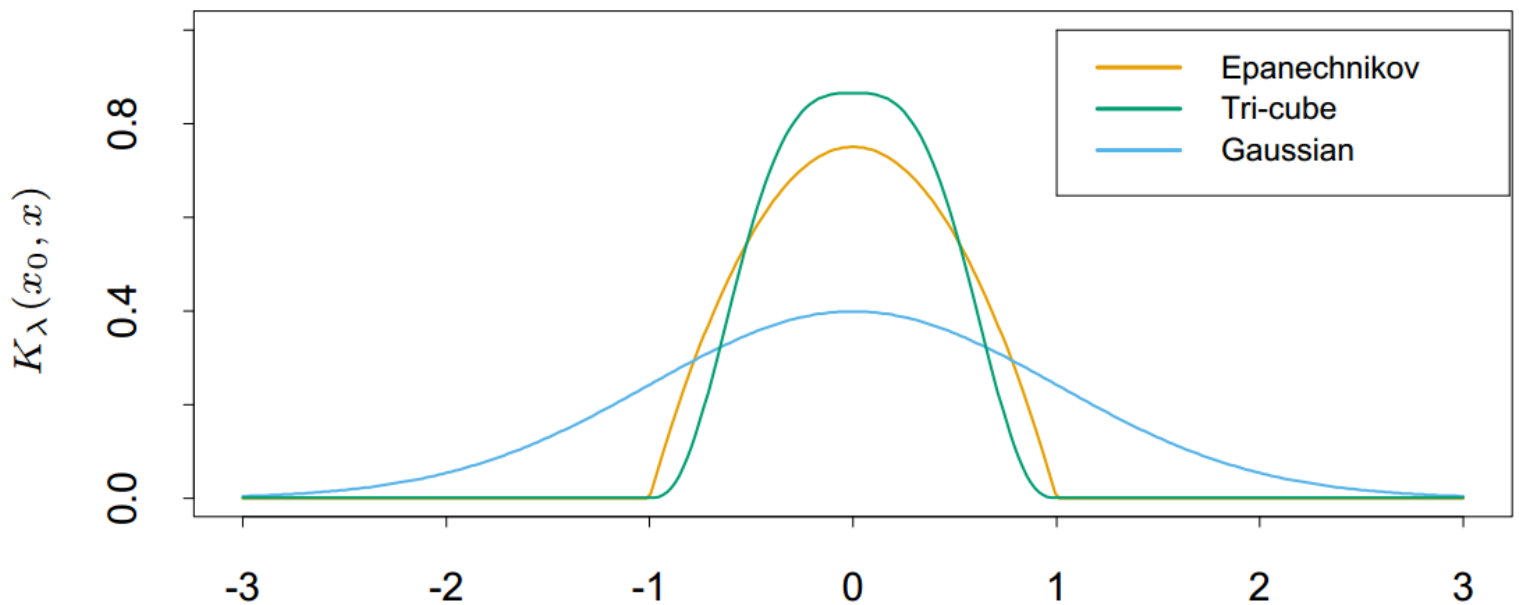
- Large λ implies lower variance but higher bias.
- Metric window widths(constant $h_\lambda(x)$)
  keep the bias of the estimate constant but the variance is inversely proportional to the local density.
- Nearest-neighbor window
  the variance stays constant and the absolute bias varies inversely with local density.

Details

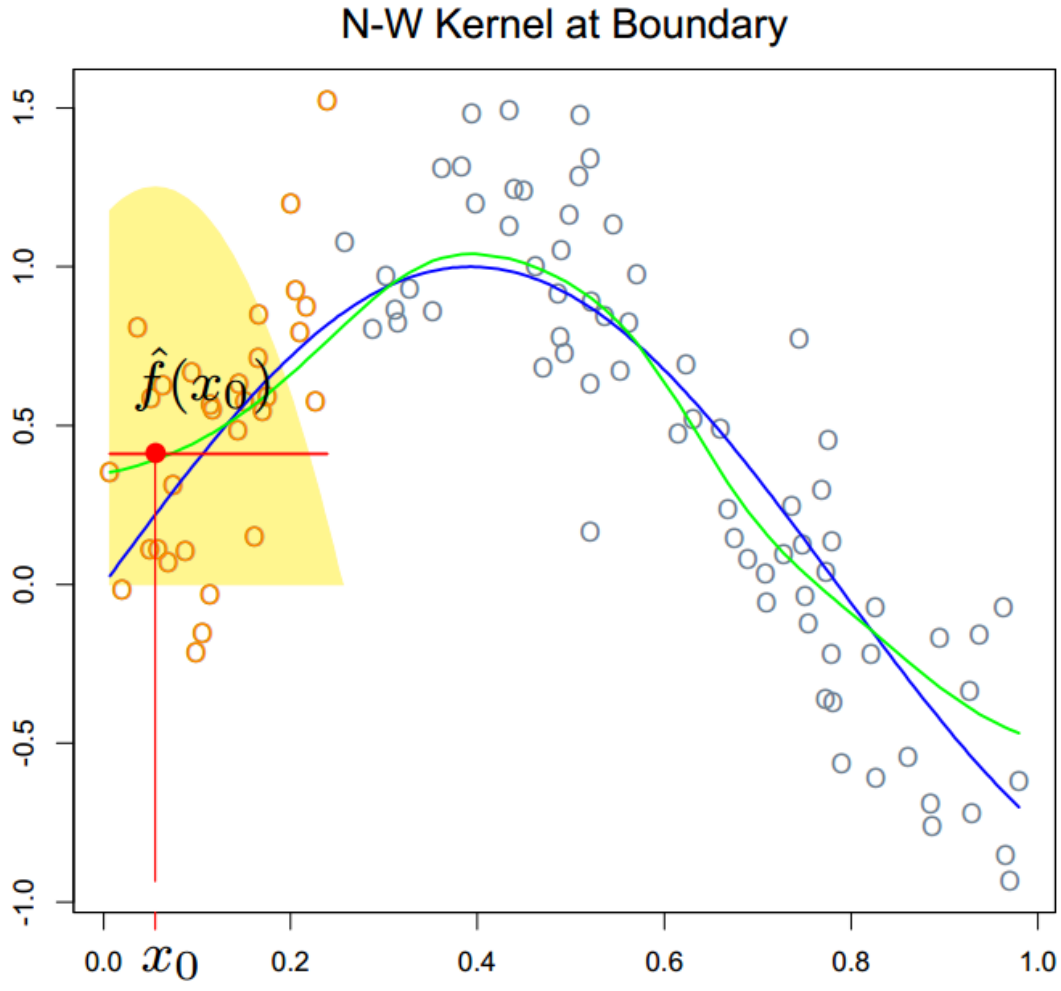
- When there are ties in the x_i.
- Observation weights w_i. Operationally we simply multiply them by the kernel weights before computing the weighted average.
- Boundary issues arise. The metric neighborhoods tend to contain less points on the boundaries, while the nearest-neighborhoods get wider.
- The Epanechnikov kernel has compact support (needed when used with nearest-neighbor window size). Another popular compact kernel is based on the tri-cube function

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

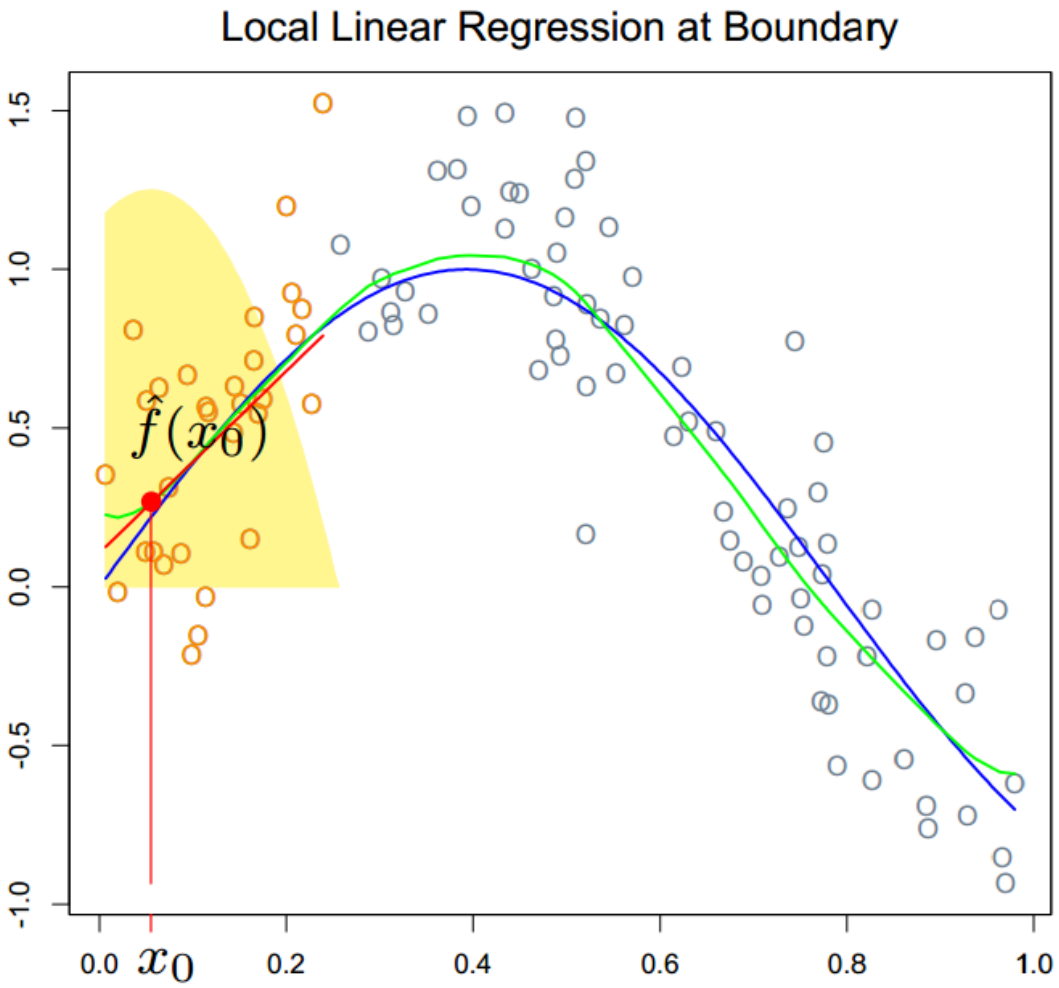
This is flatter on the top (like the nearest-neighbor box) and is differentiable at the boundary of its support. The Gaussian density function $D(t) = \Phi(t)$ is a popular noncompact kernel, with the standard deviation playing the role of the window size.



The smooth kernel fit still has problems: Locally-weighted averages can be badly biased on the boundaries of the domain, because of the asymmetry of the kernel in that region.



By fitting straight lines rather than constants locally, we can remove this bias exactly to first order



Locally weighted regression

Locally weighted regression solves a separate weighted least squares problem at each target point x_0:

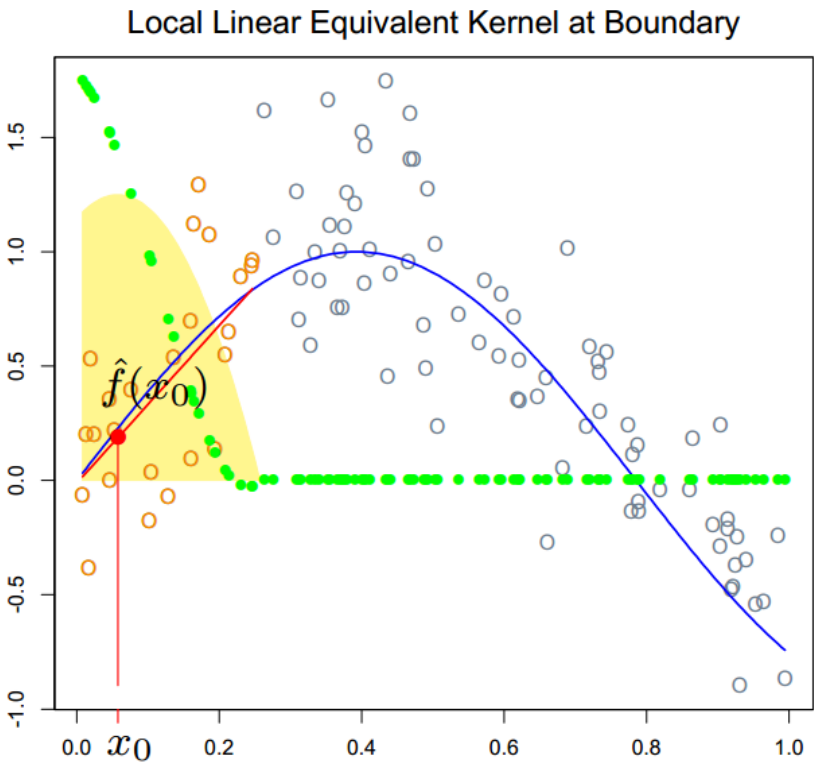
$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)\Big[y_i - \alpha(x_0) - \beta(x_0)x_i\Big]^2.$$

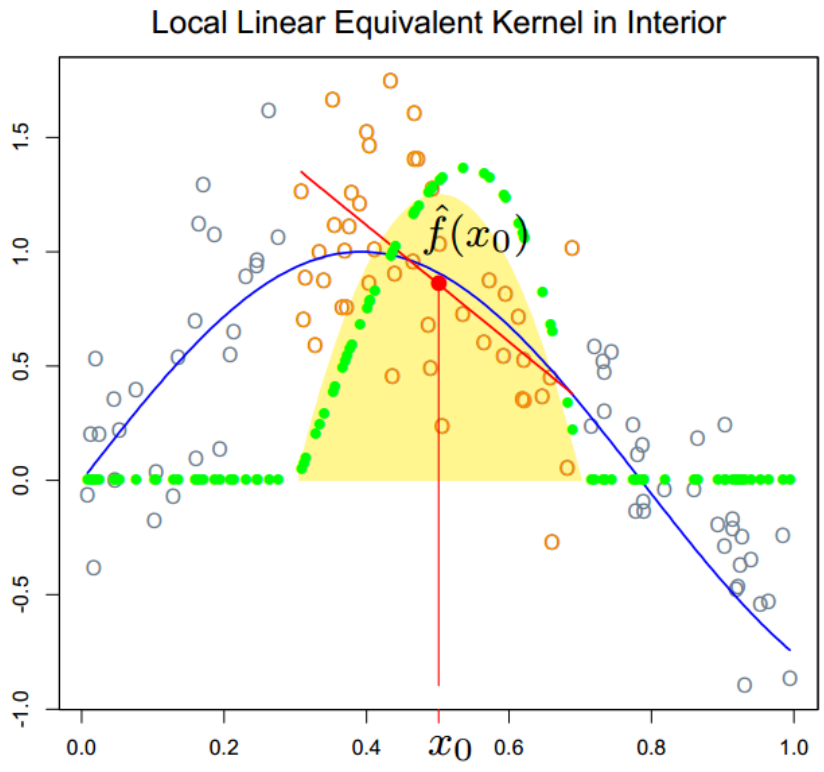
The estimate is then $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$.

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

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

The equivalent kernel $l_i(x_0)$ for local regression





Consider the following expansion

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

for local linear regression

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

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

Local Polynomial Regression

We can fit local polynomial fits of any degree d

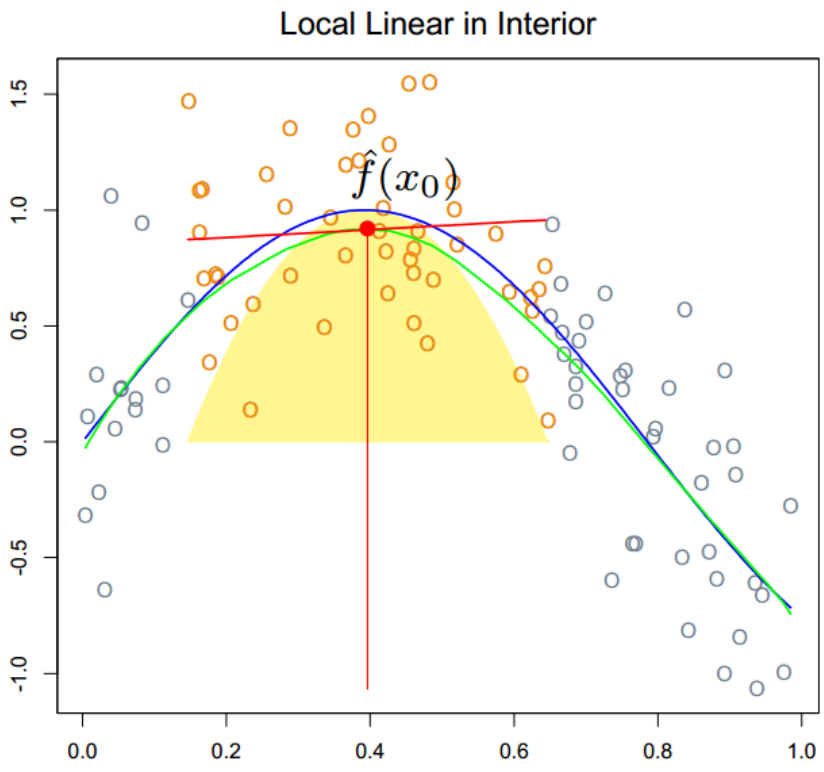
$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\cdots,d} \sum_{i=1}^{N} K_\lambda(x_0,x_i)\left[y_i - \alpha(x_0) - \sum_{j=1}^{d}\beta_j(x_0)x_i^j\right]^2$$

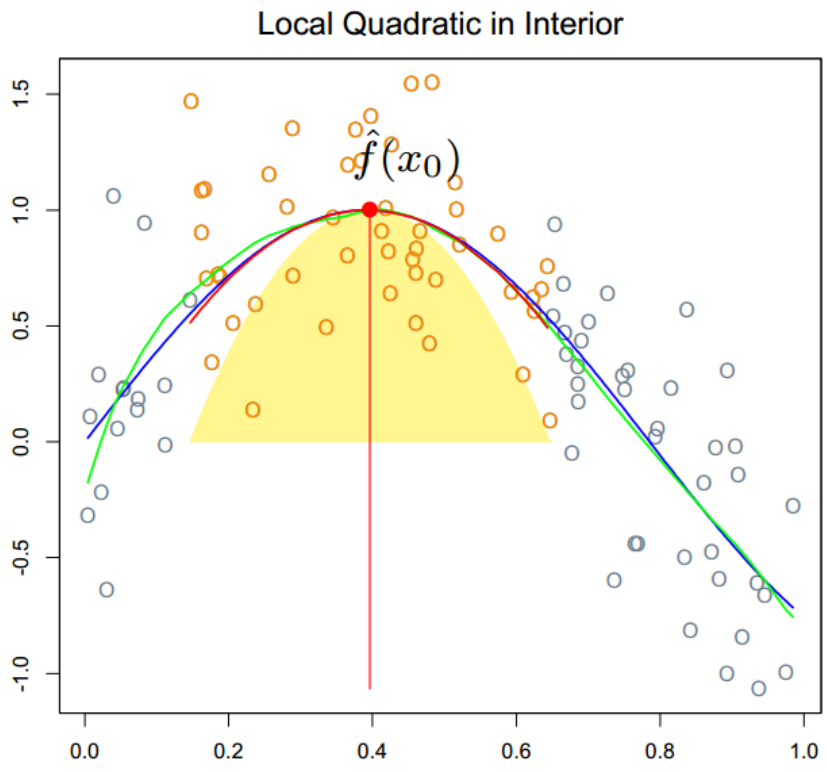
with solution $\hat{f}(x_0) = \hat{\alpha}(x_0) + \sum_{j=1}^{d}\hat{\beta}_j(x_0)x_0^j$
the bias will only have components of degree $d+1$ and higher

- **increased variance is a price to be paid for this bias reduction**

  Assuming the model $y_i = f(x_i) + \varepsilon_i$, with ε_i independent and identically distributed with mean zero and variance σ^2, $Var(\hat{f}(x_0)) = \sigma^2\|l(x_0)\|^2$, where $l(x_0)$ is the vector of equivalent kernel weights at x_0. $\|l(x_0)\|$ increases with d.

An example


Local Linear in Interior
$\hat{f}(x_0)$
1.5
1.0
0.5
0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0


Local Quadratic in Interior
$\hat{f}(x_0)$
1.5
1.0
0.5
0.0
-0.5
-1.0
0.0
0.2
0.4
0.6
0.8
1.0

- Local linear fits can help bias dramatically at the boundaries at a modest cost in variance. Local quadratic fits do little at the boundaries for bias, but increase the variance a lot.

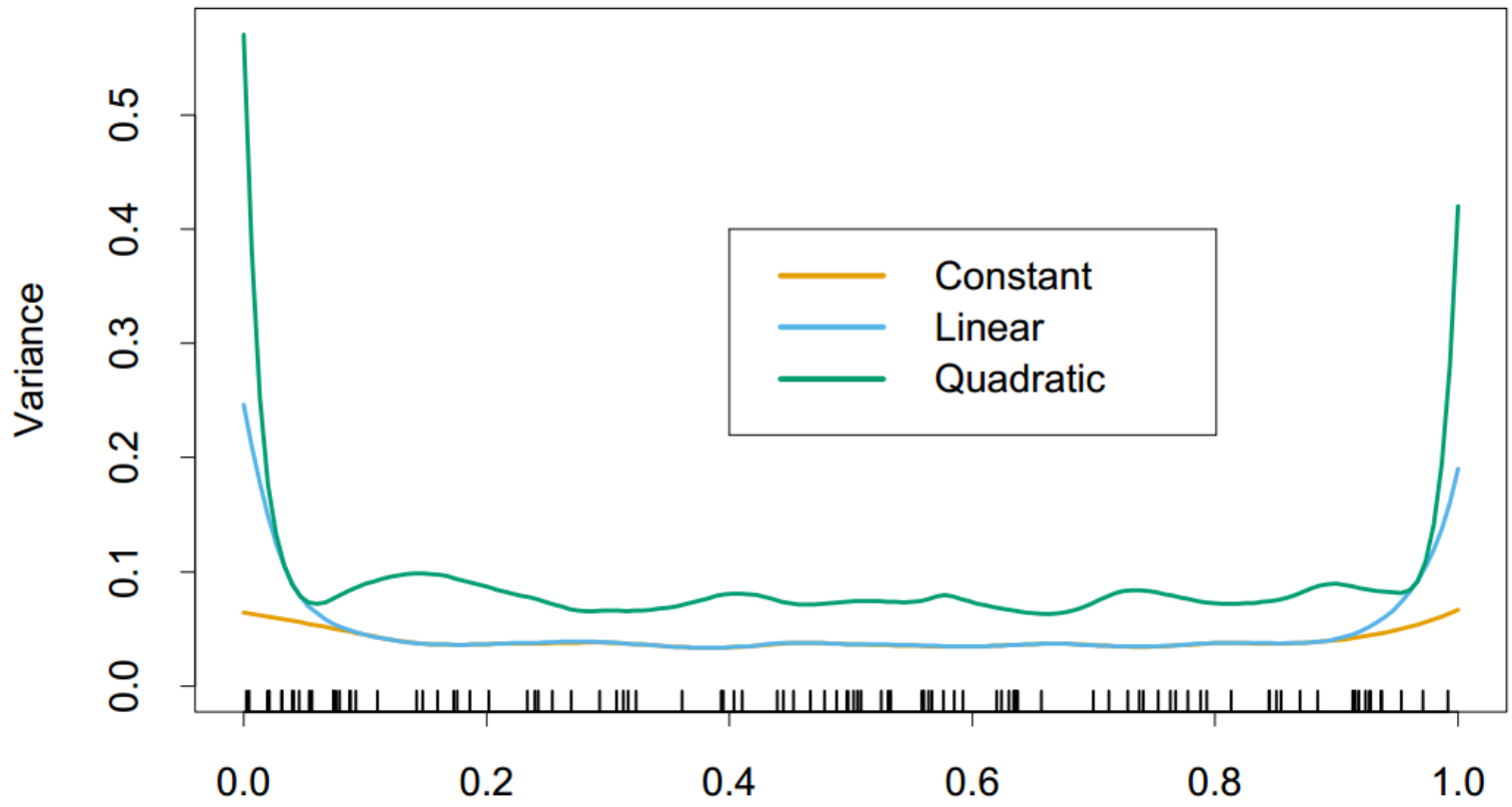


Figure: The variances functions $\|l(x)\|^2$ for local constant, linear and quadratic regression, for a metric bandwidth $(\lambda = 0.2)$ tri-cube kernel.

Local Regression in R^p

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

At each $x_0 \in R^p$ solve

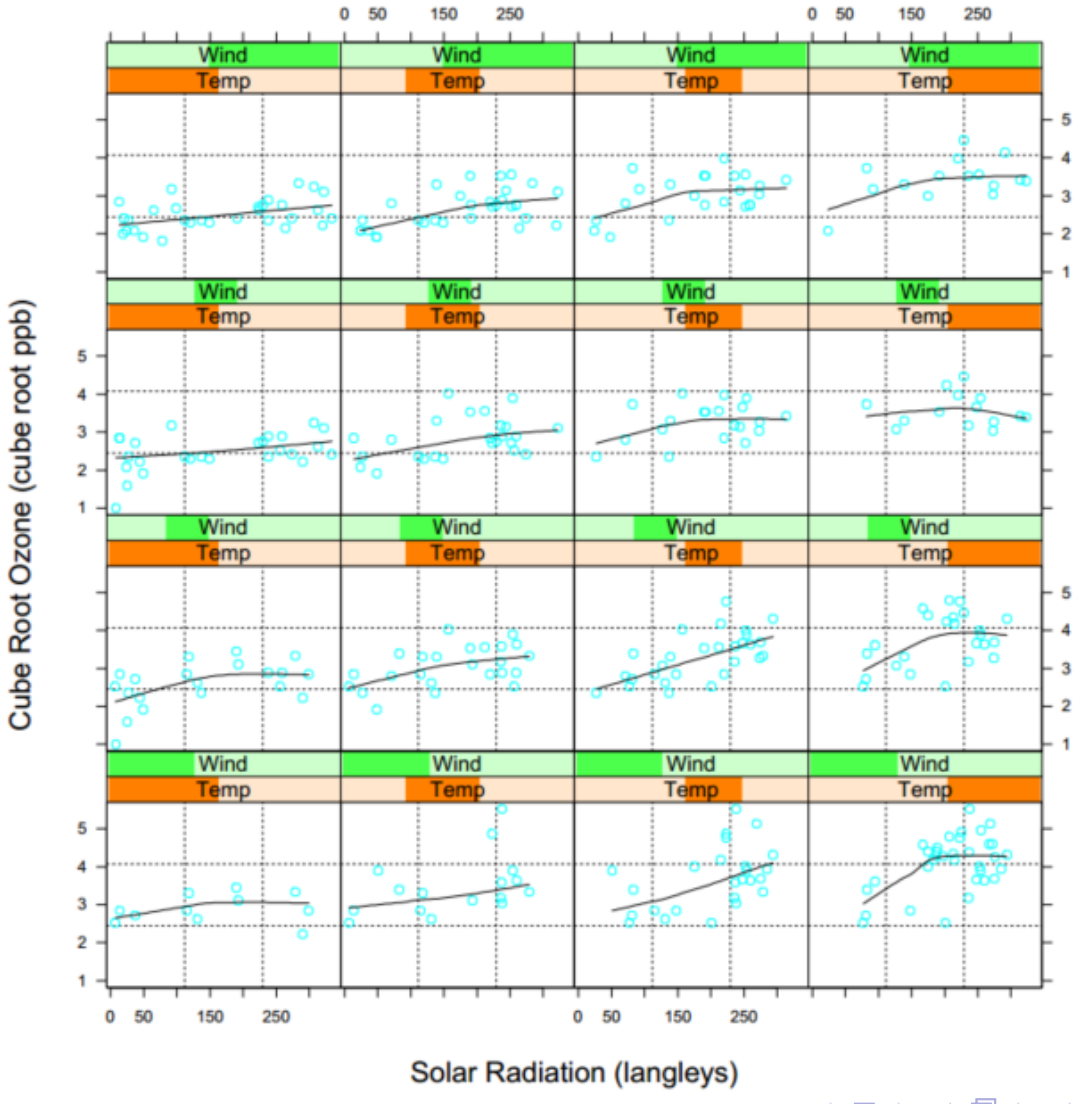
$$\min_{\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - b(x_i)^T \beta(x_0))^2$$

to produce the fit $\hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0)$. Typically the kernel will be a radial function, such as the radial Epanechnikov or tri-cube kernel

$$K_\lambda(x_0, x) = D\left(\frac{\|x - x_0\|}{\lambda}\right) \qquad \|\cdot\| \text{is the Euclidean norm}$$

- Boundary effects are a much bigger problem in two or higher dimensions, since the fraction of points on the boundary is larger.
- Local regression becomes less useful in dimensions much higher than two or three.
- It is impossible to simultaneously maintain localness ($\geq$low bias) and a sizable sample in the neighborhood ($\geq$low variance) as the dimension increases, without the total sample size increasing exponentially in p.

It is probably more useful in terms of understanding the joint behavior of the data.



Structured Local Regression Models in R^p

Structured Kernels

- standardize each variable to unit standard deviation
- use a positive semidefinite matrix A to weigh the different coordinates:

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

Structured Regression Functions

We are trying to fit a regression function $E(Y|X) = f(X_1, X_2, \cdots, X_p)$ in R^p, in which every level of interaction is potentially present. Analysis-of-variance (ANOVA) decompositions

$$f(X_1, X_2, \cdots, X_p) = \alpha + \sum_j g_j(X_j) + \sum_{k<l} g_{kl}(X_k, X_l) + \cdots$$

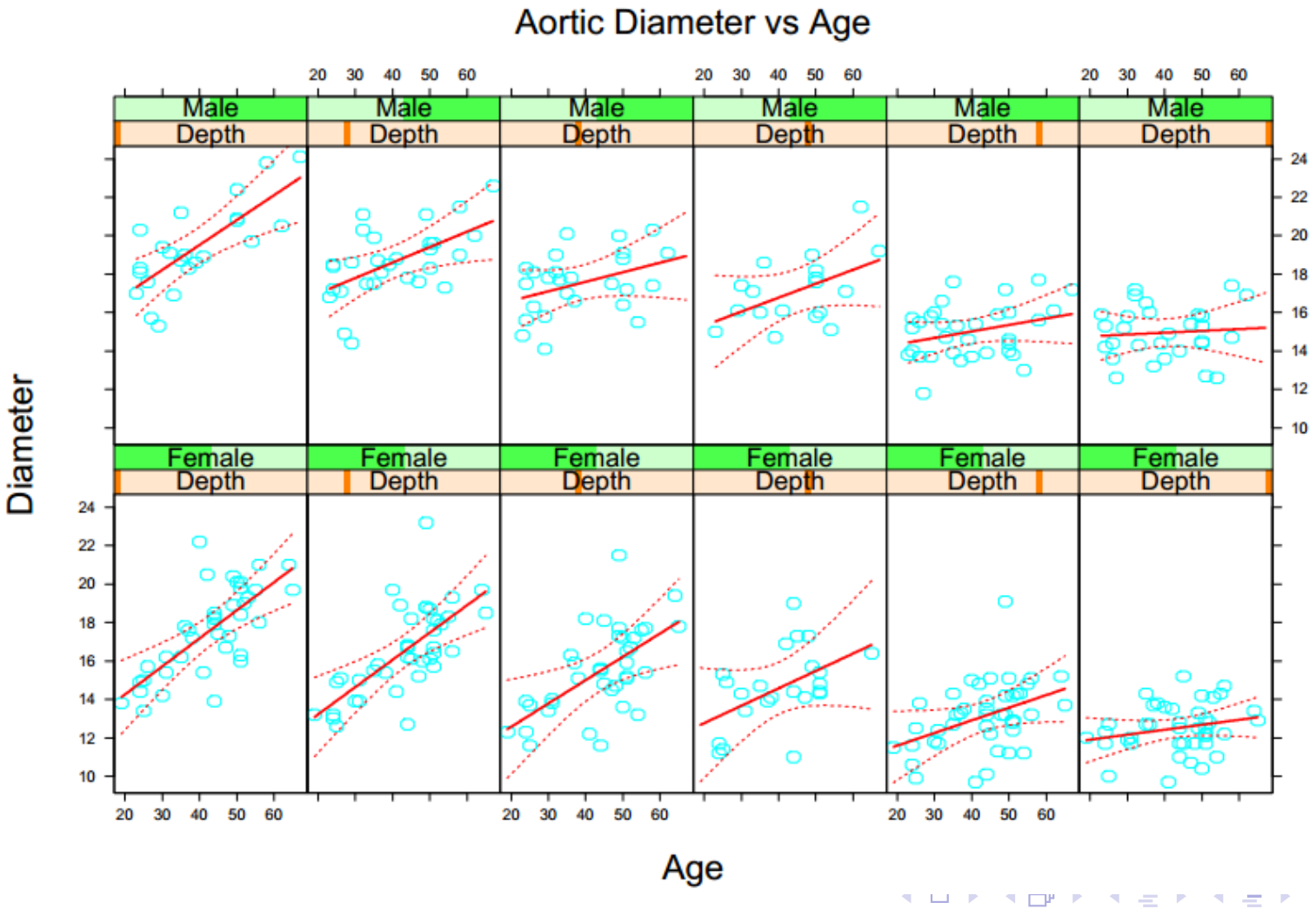
- **varying coefficient models**
  Suppose, for example, that we divide the p predictors in X into a set $(X_1, X_2, \cdots, X_q)$ with $q < p$, and the remainder of the variables we collect in the vector Z. We then assume the conditionally linear model

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

  For given Z, this is a linear model, but each of the coefficients can vary with Z. It is natural to fit such a model by locally weighted least squares:

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

Here we model the diameter of the aorta as a linear function of age, but allow the coefficients to vary with gender and depth down the aorta. We used a local regression model separately for males and females.



Local Likelihood and Other Models

- Associated with each observation y_i is a parameter $\theta_i = \theta(x_i) = x_i^T\beta$ linear in the covariate(s) x_i, and inference for β is based on the log-likelihood $l(\beta) = \sum_{i=1}^N l(y_i, x_i^T\beta)$. We can model $\theta(X)$ more flexibly by using the likelihood local to x_0 for inference of $\theta(x_0) = x_0^T\beta(x_0)$:

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

Many likelihood models, in particular the family of generalized linear models including logistic and log-linear models, involve the covariates in a linear fashion. Local likelihood allows a relaxation from a globally linear model to one that is locally linear.

Local Likelihood and Other Models

- As above, except different variables are associated with θ from those used for defining the local likelihood:

$$l(\theta(z_0)) = \sum_{i=1}^{N} K_\lambda(z_0, z_i) l(y_i, \eta(x_i, \theta(z_0))).$$

  For example, $\eta(x, \theta) = x^T\theta$ could be a linear model in x. This will fit a varying coefficient model $\theta(z)$ by maximizing the local likelihood.

- Autoregressive time series models of order k have the form $y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_k y_{t-k} + \varepsilon_t$. Denoting the *lag set* by $z_t = (y_{t-1}, y_{t-2}, \ldots, y_{t-k})$, the model looks like a standard linear model $y_t = z_t^T\beta + \varepsilon_t$, and is typically fit by least squares. Fitting by local least squares with a kernel $K(z_0, z_t)$ allows the model to vary according to the short-term history of the series. This is to be distinguished from the more traditional dynamic linear models that vary by windowing time.

Time Series Analysis?

Local Likelihood and Other Models

As an illustration of local likelihood, we consider the local version of the multiclass linear logistic regression model The data consist of features x_i and an associated categorical response $g_i \in \{1, 2, \ldots, J\}$, and the linear model has the form

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

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

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

we have centered the local regressions at x_0, so that the fitted posterior probabilities at x_0 are simply

$$\hat{\Pr}(G = j|X = x_0) = \frac{e^{\hat{\beta}_{j0}(x_0)}}{1 + \sum_{k=1}^{J-1} e^{\hat{\beta}_{k0}(x_0)}}.$$

Kernel Density Estimation and Classification

- **Kernel Density Estimation**

  Parzen estimate

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

  In this case a popular choice for K_λ is the Gaussian kernel

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

  Letting ϕ_λ denote the Gaussian density with mean zero and standard-deviation λ, then

$$\hat{f}_X(x) = \frac{1}{N}\sum_{i=1}^{N} \phi_\lambda(x - x_i) = (\hat{F} * \phi_\lambda)(x)$$

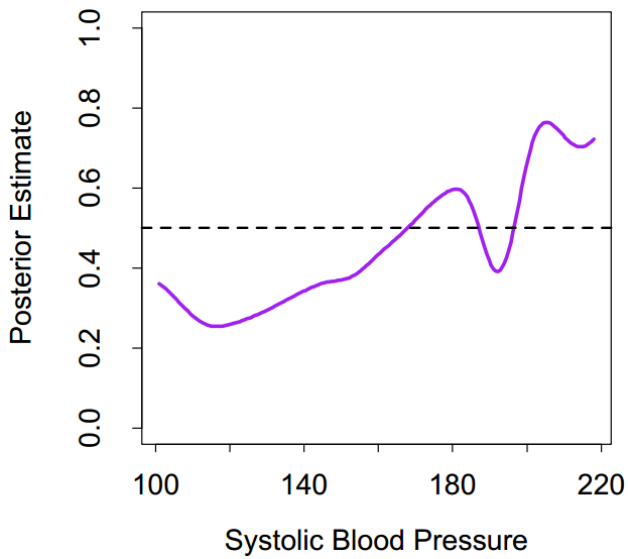
  In R^p the natural generalization of the Gaussian density estimate amounts to using the Gaussian product kernel

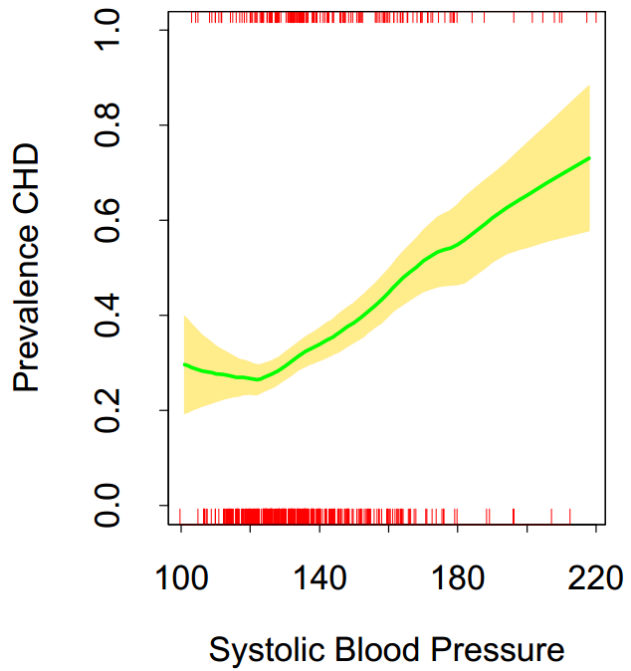
$$\hat{f}_X(x_0) = \frac{1}{N(2\lambda^2\pi)^{p/2}}\sum_{i=1}^{N} e^{-\frac{1}{2}(\|x_i - x_0\|/\lambda)^2}$$

Kernel Density Classification

Use nonparametric density estimates for classification in a straightforward fashion using Bayes theorem.

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





The Naive Bayes Classifier

The naive Bayes model assumes that given a class $G = j$, the features X_k are independent:

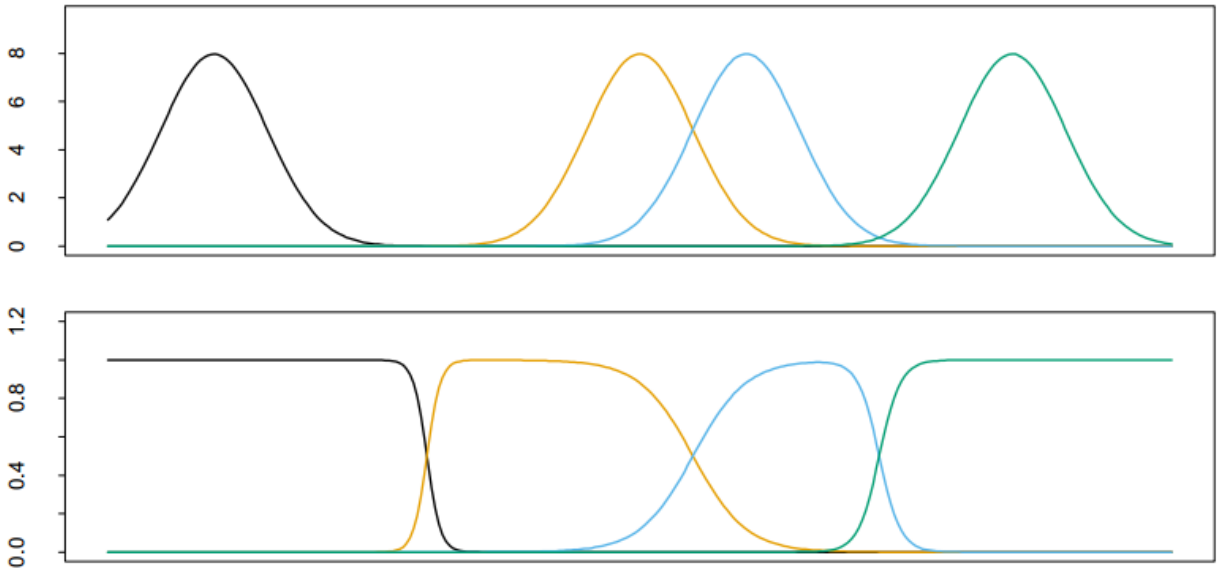
$$f_j(X) = \prod_{k=1}^{p} f_{jk}(X_k)$$

we can derive the logit-transform (using class J as the base):

$$\log \frac{Pr(G = l|X)}{Pr(G = J|X)} = \log \frac{\pi_l f_l(X)}{\pi_J f_J(X)} = \log \frac{\pi_l \prod_{k=1}^{p} f_{lk}(X_k)}{\pi_J \prod_{k=1}^{p} f_{Jk}(X_k)}$$

$$= \log \frac{\pi_l}{\pi_J} + \sum_{k=1}^{p} \log \frac{f_{lk}(X_k)}{f_{Jk}(X_k)}$$

$$= \alpha_l + \sum_{k=1}^{p} g_{lk}(X_k)$$



While it would seem attractive to reduce the parameter set and assume a constant value for $\lambda_j = \lambda$, this can have an undesirable side effect of creating holes-regions of R^p where none of the kernels has appreciable support. Renormalized radial basis functions,

$$h_j(x) = \frac{D(\|x - \xi_j\|)/\lambda}{\sum_{k=1}^{M} D(\|x - \xi_k\|)/\lambda}$$

avoid this problem.

- Optimize the sum-of-squares with respect to all the parameters:

$$\min_{\{\lambda_j,\xi_j,\beta_j\}_1^M} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{M} \beta_j \exp\left\{ -\frac{(x_i-\xi_j)^T(x_i-\xi_j)}{\lambda_j^2} \right\} \right)^2$$

- Estimate the $\{\lambda_j, \xi_j\}$ separately from the β_j

- Given the former, the estimation of the latter is a simple least squares problem. Often the kernel parameters λ_j and ξ_j are chosen in an unsupervised way using the X distribution alone.

An example

The Nadaraya-Watson kernel regression estimator in R^p can be viewed as an expansion in renormalized radial basis functions

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

Mixture Models for Density Estimation and Classification

The mixture model is a useful tool for density estimation, and can be viewed as a kind of kernel method. The Gaussian mixture model has the form

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \mathbf{\Sigma}_m) \qquad (*)$$

with mixing proportions α_m, $\sum_m \alpha_m = 1$, and each Gaussian density has a mean μ_m and covariance matrix $\mathbf{\Sigma}_m$. In general, mixture models can use any component densities in place of the Gaussian in $(*)$: the Gaussian mixture model is by far the most popular.

The parameters are usually fit by maximum likelihood, using the EM algorithm as described in Chapter 8. Some special cases arise:

- If the covariance matrices are constrained to be scalar: $\mathbf{\Sigma}_m = \sigma_m \mathbf{I}$, then $(*)$ has the form of a radial basis expansion.
- If in addition $\sigma_m = \sigma > 0$ is fixed, and $M \uparrow N$, then the maximum likelihood estimate for $(*)$ approaches the kernel density estimate (6.22) where $\hat{\alpha}_m = 1/N$ and $\hat{\mu}_m = x_m$.

The mixture model also provides an estimate of the probability that observation i belongs to component m,

$$\hat{r}_{im} = \frac{\hat{\alpha}_m \phi(x_i; \hat{\mu}_m, \hat{\Sigma}_m)}{\sum_{k=1}^{M} \hat{\alpha}_k \phi(x_i; \hat{\mu}_k, \hat{\Sigma}_k)}$$