CS 480/680 1/15/26

Lecture 4

Losses

Linear Regression by LS

HW1 due Next Wed

HW2 NOT GRADED

LII 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

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
- Support Vector Machines

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

regression
classification

- Linear predictors for regression

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

$$f(x) = \beta^T x \qquad f(x) \in \mathbb{R} \tag{1}$$

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

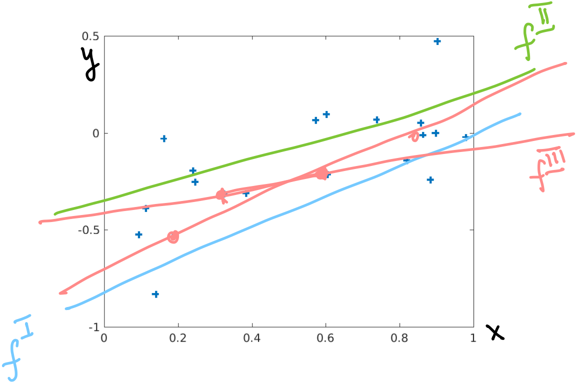
$$\text{e.g. } \hat{y}(x) = \mathrm{sgn}(\beta^T x) \qquad \hat{y}, y \in \{\pm 1\} \tag{2}$$

i.e. the decision boundary is linear

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

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

monotonically increasing



$d = 1$

$x \in \mathbb{R}$

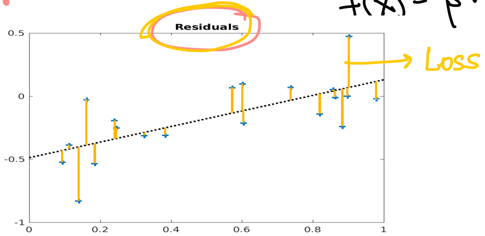
$f(x) = \beta_0 + \beta_1 x$

intercept

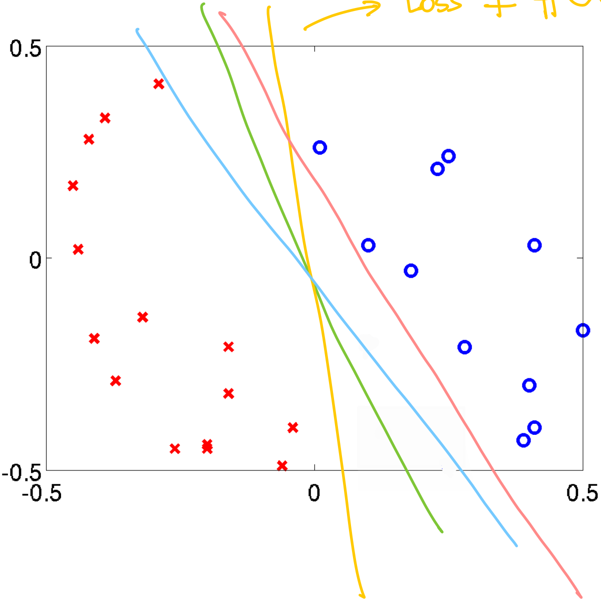
$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix} \in \mathbb{R}^2$

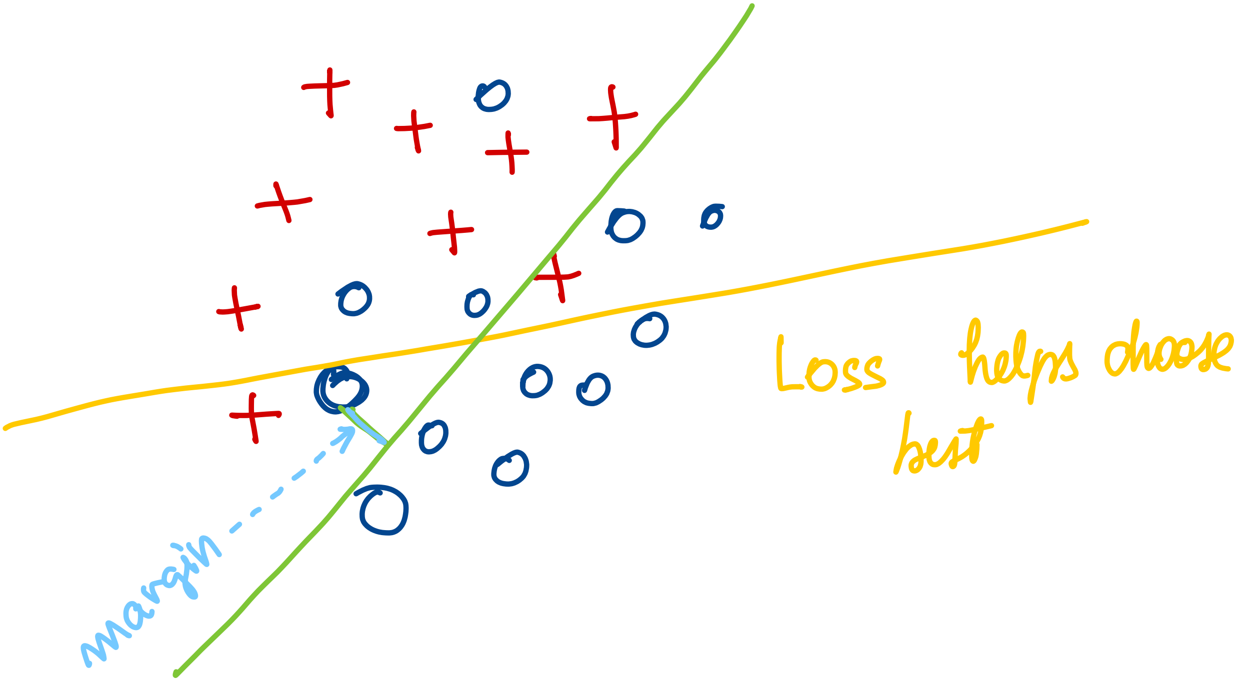
$\beta = \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} \in \mathbb{R}^2$

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



want
Loss ≠ #errors





margin
Loss helps choose best

How good is a regressor? Measuring the "Error"

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

$\frac{1}{n}\|e\|_2$

Euclidean norm, 2-Norm $\Longleftrightarrow$ Mean Squared Error

$\frac{1}{n}\|e\|_2^2$

How good is a regressor? Measuring the "Error"

- Prediction error for y^i: $e^i = y^i - f(x^i)$
- "Error" of f on $\mathcal{D}$
  - $''Err'' = \frac{1}{n}\sum_{i=1}^{n} e^i$ **X**
  - $''Err'' = \frac{1}{n}\sum_{i=1}^{n} |e^i|$?
  - . . . norms!
- Let $e = [e^1\ e^2\ \ldots\ e^n]$.
- 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.

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 $\operatorname{sgn} f(x)$, we will typically abuse notation and write $L(y, f(x))$.

$\frac{1}{n}\|e\|_1 \dashrightarrow L_1 = |y - \hat{y}| \longrightarrow L_1^{train} = \frac{1}{n}\sum_{i=1}^{n} L_1(y^i, f(x^i))$

$\frac{1}{n}\|e\|_2^2 \dashrightarrow L_2 = (y - \hat{y})^2 \longrightarrow L_2^{train} = \frac{1}{n}\sum_{i=1}^{n} L(y^i, f(x^i))$

Loss fction

Training set loss

$\mathcal{D} = \{(x^1, y^1), \ldots (x^n, y^n)\}$

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 $\mathrm{sgn} f(x)$, 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...

$$L_{01}(y, \hat{y}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & y = \hat{y} \end{cases} = 1_{[\hat{y} \neq y]}$$

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	$\boxed{100}$
−	1	0

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

f(x)	0	battery	gas	....
y = 0	0			
b		0		
gas..			0	

Training set loss and expected loss

- **Training set loss** $\frac{1}{n}\sum_{i=1}^{n} L(y^i, f(x^i)) = L^{train}(f)$ learned predictor
- 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**. use $\underbrace{L^{test}(f) = \frac{1}{n'}\sum_{i'=1}^{n'} L(y^{i'}, f(x^{i'}))}_{\mathcal{D}^{test}}$

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

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

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

$P(X,Y) \equiv P_{XY}$

$\mathcal{D}, \mathcal{D}^{test} \sim \text{iid } P_{XY}$

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

f Loss $\mathcal{D}$ Pb: learn f

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

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

- 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.

LS Regression Problem

1. error in matrix-vector form

$\mathcal{D} = \{(x^i, y^i),\ i = 1:n\}$

$f(x) = \beta^T x$

Want $\beta = ?$

$\text{Loss} = L_2 = (y - f(x))^2 \quad \Big| \Rightarrow$ want $\beta^* = \underset{\beta}{\arg\min}\ L_2^{train}$

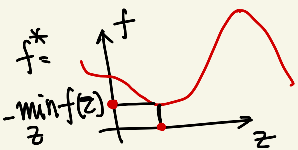
$$L_2^{train} = \frac{1}{n}\sum_{i=1}^{n}(\underbrace{y^i - \beta^T x^i}_{e^i})^2 = \frac{1}{n}\|e\|_2^2 = \frac{1}{n}$$

$$e = \begin{bmatrix} y^1 - \beta^T x^1 \\ y^2 - \beta^T x^2 \\ \cdots \end{bmatrix} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \end{bmatrix} - \begin{bmatrix} (x^1)^T\beta \\ (x^2)^T\beta \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} y^1 \\ y^2 \\ \vdots \end{bmatrix}}_{y} - \underbrace{\begin{bmatrix} \overbrace{(x^1)^T}^{d} \\ (x^2)^T \\ \cdots \end{bmatrix}}_{X} \underset{\mathbb{R}^d}{\beta} = y - X\beta$$

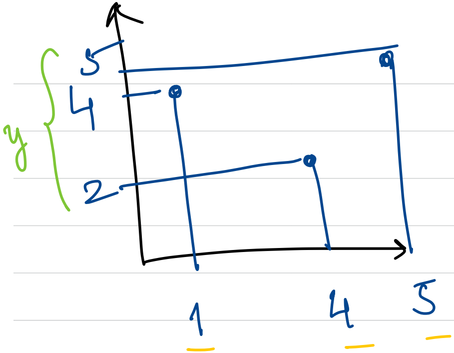
$\beta^T x = x^T \beta$

$y \in \mathbb{R}^n \qquad X = \mathbb{R}^{n \times d}$

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



$z^* = \underset{z}{\arg\min}\ f(z)$



$$\beta = [\beta_0 , \beta_1]$$

$$f(x^1) = \beta_0 \cdot 1 + \beta_1 \cdot 1 = \underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{= x^1} \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$

$$f(x^2) = \begin{bmatrix} 4 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$

$$e = \begin{bmatrix} 4 \\ 2 \\ 5 \end{bmatrix} - X\beta = y - X\beta$$

$$X = \begin{array}{c} \begin{bmatrix} 1 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} \\ \begin{array}{cc} \uparrow & \uparrow \\ \beta_1 & \beta_0 \end{array} \end{array}$$

$$e = \underset{y}{\begin{bmatrix} \; \end{bmatrix}} - \underset{X}{\begin{bmatrix} \; \end{bmatrix}} \, \beta$$

Finding the optimal β

2. $nL_2^{train}(\beta) = \|e\|_2^2 = \|y - X\beta\|^2 = (y - X\beta)^T(y - X\beta)$

$= \underbrace{y^Ty}_{0} - \beta^T X^T y - y^T X \beta + \beta^T X^T X \beta$

$a^Tz \quad = \qquad\qquad z^TAz$

Loss in matrix-vector form

$\|e\|^2 = e^Te$

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

$a^Tz = g(z) = z^Ta$

$\nabla g(z) = a$

$h(z) = z^TAz$

$\nabla h = 2Az$

A symmetric

$\min L_2^{train} \iff \boxed{\nabla L_2^{train} = 0}$

3. $\nabla L_2^{train} = 0 - X^Ty \cdot 2 + 2X^TX\beta = 0$

Find β by solving $\nabla L_2^{train} = 0$

(t.b. continued)

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.

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

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