Lecture 4

Bias and Variance (finish)

Losses

Linear Regression by LS

HW1 due Next Wed

[HW2 NOT GRADED]

L11 linear predictors

Prob/Stat refresher

- Fri 9:30, 10:30 Gavin

MC 2035

Predictors

- K-Nearest-Neighbor
- Linear - for regression
  - for classification

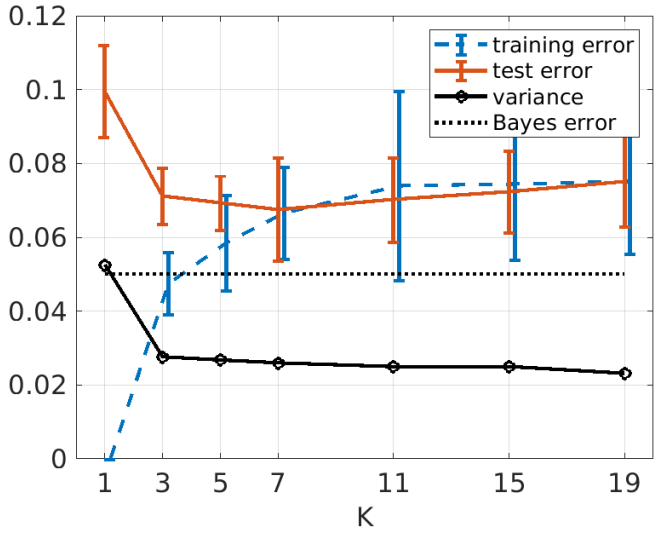
Algorithms

LS Regression

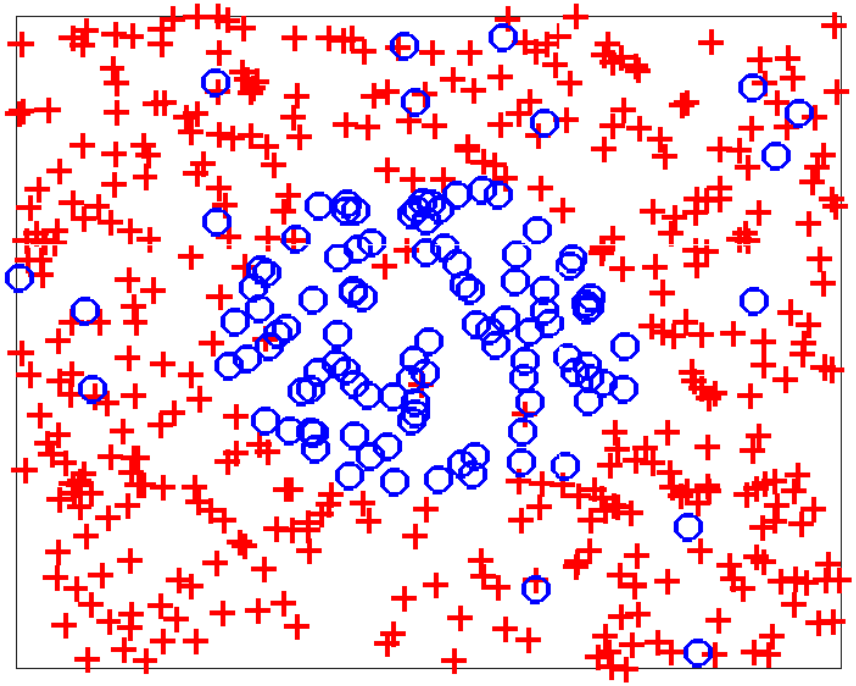
Concepts

- Decision Region, Dec. Boundary
- Training error, Test error
  - Expected error
- Variance, Bias
- Loss functions - training / test / expected loss

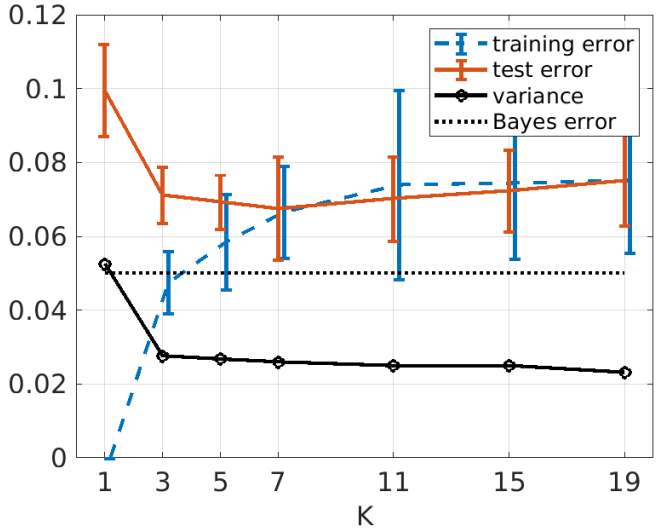
The case $K = 1$: Variance

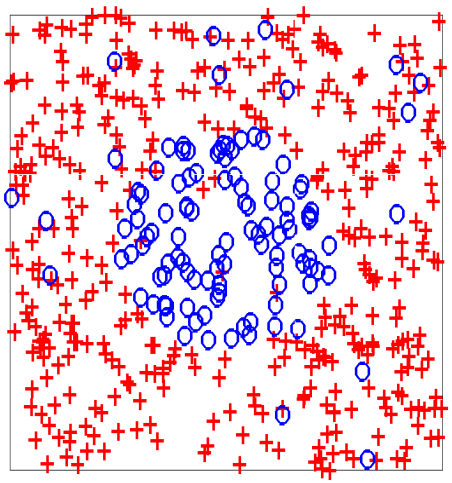


- $\mathcal{D} \sim P_{XY} \Rightarrow \mathcal{D}$ is **random**



The case $K = 1$: Variance





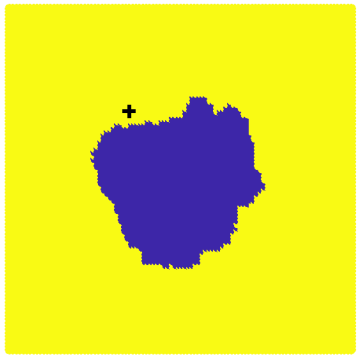
$(K = 1)$

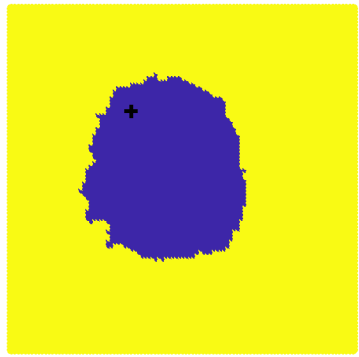
- $\mathcal{D} \sim P_{XY} \Rightarrow \mathcal{D}$ is **random**
- Hence any function f_K we estimate from $\mathcal{D}$ is also random

- Formally, for any fixed x, $f_K(x)$ is a random variable, hence it has a variance.
- In this course, we do not explicitly calculate the variance, but we want to know what increases or decreases it.

The case of K large: Bias

($K = 11$)





- **Bias** means to let one's own prior beliefs override the evidence.
- In data science/ML/statistics **every model/prediction** is a combination of **prior belief** and data

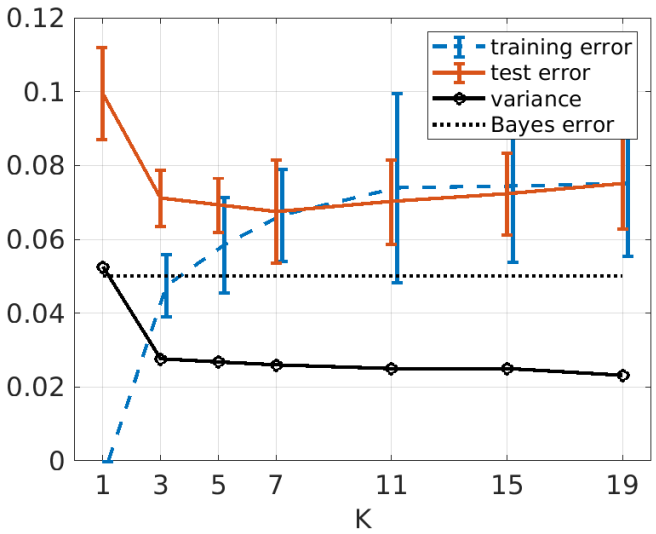
- **prior** = before seeing the data
- (usually) **prior belief** = prior **knowledge**, e.g. from previous experiments

- Bias can take many forms – in this course you will encounter several
- We do not explicitly calculate bias, but we want to identify where it is coming from, and what increases/decreases it
- One way to look for bias: if a predictor f cannot exactly/accurately predict a training set, "whatever is causing this" is bias.

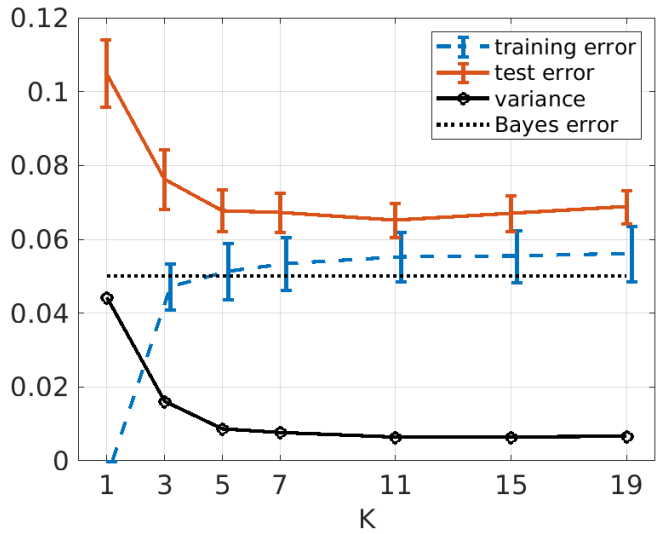
The Bias-Variance trade-off

- When bias ↗, variance ↘

- When data set size n ↗, variance ↘ ⇒ bias ↘



$n = 2000$



$n = 2000$

Lecture II: Linear regression and classification. Loss functions

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

January 12, 2026

Linear predictors generalities

Loss functions

Least squares linear regression
- Linear regresssion as minimizing L_{LS}
- Linear regresssion as maximizing likelihood
- Linear Discriminant Analysis (LDA)
- QDA (Quadratic Discriminant Analysis)
- Logistic Regression
- The Perceptron algorithm

Reading HTF Ch.: 2.1–5,2.9, 7.1–4 bias-variance tradeoff, Murphy Ch.: 1., 8.6[1], Bach Ch.:

[1] Neither textbook is close to these notes except in a few places; take them as alternative perspectives or related reading

Linear predictors

regr. / classifiers

$\beta, x \in \mathbb{R}^d$

- Linear predictors for regression

$$f(x) = \beta^T x \qquad \hat{y} = f(x) \tag{1}$$

where $Y \in \mathbb{R}$, $X \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$ is a **vector of parameters**.
Hence, the model class is $\mathcal{F} = \{\beta \in \mathbb{R}^d\}$ the set of all linear functions over $\mathbb{R}^d$.

- Linear predictors for classification

$$\hat{y}(x) = \text{sgn}(\beta^T x) \leftarrow \text{for example} \tag{2}$$

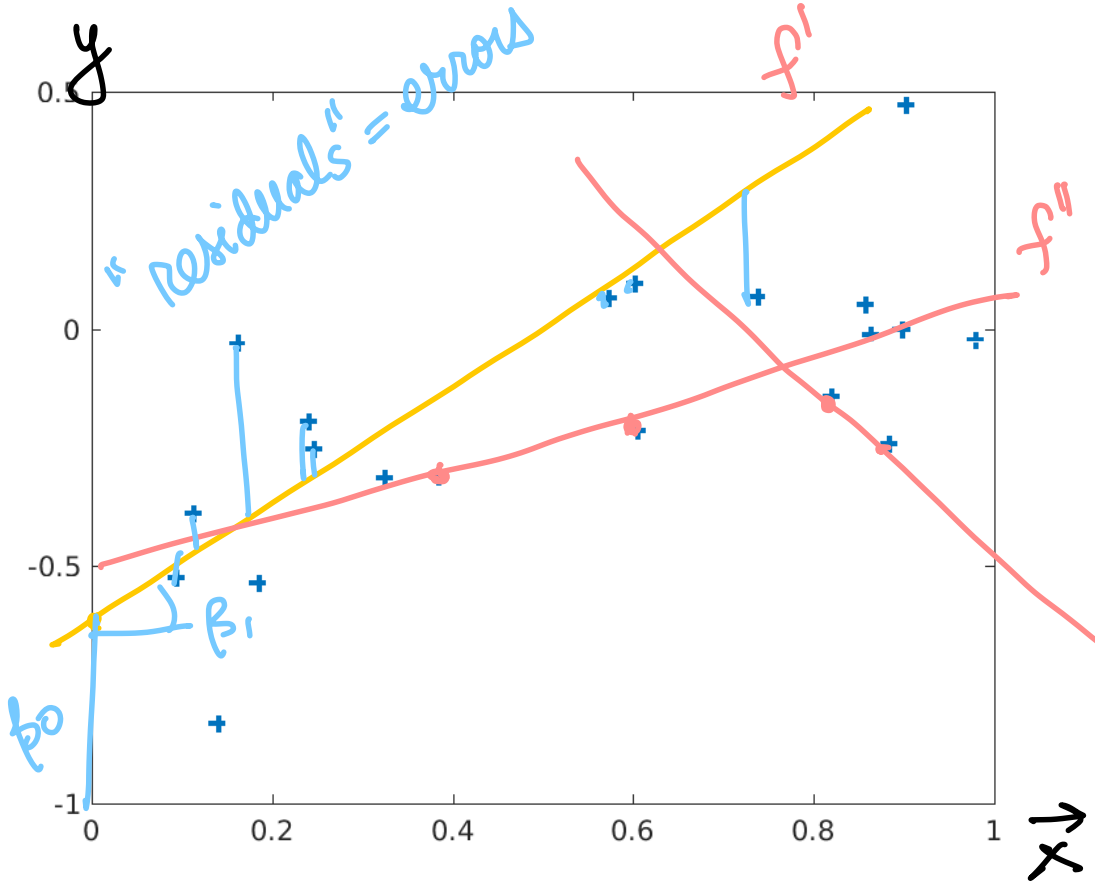
i.e. the decision boundary is linear

$$\text{sgn}\ \beta^T x = \text{sgn}\left(e^{\beta^T x} - 1\right)$$

$$= \text{sgn}\ g(\beta^T x)$$

$g(0) = 0$

$g \uparrow$



$d=1$

$$f(x) = \beta_1 x + \beta_0 = \tilde{\beta}^T \tilde{x}$$

$$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} \} d \quad \in \mathbb{R}^{d+1}$$

$$\tilde{\beta} = \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} \} d \quad \in \mathbb{R}^{d+1}$$

"Loss" = "avg" of errors

Transforming categorical inputs into real values

- if X_j takes two values (e.g "yes", "no"), map it to $\{\pm 1\}$ or $\{0, 1\} \subset \mathbb{R}$.
- discrete multivariate inputs
  - Let X_j take values in $\Omega_j = \{0, \ldots r-1\}$.
  - One defines the $r-1$ binary variables $\tilde{X}_{jk} = \mathbf{1}_{[]} X_j = k,\ k = 1 : r-1$. The variable X_j is replaced with $\tilde{X}_{jk},\ k = 1 : r-1$
  - the parameter β_j with $r-1$ parameters $\beta_{j1} \ldots \beta j, r-1$ representing the coefficients of $\tilde{X}_{j1}, \ldots \tilde{X}_{j,r-1}$.

  This substitution is widely used to parametrize any function of a discrete variable as a linear function

Example: The demographic variable Race takes values in { African, Asian, Caucasian, ...}; the corresponding parameters in the model will be $\hat{\beta}_{\text{Asian}}, \hat{\beta}_{\text{Caucasian}}, \ldots$.

The intercept as a slope

- Sometimes we like f to have an intercept $f(x) = \beta^T x + \beta_0$, with $x, \beta \in \mathbb{R}^d$. Such a function is affine, not linear, and not homogeneous. Here is a trick to get the best of both worlds.
- Add a dummy input $x_0 \equiv 1$ to x. Then its coefficient β_0 is the intercept.

$$\tilde{x} \leftarrow \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_d \end{bmatrix} \in \mathbb{R}^{d+1} \quad \tilde{\beta} \leftarrow \begin{bmatrix} \beta_0 \\ \beta_1 \\ \cdots \\ \beta_d \end{bmatrix} \in \mathbb{R}^{d+1} \quad f(x) = \tilde{\beta}^T \tilde{x} \tag{3}$$

- in classification, β_0 is called **threshold** or **bias term**

How good is a regressor? Measuring the "Error"

- Prediction error for y^i: $e^i = y^i - f(x^i)$
- "Error" of f on $\mathcal{D}$
  - [redacted] ✗
  - $''Err'' = \frac{1}{n}\sum_{i=1}^{n} |e^i|$? ← $L_1 = \|e\|_1 \cdot \frac{1}{n}$
  - ... norms!
- Let $e = [e^1\ e^2\ \dots\ e^n]$.
- e is a vector in $\mathbb{R}^n$. $\frac{1}{n}\sum_{i=1}^{n} |e^i| = \frac{1}{n}\|e\|_1$
- But we can use other norms, e.g. $\frac{1}{n}\|e\|_2$, $\frac{1}{n}\|e\|_\infty$.

- Formally, $''Err''$ as above is called **loss** function.

$e = \begin{bmatrix} e^1 \\ e^2 \\ \vdots \end{bmatrix} \} n$

$L_2^{train} = \frac{1}{n}\|e\|_2^2$ Mean Squared Error / Least Squares Loss

$L_\infty^{train} = \frac{1}{n} \max_{i=1:n} |e^i|$

Loss functions

The **loss function** represents the cost of error in a prediction problem. We denote it by L, where

$$L(y, \hat{y}) = \text{the cost of predicting } \hat{y} \text{ when the actual outcome is } y$$

true predicted

As usually $\hat{y} = f(x)$ or $\text{sgn} f(x)$, we will typically abuse notation and write $L(y, f(x))$.

$L(e) \rightarrow L(y, \hat{y})$

$e = y - \hat{y}$

Loss functions

The **loss function** represents the cost of error in a prediction problem. We denote it by L, where

$$L(y, \hat{y}) = \text{the cost of predicting } \hat{y} \text{ when the actual outcome is } y$$

As usually $\hat{y} = f(x)$ or $\text{sgn} f(x)$, we will typically abuse notation and write $L(y, f(x))$.

- For **Regression**
  - **Least-Squares** L_2 Loss $L_{LS}(y, f(x)) = \frac{1}{n}\|e\|_2^2$
  - L_1 Loss $L_{LS}(y, f(x)) = \frac{1}{n}\|e\|_1$
  - Statistical losses.
- For **Classification**
  - **Misclassification Error (0-1 Loss)** $L_{01} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{[y^i \neq \hat{y}^i]}$
  - Statistical losses...
  - Imbalanced losses

Loss functions for classification

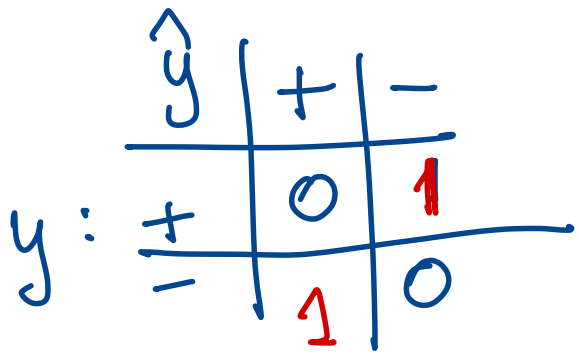
For classification, a natural loss function is the **misclassification error** (also called **0-1 loss**)

$$L_{01}(y, f(x)) = \mathbf{1}_{[y \neq f(x)]} = \begin{cases} 1 & \text{if } y \neq f(x) \\ 0 & \text{if } y = f(x) \end{cases} \tag{5}$$

Sometimes different errors have different costs. For instance, classifying a HIV+ patient as negative (**a false negative** error) incurs a much higher cost than classifying a normal patient as HIV+ (**false positive** error). This is expressed by **asymmetric misclassification costs**. For instance, assume that a false positive has cost one and a false negative has cost 100. We can express this in the matrix

$f(x):$	$+$	$-$
true $:+$	0	100
$-$	1	0

In general, when there are p classes, the matrix $L = [L_{kl}]$ defines the loss, with L_{kl} being the cost of misclassifying as l an example whose true class is k.



$$L^{train} = \frac{1}{n} \sum_{i=1}^{n} L_{y^i \hat{y}^i} =$$

$$= \frac{1}{n} \{ \#(\hat{y}^i = 1, y^i = 0) \cdot L_{01} + \#(\hat{y}^i = 0, y^i = 1) \cdot L_{10} \}$$

Training set loss and expected loss

- **Training set loss**
- Objective of prediction = to minimize loss on future data,

$$\text{minimize } L(f) = E_{P(X,Y)}[L(Y, f(X)] \text{ over } f \in \mathcal{F} \tag{6}$$

We call $L(f)$ above **expected loss**.

Example (Misclassification error $L_{01}(f)$)

$L_{01}(f)$ = probability of making an error on future data.

$$L_{01}(f) = P[Yf(X) < 0] = E_{P_{XY}}[\mathbf{1}_{[Yf(X)<0]}] \tag{7}$$

Training set loss and expected loss

- **Training set loss**
- Objective of prediction = to minimize loss on future data,

$$\text{minimize } L(f) = E_{P(X,Y)}[L(Y, f(X)] \text{ over } f \in \mathcal{F} \quad (6)$$

We call $L(f)$ above **expected loss**.

- $L(f)$ cannot be minimized or even computed directly, because we don't know the data distribution P_{XY}.
- Therefore, in training we use the **traing set** loss.
- . . . we approximate data distribution P_{XY} by the sample $\mathcal{D}$.
- The empirical loss (or **empirical error** or **training error**) is the average loss on $\mathcal{D}$

$$\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} 1_{[y^i f(x^i)<0]} \quad (7)$$

Training set loss and expected loss

- **Training set loss**
- Objective of prediction = to minimize loss on future data,

$$\text{minimize } L(f) = E_{P(X,Y)}[L(Y, f(X)] \text{ over } f \in \mathcal{F} \tag{6}$$

  We call $L(f)$ above **expected loss**.
- Therefore, in training we use the **traing set** loss.
- . . . we approximate data distribution P_{XY} by the sample $\mathcal{D}$.
- The empirical loss (or **empirical error** or **training error**) is the average loss on $\mathcal{D}$

$$\hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} 1_{[y^i f(x^i) < 0]} \tag{7}$$

- And we approximate $L(f)$ the expected loss by a different data set $\mathcal{D}^{\text{test}}$ from the same P_{XY}.
- The size of $\mathcal{D}^{\text{test}}$ is n', not necessarily equal to n.

(Linear) least squares regression

- define **data matrix** or (transpose) **design matrix**

$$\mathbf{X} = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \cdots \\ (x^i)^T \\ \cdots \\ (x^n)^T \end{bmatrix} \in \mathbb{R}^{N \times n} \quad \text{and} \quad Y = \begin{bmatrix} y^1 \\ y^2 \\ \cdots \\ y^n \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon^1 \\ \varepsilon^2 \\ \cdots \\ \varepsilon^d \end{bmatrix} \in \mathbb{R}^d$$

- Then we can write

$$Y = X\beta + E$$

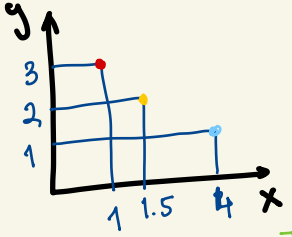
- The solution $\hat{\beta}$ is chosen to minimize the sum of the squared errors $\sum_{i=1}^{d}(\varepsilon^i)^2 = \sum_{i=1}^{d}(y^i - \beta^T x^i)^2 = ||E||^2$

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{d}(y^i - \beta^T x_i)^2$$

- This optimization problem is called a **least squares** problem. Its solution is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y \tag{8}$$

- Underlying statistical model $y = \beta^T x + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$ (and i.i.d. sampling of $(x^{1:N}, y^{1:N})$ of course).
  Then $\hat{\beta}$ from (8) is the Maximum Likelihood (ML) estimator of the parameter β.



$d=1$

$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} \quad \tilde{\beta} = \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$

$\tilde{x}^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \tilde{x}^2 = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} \quad \tilde{x}^3 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$

$X = \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \\ 4 & 1 \end{bmatrix} \quad (\tilde{x}^1)^T$

$\hat{y} = f(\tilde{x}) = \tilde{x}^T \tilde{\beta} = \beta_1 x + \beta_0$

prediction

$\hat{y} = \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \hat{y}^3 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$

$$e = y - \hat{y} = y - X\beta$$

1. $e = y - X\beta$

2. Loss in matrix vector form

$$n \cdot L_2^{\text{train}}(\beta) = \|e\|_2^2 = (y - X\beta)^T (y - X\beta)$$
$$= y^T y - \beta^T X^T y - y^T X \beta + \beta^T X^T X \beta$$

$y^T X \beta = a^T$ (symmetric)

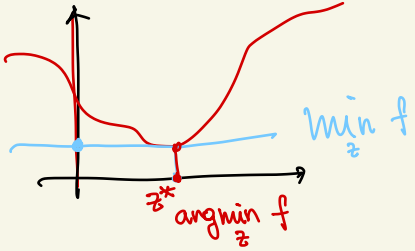
3. Find $\hat{\beta} = \arg\min_\beta L_2^{\text{train}}(\beta)$

by solving $\nabla L_2^{\text{train}}(\beta) = 0$

3.1.
$$\nabla L_2^{\text{train}}(\beta) = 0 - 2X^T y + 2X^T X\beta = 0$$

3.2 solve linear system
$$(X^T X)\beta = X^T y$$
A $\quad b$

$A = d \times d$
$\beta \in \mathbb{R}^d$
$b = \mathbb{R}^d$



$\|e\|_2^2 = e^T e$

$(X\beta)^T = \beta^T X^T$

$g(z) = a^T z \in \mathbb{R}$

$\nabla g = a$ quadratic

$h(z) = z^T A z$

$\nabla h = 2Az$

A symmetric

$Az = b$