Kernel smoothing generalizes to multiple predictors but

The curse of dimensionality gets in the way

High dimensional samples are always sparse

Suppose we have training sample $(\underline{x}_1, y_1) \dots (\underline{x}_n, y_n)$, with $\underline{x}_i \sim U[0,1]^{10}$

Suppose we want to estimate $E(Y|\underline{x})$ using box kernel with side length 0.1
$\Rightarrow$ volume (footprint) of kernel $= 0.1^{10}$
$\Rightarrow$ Expected # of training obs in box $= n * 0.1^{10}$
$\Rightarrow$ not much averaging

Suppose we use a kernel with volume $= 0.1 \Rightarrow$ width of box $= 0.8$
$\Rightarrow$ "local" averaging is not local.

High dim data has counter-intuitive geometry

Consider a regular grid

In 2 dim, every grid cell has $3^2 - 1 = 8$ neighboring cells

In 10 dim, each cell has $3^{10} - 1 \approx 59,000$ neighbors

Need to decide how many apples make an orange

Consider the case $p = 2$. Assume X_1 is measured in gallons and X_2 is measured in pounds

Define distance in predictor space by

$$d^{\infty}(\underline{x}_1, \underline{x}_2; \underline{w}) = \max_i w_i |x_{1i} - x_{2i}|$$

The ratio w_1 / w_2 defines how many gallons make a pound when we measure distances.

Suppose we have TS $(\underline{x}_1, y_1) \dots (\underline{x}_n, y_n)$

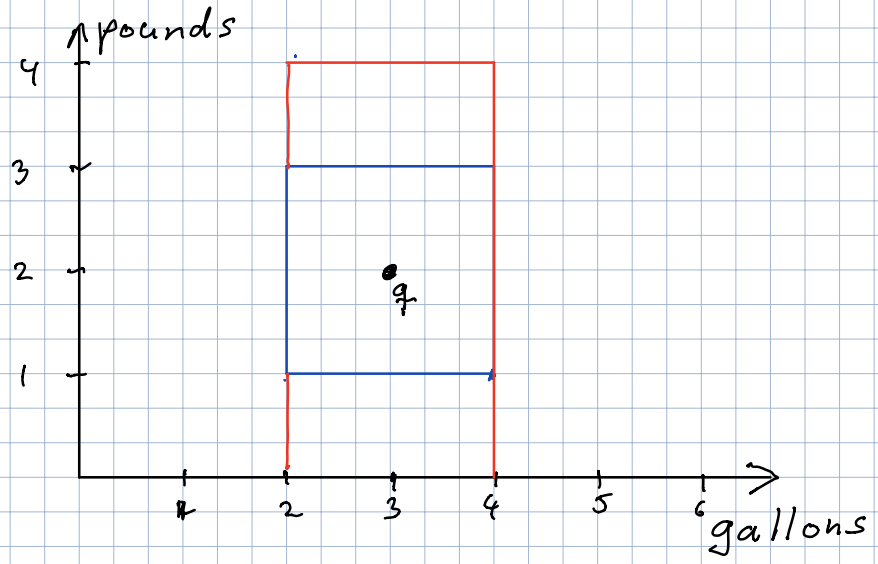
and we estimate $f(q) = E(Y | \underline{X} = q)$ by local averaging:

$$\hat{f}(q) = \text{mean}(y_i \mid d^{\infty}(q, \underline{x}_i; \underline{w}) \le c)$$

We are smoothing with a box kernel.

The width of the box along dimension i is $\ell_i = 2c / w_i$.

Here is a picture for $c=1$, $w_1 = w_2 = 1$



box for $c=1$, $w_1 = w_2 = 1$

box for $c=1$, $w_1 = 1$, $w_2 = 0.5$

Good choice of w_1, w_2 depends on nature of response variation

Consider 2 scenarios

(1) $E(y \mid \underline{x}) = f(x_1)$

The optimal choice is $w_2 = 0$.

Stretching the box in the x_2 direction

will not increase the bias of $\hat{f}(q)$ because f does not depend on x_2.

It will decrease the variance because we average over more training observations.

(2) $E(Y|\underline{x}) = f(x_2)$

The optimal choice is $w_1 = 0$

Same argument as above.

Message: Optimal weights for predictor variables depend on $E(Y|\underline{x})$ which is unknown.

Basic idea of CART

Estimate $E(Y|\underline{x})$ by local averaging, but choose neighborhoods over which to average by looking at response values for training sample.

CART

Given TS $(\underline{x}_1, y_1) \dots (\underline{x}_n, y_n)$ $\underline{x}_i \in \mathbb{R}^p$

$y_i \in \mathbb{R}$

Goal: Estimate $E(Y|\underline{x})$

Idea:

- Estimate $E(Y|\underline{x})$ by local averaging
- Choose neighborhood adaptively by looking at the observed responses

Form of model

$$\hat{f}(\underline{x}) = \sum_{i=1}^{K} c_i \, I(\underline{x} \in N_i)$$

constant over disjoint hyper-rectangles $N_1 \dots N_K$ with $\bigcup_i N_i = \mathbb{R}^p$

$$RSS = \sum_{j=1}^{n} \left(y_j - \sum_{i=1}^{K} c_i \, I(\underline{x}_j \in N_i)\right)^2$$

$$= \sum_{m=1}^{K} \left(\sum_{j | x_j \in N_m} \left(y_j - \sum_{i=1}^{K} c_i \, I(x_j \in N_i)\right)^2 \right)$$

$\Rightarrow$ $c_m =$ mean$(y_j \mid \underline{x}_j \in N_m)$

This choice minimizes the resubstitution error.

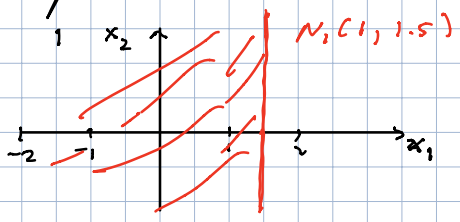
Key question: How to determine K and

$R_1 \dots R_K$

Special case $K = 2$

Split of $\mathbb{R}^p$ into two disjoint axis parallel hyper-rectangles is defined by a split coordinate i and a split point s.

$$R_1(i,s) = \{x \mid x_i \le s\}$$
$$R_2(i,s) = \{x \mid x_i > s\}$$



(i,s) also defines a split of the training sample into subsets

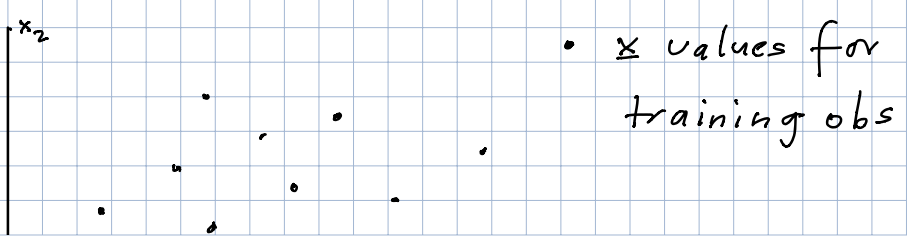
$$S_1(i,s) = \{j \mid x_j \in R_1(i,s)\}$$
$$S_2(i,s) = \{j \mid x_j \in R_2(i,s)\}$$

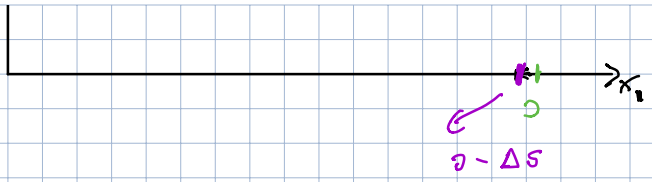
$$c_1(i,s) = \text{mean}(y_j \mid j \in S_1(i,s))$$
$$c_2(i,s) = \text{mean}(y_j \mid j \in S_2(i,s))$$

$$RSS(i,s) = (n_1 - 1)\,\text{var}(y_j \mid j \in S_1(i,s)) + (n_2 - 1)\,\text{var}(y_j \mid j \in S_2(i,s))$$

How to find optimal i, s?

Note: RSS only changes when S_1 and S_2 change





For each feature we only have to try $(n-1)$ split points

Moreover, means and variances can be updated.

We can find optimal split in $O(np)$ operations (p = # of predictor variables) using exhaustive search.