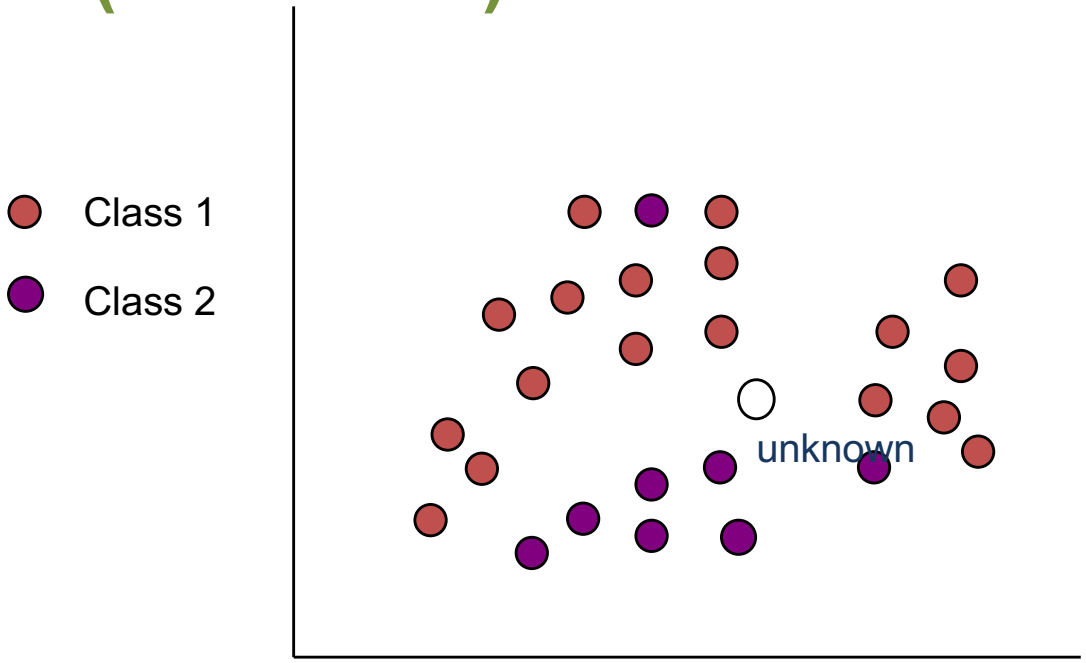
Logistic Regression

Classification

Learn a method for predicting the instance class from pre-labeled (classified) instances



Many approaches: Regression, Decision Trees, Nearest Neighbor, Support Vector Machines, Neural Networks,

...

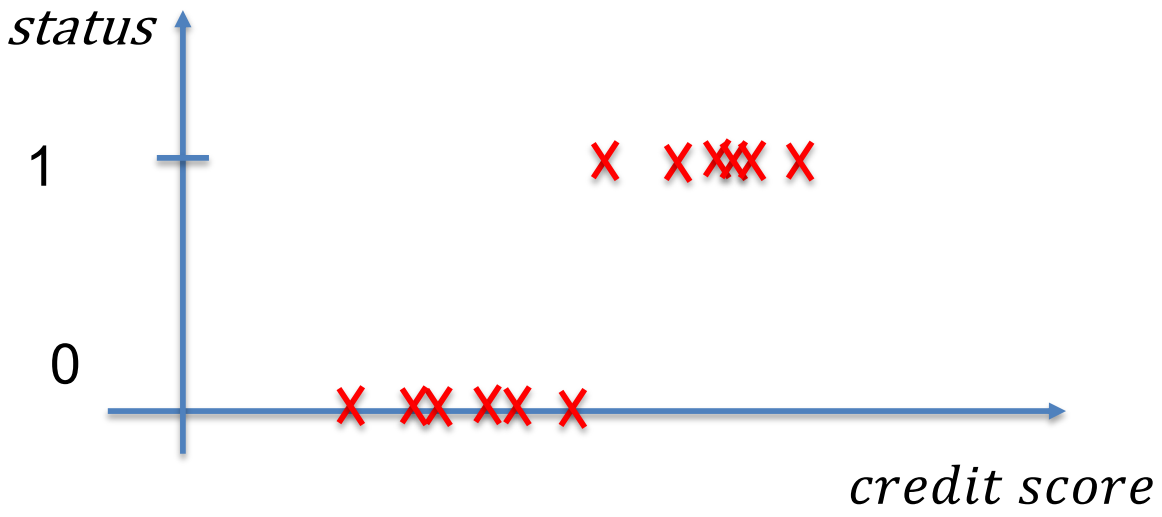
Classification Problem

Loan approval problem with a single variable

x_1: credit score (FICO score)

y: 1-approve, 0-deny

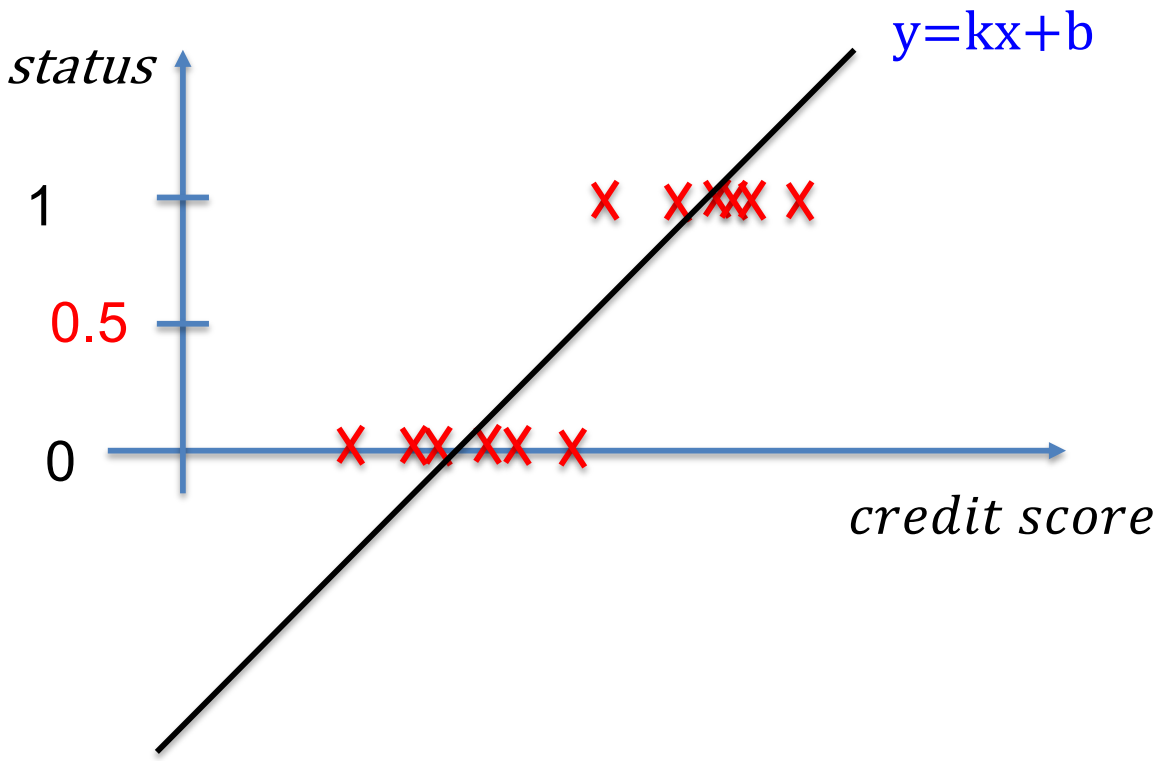
Credit Score	Loan Status
750	1
725	0
700	0
650	0
726	1
645	0
800	1
…	…



Classification Problem

Loan approval problem with a single variable

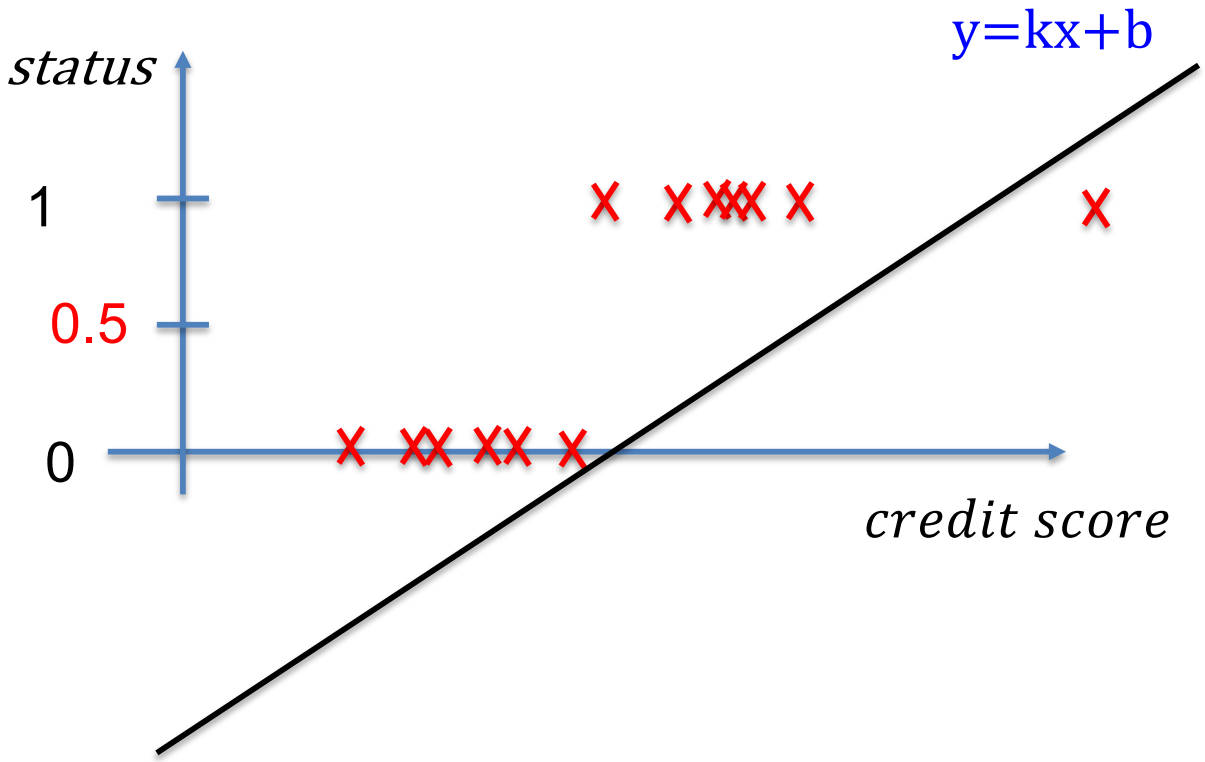
Credit Score	Loan Status
750	1
725	0
700	0
650	0
726	1
645	0
800	1
…	…



Classification Problem

Loan approval problem with a single variable

Credit Score	Loan Status
750	1
725	0
700	0
650	0
726	1
645	0
800	1
…	…



Classification Problem

Loan approval problem

x_1: credit score (FICO score)
x_2: income
(may include other features)

y: 1-approve, 0-deny

Training Data

Credit Score	Income	Loan Status
750	113000	1
725	26000	0
700	54000	0
650	45000	0
726	89500	1
645	78500	0
800	87050	1
...	...	...

Test data:
for a new applicant with credit score 715 and
income 68500, will the loan application be approved?

Binary Classification Data

Given:

Training data set:

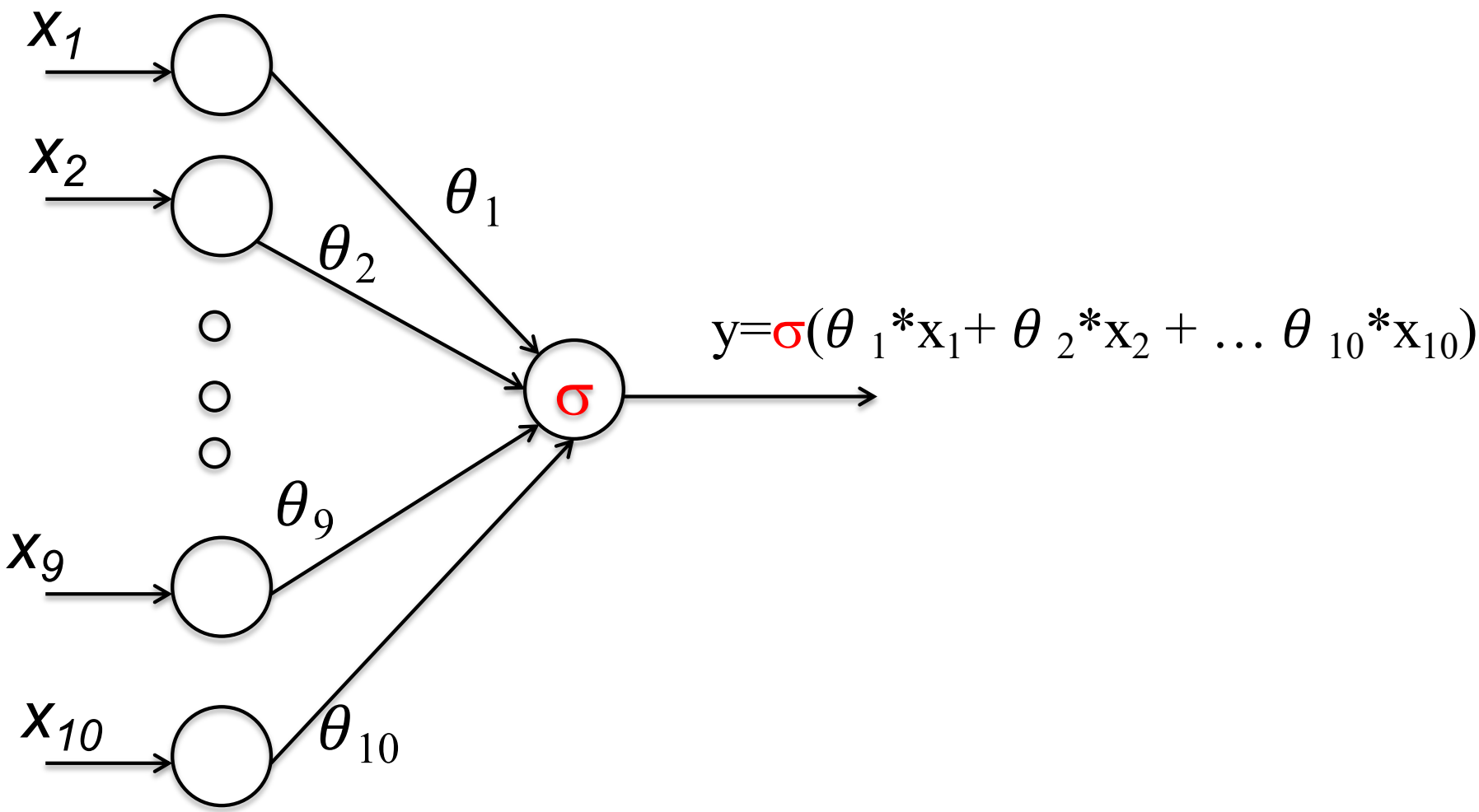
$\{ \{x^1, y^1\},$
$\{x^2, y^2\},$
$\{x^3, y^3\},$
...
$\{x^m, y^m\}\}$

m examples

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$$

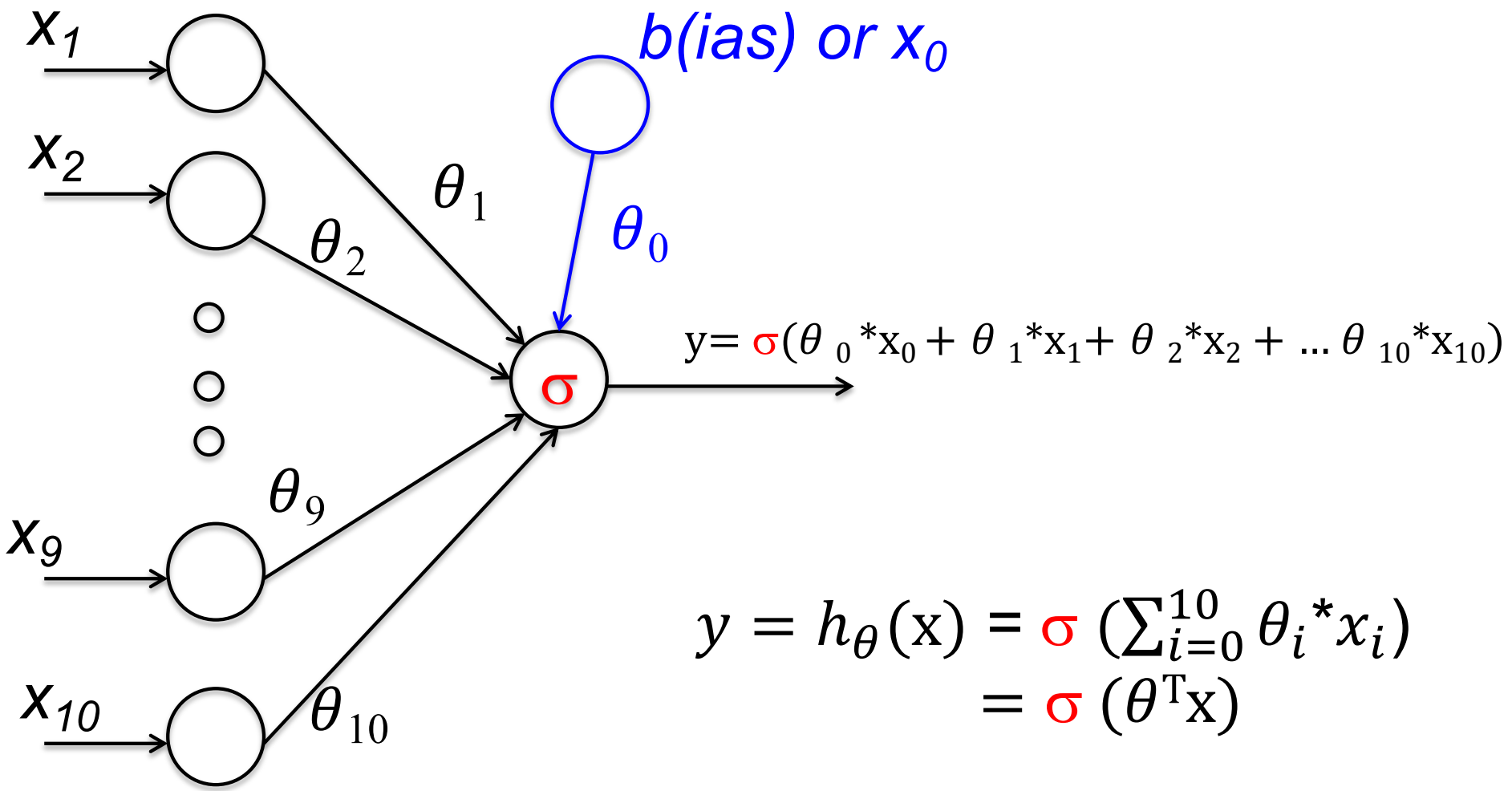
x_0=1, y$\in \{0, 1\}$

Logistic Regression For Binary Classification



10 features

Logistic Regression For Binary Classification



$$y = h_\theta(\mathrm{x}) = \sigma\left(\sum_{i=0}^{10} \theta_i{}^* x_i\right)$$
$$= \sigma\left(\theta^{\mathrm{T}}\mathrm{x}\right)$$

10 features

Activation Function σ

- Tanh()

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

$$f'(x) = 1 - f(x)^2$$

- Sigmoid/Logistic

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

$$f'(x) = f(x)[1 - f(x)]$$

- Bipolar Sigmoid

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

$$f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)]$$

Derivatives

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

quotient rule

$$\frac{d}{dx}\sigma(x) = \frac{0 \cdot (1+e^{-x}) - (1) \cdot (e^{-x} \cdot (-1))}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1+e^{-x}-1}{(1+e^{-x})^2}$$

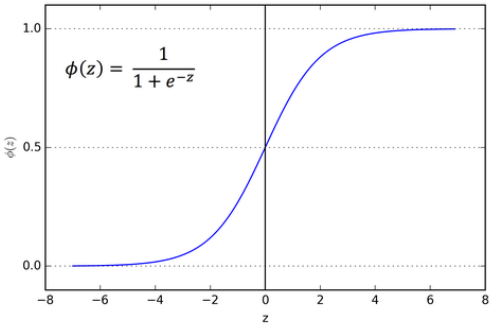
$$= \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2}$$

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

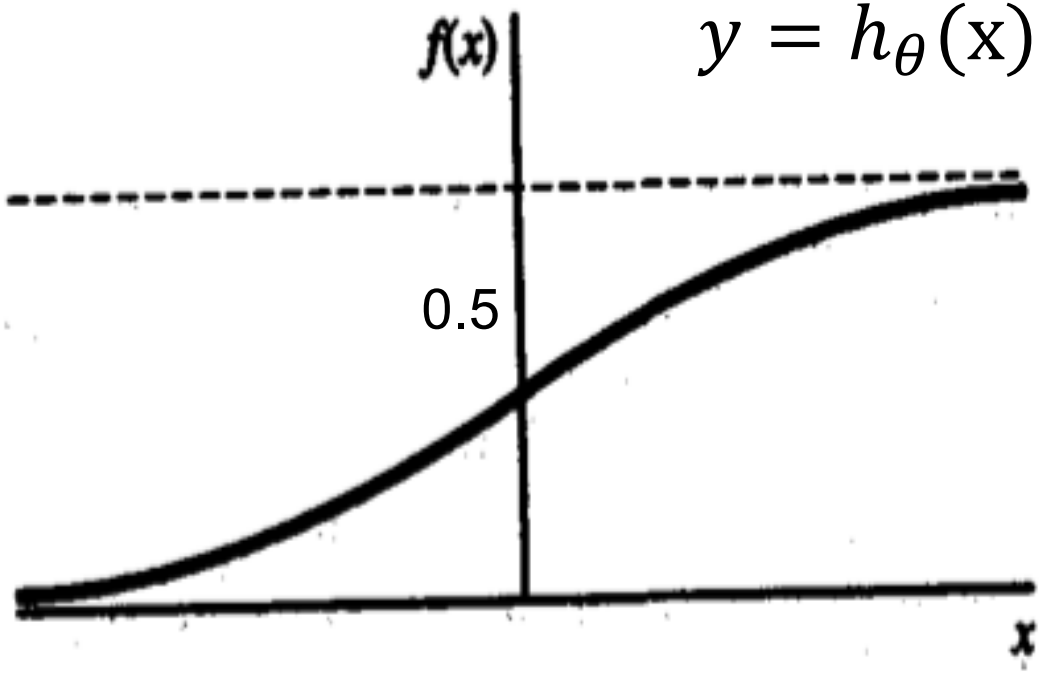
$$= \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right)$$

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

Sigmoid Function for Classification



$$y = h_\theta(\mathrm{x}) = \sigma(\theta^{\mathrm{T}}\mathrm{x})$$



if $\sigma(\theta^{\mathrm{T}}\mathrm{x})<0.5$,
predict class 0

if $\sigma(\theta^{\mathrm{T}}\mathrm{x})>0.5$,
predict class 1

($\theta^{\mathrm{T}}\mathrm{x} < 0$,
predict class 0)

($\theta^{\mathrm{T}}\mathrm{x} \geq 0$,
predict class 1)

Logistic Regression Model

- Logistic Regression assumes:

$$p(y=1 \mid x, \theta) = \sigma(\theta^T x)$$

(e.g., the probability that the label is 1, given input x, follows a sigmoid of a linear function.)

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

$estimate\ the\ probablity\ that\ y = 1\ given\ input\ x$

$$p(y=1 \mid x, \theta)$$

$$p(y=0 \mid x, \theta) = 1 - p(y=1 \mid x, \theta)$$

How to use it in credit assignment or medical diagnosis problems?

Estimate the Parameters θ

Given:

Training data set:

$\{\{x^1, y^1\},$
$\{x^2, y^2\},$
$\{x^3, y^3\},$
...
$\{x^m, y^m\}\}$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$$

x_0=1, y$\in \{0, 1\}$

m examples

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

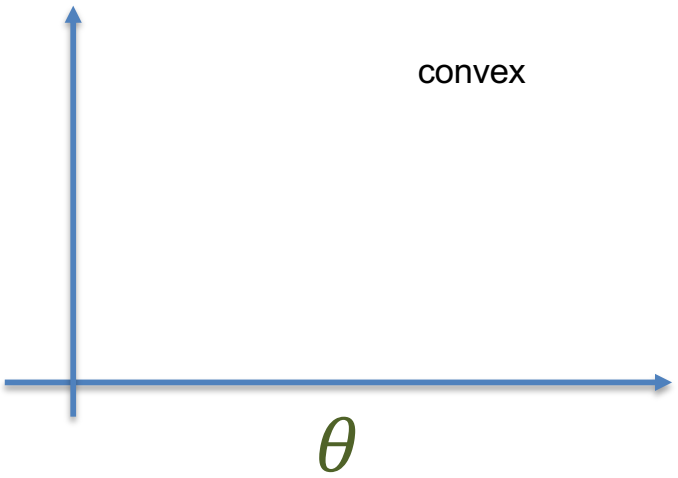
How to estimate the parameters θ from data?

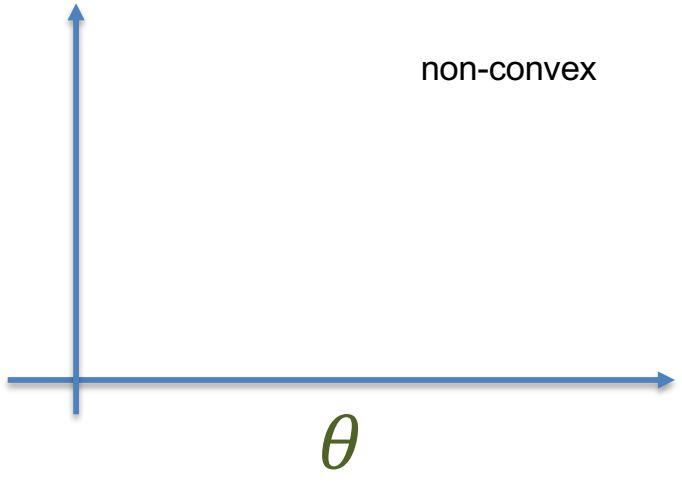
Cost Function

- Linear Regression:

$$loss\ function: J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

- In logistic regression, $(h_\theta(x^{(i)}) - y^{(i)})^2$ is not a convex curve, not suitable for gradient descent approximation approach.





Cost Function

Logistic regression assumes:

$$h_\theta(x) = P(y = 1 \mid x) = \sigma(\theta^T x)$$

Since $y \in \{0, 1\}$, this is a **Bernoulli random variable**.

So:

- $P(y = 1|x) = h_\theta(x)$
- $P(y = 0|x) = 1 - h_\theta(x)$

These two cases can be combined into one formula:

$$P(y|x;\theta) = \big(h_\theta(x)\big)^y \big(1 - h_\theta(x)\big)^{1-y}$$

Check:

- If $y = 1$: becomes $h_\theta(x)$
- If $y = 0$: becomes $1 - h_\theta(x)$

Cost Function

For training example (x_i, y_i), the likelihood is:

$$L(\theta) = P(y_i \mid x_i; \theta)$$

Using Bernoulli form:

$$\boxed{L(y_i \mid x_i, \theta) = \big(h_\theta(x_i)\big)^{y_i} \big(1 - h_\theta(x_i)\big)^{1-y_i}}$$

We maximize likelihood — equivalently, maximize **log-likelihood**:

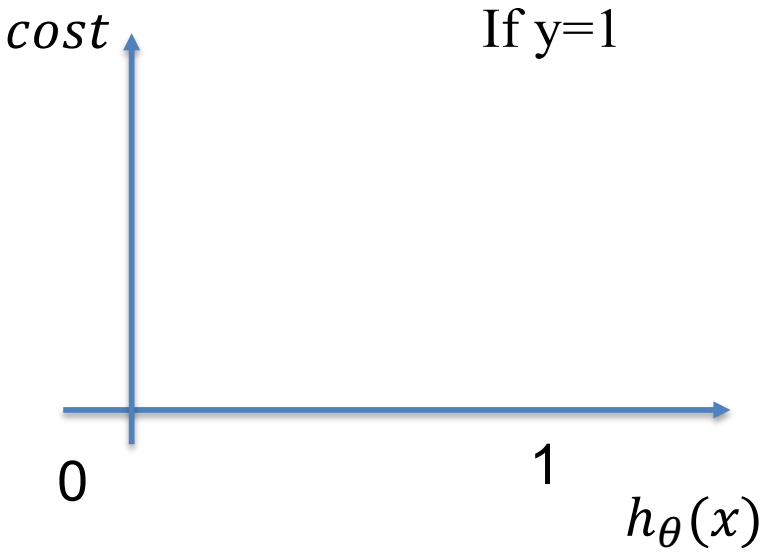
$$\log L(\theta) = y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

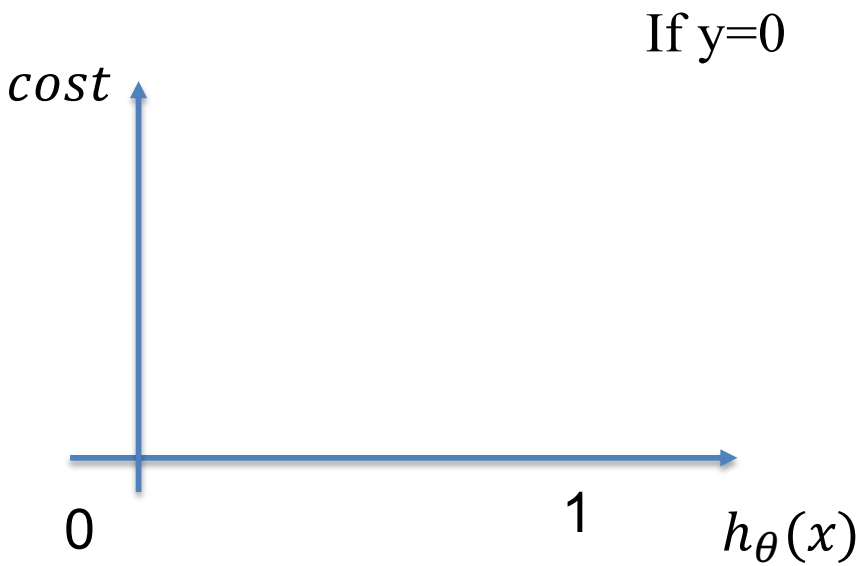
Turn maximum log-likelihood into minimizing cost (negative log-likelihood):

$$\text{Cost} = -\Big[y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\Big]$$

Logistic Regression Cost Function

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1 - h_\theta(x)) & \text{if } y=0 \end{cases}$$





Gradient Descent

- To minimize the Cost function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log\, h_\theta(x^{(i)}) + (1-y^{(i)})\log(1 - h_\theta(x^{(i)}))]$$

- To minimizing the cost function over the entire data set
  - Generally, there is no closed form solution for this minimization problem, except for special cases
  - Approach: Gradient descent

Repeat for each iteration:

$$\theta_j := \theta_j - \lambda \frac{\partial}{\partial \theta_j} J(\theta)$$

where: $\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$ *derivation at the end*

Weight Updates with Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log h_\theta(x^{(i)}) + (1\text{-}y^{(i)})\log(1 - h_\theta(x^{(i)}))]$$

Want to minimize $J(\theta)$:

Repeat for each iteration:

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

Simultaneously update all θ_j

α: Learning Rate → step size

Gradient Descent -- A Toy Problem

Data (m = 1):

$$x^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (\text{bias } x_0 = 1,\ x_1 = 2), \qquad y^{(1)} = 1$$

Initialize:

$$\theta^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \alpha = 0.1$$

Hypothesis:

$$h_\theta(x) = \sigma(\theta^T x), \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$z^{(1)} = \theta^T x^{(1)} = 0 \cdot 1 + 0 \cdot 2 = 0 \qquad h^{(1)} = \sigma(0) = 0.5$$

$$h^{(1)} - y^{(1)} = 0.5 - 1 = -0.5$$

Gradient Descent -- A Toy Problem

Compute gradients for each parameter

$$\frac{\partial J}{\partial \theta_j} = (h - y)x_j$$

$$\frac{\partial J}{\partial \theta_0} = (-0.5) \cdot 1 = -0.5$$

$$\frac{\partial J}{\partial \theta_1} = (-0.5) \cdot 2 = -1.0$$

$$\nabla J = \begin{bmatrix} -0.5 \\ -1.0 \end{bmatrix}$$

Gradient descent update

$$\theta := \theta - \alpha \nabla J$$

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} -0.5 \\ -1.0 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.10 \end{bmatrix}$$

$$z^{(1)}_{\text{new}} = \theta^{(1)T} x^{(1)} = 0.05 \cdot 1 + 0.10 \cdot 2 = 0.25$$

$$h_{\text{new}} = \sigma(0.25) \approx 0.562$$

Cross Entropy Error Cost Function

- Logistic Regression Error
  - 0 if correct, >0 if not correct, more wrong → bigger cost
- Cross-Entropy Error cost function

$$Cost(h_\theta(x), \mathrm{y}) = -y * \log(h_\theta(x)) - (1\text{-}\mathrm{y})^* \log(1 - h_\theta(x))$$

y is the target, $h_\theta(x)$ is the predicted value

y	$h_\theta(x)$	cost
1	1	0
0	0	0
1	0.9	0.11
1	0.5	0.69
1	0.1	2.3

The Derivative of Cost Function for Logistic Regression (1) (for math enthusiasts)

Hypothesis (sigmoid):

$$h_\theta(x) = \sigma(z), \quad z = \theta^T x$$

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

Cost function (average negative log-likelihood / binary cross-entropy):

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

We want:

$$\frac{\partial J(\theta)}{\partial \theta_j}$$

The Derivative of Cost Function for Logistic Regression (2)

$$\frac{\partial J}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{\partial}{\partial \theta_j}\log h_\theta(x^{(i)}) + (1-y^{(i)})\frac{\partial}{\partial \theta_j}\log(1-h_\theta(x^{(i)}))\right]$$

Use:

$$\frac{d}{du}\log u = \frac{1}{u}$$

So:

$$\frac{\partial}{\partial \theta_j}\log h_\theta(x^{(i)}) = \frac{1}{h_\theta(x^{(i)})}\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j}$$

and

$$\frac{\partial}{\partial \theta_j}\log(1-h_\theta(x^{(i)})) = \frac{1}{1-h_\theta(x^{(i)})}\frac{\partial (1-h_\theta(x^{(i)}))}{\partial \theta_j}$$

But

$$\frac{\partial (1-h)}{\partial \theta_j} = -\frac{\partial h}{\partial \theta_j}$$

So:

$$\frac{\partial}{\partial \theta_j}\log(1-h_\theta(x^{(i)})) = -\frac{1}{1-h_\theta(x^{(i)})}\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j}$$

The Derivative of Cost Function for Logistic Regression (3)

So, we get:

$$\frac{\partial J}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\frac{1}{h^{(i)}}\frac{\partial h^{(i)}}{\partial \theta_j} - (1-y^{(i)})\frac{1}{1-h^{(i)}}\frac{\partial h^{(i)}}{\partial \theta_j}\right]$$

$$h^{(i)} = h_\theta(x^{(i)}).$$

$$\frac{\partial J}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[\left(\frac{y^{(i)}}{h^{(i)}} - \frac{1-y^{(i)}}{1-h^{(i)}}\right)\frac{\partial h^{(i)}}{\partial \theta_j}\right]$$

The Derivative of Cost Function for Logistic Regression (4)

Compute $\frac{\partial h^{(i)}}{\partial \theta_j}$

Given: $h^{(i)} = \sigma(z^{(i)}), \quad z^{(i)} = \theta^T x^{(i)}$

We have: $$\frac{\partial h^{(i)}}{\partial \theta_j} = \sigma'(z^{(i)}) \frac{\partial z^{(i)}}{\partial \theta_j}$$

Since: $$\sigma'(z^{(i)}) = \sigma(z^{(i)})(1 - \sigma(z^{(i)})) = h^{(i)}(1 - h^{(i)})$$

$$\frac{\partial z^{(i)}}{\partial \theta_j} = x_j^{(i)}$$

We have:

$$\frac{\partial h^{(i)}}{\partial \theta_j} = h^{(i)}(1 - h^{(i)}) x_j^{(i)}$$

The Derivative of Cost Function for Logistic Regression (5)

Substitute into the earlier expression:

$$\frac{\partial J}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left(\frac{y^{(i)}}{h^{(i)}} - \frac{1-y^{(i)}}{1-h^{(i)}}\right) h^{(i)}(1-h^{(i)})x_j^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}}{h^{(i)}}h^{(i)}(1-h^{(i)}) - \frac{1-y^{(i)}}{1-h^{(i)}}h^{(i)}(1-h^{(i)})\right] x_j^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\Big[\underbrace{y^{(i)}(1-h^{(i)}) - (1-y^{(i)})h^{(i)}}_{y^{(i)} - y^{(i)}h^{(i)} - h^{(i)} + y^{(i)}h^{(i)}}\Big] x_j^{(i)}$$

$$y^{(i)} - h^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h^{(i)})x_j^{(i)}$$

$$\boxed{\frac{\partial J}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h^{(i)} - y^{(i)})x_j^{(i)}}$$