Basis expansions, kernels and support vector machines

Hastie, Tibshirani, Friedman Ch 5, 6, 12

Kevin Murphy Ch. 14

CS 6140
Machine Learning
Professor Olga Vitek

March 2, 2017

Basis Expansion

HTF Ch5

Move beyond linearity

- Linear regression, logistic regression, LDA
  - Classification by linear hyperplanes
  - Easy to fit and to interpret

- $f(Y|\mathbf{X})$ is typically non-linear and non-additive
  - augment X with transformations of $\mathbf{X}$
  - use as input in linear models
  - Denote mth transformation $h_m(\mathbf{X}) : \mathbb{R}^p \to \mathbb{R}^p$
  - Model $f(\mathbf{X}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{X})$
  - $f(\mathbf{X})$ is linear in $h_m(\mathbf{X})$
  - This is a *linear basis expansion* in $\mathbf{X}$

Linear basis expansion

- Linear model
  - $h_m(\mathbf{X}) = X_m$
- Polynomial terms (Taylor expansion)
  - $h_m(\mathbf{X}) = X_m^2$ or $h_m(\mathbf{X}) = X_i X_j^2$
  - # variables $\uparrow$ exponentially in p
  - tweak one region $\uparrow$ flap another
- Functions of a vector
  - $h_m(\mathbf{X}) = log(X_j)$, $h_m(\mathbf{X}) = \sqrt{X_j}$, $h_m(\mathbf{X}) = ||\mathbf{X}||$
- Indicators $h_m(\mathbf{X}) = I(L_m \leq X_k < U_m)$
  - Break the range of X_k into regions
  - Piecewise constant model in each region
- Piece-wise polynomials and splines
  - Dictionary $\mathcal{D}$ of basis functions
  - Method for controlling model complexity: *restriction, selection, regularization*
  - Restriction: $f(\mathbf{X}) = \sum_{j=1}^{p} f_j(X_j) = \sum_{j=1}^{p} \sum_{m=1}^{M_j} \beta_{jm} h_{jm}(X_j)$

Piecewise fits

- Assume $\mathbf{X}$ is one-dimensional (i.e., X)
  - Divide domain(X) into contiguous intervals
  - $f(X)$: a separate polynomial in each interval

- Example: piecewise constant
  - $h_1(X) = I(X < \xi_1),\ h_2(X) = I(\xi_i \leq X \leq \xi_2),$
    $h_3 = I(\xi_2 \leq X)$
  - $f(X) = \sum_{m=1}^{3} \beta_m h_m(X)$
  - $\widehat{\beta}_m = \bar{Y}_m$, the mean of mth region

- Example: piecewise linear
  - Need extra basis $h_{m+3} = h_m(X) \cdot X,\ m = 1, \ldots, 3$

- Example: restricted piecewise linear
  - $f(\xi_1^-) = f(\xi_1^+) \ \rightarrow\ \beta_1 + \xi_1\beta_4 = \beta_2 + \xi_1\beta_5$
  - 3 intervals: 4 free parameters out of 6
  - Alternatively: $h_1(X) = 1,\ h_2(X) = X,$
    $h_3(X) = (X - \xi_1)_+,\ h_4(X) = (X - \xi_2)_+$

Piecewise fit

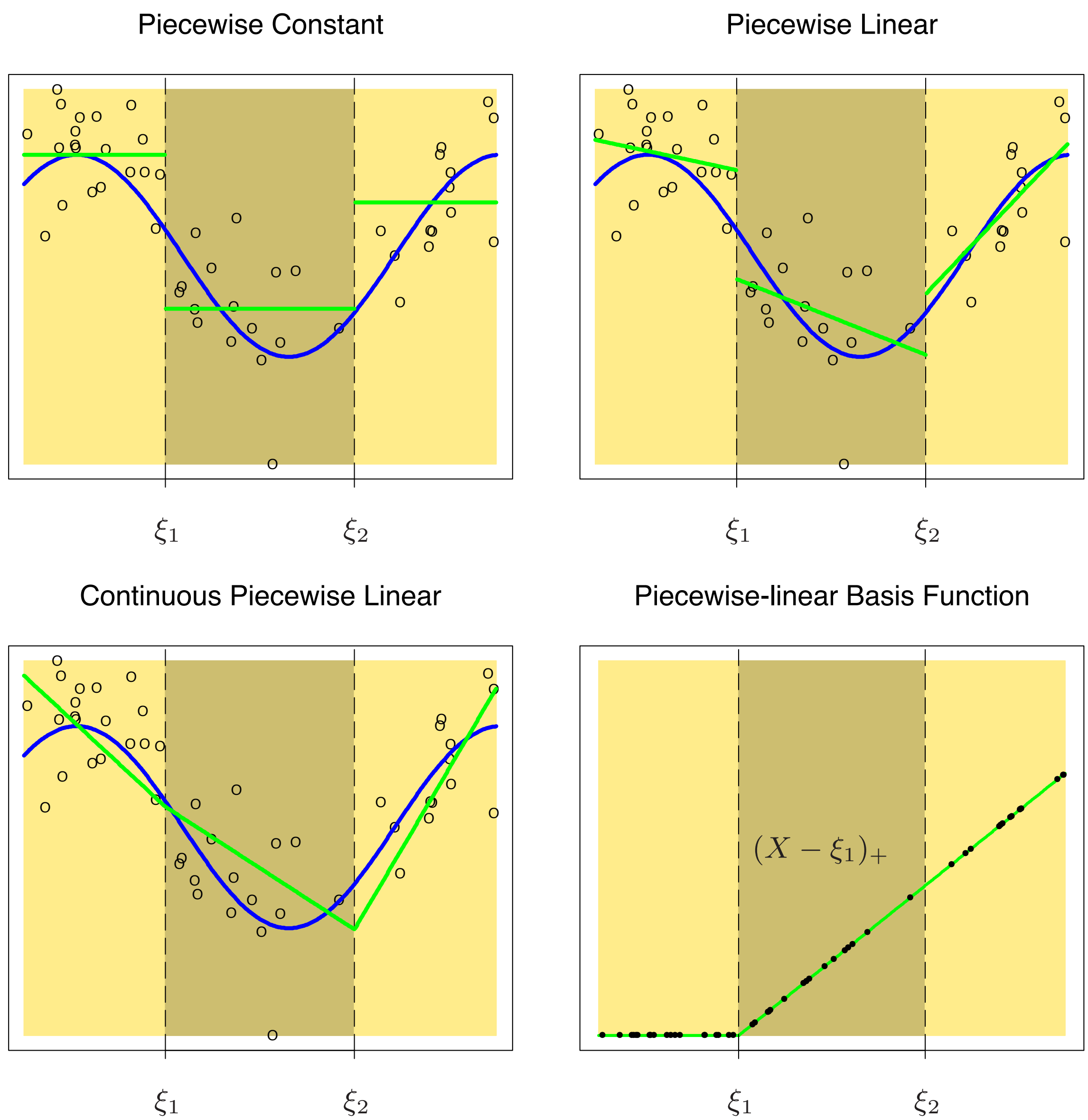


Fig. 5.1. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

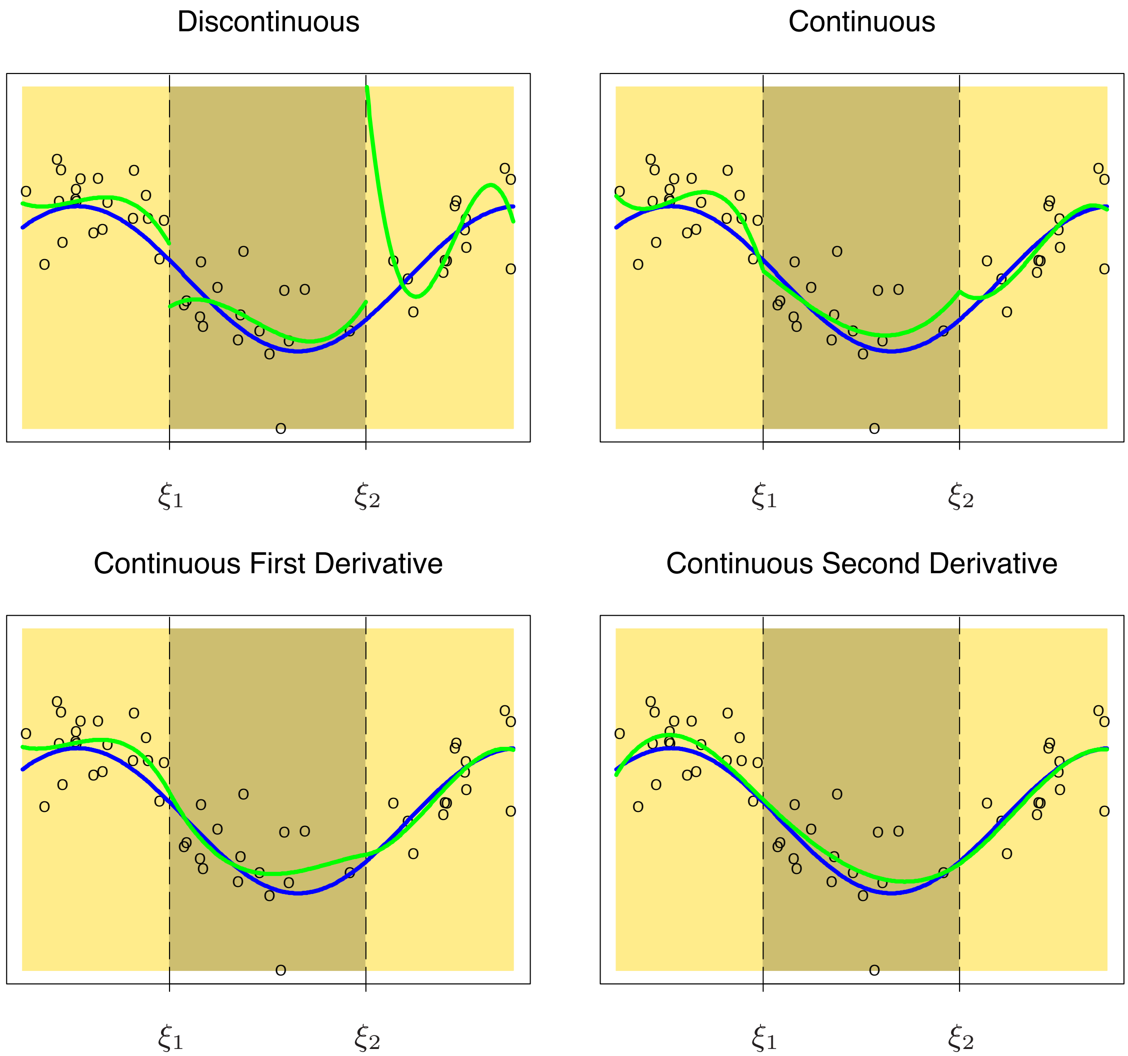
Splines

- Smoother functions
  - Increase the order of polynomials
  - Cubic splines: continuous to second derivative
  - $h_1(X) = 1, h_s(X) = X, h_3(X) = X^2, h_4(X) = X^3$
    $h_5(X) = (X - \xi_1)_+^3,\ h_6(X) = (X - \xi_2)_+^3$

- Example on the next page
  - 6 basis functions
  - 6-dimensional linear space of functions
  - (3 regions) $\times$ (4 parameters per region)
    - (2 knots) $\times$ (3 constrains per knot) $= 6$

- Order-M spline with knots ξ_j, $j = 1, \ldots, K$
  - Piecewise polynomial up to order M
  - Has continuous derivatives up to order $M - 2$.
  - Piecewise constant fit is order-1 spline.
    Piecewise continuous fit is order-2 spline.
    Cubic spline is order-4 spline.
  - Basis set: $h_j(X) = X^{j-1},\ j = 1, \ldots, M$ and
    $h_{M+l}(X) = (X - \xi_l)_+^{M-1},\ l = 1, \ldots, K$

Piecewise cubic polynomials



Cubic spline is the lowest-order spline for which the knot-discontinuity is not visible to human eye

Fig. 5.2. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Natural cubic splines

- Need for extra stability
  - Polynomials and splines have erratic behavior near the boundaries
  - Variance explodes
  - Extrapolation is problematic

- Natural cubic splines: extra constraints
  - Linear functions beyond boundary knots
  - K knots $=$ K basis functions
  - Basis for cubic splines $\rightarrow$ impose constraints
  - Start from basis set, impose boundary constraint, derive reduced basis:

$$N_1(X) = 1,\ N_2(X) = X,$$
$$N_{k+2}(X) = d_k(X) - d_{K-1}(X)$$
$$\text{where } d_k(X) = \left\{(X-\xi_k)_+^3 - (X-\xi_K)_+^3\right\} / \{\xi_K - \xi_k\}$$

  - Second and third derivatives are 0 for $X \geq \xi_K$

Pointwise variance curve

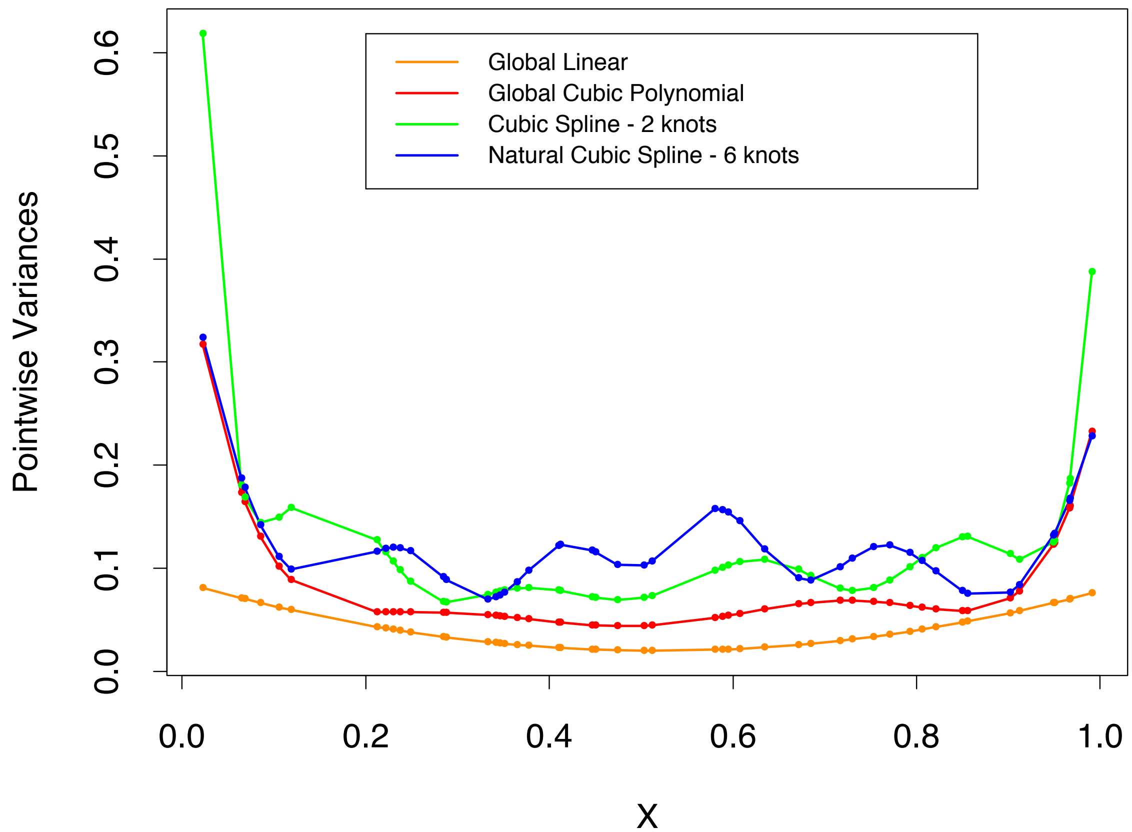


Fig. 5.3. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Example: South African Heart Disease

- Logistic regression

$$\frac{\Pr\{\texttt{chd} = 1|X\}}{1 - \Pr\{\texttt{chd} = 1|X\}} = \theta_0 + h_1(X_1)'\theta_1 + \ldots + h_p(X_p)'\theta_p$$

  - θ_j are vectors of coefficients multiplying the vector of natural spline basis functions h_j
  - Use 4 natural spline basis functions for each term in the model
  - Knots chosen at uniform quantiles of X_j:
    3 internal knots +
    2 boundary knots at the extremes of X_j
  - Binary predictor has a single coefficient
  - More compactly: combine p vectors of basis functions + constant term in a big vector $h(X)$, and model $h(X)'\theta$
  - Backward stepwise selection + AIC to drop terms
  - Plot prediction $\pm 2 \cdot SE$

Fit: $\hat{f}_j(X_j) = h_j(X_j)' \hat{\theta}_j$

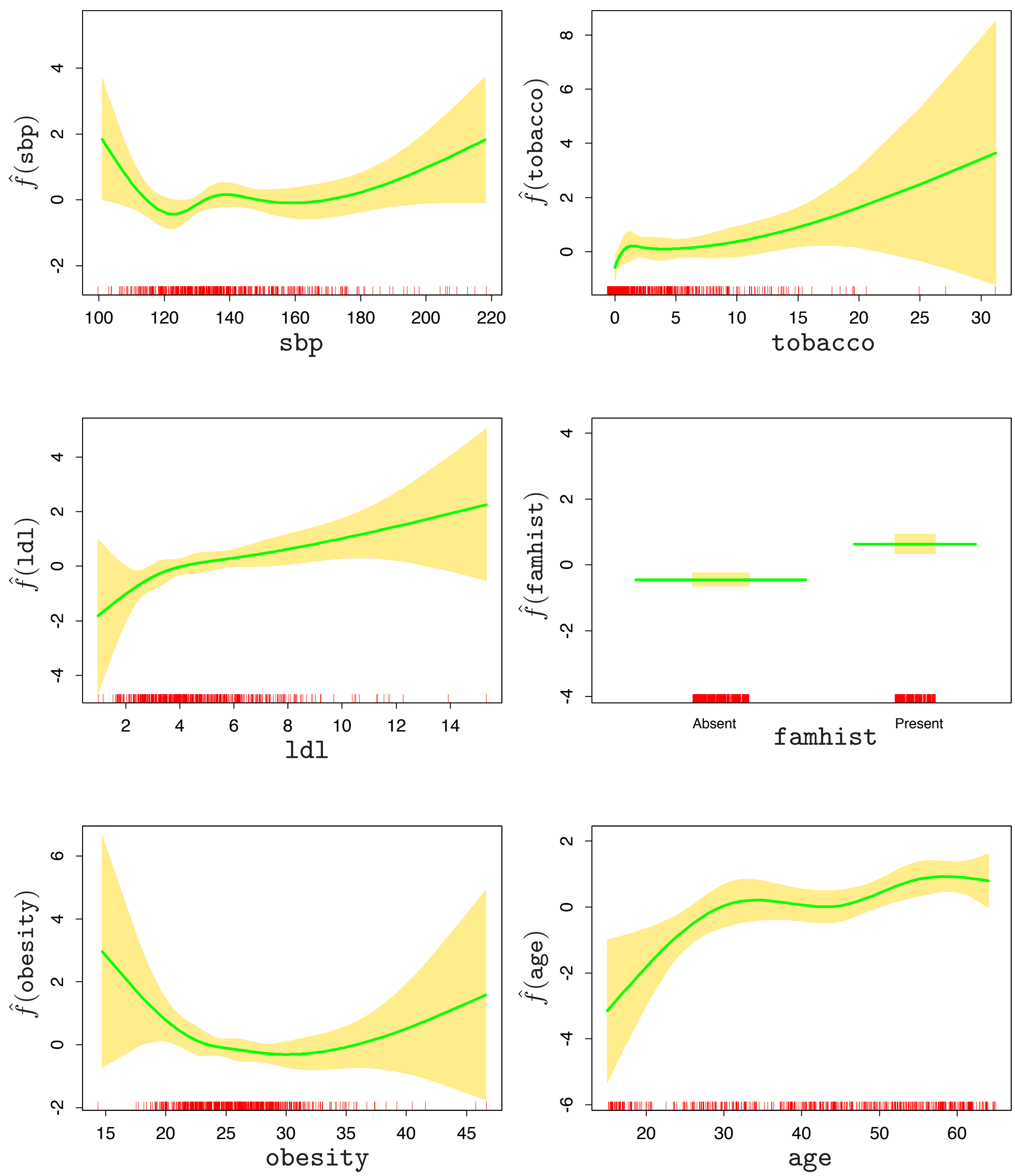
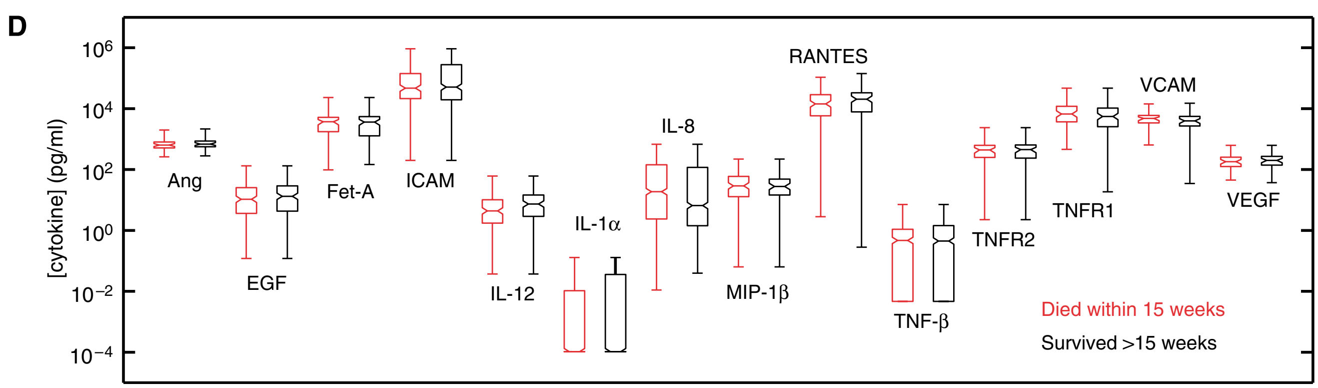


Fig. 5.4. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Example: disease classification

Knickerbocker *et al.* "An integrated approach to prognosis using protein microarrays and nonparametric methods". *Molecular Systems Biology*, 2007

- Goal: predict mortality in patients w/kidney dialysis
  - 468 patients, 208 died within 15 weeks of diagnoses
  - 14 proteins ("messenger" molecule that allows cells to communicate and alter function)
  - 11 clinical characteristics (age, race, bmi...)
  - Proteins were uninformative in isolation



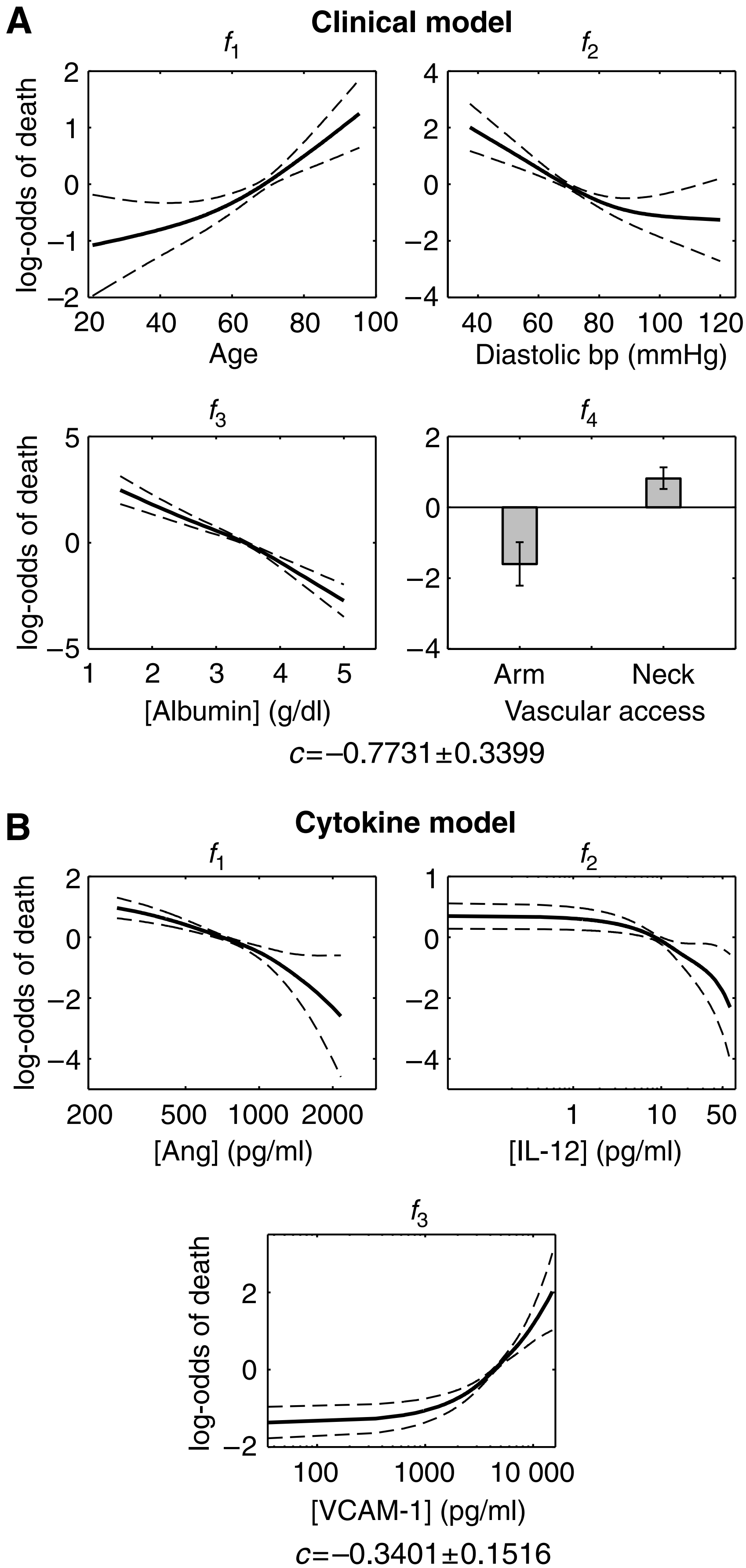
Approach

- Separate predictive model for clinical and molecular measurements
  - Logistic regression (Y = death within 15 weeks)

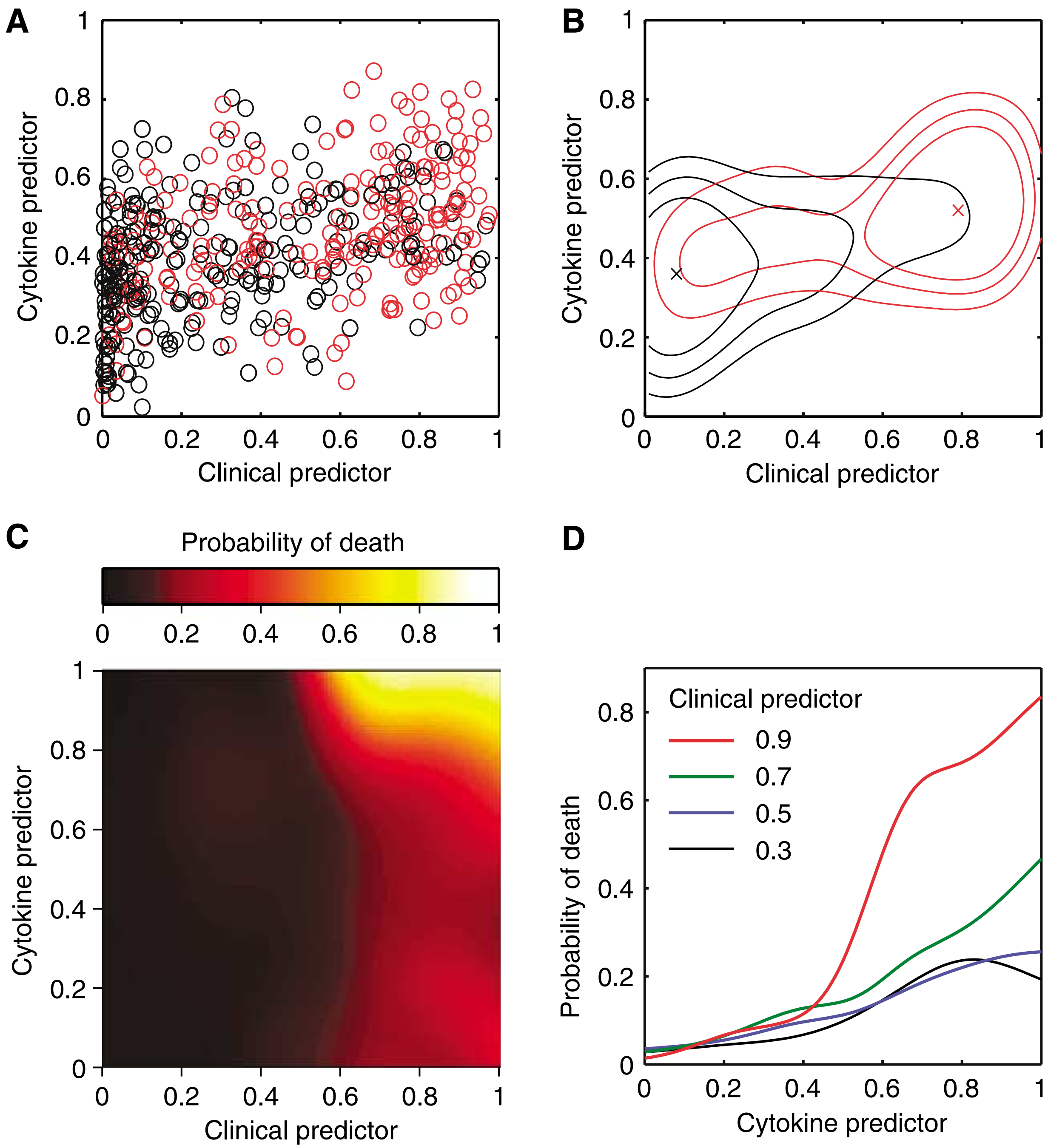
$$\text{log-odds of death} = \log(P_{\text{sample}}(\text{death})/P_{\text{sample}}(\text{survival})) = c + \sum_{p=1}^{M} b_p x_p$$

  - Additive model to reduce feature space
  - Exhaustive search for best M-variable model
  - Non-parametric version of the best model: splines

$$\log - \text{odds of death} = \log(P_{\text{sample}}(\text{death})/P_{\text{sample}}(\text{survival})) = c + \sum_{p=1}^{M} f_p(x_p)$$


A
Clinical model
f_1
log-odds of death
2
1
0
−1
−2
20
40
60
80
100
Age
f_2
4
2
0
−2
−4
40
60
80
100
120
Diastolic bp (mmHg)
f_3
log-odds of death
5
0
−5
1
2
3
4
5
[Albumin] (g/dl)
f_4
2
0
−2
−4
Arm
Neck
Vascular access
c=−0.7731±0.3399
B
Cytokine model
f_1
log-odds of death
2
0
−2
−4
200
500
1000
2000
[Ang] (pg/ml)
f_2
1
0
−2
−4
1
10
50
[IL-12] (pg/ml)
f_3
log-odds of death
2
0
−2
100
1000
10 000
[VCAM-1] (pg/ml)
c=−0.3401±0.1516

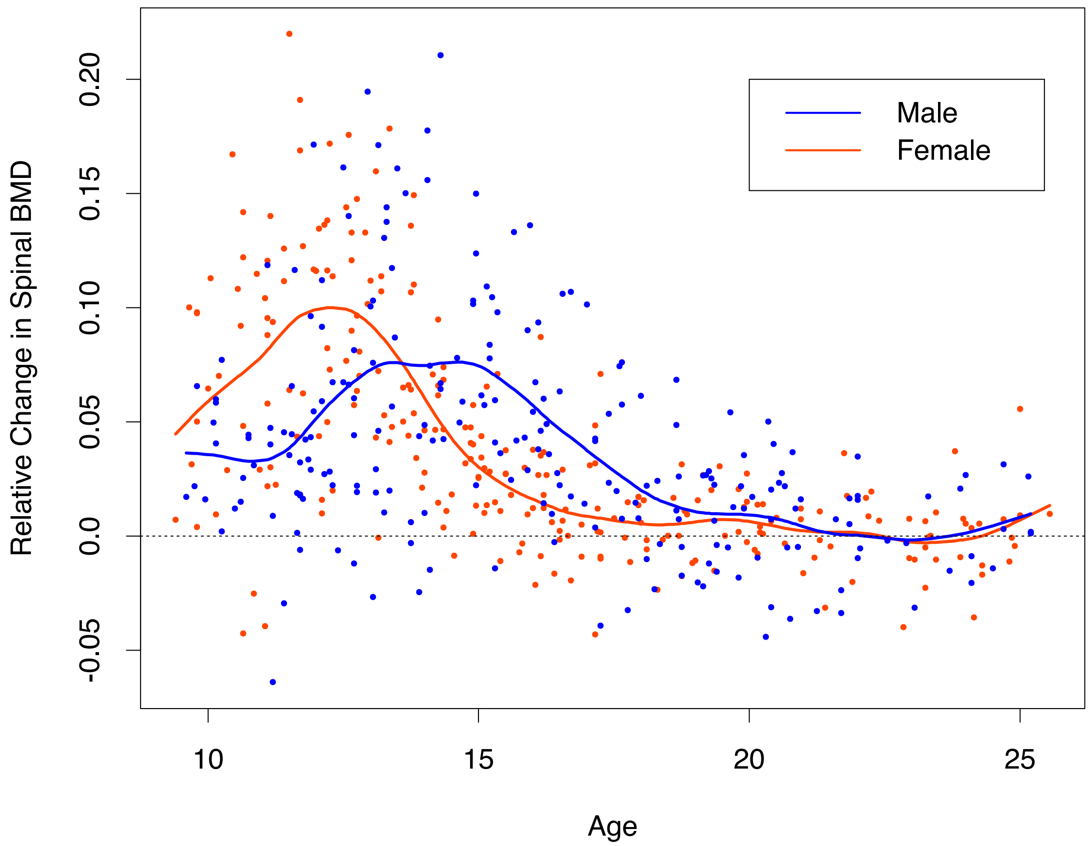
Combined prediction



Controlling model complexity (fixed knots)

- Restriction
  - Limit the class of functions
    $$f(\mathbf{X}) = \sum_{j=1}^{p} f_j(X_j) = \sum_{j=1}^{p} \sum_{m=1}^{M_j} \beta_{jm} h_{jm}(X_j)$$
  - Model complexity $\sim$ number of basis functions
- Selection
  - AIC, BIC, significance testing
  - Adaptively include basis functions that contribute to prediction (include CART, boosting...)
- Regularization
  - Include the entire dictionary of basis functions, but restrict the coefficients
    $$\text{RSS}(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \left\{f''(t)\right\}^3 dt$$
  - λ is a smoothing parameter ($\lambda = 0$: any function; $\lambda = \infty$: linear fit)
  - Maximized with a natural cubic spline
  - Equivalent to generalized ridge regression

Example



Rel. change in bone mineral density $\sim$ age, $\lambda = 0.00022$

$$\begin{aligned} \text{EPE}(\hat{f}_\lambda) &= E\left(Y - \hat{f}_\lambda\right)^2 \\ &= \text{Var}(Y) + E\left[\text{Bias}^2(\hat{f}_\lambda) + \text{Var}(\hat{f}_\lambda)\right] \\ &= \sigma^2 + \text{MSE}(\hat{f}_\lambda) \end{aligned}$$

Choose λ by cross-validation

Fig. 5.6. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Kernel Methods

HTF Ch6

One-dimensional kernel smoothers

- Different model at each point x_0
  - Only use observations close to target x_0
  - Weigh the neighbors x_1 with a kernel $K_\lambda(x_0, x_i)$

  - Weight based on distance from x_0
  - λ is a parameter
  - The resulting function is smooth

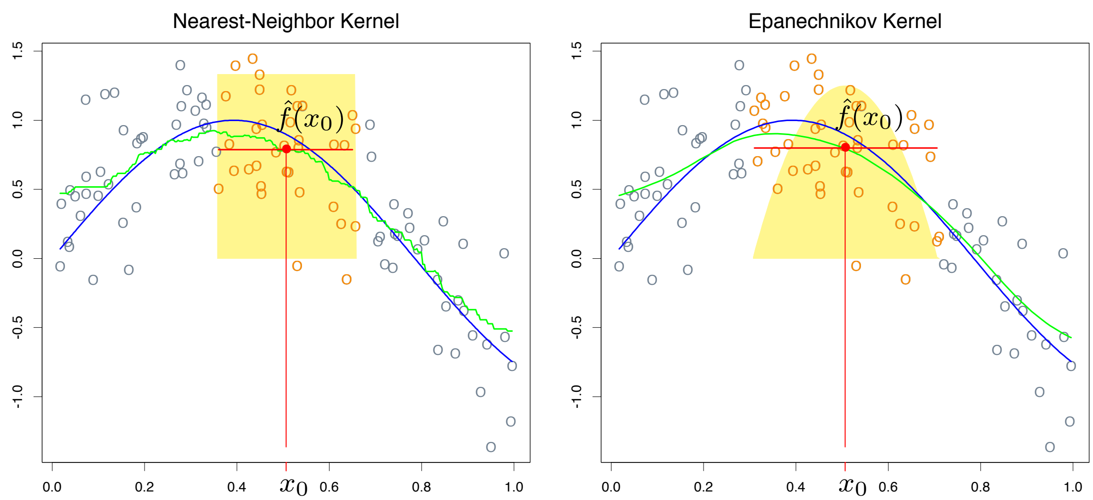
- K nearest neighbors
  - $\hat{f}(x_0) = \text{Ave}(y_i | x_i \in N_k(x_0))$ (discontinuous in x)

- Nadaraya-Watson kernel-weighted averaga
  - $\hat{f}(x_0) = \frac{\sum_{i=1}^N K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^N K_\lambda(x_0, x_i)}$, where
    $K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right)$ and

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

  - Points near the boundary have weight ~ 0
    $\rightarrow$ smoothing

Example



- 100 pairs (x_i, y_i)
- Green: Left: 30-NN running mean.
  Right: Kernel-weighted average, $\lambda = 0.2$
- Orange: observations contributing to the fit

Fig. 6.1. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Adaptive width

- Define $h_\lambda(x_0)$ a width function that determines the neighborhood of x_0

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

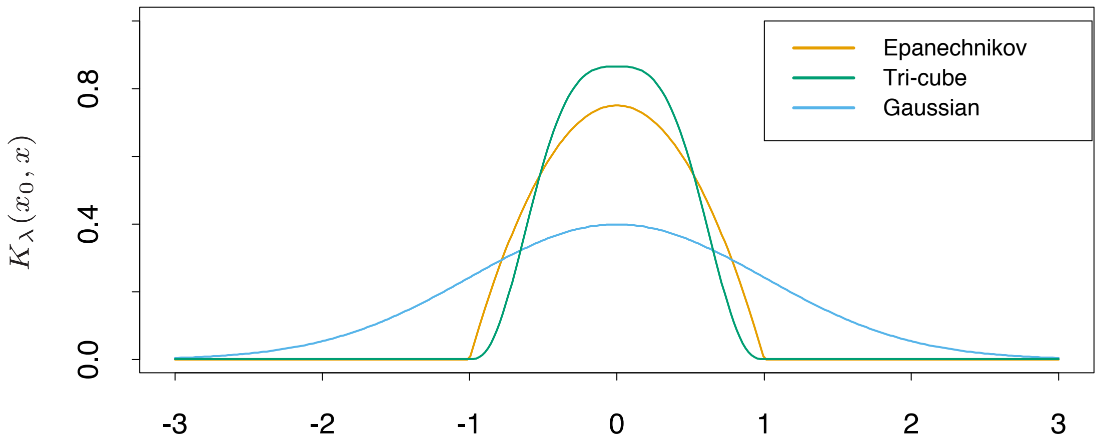
- Concerns:

  - Determine λ. Larger $\lambda \rightarrow$ lower variance but higher bias

  - $h_\lambda(x)$ constant $\rightarrow$ variance inversely proportional to density of points

  - Nearest neighbor $\rightarrow$ bias inversely proportional to density of points

Kernel examples



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

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

- Tri-cube function

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

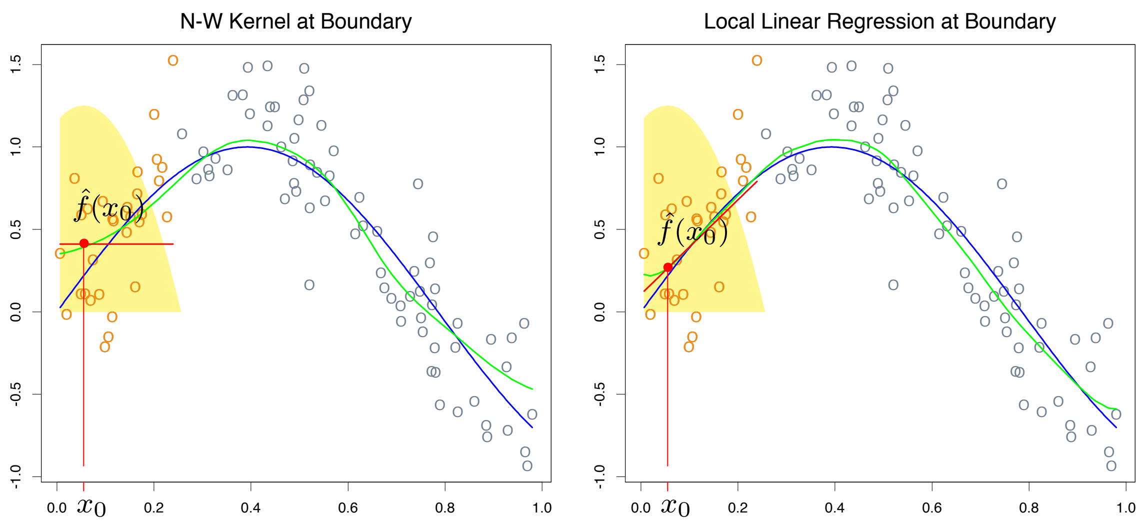
Fig. 6.2. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Local linear regression

- Smoothly varying locally weighted average
  - Problem: asymmetry of the kernel on the boundary of the domain $\rightarrow$ bias
  - Solution: fit straight lines rather than constants
  - Also helps if values of x are unequally spaced

- Separate weighted least squares problem
  - At each target point x_0:
    $$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)\,[y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$
  - Define $b(x)' = (1, x)$
    $\mathbf{B}$ the $X \times 2$ regression matrix with ith row $b(x)'$
    $\mathbf{W}(x_0)$ the $N \times N$ diagonal matrix diag$(K_\lambda(x_0, x_i))$.
    Then $\hat{f}(x_0) = b(x_0)'\,(\mathbf{B}'\mathbf{W}(x_0)\mathbf{B})^{-1}\,\mathbf{B}'\mathbf{W}(x_0)\mathbf{y}$
  - Estimate at a single point
    $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$

Example



- Left: locally weighted average:
  bias near the boundaries of X

- True function is linear, but most observations in the neighborhood exceed the target point.

- Right: locally weighted linear regression removes the bias to first order

Fig. 6.3. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Extensions / comments

- Local polynomial regression
  - Minimize
    $$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{N} K_\lambda(x_0,x_i)\left[y_i-\alpha(x_0)-\sum_{j=1}^{d}\beta_j(x_0)x_i^j\right]^2$$
  - Solution $\hat{f}(x_0)=\hat{\alpha}(x_0)+\sum_{j=1}^{d}\hat{\beta}_j(x_0)x_0^j$
  - Avoid "trimming the hills and filling the valleys"

- Bias-variance tradeoff
  - Small window $\rightarrow$ low bias, high variance

- Default hoice of λ
  - Epanechnikov or tri-cube: radius of support region
  - Gaussian: standard deviation
  - Better option: cross-validation

Example

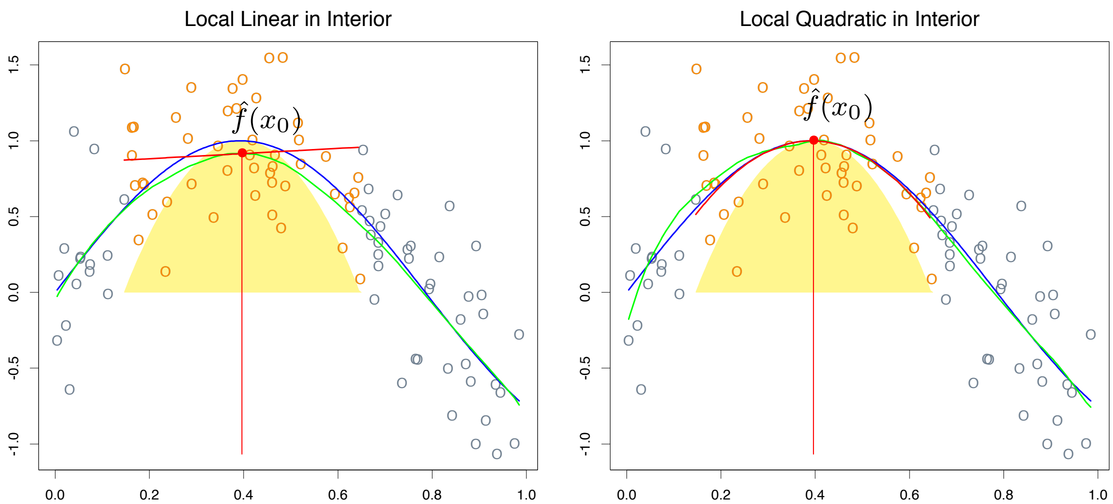
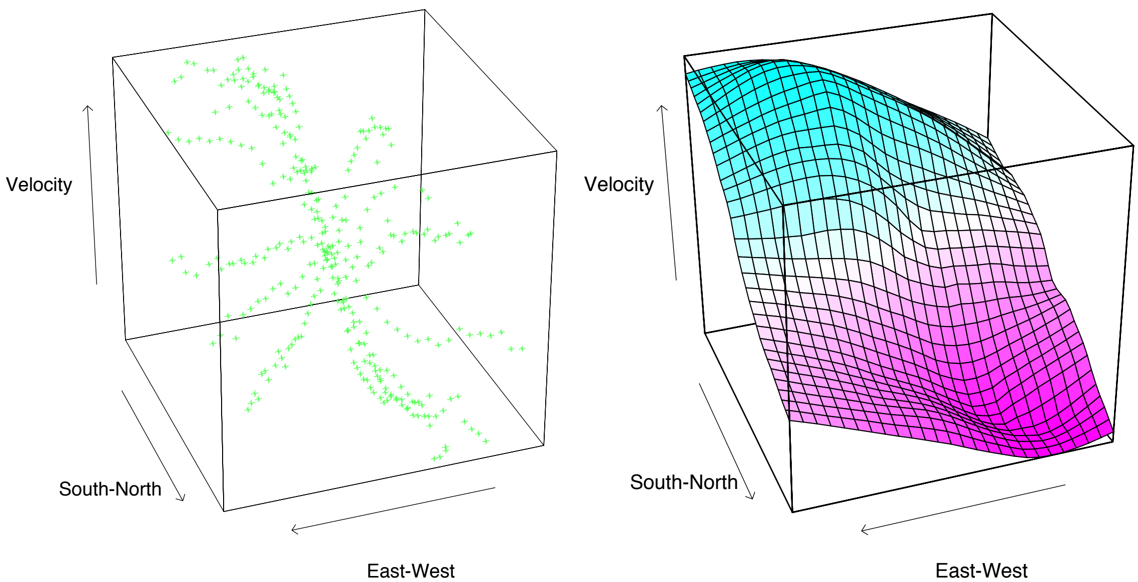


Fig. 6.5. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008

Local regression in $\mathbb{R}^p$



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

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

Fig. 6.8. Hastie, Tibshirani, Friedman
The Elements of Statistical Learning 2008