K-NN Regression

predicting continuous outcomes Y

K nearest neighbors

Setup: training data $\{(y_n, x_n)\}$ $y_n \in \mathbb{R}$, $x_n \in \mathbb{R}^p$

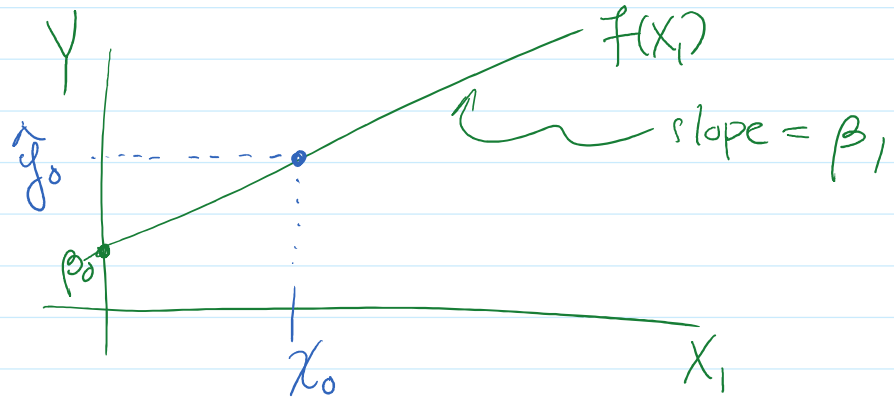
we assume that

$$Y \approx f(X_1, X_2, \ldots, X_p)$$

Linear Regression: makes the assumption about f that it is linear

$$Y = f(X_1, \ldots, X_p) = \beta_0 + \sum_{j=1}^{N} \beta_j X_j$$

$p = 1$



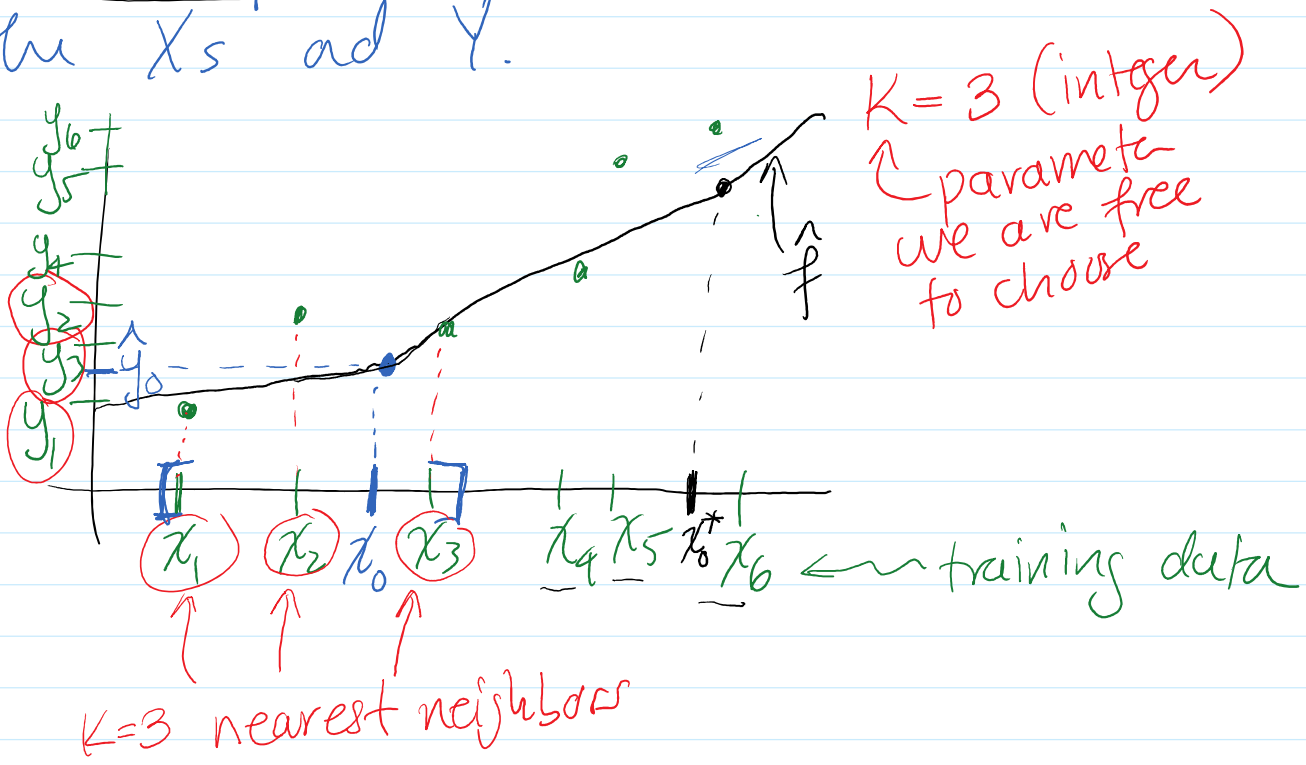
this is a strong global assumption about f, because it says f is linear w/ slope β_1 everywhere

What this does for us is reduce dim of

What this does for us is reduce dim of problem from

$$|\mathcal{F}| = \infty \quad \text{to} \quad |\mathcal{F}| = p+1$$

K-NN regression: make a weaker local assumption abat the rel. between the Xs and Y.



Math:

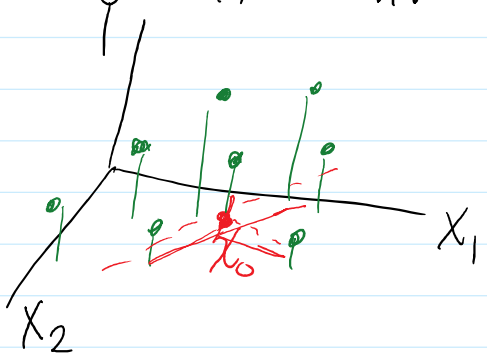
$$\hat{y} = \hat{f}(x) = \frac{1}{K} \sum y_i$$

K nearest neighbors to x

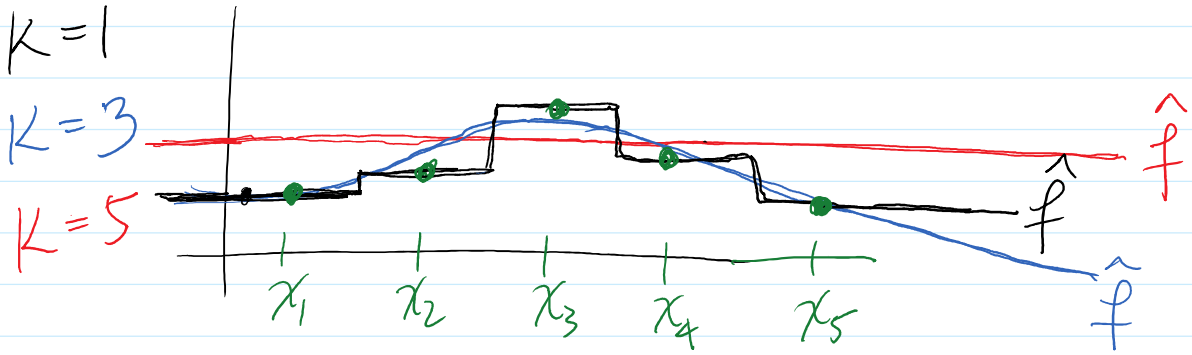
$\{i \mid x_i \in N_K(x)\}$

We can do this for $x_n \in \mathbb{R}^p$ we just use euclidean distance to determine "closeness"

$\|x - x_n\|$



K-NN, what about K?



General Rule: K controls flexibility of the method.

flexibility → how complicated of a shape it can create.

High K → very inflexible method

Low K → very flexible method

Flexibility compared to linear regression

parameters fit

Flexibility compared to linear regression

parameters fit

(1) Linear Regression: p = # cols in design mtx

(2) KNN: $\sim N/K$ as $K\uparrow$ flex. $\downarrow$, $K\downarrow$ flex $\uparrow$

(properly VC-dimension is N/K)

Evaluation: How do I choose a method or parameters?

Residual: $e = \hat{y} - y$ = predicted − actual

want in a method fit $\hat{f}$ is that

$$\hat{y} = \hat{f}(x) \approx y$$

i.e. small residuals.

Setup: training data $\{(y_n, x_n)\}_{n=1}^{N}$ and we fit some $\hat{f}$ so we can summarize the residuals in several ways:

(1) training residual sum of squares (RSS_{train})

$$RSS_{train} = \sum_{n=1}^{N} e_n^2 = \sum_{n=1}^{N} (\hat{y}_n - y_n)^2$$

$$RSS_{train} = \sum_{n=1} e_n^2 = \sum_{n=1} (\hat{y}_n - y_n)^2$$

(2) training mean squared error: (MSE_{train})

$$MSE_{train} = \frac{1}{N} \sum_{n=1}^{N} (\hat{y}_n - y)^2 = \frac{1}{N} RSS$$

(3) training Coefficient of Determination (R^2_{train})

$$R^2_{train} = 1 - \frac{RSS_{train}}{TSS_{train}}$$

↑ total sum of squares

$$TSS_{train} = \sum_{n=1}^{N} (y_n - \bar{y})^2$$

RSS = how much variation is left in data after $\hat{f}$ predicts

TSS = total amt. of variation.

$$R^2 = 1 - \frac{RSS}{TSS}$$

% of total variation remain after acct for $\hat{f}$

% of variation explained by $\hat{f}$.