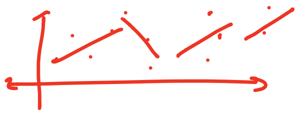
Local Learning



e.g. local linear regression

Idea: Only build model based on local points near x_0.

Estimate: $\alpha(x_0)$, $\beta(x_0) = (\beta_1(x_0) \dots, \beta_p(x_0))^T$

Under the least squared model we see

$$(\hat{\alpha}, \hat{\beta}) = \underset{(\alpha,\beta)}{\operatorname{argmin}} \sum_{i=1}^{N} \{ w_i [y_i - \alpha_i(x_i) - \beta(x_i)^T x_0 \}^2$$

Ex: $d = 0$, local constant regression

$$\hat{\alpha}(x_0) = \sum_{i=1}^{N} w_i y_i \Big/ \sum_{i=1}^{n} w_i \qquad \text{Kernel-Nearest Neighbors}$$

$$= \sum_{i=1}^{n} \left(\frac{w_i}{\sum_{j=1}^{n} w_j} \right) y_i \equiv \sum_{i=1}^{n} \ell_i(x_0) y_i \equiv \ell^T(x_0) Y$$

So all together we have

$$\hat{y} = L^T Y \qquad L = \left(\ell(x_1) \cdots | \ell(x_n) \right)$$

Local Linear Regression

$$(\alpha(x_0), \beta(x_0)) = \underset{(\alpha,\beta)}{\arg\min} = \sum_{i=1}^{n} w_i^{0} (y_i - \alpha(x_0) - \beta(x_0) x_i)^2$$

$$\begin{pmatrix} \hat{\alpha}(x_0) \\ \hat{\beta}(x_0) \end{pmatrix} = (X^T W^{(0)} X)^{-1} X^T W^{(0)} y$$

Choice of Weights

$w_i^{(0)} = D(x_i - x_0)$ that has certain properties.

i.e. $\int D(u)\,du = 1$, $-D(u) = D(-u)$

Epanechnikov Kernel: $D(u) = \frac{3}{4}(1-u^2)$

Gaussian: $D(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$

The weights then are given $w_i^{(0)} = K_\lambda(x_i, x_0) = \frac{1}{\lambda} D\left(\frac{|x_i - x_0|}{\lambda}\right)$

Definition of Sample Training Error

$$\overline{err} = \frac{1}{N} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$$

$$Err = L_{(x,y)}(Y, \hat{f}(X))$$