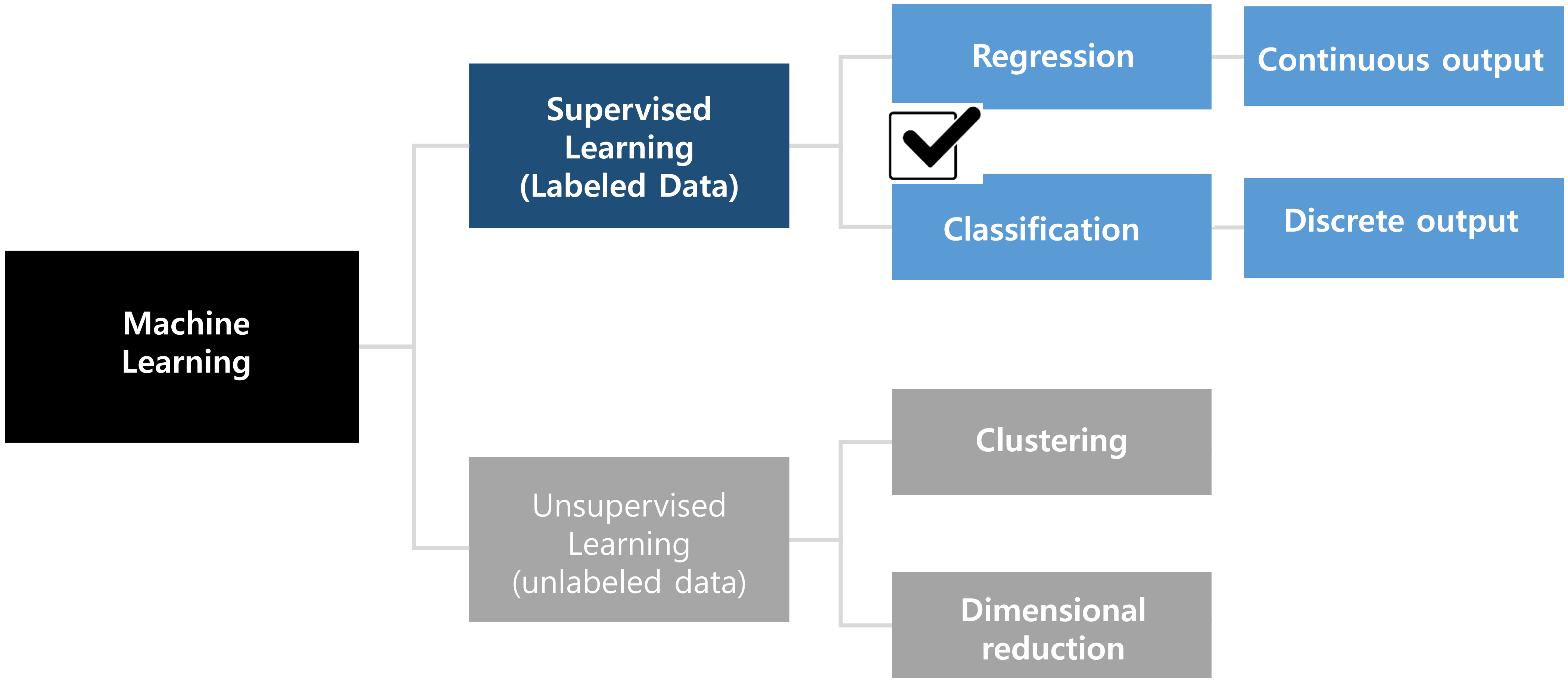
Linear Classification

Prof. Je-Won Kang
Electronic & Electrical Engineering
Ewha Womans University

Machine learning problems



Linear classification

hyper plane 구해서
linear combination에 의해서
구분하자.

- Predict a discrete output y (classification ID) from $\boldsymbol{x}$ when $D = (\boldsymbol{x}, y)$ is given
  - ID = 0 or 1 (binary classification)
  - ID = 0, 1, ..., N-1 (multi classification)

- Hypothesis set $\mathcal{H}$: a set of lines

model parameter
feature

$$\boxed{h_w(x) = w_0 + w_1 x_1 + \cdots + w_d x_d = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}}$$

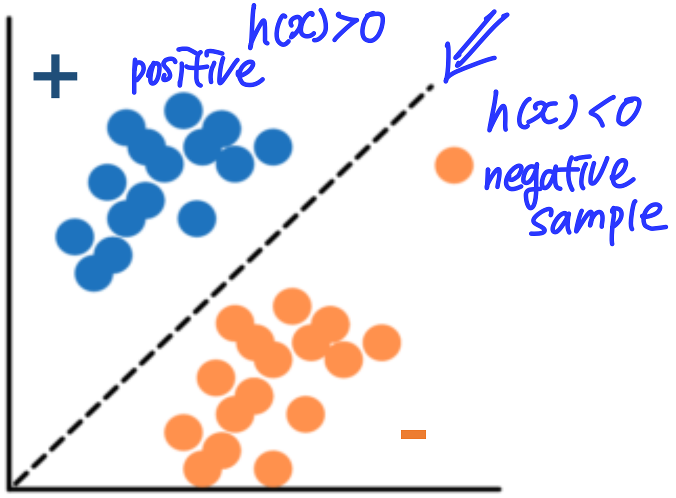
$\boldsymbol{w}$: model parameter (learnable parameter)

$$\boxed{h_w(x) = w_0 + w_1 \phi(x_1) + \cdots + w_d \phi(x_d) = \boldsymbol{w}^{\mathrm{T}} \phi(\boldsymbol{x})}$$

Linear model with a set of features

h(x)=0

Hyper plane
(decision boundary)



Linear separable

Example: image recognition

hyper plane이
다수 존재

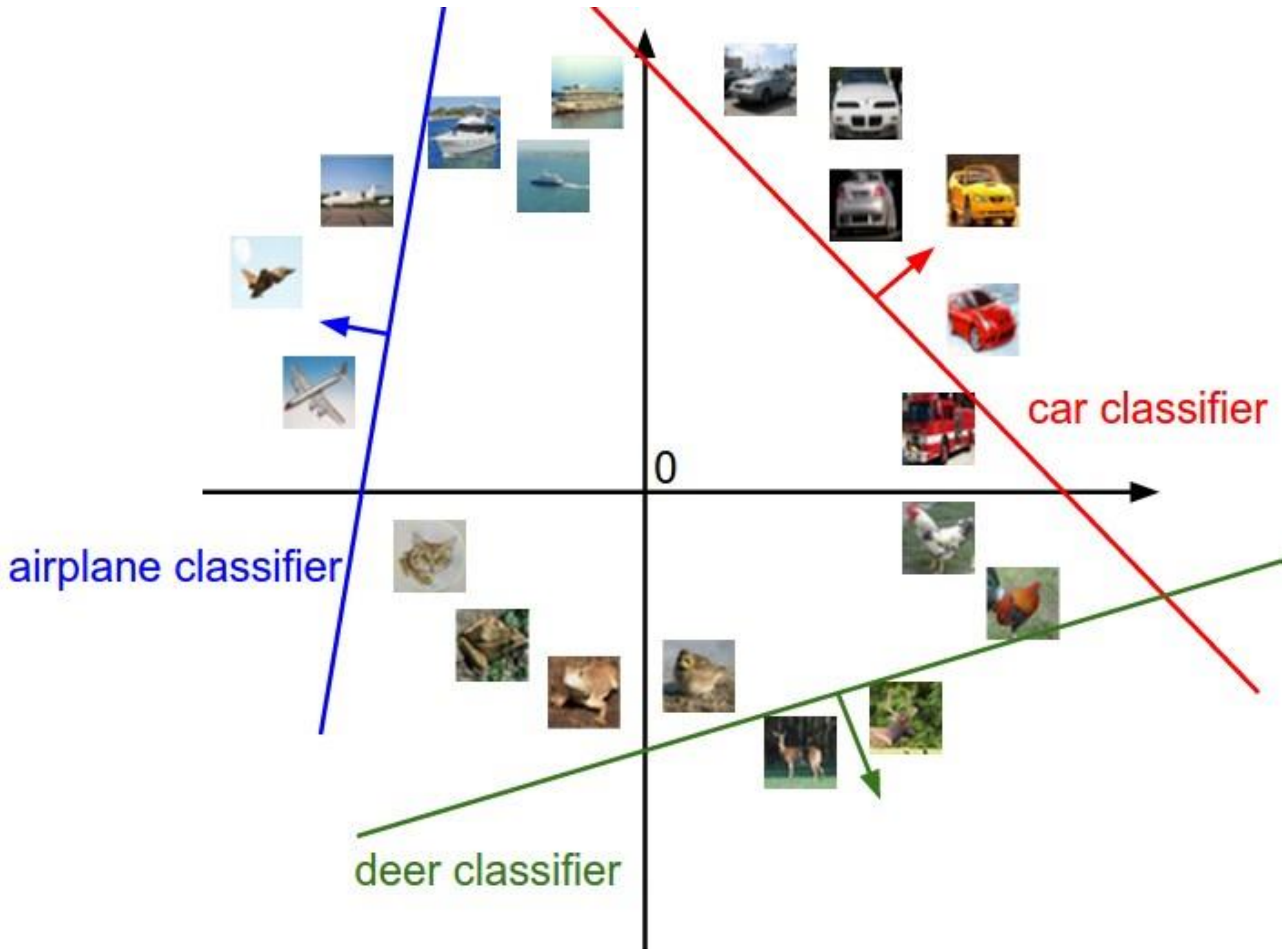


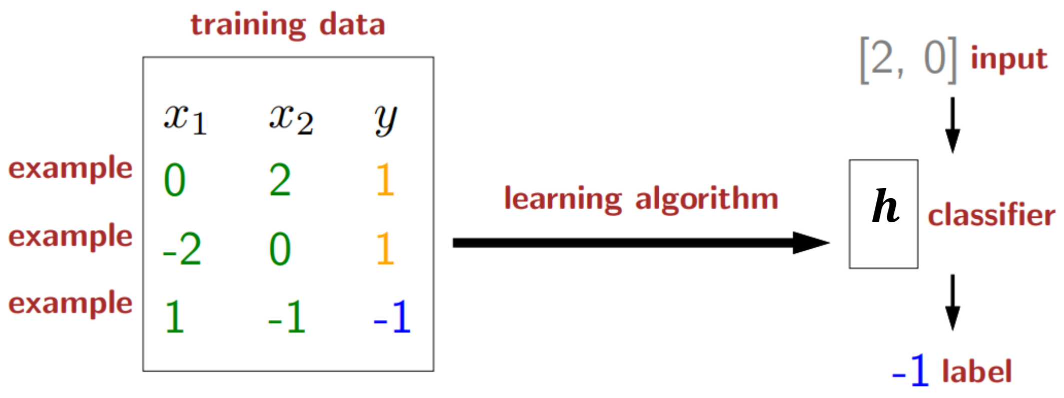
Image from : CS231n: Deep Learning for Computer Vision

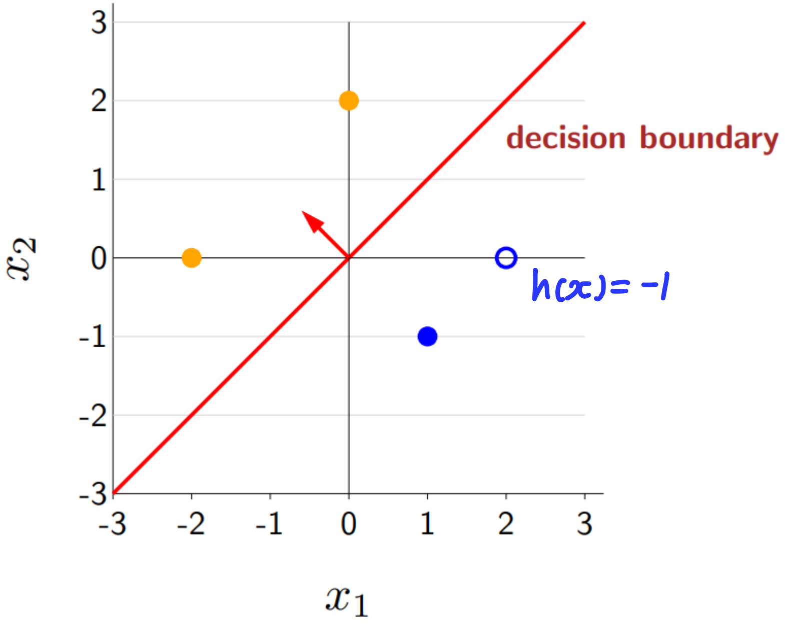
Problem formulation

- $\boldsymbol{X} = \mathbf{R}^d$ is an input space
  - $\mathbf{R}^d$: a d-dimensional Euclidean space
  - input vector $\boldsymbol{x} \in \boldsymbol{X}: \boldsymbol{x} = (x_1, x_2, \dots, x_d)$, e.g. $d = 2$
- $\boldsymbol{Y} = \{+1, -1\}$ is an output space
  - Binary (yes/no) decision
- Now, we want to approximate a target function f
  - $f: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ (unknown ideal function)
  - Data $(\boldsymbol{x}^1, y^1)$, … , $(\boldsymbol{x}^N, y^N)$; dataset where $y^N = f(\boldsymbol{x}^N)$
  - Correct label is ready for a training set
  - **Hypothesis** $g: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ (ML model to approximate f) : $g \in \boldsymbol{H}$

Linear classification framework

Hypothesis function to build a decision boundary





Which predictor?
Hypothesis class

$$h(x) = \text{sign}\ (w^{\mathrm{T}} x)$$

How good is a predictor?
Loss function

Zero-one loss
Hinge loss
Cross-entropy loss

How to compute the best predictor?
Optimization algorithm

Gradient descent algorithm

Linear classification model

- The linear formula $g \in \boldsymbol{H}$ can be written as

$$h(x) = \text{sign}\,\left(\left(\sum_{i=1}^{d} w_i x_i\right) + w_0 \right)$$

$$= \text{sign}\,\left(\left(\sum_{i=0}^{d} w_i x_i\right)\right),\; x_0 = 1$$

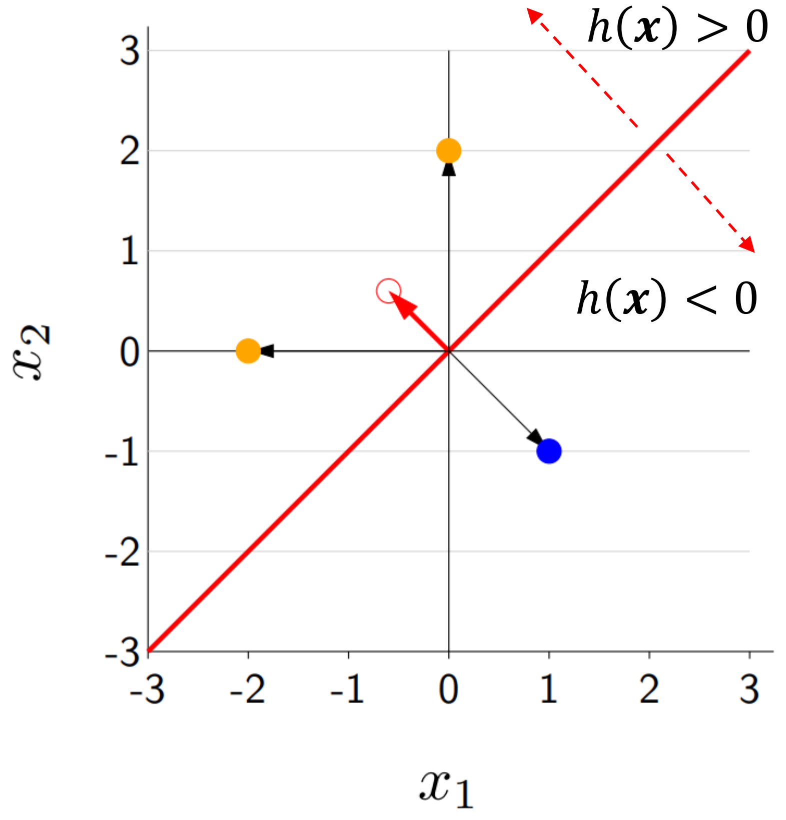
$$= \text{sign}\,(w^{\mathrm{T}} x)$$

w_o: *a bias term*

$$sign(x) = \begin{cases} 1, & if\ x > 0 \\ -1, & if\ x < 0 \end{cases}$$

x_o: 1

$h(\boldsymbol{x}) = w_0 + w_1x_1 + w_2x_2 = 0$



Example of linear classifier

$$h(\boldsymbol{x}) = \text{sign}\left(\overbrace{[-1,1]}^{\boldsymbol{w}}\ \overbrace{[x_1,x_2]^{\mathrm{T}}}^{\boldsymbol{\phi}(\boldsymbol{x})}\right)$$

$$\text{sign}(z) = \begin{cases} +1 & \text{if } z > 0 \\ -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \end{cases}$$

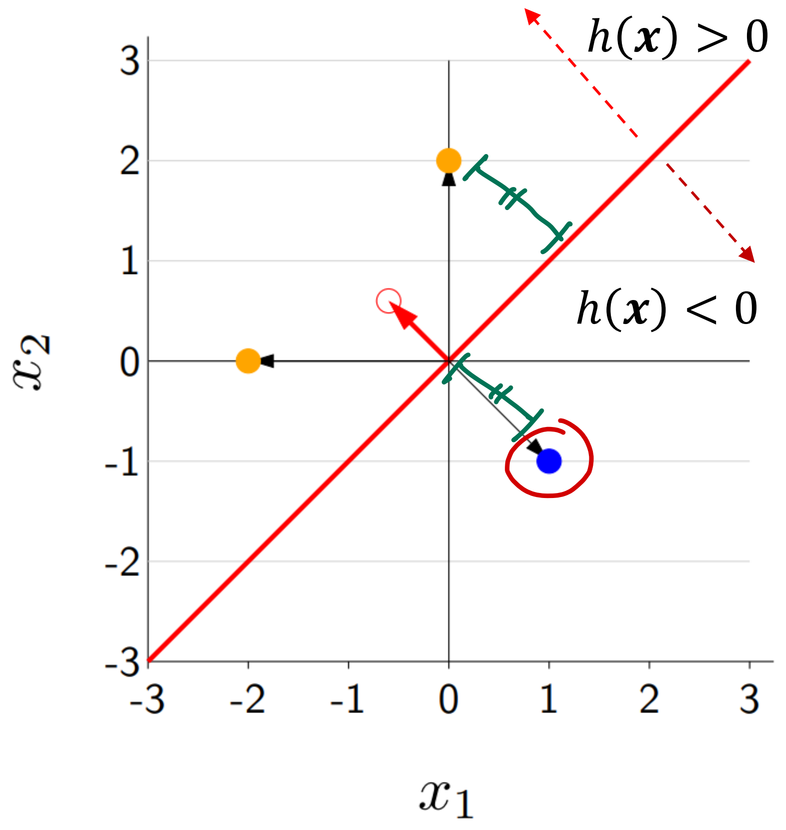
$$h([0,2]) = \text{sign}\left(\underline{[-1,1]}\ \underline{[0,2]^{\mathrm{T}}}\right) = 1$$

2

$$h([1,-1]) = \text{sign}\left(\underline{[-1,1]}\ \underline{[1,-1]^{\mathrm{T}}}\right) = -1$$

$w^T x$

$$h(\boldsymbol{x}) = w_0 + w_1x_1 + w_2x_2 = 0$$

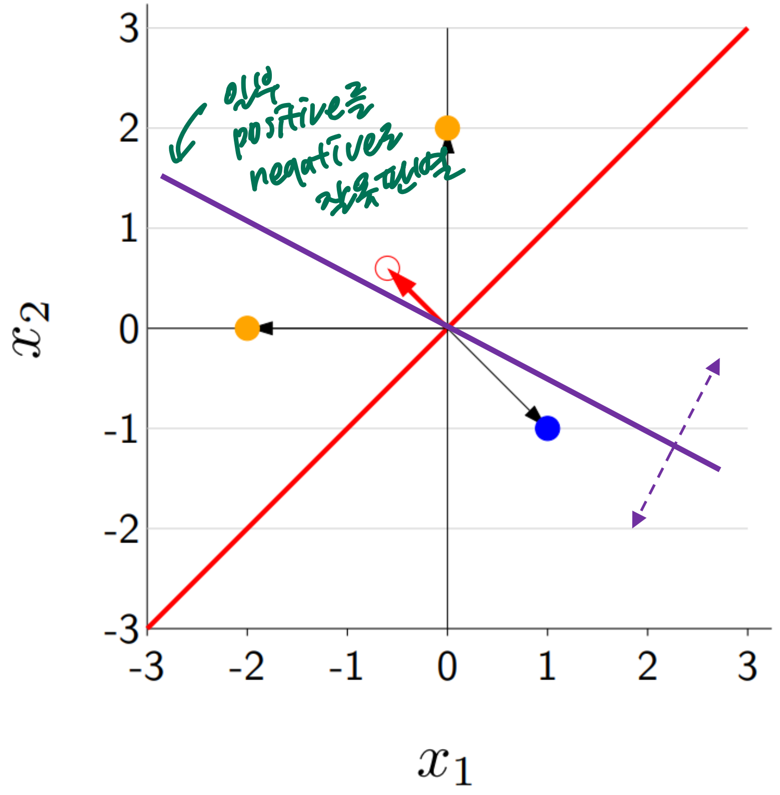


Hypothesis class : which classifier?

$h(\boldsymbol{x}) = \text{sign}\,([-1, 1]\,[x_1, x_2]^{\mathrm{T}})$

$h(\boldsymbol{x}) = \text{sign}\,([0.5, 1]\,[x_1, x_2]^{\mathrm{T}})$

$h(\boldsymbol{x}) = w_0 + w_1 x_1 + w_2 x_2 = 0$



For optimization

Define a metric and compute an error

$$\text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) = \mathbf{1}[f_{\mathbf{w}}(x) \neq y]$$ zero-one loss

zero-one-function

$$\text{Loss}([0, 2], 1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [0, 2]) \neq 1] = 0$$

$$\text{Loss}([-2, 0], 1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [-2, 0]) \neq 1] = 1$$

$$\text{Loss}([1, -1], -1, [0.5, 1]) = \mathbf{1}[\text{sign}([0.5, 1] \cdot [1, -1]) \neq -1] = 0$$

Score and margin

결정과정에서
모델이 얼마나
confident한지 측정

- Input data : x
- Predicted label : $h(\boldsymbol{x}) = \text{sign}\,(w^T \phi(\boldsymbol{x}))$
- Target label: y

✓ Score : the score on an example (x, y) is $w \cdot \phi(x)$, how **confident** we are in predicting +1.

score x y

✓ Margin : the margin on an example (x, y) is $(w \cdot \phi(x))y$, how **correct** we are.

score · y

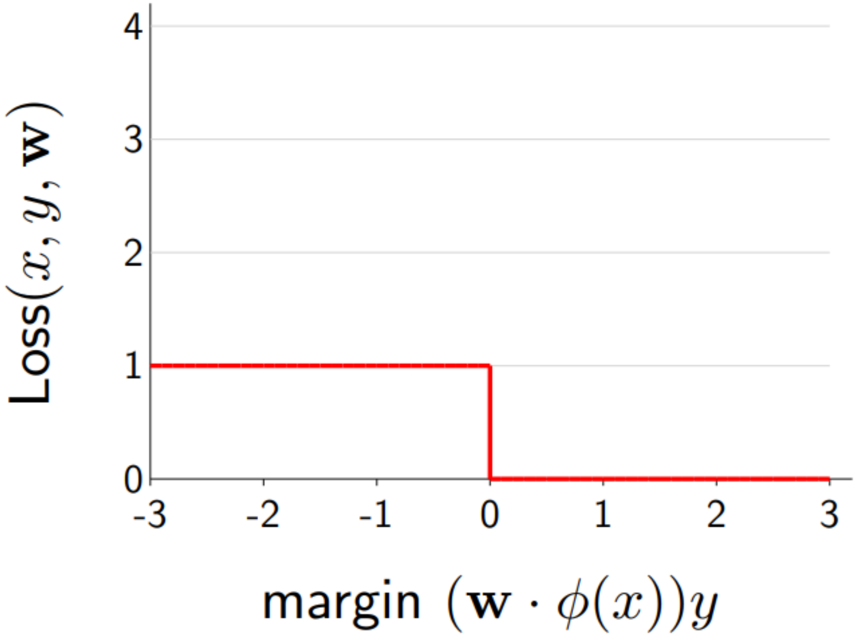
$\boxed{\oplus \quad y=1}$ → margin 大 → 정답맞춤.

$\boxed{\ominus \quad y=-1}$ → margin 大 → 〃

$\oplus \quad y=-1$ → margin 小 → 틀림.

Zero-one loss

$$\text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) = \mathbf{1}[\underbrace{(\mathbf{w} \cdot \phi(x))y}_{\text{margin}} \leq 0]$$



The goal is to minimize the loss

To run gradient descent, compute the gradient:

Gradient is zero almost everywhere!

학습불가

$$\nabla_{\mathbf{w}} \text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \nabla \text{Loss}_{0\text{-}1}(x, y, \mathbf{w})$$

$$\nabla_{\mathbf{w}} \text{Loss}_{0\text{-}1}(x, y, \mathbf{w}) = \nabla \mathbf{1}[(\mathbf{w} \cdot \phi(x))y \leq 0]$$

Slides from regression and classification, CS221 Stanford

✓ 문제해결

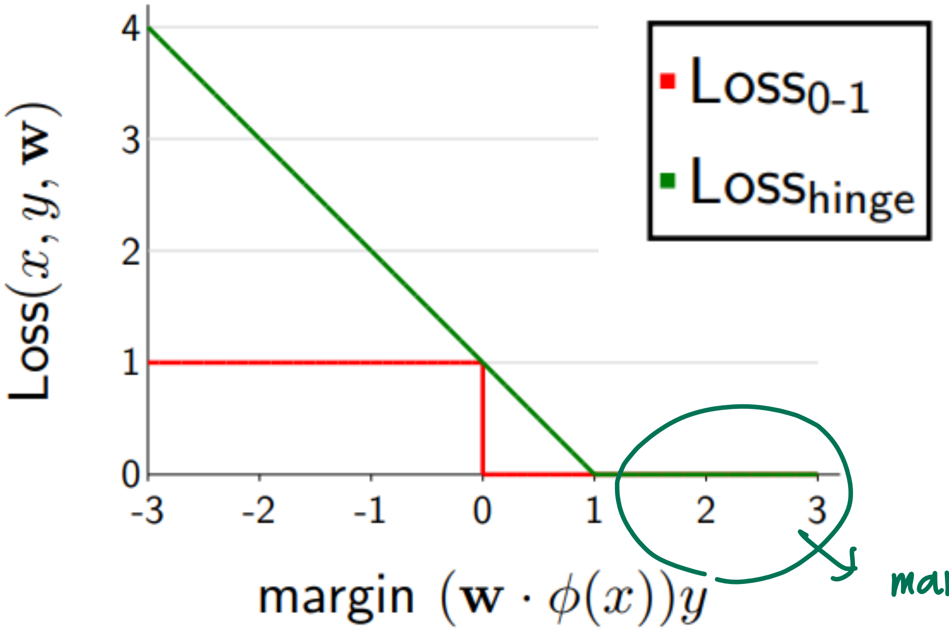
Hinge loss

1-margin ⇒ 정답 잘 맞추고 있는 상태면 1-margin 음수값일것.

margin

$$\text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \max\{1 - \underline{(\mathbf{w} \cdot \phi(x))y}, 0\}$$

- Zero loss if it is classified confidently and correctly
- Misclassification incurs a linear penalty w.r.t. confidence



margin이 ⊕일때

margin

$$\nabla \text{Loss}_{\text{hinge}}(x, y, \mathbf{w}) = \begin{cases} -\phi(x)y & \text{if } 1 > \{(\mathbf{w} \cdot \phi(x))y\} \\ 0 & \text{otherwise} \end{cases}$$

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

Slides from regression and classification, CS221 Stanford

Cross-entropy loss

- Considers two probability mass functions (pmf) $\{p, 1-p\}$ and $\{q, 1-q\}$ with a binary outcomes
- Cross entropy for these two pmfs : defined by

$$D(\mathrm{p} \,\|\, \mathrm{q}) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

$$p \log \frac{1}{q} + (1-p) \log \frac{1}{1-q}$$

Cf. $\displaystyle \sum_{x \in X} p(x) \log \frac{1}{q(x)} = H(p) + \underbrace{D(\mathrm{p} \,\|\, \mathrm{q})}_{\text{K-L divergence}}$ ← p와 q의 유사도에 따라 값 바뀜.

거의 안바뀜.

Kullback-Leibler (K-L) divergence is a measure of dissimilarity of two distributions

p와 q 유사하면 loss 줄어든다

- Cross entropy measures the error when approximating an observed pmf $\{p, 1-p\}$ between a fitted pmf $\{q, 1-q\}$

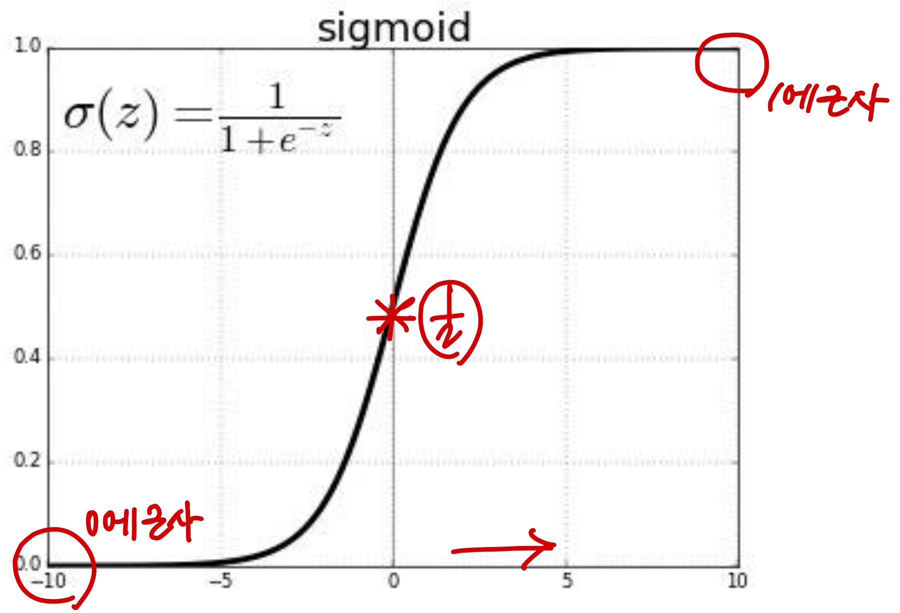
Q. 계산한 Score 값을 어떻게 확률값으로 Mapping 할 수 있을까?

A. Sigmoid

Cross-entropy loss

Real value $h = w^T \phi(\boldsymbol{x})$ 0 or 1

Height (cms)	Weight (kg)	Fitness
150	50	Fit
187	75	Fit
156	80	Not Fit
163	60	Fit
170	49	Not Fit
179	70	Fit



score | $h = w^T \phi(\boldsymbol{x})$

Sigmoid | $\sigma(z) = \frac{1}{1+e^{-z}}$

Logistic Model

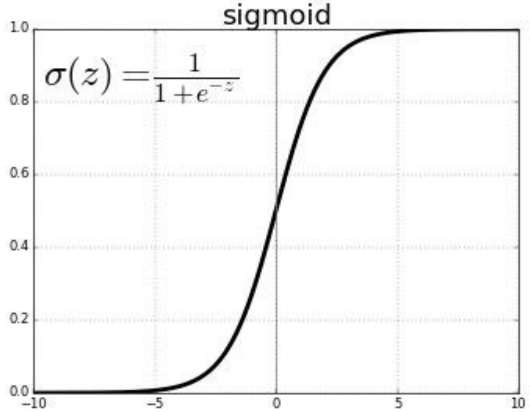
$$\sigma(z) = \frac{1}{1+e^{-W^T X}}$$

Sigmoid function

- Squash the output of the linear function

$$\sigma(-\mathrm{w}^T x) = \frac{1}{1 + e^{-\mathrm{w}^T x}}$$

0~1 사이 값으로…



- A better approach : interpret as a probability

$$P_{\mathrm{w}}(y = 1|x) = \sigma(-\mathrm{w}^T x) = \frac{1}{1 + e^{-\mathrm{w}^T x}}$$

$$P_{\mathrm{w}}(y = 0|x) = 1 - \sigma(-\mathrm{w}^T x) = \frac{e^{-\mathrm{w}^T x}}{1 + e^{-\mathrm{w}^T x}}$$

Cross-entropy loss

$\bar{Y}$ (estimated)	
0.7	0.9
0.2	0.1
0.1	0.0

더 정확한 모델이라면

$$\mathrm{CE}(S, L) = -\sum_{\forall i} L_i \log(S_i)$$

$L = Y$ (real)
1.0
0.0
0.0

$\Sigma y_i = 1$

Understanding this Cost Function

- Suppose that $L = [1,0,0]$,
  - If $\bar{Y} = [1,0,0]$, then $D = -1 \cdot \log 1 - 0 \cdot \log 0 - 0 \cdot \log 0 = -1 \cdot 0 - 0 \cdot (-\infty) - 0 \cdot (-\infty) = 0$ (no cost).
  - If $\bar{Y} = [0,1,0]$, then $D = -1 \cdot \log 0 - 0 \cdot \log 1 - 0 \cdot \log 0 = -1 \cdot (-\infty) - 0 \cdot 0 - 0 \cdot (-\infty) = \infty$ (huge cost).
  - If $\bar{Y} = [0,0,1]$, then $D = -1 \cdot \log 0 - 0 \cdot \log 0 - 0 \cdot \log 1 = -1 \cdot (-\infty) - 0 \cdot (-\infty) - 0 \cdot 0 = \infty$ (huge cost).

Gradient Descent Method

$$W \leftarrow W - \alpha \frac{\partial}{\partial W} \mathrm{CE}$$

Training a linear classifier

- Iterative optimization using gradient descent

1. Initialize weights at time step $t = 0$
2. Compute the gradients

$$\nabla E_{Train}(w_t) = \boxed{-\frac{1}{N}\sum_{n=1}^{N}\frac{y_n x_n}{1 + e^{-y_n \mathrm{w}_t^T x_n}}}$$

3. Set the direction to move : $v_t = -\nabla E_{Train}(w_t)$

4. Update weights $w_{t+1} = w_t + \alpha v_t$

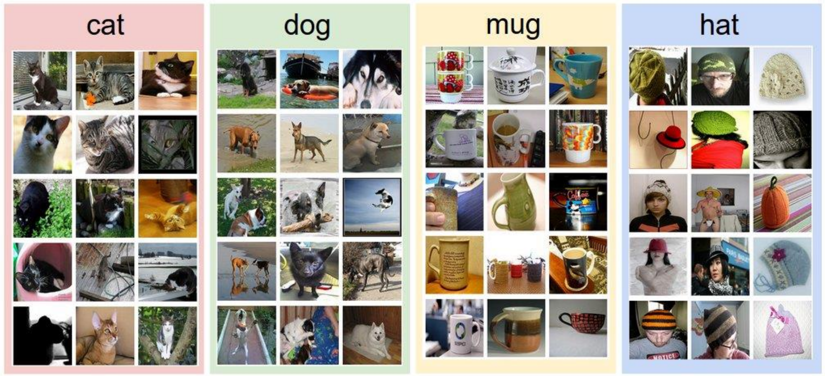
5. Iterate to next step until converging

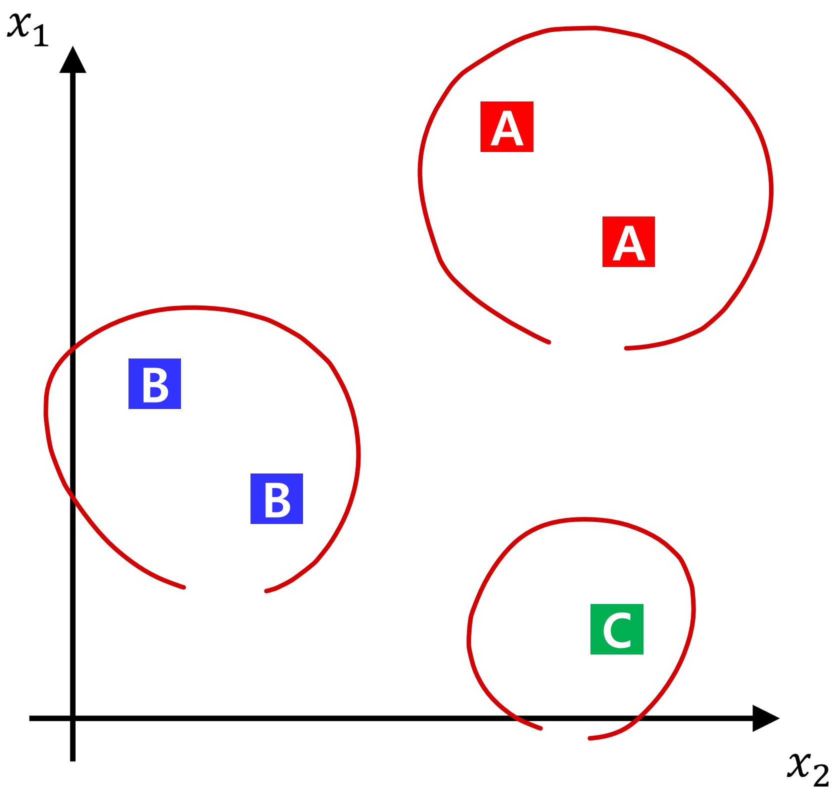
Multiclass classification

- Not all classification predictive models support multi-class classification.
- split the multi-class classification dataset into multiple binary classification datasets and fit a binary classification model on each.

Classification (Examples)

- Pass/Fail (Binary Classification)
- **(Multi-Level Classification)**

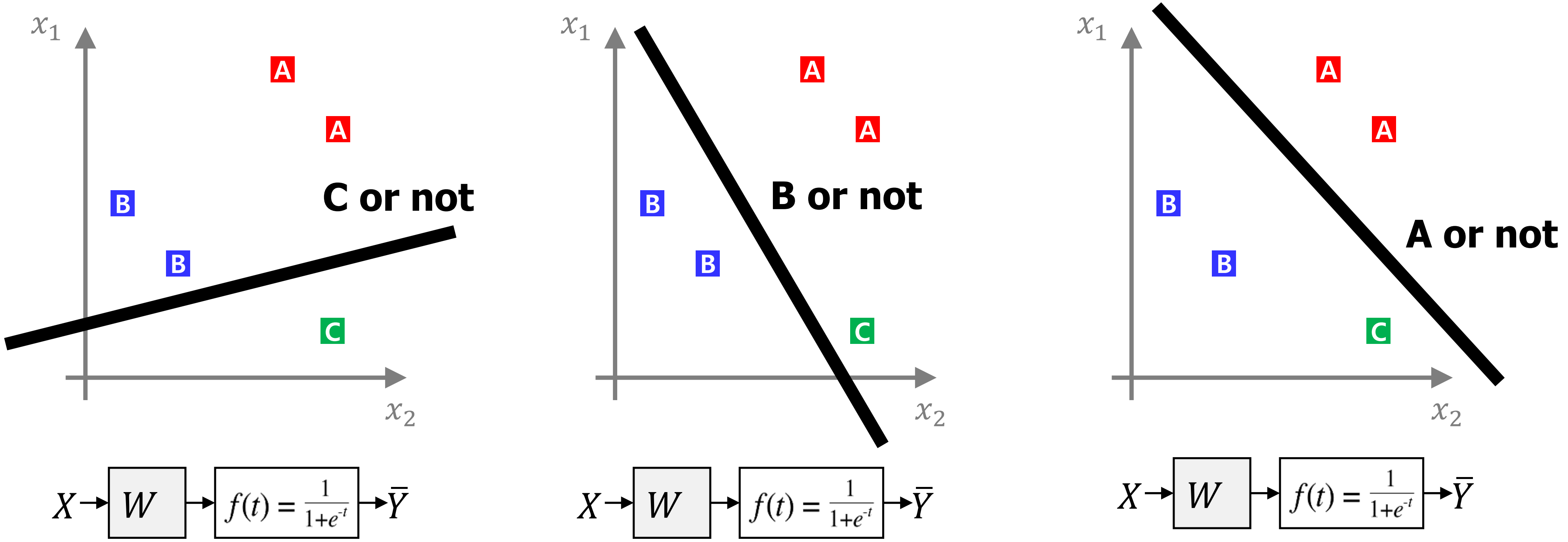




Multiclass classification

One-VS-All

multiclass를 binary로

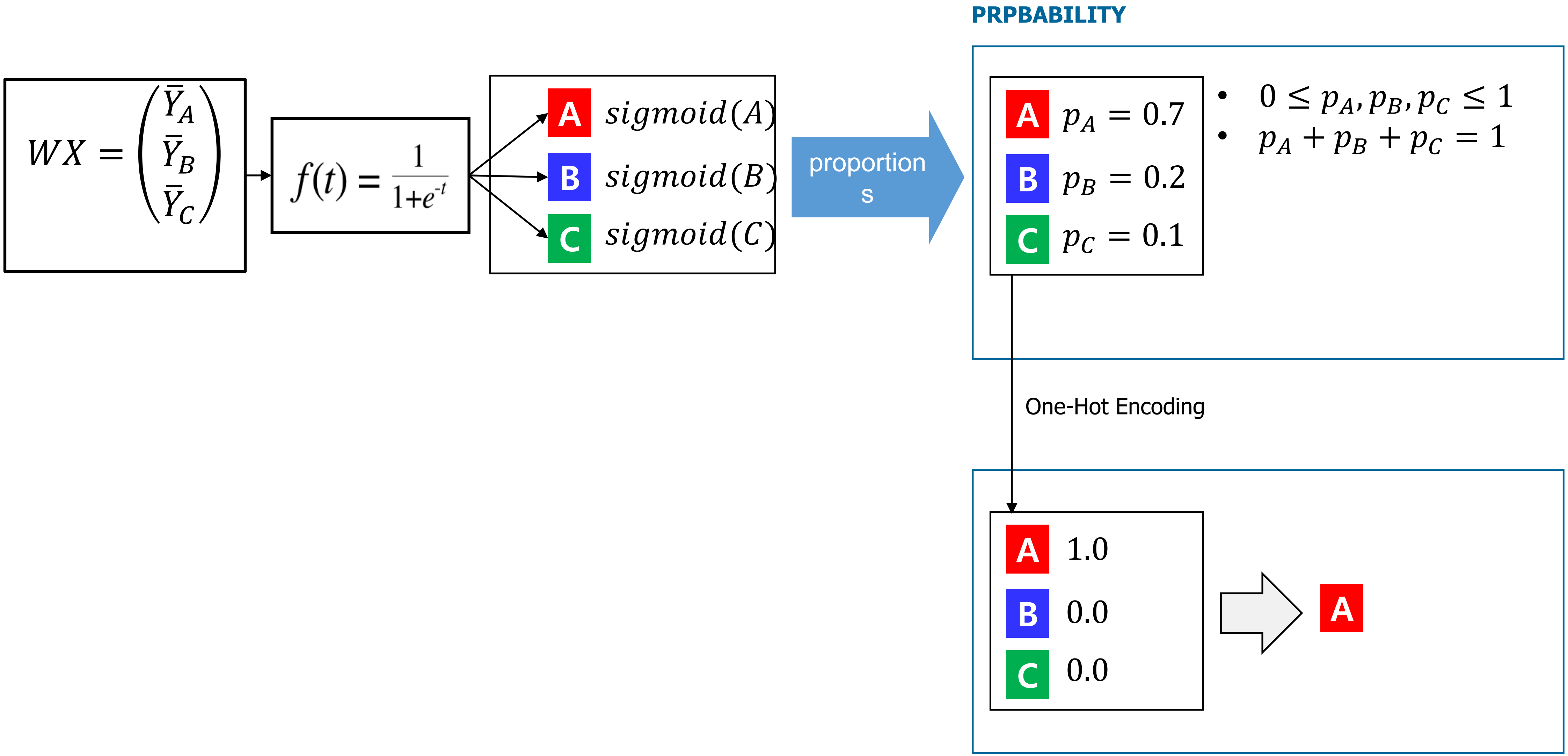


score

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \cdot \begin{array}{c} \begin{array}{ccc} A & \quad B & \quad C \end{array} \\ \begin{pmatrix} W_{A1} & W_{B1} & W_{C1} \\ W_{A2} & W_{B2} & W_{C2} \end{pmatrix} \end{array} = \begin{pmatrix} \overset{A}{\boxed{x_1 \cdot W_{A1} + x_2 \cdot W_{A2}}} & \overset{B}{\boxed{x_1 \cdot W_{B1} + x_2 \cdot W_{B2}}} & \overset{C}{\boxed{x_1 \cdot W_{C1} + x_2 \cdot W_{C2}}} \end{pmatrix}$$

like one-hot-encoding.

Multiclass classification



Advantage of linear classification

- Simple!
- Interpretability (example in Murphy 2012) 해석가능성 증가
  - x_1: the number of cigarettes per day , x_2: minutes of exercise per day
  - The goal is to predict $P(Y = \text{lung cancer})$
  - Assume we have estimated the best parameter $w = (1.3, -1.1)$ to have $h(x) = 1.3x_1 - 1.1x_2$

$\longrightarrow$ For every cigarettes per day, the risk increased by a factor of $e^{1.3}$

$$\frac{p(y=+1 \mid x)}{p(y=-1 \mid x)} = e^{w^T x} = e^{w_1 x_1 + w_2 x_2}$$

Reference

- Book: Pattern Recognition and Machine Learning (by Christopher M. Bishop)
- Book: Machine Learning: a Probabilistic Perspective (by Kevin P. Murphy)
- https://www.andrewng.org/courses/