Logistic Regression

Data Intelligence and Learning (DIAL) Lab
Prof. Jongwuk Lee

Linear Regression for Classification

Recap: Linear Regression

- **Given** $\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right): 1 \le i \le n\right\}$
  - $\mathbf{x}^{(i)} = (1, x_{i1}, \dots, x_{id}), y^{(i)} \in \mathbb{R}$

- **Find** $f\left(\mathbf{x}^{(i)}\right) = \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}$ **that minimizes error function** $E(\mathbf{w})$.

$$E(\mathbf{w}) = \sum_{i=1}^{n} \left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right)^2$$

$$f\left(\mathbf{x}^{(i)}\right) = \sum_{j=0}^{d} w_j x_{ij} = w_0 + w_1 x_{i1} + \cdots + w_d x_{id}$$

Example: Linear Regression

- Fitting a linear model with a set of variables $x_0, x_1, \dots, x_d$

$$f(\mathbf{x}) = w_0 x_0 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d$$

- Age and systolic blood pressure (SBP)

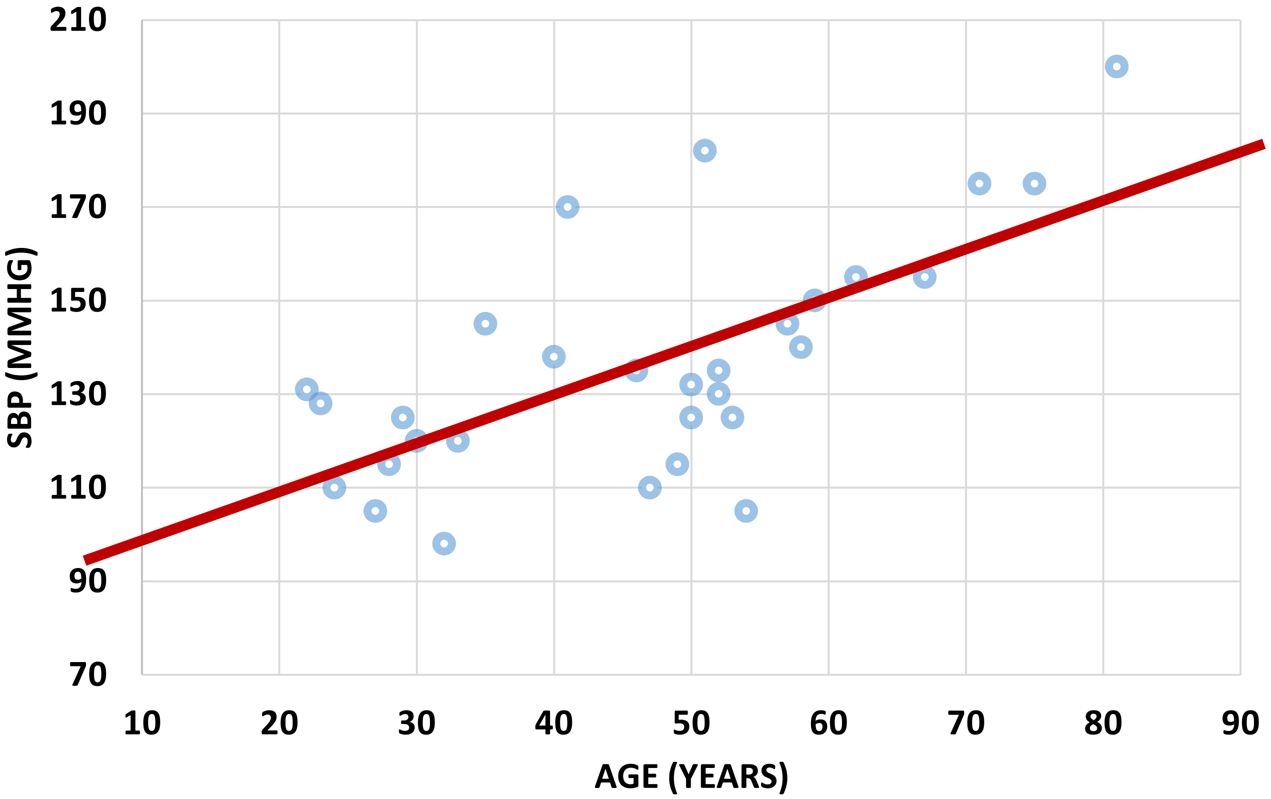
Age	SBP
22	131
23	128
24	110
27	105
28	115
29	125
30	120
32	98
33	120
35	145

Age	SBP
40	138
41	170
46	135
47	110
49	115
50	132
50	125
51	182
52	130
52	135

Age	SBP
53	125
54	105
57	145
58	140
59	150
62	155
67	155
71	175
75	175
81	200

Example: Linear Regression

$$SBP = 1.0538 \times Age + 87.361$$



Classification Problem

- Age and coronary heart disease (CD)

Age	CD
22	0
23	0
24	0
27	0
28	0
29	0
30	1
32	0
33	1
35	0

Age	CD
40	1
41	0
46	1
47	0
49	0
50	1
50	0
51	0
52	1
52	0

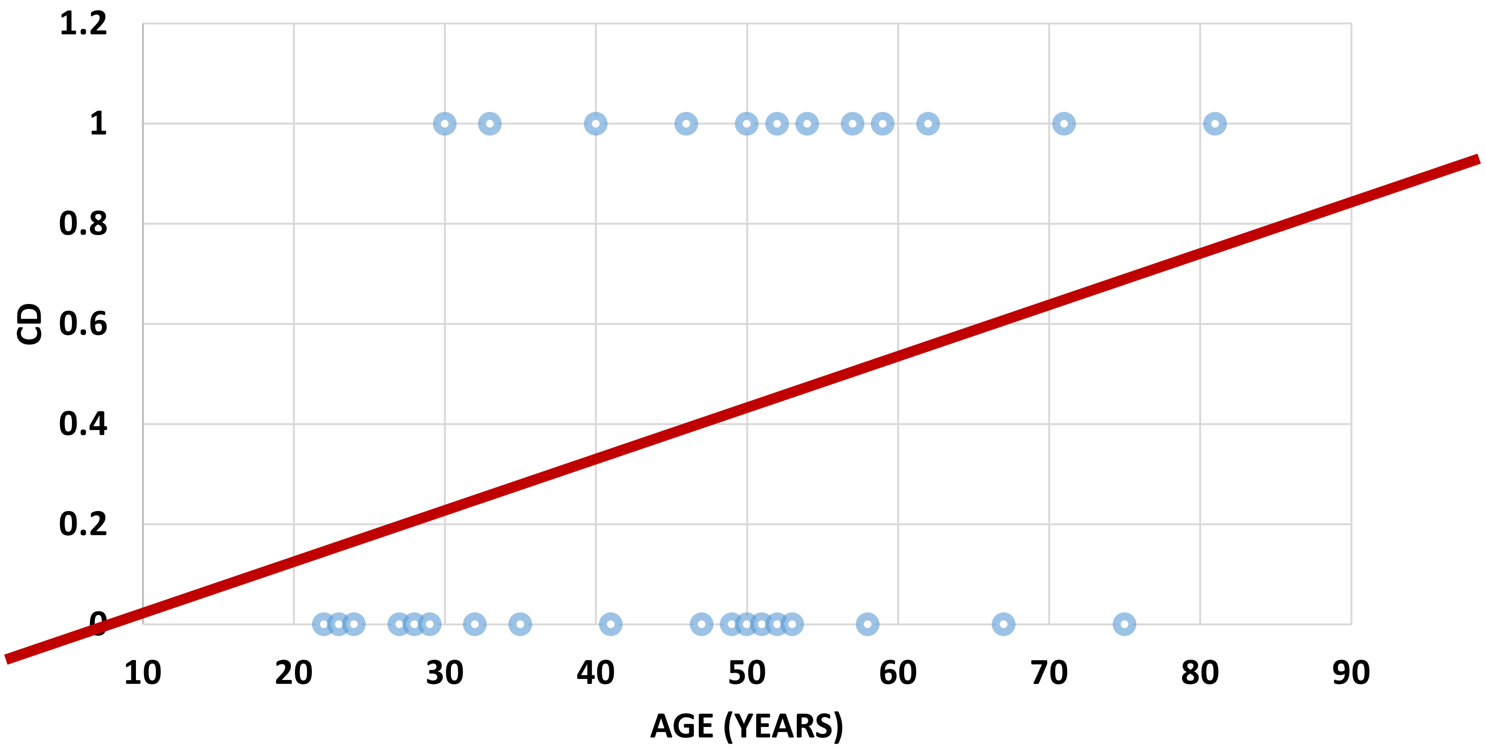
Age	CD
53	0
54	1
57	1
58	0
59	1
62	1
67	0
71	1
75	0
81	1

- What about applying the linear regression model?

Classification Problem

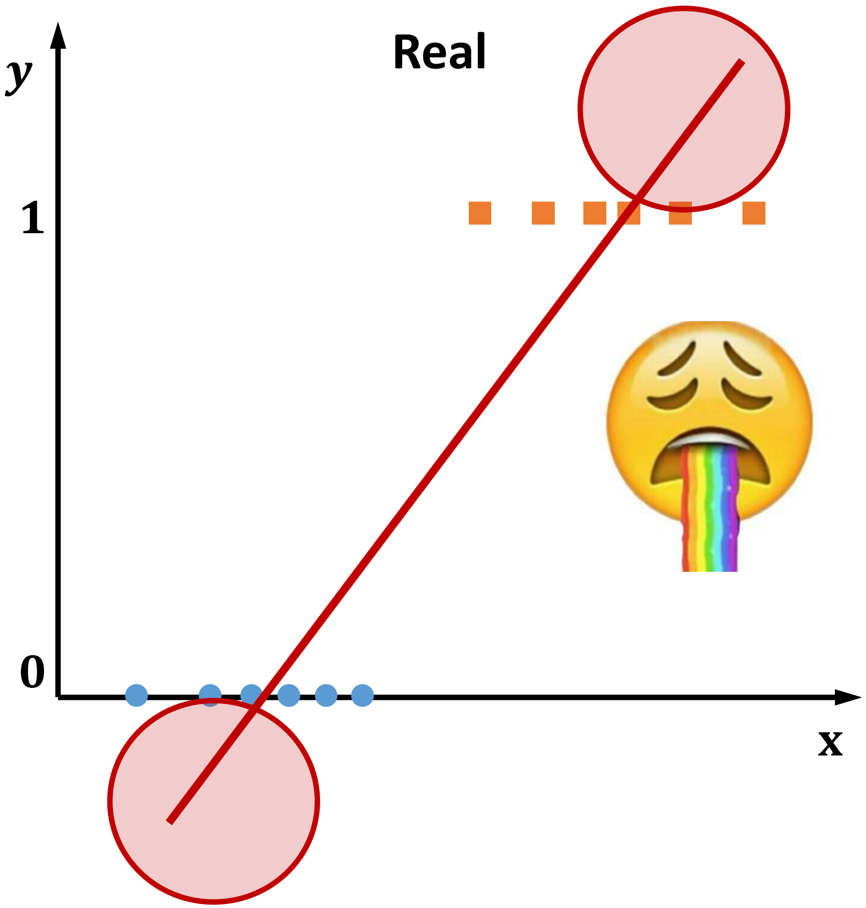
- In this case, the output can be > 1 or < 0.

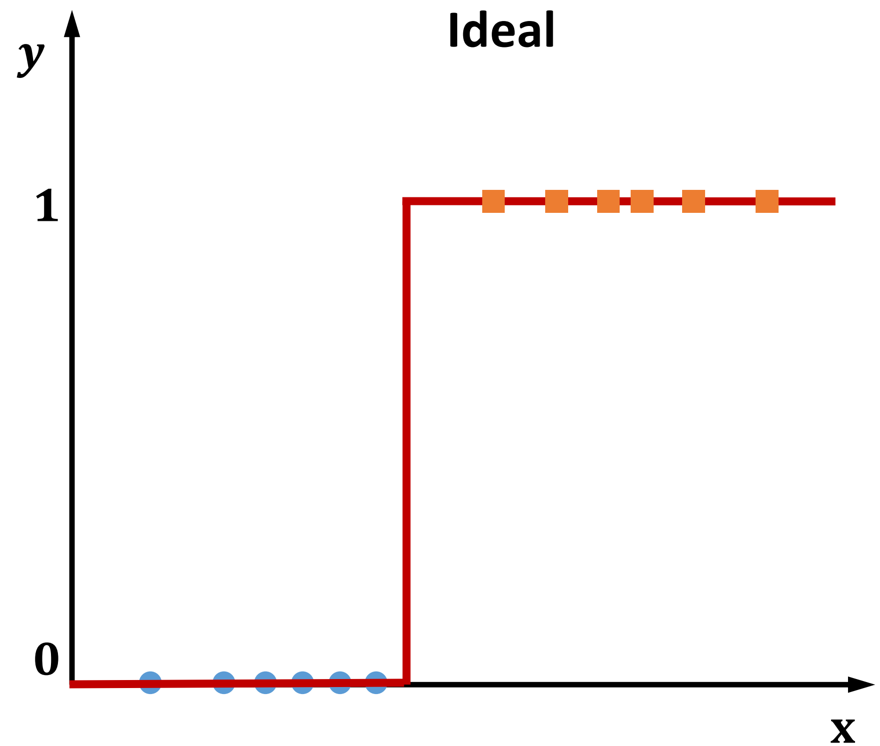
$$CD = 0.0102 \times Age - 0.0755$$



Classification Problem

- For binary classification, the output is either 0 or 1.





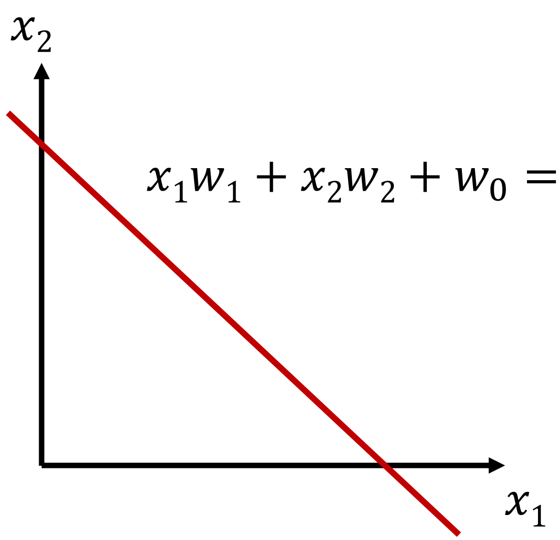
Simple Classification

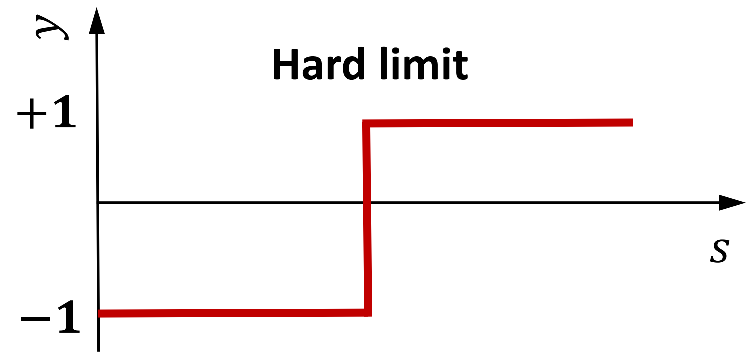
Finding a Linear Decision Boundary

- **Linear combination of input x:**

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{i=0}^{d} w_i x_i$$

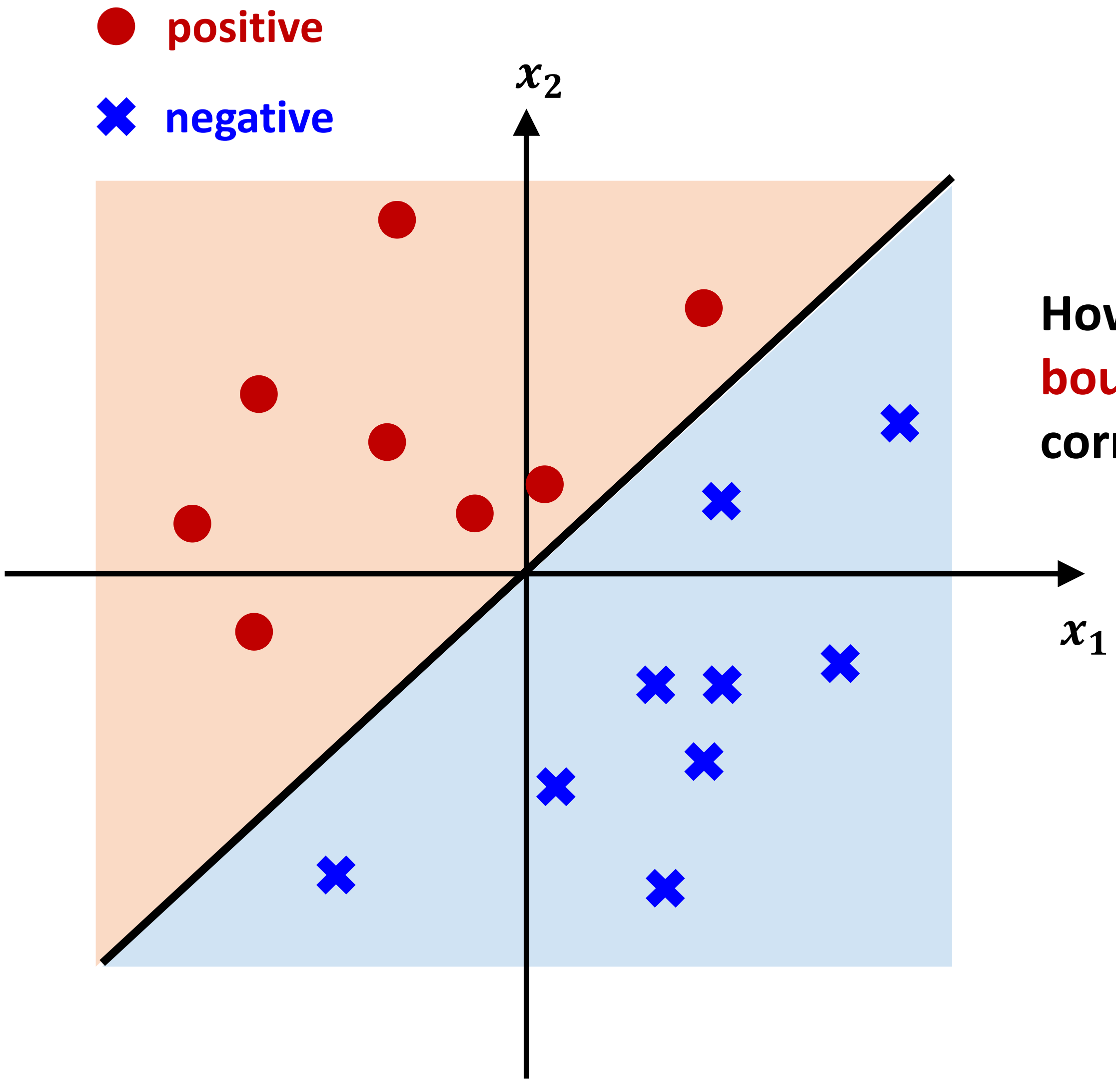
- **Nonlinear transformation of s:**





$$f(s) = \begin{cases} +1 & if\ s \geq 0 \\ -1 & otherwise \end{cases}$$

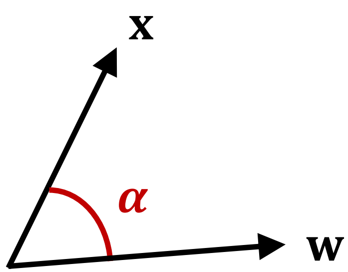
Finding a Linear Decision Boundary



How do we find a **linear decision boundary** that classifies data correctly?

Geometric Relation of Two Vectors

- Calculating the angle between two vectors



$$cos\alpha = \frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|\,\|\mathbf{x}\|} \Rightarrow cos\alpha \propto \mathbf{w}^T\mathbf{x}$$

- The angle is proportional to the inner product.

- If $\mathbf{w}^T\mathbf{x} > 0 \Rightarrow cos\alpha > 0 \Rightarrow \alpha < 90$
- If $\mathbf{w}^T\mathbf{x} < 0 \Rightarrow cos\alpha < 0 \Rightarrow \alpha > 90$

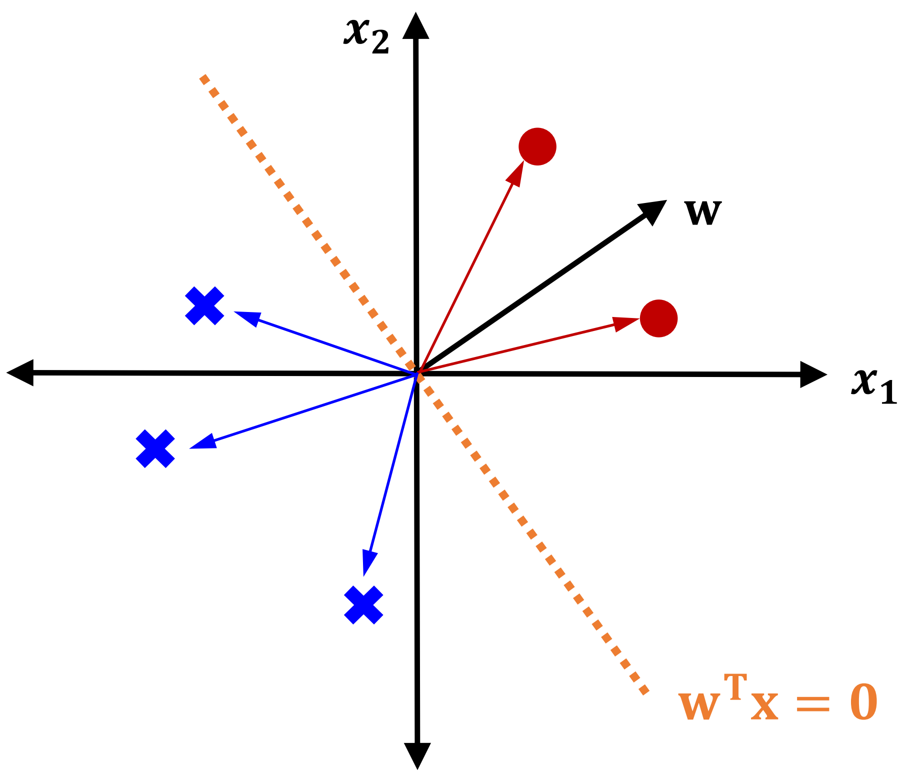
Geometric Relation of Two Vectors

- **Ideally, the weight vector should be like this:**
  - For **positive samples**, an angle is **less than 90 degrees**.
  - For **negative samples**, an angle with **more than 90 degrees**.

$$\boldsymbol{cos\alpha} = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|\,\|\mathbf{x}\|}$$

$$\Downarrow$$

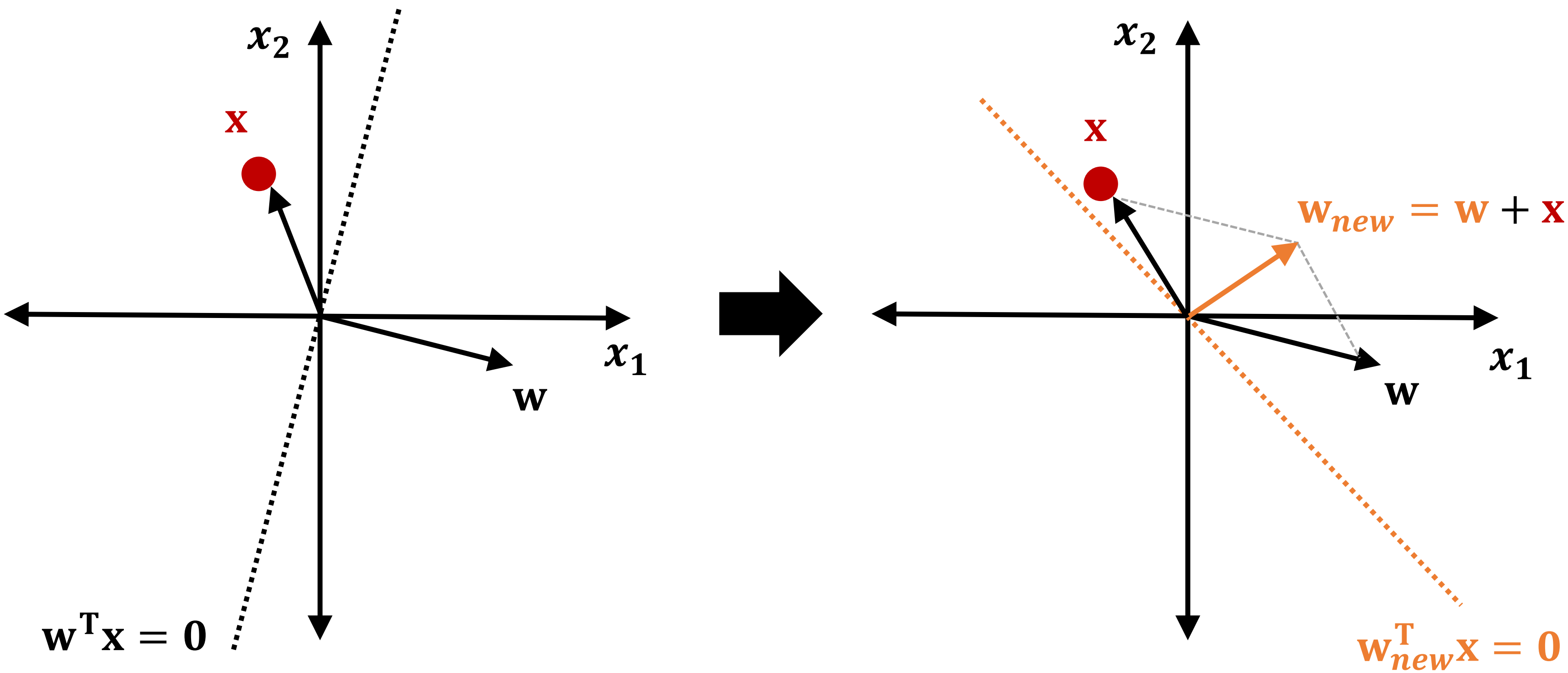
$$\boldsymbol{cos\alpha} \propto \mathbf{w}^{\mathrm{T}}\mathbf{x}$$



Case 1: How to Adjust an Angle

- When $\mathbf{x}$ belongs to the **positive sample** and $\mathbf{w}^{\mathrm{T}}\mathbf{x} < 0$,
- We need to increase the $cos\ \alpha$ value.

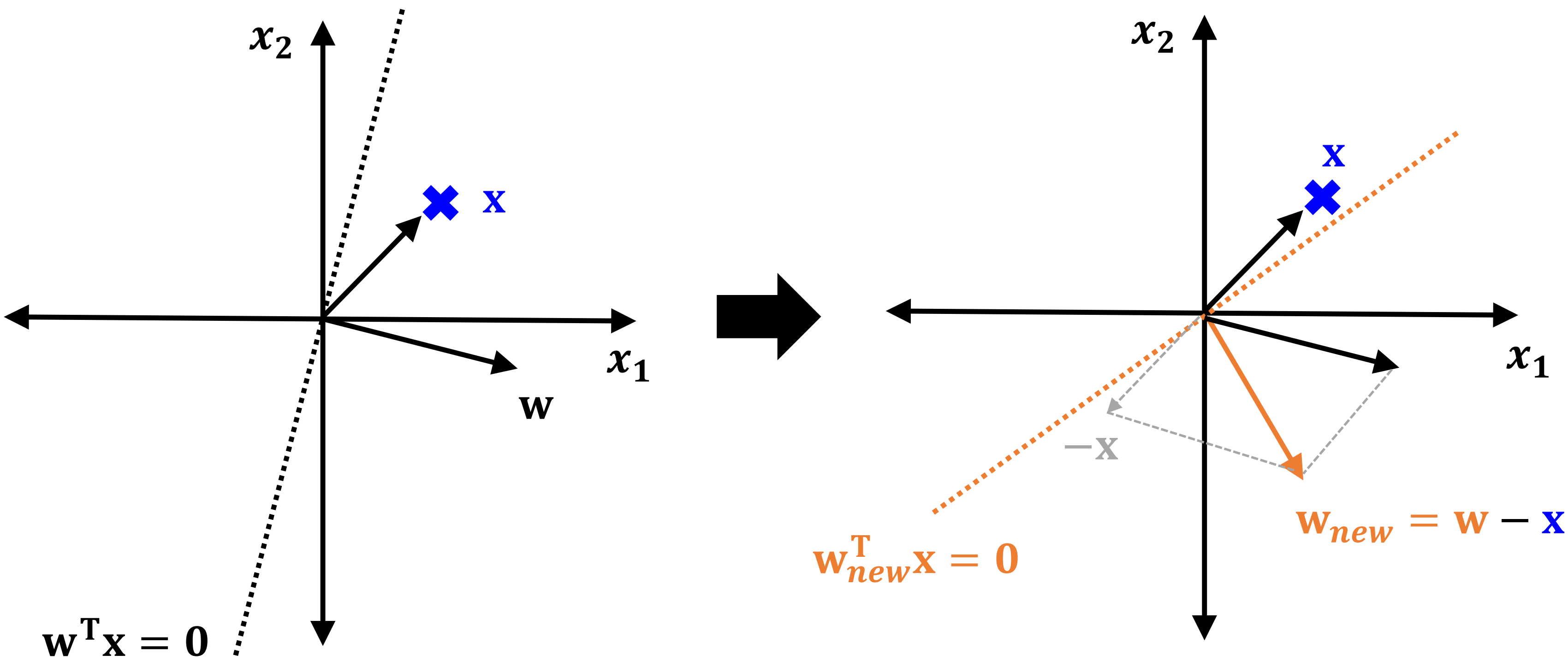
  $\Longrightarrow$ we need to **decrease the α value.**



Case 2: How to Adjust an Angle

- When $\mathbf{x}$ belongs to the negative sample and $\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$,
- We need to decrease the $cos\ \alpha$ value.

  $\Longrightarrow$ we need to increase the α value.



Summary: How to Adjust an Angle

- **When $\mathbf{w}_{new} = \mathbf{w} + \mathbf{x}$, the angle is α_{new}.**
  - $\cos\alpha_{new} \propto \mathbf{w}_{new}^{\mathbf{T}}\mathbf{x} = (\mathbf{w}+\mathbf{x})^{\mathbf{T}}\mathbf{x} = \mathbf{w}^{\mathbf{T}}\mathbf{x} + \mathbf{x}^{\mathbf{T}}\mathbf{x}$
  - Because $\mathbf{x}^{\mathbf{T}}\mathbf{x} > 0$, $cos\alpha_{new} > \cos\alpha$.

$\Longrightarrow$ **increasing the cosα value, i.e., decreasing the α value**

- **When $\mathbf{w}_{new} = \mathbf{w} - \mathbf{x}$, the angle is α_{new}.**
  - $\cos\alpha_{new} \propto \mathbf{w}_{new}^{\mathbf{T}}\mathbf{x} = (\mathbf{w}-\mathbf{x})^{\mathbf{T}}\mathbf{x} = \mathbf{w}^{\mathbf{T}}\mathbf{x} - \mathbf{x}^{\mathbf{T}}\mathbf{x}$
  - Because $\mathbf{x}^{\mathbf{T}}\mathbf{x} > 0$, $cos\alpha_{new} < \cos\alpha$.

$\Longrightarrow$ **decresing the cosα value, i.e., increasing the α value**

Learning a Linear Classifier

- **Execute the Perceptron Learning Algorithm (PLA) until not encountering mistakes.**

Randomly choose an initial solution $\mathbf{w}^{\mathbf{0}}$.

For $t = 0, 1, \ldots$

Find a mistake sample $(\mathbf{x}^{(i)}, y^{(i)})$ of $\mathbf{w}^{\mathbf{t}}$

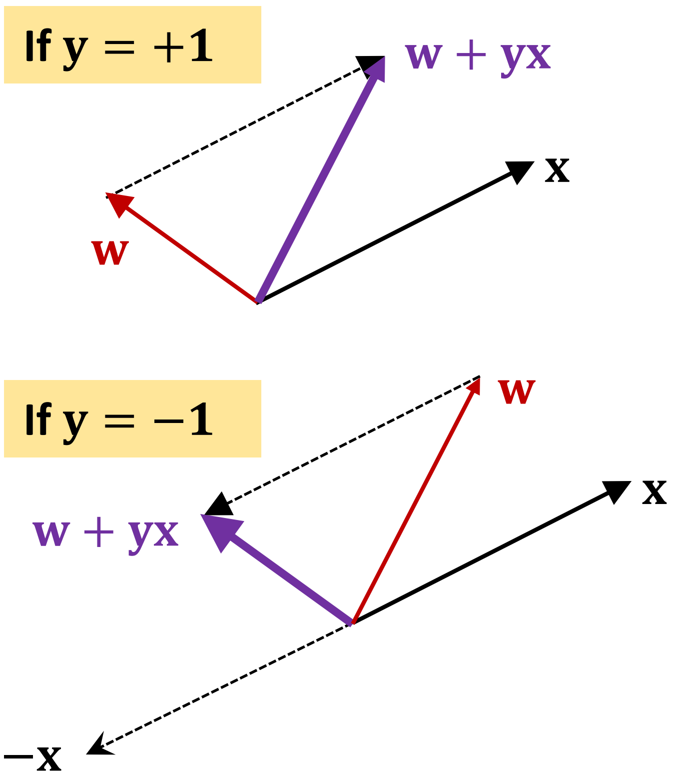
$$\mathrm{sign}\left(\mathbf{w}^{\mathbf{T}}\mathbf{x}^{(i)}\right) \neq y^{(i)}$$

Correct the mistake by

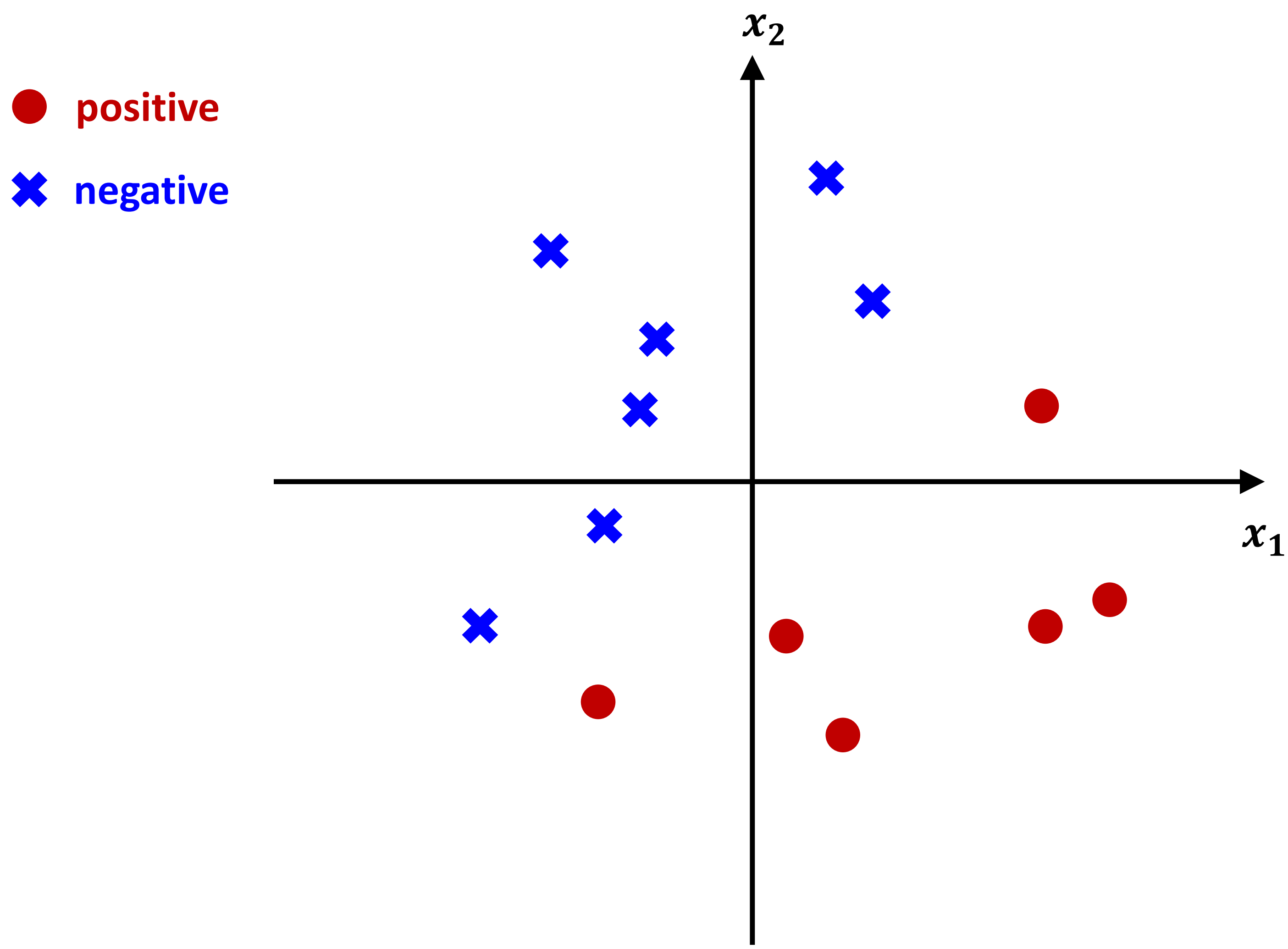
$$\mathbf{w}^{\boldsymbol{t}+\mathbf{1}} = \mathbf{w}^{\boldsymbol{t}} + y^{(i)}\mathbf{x}^{(i)}$$

Until no more mistake is found

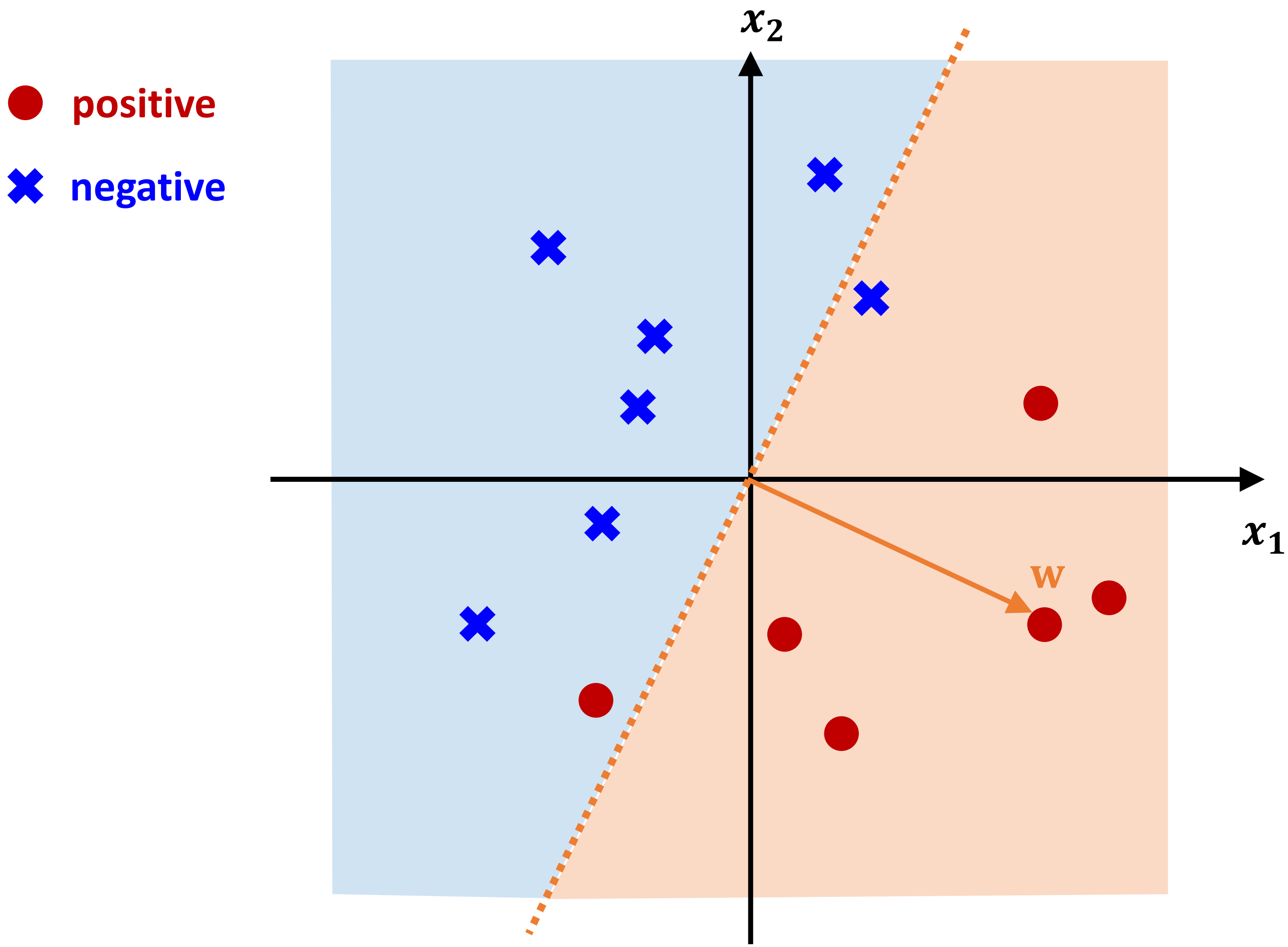
Return last $\mathbf{w}^{\boldsymbol{t}}$ as the learned model.



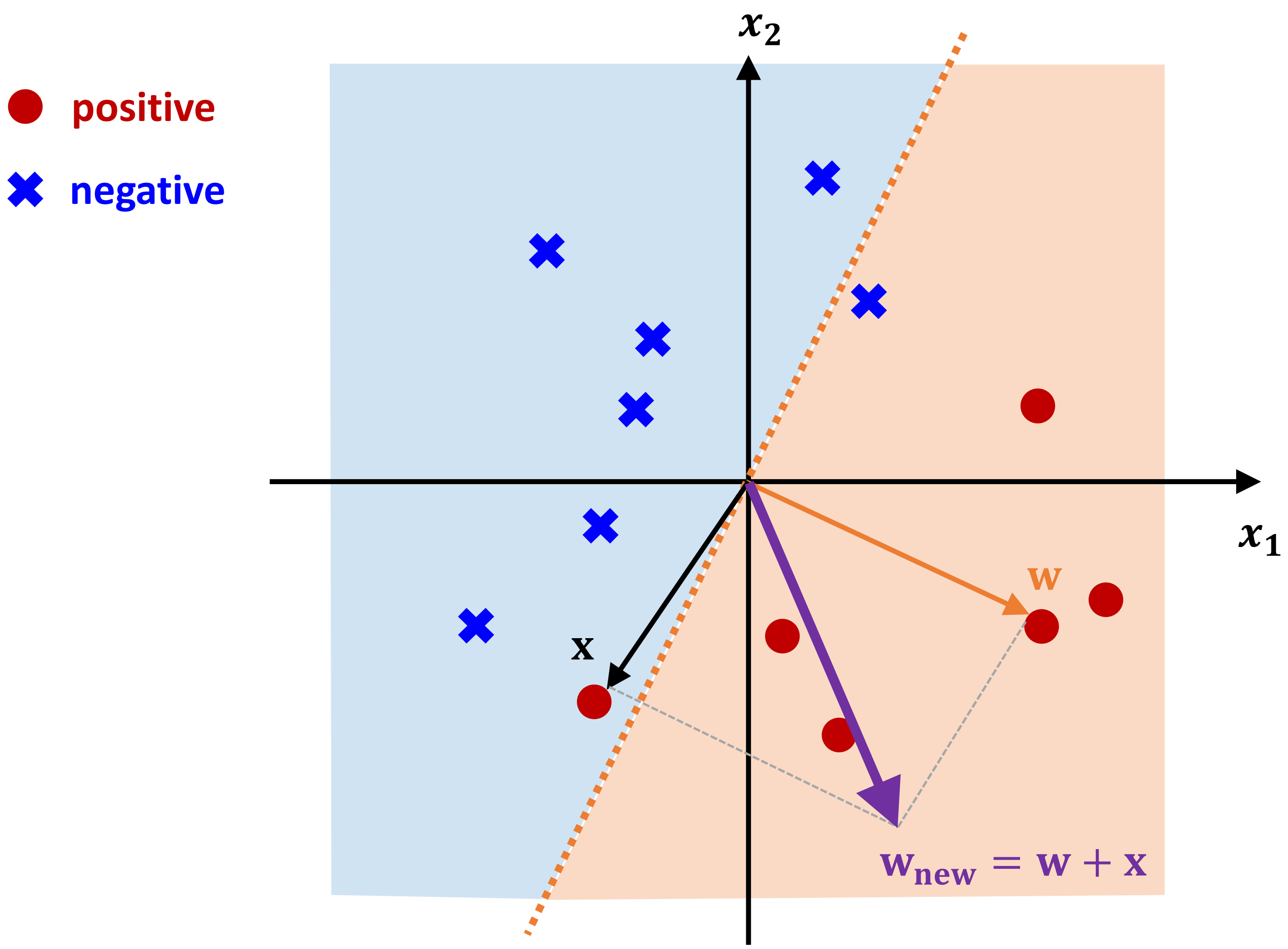
Example: Learning a Linear Classifier



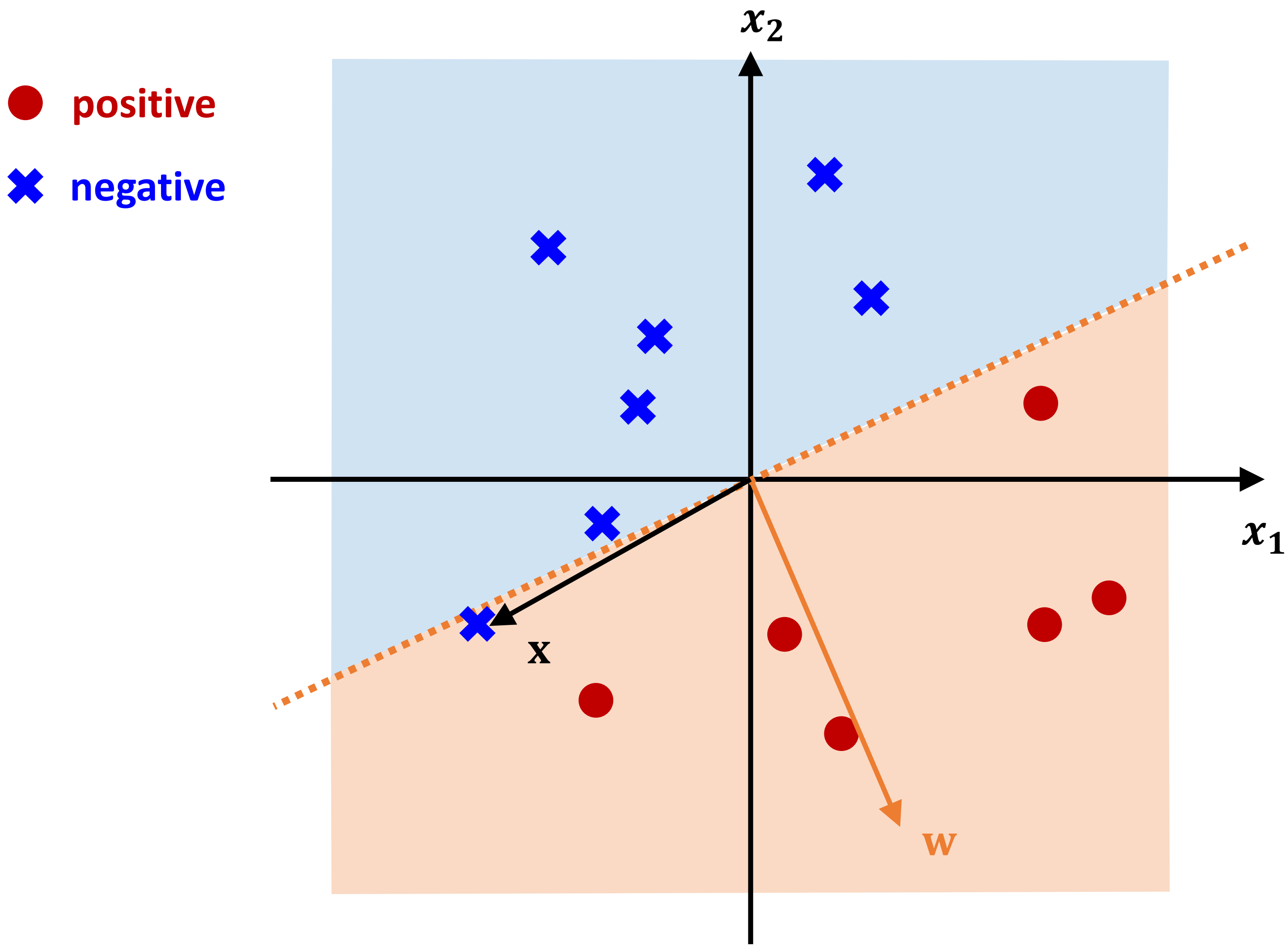
Example: Learning a Linear Classifier



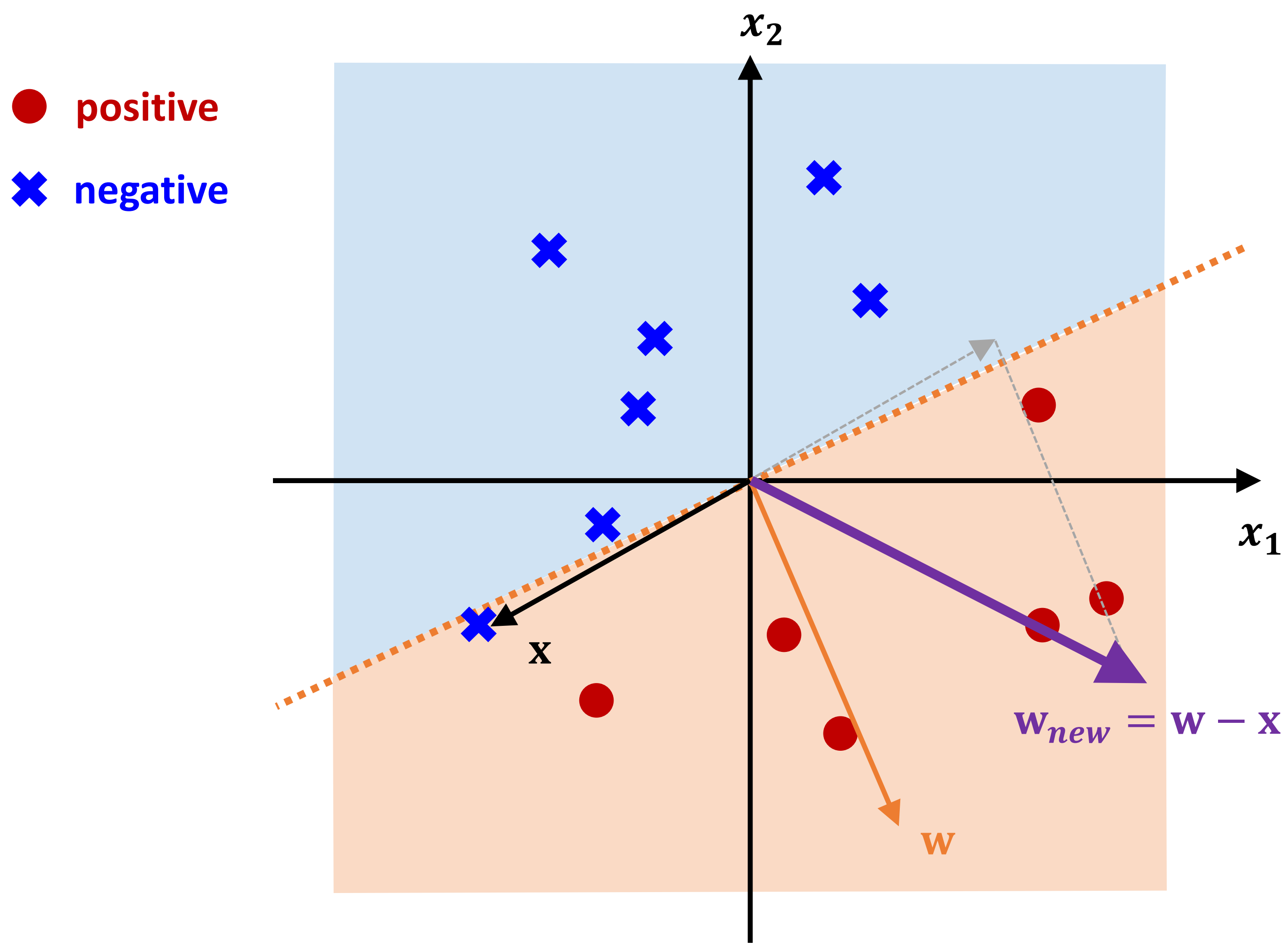
Example: Learning a Linear Classifier



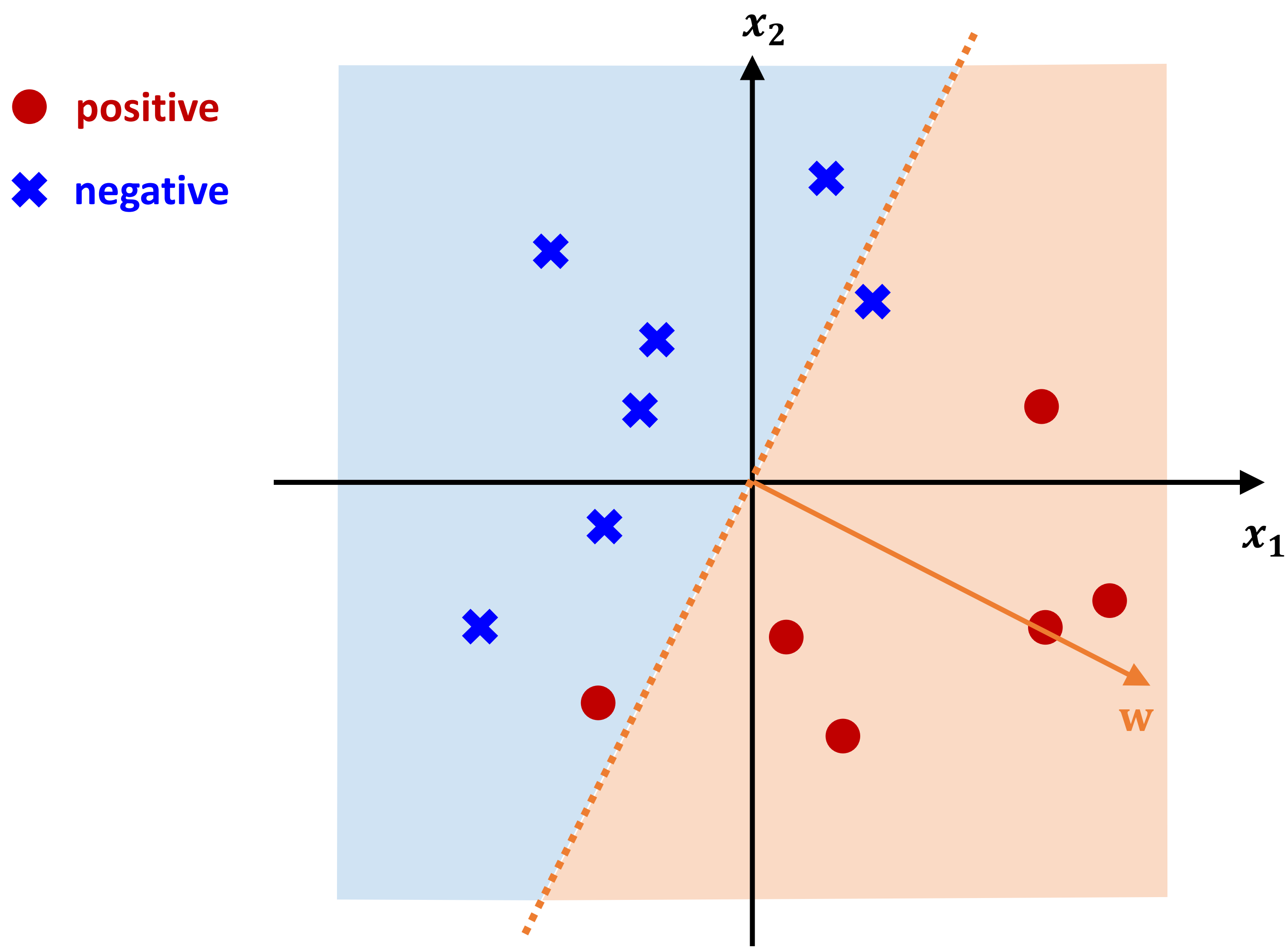
Example: Learning a Linear Classifier



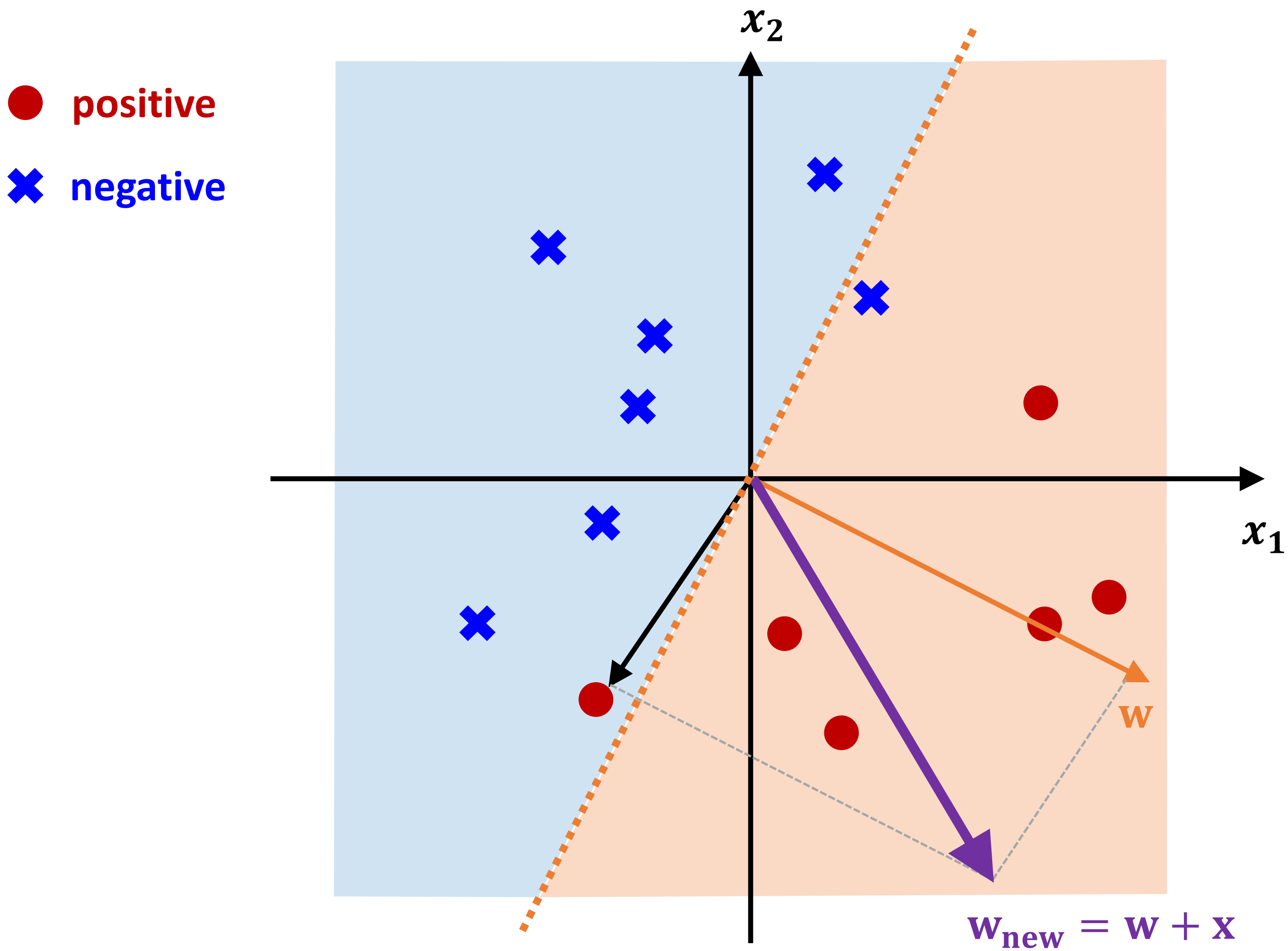
Example: Learning a Linear Classifier



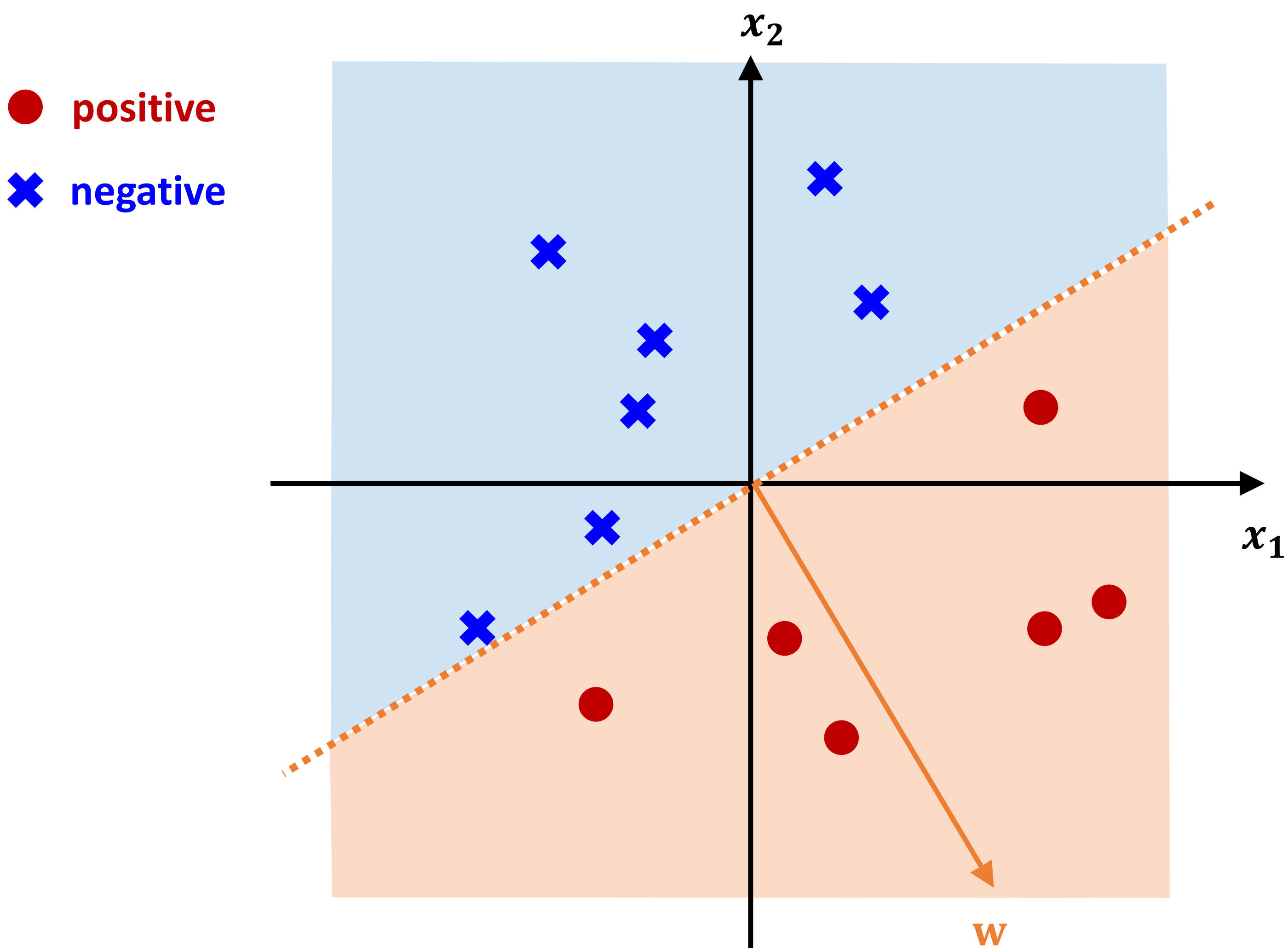
Example: Learning a Linear Classifier



Example: Learning a Linear Classifier

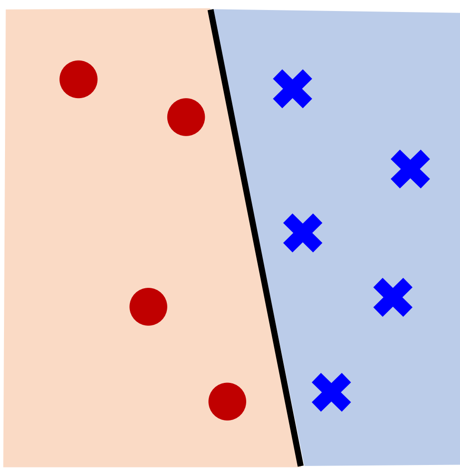


Example: Learning a Linear Classifier



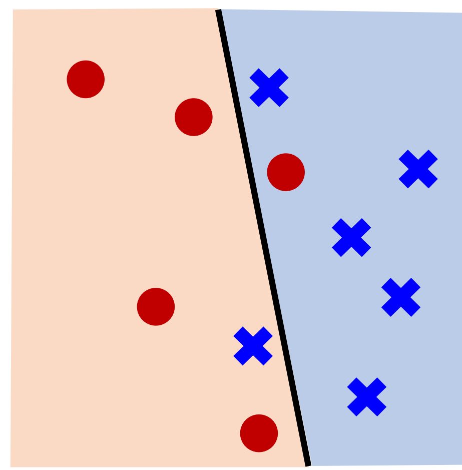
Linear Separability

- **If PLA halts (i.e., no more mistakes),**
  - **(necessary condition)** $\boldsymbol{D}$ allows some $\mathbf{w}$ to make no mistake.
- **Call such D linearly separable.**



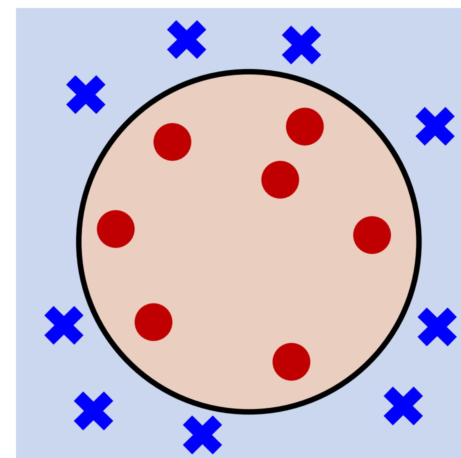
Linear separable

Good!



Linear non-separable

Need a linear model that allows some errors.

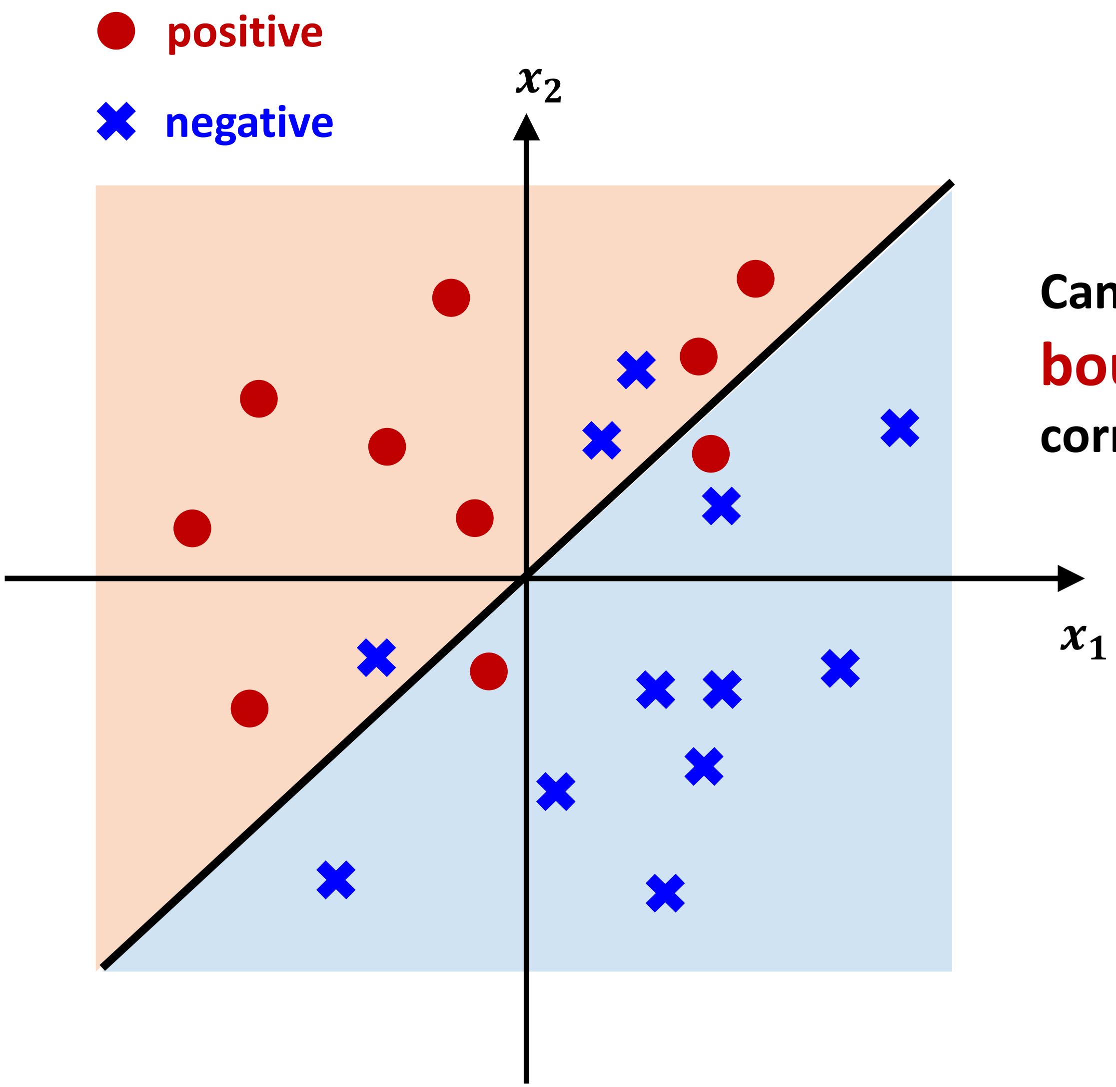


Linear non-separable

Need a non-linear model.

Logistic Regression Basics

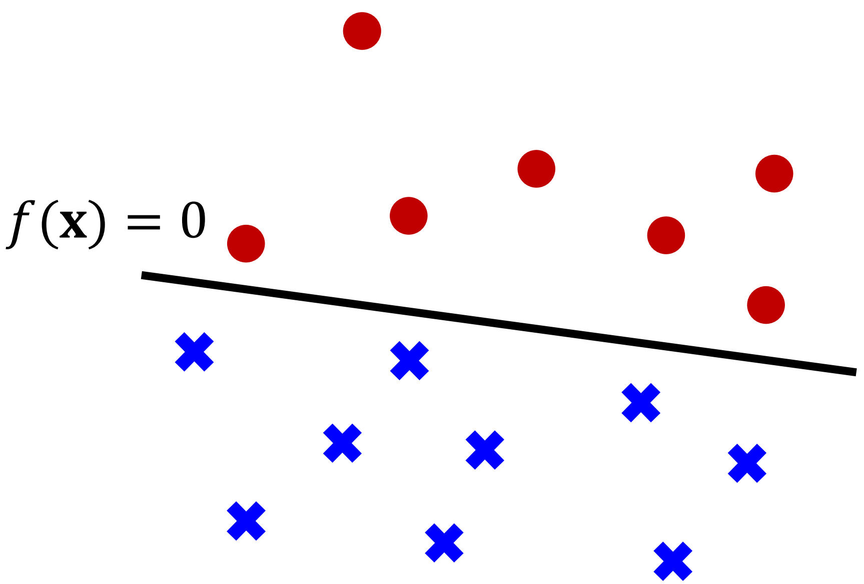
Learning a Linear Classifier

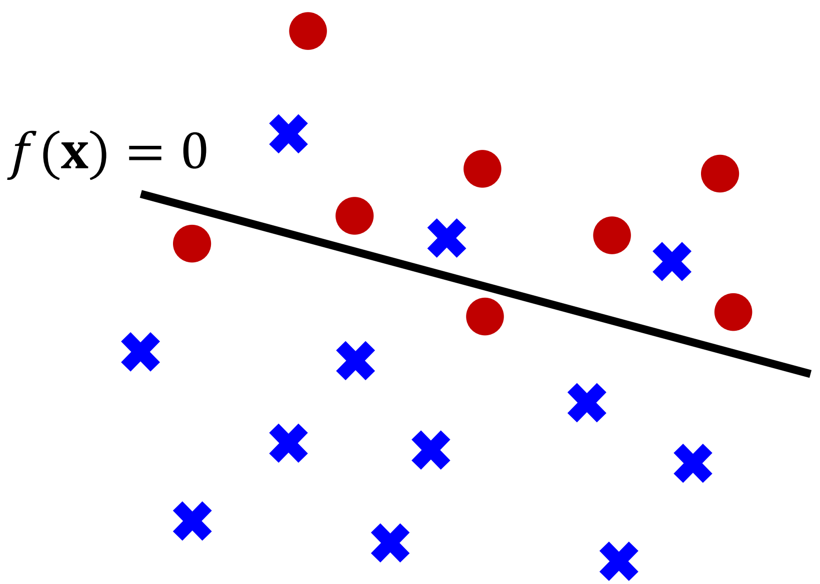


Can we find a **linear decision boundary** that classifies data correctly?

Probabilistic View for a Linear Classifier

- $h(\mathbf{x})$ **can be interpreted as the probability of "being red."**
  - As $\mathbf{x}$ goes upward from $f(\mathbf{x})$, $\mathbf{x}$ is more likely to be 1 (**Red**).
  - As $\mathbf{x}$ goes downward from $f(\mathbf{x})$, $\mathbf{x}$ is more likely to be 0 (**Blue**).

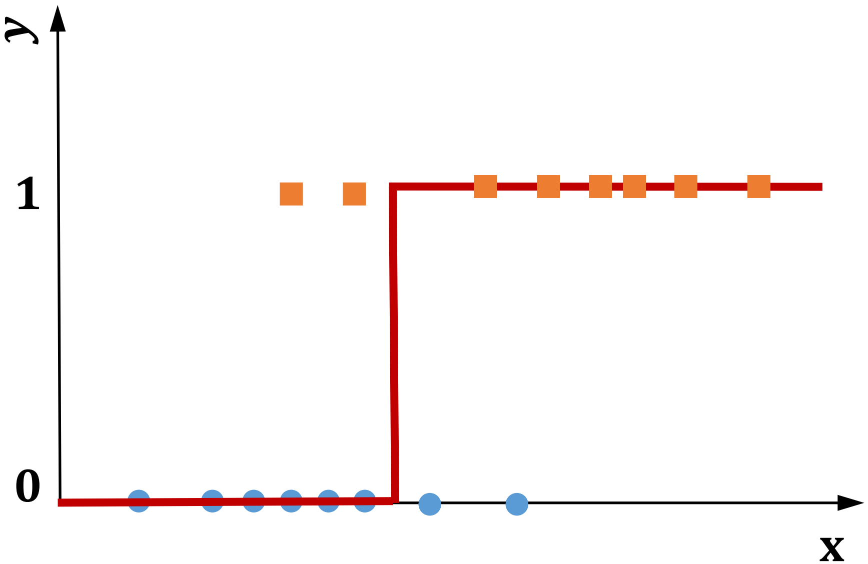


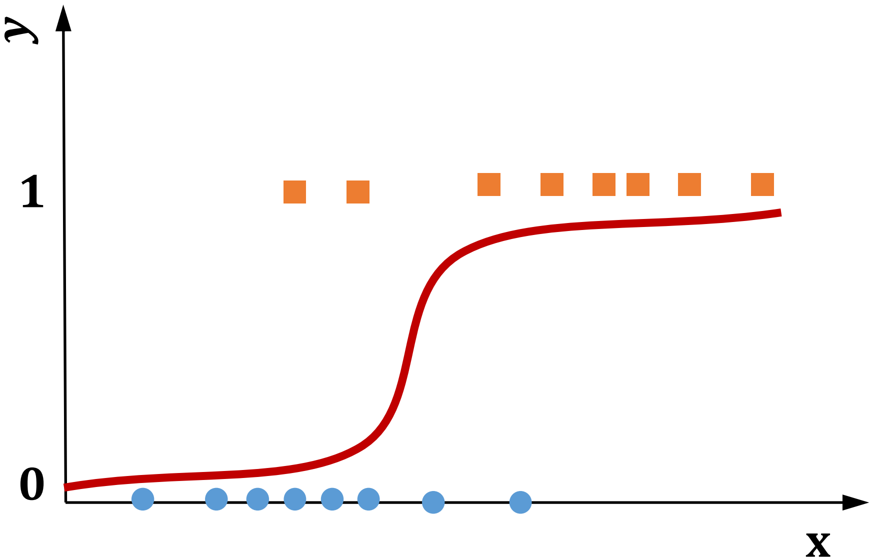


Probabilistic View for a Linear Classifier

- **What if we consider the output as $P(y = 1|\, \mathrm{x})$?**
  - As $h(\mathbf{x})$ is close to 1, $\mathbf{x}$ is more likely to be 1 (**Red**).
  - As $h(\mathbf{x})$ is close to 0, $\mathbf{x}$ is more likely to be 0 (**Blue**).

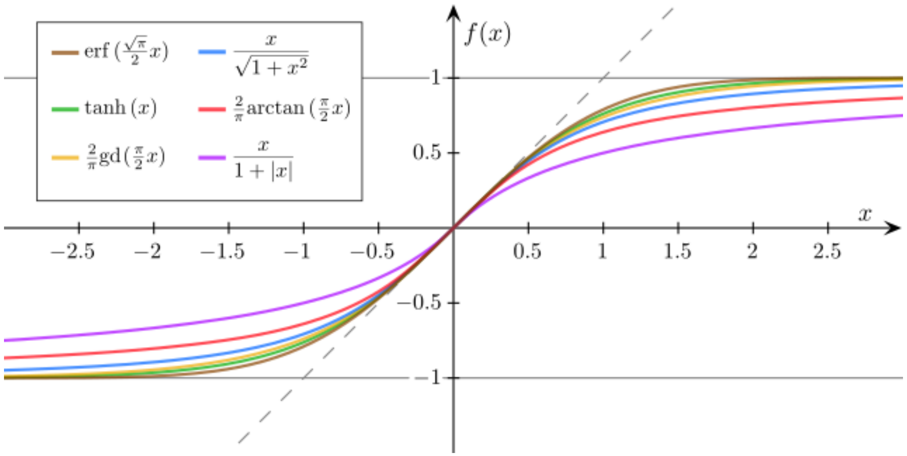
$$h(\mathbf{x}) = \begin{cases} 1\ (\textbf{Red}) & \text{if } f(\mathbf{x}) \geq 0 \\ 0\ (\textbf{Blue}) & \text{otherwise} \end{cases} \quad \Rightarrow \quad h(\mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))}$$





What is the Sigmoid Function?

- **The sigmoid function is an S-curve shape.**
  - Bounded
  - Differential
  - Defined for all real inputs
  - With a positive derivative



- **Logistic function**

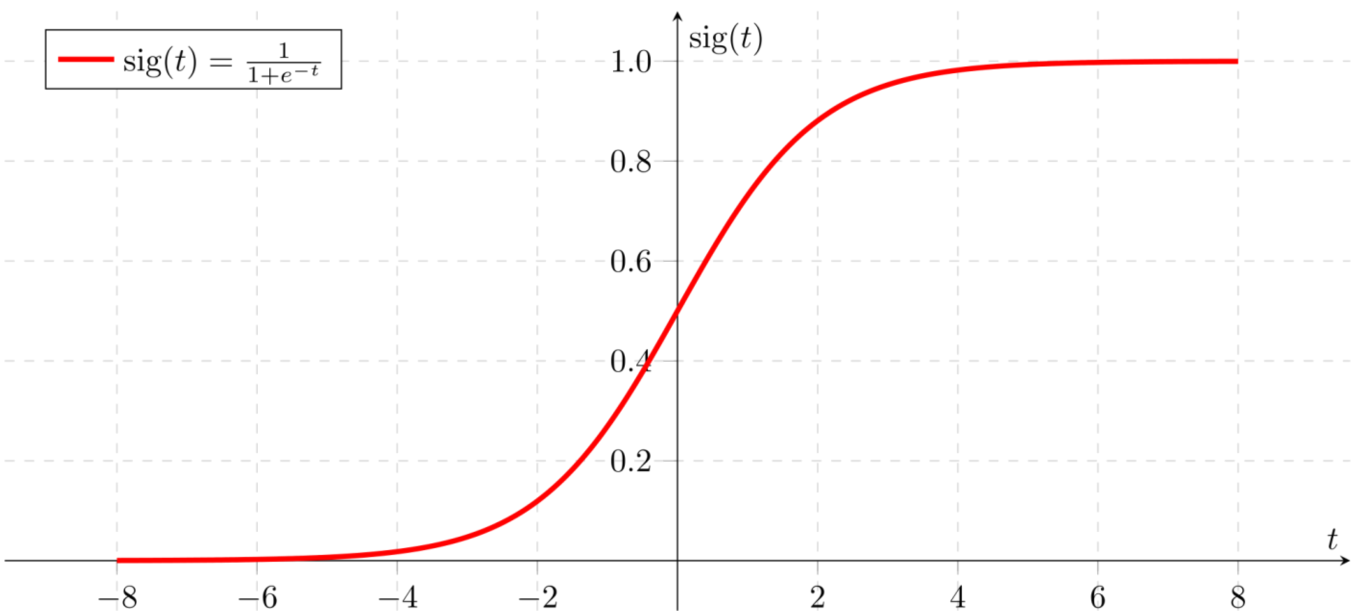
$$\sigma(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

  - x_0: the midpoint of the x-value
  - L: the curve's maximum value
  - k: the steepness of the curve

Logistic Function

- As the input of the logistic function, $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ is used.

$$h(\mathbf{x}) = g(f(\mathbf{x})) = \sigma\big(f(\mathbf{x})\big) = \frac{1}{1+e^{-f(\mathbf{x})}} = \frac{1}{1+e^{-(\mathbf{w}^{\mathrm{T}}\mathbf{x})}}$$



Formulating Binary Classification

➢ **Use Bayes' rule to calculate the relevant posterior probability.**

$$P(y=1\mid\mathbf{x})=\frac{P(\mathbf{x}\mid y=1)P(y=1)}{P(\mathbf{x})}$$

$$=\frac{P(\mathbf{x}\mid y=1)P(y=1)}{P(x\mid y=1)P(\mathrm{y}=1)+P(x\mid y=0)P(\mathrm{y}=0)}$$

$$=\frac{1}{1+\dfrac{P(\mathbf{x}\mid y=0)P(\mathrm{y}=0)}{P(\mathbf{x}\mid y=1)P(\mathrm{y}=1)}}$$

$$=\frac{1}{1+\exp\left\{\ln\dfrac{P(\mathbf{x}\mid y=0)P(\mathrm{y}=0)}{P(\mathbf{x}\mid y=1)P(\mathrm{y}=1)}\right\}}\qquad \exp\{\ln a\}=a$$

$$=\frac{1}{1+\exp\left\{-\ln\dfrac{P(\mathbf{x}\mid y=1)}{P(\mathbf{x}\mid y=0)}-\ln\dfrac{P(y=1)}{P(y=0)}\right\}}$$

Formulating Binary Classification

- It is the form of the logistic function.

$$P(y=1\mid \mathbf{x}) = \frac{1}{1+\exp\{-z\}}$$

$$\text{where} \quad z = \ln\frac{P(\mathbf{x}\mid y=1)}{P(\mathbf{x}\mid y=0)} + \ln\frac{P(y=1)}{P(y=0)} \propto \ln\frac{P(y=1\mid \mathbf{x})}{P(y=0\mid \mathbf{x})}$$

Likelihood ratio   Prior ratio

- We simply design it as a linear model.

What are Odds?

- Instead of the probability, we introduce the **odds**.

- It is defined as the probability that the event will occur divided by the probability that the event will not occur.

$$odds = \frac{P(y=1 \mid \mathbf{x})}{P(y=0 \mid \mathbf{x})} = \frac{P(y=1 \mid \mathbf{x})}{1 - P(y=1 \mid \mathbf{x})}$$

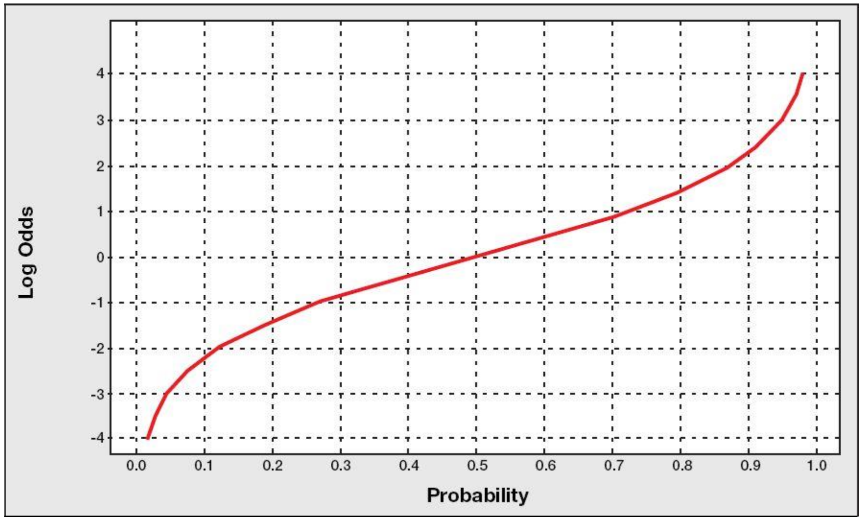
- For binary classification, evaluating the odds is also okay.
  - If $P(y=1 \mid \mathbf{x}) > P(y=0 \mid \mathbf{x})$, then x is likely to be 1.
  - If $P(y=1 \mid \mathbf{x}) < P(y=0 \mid \mathbf{x})$, then x is likely to be 0.

Applying Log Odds (Logit) to $f(\mathbf{x})$

- What if we represent $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ as $\ln \frac{P(y=1\mid\mathbf{x})}{1-P(y=1\mid\mathbf{x})}$?

$$\ln \frac{P(y=1\mid\mathbf{x})}{1-P(y=1\mid\mathbf{x})} = w_0x_0 + w_1x_1 + w_2x_2 + \ldots + w_dx_d$$

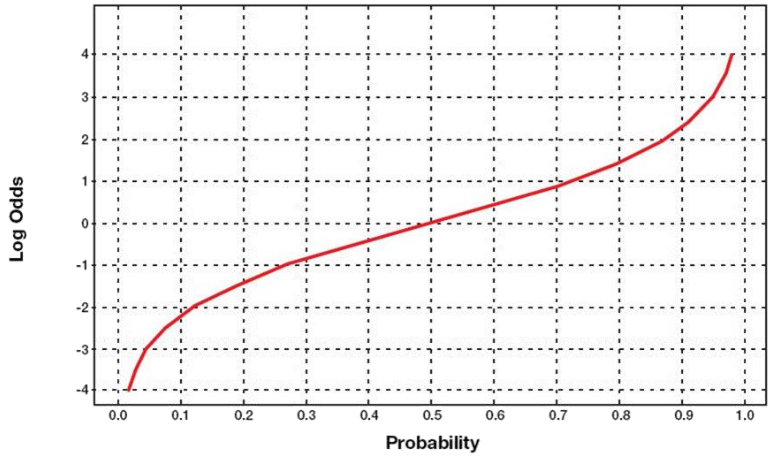
- The logarithm of the odds
  - $-\infty < \ln(odds) < \infty$
  - Symmetric



Formulating the Logistic Function

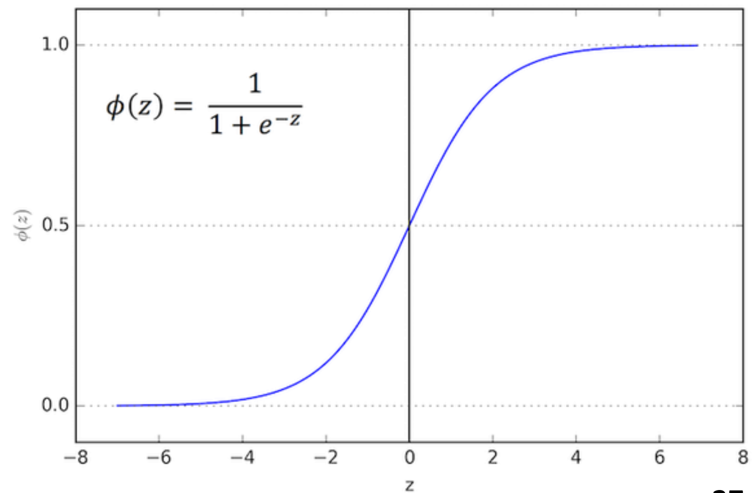
- **Mapping the linear equation to the log odds**

$$\ln(odds) = \ln\left(\frac{p}{1-p}\right) = f(\mathbf{x})$$



- **Taking the exponent for both sides**

$$odds = \frac{p}{1-p} = e^{f(\mathbf{x})}$$



- **The probability of $y = 1$ given $\mathbf{x}$ is**

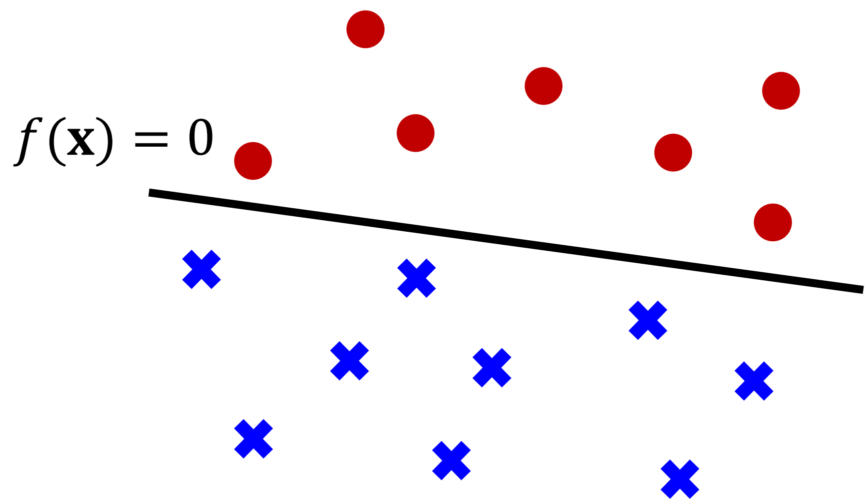
$$P(y = 1 \mid \mathbf{x}) = \frac{e^{f(\mathbf{x})}}{1 + e^{f(\mathbf{x})}} = \frac{1}{1 + e^{-(f(\mathbf{x}))}}$$

Making a Linear Classifier

- $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ is used as a linear decision boundary.

$$\ln(odds) = \ln\left(\frac{P(y=1\,|\,\mathbf{x})}{1-P(y=1\,|\,\mathbf{x})}\right) = \ln\left(\frac{P(y=1\,|\,\mathbf{x})}{P(y=0\,|\,\mathbf{x})}\right) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

- If $\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0$,
  - $P(y = 1\,|\,\mathbf{x}) > P(y = 0\,|\,\mathbf{x})$
- If $\mathbf{w}^{\mathrm{T}}\mathbf{x} < 0$,
  - $P(y = 1\,|\,\mathbf{x}) < P(y = 0\,|\,\mathbf{x})$



The line on the decision boundary is $\mathbf{w}^{\mathrm{T}}\mathbf{x} = \mathbf{0}$.

Formulating Logistic Regression

Formulating Logistic Regression

- **Given** $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}): 1 \le i \le n\}$
  - $\mathbf{x}^{(i)} = (1, x_{i1}, \dots, x_{id}), y^{(i)} \in \{0, 1\}$

- **Finding** $h(\mathbf{x}^{(i)}) = \sigma\left(f(\mathbf{x}^{(i)})\right)$ **that minimizes** $E(\mathbf{w})$

$$E(\mathbf{w}) = \sum_{i=1}^{n} \left(\sigma\left(f(\mathbf{x}^{(i)})\right) - y^{(i)}\right)^2, \text{where } \sigma\left(f(\mathbf{x}^{(i)})\right) = \frac{1}{1 + e^{-(f(\mathbf{x}^{(i)}))}}$$

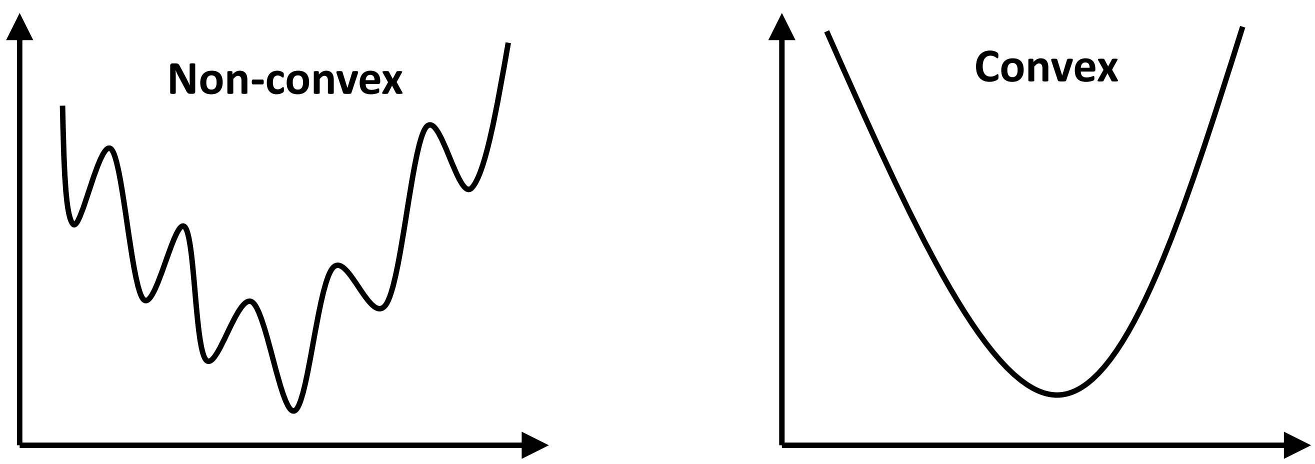
- **How?**
  - Using the gradient descent method

Training Logistic Regression

- Simply, use the error function used in linear regression!

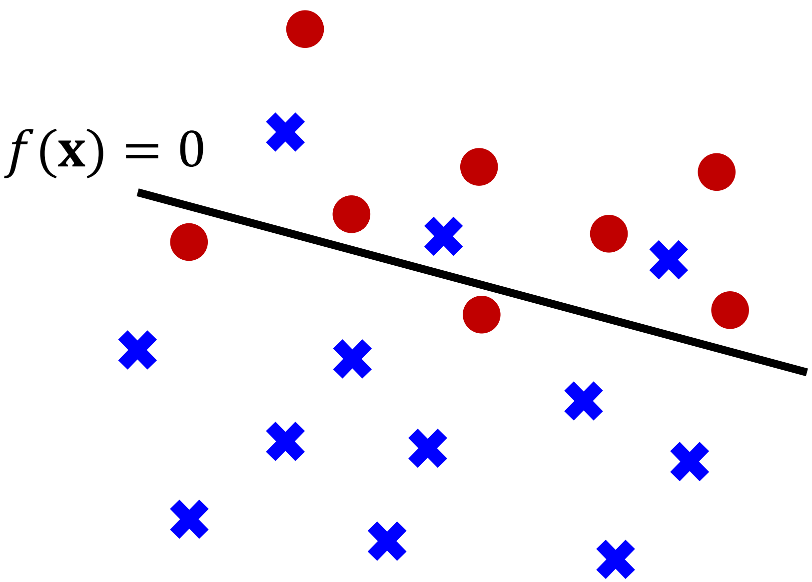
$$E(\mathbf{w}) = \sum_{i=1}^{n} \left(y^{(i)} - h(\mathbf{x}^{(i)})\right)^2$$

- This gives the **non-convex function** for w, which does not guarantee the global minimum.



Probabilistic View: Linear Classifier

- $h(\mathbf{x})$ **can be interpreted as the probability of "being red."**
  - As $\mathbf{x}$ goes upward from $f(\mathbf{x})$, $\mathbf{x}$ is more likely to be 1 (**Red**).
  - As $\mathbf{x}$ goes downward from $f(\mathbf{x})$, $\mathbf{x}$ is more likely to be 0 (**Blue**).



$$h(\mathbf{x}) = \frac{1}{1 + \exp(-f(\mathbf{x}))}$$

$$P(y|\mathbf{x}, \mathbf{w}) = \begin{cases} h(\mathbf{x}) & \text{if } \boldsymbol{y} = \mathbf{1} \\ 1 - h(\mathbf{x}) & \text{if } \boldsymbol{y} = \mathbf{0} \end{cases}$$

Recap: Maximum Likelihood Estimation

- Estimate the **maximum likelihood** given **independent** observations $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$.

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f\left(\mathbf{x}^{(i)} \mid \theta\right)$$

- What θ **maximizes the likelihood** of the observed data?

$$\frac{\partial}{\partial \theta} \mathcal{L}(\theta) = 0$$

Formulating the Error Function

- **Find a boundary which makes**
  - **Positive samples** are likely to be $P(y = 1 \mid \mathbf{x}, \mathbf{w})$.
  - **Negative samples** are likely to be $P(y = 0 \mid \mathbf{x}, \mathbf{w})$.

- **Find w that maximizes**

$$\underset{\mathbf{w}}{\operatorname{argmax}} \left(\prod_{\{\mathbf{x} \mid y=1\}} P(y = 1 \mid \mathbf{x}, \mathbf{w}) \right) \left(\prod_{\{\mathbf{x} \mid y=0\}} P(y = 0 \mid \mathbf{x}, \mathbf{w}) \right)$$

Positive samples **Negative samples**

Formulating the Error Function

$$\underset{\mathbf{w}}{\operatorname{argmax}} \left(\prod_{\{\mathbf{x} \mid y=1\}} P(y = 1 \mid \mathbf{x}, \mathbf{w}) \right) \left(\prod_{\{\mathbf{x} \mid y=0\}} P(y = 0 \mid \mathbf{x}, \mathbf{w}) \right)$$

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \, \mathbf{ln} \left(\prod_{\{\mathbf{x} \mid y=1\}} P(y = 1 \mid \mathbf{x}, \mathbf{w}) \right) \left(\prod_{\{\mathbf{x} \mid y=0\}} P(y = 0 \mid \mathbf{x}, \mathbf{w}) \right)$$

Note: The log function is monotonic.

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{\{\mathbf{x} \mid y=1\}} \ln P(y = 1 \mid \mathbf{x}, \mathbf{w}) + \sum_{\{\mathbf{x} \mid y=0\}} \ln(1 - P(y = 1 \mid \mathbf{x}, \mathbf{w}))$$

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{\{\mathbf{x} \mid y=1\}} \ln h(\mathbf{x}) + \sum_{\{\mathbf{x} \mid y=0\}} \ln(1 - h(\mathbf{x})) \text{ where } h(\mathbf{x}) = \frac{1}{1 + e^{-(f(\mathbf{x}))}}$$

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + \left(1 - y^{(i)}\right) \ln\left(1 - h\left(\mathbf{x}^{(i)}\right)\right) \text{ where } h\left(\mathbf{x}^{(i)}\right) = \frac{1}{1 + e^{-\left(f\left(\mathbf{x}^{(i)}\right)\right)}}$$

Formulating the Error Function

- Finding a linear boundary $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ that minimizes the error function

$$= \underset{\mathbf{w}}{\operatorname{argmax}} \left(\sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)})) \right)$$

$$\Downarrow$$

$$= \underset{\mathbf{w}}{\operatorname{argmin}} - \left(\sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)})) \right)$$

- How to solve this?
  - A closed-form equation
  - Gradient descent method

Solving the Optimization Problem

- Bad news: there is no closed-form solution to minimize the error function.

$$E(\mathbf{w}) = -\sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)}))$$

Details: Optimization Problem

$$y \ln P(y = 1 \mid \mathbf{x}, \mathbf{w}) + (1 - y) \ln(1 - P(y = 1 \mid \mathbf{x}, \mathbf{w}))$$

Substituting $P(y = 1 \mid \mathbf{x}, \mathbf{w})$ to $h(\mathbf{x})$

$$y \ln h(\mathbf{x}) + \ln(1 - h(\mathbf{x})) - y \ln(1 - h(\mathbf{x}))$$

$$y\big(\ln h(\mathbf{x}) - \ln\big(1 - h(\mathbf{x})\big)\big) + \ln(1 - h(\mathbf{x}))$$

$$y\left(\ln \frac{P(y = 1 \mid \mathbf{x}, \mathbf{w})}{1 - P(y = 1 \mid \mathbf{x}, \mathbf{w})}\right) + \ln(1 - h(\mathbf{x}))$$

Details: Optimization Problem

$$y\left(\ln\frac{P(y=1\,|\,\mathbf{x},\mathbf{w})}{1-P(y=1\,|\,\mathbf{x},\mathbf{w})}\right)+\ln(1-h(\mathbf{x}))$$

Substituting $\mathbf{w}^{\mathbf{T}}\mathbf{x} = \ln\frac{P(y=1\,|\mathbf{x},\mathbf{w})}{1-P(y=1\,|\mathbf{x},\mathbf{w})}$

$$y\mathbf{w}^{\mathbf{T}}\mathbf{x}+\ln(1-h(\mathbf{x}))$$

- **Apply the partial derivative to find optimal w.**

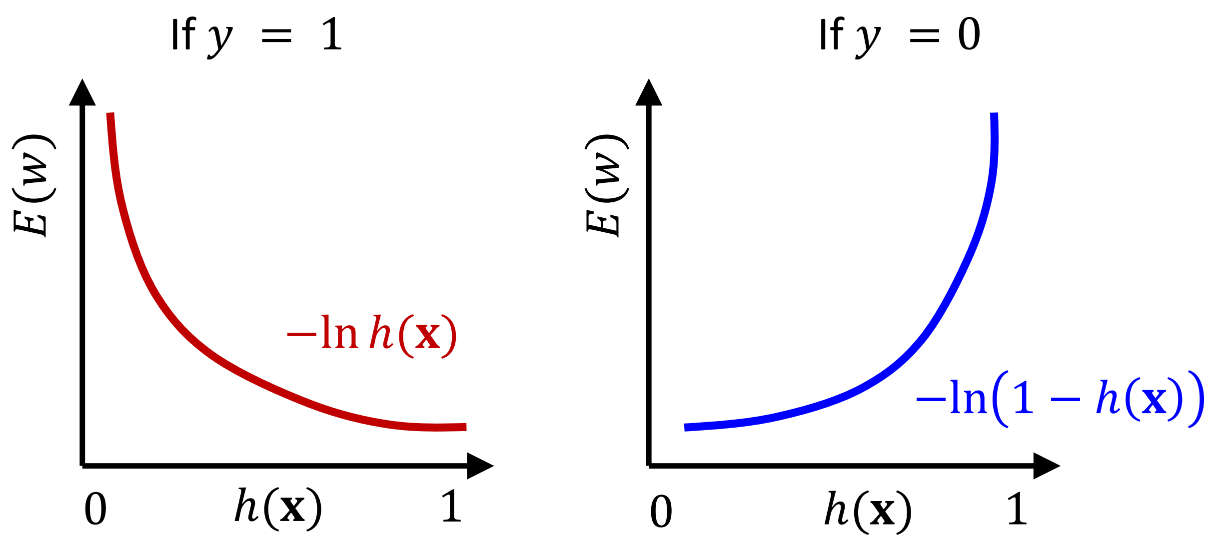
$$\frac{\partial}{\partial w_j}\left(y\mathbf{w}^{\mathbf{T}}\mathbf{x}+\ln(1-h(\mathbf{x}))\right)=0$$

We cannot solve this problem.

Solving the Optimization Problem

- **Good news: the error function is convex.**
  - Unique maximum: The convex function is easy to optimize.

$$E(\mathbf{w}) = -\sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)}))$$

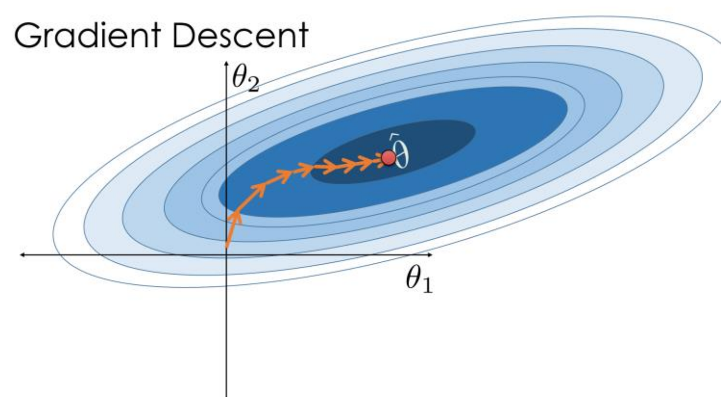


Training Logistic Regression

Recap: Gradient Descent (GD)

- **Simple concept: follow the gradient *downhill***
- **Process:**
  1. Pick a starting position: $\mathbf{w}^0 = (w_0, w_1, w_2, \dots, w_d)$
  2. Determine the descent direction: $\Delta\mathbf{w} = \nabla E(\mathbf{w}^t)$
  3. Choose a learning rate: $\boldsymbol{\eta}$
  4. Update your position: $\mathbf{w}^{t+1} = \mathbf{w}^t - \boldsymbol{\eta}\Delta\mathbf{w}$
  5. Repeat from 2) until stopping criterion is satisfied.

- **Key issues in GD**
  - **How to compute Δw?**
    - **Batch size in $\mathcal{D}$**
  - **How to determine η?**



Computing the Partial Derivative

- **Using the chain rule**

$$\frac{\partial E}{\partial \mathbf{w}} = \frac{\partial E}{\partial h}\frac{\partial h}{\partial f}\frac{\partial f}{\partial \mathbf{w}} \quad \textit{where } h = h\big(\boldsymbol{f}(\mathbf{x})\big) \textit{ and } f = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\frac{\partial}{\partial \mathbf{w}}\sum_{i=1}^{n}[y^{(i)}\ln h + \left(1-y^{(i)}\right)\ln(1-h)]$$

Applying the derivative of $\ln(x) = x^{-1}$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left[y^{(i)}\frac{1}{h}\frac{\partial h}{\partial \mathbf{w}} + \left(1-y^{(i)}\right)\left(-\frac{1}{1-h}\right)\frac{\partial h}{\partial \mathbf{w}}\right]$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{h} - \left(1-y^{(i)}\right)\frac{1}{1-h}\right)\frac{\partial h}{\partial \mathbf{w}}$$

Computing the Partial Derivative

- **Using the chain rule**

$$\frac{\partial E}{\partial \mathbf{w}} = \frac{\partial E}{\partial h}\frac{\partial h}{\partial f}\frac{\partial f}{\partial \mathbf{w}} \quad \textit{where } h = h\big(\boldsymbol{f}(\mathbf{x})\big) \textit{ and } f = \mathbf{w}^{\mathbf{T}}\mathbf{x}$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{h} - \left(1-y^{(i)}\right)\frac{1}{1-h}\right)\frac{\partial h}{\partial \mathbf{w}}$$

Applying the derivative of $h(f) = \frac{1}{1+e^{-f}}$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{h} - \left(1-y^{(i)}\right)\frac{1}{1-h}\right)h(1-h)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}\right)$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(\frac{y^{(i)}(1-h) - \left(1-y^{(i)}\right)h}{h(1-h)}\right)h(1-h)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}\right)$$

Details!

Derivative of Sigmoid Function

- Denote the sigmoid function as $\sigma(x) = \frac{1}{1+e^{-x}}$

$$\frac{d}{dx}\sigma(x) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{d}{dx}(1+e^{-x})^{-1} = -(1+e^{-x})^{-2}(-e^{-x})$$

$$\Downarrow$$

$$-(1+e^{-x})^{-2}(-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right)$$

$$\Downarrow$$

$$\left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{(1+e^{-x})-1}{1+e^{-x}}\right) = \left(\frac{1}{1+e^{-x}}\right)\left(1-\frac{1}{1+e^{-x}}\right)$$

$$\Downarrow$$

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

Computing the Partial Derivative

- Using the chain rule

$$\frac{\partial E}{\partial \mathbf{w}} = \frac{\partial E}{\partial h}\frac{\partial h}{\partial f}\frac{\partial f}{\partial \mathbf{w}} \quad where\ h = h\big(\boldsymbol{f}(\mathbf{x})\big)\ and\ f = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(\frac{y^{(i)}(1-h) - \left(1-y^{(i)}\right)h}{h(1-h)}\right) h(1-h)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}\right)$$

$$\Downarrow$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(y^{(i)} - h(\mathbf{x})\right)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)}\right)$$

$$\Downarrow$$

$$\frac{\partial E}{\partial \mathbf{w}} = -\sum_{i=1}^{n}\left(y^{(i)} - h(\mathbf{x})\right)\mathbf{x}^{(i)}$$

Computing the Partial Derivative

- The error function for logistic regression is

$$E(\mathbf{w}) = -\left(\sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} y^{(i)} \ln h(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \ln\left(1 - h(\mathbf{x}^{(i)})\right) \right)$$

$$h(\mathbf{x}) = \sigma(f(\mathbf{x})) = \frac{1}{1 + e^{(-f(\mathbf{x}))}} = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + \cdots + w_d x_d)}}$$

- The gradient of $E(\mathrm{w})$ is

$$\frac{\partial}{\partial \mathbf{w}} E(\mathbf{w}) = \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}} \left(h(\mathbf{x}^{(i)}) - y^{(i)}\right) \mathbf{x}^{(i)}$$

Training Logistic Regression

Randomly choose an initial solution $\mathbf{w}^0$,

Repeat

Choose a random sample set $\mathcal{B} \subseteq \mathcal{D}$.

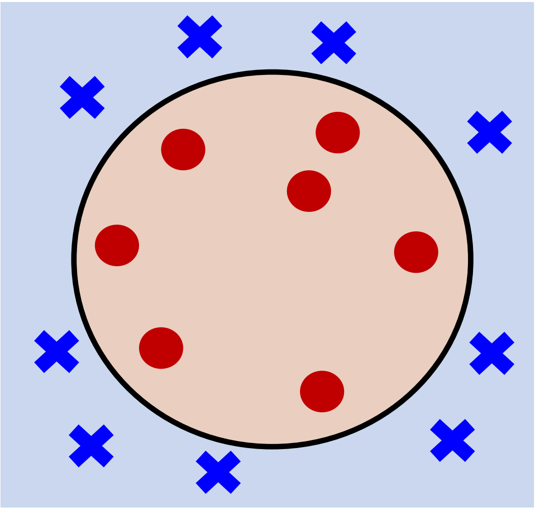
$$\Delta \mathbf{w} = \sum_{\left(\mathbf{x}^{(i)}, y^{(i)}\right) \in \mathcal{B}} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right) \mathbf{x}^{(i)}$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \Delta \mathbf{w}$$

Until stopping condition is satisfied

Discussion and Summary

- **No closed-form solution**
  - Optimized by the **gradient descent method**
- **A linear boundary**
  - How about a **non-linear** classifier?
- **Binary classifier**
  - How about **three or more classes**?



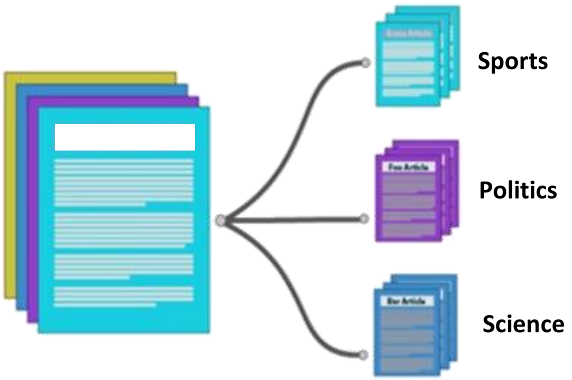
Non-linear separable

Multinomial Logistic Regression

Multinomial Logistic Regression

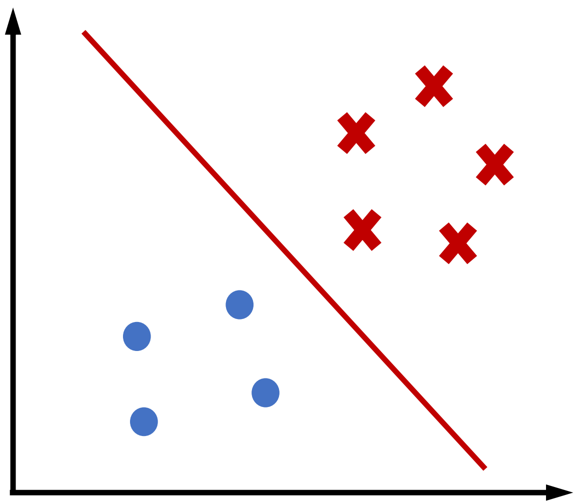
- **It is a classification method that generalizes logistic regression to the multiclass problem, i.e., with more than two possible discrete outcomes.**
  - It is also called **softmax regression** and **multinomial logit**.

- **Examples**
  - Which **major** will a student choose, given the status of the student?
  - Which **blood type** does a person have, given the results of various diagnostic tests?

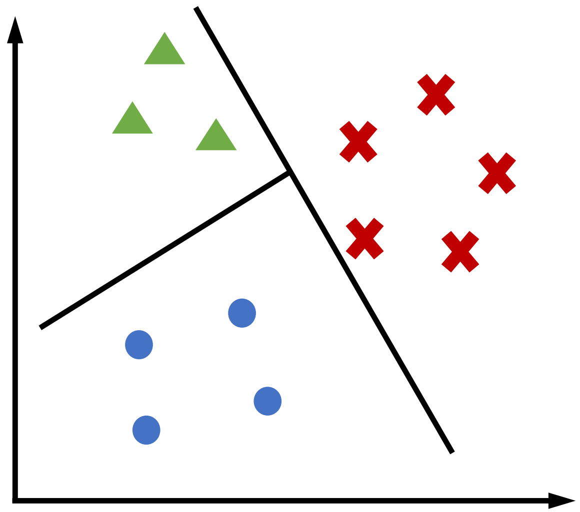


Multi-class Classification

- Classifying instances into one of **three or more classes**
  - **Binary classification**: Classifying instances into one of two classes



Binary classification

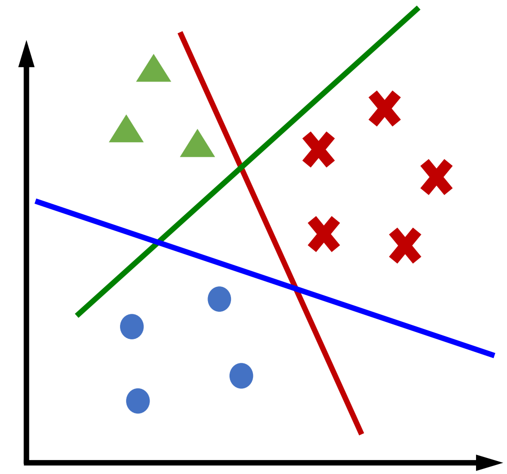


Multi-class classification

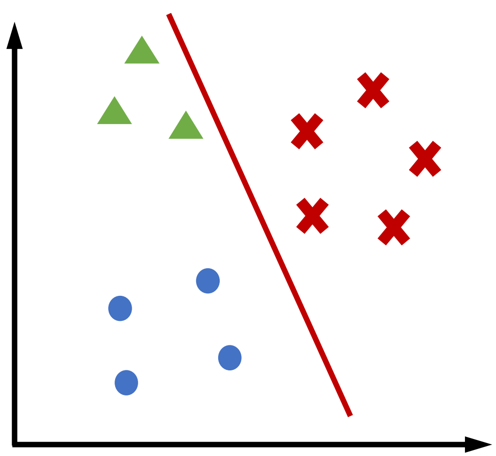
- How to classify **multiple classes** with some boundaries?

Multi-class Classification

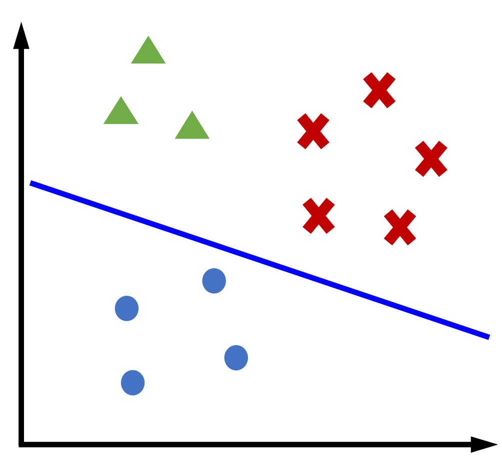
- **Considering a linear decision boundary for each class**
  - Use k classifiers for k classes.



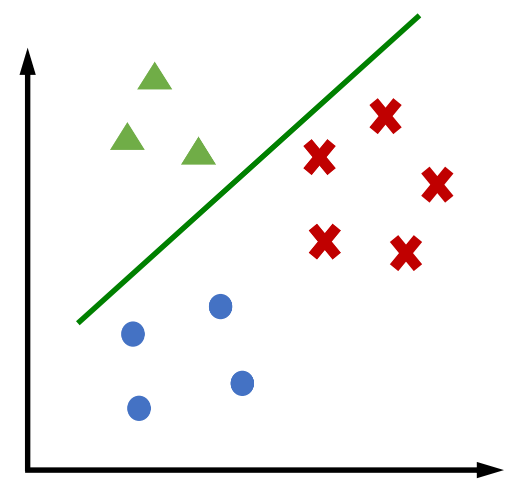
Red vs. **Others**



Blue vs. **Others**



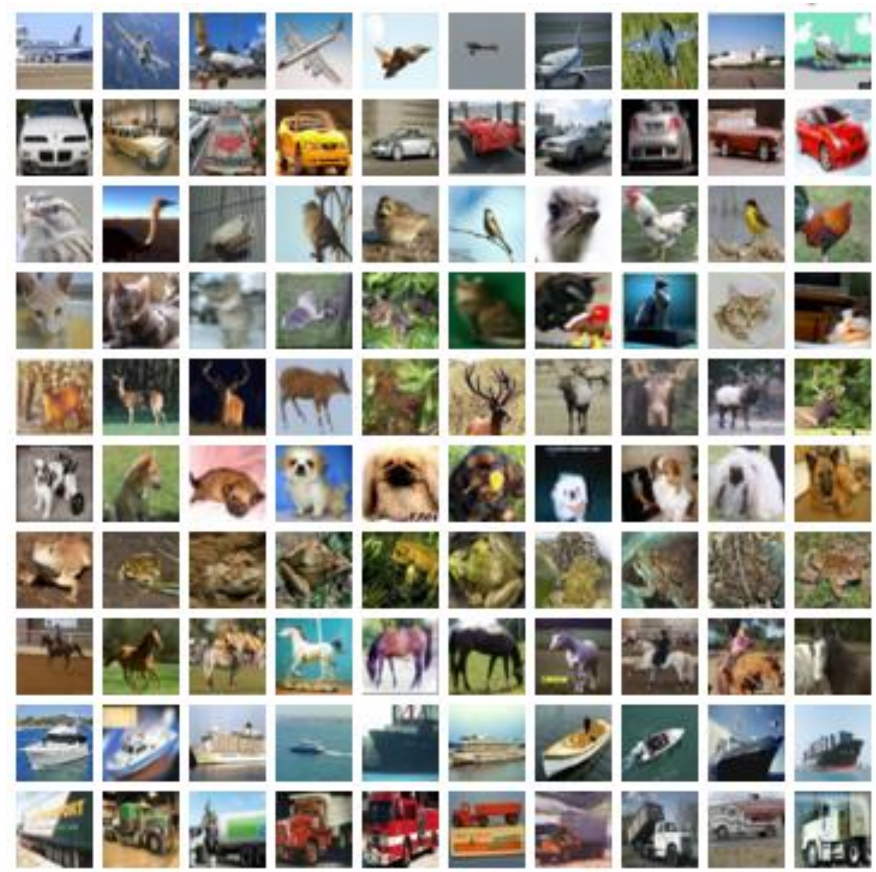
Green vs. **Others**



Example: Image Classification

CIFAR-10
- **10** labels
- **50,000** training images
- **10,000** test images
- Each image is **32x32x3**.

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck



https://www.cs.toronto.edu/~kriz/cifar.html

Multi-class Classification

- Given an input $\mathbf{x} \in \mathbb{R}^{3072\times 1}$, $f(\mathbf{x}; \mathbf{W}, \mathbf{b})$ returns $\mathbf{y} \in \mathbb{R}^{10\times 1}$.
  - $\mathbf{W}^T \in \mathbb{R}^{10\times 3072}$, $\mathbf{b} \in \mathbb{R}^{10\times 1}$

$f(\mathbf{x};\ \mathbf{W}, \boldsymbol{b})$

10 numbers, indicating class scores

[32x32x3]
(3072 numbers total)

Multi-class Classification

➢ **Given an input $\mathbf{x} \in \mathbb{R}^{3072\times 1}$, $f(\mathbf{x}; \mathbf{W}, \mathbf{b})$ returns $\mathbf{y} \in \mathbb{R}^{10\times 1}$.**
- $\mathbf{W}^T \in \mathbb{R}^{10\times 3072}, \mathbf{b} \in \mathbb{R}^{10\times 1}$

$$\underbrace{f(\mathbf{x};\ \mathbf{W}, b)}_{10\text{x}1} = \underbrace{\mathbf{W}^T}_{10\text{x}3072} \underbrace{\mathbf{x}}_{3072\text{x}1} + \underbrace{\mathbf{b}}_{10\text{x}1}$$

10 numbers, indicating class scores

[32x32x3]
(3072 numbers total)

Example: Multi-class Classification

Stretch pixels into a column vector.

$$\begin{bmatrix} 0.2 & 0.3 & \cdots & \cdots & 1.2 \\ 1.5 & 2.3 & \cdots & \cdots & 2.9 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2.1 & 0.3 & \cdots & \cdots & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 13 \\ 24 \\ \vdots \\ \vdots \\ 71 \end{bmatrix} + \begin{bmatrix} 1.1 \\ 3.2 \\ \vdots \\ 0.4 \end{bmatrix} = \begin{bmatrix} 42.1 \\ -52.4 \\ \vdots \\ 102.5 \end{bmatrix}$$

$$\boxed{\mathbf{W}^{\mathrm{T}}} \quad \boxed{\mathbf{x}} \quad \boxed{\mathbf{b}} \quad \boxed{f(\mathbf{x}; \mathbf{W}, \mathbf{b})}$$

10x3072 **3072x1** **10x1** **10x1**

Example: Multi-class Classification

➢ Concatenating **W** and **b**

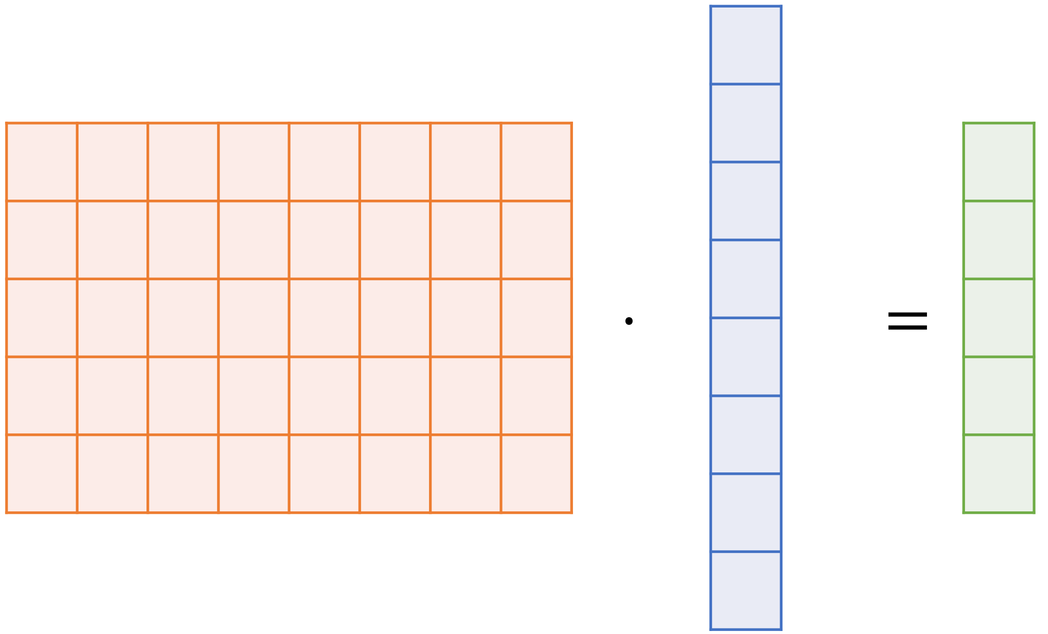
Stretch pixels into a column vector.

$$\begin{bmatrix} 1.1 & 0.2 & 0.3 & \cdots & 1.2 \\ 3.2 & 1.5 & 2.3 & \cdots & 2.9 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.4 & 2.1 & 0.3 & \cdots & 0.2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 13 \\ 24 \\ \vdots \\ \vdots \\ 71 \end{bmatrix} = \begin{bmatrix} 42.1 \\ -52.4 \\ \vdots \\ 102.5 \end{bmatrix}$$

$$\boxed{\mathbf{W}^{\mathrm{T}}} \qquad \boxed{\mathbf{x}} \qquad \boxed{f(\mathbf{x}; \mathbf{W}, \mathbf{b})}$$

10x3073 **3073x1** **10x1**

Example: Multi-class Classification



$\mathbf{W}^{\mathbf{T}} \in \mathbb{R}^{\boldsymbol{k}\times(\boldsymbol{d}+\mathbf{1})}$ $\mathbf{x} \in \mathbb{R}^{(\boldsymbol{d}+\mathbf{1})\times\mathbf{1}}$ $f(\mathbf{x}; \mathbf{W}, \mathbf{b}) \in \mathbb{R}^{k\times 1}$

➢ **Note: the output is not a probability.**

https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/

Computing the Logit

- For each class, the **logit** is computed.

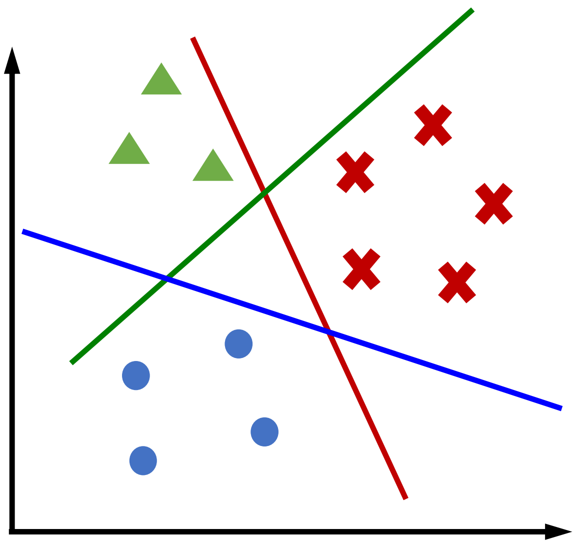
$$logit = \ln \frac{P(y=j \mid \mathbf{x}, \mathbf{W})}{1 - P(y=j \mid \mathbf{x}, \mathbf{W})} = \mathbf{w}_j^{\mathrm{T}}\mathbf{x}$$

$$\Rightarrow \quad \frac{P(y=j \mid \mathbf{x}, \mathbf{W})}{1 - P(y=j \mid \mathbf{x}, \mathbf{W})} = e^{\mathbf{w}_j^{\mathrm{T}}\mathbf{x}}$$

$$e^{[w_{10}\; w_{11}\; \dots \; \dots \; w_{1d}] \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_d \end{bmatrix}}$$

$$e^{[w_{20}\; w_{21}\; \dots \; \dots \; w_{2d}] \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_d \end{bmatrix}}$$

$$e^{[w_{30}\; w_{31}\; \dots \; \dots \; w_{3d}] \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_d \end{bmatrix}}$$



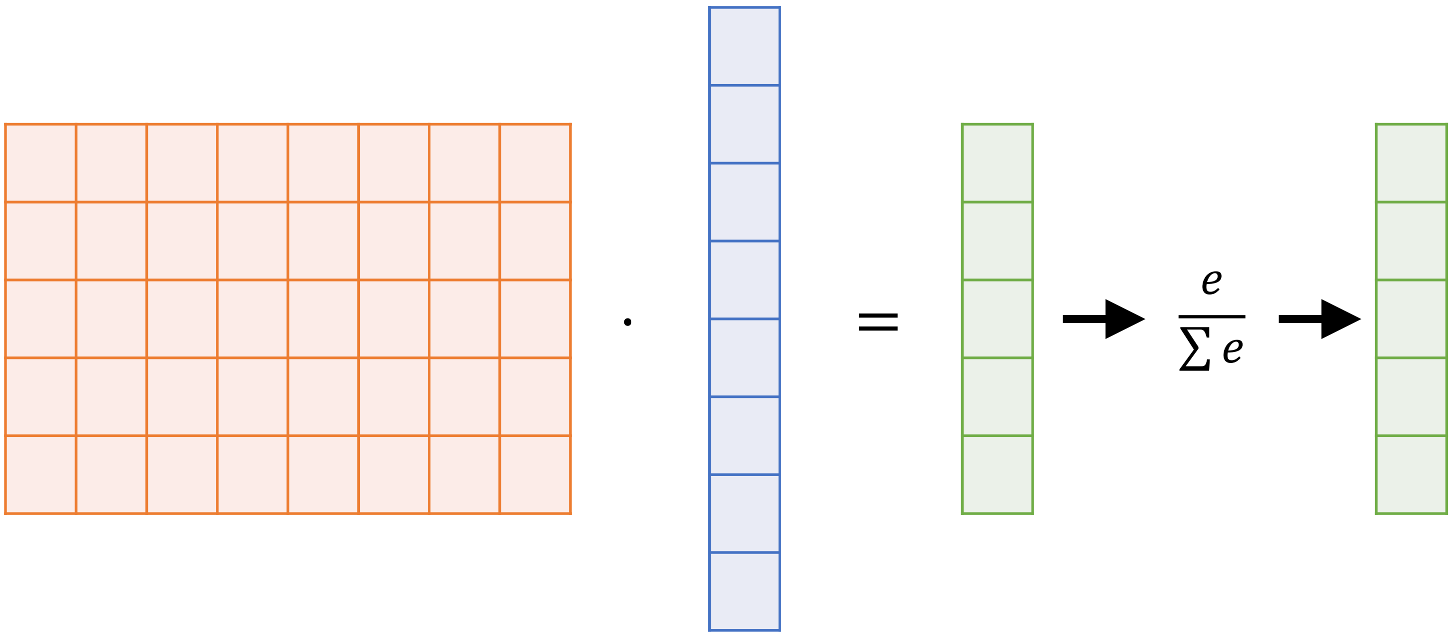
What is the Softmax Function?

- To represent a probability, the odds are normalized

$$P(y = j \mid \mathbf{x}, \mathbf{W}) = \frac{e^{\mathbf{w}_j^{\mathbf{T}}\mathbf{x}}}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathbf{T}}\mathbf{x}}}$$

$$\begin{bmatrix} P(y = 1 \mid \mathbf{x}, \mathbf{W}) \\ P(y = 2 \mid \mathbf{x}, \mathbf{W}) \\ \vdots \\ \vdots \\ P(y = k \mid \mathbf{x}, \mathbf{W}) \end{bmatrix} = \frac{1}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathbf{T}}\mathbf{x}}} \begin{bmatrix} e^{\mathbf{w}_1^{\mathbf{T}}\mathbf{x}} \\ e^{\mathbf{w}_2^{\mathbf{T}}\mathbf{x}} \\ \vdots \\ \vdots \\ e^{\mathbf{w}_k^{\mathbf{T}}\mathbf{x}} \end{bmatrix}$$

What is the Softmax Function?



$\mathbf{W}^{\mathbf{T}} \in \mathbb{R}^{\boldsymbol{k}\times(\boldsymbol{d}+\mathbf{1})}$ $\mathbf{x} \in \mathbb{R}^{(\boldsymbol{d}+\mathbf{1})\times\mathbf{1}}$ $e^{\mathbf{W}^{\mathrm{T}}\mathbf{x}} \in \mathbb{R}^{k\times 1}$ $probability$

https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/

What is the Softmax Function?

➢ **We want to maximize the probability of the correct class.**

$$P(y = j \mid \mathbf{x}, \mathbf{W}) = \frac{e^{\mathbf{w}_j^{\mathrm{T}}\mathbf{x}}}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}}}$$

$e^{\mathbf{w}_j^{\mathrm{T}}\mathbf{x}}$

	unnormalized probabilities	probabilities
cat	**24.5**	**0.12**
car	164.0	0.87
frog	0.18	0.01

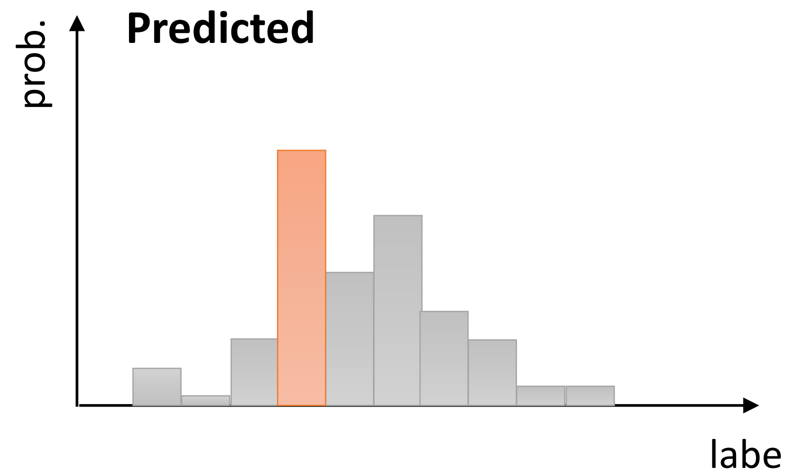
normalize

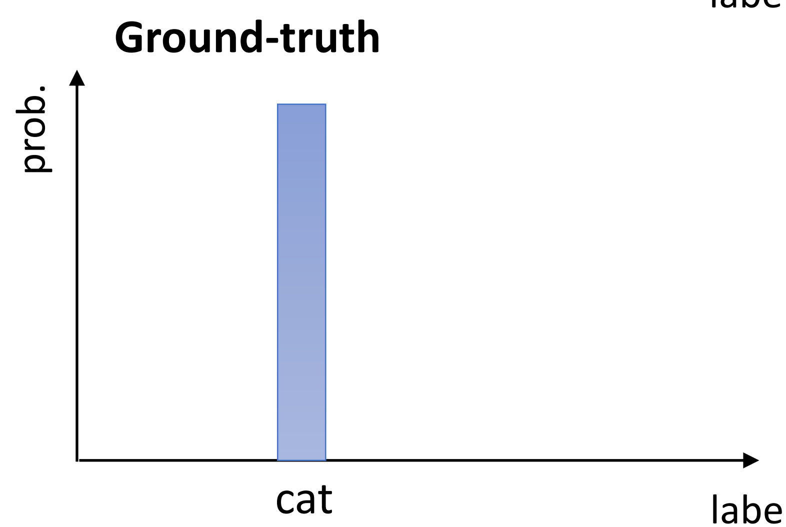
unnormalized probabilities → **probabilities**

How to Train the Softmax Regression?

➢ **Maximizing the class probability of the ground truth**

$$P(y = j \mid \mathbf{x}, \mathbf{W}) \frac{e^{\mathbf{w}_j^{\mathbf{T}}\mathbf{x}}}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathbf{T}}\mathbf{x}}}$$

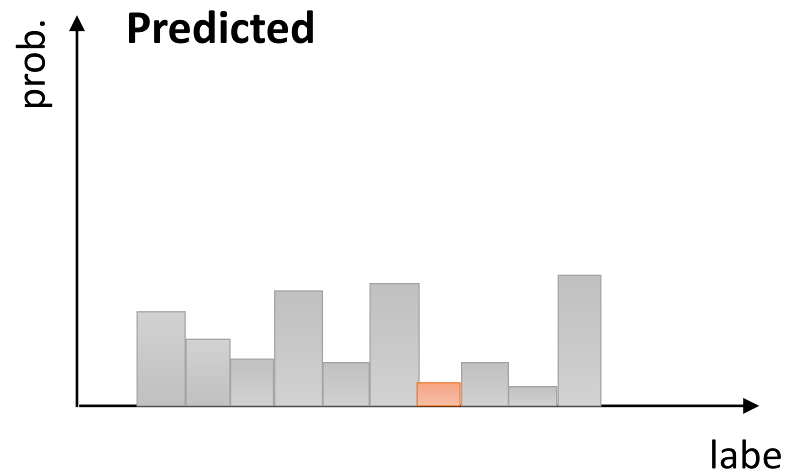


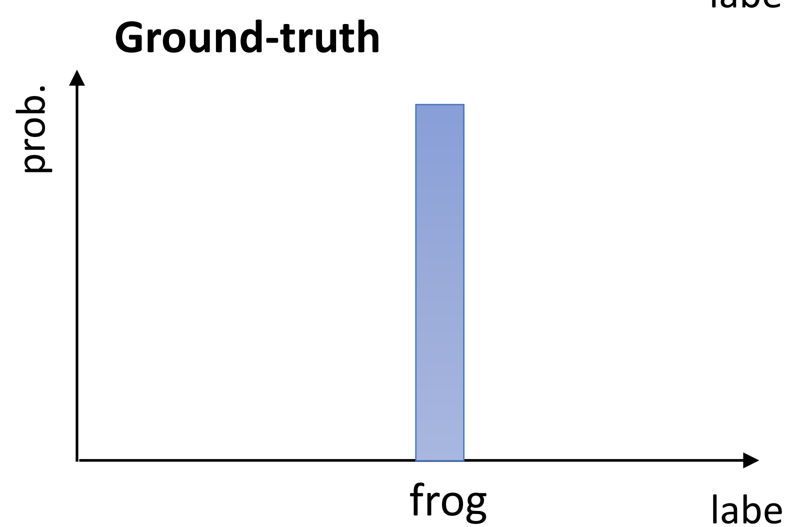


How to Train the Softmax Regression?

- **Maximizing the class probability of the ground truth**

$$P(y = j \mid \mathbf{x}, \mathbf{W}) \frac{e^{\mathbf{w}_j^{\mathbf{T}}\mathbf{x}}}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathbf{T}}\mathbf{x}}}$$





Formulating the Error Function

- **Generalizing the error function of binary classification**

$$E(\mathbf{w}) = -\sum_{i=1}^{n} y^{(i)} \ln\left(P\left(y^{(i)} = 1 \mid \mathbf{x}^{(i)}, \mathbf{w}\right)\right) + \left(1 - y^{(i)}\right) \ln\left(1 - P\left(y^{(i)} = 1 \mid \mathbf{x}^{(i)}, \mathbf{w}\right)\right)$$

$$\Downarrow$$

$$E(\mathbf{w}) = -\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}\left[y^{(i)} = j\right] \ln\left(P\left(y^{(i)} = j \mid \mathbf{x}^{(i)}, \mathbf{w}\right)\right)$$

$$\mathbb{I}\left[y^{(i)} = j\right] = \begin{cases} 1 & \mathit{if}\ y^{(i)} = j \\ 0 & \mathit{otherwise} \end{cases}$$

$$P\left(y^{(i)} = j \mid \mathbf{x}^{(i)}, \mathbf{w}\right) = \frac{e^{\mathbf{w}_j^{\mathbf{T}} \mathbf{x}^{(i)}}}{\sum_{i=1}^{k} e^{\mathbf{w}_i^{\mathbf{T}} \mathbf{x}^{(i)}}}$$

Training Sofmax Regression

How to train softmax regression?

- **How to solve this?**
  - A closed-form equation
  - Gradient descent method

Recap: Training Logistic Regression

Randomly choose an initial solution $\mathbf{w}^0$,

Repeat

Choose a random sample set $B \subseteq D$.

$$\Delta\mathbf{w} = \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in B} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right) \mathbf{x}^{(i)}$$

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta \Delta\mathbf{w}$$

Until the stopping condition is satisfied

Solving Softmax Regression by GD

- For the error function, compute the partial derivative of $\mathbf{w}$.

$$E(\mathbf{w}) = -\sum_{i=1}^{n}\sum_{j=1}^{k} \mathbb{I}\left[y^{(i)} = j\right] \ln\left(P\left(y^{(i)} = j \,\middle|\, \mathbf{x}^{(i)}, \mathbf{w}\right)\right)$$

- Then, apply $\nabla_{\mathbf{w}} E = \frac{\partial E}{\partial \mathbf{w}}$ to the gradient descent method.

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta \boldsymbol{\nabla}_{\mathbf{w}} \boldsymbol{E}$$

Q&A