Lecture 5. ML Algorithm Basics

1. Two Categories of ML algorithms

1) supervised learning: we learn ML models to associate the input (x) with output (y)

$x \rightarrow \boxed{\text{ML model}} \rightarrow y$

x: input / feature vector
↓
quantitative representation

y: output / target value
↓
ground truth

Linear regression

Logistic regression

Naive Bayes

Support vector machines (SVMs)

Decision trees

K-nearest neighbor (KNN)

* Artificial Neural Networks (ANNs)

2) Unsupervised learning: We only have input feature vectors

$$x \rightarrow \boxed{ML} \rightarrow \text{no } y \text{ provided.}$$

① Dimensionality reduction (DR)

high-dimensional data

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{pmatrix} \longrightarrow \boxed{DR} \rightarrow \bar{x}_i = \begin{pmatrix} \bar{x}_{i1} \\ \bar{x}_{i2} \\ \vdots \\ \bar{x}_{iK} \end{pmatrix} \qquad K \ll m$$

m is large

- Principal component analysis (PCA) → linear DR
- manifold learning ⟶ a set of algorithms for non-learn DR.

② Clustering: group data. e.g., k-means

2. Four Components in ML

1) Data sets : training set (60%-70%) + validation set (15%-20%) ← training stage.

- training set: find best model
- validation set: determine the hyperparameters

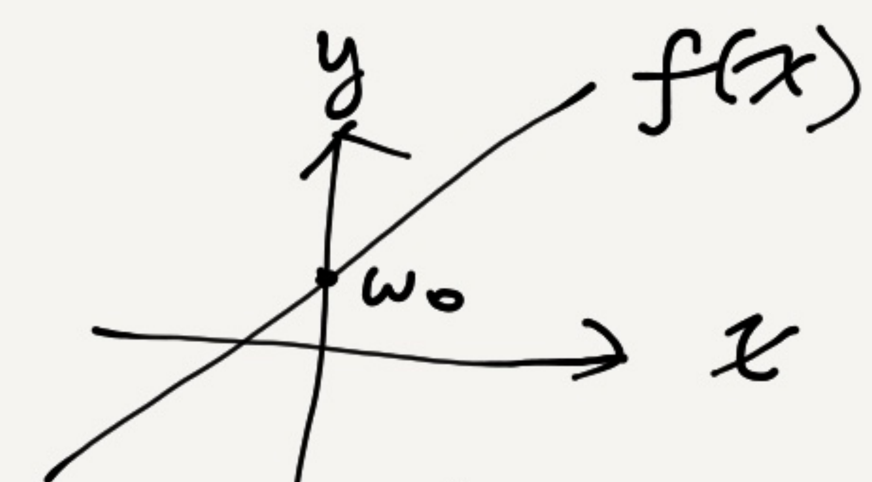
test set (15%-20%) : evaluate generalization performance, i.e., performance on new data

2) Model : defines a function to compute prediction from input

$$\underset{\text{input}}{x} \longrightarrow \underset{\text{model}}{\boxed{f(x)}} \longrightarrow \underset{\text{prediction}}{\hat{y}}$$

simple model : linear model.

1D: $f(x) = w_0 + w_1 x$



2D: $f(x) = w_0 + w_1 x_1 + w_2 x_2$ plane $\quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

3D: $f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_4$ hyperplane $\quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$

mD:

$$f(x) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$= w^T x + w_0$$

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

generic form of linear function.

$$f(x) = w^T x$$

$$w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{pmatrix}_{m+1} \qquad x = \begin{pmatrix} x_0 = 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix}$$

3) Loss / cost / objective function : defines training goal, e.g., prediction $(\hat{y})$ should match the target value (y)

$$x \longrightarrow \boxed{f(x)} \longrightarrow \boxed{\hat{y} = y}$$

① Exact match

$$\mathcal{L}_0 = \begin{cases} 0 & \text{if } \hat{y} = y \\ +\infty & \text{if } \hat{y} \neq y \end{cases}$$

residual : $y - \hat{y}$

② Residual-based Loss

n: # of data samples.

$$\mathcal{L}_1 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 .$$ residual sum of squares. RSS.

$$\mathcal{L}_2 = \frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\mathcal{L}_3 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$: mean square error (MSE)

$$\mathcal{L}_4 = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\mathcal{L}_5 = \sum_{i=1}^{n} y_i \cdot \log \hat{y}_i$$ ⟶ cross-entropy

4). Optimization algorithm / optimizer / solver

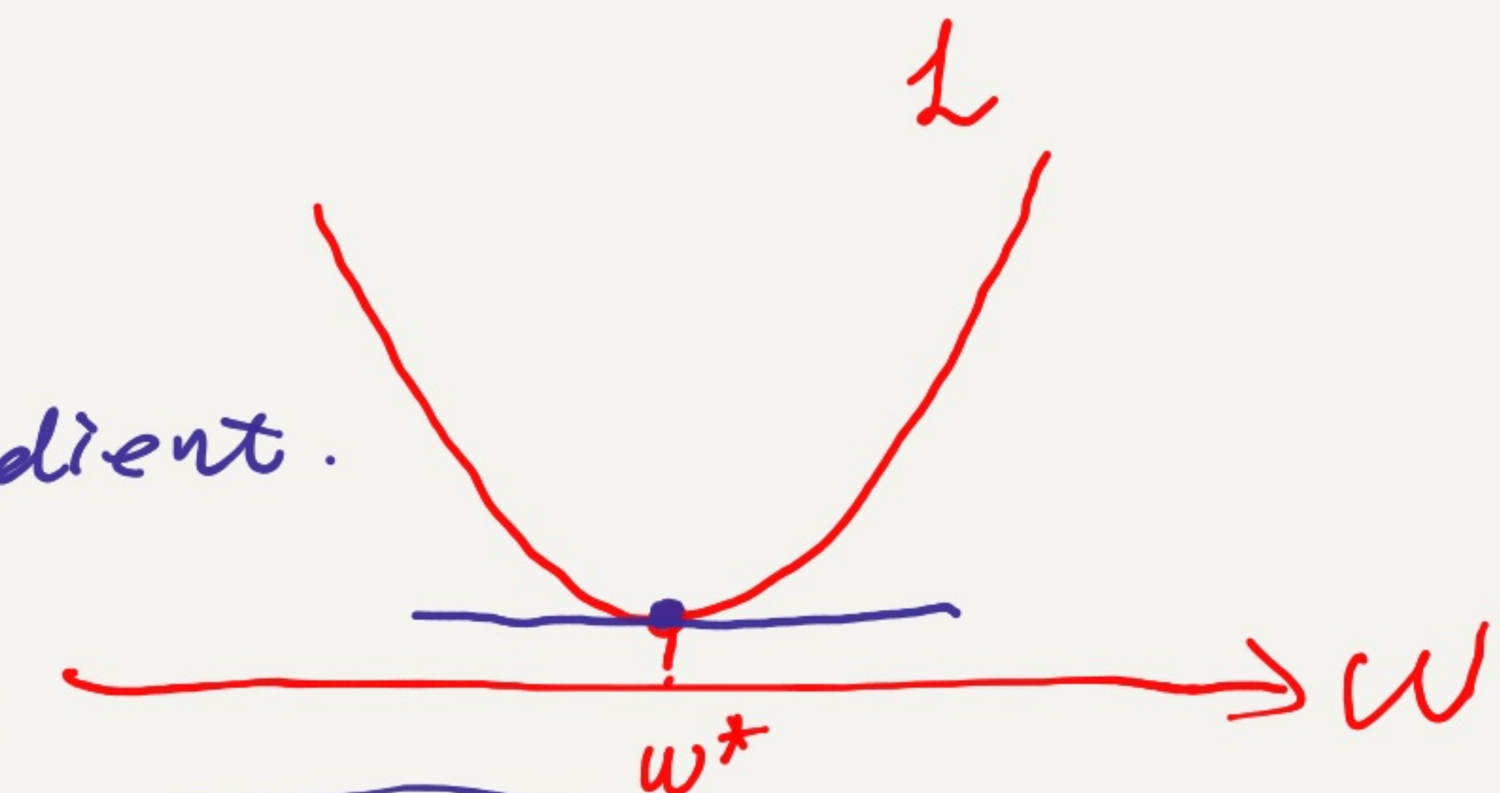
defines steps to find $f(x)$ that minimize the loss function

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$w^* = \arg\min_{w} \overbrace{\frac{1}{2}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$ w^*: best model parameters

$\downarrow$ $\hat{y} = f(x) = w^T x$ (linear model)

$$= \arg\min_{w} \frac{1}{2}\sum_{i=1}^{n} (w^T x_i - y_i)^2$$



How ??

The extrem points have zero gradient.

We can solve $\nabla \mathcal{L}|_{w=w^*} = 0$ to get w^*.

$\boxed{\nabla \mathcal{L}|_{w=w^*} = 0}$

$\mathcal{L}_0 = w^T x - y \longrightarrow \nabla_w \mathcal{L}_0 = x \longrightarrow$ $\boxed{\text{Obtain } w^*}$

$\mathcal{L}_1 = (w^T x - y)^2 \longrightarrow$