Recall, we assume for regression

$$y = f(x) + \varepsilon$$

want to determine $\hat{f}$ so that $\hat{y} = \hat{f}(x) \approx y$.

Soln: minimize expected loss (Risk)

$$\hat{f} = \arg\min_f E[(y - f(x))^2]$$

claim: $\hat{f}(x) = E[y|x]$

So we model $E[y|x]$.

Ways to do this:

global assumption about form of $E[y|x]$

1. Linear Regression

$$\hat{f}(x) = x^T \hat{\beta}$$

2. Polynomial Regression $x \in \mathbb{R}$ $(p = 1)$

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3 + \cdots$$

local assumption

3. relax this global (linear) assumption

local assump

(b) relax this global (linear) assumption
e.g. KNN regression.

→ non-parametric regression

model: $\mathbb{E}[y|x] = m(x)$

some function w/ no particular predefined form!

KNN Regression

$$\hat{f}(x) = \frac{1}{K} \sum_{n: x_n \in N_K(x)} y_n$$

if we let

$$\omega_n(x) = \frac{1}{K} \mathbb{1}(x_n \in N_K(x)) = \begin{cases} 1/K, & x_n \in N_K(x) \\ 0, & \text{else} \end{cases}$$

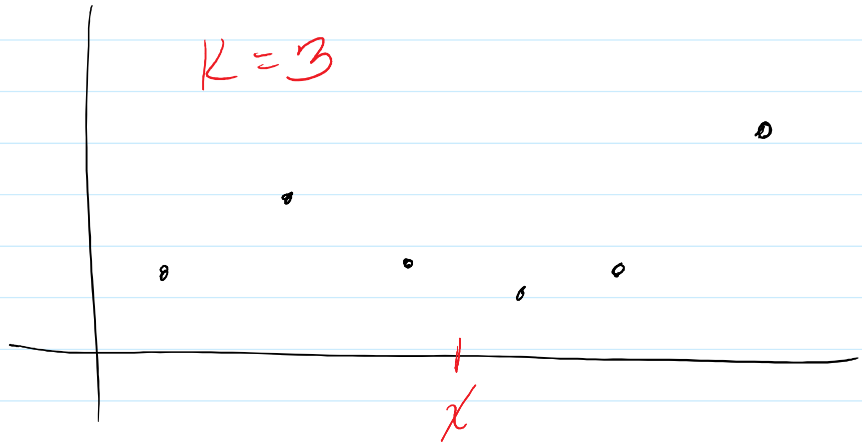
then for KNN

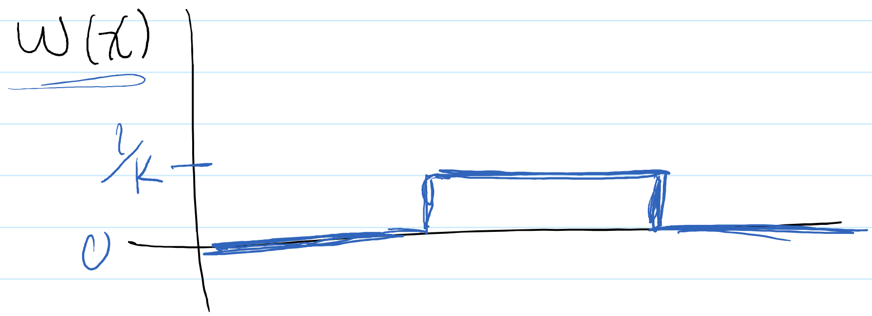
$$\boxed{\hat{f}(x) = \sum_{n=1}^{N} \omega_n(x) y_n .}$$

weighted average of training y_n's

weight for n^{th} training pair

weight for n^{th} training pair (depends on x)





KNN may be called a linear smoother

<u>Notice:</u> w_ns are discontinuous.

Generalization is to use "nicer" weighting function w_n

continuous differentiable

leads to <u>Kernel Smoothing</u>.

Defn: a Kernel is a function K

(1) $K: \mathbb{R}^P \to \mathbb{R}$

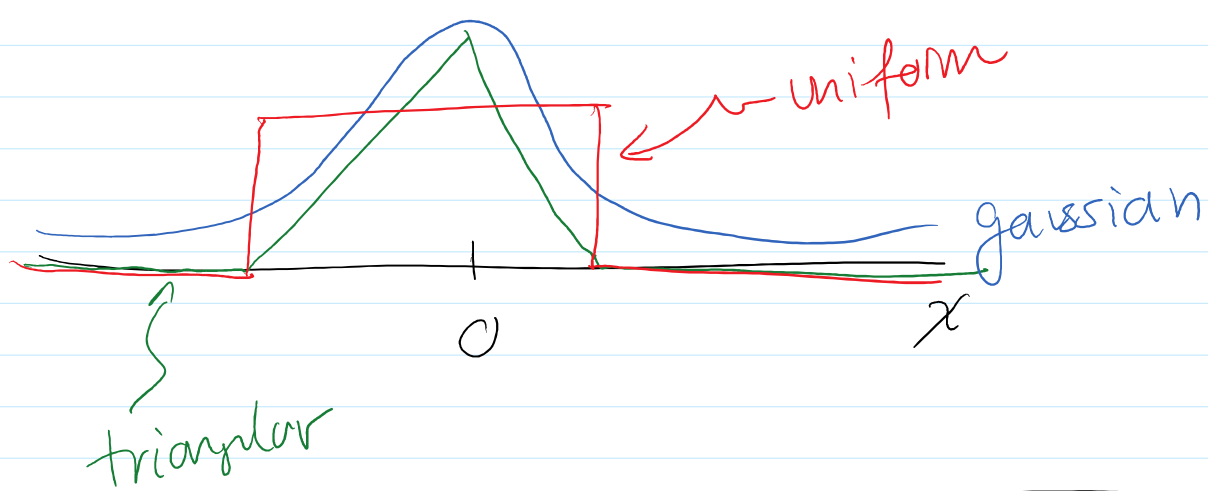
(2) $K(x) \geq 0$

typically also assume

(3) $\int_{\mathbb{R}^P} K(x)\,dx = 1$ (PDF of a P-variate RV)

(4) $K(x) = K(-x)$ (symmetric)

Ex.



We can use kernels to define a measure of "closeness":

$K(x-y) \approx$ closeness between x and y

$K(x-y) \approx$ closeness between x and y

$p=1$ and $K(\cdot) = \frac{1}{|\cdot|}$ then $K(x-y) = \frac{1}{|x-y|}$.

Ex. $K(\cdot) = \exp(-\cdot^2)$

$$K(x-y) = \exp\left(-(x-y)^2\right)$$

linear smoother: Kernel Smoothing

$$\hat{f}(x) = \sum_{n=1}^{N} w_n(x) y_n$$

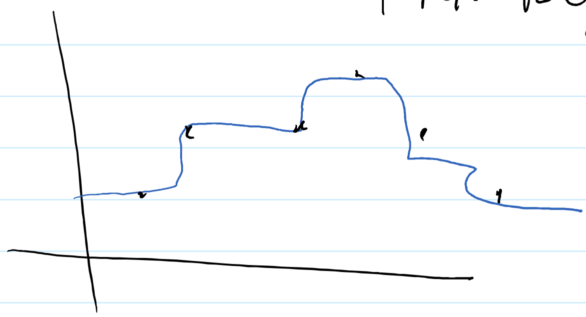
where $w_n(x) \propto \underbrace{K(x-x_n)}_{\text{closeness of } x_n \text{ to } x}$

Typically req that
$$\sum_{n=1}^{N} w_n(x) = 1$$

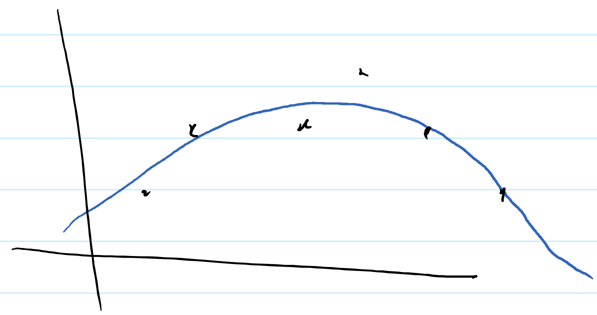
i.e.
$$\boxed{w_n(x) = \frac{K(x-x_n)}{\sum_{i=1}^{N} K(x_n - x_i)}}.$$

$\hookrightarrow$ Nadaraya-Watson (NW) Kernel Regression estimator.

KNN Reg



Kernel Reg



Generalize a Bit more

For weights ω_n then the weighted avg of y_ns

$$\rightarrow \bar{y}_w = \frac{1}{\sum_n \omega_n} \sum_{n=1}^{N} \omega_n y_n$$

if $\omega_n = 1/N \ \forall n$ we get typical avg.

Model : $\boxed{Y \approx \alpha}$

$$\hat{\alpha} = \arg\min_{\alpha} \sum_n (y_n - \alpha)^2 = \bar{y}$$

$$\rightarrow \hat{\alpha} = \arg\min_{\alpha} \sum_n \omega_n (y_n - \alpha)^2 = \bar{y}_w$$

Model: $\boxed{Y \approx \alpha + \beta x}$

$$\hat{\alpha}, \hat{\beta} = \arg\min_{\alpha,\beta} \sum_n (y_n - \alpha - \beta x_n)^2 = \text{Least squares reg. ests}$$

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha,\beta}{\arg\min} \sum_n w_n (y_n - \alpha - \beta x_n)^2 = \text{weighted LS reg. ests.}$$

can depend on x_ns

Model: $Y = X\beta$

$$\hat{\beta}^{(OLS)} = \underset{\beta}{\arg\min} \|Y - X\beta\|^2 = (X^T X)^{-1} X^T Y$$

diag(w_ns)

$$\hat{\beta}^{(WLS)} = \underset{\beta}{\arg\min} \|W(Y - X\beta)\|^2 = (X^T W X)^{-1} X^T W Y$$

Punchline: $x \in \mathbb{R}$ $(P=1)$

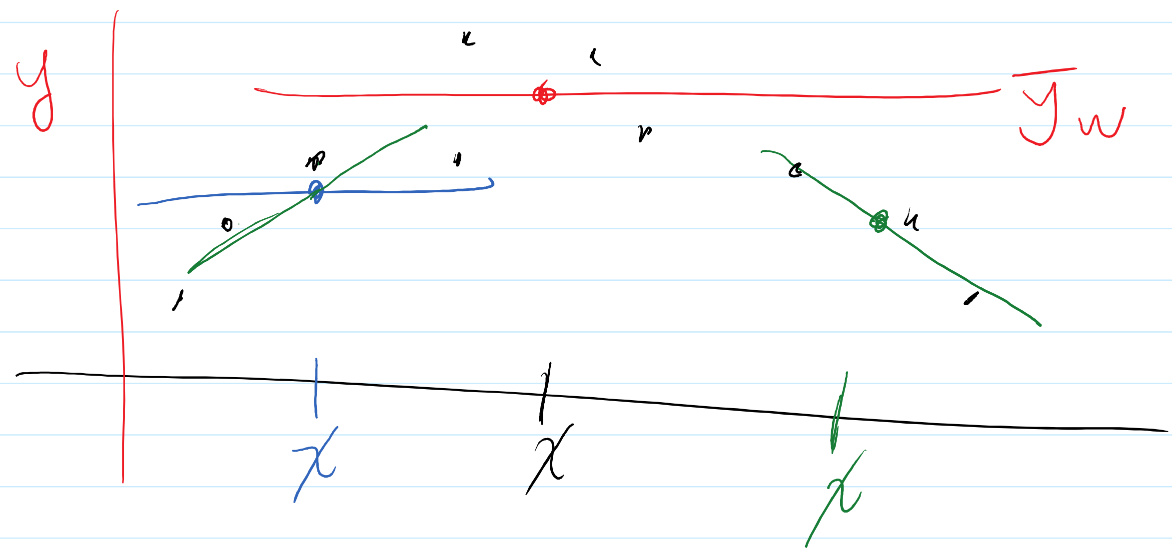
$$\hat{f}(x) = \underset{\theta}{\arg\min} \sum_{n=1}^{N} \underbrace{K(x - x_n)}_{w_n} (y_n - \theta)^2$$

then

$$\hat{f}(x) = \bar{y}_w \quad w/ \; w_n \propto K(x - x_n)$$

$= $ NW Kernel Regression est. w/ $w_n(x) \propto K(x - x_n)$.

NW K-Reg. est. solves a simple weighted regression model where weights depend on x so that $w_n(x) \propto K(x - x_n)$



Local Polynomial Regression

Simple example

$$\boxed{\hat{f}(x) = \hat{\alpha}(x) + \hat{\beta}(x)\,x}$$

$$\hat{\alpha}(x), \hat{\beta}(x) = \underset{\alpha,\beta}{\operatorname{argmin}} \sum_{n=1}^{N} \overbrace{K(x-x_n)}^{w_n(x)}\left(y_n - \alpha - \beta x_n\right)^2$$

weighted regression ests
w/ $w_n(x) \propto K(x-x_n)$

Fully $P=1$ Local Poly

$$\{\hat{\beta}_j(x)\}_j = \underset{\{\beta_j\}}{\operatorname{argmin}} \sum_{n=1}^{N} K(x-x_n)\left(y_n - \sum_{j=0}^{d} \beta_j x_n^j\right)$$

weighted poly regression

Ultimately: at some x

$$W = \mathrm{diag}(w_n(x)) = \mathrm{diag}(K(x - x_n))$$

$$X = \begin{bmatrix} | & | & | & | & \\ 1 & x_n & x_n^2 & x_n^3 & \cdots \\ | & | & | & | & \end{bmatrix}$$

and $\hat{\beta} = (X^T W X)^{-1} X^T W y$

and so

$$\hat{y} = \hat{f}(x) = x^T \hat{\beta}$$

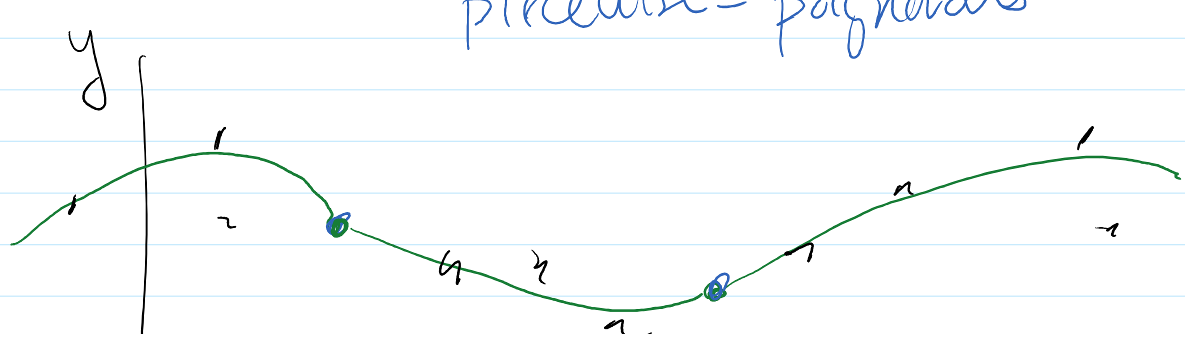
linear smoother.

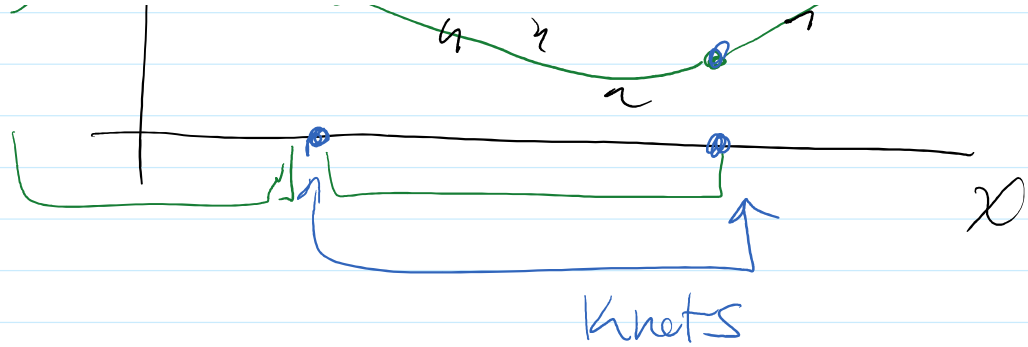
Splines

Kernel Reg/Local Poly: local methods

Splines: Fit a better global model.

piecewise-polynomials





Spline: k-degree polynomial w/ some smoothness condition at break-points