Lecture 4

Machine Learning – III

Classification

Outline

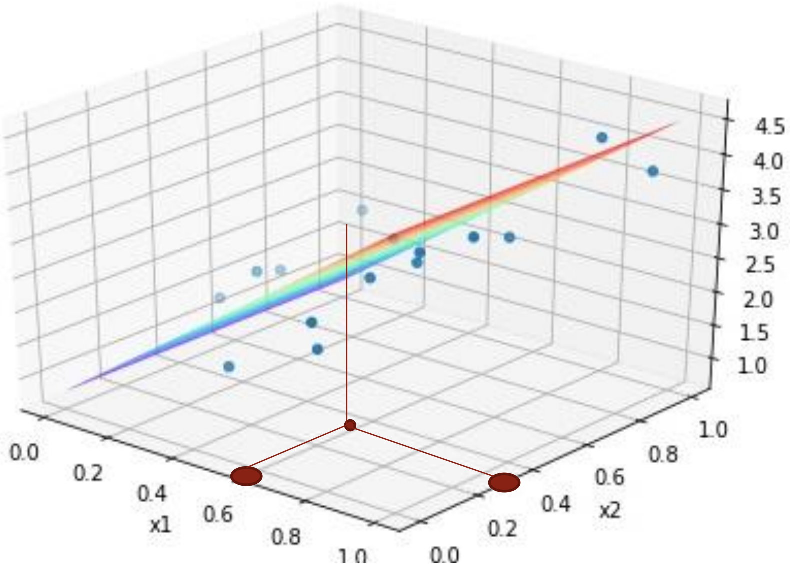
- Brief review
- Binary Classification
- Multi-class classification

Multivariate Linear Regression

Size	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
prize	**2.737**	**2.237**	**2.869**	**1.626**	**2.22**	**3.038**	**3.929**	**3.351**	**4.308**	**1.304**	**1.917**	**3.189**	**2.656**	**3.517**	**1.663**

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=2 is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^2, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.

$$\left(\boldsymbol{x}^i = \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix}, y^i\right)$$



Multivariate Linear Regression

Size	0.488	0.681	0.655	0.088
Color	0.609	0.112	0.324	0.669
bias	1	1	1	1
prize	**2.737**	**2.237**	**2.869**	**1.626**

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=2 is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{w}\boldsymbol{x} + b = w_1 x_1 + w_2 x_2 + b,$$

$$\text{where } \boldsymbol{w} = [w_1, w_2], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Object: $\underset{w,b}{\operatorname{argmin}} J(w, b) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$

$$\begin{aligned} a^i &= \boldsymbol{w}\boldsymbol{x}^{\boldsymbol{i}} + b \\ &= w_1 x_1^i + w_2 x_2^i + b \cdot 1 \\ &= w_1 x_1^i + w_2 x_2^i + w_3 \cdot 1 \\ &= w_1 x_1^i + w_2 x_2^i + w_3 \cdot x_3^i \\ &= \sum_{j=1}^{n} w_j \cdot x_j^i \end{aligned}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longrightarrow \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Multivariate Linear Regression

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where *n*=2 is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{wx} + b = w_1 x_1 + w_2 x_2 + b,$$

$$\text{where } \boldsymbol{w} = [w_1, w_2], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Object:

$$\operatorname*{argmin}_{w,b} J(w, b) = \operatorname*{argmin}_{w,b} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where *n*=3 is the *dimension* of the input vector $\boldsymbol{x}$ *and* x_3=1.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^{\mathrm{n}}, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{wx} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

$$\text{where } \boldsymbol{w} = [w_1, w_2, w_3], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Object:

$$\operatorname*{argmin}_{w} J(\boldsymbol{w}) = \operatorname*{argmin}_{w} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Least Squares Approximations

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=3 is the *dimension* of the input vector $\boldsymbol{x}$ *and* x_3=1.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{wx} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

$$\text{where } \boldsymbol{w} = [w_1, w_2, w_3], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Object:

$$\underset{w}{\operatorname{argmin}} J(\boldsymbol{w}) = \underset{w}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

For a differentiable function $J\,(w)$, if w^* is a minimum point of J, then the following equation holds:

$$\left.\frac{\partial J}{\partial w}\right|_{w=w^*} = 0$$

$$\Downarrow$$

$$\begin{cases} \left.\dfrac{\partial J}{\partial w_1}\right|_{w_1=w_1^*} = 0 \\ \left.\dfrac{\partial J}{\partial w_2}\right|_{w_2=w_2^*} = 0 \\ \left.\dfrac{\partial J}{\partial w_3}\right|_{w_3=w_3^*} = 0 \end{cases}$$

Least Squares Approximations

$$\frac{1}{m}\left[(a^1-y^1),\cdots(a^i-y^i),\cdots,(a^m-y^m)\right]\begin{bmatrix}x_1^1\\x_1^2\\\vdots\\x_1^i\\\vdots\\x_1^m\end{bmatrix} = \frac{\partial J}{\partial w_1}$$

$$\frac{1}{m}\left[(a^1-y^1),\cdots(a^i-y^i),\cdots,(a^m-y^m)\right]\begin{bmatrix}x_2^1\\x_2^2\\\vdots\\x_2^i\\\vdots\\x_2^m\end{bmatrix} = \frac{\partial J}{\partial w_2}$$

$$\frac{1}{m}\left[(a^1-y^1),\cdots(a^i-y^i),\cdots,(a^m-y^m)\right]\begin{bmatrix}x_3^1\\x_3^2\\\vdots\\x_3^i\\\vdots\\x_3^m\end{bmatrix} = \frac{\partial J}{\partial w_3}$$

Least Squares Approximations

$$\frac{1}{m}\left[(a^1 - y^1), \cdots (a^i - y^i), \cdots, (a^m - y^m)\right] \quad X^T = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \frac{\partial J}{\partial w_3}\right]$$

$$\downarrow \; \boldsymbol{wX - y}$$

$$\frac{1}{m}(\boldsymbol{wX - y})\boldsymbol{X^T} = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \frac{\partial J}{\partial w_3}\right]$$

$$\downarrow$$

$$(\boldsymbol{wX - y})\boldsymbol{X^T} = \boldsymbol{0}$$

$$\downarrow$$

$$\boldsymbol{wXX^T} = \boldsymbol{yX^T}$$

$$\downarrow$$

$$\boldsymbol{w} = \boldsymbol{yX^T}\left(\boldsymbol{XX^T}\right)^{-\boldsymbol{1}}$$

Steepest Gradient Descend Method

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=3 is the *dimension* of the input vector $\boldsymbol{x}$ *and* x_3=1.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{w}\boldsymbol{x} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

$$\text{where } \boldsymbol{w} = [w_1, w_2, w_3], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Object:

$$\underset{w}{\operatorname{argmin}} J(\boldsymbol{w}) = \underset{w}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Vector form

$$\frac{\partial J}{\partial w} = \frac{1}{m} (\boldsymbol{wX} - \boldsymbol{y})\boldsymbol{X}^T$$

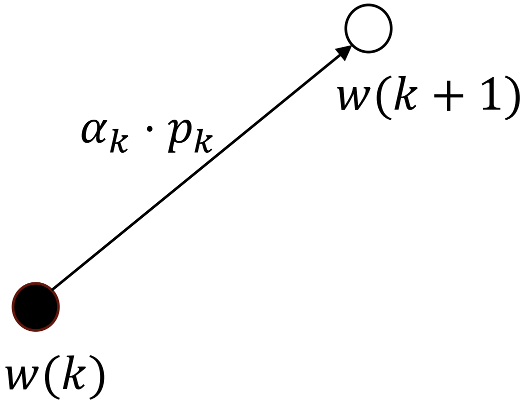
Component form

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (a^i - y^i) \cdot x_j^i$$

Steepest Gradient Descend Method

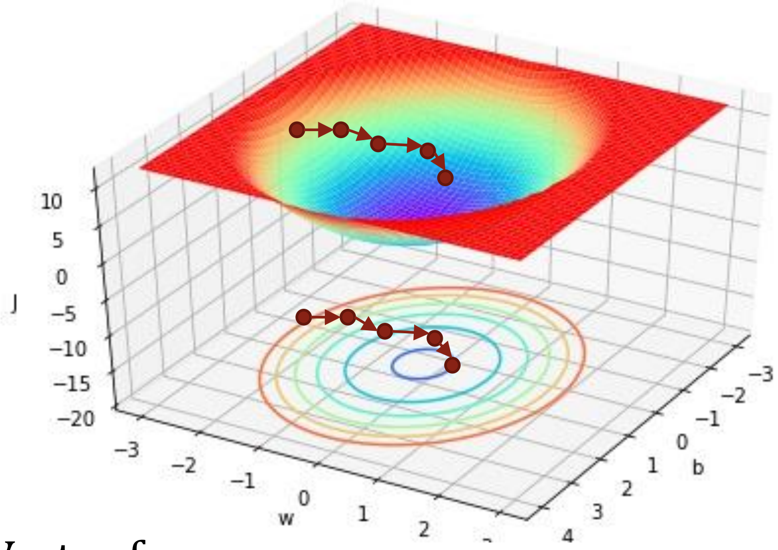
Finding a minimum point step by step

$$w(k+1) = w(k) \ + \alpha_k \cdot p_k^w$$



p_k , is called searching direction

α_k is learning rate at step k.



Vector form

$$\boldsymbol{w(k+1) = w(k)} - \alpha_k \frac{1}{m} \boldsymbol{(wX - y)X^T}$$

Component form

$$w_j(k+1) = w_j(k) - \alpha_k \frac{1}{m} \sum_{i=1}^{m} (a^i - y^i) \cdot x_j^i$$

Steepest Gradient Descend Method

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=3 is the *dimension* of the input vector $\boldsymbol{x}$ *and* x_3=1.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{w}\boldsymbol{x} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

$$\text{where } \boldsymbol{w} = [w_1, w_2, w_3], \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Object:

$$\underset{w}{\operatorname{argmin}} J(\boldsymbol{w}) = \underset{w}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Steepest Descent Algorithm

Input: D, w

for k in 1,2,…, K:

{

for i in 1,2,…,m:

{

$$a^i \leftarrow \sum_{j=1}^{n} w_j x_j^i$$

}

for j in 1,2,…, n:

{

$$\frac{\partial J}{\partial w_j} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (a^i - y^i) \cdot x_j^i$$

}

for j in 1,2,…, n:

{

$$w_j \leftarrow w_j - \alpha \frac{\partial J}{\partial w_j}$$

}

}

Steepest Gradient Descend Method

Multivariate Linear Regression

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in R)$, where n=3 is the *dimension* of the input vector $\boldsymbol{x}$ *and* x_3=1.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1, m]\}$, where m=15 is the number of the samples.
- Linear Model:

$$a = \boldsymbol{w}\boldsymbol{x} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

where $\boldsymbol{w} = [w_1, w_2, w_3]$, $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

- Object:

$$\underset{w}{\operatorname{argmin}} J(\boldsymbol{w}) = \underset{w}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Steepest Descent Algorithm

Input: D, w

for k in $1, 2, \ldots, K$:

{

$$\boldsymbol{a} \leftarrow \boldsymbol{w}\boldsymbol{X}$$

$$\frac{\partial J}{\partial w} = \frac{1}{m}(\boldsymbol{w}\boldsymbol{X} - \boldsymbol{y})\boldsymbol{X}^{\boldsymbol{T}}$$

$$w \leftarrow w - \alpha \frac{\partial J}{\partial w}$$

}

$$J = \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

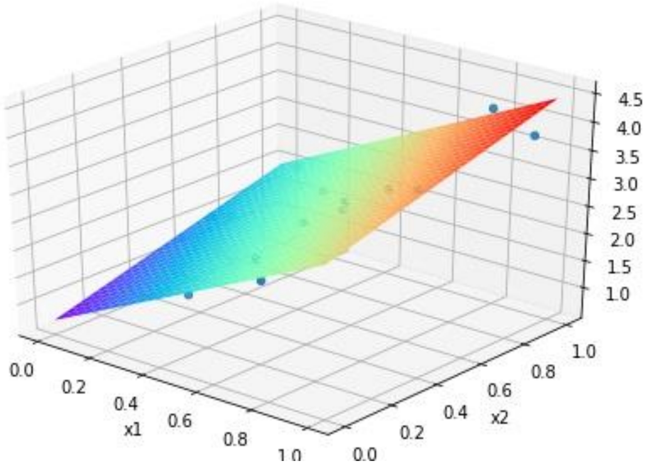
$$= \frac{1}{2m}(\boldsymbol{a} - \boldsymbol{y})(\boldsymbol{a} - \boldsymbol{y})^{\boldsymbol{T}}$$

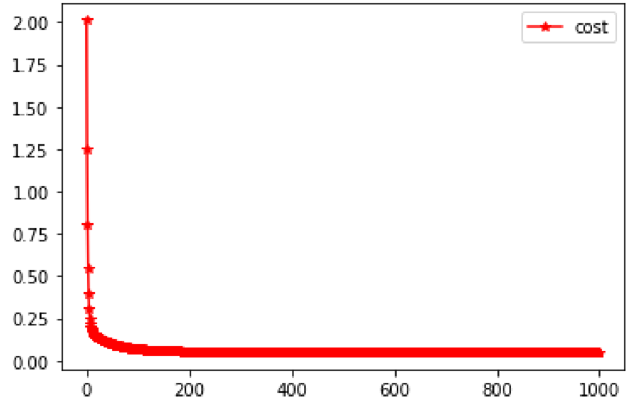
Steepest Gradient Descend Method

Size	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
bias	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
prize	**2.737**	**2.237**	**2.869**	**1.626**	**2.22**	**3.038**	**3.929**	**3.351**	**4.308**	**1.304**	**1.917**	**3.189**	**2.656**	**3.517**	**1.663**

Please predict the price for an apple (size=0.6, color=0.3)

2.47*0.6+1.25*0.3+0.69=2.56

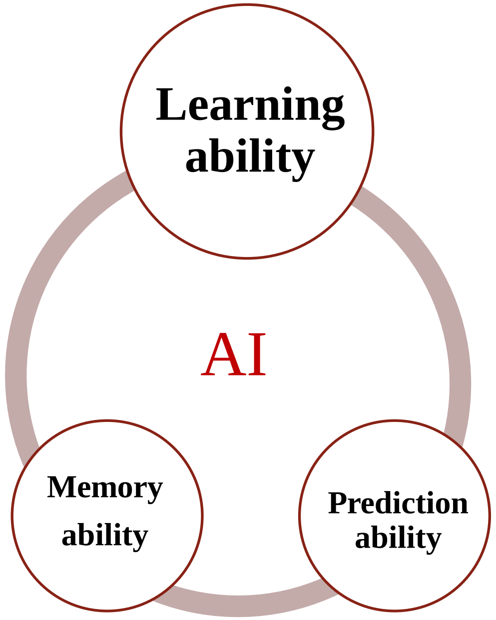




$w = [2.47,\ 1.25,\ 0.69]$

machine learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning



Supervised Learning

- **The dataset contains the true answer(label):**

$$D=\{\ (x,y)\}$$

- **Regression**
- **Classification**

Unsupervised Learning

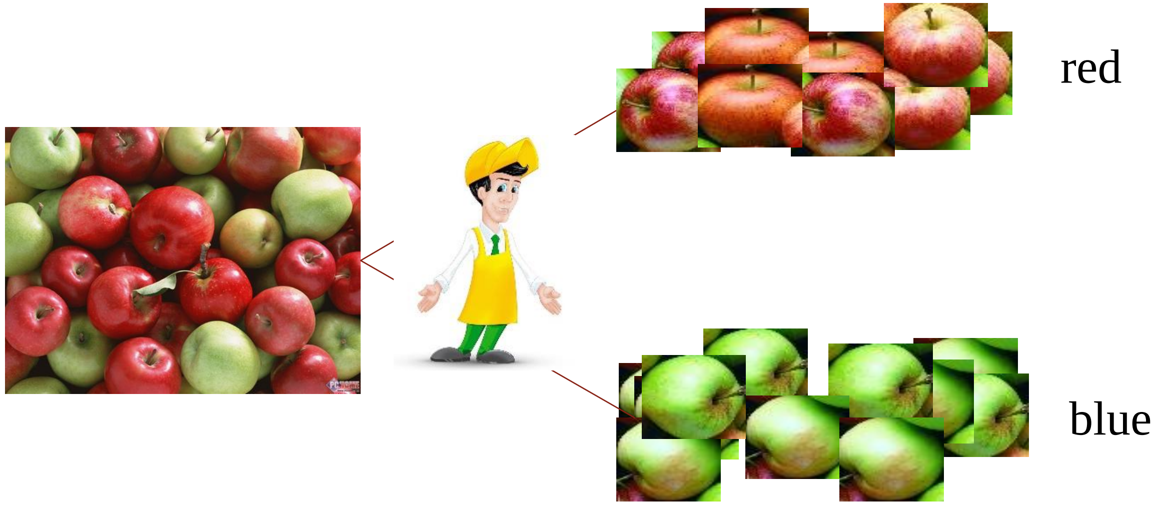
Reinforcement Learning

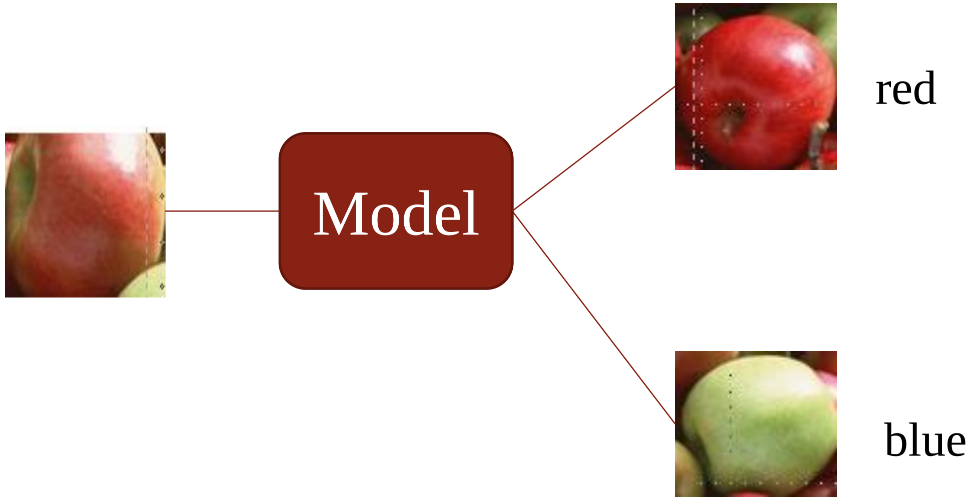
Outline

- Brief review
- Binary Classification
- Multi-class classification

Examples

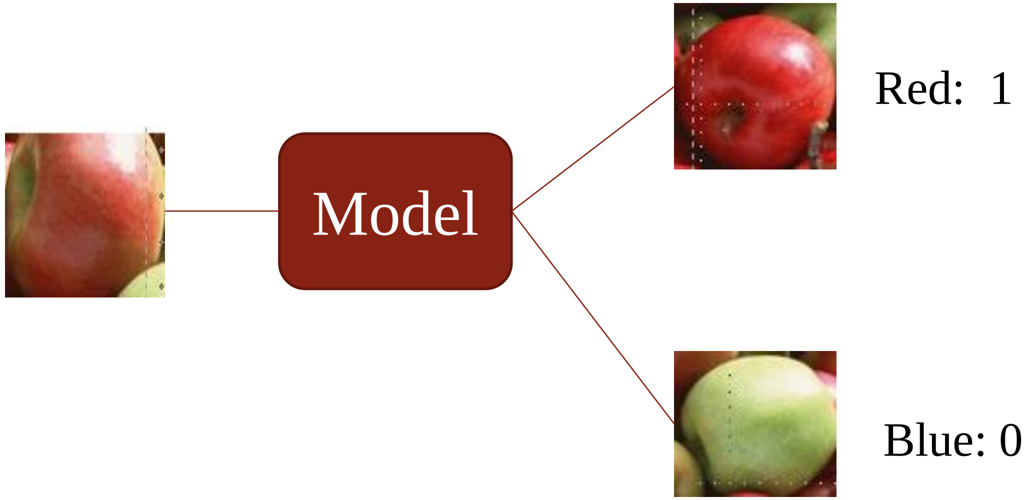
Color x	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
class	**red**	blue	blue	**red**	**red**	blue	**red**	blue	**red**	**red**	**red**	**red**	blue	blue	blue

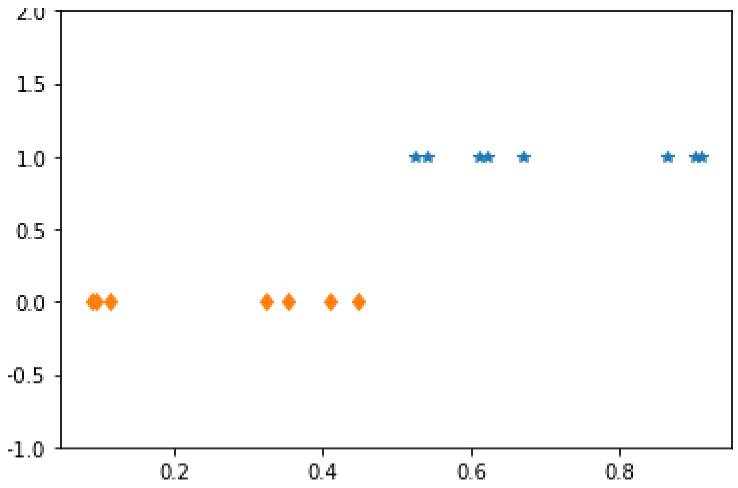




Examples

Color x	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
class	red	blue	blue	red	red	blue	red	blue	red	red	red	red	blue	blue	blue
label y	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0

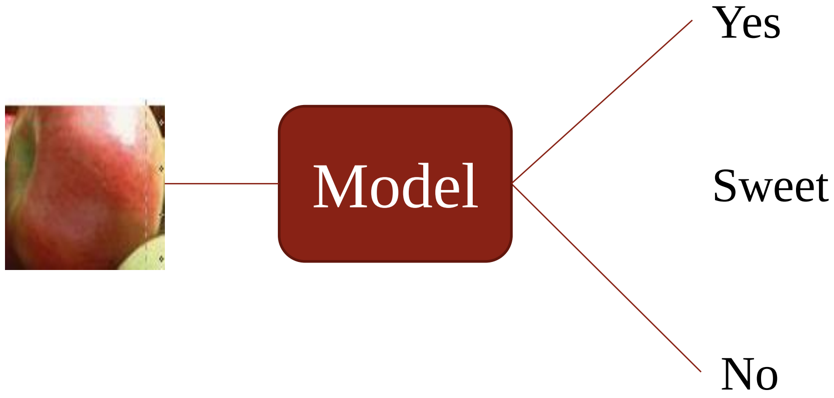




- Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$
- Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$

Examples

Size x_1	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color x_2	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Class sweet	**No**	**No**	**No**	**No**	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**Yes**	**No**	**Yes**	**No**



Examples

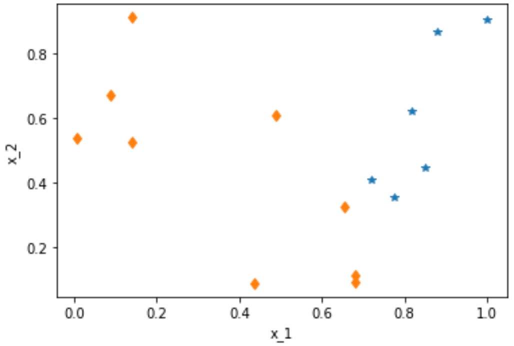
Size $\boldsymbol{x_1}$	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color $\boldsymbol{x_2}$	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Class **sweet**	No	No	No	No	No	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	No
Label y	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0

Model

Yes: 1

Sweet

No: 0



- Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$
- Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$

Examples

Size x_1	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color x_2	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Class **sweet**	No	No	No	No	No	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	No
Label y	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0

$\boldsymbol{x}$

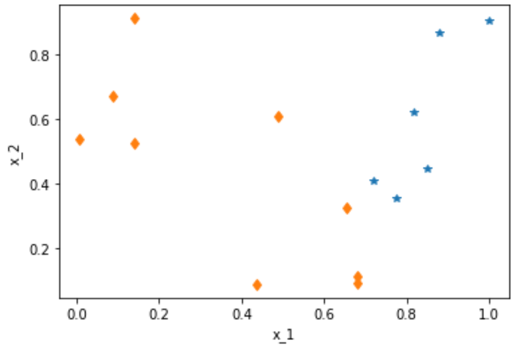
Model

$a = F(\boldsymbol{x}, \boldsymbol{w})$

Yes: 1

Sweet

No: 0



- Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$
- Dataset: $D = \left\{\left(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}\right) \middle| i \in [1, m]\right\}$

Binary classification

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Model:

$$a = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{w})$$

- Object:

$\forall(\boldsymbol{x}, y) \in D$, the output a is close to y

Model

positive: 1

Sweet

negative: 0

$$a = F(\boldsymbol{x}, \boldsymbol{w})$$

$\boldsymbol{x}$ —— y

How to define $F(\boldsymbol{x}, \boldsymbol{w})$

Binary classification

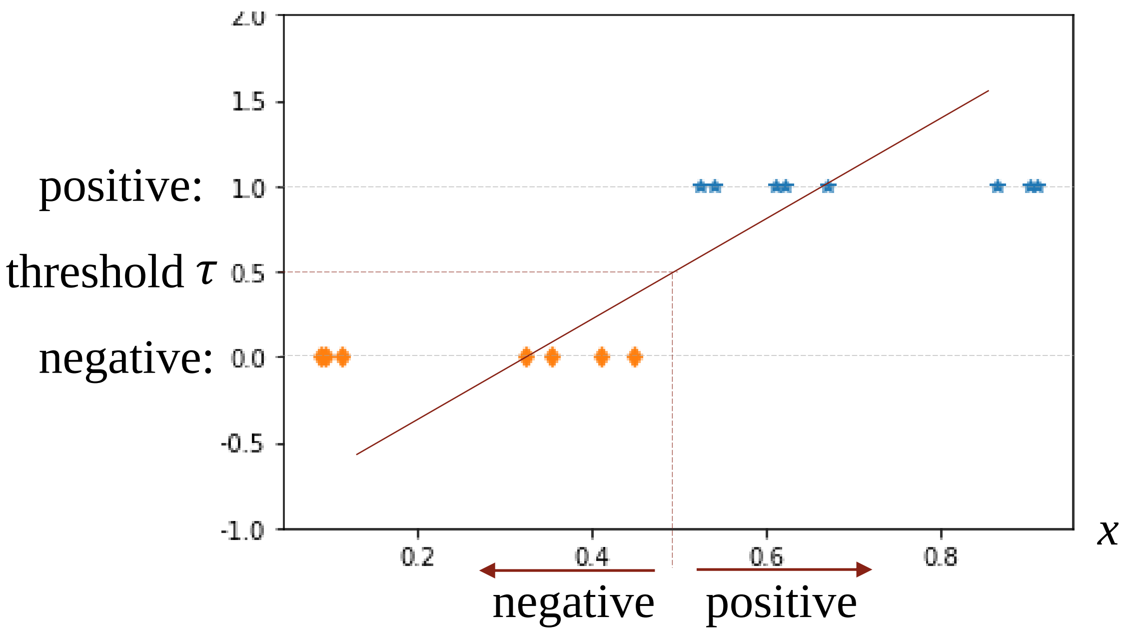
Color x	0.609	0.112	0.324	0.669
label y	1	0	0	1

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \left\{\left(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}\right) \middle| i \in [1, m]\right\}$, where m is the number of the samples.
- Linear Regression Model：

$$a = \boldsymbol{w}\boldsymbol{x} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

- Object：

$$\forall(\boldsymbol{x}, y) \in D, \text{the output } a \text{ is close to } y$$



Threshold classifier output a at τ :

if $a \geq \tau$, predict "y=1"

if $a < \tau$, predict "y=0".

For example $\tau = 0.5$

Binary classification

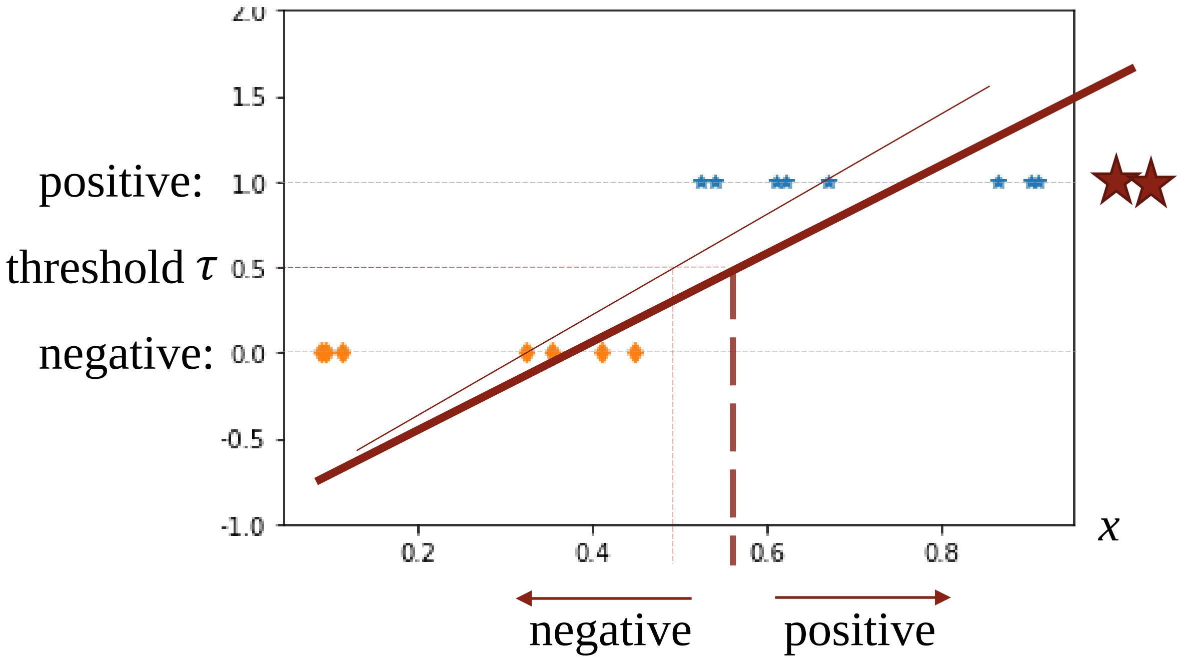
Color x	0.609	0.112	0.324	0.669
label y	1	0	0	1

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Linear Regression Model:

$$a = \boldsymbol{w}\boldsymbol{x} = \sum_{j=1}^{n} w_j \cdot x_j^i$$

- Object:

$\forall(\boldsymbol{x}, y) \in D$, the output a is close to y



label $y \in \{0,1\}$

output $a \in [-\infty, +\infty]$

Binary classification

Color x	0.609	0.112	0.324	0.669
label $\boldsymbol{y}$	1	0	0	1

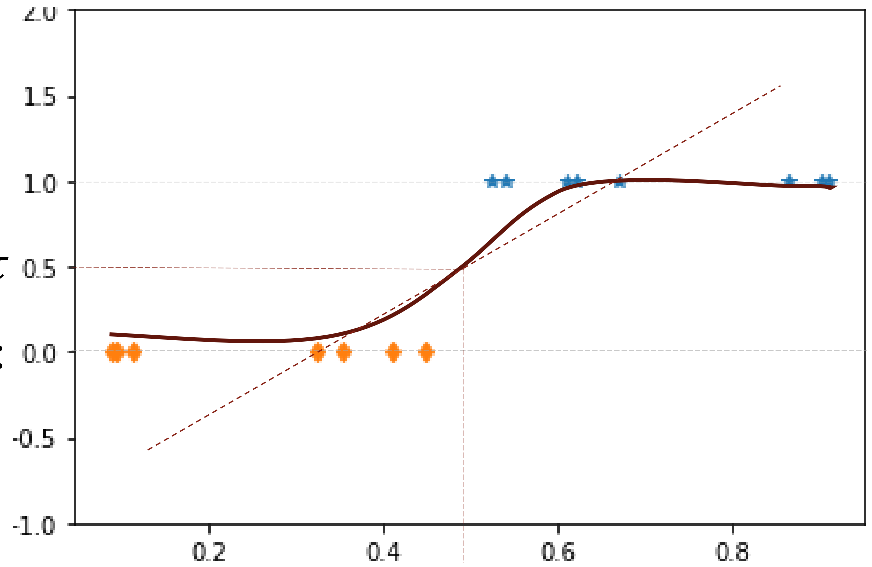
- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

$$\text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$

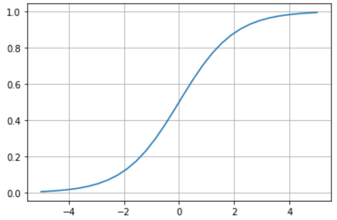
- Object:

  $\forall(\boldsymbol{x}, y) \in D$, the output a is close to y



label $y \in \{0,1\}$

output $a \in (0,1)$



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

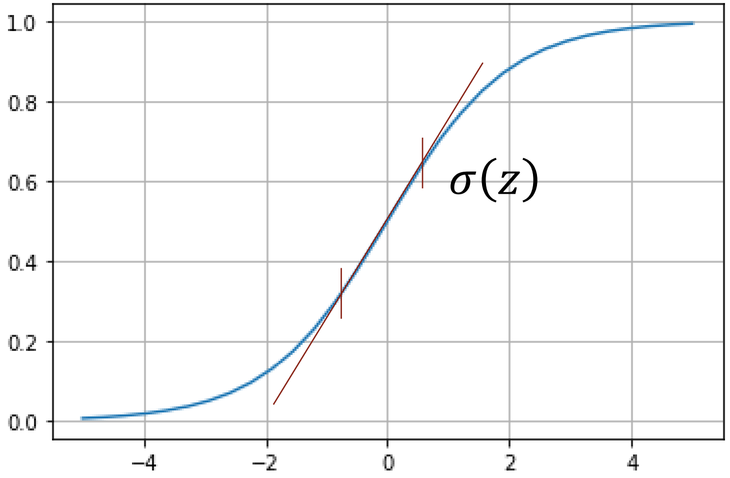
Sigmoid function

Sigmoid Function

A sigmoid function is a bounded, monotonic, differentiable function that is defined for all real input values and has a non-negative derivative at each point.

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

If $z_1 > z_2$ then $\sigma(z_1) > \sigma(z_2)$



Sigmoid Function

A sigmoid function is a bounded, monotonic, differentiable function that is defined for all real input values and has a non-negative derivative at each point.

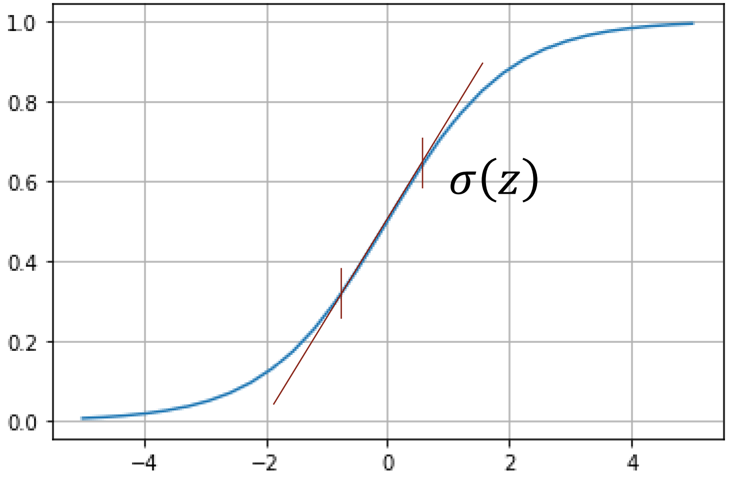
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

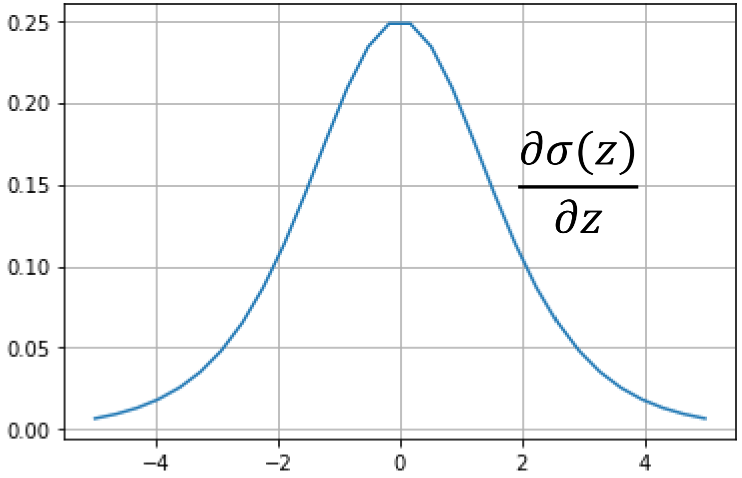
If $z_1 > z_2$ then $\sigma(z_1) > \sigma(z_2)$

$$\frac{\partial \sigma(z)}{\partial z} = -\frac{-e^{-z}}{(1 + e^{-z})^2}$$

$$\frac{\partial \sigma(z)}{\partial z} = \frac{1}{1 + e^{-z}} \cdot \frac{e^{-z}}{1 + e^{-z}}$$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z) \cdot (1 - \sigma(z))$$





Interpretation of output

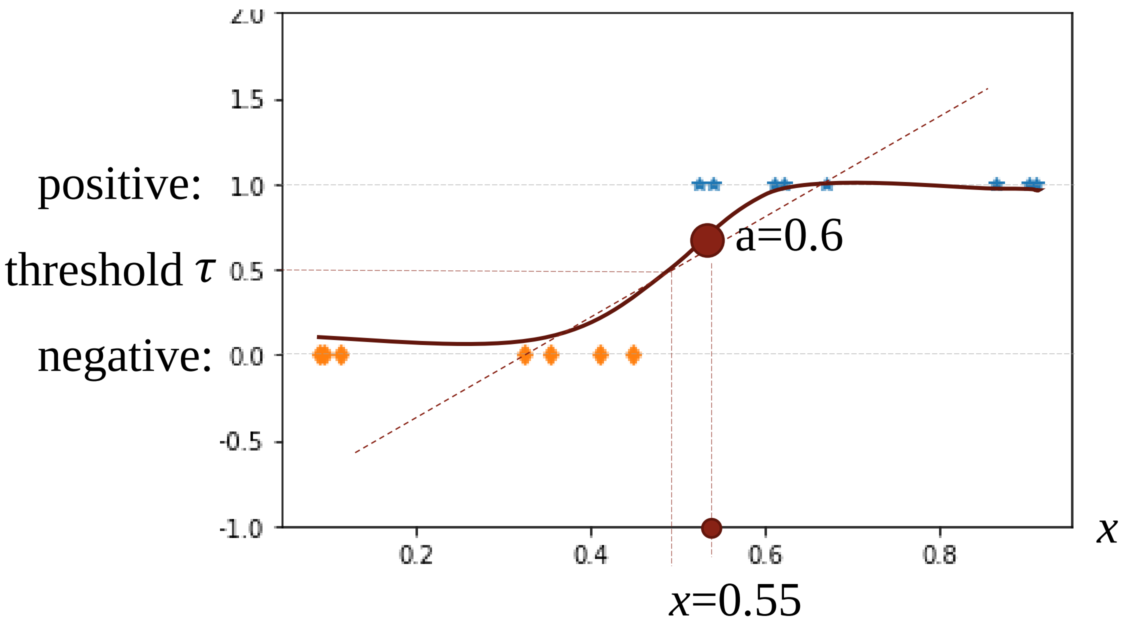
- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

- Object:

$\forall(\boldsymbol{x}, y) \in D$, the output a is close to y



Given an input x,
the output a estimates the probability that "$y=1$"

$$p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = a$$
$$p(y = 0|\boldsymbol{x}, \boldsymbol{w}) = 1 - a$$

Interpretation of output

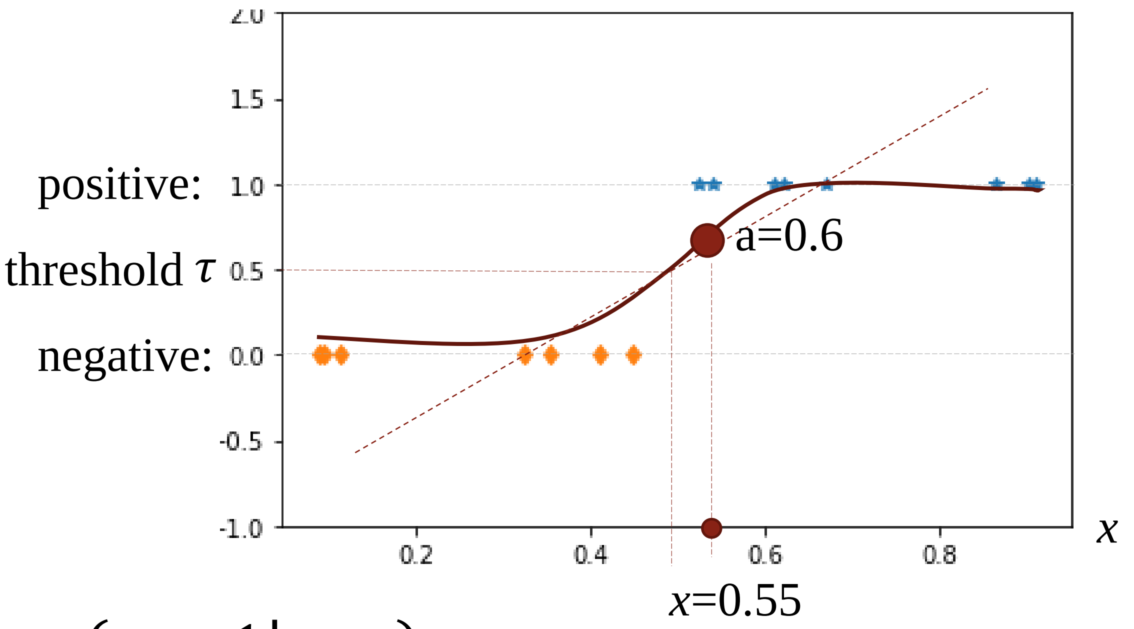
- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{w}\boldsymbol{x}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

$$\text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$

- Object:

$\forall(\boldsymbol{x}, y) \in D$, the output a is close to y



$$p(y = 1|\boldsymbol{x}, \boldsymbol{w}) = a$$

$$p(y = 0|\boldsymbol{x}, \boldsymbol{w}) = 1 - a$$

How to find w ?

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = a^y \cdot (1-a)^{1-y}$$

Cost function

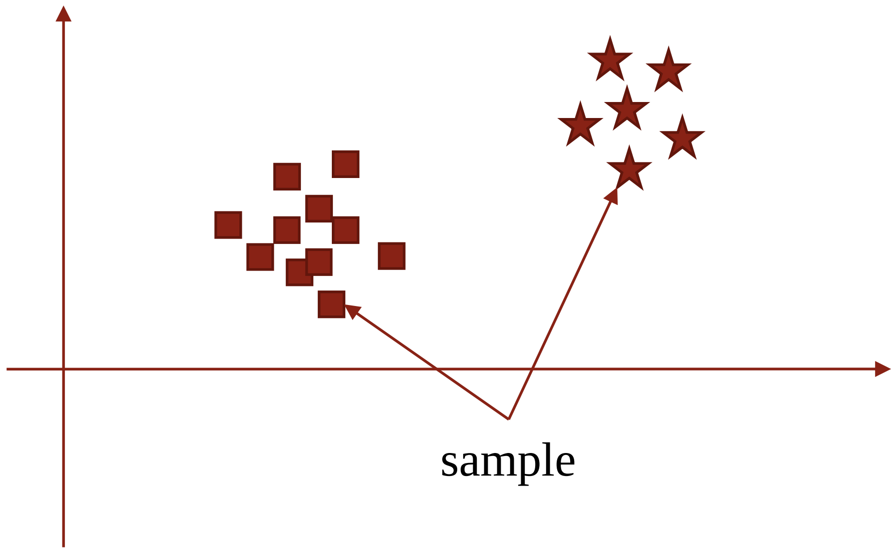
$$p(y|\boldsymbol{x},\boldsymbol{w}) = a^{y} \cdot (1-a)^{1-y}$$

Joint probability for sampled dataset D

$$P(Y|\boldsymbol{X},\boldsymbol{w}) = \prod_{i=1}^{m} p(y^{i}|\boldsymbol{x}^{\boldsymbol{i}},\boldsymbol{w})$$

Size $\boldsymbol{x_1}$	0.088	0.999	0.007	0.14	0.817	0.681	0.851	0.436
Color $\boldsymbol{x_2}$	0.669	0.902	0.539	0.524	0.62	0.093	0.447	0.088
Label y	0	1	0	0	1	0	1	0

$$\boldsymbol{X} = \begin{bmatrix} x_1^1, x_1^2, \cdots x_1^i, \cdots, x_1^m \\ x_2^1, x_2^2, \cdots x_2^i, \cdots, x_2^m \\ x_3^1, x_3^2, \cdots x_3^i, \cdots, x_3^m \end{bmatrix} \quad Y = \left[y^1, \cdots y^i, \cdots, y^m\right]$$



we would like to maximize the joint probability $P(Y|\boldsymbol{X},\boldsymbol{w})$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} P(Y|\boldsymbol{X},\boldsymbol{w})$$

Cost function

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \prod_{i=1}^{m} p(y^i|\boldsymbol{x^i},\boldsymbol{w})$$

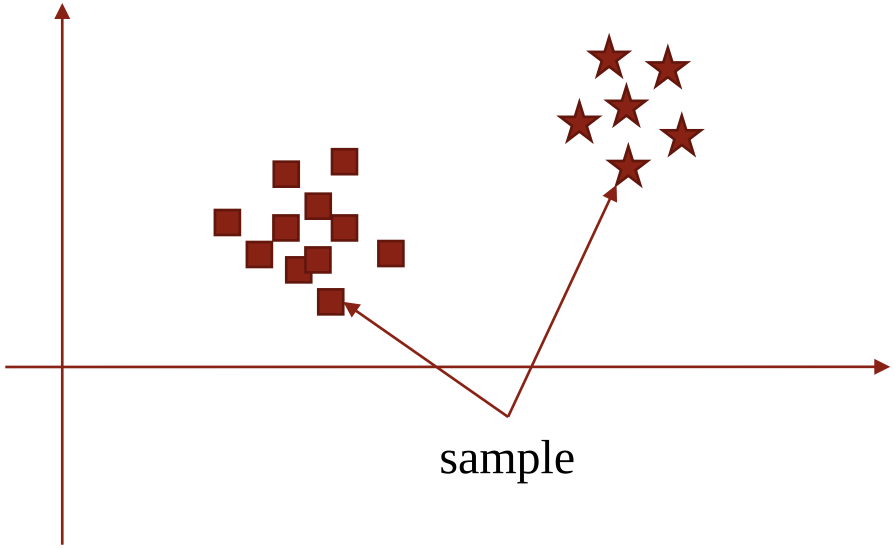
$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \log \prod_{i=1}^{m} p(y^i|\boldsymbol{x^i},\boldsymbol{w})$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{m} \log\left(p(y^i|\boldsymbol{x^i},\boldsymbol{w})\right)$$

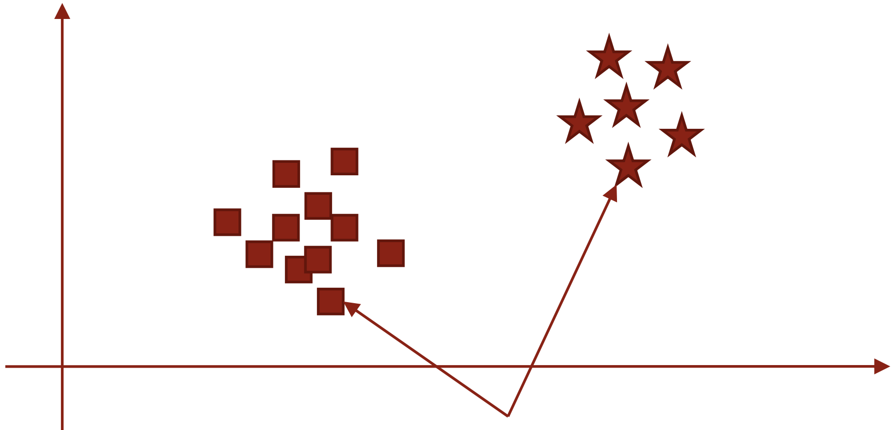
$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{m} \log\left(a^{i\,(y^i)} \cdot (1-a^i)^{1-y^i}\right)$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^{\mathrm{i}}) \cdot \log(1-a^i)\right]$$



$$p(y|\boldsymbol{x},\boldsymbol{w}) = a^y \cdot (1-a)^{1-y}$$

Cost function



$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \prod_{i=1}^{m} p(y^i|\boldsymbol{x^i},\boldsymbol{w})$$

$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^{\mathrm{i}}) \cdot \log(1-a^i)\right]$$

$$\Updownarrow$$

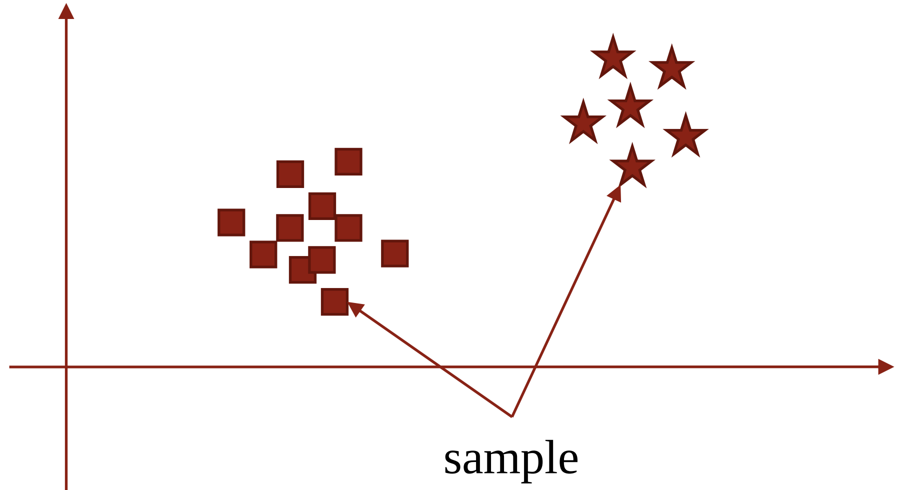
$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{1}{m} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^{\mathrm{i}}) \cdot \log(1-a^i)\right]$$

$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^{\mathrm{i}}) \cdot \log(1-a^i)\right]$$

$$\boxed{\boldsymbol{J}(\boldsymbol{w}) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^{\mathrm{i}}) \cdot \log(1-a^i)\right]}$$

Cost function



$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \prod_{i=1}^{m} p(y^i|\boldsymbol{x^i},\boldsymbol{w})$$

$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i) \right]$$

$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{1}{m} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i) \right]$$

$$\Updownarrow$$

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m} \log P(Y|\boldsymbol{X},\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i) \right]$$

$$\boxed{\boldsymbol{J}(\boldsymbol{w}) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i) \right]}$$

Cross-entropy cost function

Binary classification

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \overbrace{-[y^i \cdot \log(a^i) + (1 - y^i) \cdot \log(1 - a^i)]}^{e(a^i, y^i)}$$

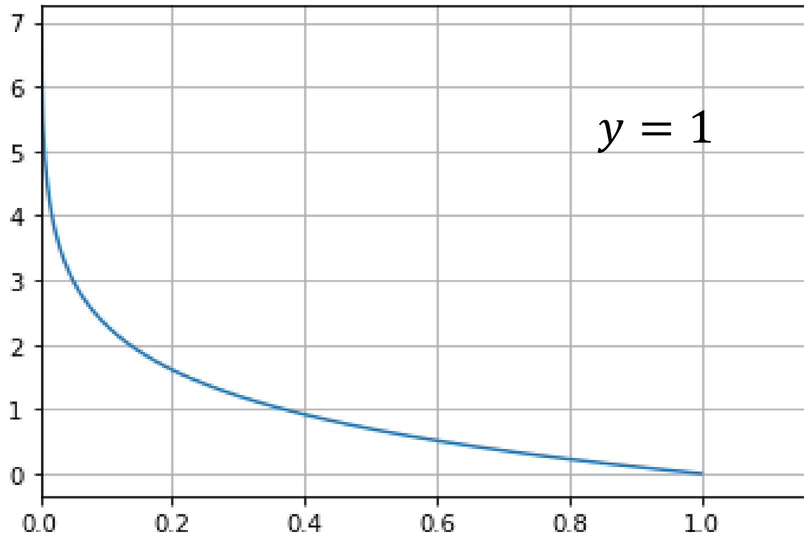
$$e(a\ \ , y) = \begin{cases} -\log(a), & y = 1 \\ -\log(1 - a) & y = 0 \end{cases}$$

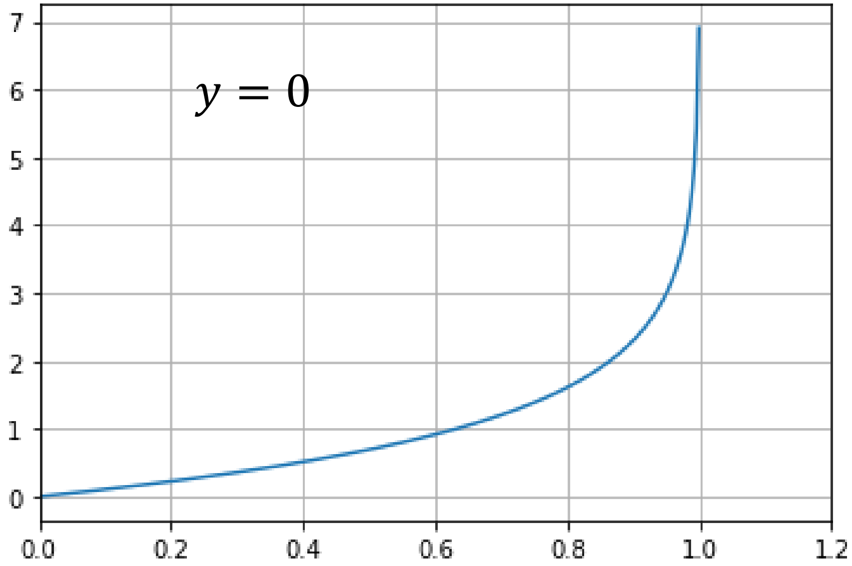
$$\begin{cases} e(a, y) \to \infty, if\ a\ is\ not\ close\ to\ y \\ e(a, y) \to 0, if\ a\ is\ close\ to\ y \end{cases}$$

Binary classification

$$e(a\ \ ,y) = \begin{cases} -\log(a)\,, & y = 1 \\ -\log(1-a) & y = 0 \end{cases}$$

$$\begin{cases} e(a,y) \to \infty, if\ a\ is\ not\ close\ to\ y \\ e(a,y) \to 0, if\ a\ is\ \ close\ to\ y \end{cases}$$





Steepest Gradient Descend Method

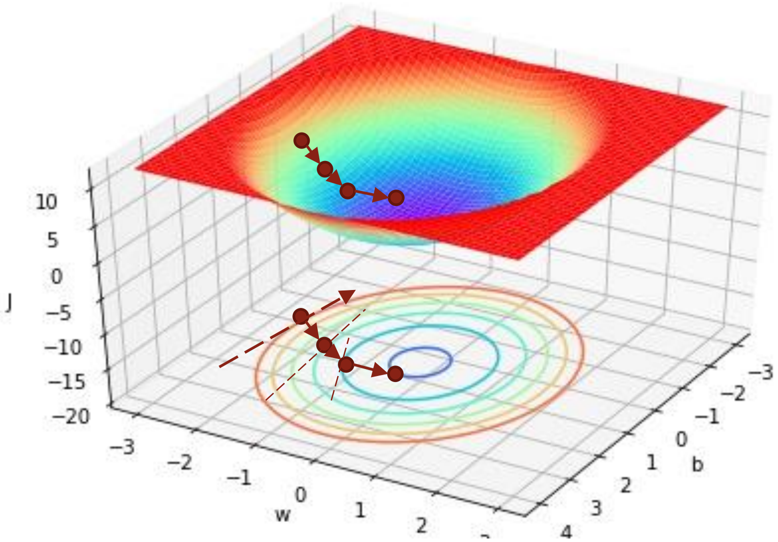
- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \overbrace{-[y^i \cdot \log(a^i) + (1 - y^i) \cdot \log(1 - a^i)]}^{e(a^i, y^i)}$$



Vector form

$$\boldsymbol{w}(\boldsymbol{k}+\boldsymbol{1}) = \boldsymbol{w}(\boldsymbol{k}) - \alpha_k \frac{\partial J}{\partial \boldsymbol{w}}$$

Component form

$$w_j(k+1) = w_j(k) - \alpha_k \frac{\partial J}{\partial w_j}$$

Steepest Gradient Descend Method

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial\left[\mathrm{y}^{\mathrm{i}}\cdot\log a^i + \left(1-y^i\right)\cdot\log\left(1-a^i\right)\right]}{\partial a^i}\cdot\frac{\partial a^i}{\partial w_j}$$

$$a = \sigma\left(\sum_{j}^{n} w_j x_j\right)$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i\cdot\frac{1}{a^i} + \left(1-y^i\right)\cdot\frac{-1}{1-a^i}\right]\cdot\frac{\partial a^i}{\partial w_j}$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\frac{y^i - a^i}{a^i\left(1-a^i\right)}\cdot\frac{\partial a^i}{\partial w_j}$$

Steepest Gradient Descend Method

$$a = \sigma\left(\sum_{j}^{n} w_j x_j\right)$$

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

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)\cdot(1-\sigma(z))$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\frac{y^i - a^i}{a^i(1-a^i)}\cdot\frac{\partial a^i}{\partial w_j}$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\frac{y^i - a^i}{a^i(1-a^i)}\cdot\frac{\partial a^i}{\partial \sum_l^n w_l x_l^i}\cdot\frac{\partial \sum_l^n w_l x_l^i}{\partial w_j}$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}\frac{y^i - a^i}{a^i(1-a^i)}\cdot\left[a^i\cdot\left(1-a^i\right)\right]\cdot x_j^i$$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - a^i)\cdot x_j^i \;=\; \frac{1}{m}\sum_{i=1}^{m}(a^i - y^i)\cdot x_j^i$$

Steepest Gradient Descend Method

$$\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(a^i - y^i)\cdot x_j^i \quad = \frac{1}{m}\left[(a^1-y^1),\quad (a^2-y^2),\quad \cdots,(a^i-y^i)\quad \cdots \quad (a^m-y^m)\right]\begin{bmatrix} x_j^1 \\ x_j^2 \\ \vdots \\ x_j^i \\ \vdots \\ x_j^m \end{bmatrix}$$

$$\left[\frac{\partial J}{\partial w_1},\frac{\partial J}{\partial w_2},\cdots,\frac{\partial J}{\partial w_n}\right] = \frac{1}{m}\left[(a^1-y^1),\cdots(a^i-y^i),\cdots,(a^m-y^m)\right]\begin{bmatrix} x_1^1 \\ x_1^2 \\ \vdots \\ x_1^i \\ \vdots \\ x_1^m \end{bmatrix}\begin{bmatrix} x_2^1 \\ x_2^2 \\ \vdots \\ x_2^i \\ \vdots \\ x_2^m \end{bmatrix}\cdots\begin{bmatrix} x_n^1 \\ x_n^2 \\ \vdots \\ x_n^i \\ \vdots \\ x_n^m \end{bmatrix}$$

$$\frac{\partial J}{\partial \boldsymbol{w}} = \frac{1}{m}[\boldsymbol{a} - Y]X^T$$

Steepest Gradient Descend Method

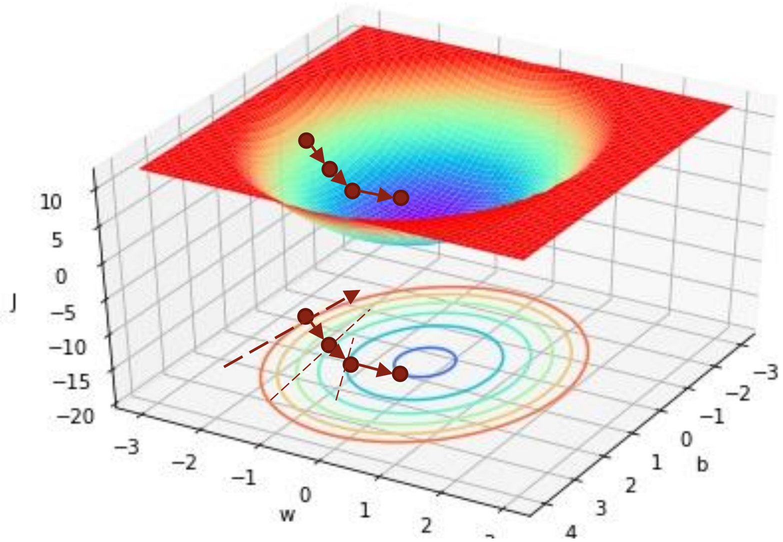
- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{w}\boldsymbol{x}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i)\right]$$



Vector form

$$\boldsymbol{w}(\boldsymbol{k}+\boldsymbol{1}) = \boldsymbol{w}(\boldsymbol{k}) - \alpha_k \frac{1}{m}[\boldsymbol{a} - Y]X^T$$

Component form

$$w_j(k+1) = w_j(k) - \alpha_k \frac{1}{m}\sum_{i=1}^{m}(a^i - y^i) \cdot x_j^i$$

Steepest Gradient Descend Method

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{w}\boldsymbol{x}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

$$\text{where } \boldsymbol{\sigma}(z) = \frac{1}{1+e^{-z}}$$

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \cdot \log(a^i) + (1-y^i) \cdot \log(1-a^i)\right]$$

Steepest Descent Algorithm

Input: D, w, α

for k in $1,2,\ldots,K$:

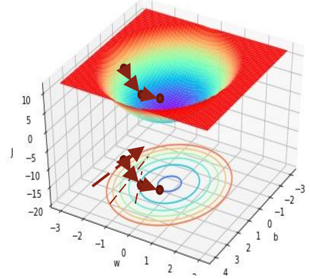
{

for i in 1,2,…,m:

{

$$a^i \leftarrow \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

}



for j in 1,2,…, n:

{

$$\frac{\partial J}{\partial w_j} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(a^i - y^i) \cdot x_j^i$$

}

for j in 1,2,…, n:

{

$$w_j \leftarrow w_j - \alpha\frac{\partial J}{\partial w_j}$$

}

}

Steepest Gradient Descend Method

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

$$\text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \cdot \log(a^i) + (1 - y^{\mathrm{i}}) \cdot \log(1 - a^i)\right]$$

Steepest Descent Algorithm

Input: D, w, α

for k in $1,2,\ldots,K$:

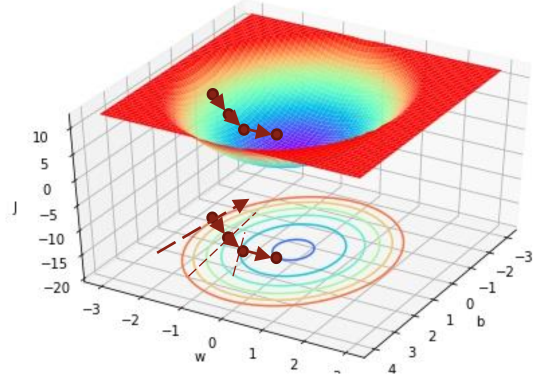
{

$$\boldsymbol{a} \leftarrow \sigma(\boldsymbol{wX})$$

$$\frac{\partial J}{\partial w} = \frac{1}{m}[\boldsymbol{a} - Y]X^T$$

$$w \leftarrow w - \alpha\frac{\partial J}{\partial w}$$

}



Steepest Gradient Descend Method

- Data:
  - Dataset: $D=\{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1,m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a=\boldsymbol{\sigma}(\boldsymbol{wx})=\sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

Vector form

$$\boldsymbol{w}(\boldsymbol{k}+\boldsymbol{1})=\boldsymbol{w}(\boldsymbol{k})-\alpha_k \frac{1}{m}[\boldsymbol{a}-Y]X^T$$

Component form

$$w_j(k+1)=w_j(k)-\alpha_k \frac{1}{m}\sum_{i=1}^{m}(a^i-y^i)\cdot x_j^i$$

- Data:
  - Dataset: $D=\{(\boldsymbol{x}^i \in R^n, y^i \in R) | i \in [1,m]\}$, where m is the number of the samples.
- Linear Regression Model:

$$a=\boldsymbol{wx}=\sum_{j=1}^{n} w_j \cdot x_j^i$$

Vector form

$$\boldsymbol{w}(\boldsymbol{k}+\boldsymbol{1})=\boldsymbol{w}(\boldsymbol{k})-\alpha_k \frac{1}{m}[\boldsymbol{a}-Y]X^T$$

Component form

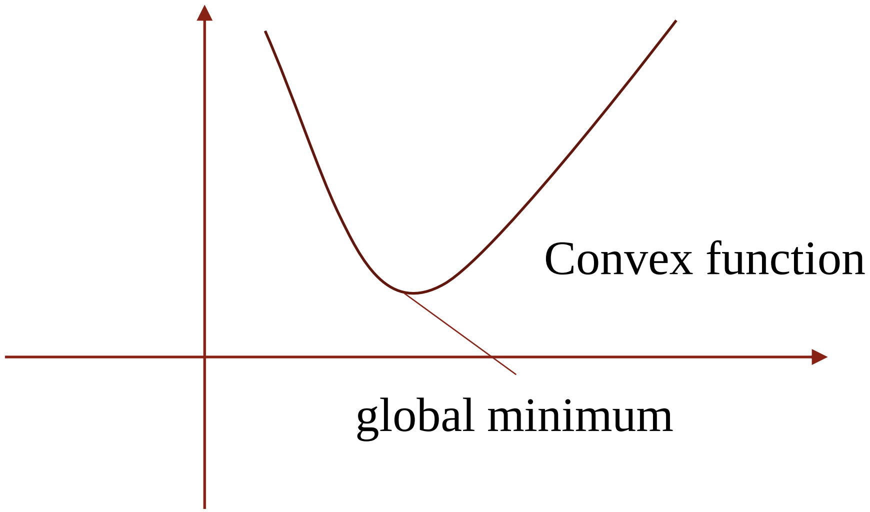
$$w_j(k+1)=w_j(k)-\alpha_k \frac{1}{m}\sum_{i=1}^{m}(a^i-y^i)\cdot x_j^i$$

Cross-entropy vs Mean Square Error

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right) \qquad \text{where } \sigma(z) = \frac{1}{1+e^{-z}}$$

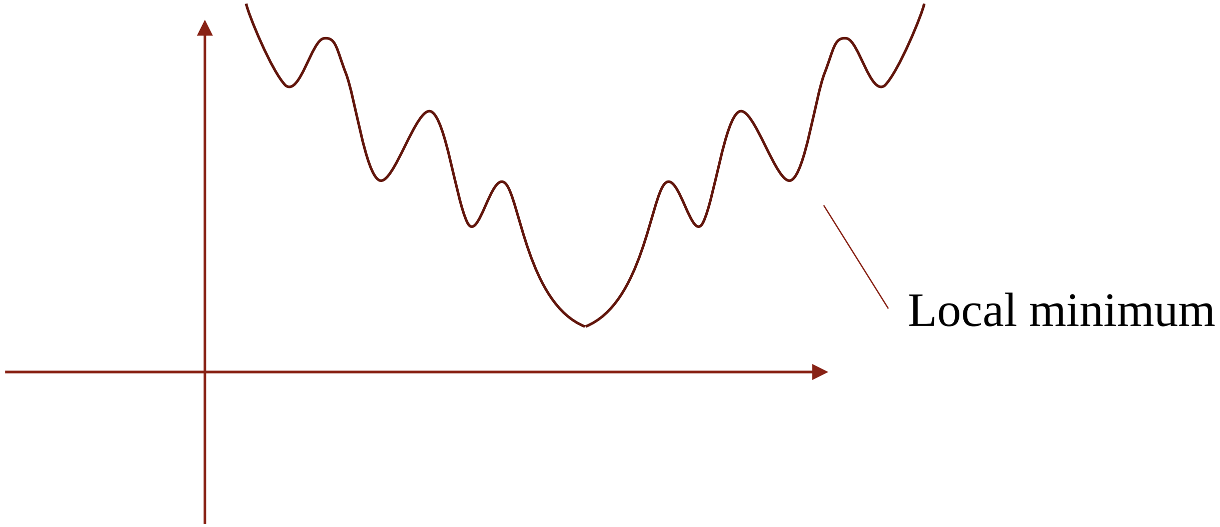
Cross-entropy

$$\boldsymbol{J}(\boldsymbol{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \cdot \log\left(a^i\right) + \left(1 - y^i\right) \cdot \log(1 - a^i)\right]$$



Mean Square Error

$$\boldsymbol{J}(\boldsymbol{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left(a^i - y^i\right)^{\boldsymbol{2}}$$



Cross-entropy cost function

If Hessian Matrix $\nabla^2 J(\mathrm{w})$ is positive determinate matrix, then, $J(w)$ is a convex function.

$$\nabla \ J(\mathrm{w}) = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \cdots, \frac{\partial J}{\partial w_n}\right]$$

$$H = \begin{bmatrix} \frac{\partial^2 J}{\partial w_1 \partial w_1} & \frac{\partial^2 J}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_1 \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_1 \partial w_n} \\ \frac{\partial^2 J}{\partial w_2 \partial w_1} & \frac{\partial^2 J}{\partial w_2 \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_2 \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_2 \partial w_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial w_p \partial w_1} & \frac{\partial^2 J}{\partial w_p \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_p \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_p \partial w_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial w_n \partial w_1} & \frac{\partial^2 J}{\partial w_n \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_n \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_n \partial w_n} \end{bmatrix}_{n \times n}$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{\partial \frac{\partial J}{\partial w_p}}{\partial w_q}$$

Cross-entropy cost function

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{\partial \frac{\partial J}{\partial w_p}}{\partial w_q}$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m} \frac{\partial \sum_{i=1}^{m} (a^i - y^i) \cdot x_p^i}{\partial w_q}$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial a^i}{\partial w_q} \cdot x_p^i$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial a^i}{\partial \sum_{l=1}^{n} w_l \cdot x_l^i} \cdot \frac{\partial \sum_{l=1}^{n} w_l \cdot x_l^i}{\partial w_q} x_p^i$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m} \sum_{i=1}^{m} a^i (1 - a^i) \cdot x_q^i \cdot x_p^i$$

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m} \sum_{i=1}^{m} a^i \left(1 - a^i\right) \cdot x_p^i \cdot x_q^i$$

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z) \cdot (1 - \sigma(z))$$

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (a^i - y^i) \cdot x_j^i$$

Cross-entropy cost function

$$\frac{\partial^2 J}{\partial w_p \partial w_q} = \frac{1}{m}\sum_{i=1}^{m} a^i(1-a^i)\cdot x_p^i \cdot x_q^i$$

$$H_{pq} = \frac{1}{m}\sum_{i=1}^{m} a^i(1-a^i)\cdot x_p^i \cdot x_q^i$$

$$\Downarrow$$

$$H = \frac{1}{m}\sum_{i=1}^{m} a^i(1-a^i)\cdot \boldsymbol{x}^i\ (\boldsymbol{x}^i)^T$$

$\Downarrow$ $\boldsymbol{x}^i\ (\boldsymbol{x}^i)^T$ *is* positive determinate matrix

Hessian Matrix H *is* positive determinate matrix,
then, $J(w)$ is a convex function.

$$H = \begin{bmatrix} \frac{\partial^2 J}{\partial w_1 \partial w_1} & \frac{\partial^2 J}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_1 \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_1 \partial w_n} \\ \frac{\partial^2 J}{\partial w_2 \partial w_1} & \frac{\partial^2 J}{\partial w_2 \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_2 \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_2 \partial w_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial w_p \partial w_1} & \frac{\partial^2 J}{\partial w_p \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_p \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_p \partial w_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial w_n \partial w_1} & \frac{\partial^2 J}{\partial w_n \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_n \partial w_q} & \cdots & \frac{\partial^2 J}{\partial w_n \partial w_n} \end{bmatrix}_{n\times n}$$

$$\begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_p^i \\ \vdots \\ x_n^i \end{bmatrix} \quad [x_1^i, x_2^i, \cdots, x_q^i, \cdots x_n^i]$$

Steepest Gradient Descend Method

- Data:
  - Sample: $(\boldsymbol{x} \in R^n, y \in \{0,1\})$, where n is the *dimension* of the input vector $\boldsymbol{x}$.
  - Dataset: $D = \{(\boldsymbol{x}^i \in R^n, y^i \in \{0,1\}) | i \in [1, m]\}$, where m is the number of the samples.
- Logistic Regression Model:

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

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

- Object:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \boldsymbol{J}(\boldsymbol{w}) = \underset{\boldsymbol{w}}{\operatorname{argmin}} -\frac{1}{m}\sum_{i=1}^{m}\left[y^i \cdot \log(a^i) + (1 - y^i) \cdot \log(1 - a^i)\right]$$

Steepest Descent Algorithm

Input: D, w

for k in $1,2,\ldots,K$:

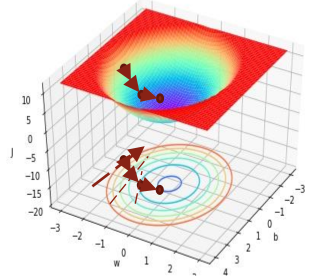
{

for i in 1,2,…,m:

{

$$a^i \leftarrow \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

}



for j in 1,2,…, n:

{

$$\frac{\partial J}{\partial w_j} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(a^i - y^i) \cdot x_j^i$$

}

for j in 1,2,…, n:

{

$$w_j \leftarrow w_j - \alpha \frac{\partial J}{\partial w_j}$$

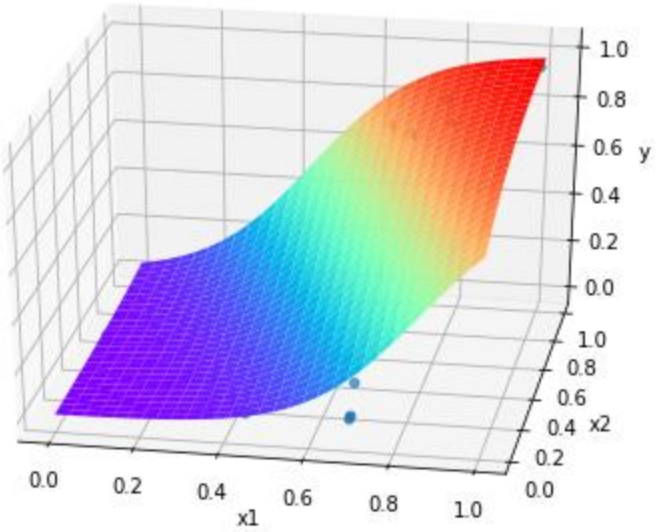
}

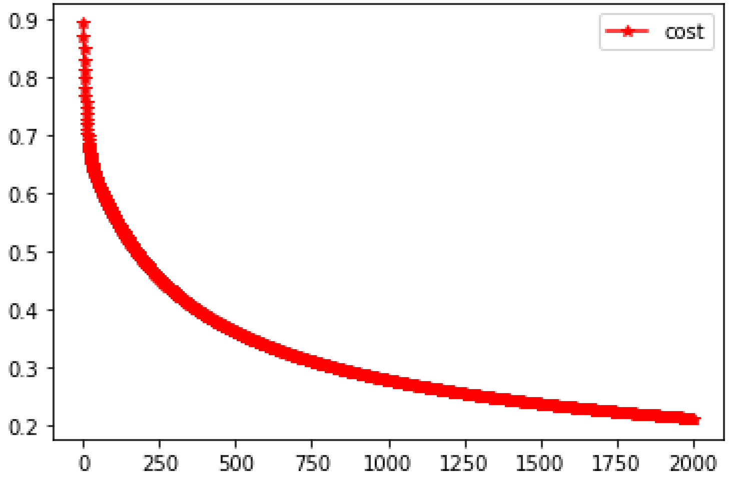
}

Examples

Size x_1	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color x_2	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Label y	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0

$\boldsymbol{w}$=[7.52, 3.19, -6.46]



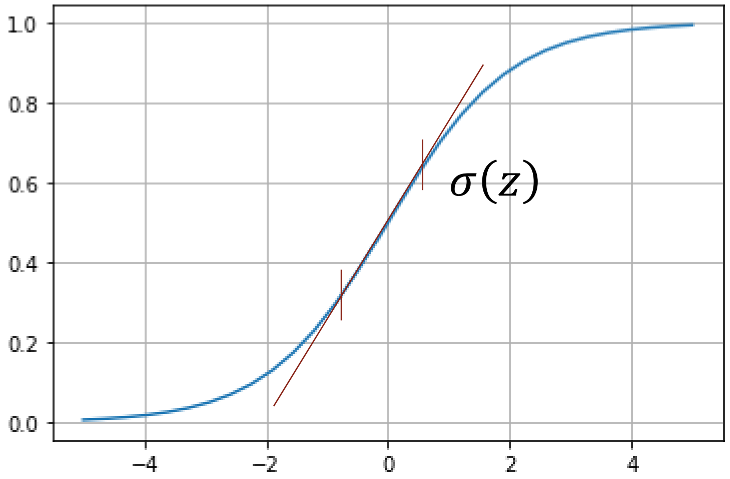


Decision Line

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

If $z_1 > z_2$ then $\sigma(z_1) > \sigma(z_2)$

If $\sigma(z_1) > \sigma(z_2)$ then $z_1 > z_2$



Threshold classifier output a at τ :

if $a \geq \tau$, predict "y=1"

if $a < \tau$, predict "y=0".

For example $\tau = 0.5$

$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

Threshold classifier output a at 0.5 :

if $a \geq 0.5$, predict "y=1"

if $a < 0.5$, predict "y=0".

$\Longleftrightarrow$

Threshold classifier output a at 0.5 :

if $\boldsymbol{wx} \geq 0$, predict "y=1"

if $\boldsymbol{wx} < 0$, predict "y=0".

$\boldsymbol{wx}$ =0 is called the decision line

Decision Line

Size $\boldsymbol{x_1}$	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color $\boldsymbol{x_2}$	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Label y	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0

$\boldsymbol{w}$=[7.52, 3.19, -6.46]

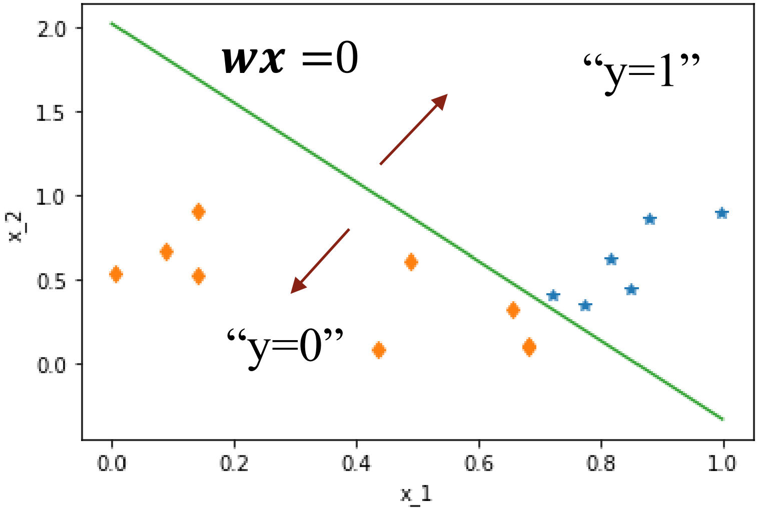
$$a = \boldsymbol{\sigma}(\boldsymbol{wx}) = \sigma\left(\sum_{j=1}^{n} w_j \cdot x_j^i\right)$$

Threshold classifier output a at τ :

if $a \geq \tau$, predict "y=1"

if $a < \tau$, predict "y=0".

For example $\tau = 0.5$



Threshold classifier output a at 0.5 :

if $\boldsymbol{wx} \geq 0$, predict "y=1"

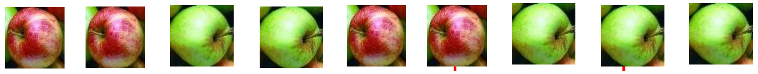
if $\boldsymbol{wx} < 0$, predict "y=0".

Accuracy

$$\text{Accuracy} = \frac{\text{number of correct prediction}}{\text{number of samples}}$$

An example

Tested data 9



Prediction 1 1 0 0 1 0 0 1 0

Correct prediction
7

Incorrect prediction
2

$$\text{Accuracy} = \frac{7}{9} = 77.78\%$$

Threshold classifier output a at τ :
if $a \geq \tau$, predict “y=1”
if $a < \tau$, predict “y=0”.

For example $\tau = 0.5$

Test on training set:

- Reflect the progress of training.
- Evaluate the ability of the model to fit given data.

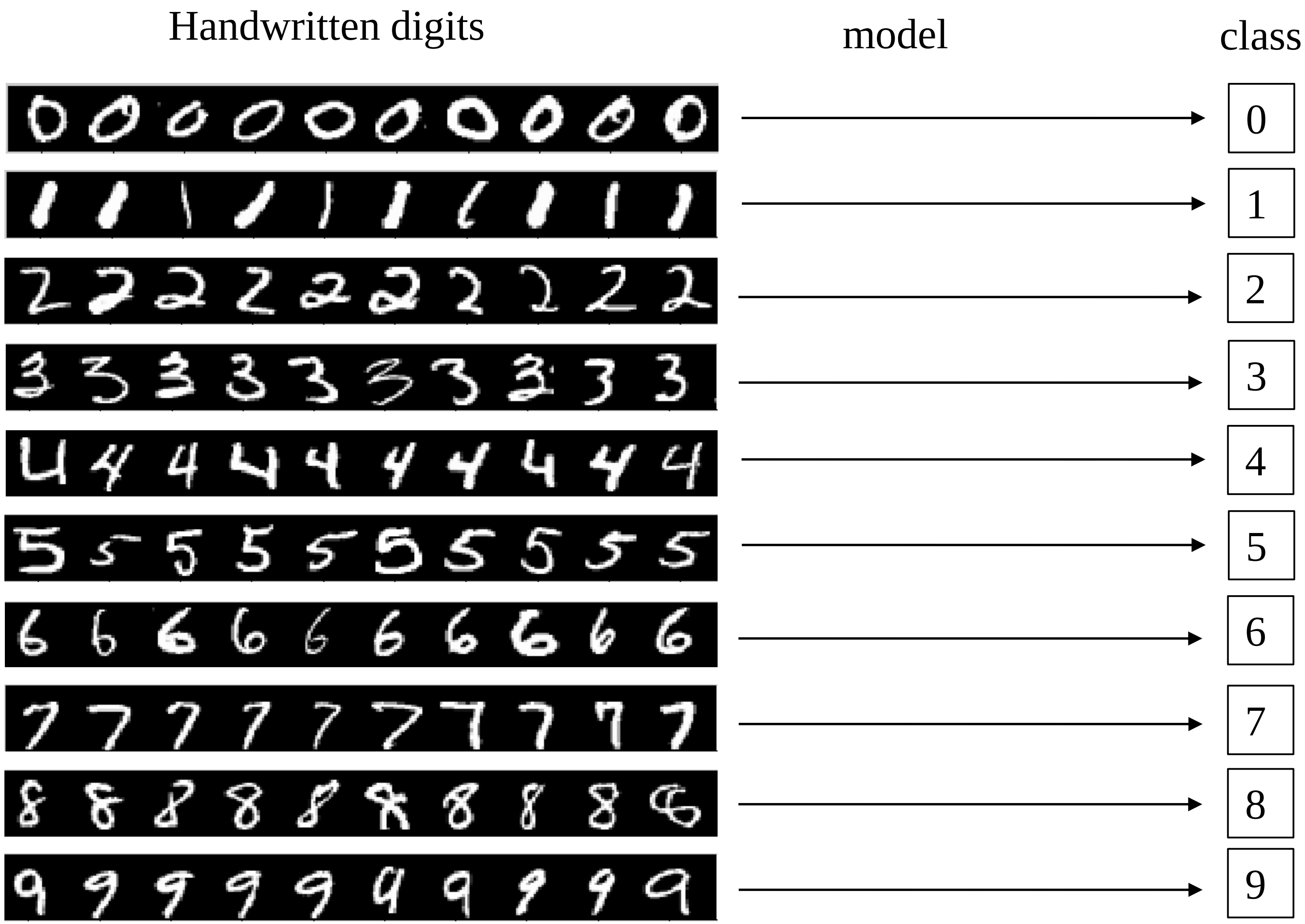
Test on testing set:

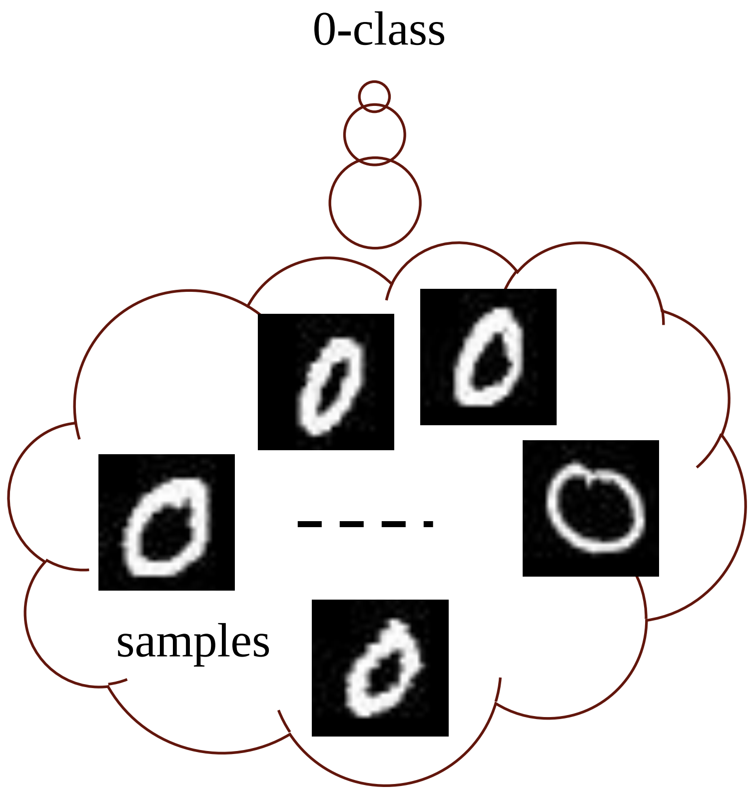
- Evaluate the ability of the model to generalize the knowledge.

Outline

- Brief review
- Binary Classification
- Multi-class classification

Examples


Handwritten digits
model
class
0
1
2
3
4
5
6
7
8
9


0-class
samples

Examples

Size x_1	0.488	0.681	0.655	0.088	0.139	0.721	0.999	0.775	0.881	0.007	0.14	0.817	0.681	0.851	0.436
Color x_2	0.609	0.112	0.324	0.669	0.91	0.41	0.902	0.353	0.865	0.539	0.524	0.62	0.093	0.447	0.088
Class sweet	**medium** 1	**medium** 1	**medium** 1	**low** 0	**medium** 1	**medium** 1	**high** 2	**high** 2	**high** 2	**low** 0	low 0	high 2	medium 1	high 2	low 0

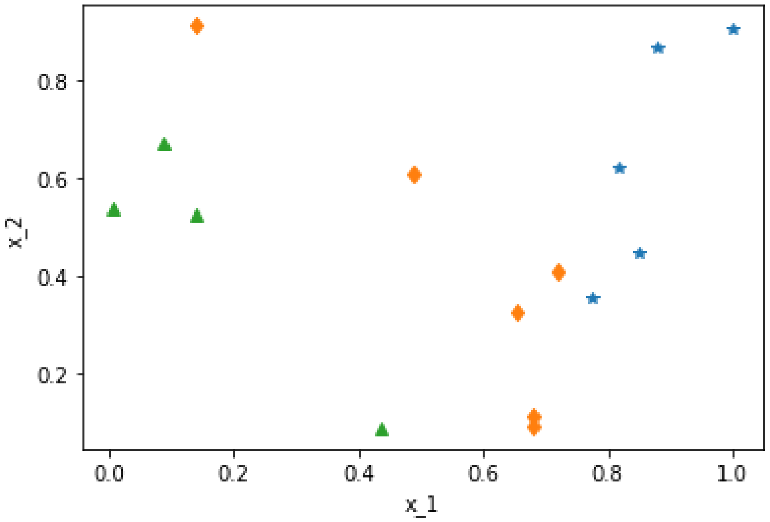
Model

Sweet

Low- 0

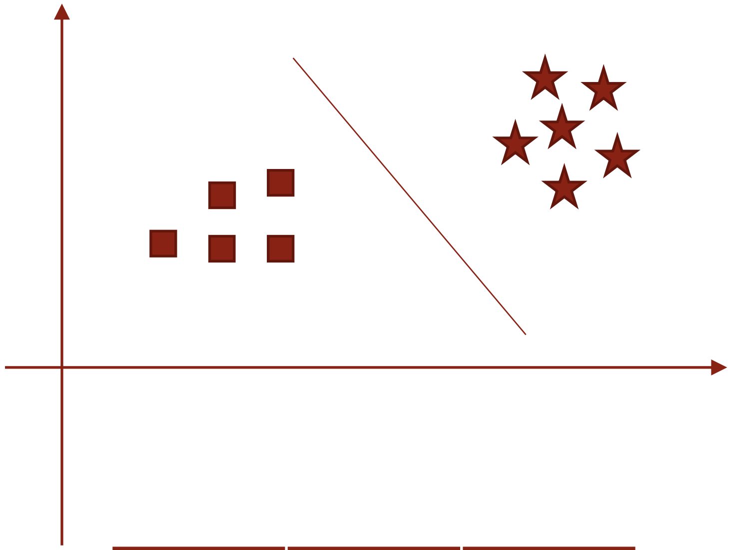
Medium -1

High- 2



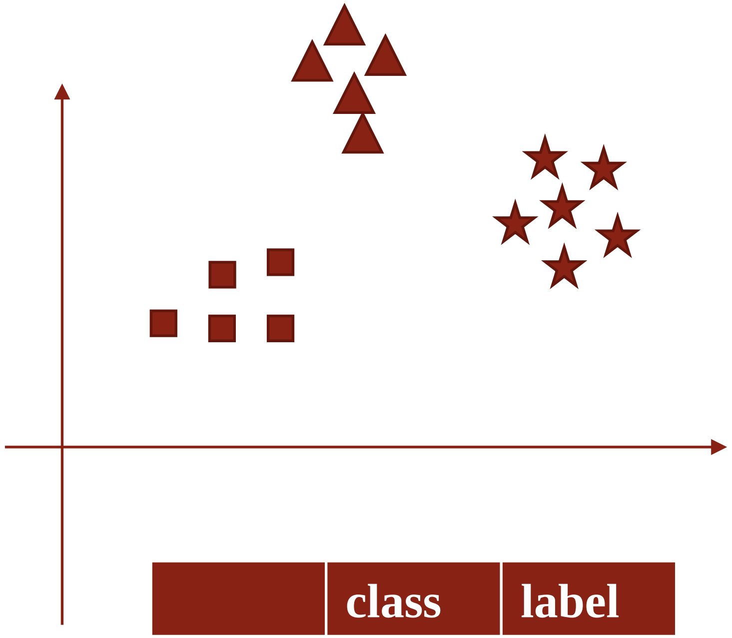
Examples

Binary classification



	class	label
★	1	0
■	2	1

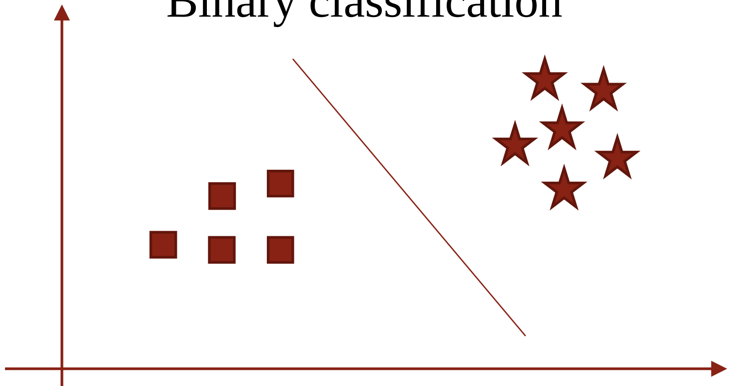
Multi-class classification



	class	label
★	1	?
■	2	?
▲	3	?

Examples

Binary classification



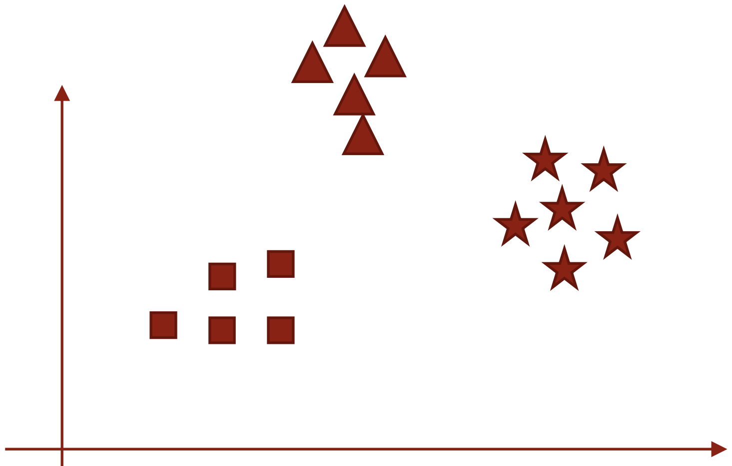
	class	label
★	1	0
■	2	1

Threshold classifier output a at τ :

if $a \geq \tau$, predict "y=1"

if $a < \tau$, predict "y=0".

Multi-class classification



	class	label
★	1	?
■	2	?
▲	3	?

How to interpret the output a ?

Thanks!!!