Data Mining

Classification
1. Logistic Regression

Classification and Regression

- Classification:
  - predicts categorical class labels
  - Constructs classification models based on training data and uses the models in classifying new data
- Regression:
  - models continuous-valued functions, i.e., predicts unknown or missing numeric values
- Example Applications
  - credit approval- classify loan application by their likelihood of defaulting on payments
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

Classification Applications

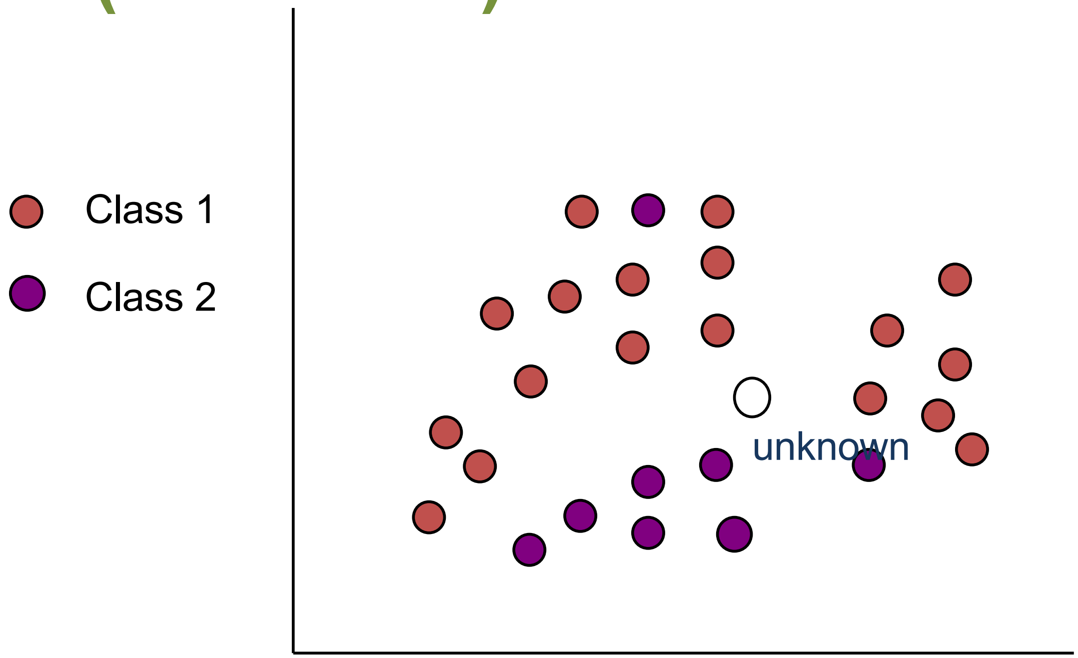
- Example Applications (continued)
  - Image processing : interpretation of digital images in radiology, recognizing 3-D objects, outdoor image segmentation
  - Language processing : text classification
  - Software development : estimate the development effort of a given software module
  - Pharmacology: drug analysis
  - Molecular biology : analyzing amino acid sequences
  - Medicine : cardiology, analyzing sudden infant death syndrome, diagnosing thyroid disorder
  - Manufacturing : classify equipment malfunctions by their cause

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction: training set
  - The model is represented as classification rules, decision trees, mathematical formulae, neural networks, or an ensemble of these
- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The set of tuples used for testing the performance of the model: test data
    - The known label of test sample is compared with the classified result from the model
    - Accuracy rate is the percentage of test set samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur

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

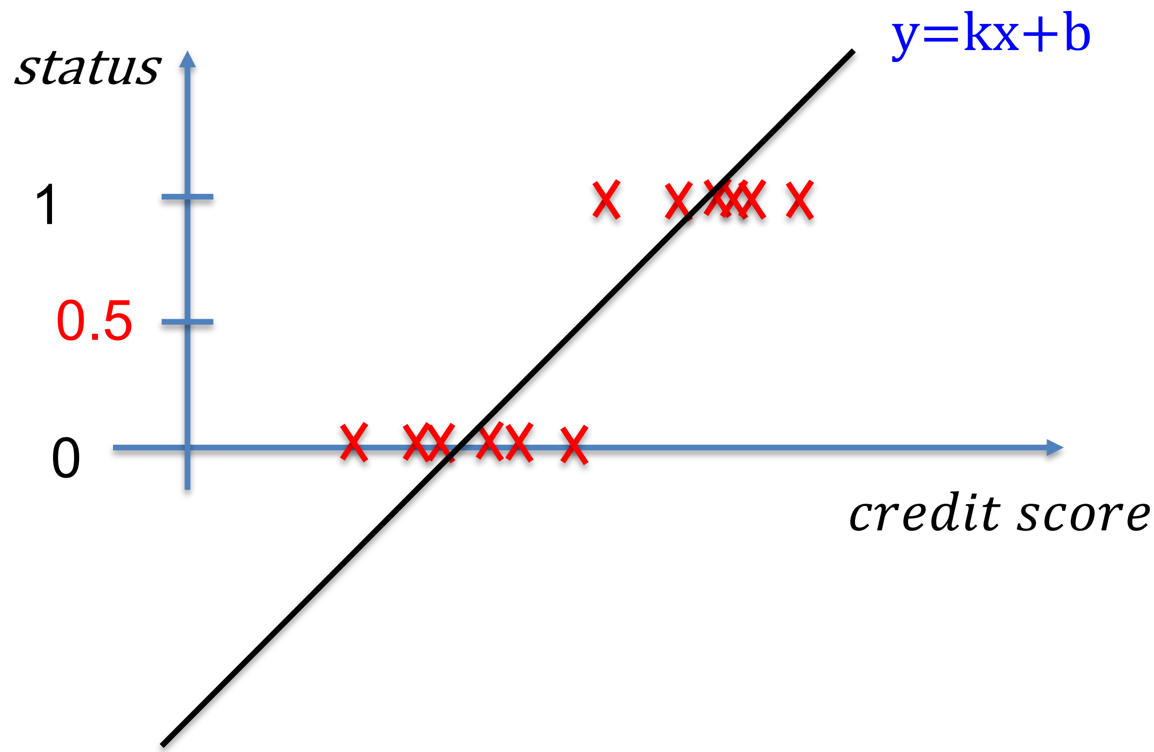
x_1: credit score (FICO score)
y: 1-approve, 0-deny

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

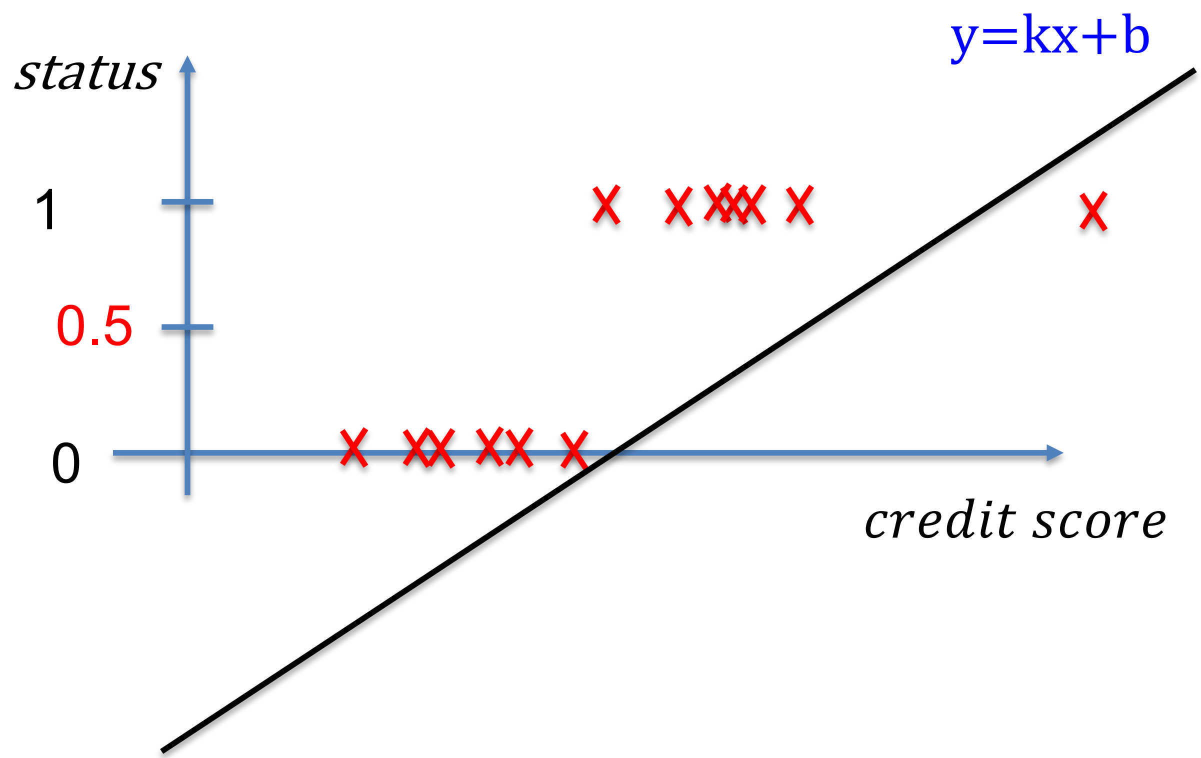
Credit Score	Loan Status
750	1
725	0
700	0
650	0
726	1
645	0
800	1
...	...



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)

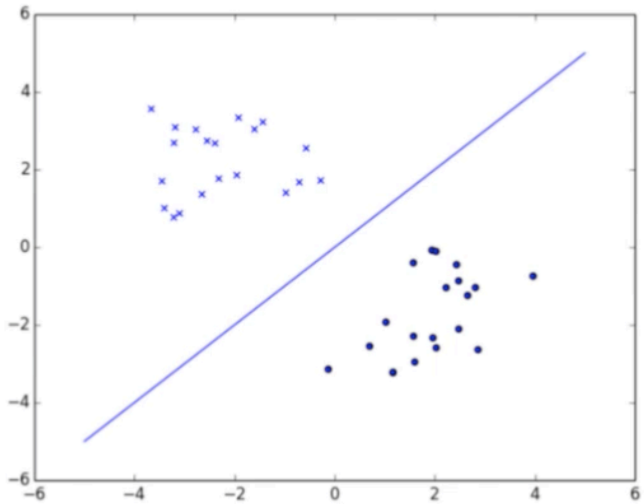
y: 1-approve, 0-deny

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
…	…	…

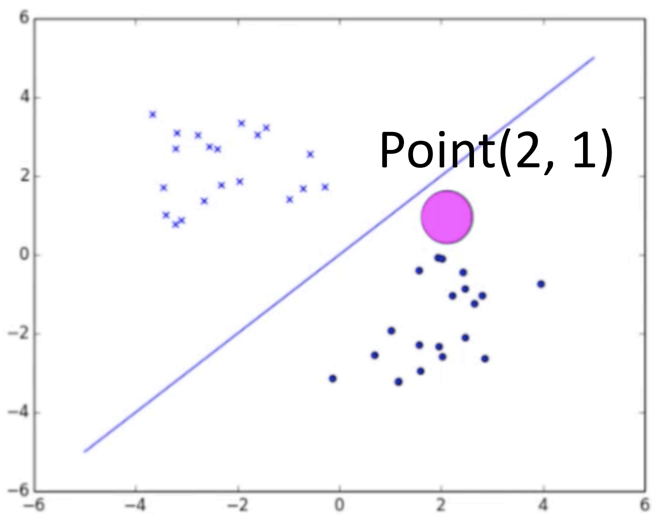
Test data:
for a new applicant with credit score 715 and income 68500, will the loan application be approved?

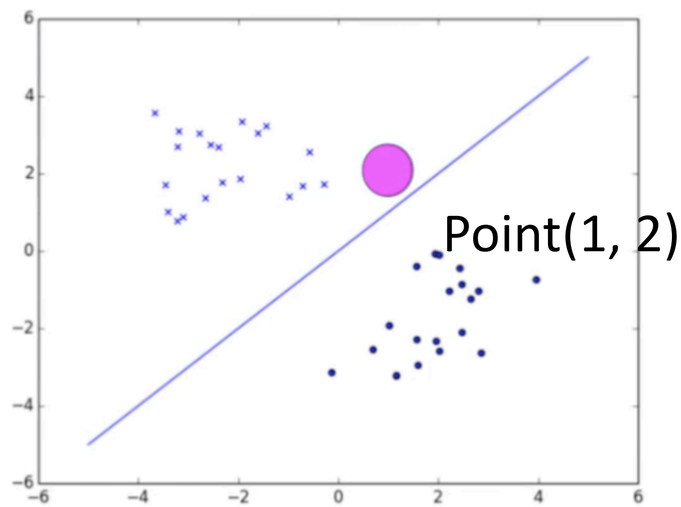
Linear Regression



$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

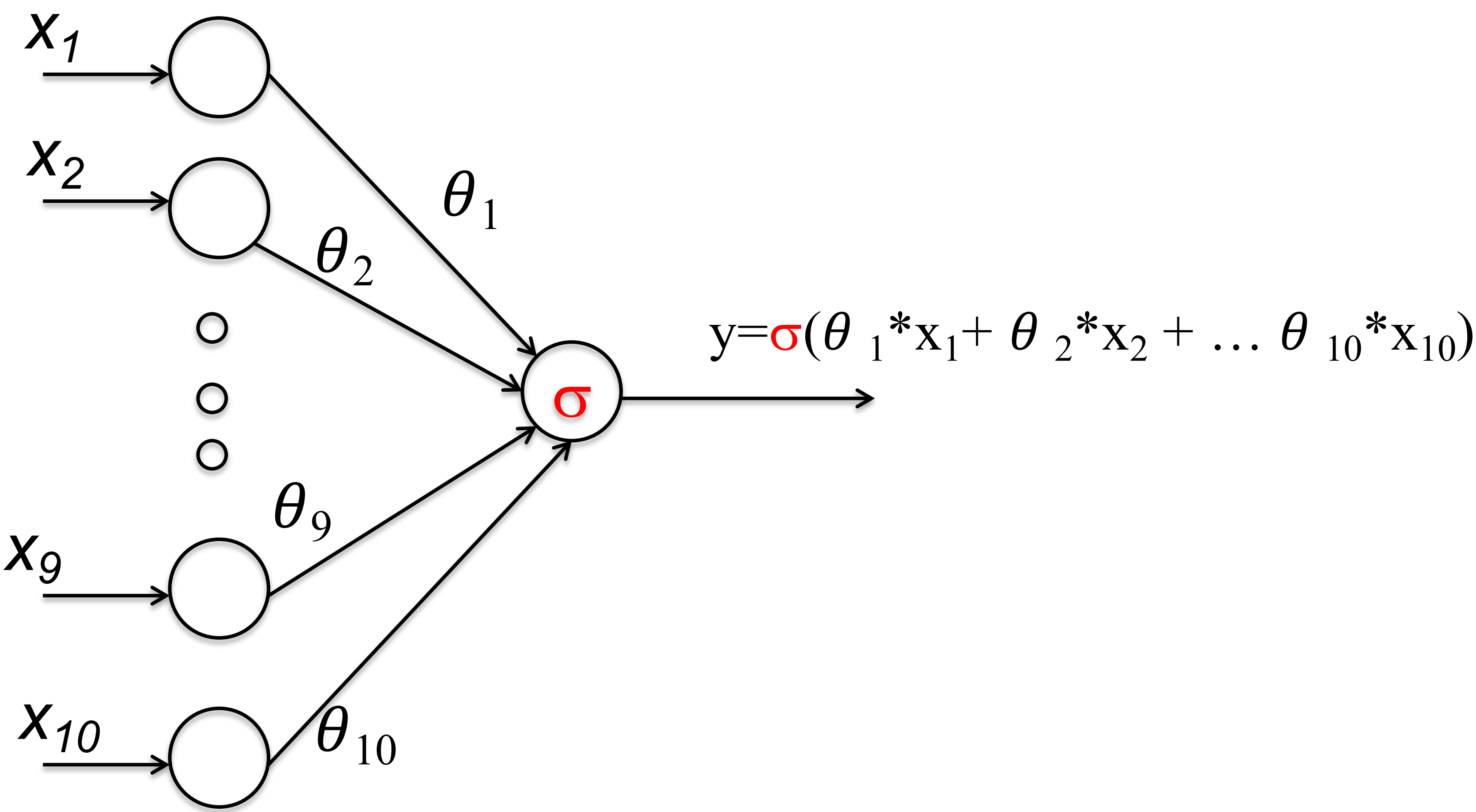




Regression

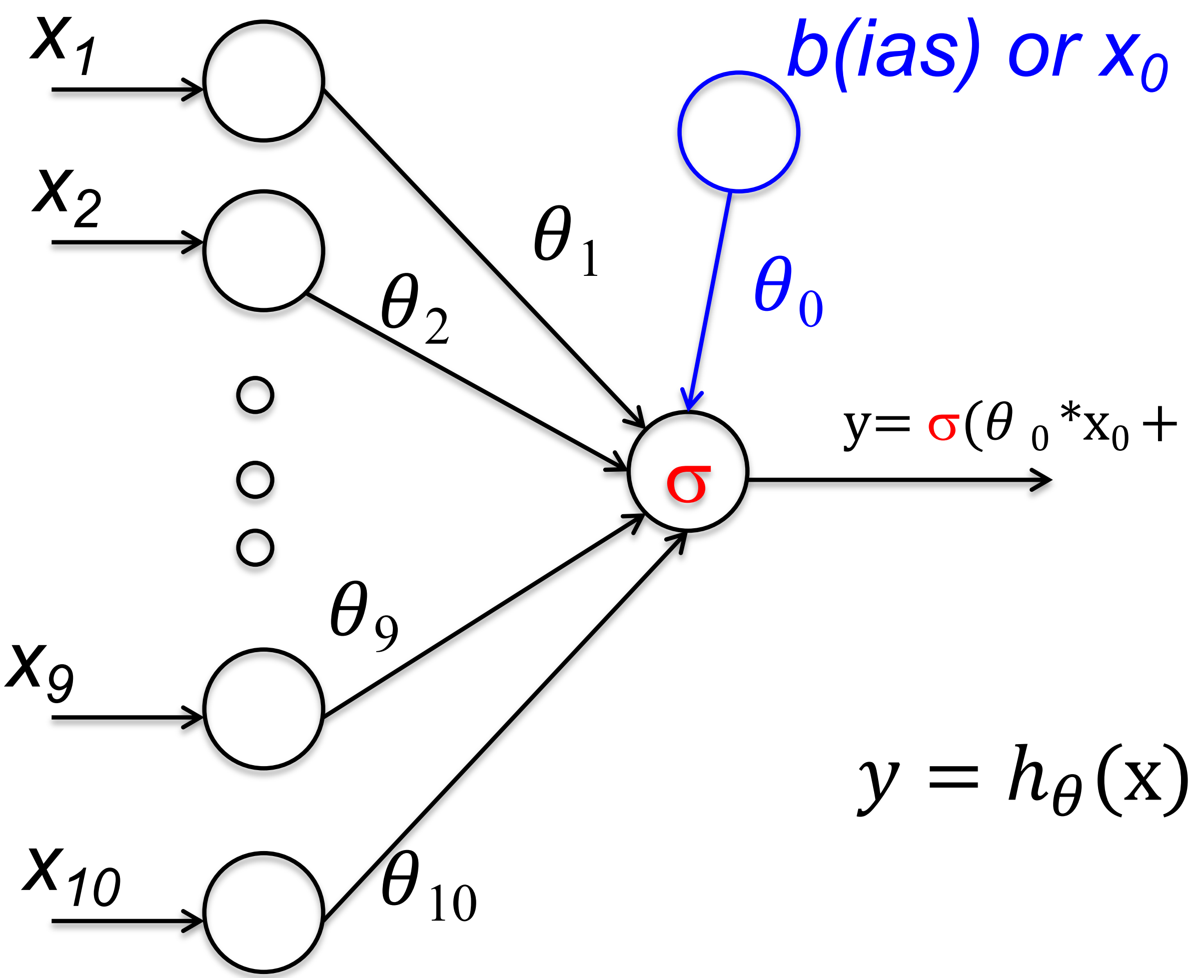
- $h_\theta(\mathrm{x}) = \theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2$, h() is a linear combination of the components of x
  - In vector form : $h_\theta(\mathrm{x}) = \theta^\mathrm{T}\mathrm{x}$
- The class separating function:
  - In 2-dimensions: a line
  - In 3-dimensions: a plane
  - In >3 dimension: hyperplane

Logistic Regression



10 features

Logistic Regression



$$y = \sigma(\theta_0 {}^* x_0 + \theta_1 {}^* x_1 + \theta_2 {}^* x_2 + \ldots \theta_{10} {}^* x_{10})$$

$$y = h_\theta(\mathrm{x}) = \sigma\left(\sum_{i=1}^{10} \theta_i {}^* x_i\right) = \sigma(\theta^{\mathrm{T}} \mathrm{x})$$

10 features

Activation Function σ

- Tanh()

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

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

- Sigmoid/Logistic

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

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

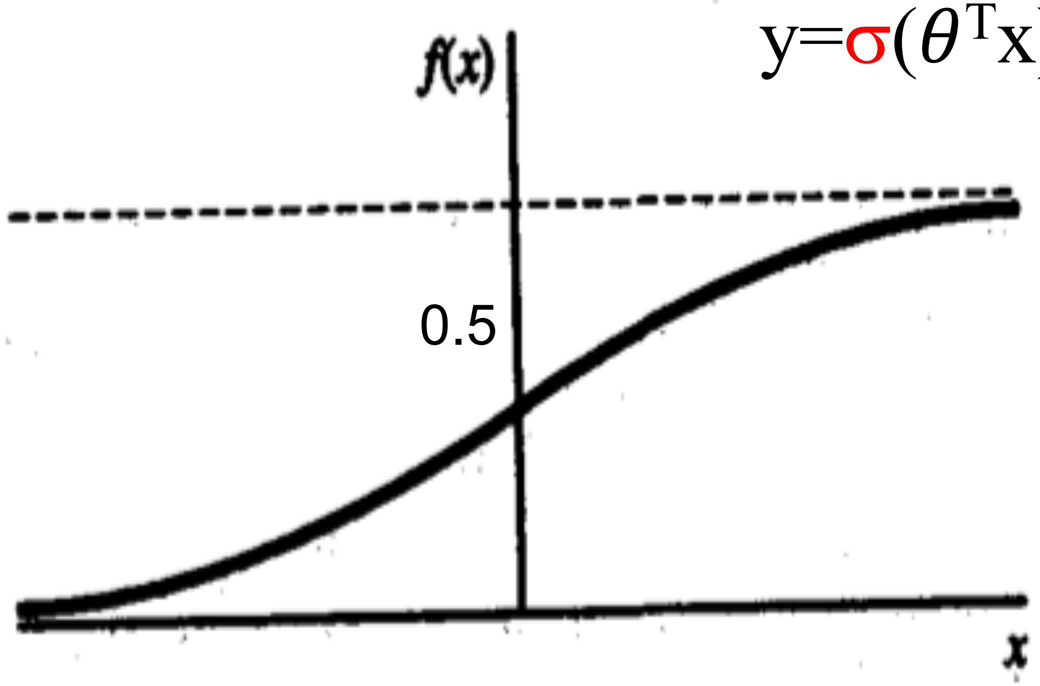
- Bipolar Sigmoid

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

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

Sigmoid Function for Classification

$y = \sigma(\theta^T x)$



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

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

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

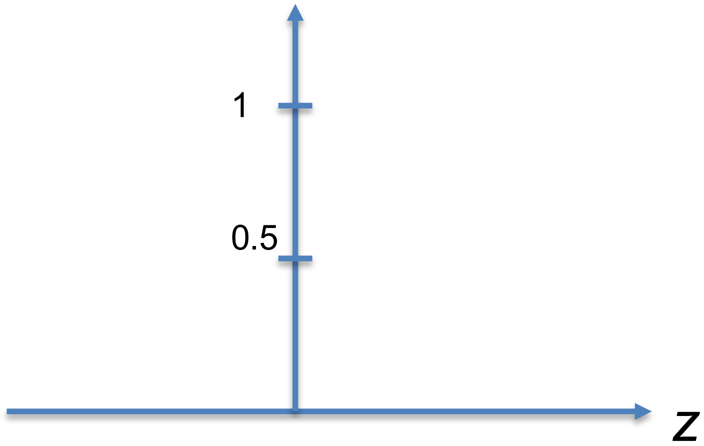
($\theta^T x \geq 0$,
predict class 1)

Logistic Regression Model

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

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

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



Sigmoid/logistic function

$h_\theta(x)$: *estimated probablity that* $y = 1$ *on 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:

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

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

x_0=1, y∈ $\{0, 1\}$

m examples

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

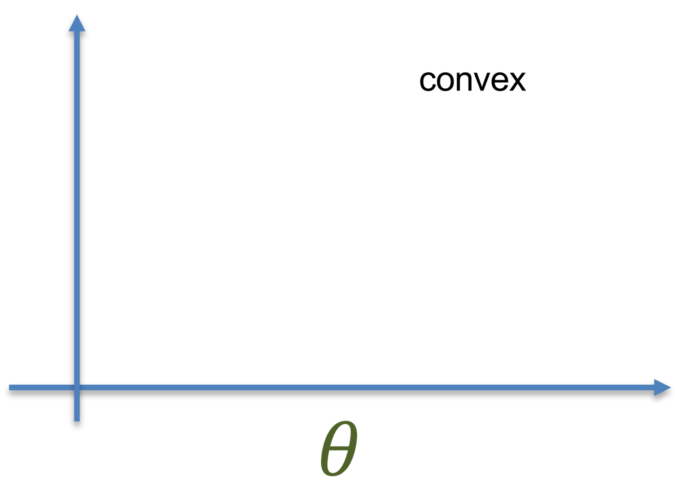
How to estimate the parameters θ from data?

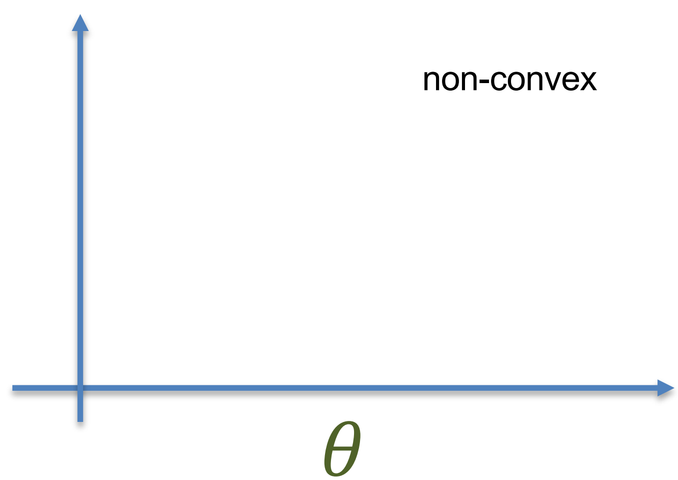
Cost Function

- Linear Regression:

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

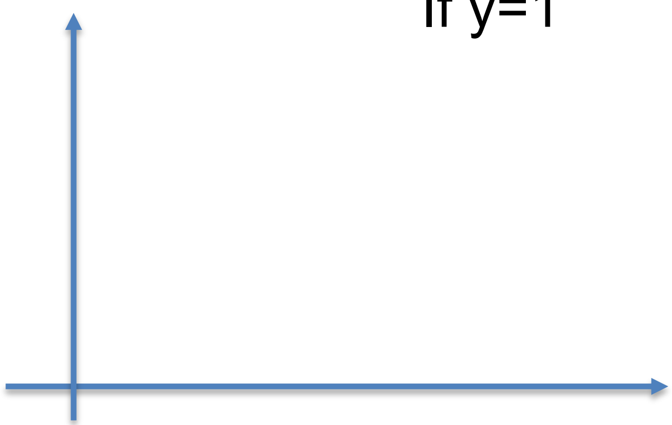
- In logistic regression, $(h_\theta(x^{(i)}) - y^{(i)})^2$ is not a convex curve, not suitable for gradient descent approximation approach.





Logistic Regression Cost Function

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



Logistic Regression Cost Function

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

Combine these two into one single cost function:

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

If y=1, $Cost(h_\theta(x), \text{y}) = -\log(h_\theta(x))$
If y=0, $Cost(h_\theta(x), \text{y}) = -\log(1 - h_\theta(x))$

Gradient Descent

- *To minimize the Cost function:*

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

- 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:

$$\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

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 minimize $J(w)$:

Repeat for each iteration:

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

Simultaneously update all θ_j

λ: Learning Rate → step size