

Social Media & Text Analysis

lecture 6 - Paraphrase Identification
and Logistic Regression



CSE 5539-0010 Ohio State University
Instructor: Wei Xu
Website: socialmedia-class.org

with slides adapted from Andrew Ng

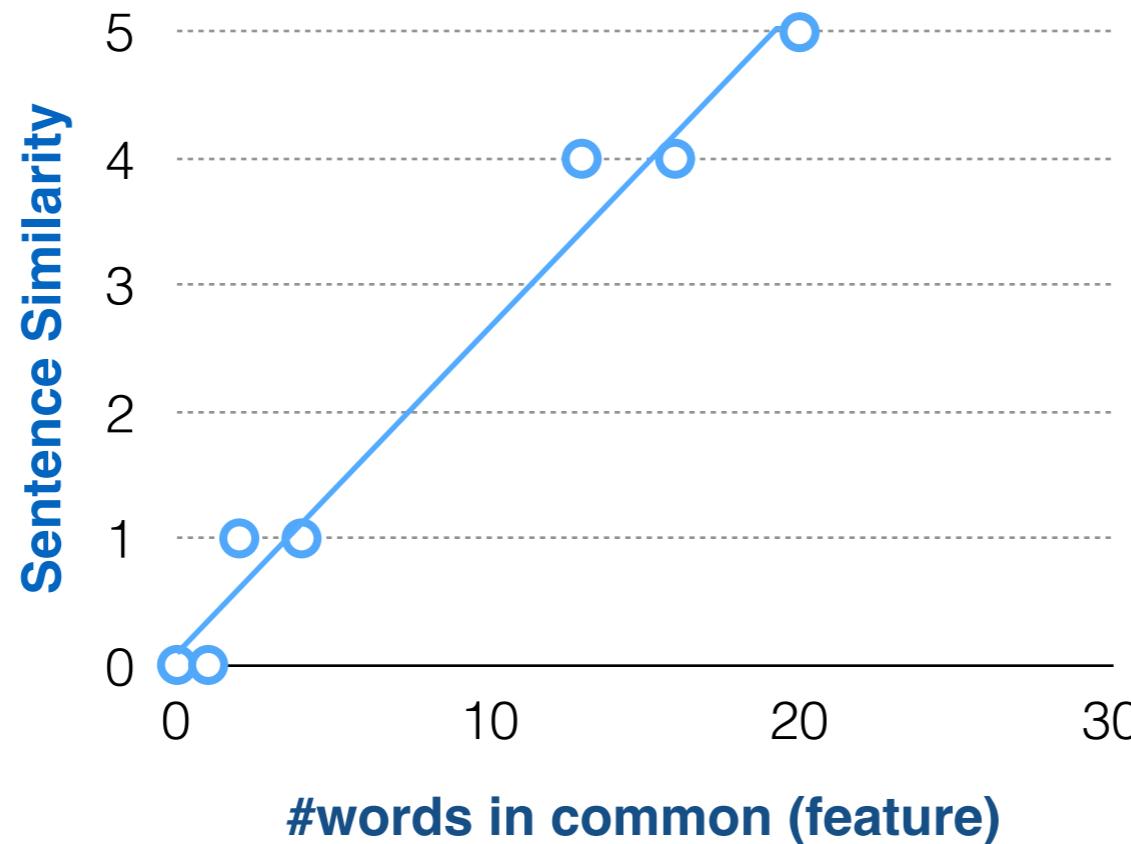
(Recap) Classification Method:

Supervised Machine Learning

- Input:
 - a sentence pair **x (represented by features)**
 - a fixed set of binary classes **$Y = \{0, 1\}$**
 - a training set of **m** hand-labeled sentence pairs
 $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- Output:
 - a learned classifier **$\gamma: x \rightarrow y \in Y$ ($y = 0$ or $y = 1$)**

(Recap)

Linear Regression



- also supervised learning (learn from annotated data)
- but for **Regression**: predict **real-valued** output
(Classification: predict discrete-valued output)

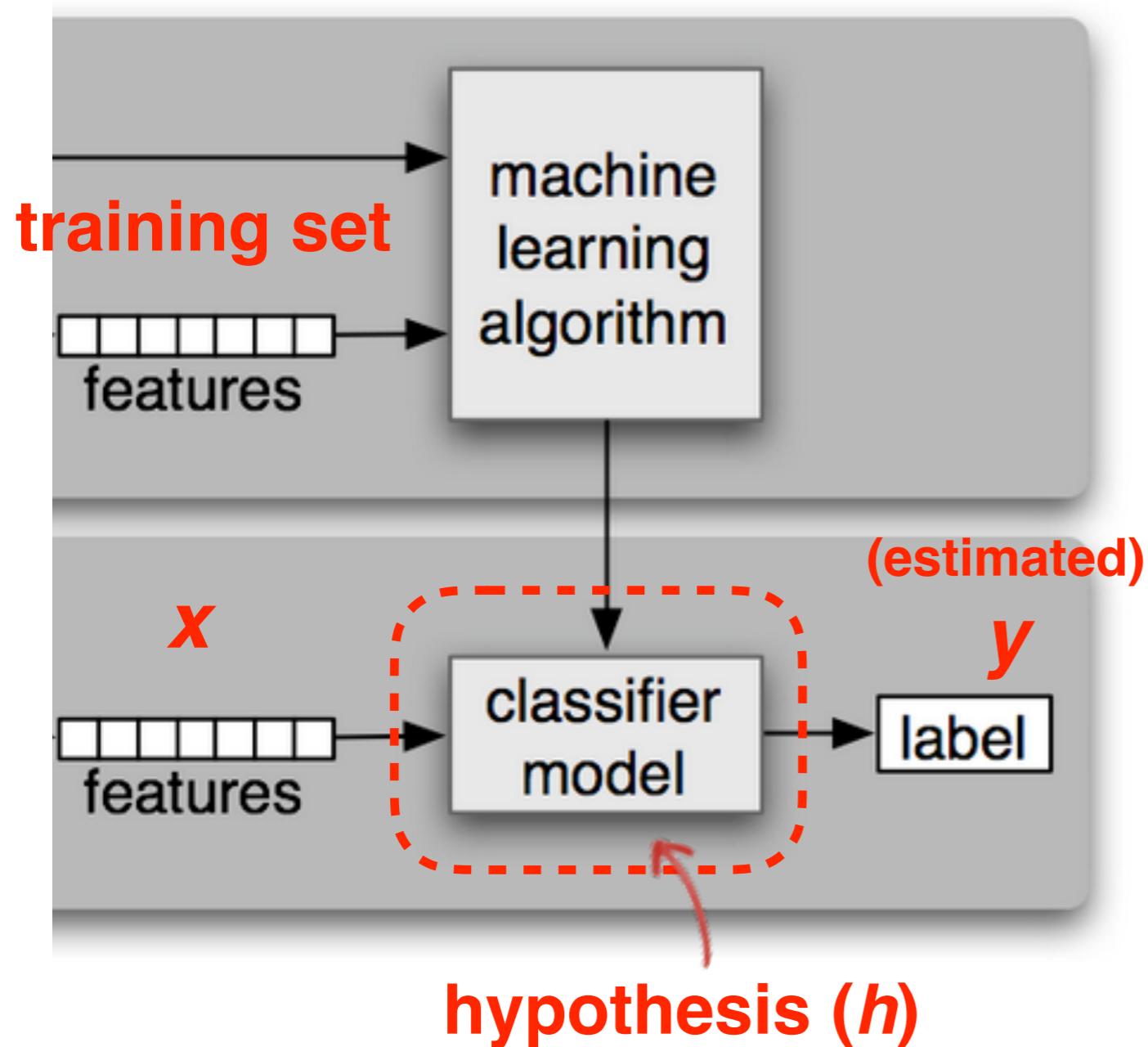
(Recap) Linear Regression w/ one variable: Model Representation

#words in common (x)	Sentence Similarity (y)
1	0
4	1
13	4
18	5
...	...

- m hand-labeled sentence pairs $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- θ 's: parameters

(Recap) Linear Regression: Model Representation

- How to represent h ?



$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Linear Regression
w/ one variable

(Recap)

Linear Regression

- **Hypothesis:**

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

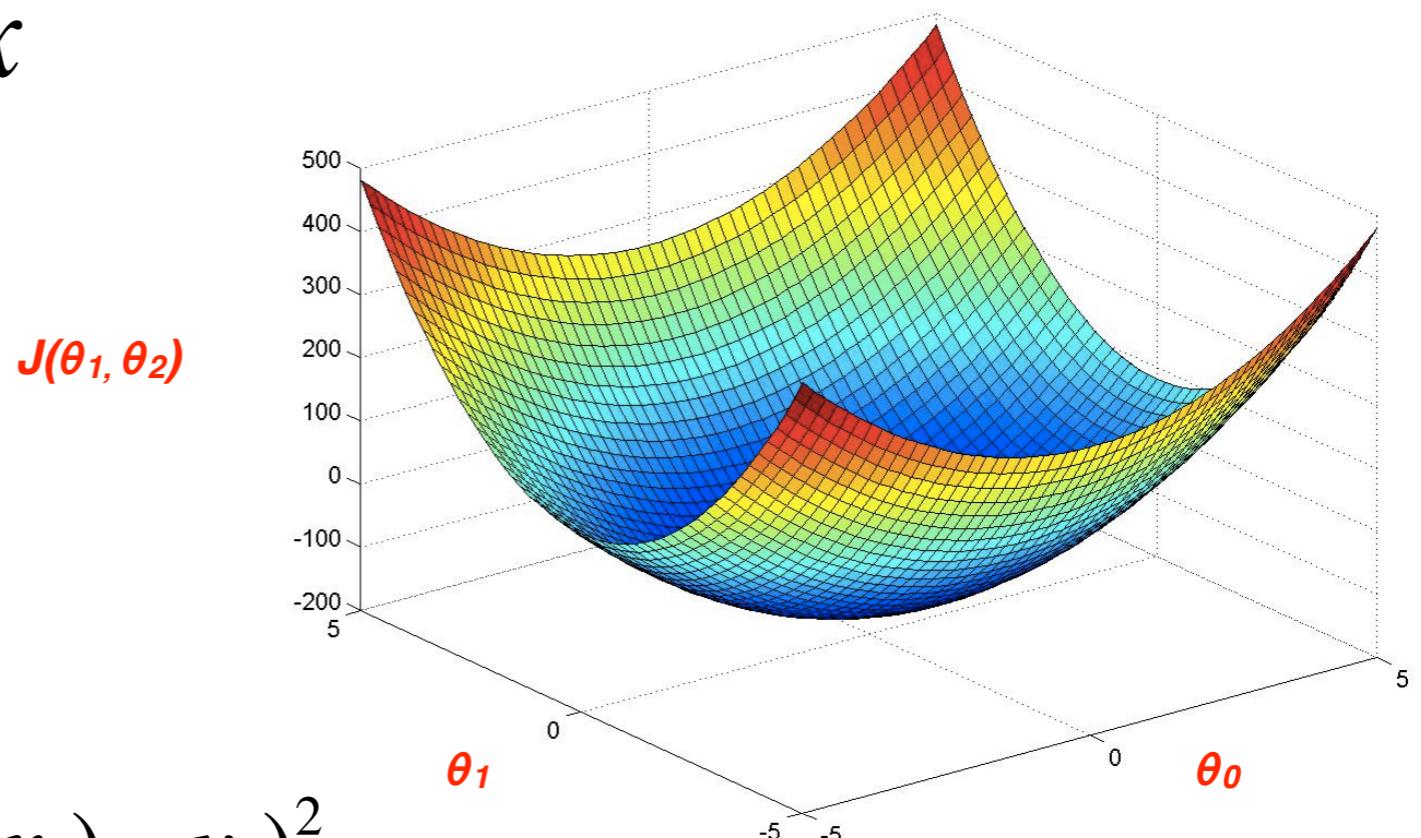
- **Parameters:**

$$\theta_0, \theta_1$$

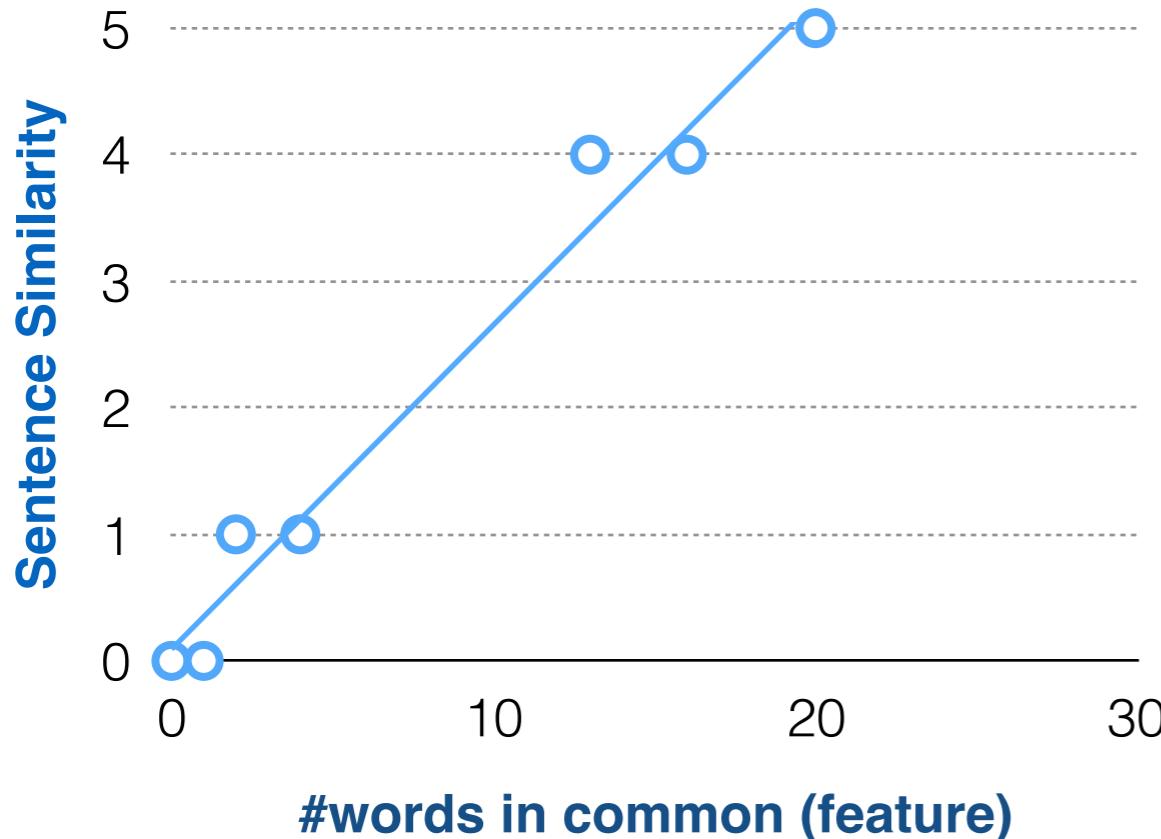
- **Cost Function:**

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

- **Goal:** $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$



(Recap) Linear Regression w/ one variable: Cost Function



squared error function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x_i) - y_i)^2$$

- **Idea:** choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for training examples (x, y)

$$\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$$

(Recap)

Gradient Descent

repeat until convergence {

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

learning rate

simultaneous update
for j=0 and j=1

(Recap) Linear Regression w/ one variable:

Gradient Descent

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x_i) - y_i)$$

simultaneous update θ_0, θ_1

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x_i) - y_i) \cdot x_i$$

}

Linear Regression w/ multiple variables (features):

Model Representation

- Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

(for convenience, define $x_0 = 1$)

Linear Regression w/ multiple variables (features):

Model Representation

- Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

(for convenience, define $x_0 = 1$)

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in R^{n+1}$$

Linear Regression w/ multiple variables (features):

Model Representation

- Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

(for convenience, define $x_0 = 1$)

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in R^{n+1}$$

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



Linear Regression w/ multiple variables (features):

Model Representation

- Hypothesis:

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

- Cost function: **# training examples**

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

(Recap)

Paraphrase Identification

obtain sentential paraphrases automatically

Mancini has been sacked by Manchester City

Yes!

Mancini gets the boot from Man City

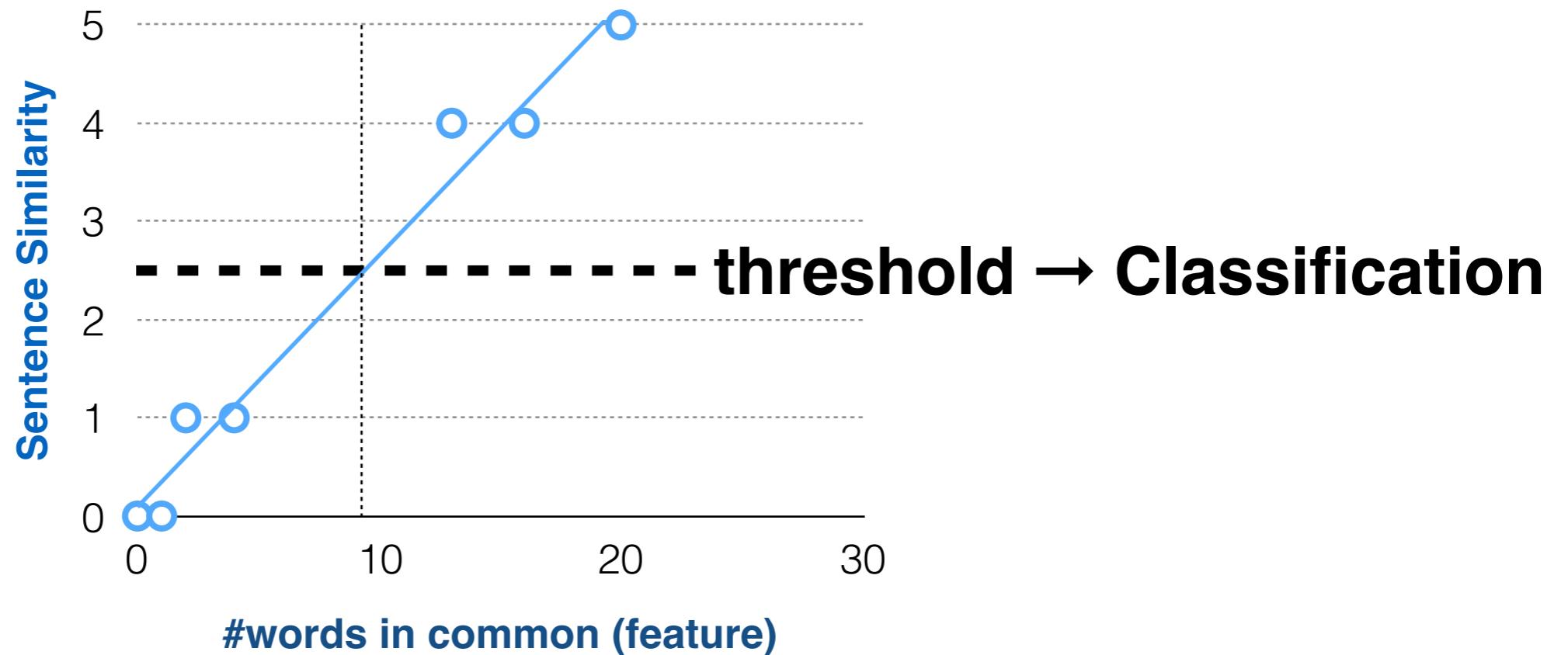
WORLD OF JENKS IS ON AT 11

No!

World of Jenks is my favorite show on tv

(Recap)

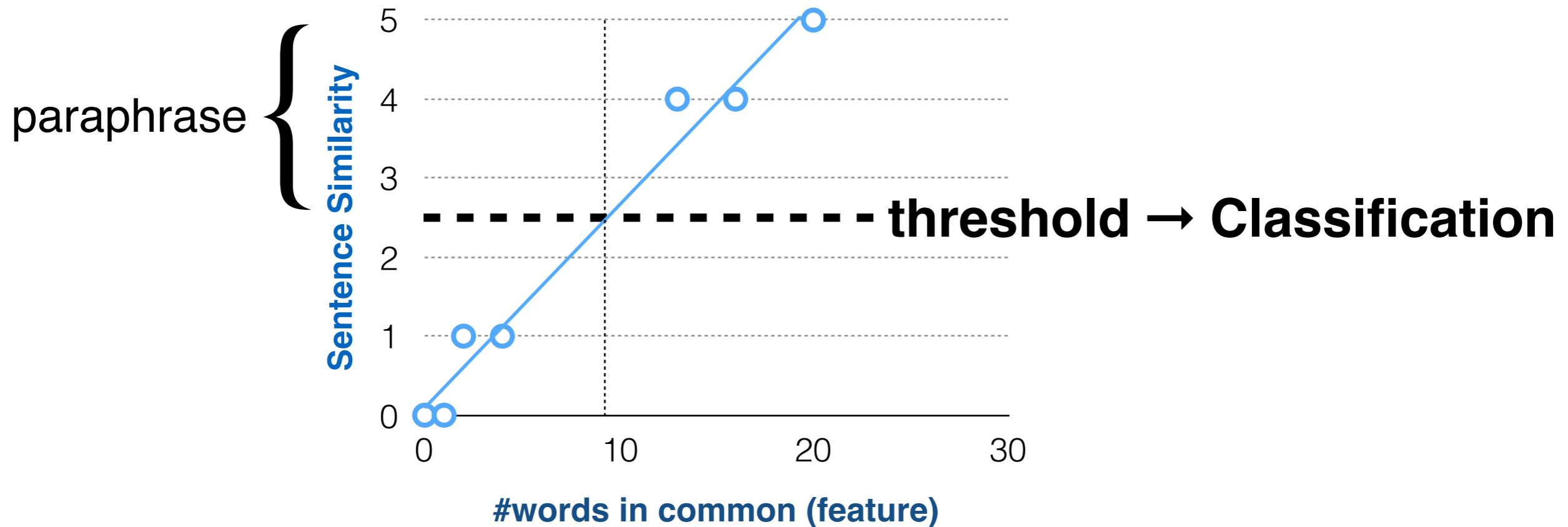
Linear Regression



- also supervised learning (learn from annotated data)
- but for **Regression**: predict **real-valued** output
(Classification: predict discrete-valued output)

(Recap)

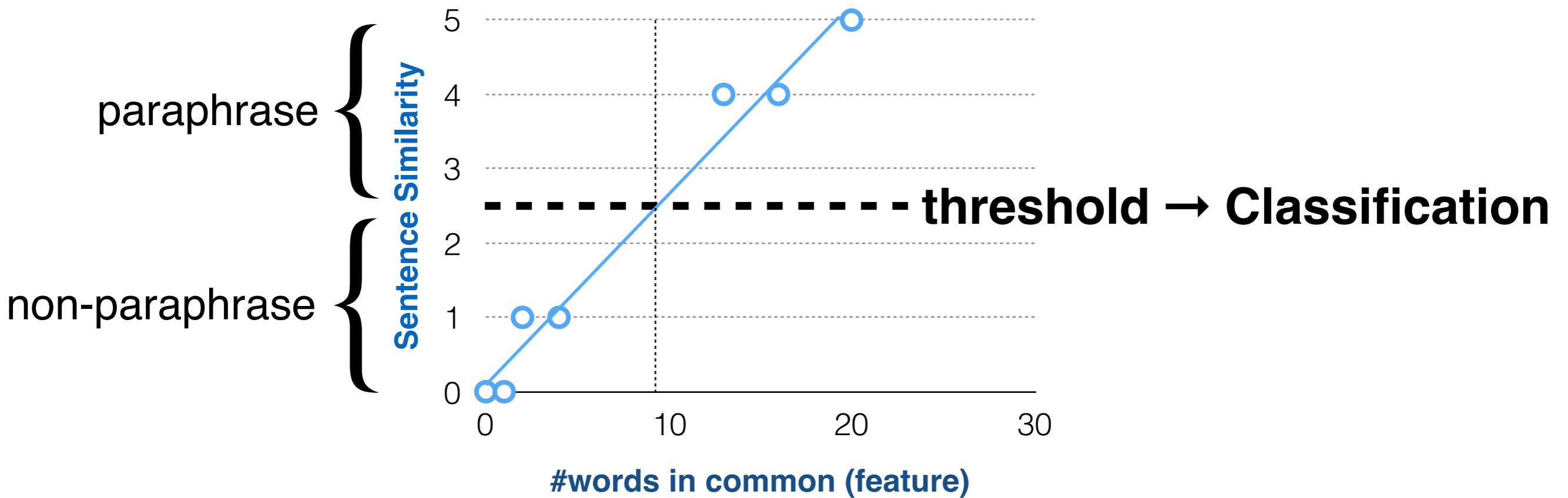
Linear Regression



- also supervised learning (learn from annotated data)
- but for **Regression**: predict **real-valued** output
(Classification: predict discrete-valued output)

(Recap)

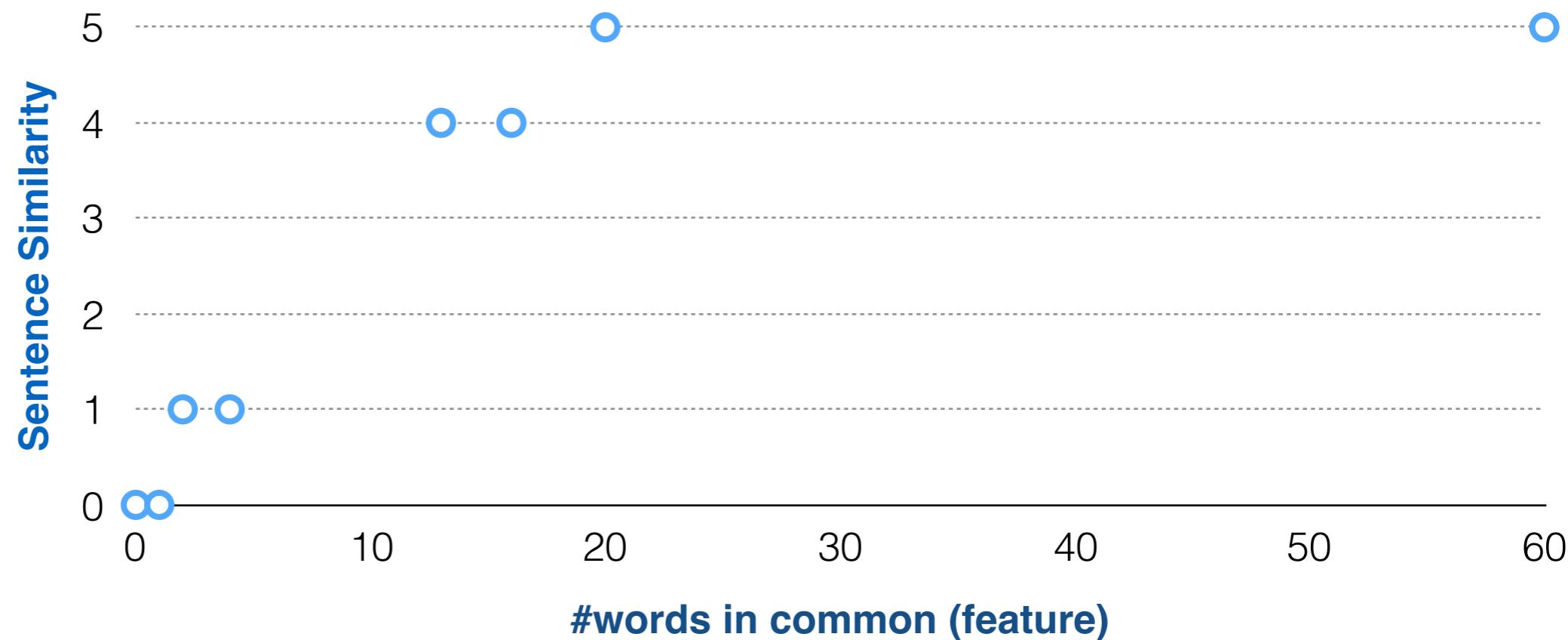
Linear Regression



- also supervised learning (learn from annotated data)
- but for **Regression**: predict **real-valued** output
(Classification: predict discrete-valued output)

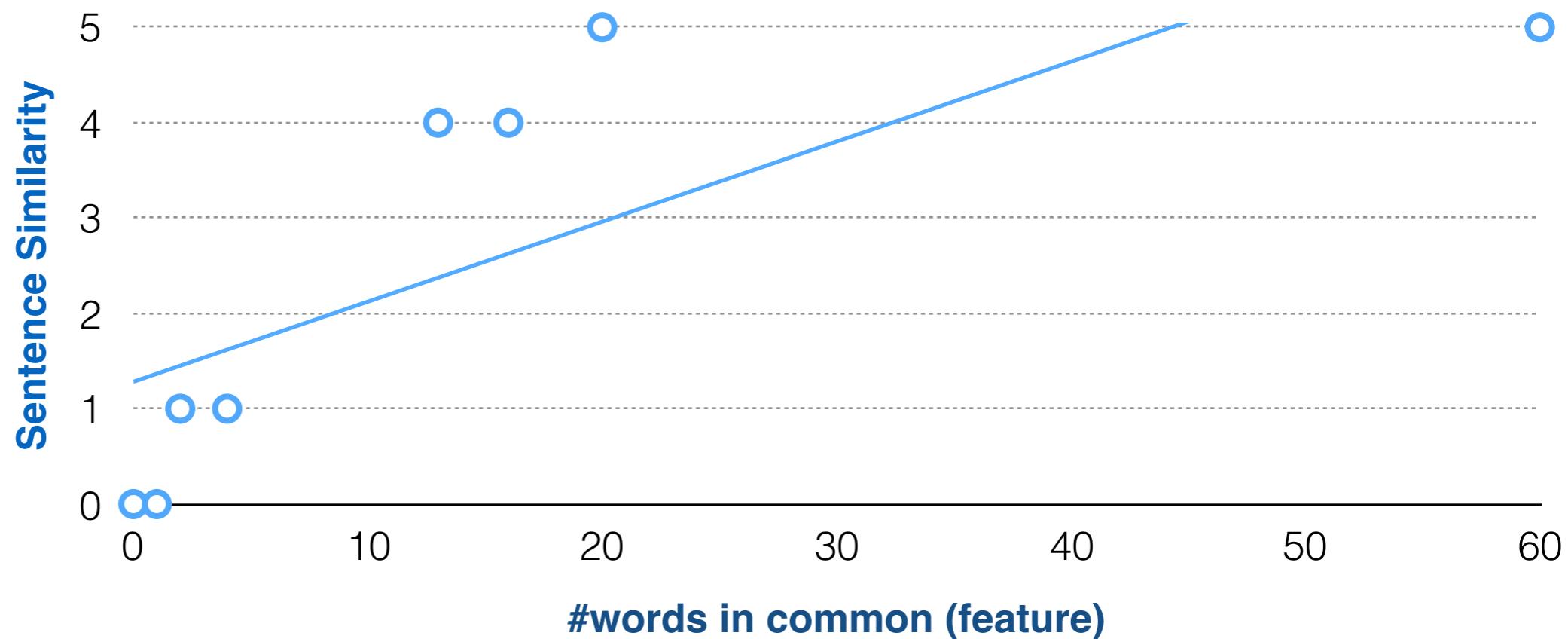
A problem in classification

Linear Regression



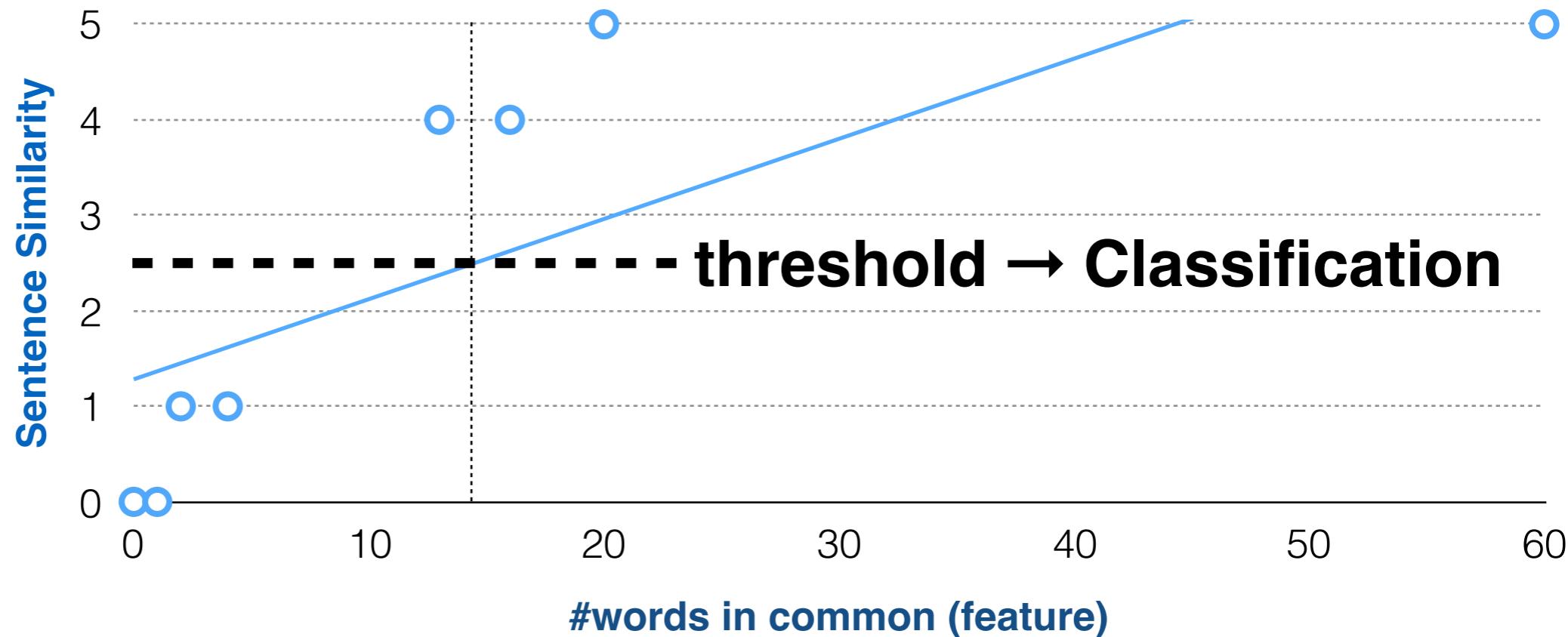
A problem in classification

Linear Regression



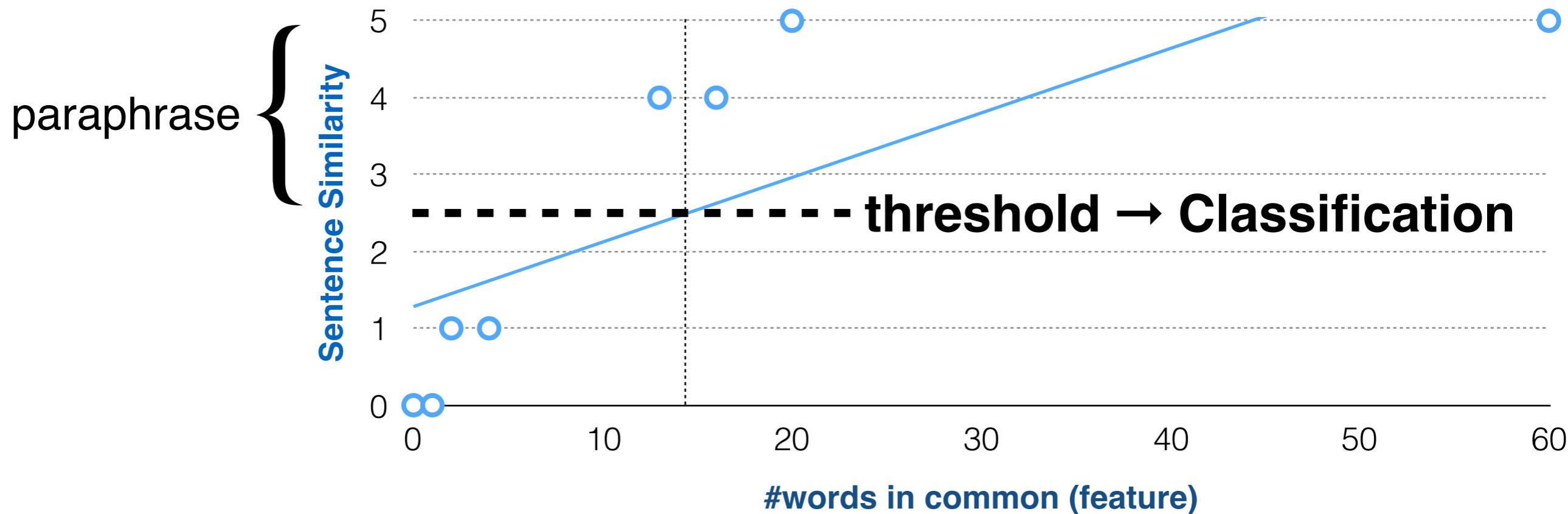
A problem in classification

Linear Regression



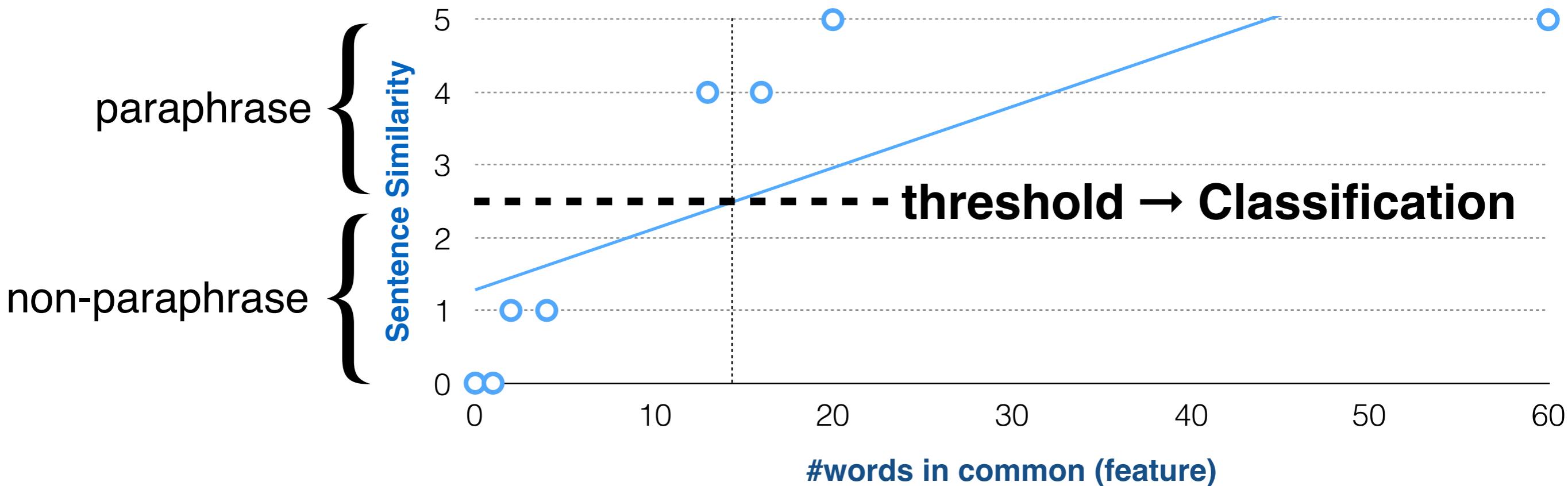
A problem in classification

Linear Regression



A problem in classification

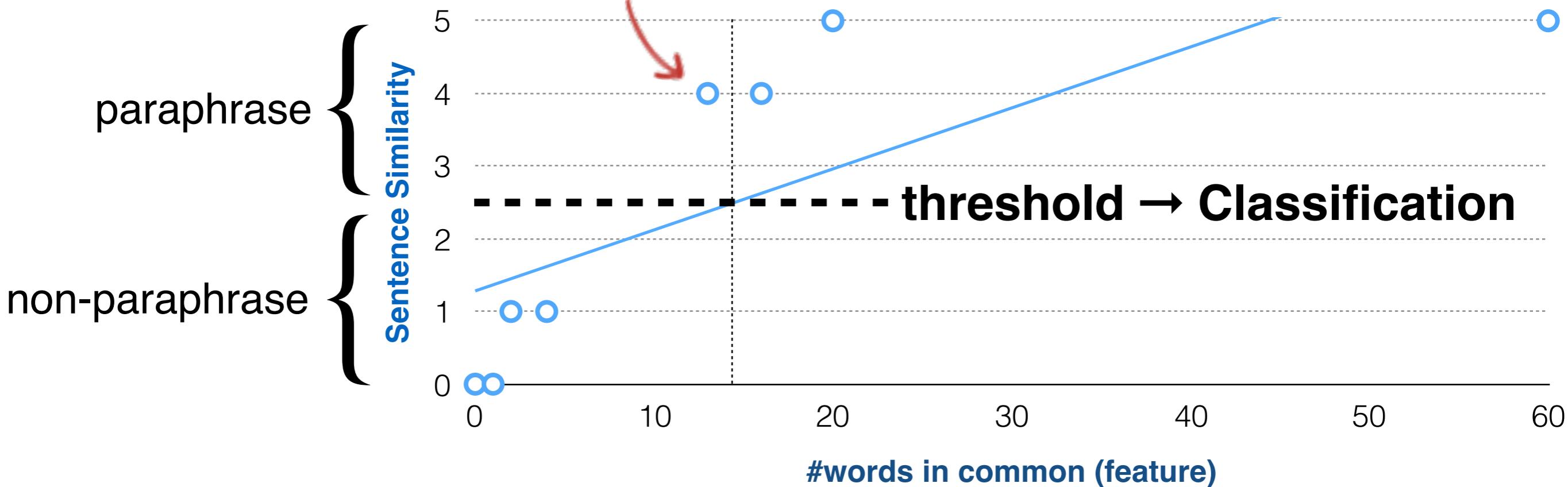
Linear Regression



A problem in classification

Linear Regression

classification error

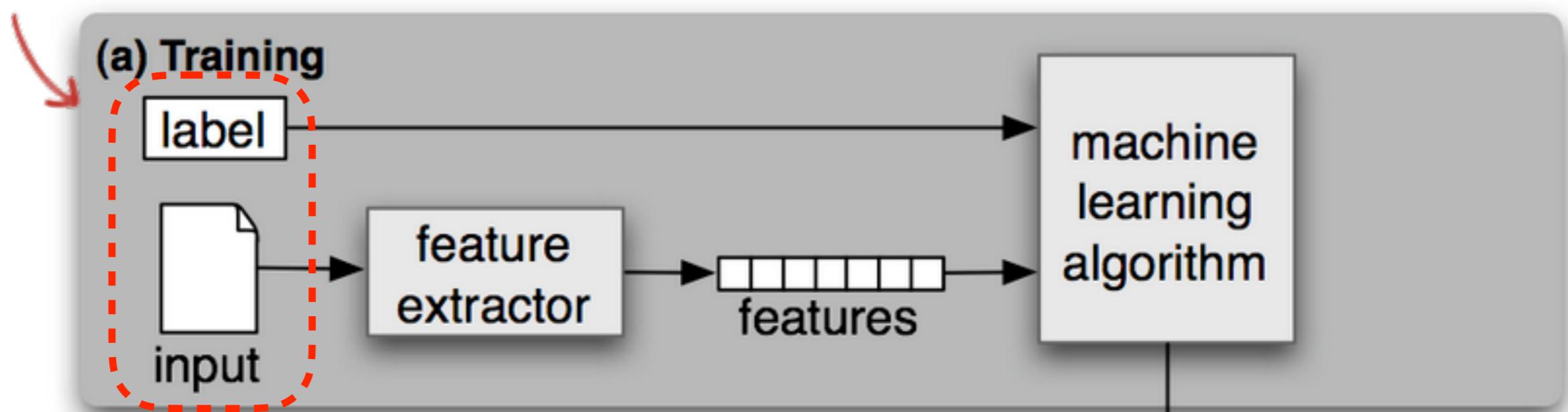


In practice, do not use linear regression for classification.

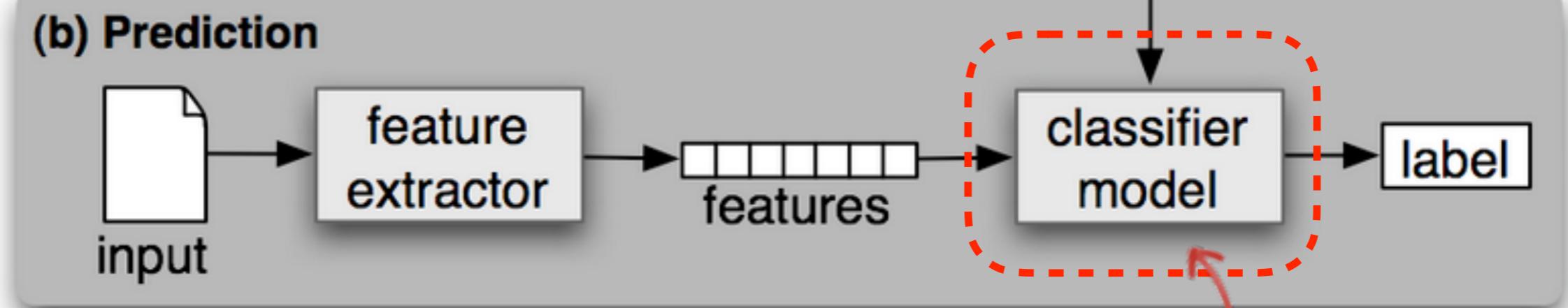
(Recap) Classification:

Supervised Machine Learning

training set



(b) Prediction



(also called) hypothesis

(Recap)

Logistic Regression

- One of the most useful **supervised machine learning algorithm** for classification!
- Generally high performance for a lot of problems.
- Much more robust than Naïve Bayes
(better performance on various datasets).

Hypothesis:

Linear → Logistic Regression

Classification: $y = 0$ or $y = 1$

- Linear Regression: $h_{\theta}(x)$ can be > 1 or < 0

Hypothesis:

Linear → Logistic Regression

Classification: $y = 0$ or $y = 1$

- Linear Regression: $h_{\theta}(x)$ can be > 1 or < 0
- Logistic Regression: want $0 \leq h_{\theta}(x) \leq 1$

Hypothesis:

Linear → Logistic Regression

Classification: $y = 0$ or $y = 1$

- Linear Regression: $h_{\theta}(x)$ can be > 1 or < 0
- Logistic Regression: want $0 \leq h_{\theta}(x) \leq 1$



a **classification (not regression) algorithm**

“Computer”

The term "**computer**", in use from the early 17th century (the first known written reference dates from 1613),^[1] meant "one who computes": a person performing mathematical **calculations**, before **electronic computers** became commercially available. "The human computer is supposed to be following fixed rules; he has no authority to deviate from them in any detail."^[2] Teams of people were frequently used to undertake long and often tedious calculations; the work was divided so that this could be done in parallel.



NACA High Speed Flight Station
"Computer Room" (1949)

“Computer”

 INDEPENDENT



Culture › Film › Features

Hidden Figures takes us back to a place where computers were women and black

Computers weren't always an electronic device – it was a job done by black women – but the relations of 1960s black politics, culture and space were far more complex than 'Hidden Figures' has time to show in terms of challenging prejudice



Hypothesis:

Linear → Logistic Regression

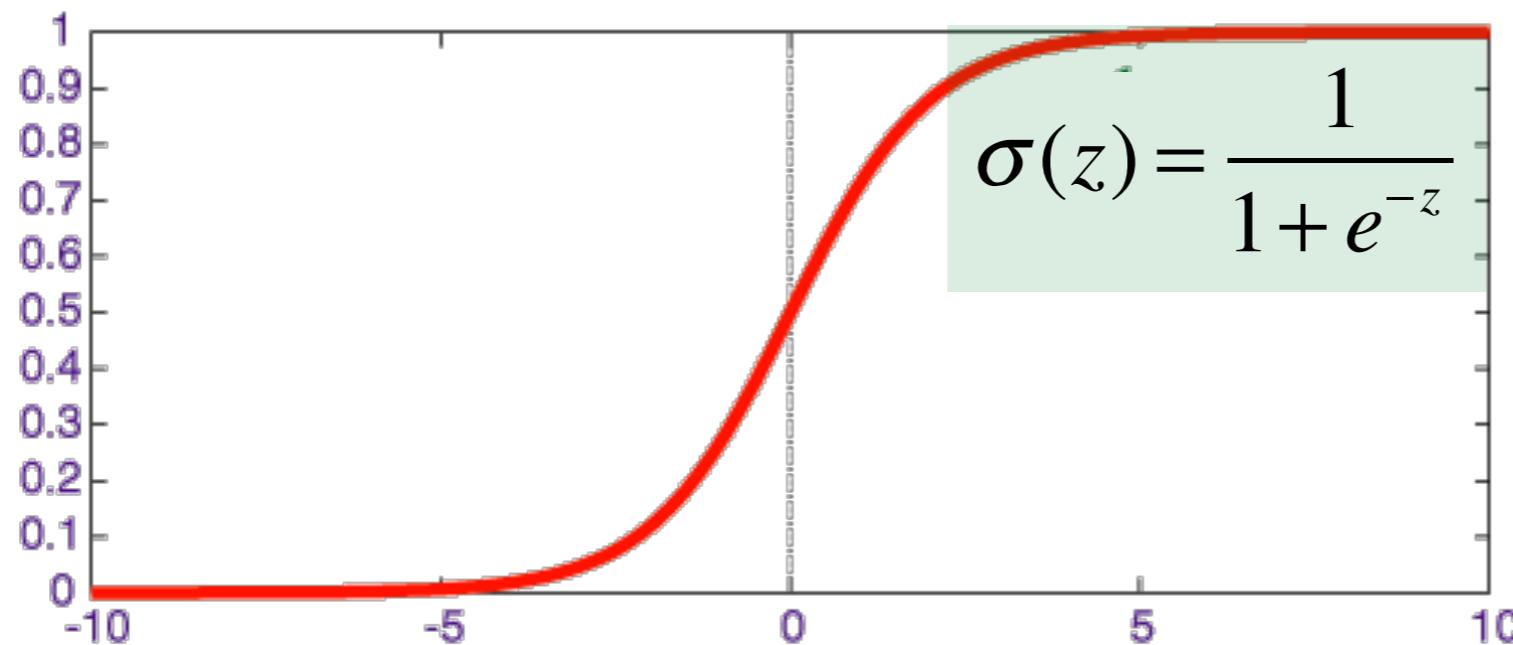
- Linear Regression: $h_{\theta}(x) = \theta^T x$
- Logistic Regression: want $0 \leq h_{\theta}(x) \leq 1$

Hypothesis:

Linear → Logistic Regression

- Linear Regression: $h_{\theta}(x) = \theta^T x$
- Logistic Regression: want $0 \leq h_{\theta}(x) \leq 1$

sigmoid (logistic) function



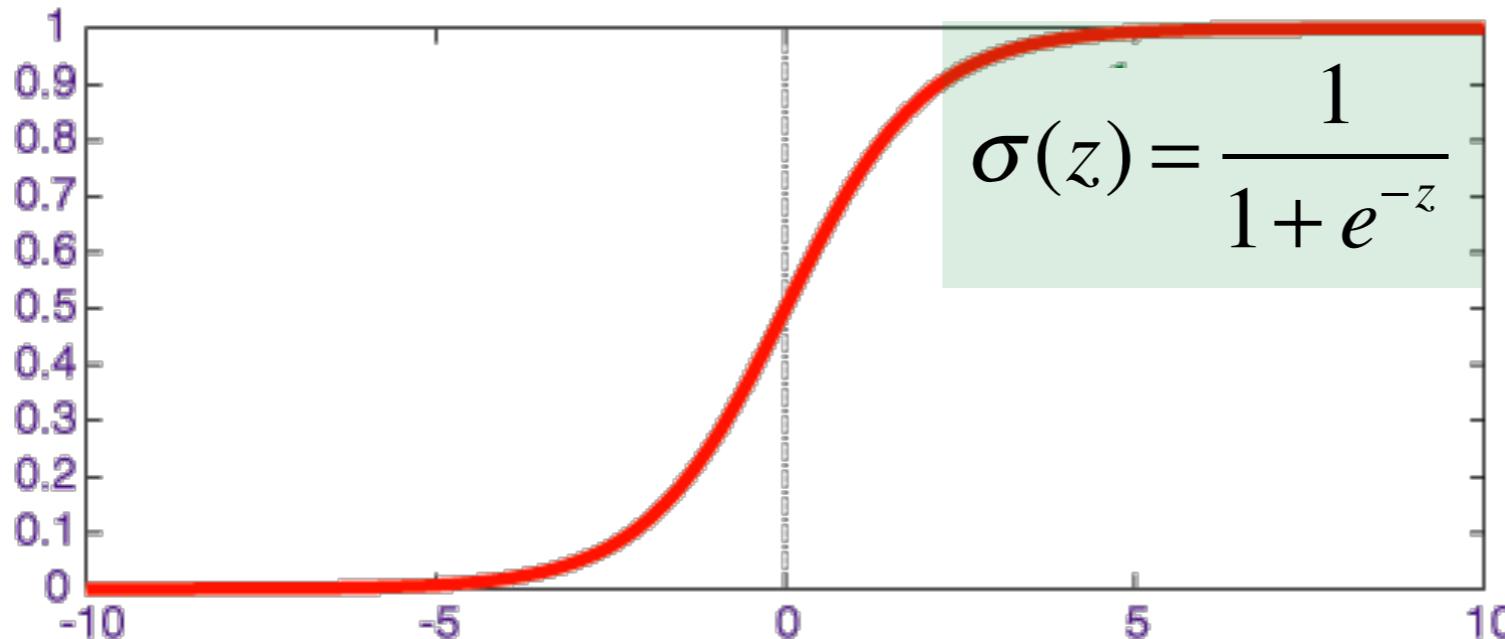
Hypothesis:

Linear → Logistic Regression

- Linear Regression: $h_{\theta}(x) = \theta^T x$
- Logistic Regression: want $0 \leq h_{\theta}(x) \leq 1$



sigmoid (logistic) function $h_{\theta}(x) = \sigma(\theta^T x)$



(Recap) Classification Method:

Supervised Machine Learning

- Input:
 - a sentence pair **x (represented by features)**
 - a fixed set of binary classes **$Y = \{0, 1\}$**
 - a training set of **m** hand-labeled sentence pairs
 $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- Output:
 - a learned classifier **$\gamma: x \rightarrow y \in Y$ ($y = 0$ or $y = 1$)**

Logistic Regression:

Interpretation of Hypothesis

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on input

Logistic Regression:

Interpretation of Hypothesis

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on input

If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \#words_in_common \end{bmatrix}$, $h_{\theta}(x) = 0.7$

70% chance of the sentence pair being paraphrases

Logistic Regression:

Interpretation of Hypothesis

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on input

If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \#words_in_common \end{bmatrix}$, $h_{\theta}(x) = 0.7$

70% chance of the sentence pair being paraphrases

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



probability that $y = 1$, given x , parameterized by θ

Logistic Regression:

Interpretation of Hypothesis

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on input

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



probability that $y = 1$, given x , parameterized by θ

Logistic Regression:

Interpretation of Hypothesis

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on input

$$P(y=1|x;\theta) + P(y=0|x;\theta) = 1$$

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



probability that $y = 1$, given x , parameterized by θ

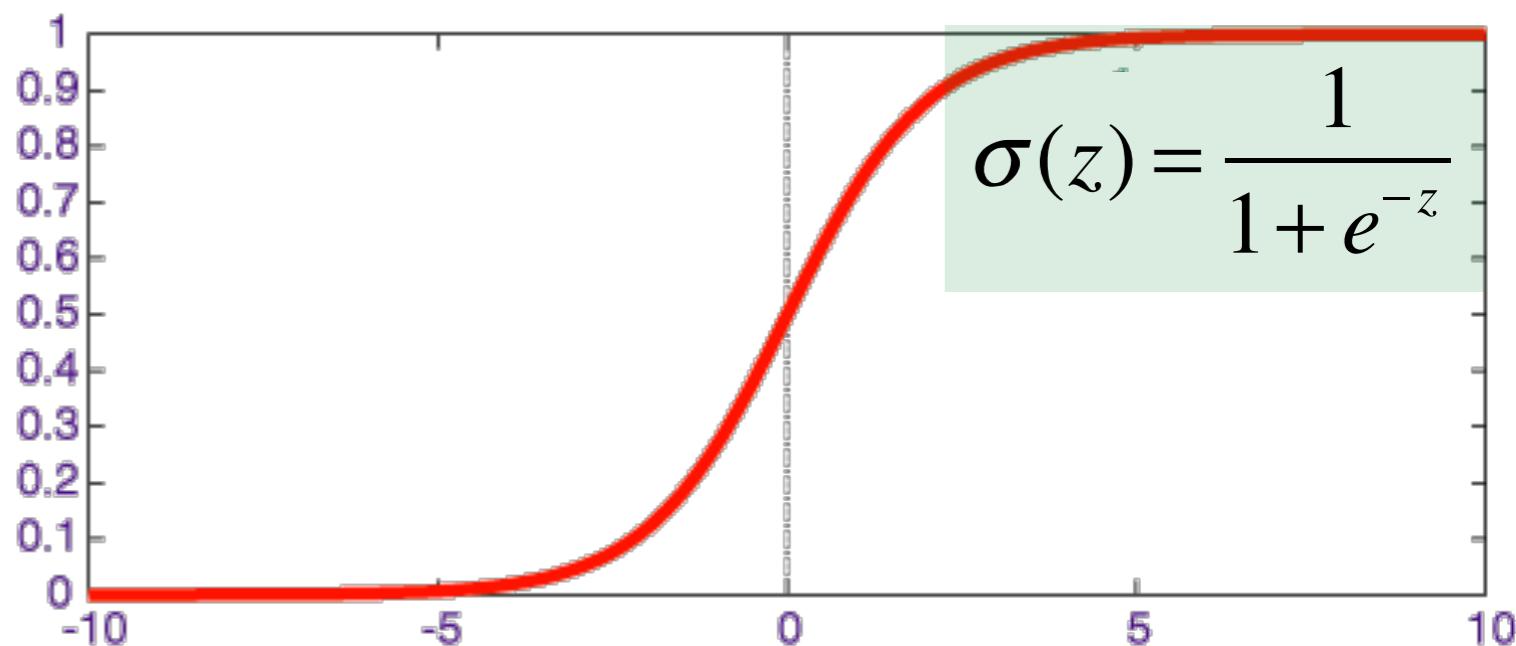
Logistic Regression:

Decision Boundary

- Logistic Regression: **sigmoid (logistic) function**

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

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



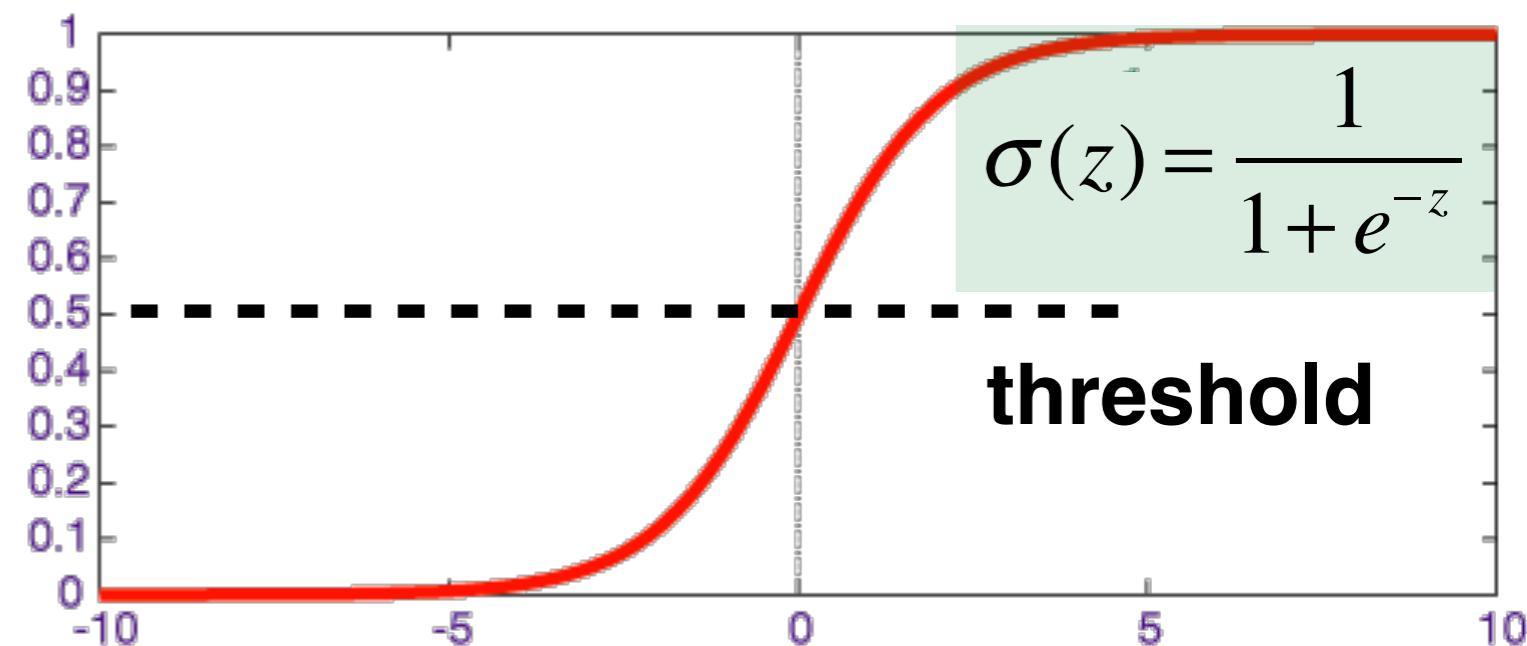
Logistic Regression:

Decision Boundary

- Logistic Regression: **sigmoid (logistic) function**

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

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



predict $y = 1$ if $h_{\theta}(x) \geq 0.5$

predict $y = 0$ if $h_{\theta}(x) < 0.5$

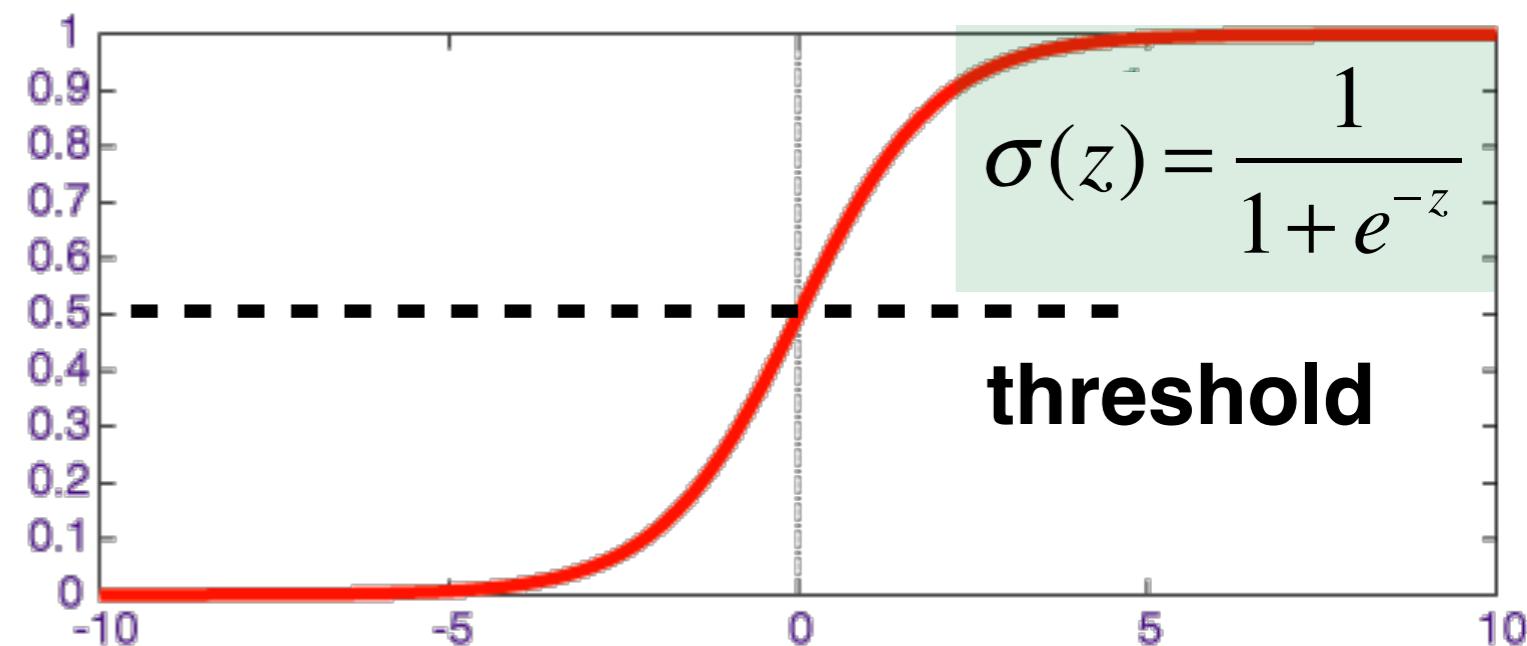
Logistic Regression:

Decision Boundary

- Logistic Regression: **sigmoid (logistic) function**

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

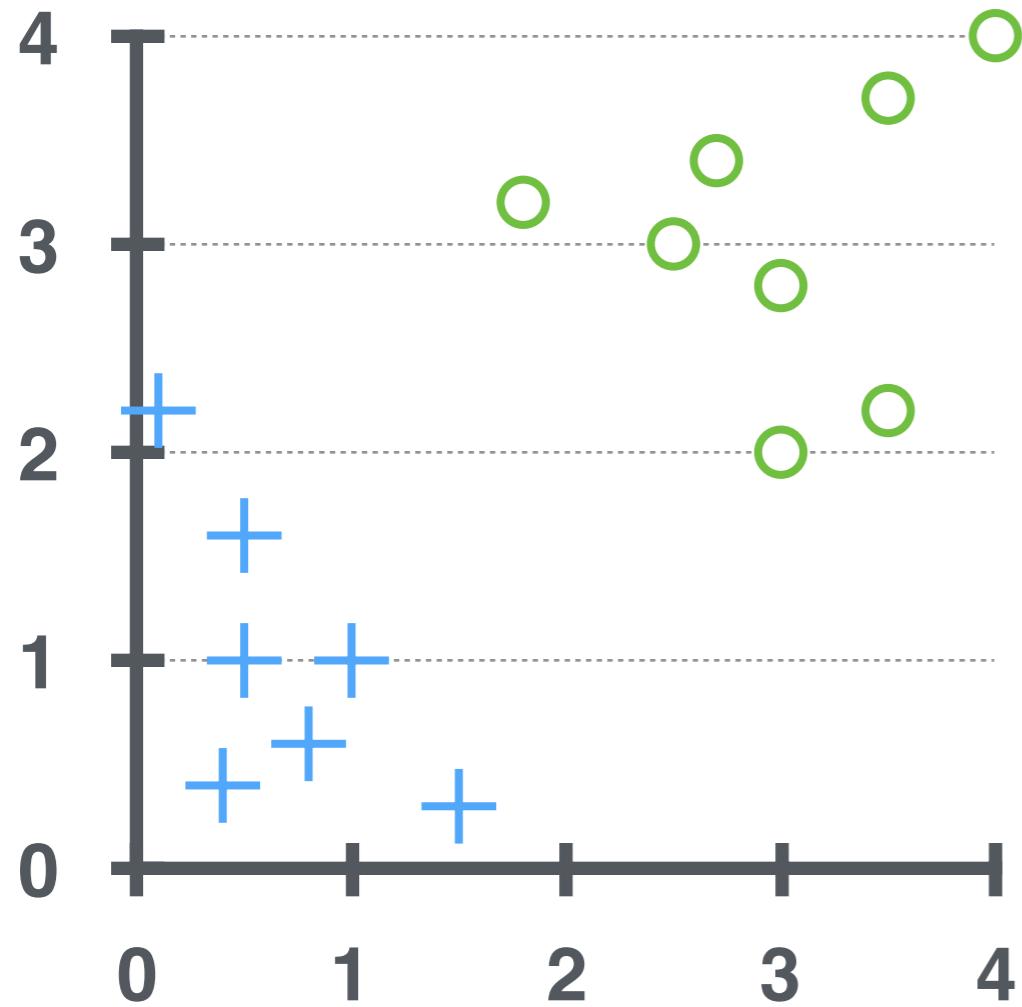
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$



predict $y = 1$ if $h_{\theta}(x) \geq 0.5$ ← when $\theta^T x \geq 0$

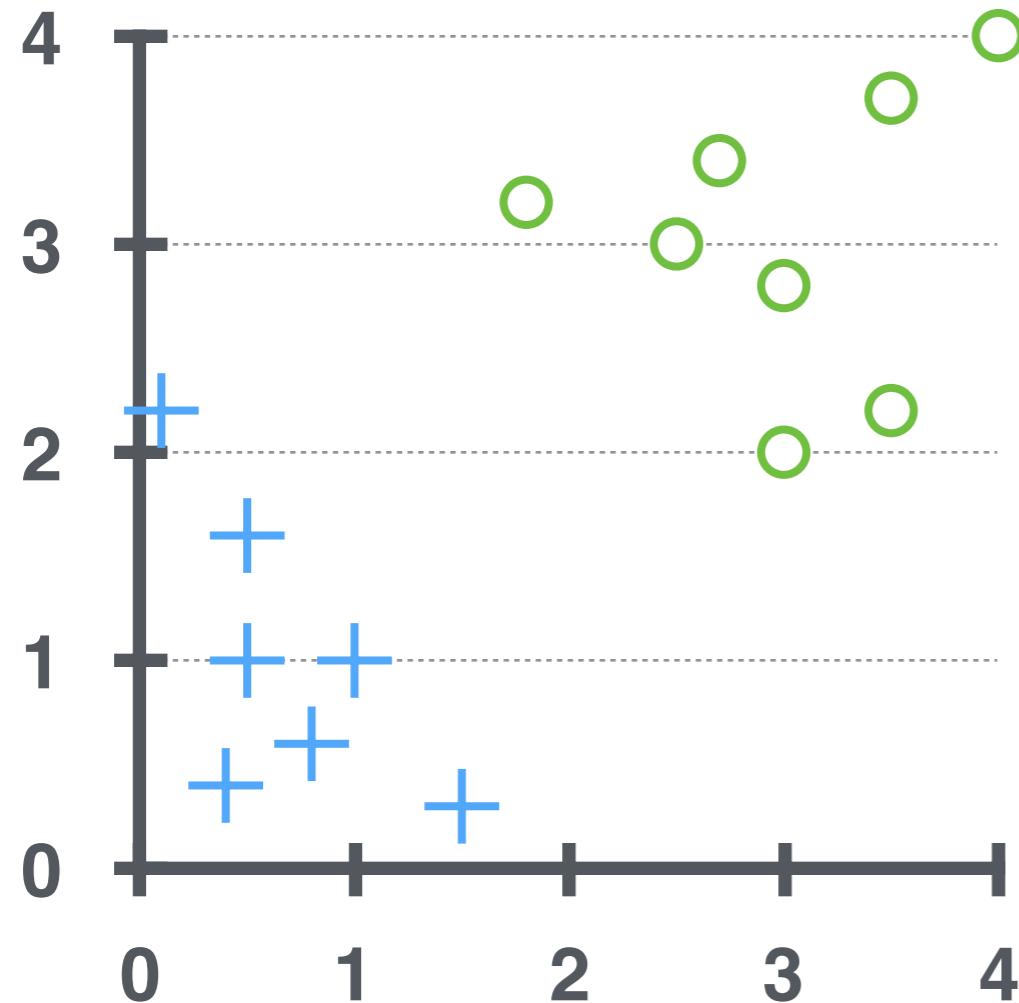
predict $y = 0$ if $h_{\theta}(x) < 0.5$ ← when $\theta^T x < 0$

Logistic Regression: Decision Boundary



Logistic Regression: Decision Boundary

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



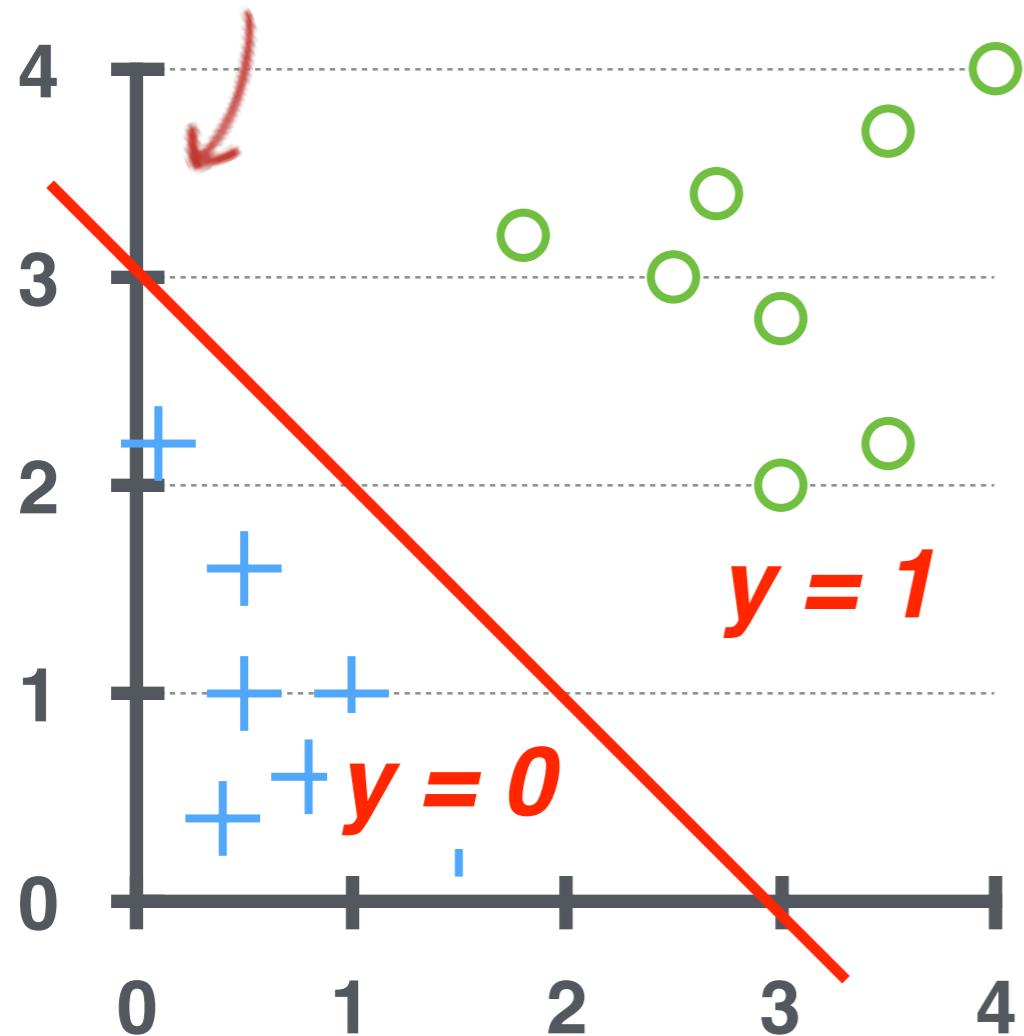
What if

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix} ?$$

predict $\textcolor{red}{y = 1}$ if $\theta^T x \geq 0$

Logistic Regression: Decision Boundary

decision boundary



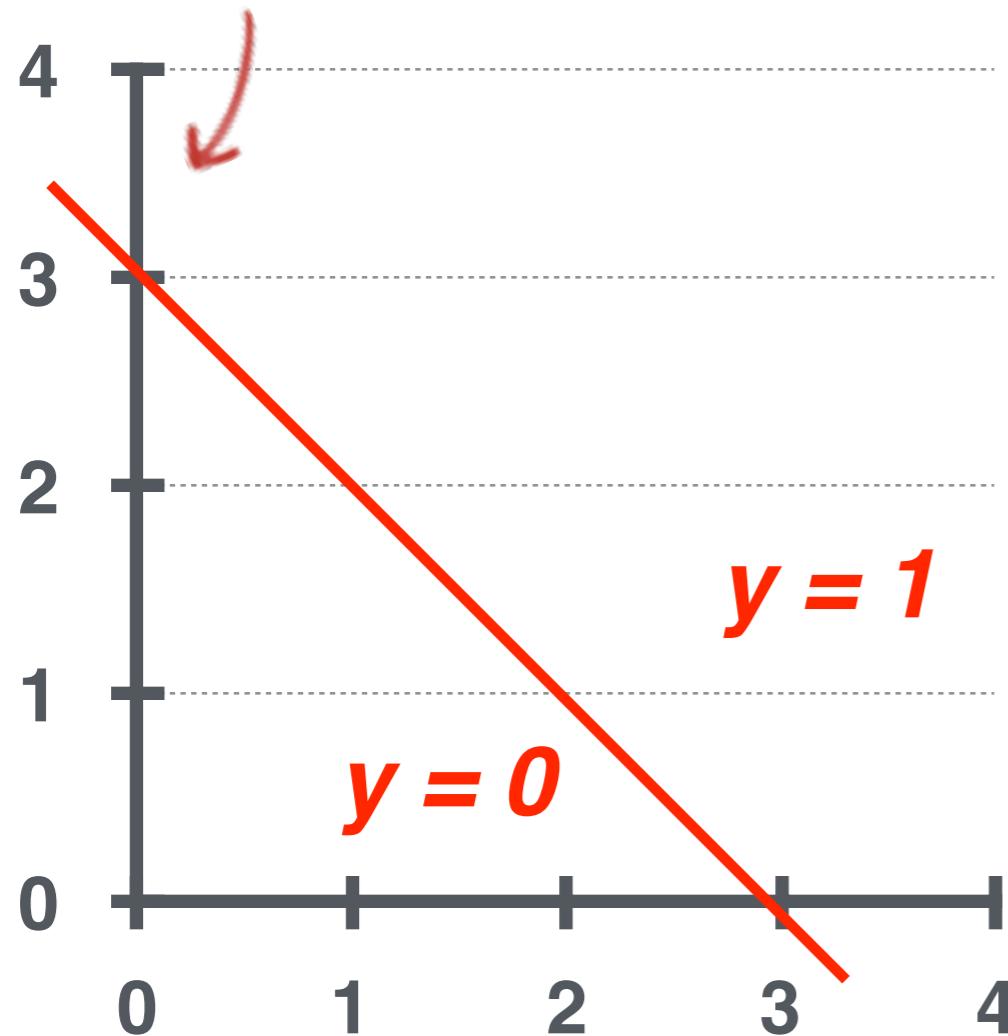
$$h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

What if $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$?

predict $y = 1$ if $\theta^T x \geq 0$

Logistic Regression: Decision Boundary

decision boundary



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

- a property of the hypothesis
- a property of the parameters
- a property of the dataset

Logistic Regression

- a training set of m hand-labeled sentence pairs
 $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(m)}, \mathbf{y}^{(m)})$ ($\mathbf{y} \in \{0, 1\}$)

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

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^{n+1} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in R^{n+1}$$

Cost function:

Linear → Logistic Regression

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

Cost function:

Linear → Logistic Regression

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

squared error function

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

this cost function is non-convex for logistic regression

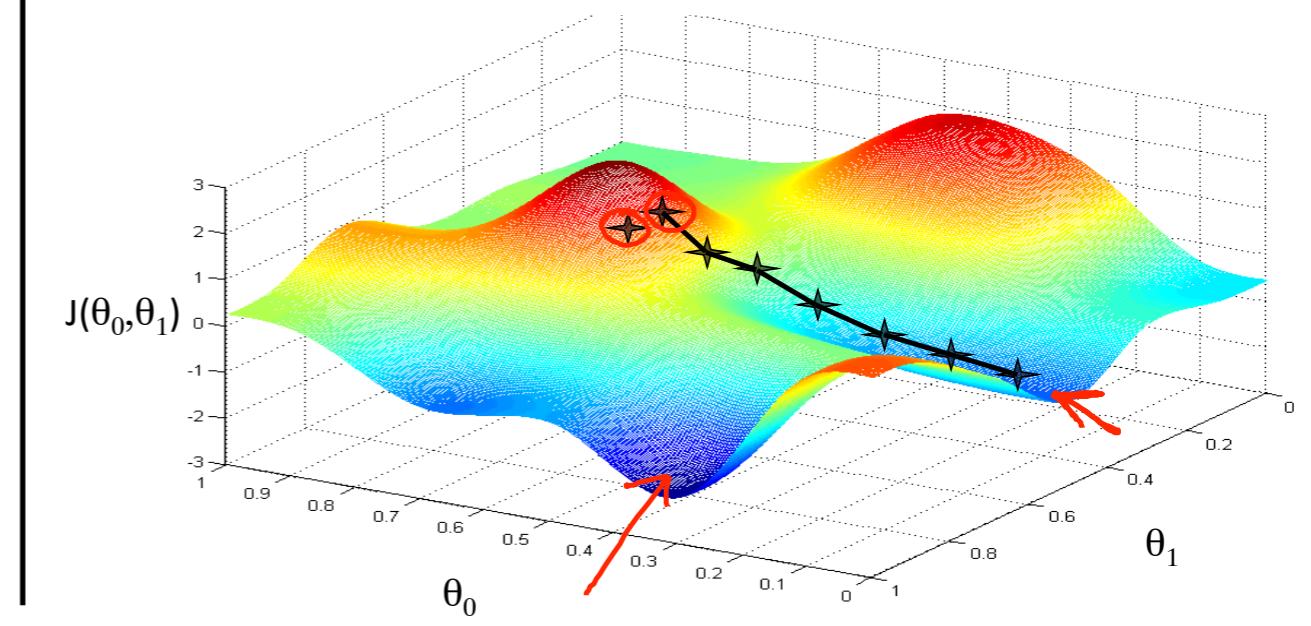
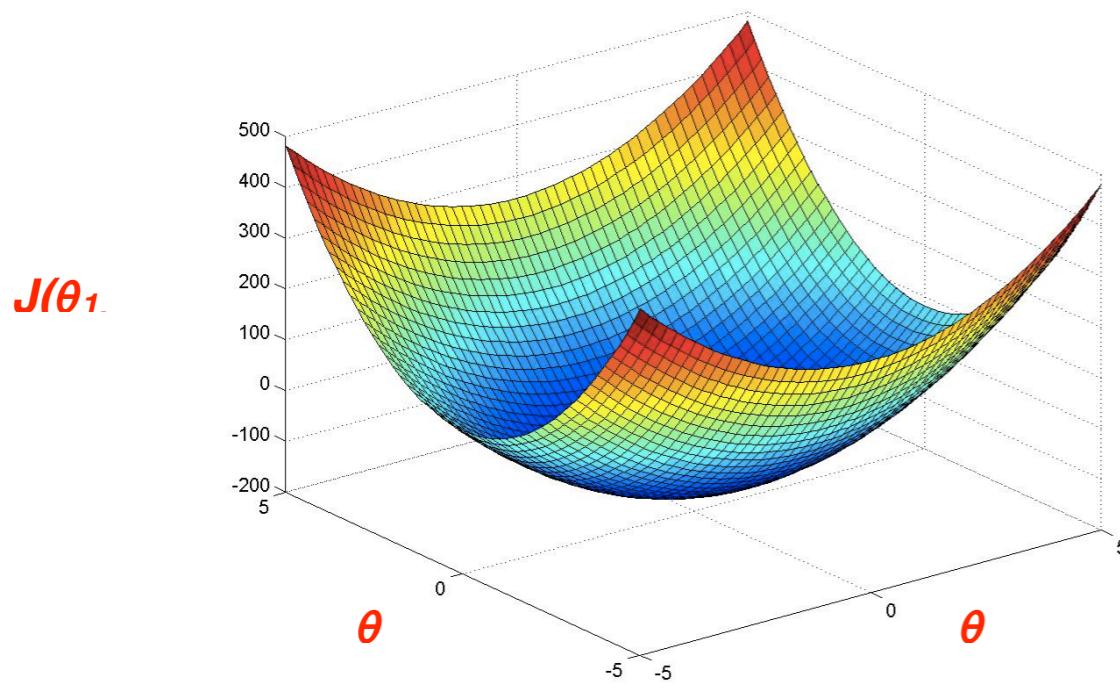
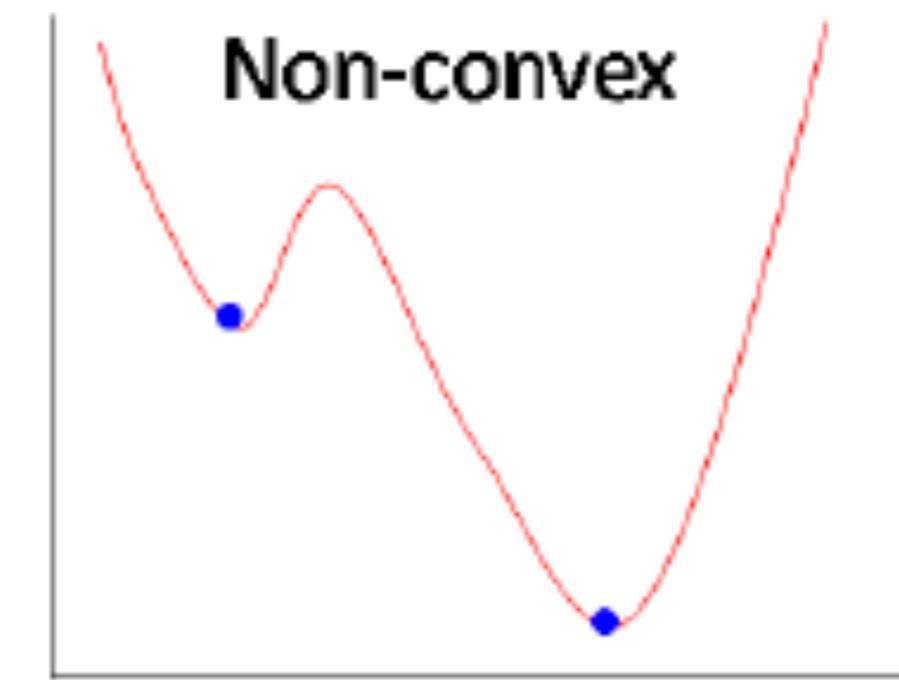
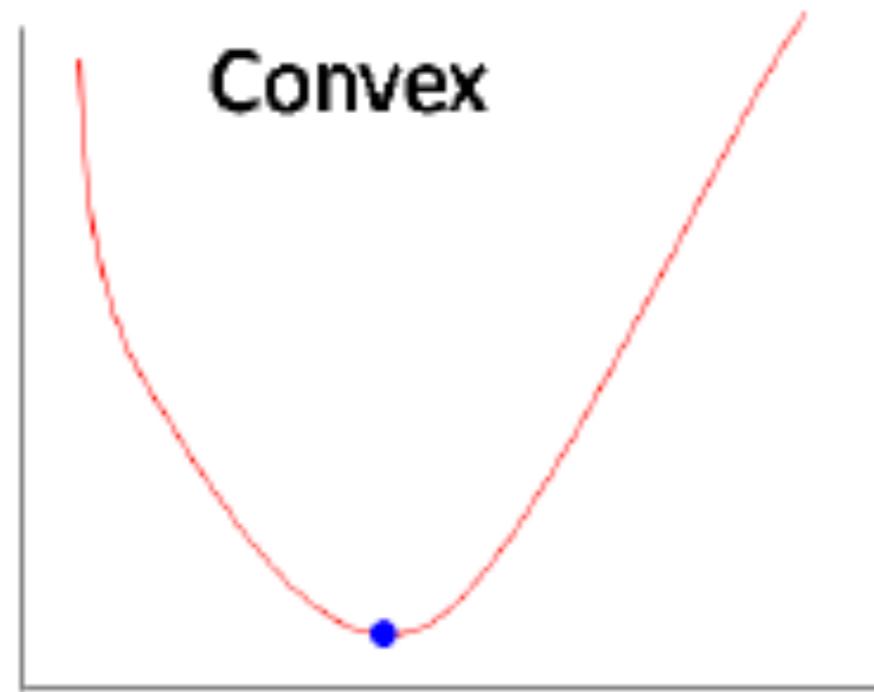
Cost function:

Linear → Logistic Regression

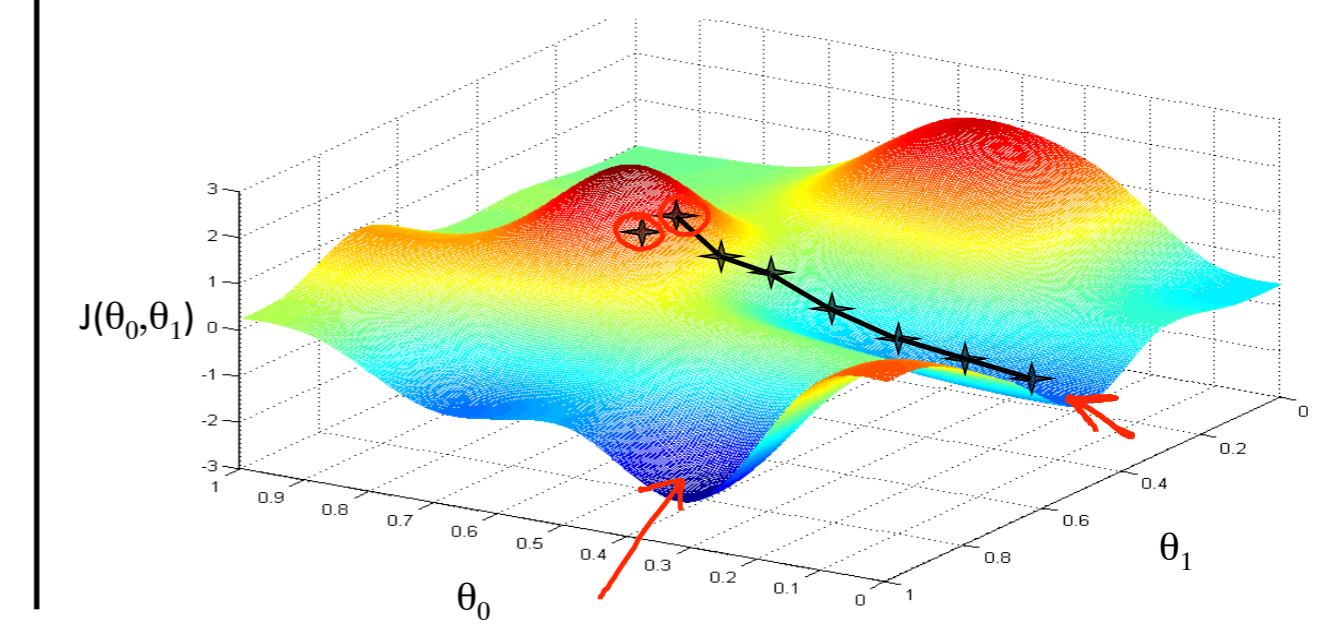
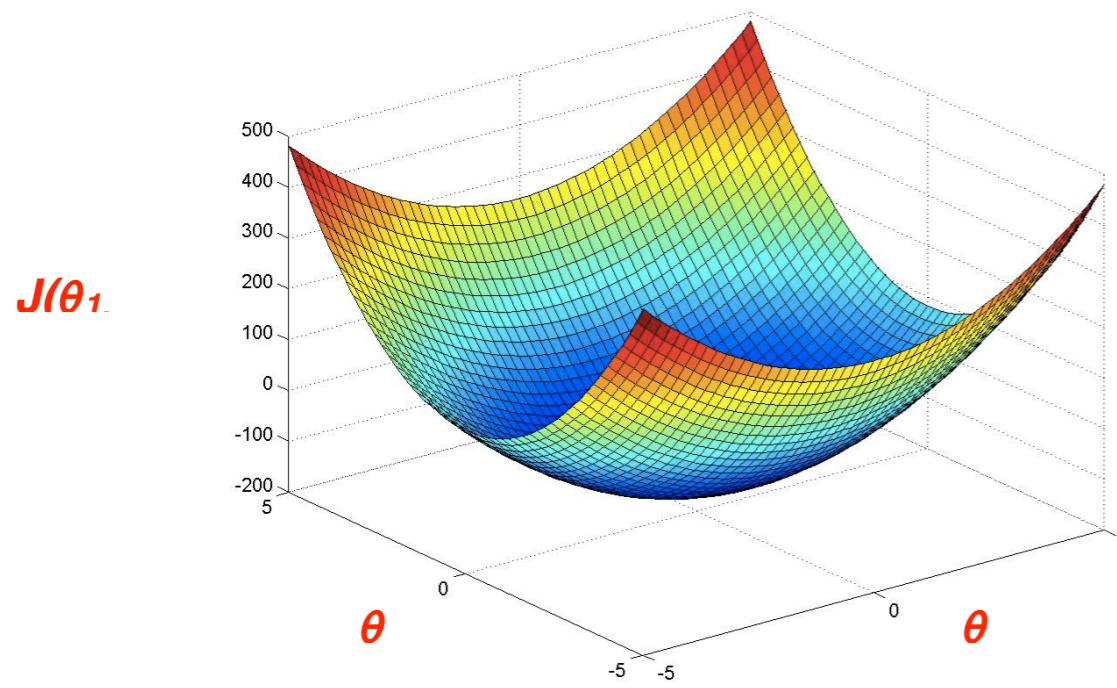
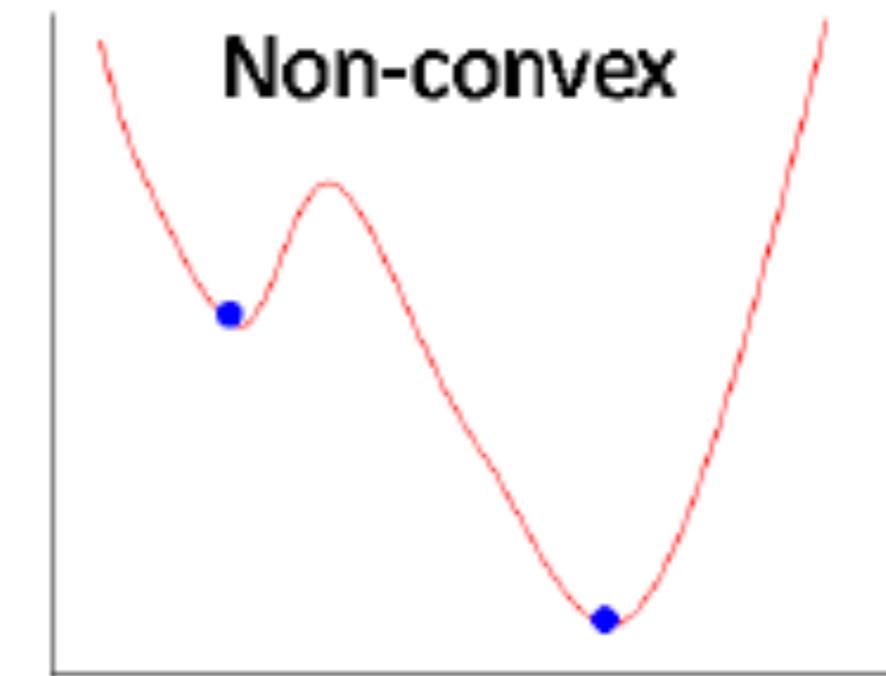
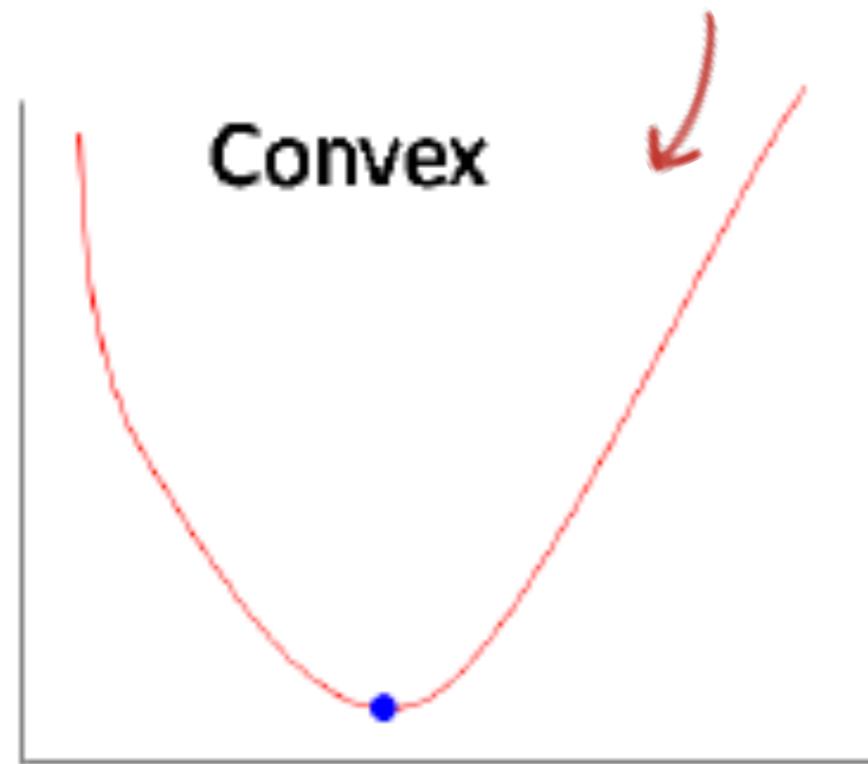
- Linear Regression:
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$$

||
$$\text{Cost}(h_\theta(x), y)$$
- Logistic Regression:
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

this cost function is non-convex for logistic regression



we want convex! easy gradient descent!



Logistic Regression:

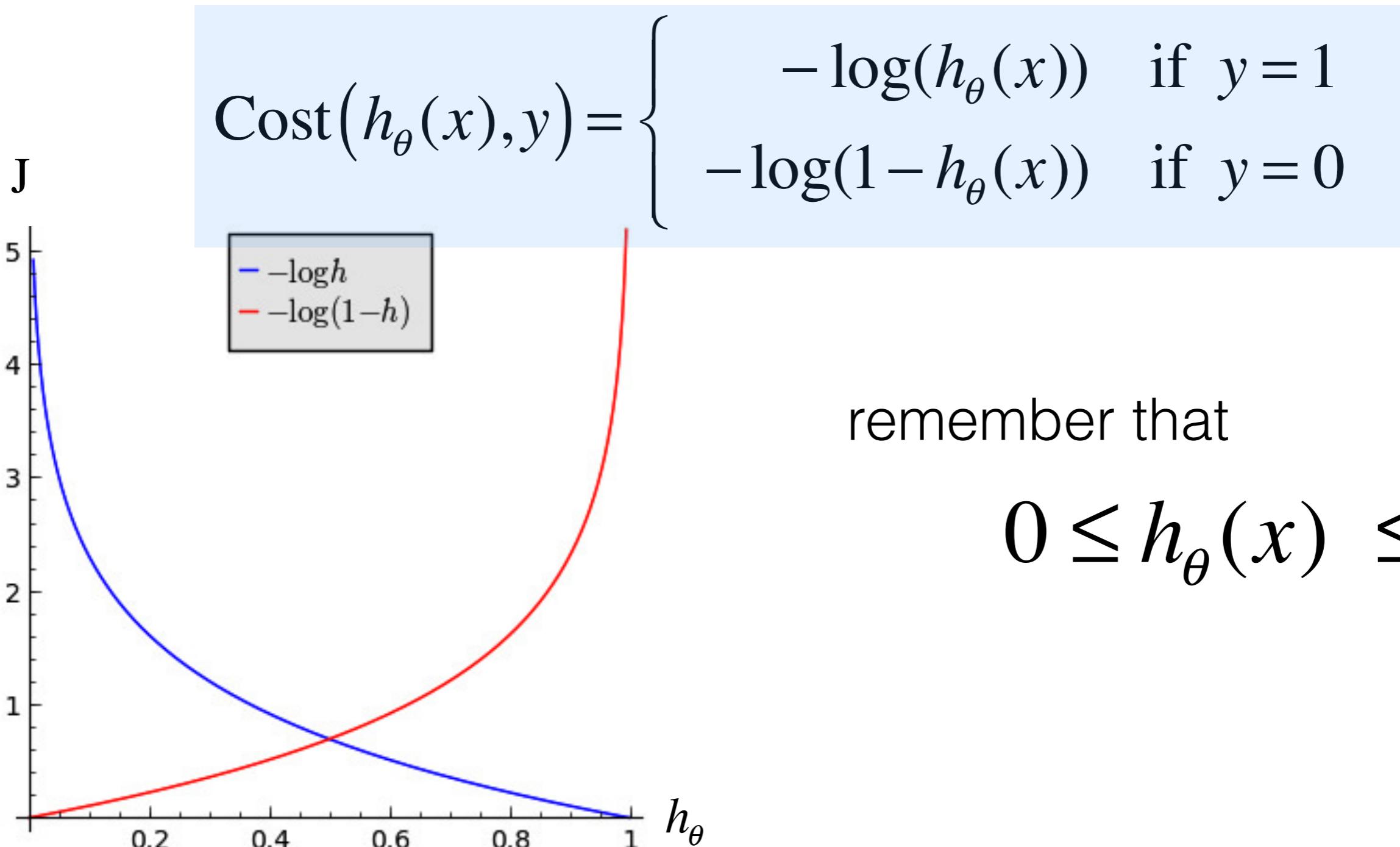
Cost Function

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

remember that

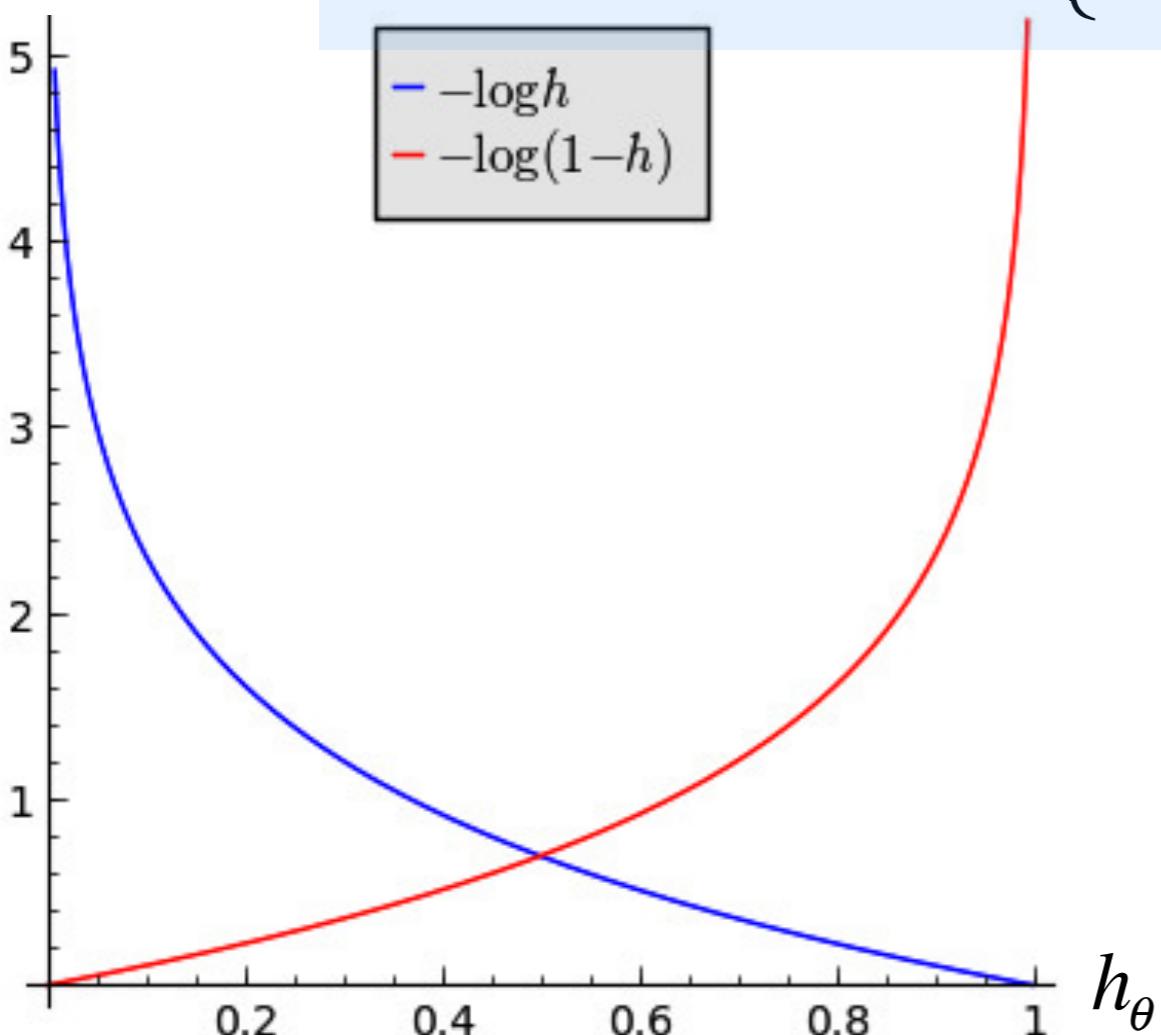
$$0 \leq h_\theta(x) \leq 1$$

Logistic Regression: Cost Function



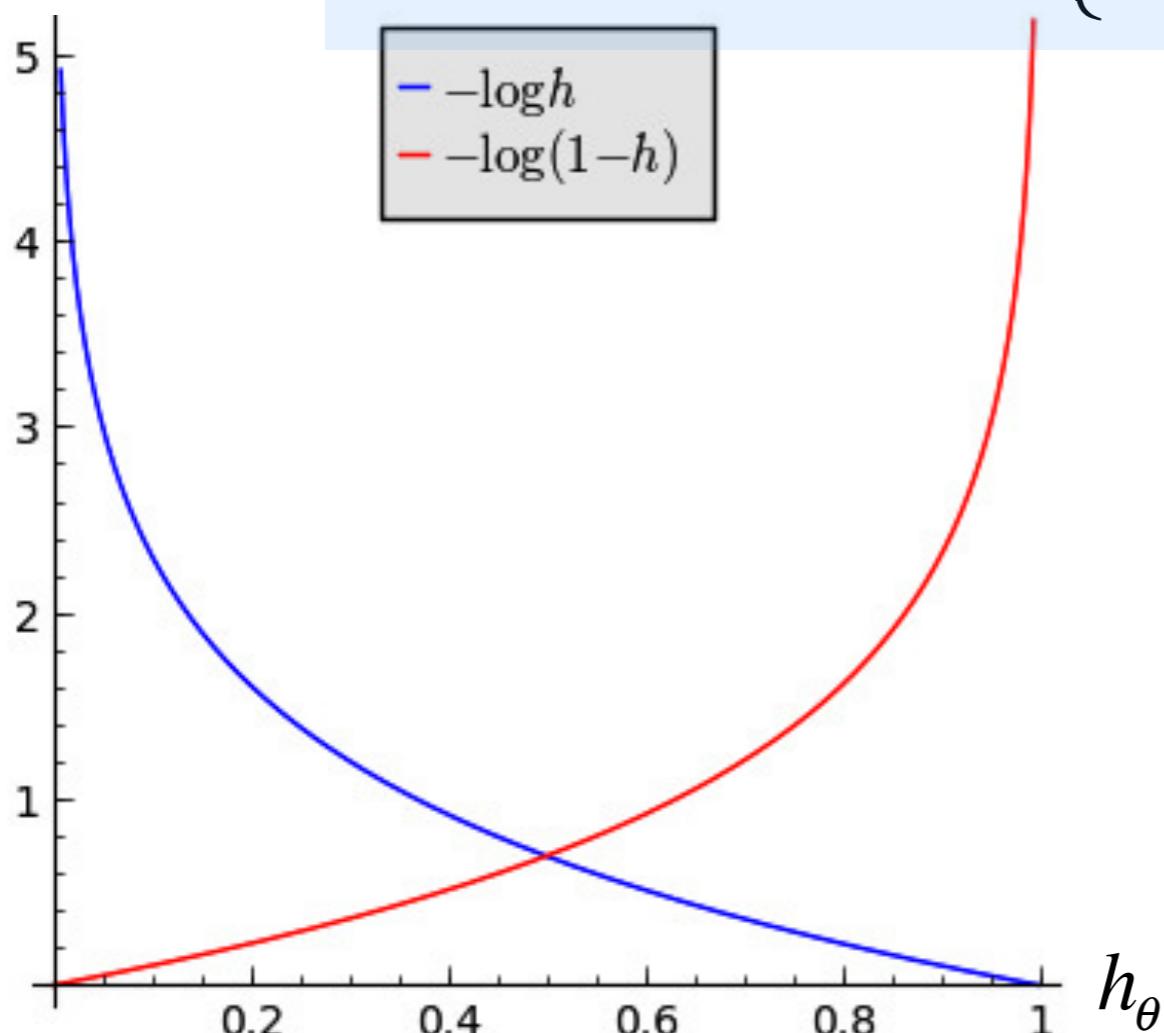
Logistic Regression: Cost Function

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



Logistic Regression: Cost Function

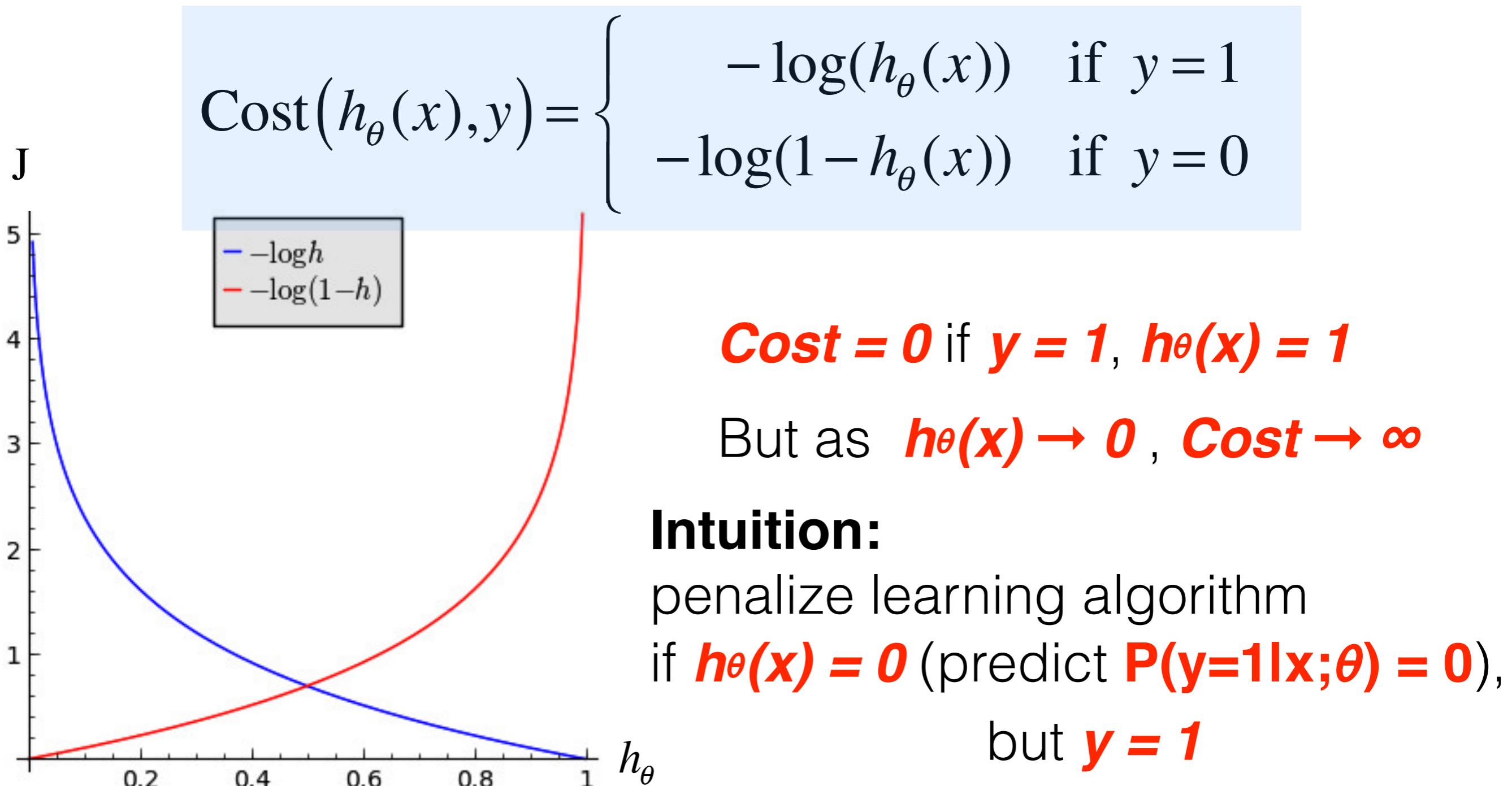
$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



Cost = 0 if $y = 1$, $h_\theta(x) = 1$

But as $h_\theta(x) \rightarrow 0$, **Cost $\rightarrow \infty$**

Logistic Regression: Cost Function



Logistic Regression: Cost Function

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

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

remember that $y = 0$ or 1 always

Logistic Regression:

Cost Function

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

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

remember that $y = 0$ or 1 always

the same



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

Logistic Regression

- **Cost Function:**

$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^m \text{Cost}\left(h_{\theta}(x^{(i)}) - y^{(i)}\right) \\ &= -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \end{aligned}$$

- **Goal:**

learn parameters θ to minimize $J(\theta)$

- **Hypothesis (to make a prediction):**
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

 $P(y=1|x;\theta)$

Logistic Regression:

Gradient Descent

repeat until convergence {

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

learning rate

simultaneous update
for all θ_j

}

Logistic Regression:

Gradient Descent

repeat until convergence {

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

simultaneous update
for all θ_j

}

learning rate

training examples

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

Logistic Regression:

Gradient Descent

repeat until convergence {

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

simultaneous update
for all θ_j

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Logistic Regression:

Gradient Descent

repeat until convergence {

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

simultaneous update
for all θ_j

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

This look the same as linear regression!!???

Logistic Regression:

Gradient Descent

repeat until convergence {

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

||

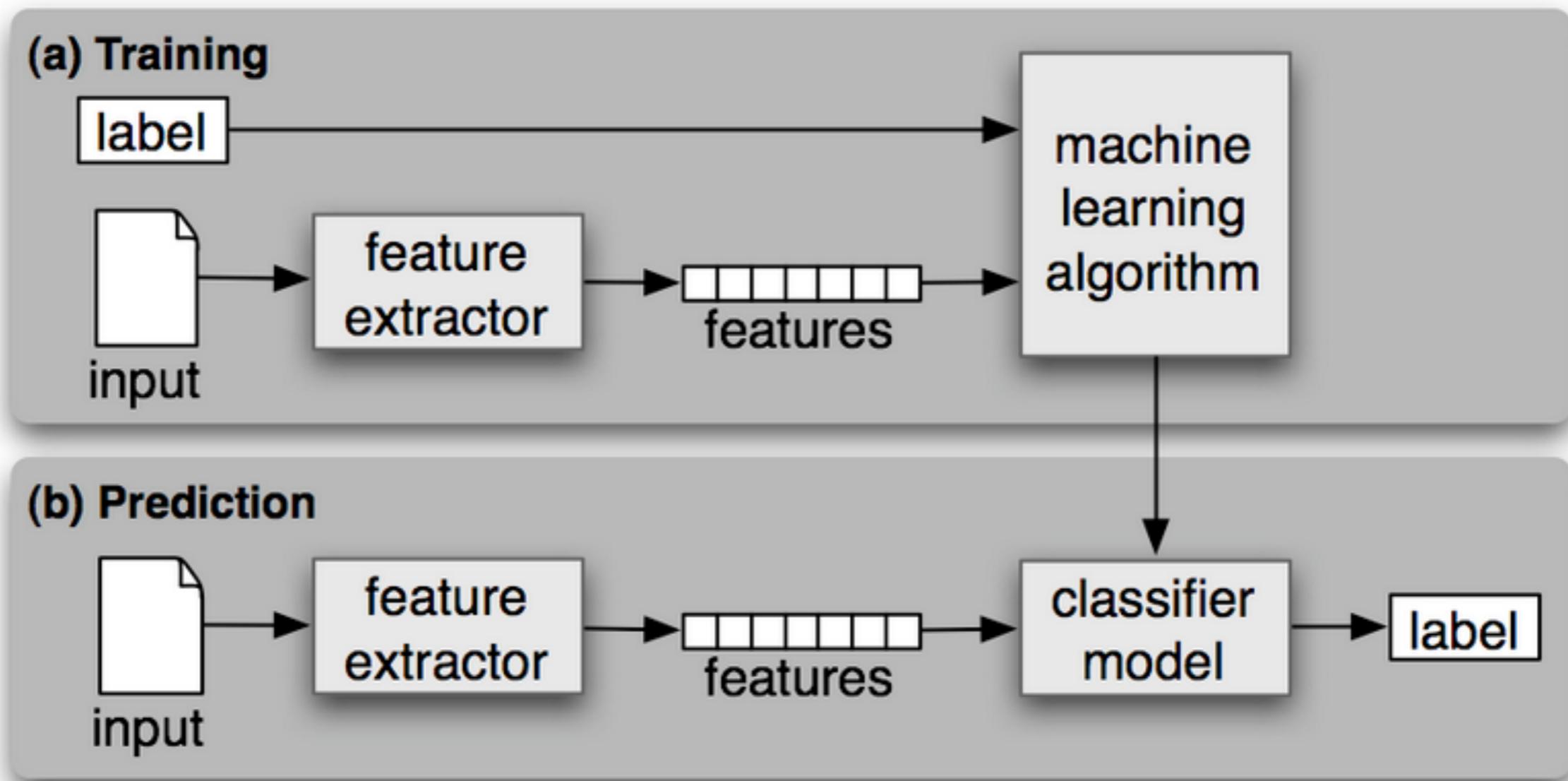
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

simultaneous update
for all θ_j

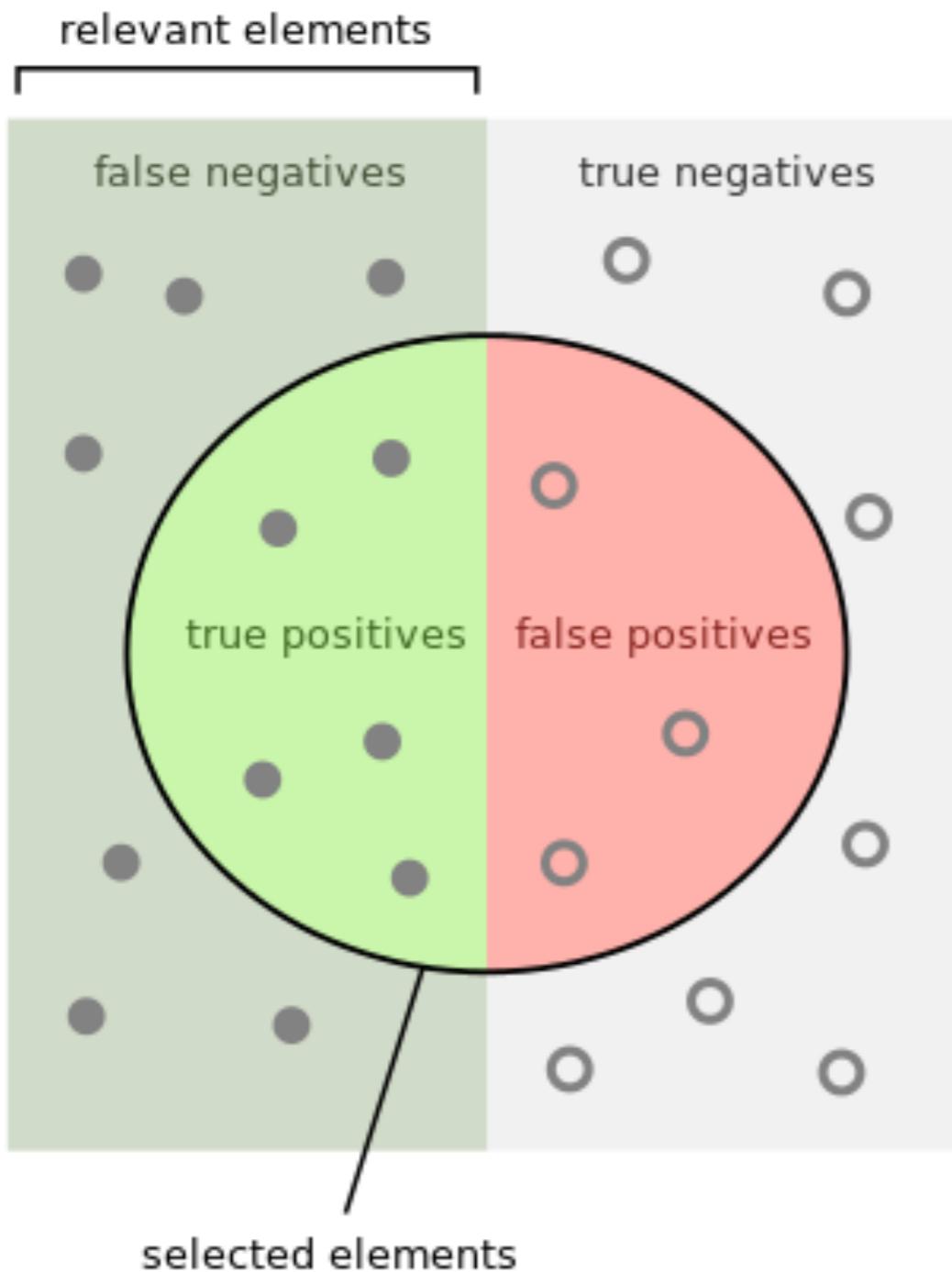
$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

using different hypothesis from linear regression

[Recap] Classification Method: Supervised Machine Learning



Classification Evaluation



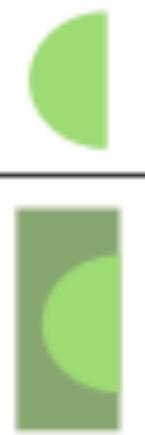
How many selected items are relevant?

Precision =



How many relevant items are selected?

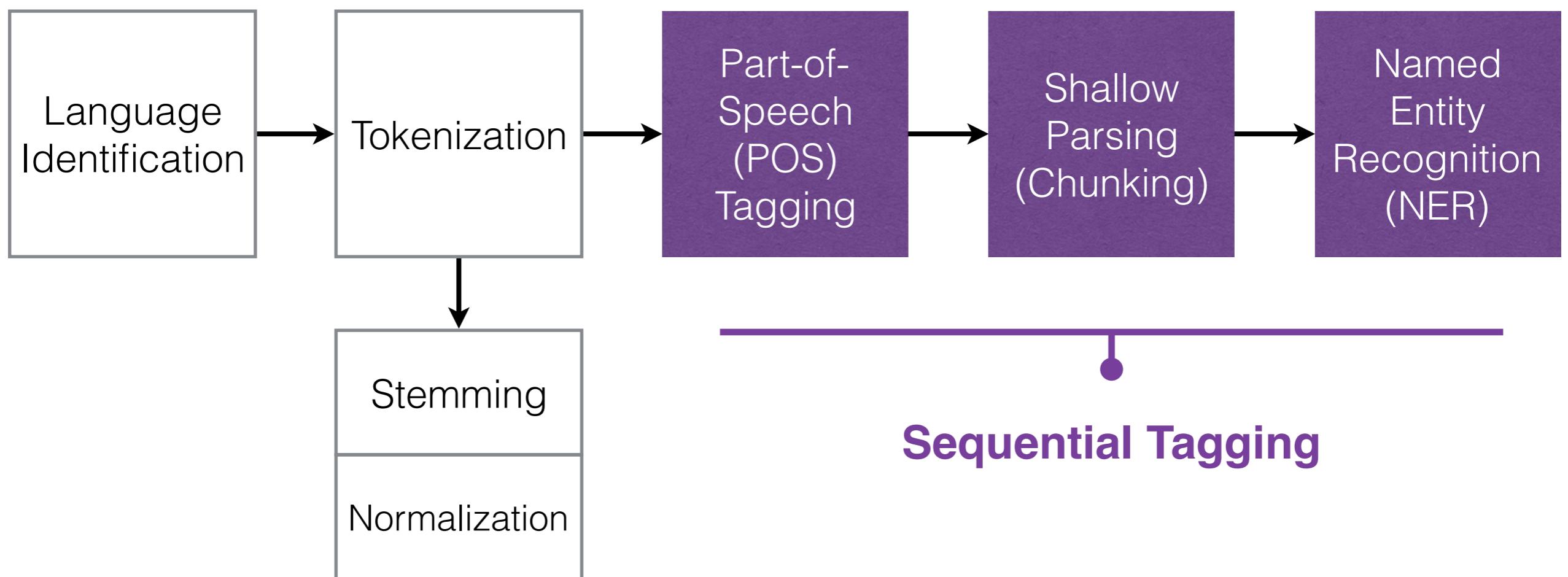
Recall =



F-measure:

$$F_1 = \frac{2 \cdot \textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

NLP Pipeline (next)

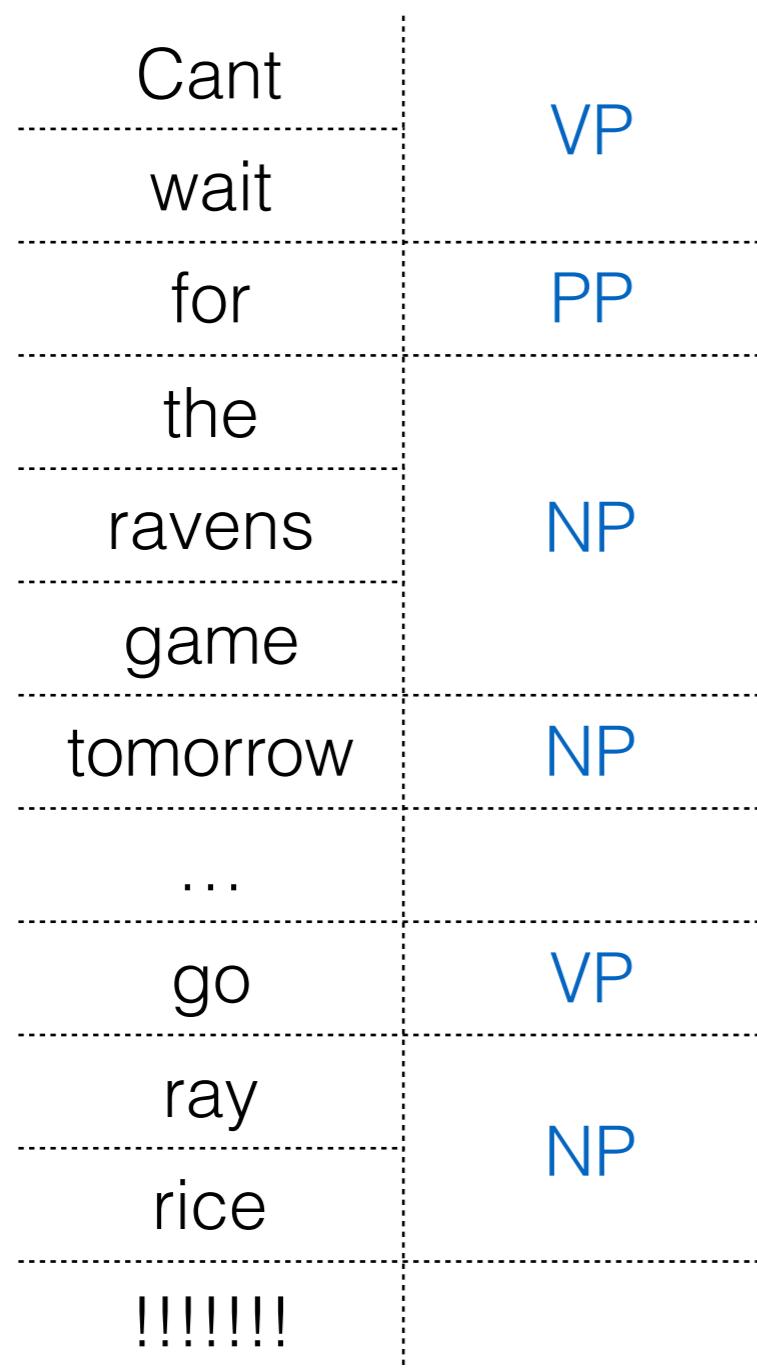


Part-of-Speech (POS) Tagging

Cant	MD
wait	VB
for	IN
the	DT
ravens	NNP
game	NN
tomorrow	NN
...	:
go	VB
ray	NNP
rice	NNP
!!!!!!	.



Chunking



Named Entity Recognition(NER)

Cant	
wait	
for	
the	
ravens	ORG
game	
tomorrow	
...	
go	
ray	
rice	PER
!!!!!!	.



ORG: organization

PER: person

LOC: location

IO tag encoding

Cant		VP	
wait		VP	
for	PP	PP	
the		NP	
ravens	NP	NP	
game		NP	
tomorrow	NP	NP	
...		O	
go	VP	VP	
ray		NP	
rice	NP	NP	
!!!!!!		O	



IO tag encoding

Cant	VP	VP	B-VP
wait		VP	I-VP
for	PP	PP	B-PP
the		NP	B-NP
ravens	NP	NP	I-NP
game		NP	I-NP
tomorrow	NP	NP	B-NP
...		O	O
go	VP	VP	B-VP
ray		NP	B-VP
rice	NP	NP	I-VP
!!!!!!		O	O



I: Inside

O: outside

B: Begin

BIO allows separation of adjacent chunks/entities

Classification Method: Supervised Machine Learning

- Naïve Bayes
- Logistic Regression
- Support Vector Machines (SVM)
- ...
- Hidden Markov Model (HMM)
- Conditional Random Fields (CRF)
- ...



**sequential
models**

Classification Method: Sequential Supervised Learning

- Input:
 - rather than just individual examples ($w_1 = \text{the}$, $c_1 = \text{DT}$)
 - a training set consists of m sequences of labeled examples
 $(x_1, y_1), \dots, (x_m, y_m)$
- $x_1 = \langle \text{the back door} \rangle$ and $y_1 = \langle \text{DT JJ NN} \rangle$
- Output:
 - a learned classifier to predict label sequences $\gamma: x \rightarrow y$

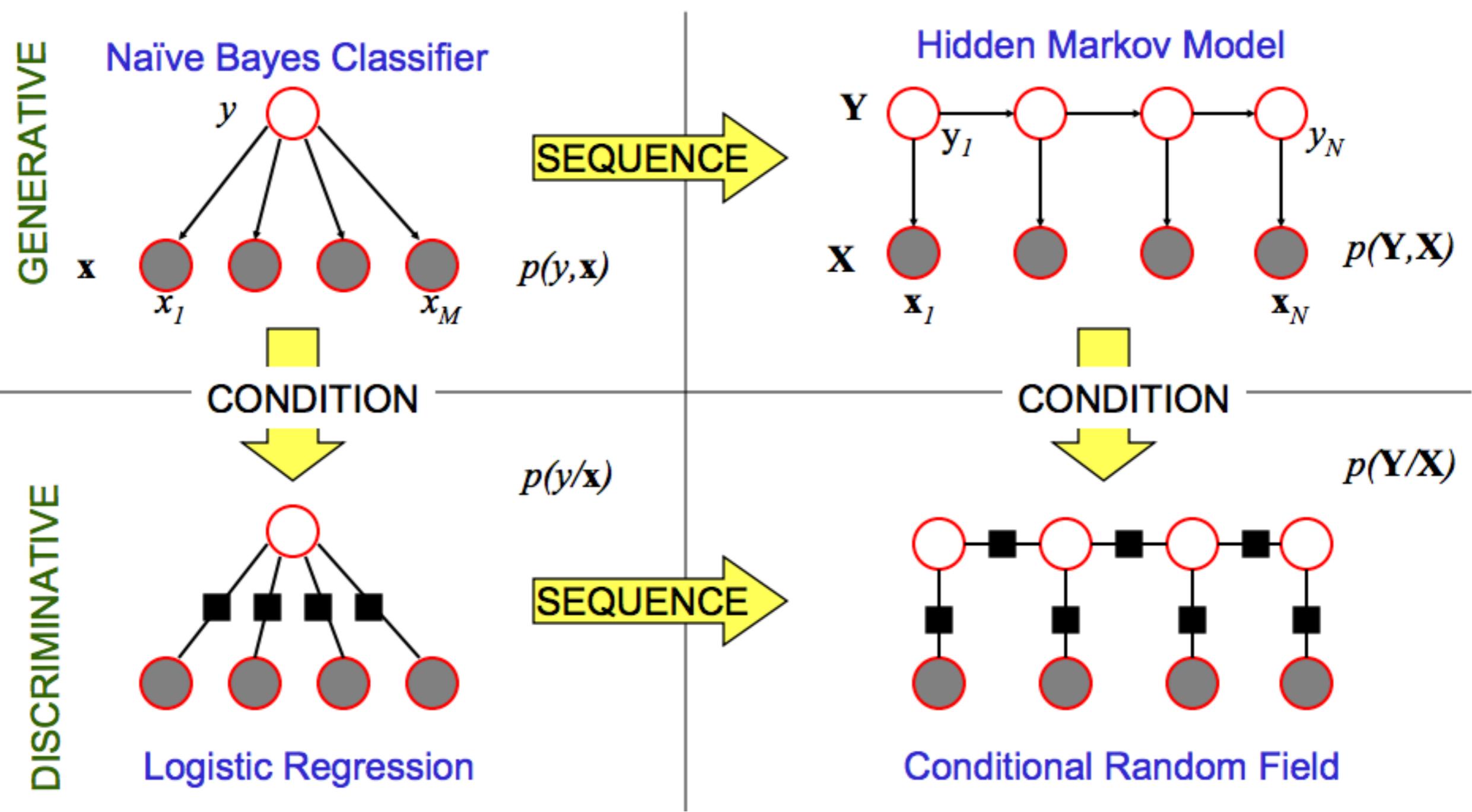
Features for Sequential Tagging

- Words:
 - current words
 - previous/next word(s) — context
- Other linguistic information:
 - word substrings
 - word shapes
 - POS tags
- Contextual Labels
 - previous (and perhaps next) labels

word shapes

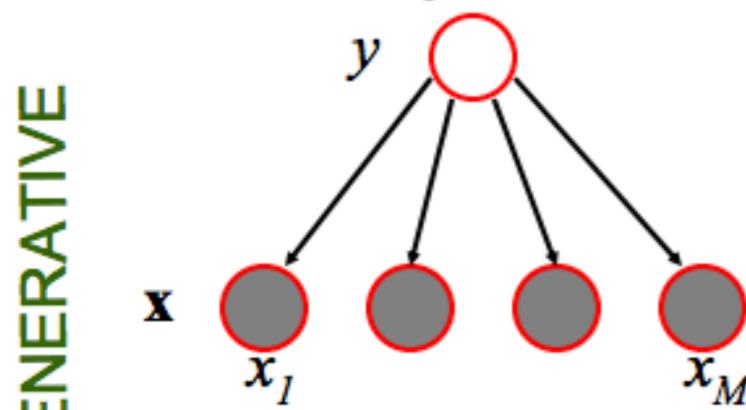
Varicella-zoster	Xx-xxx
mRNA	xXXX
CPA1	XXXd

Probabilistic Graphical Models



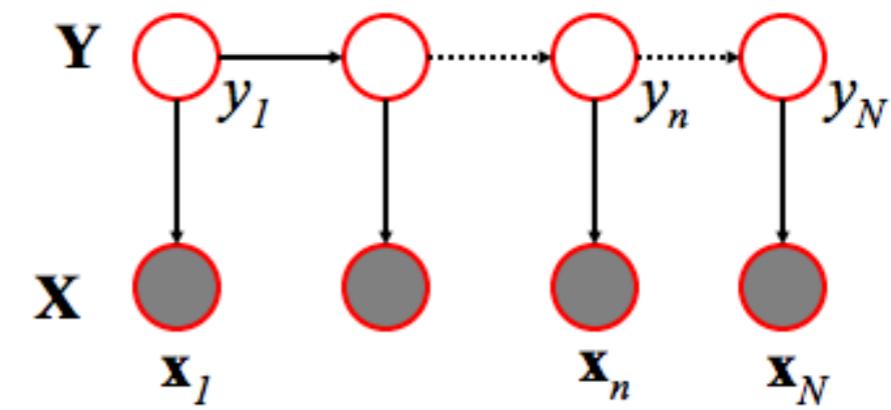
Probabilistic Graphical Models

Naïve Bayes Classifier



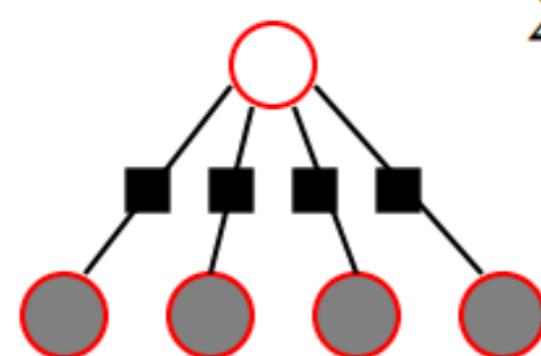
$$p(y, \mathbf{x}) = p(y) \prod_{m=1}^M p(x_m | y)$$

Hidden Markov Model



$$p(\mathbf{Y}, \mathbf{X}) = \prod_{n=1}^N p(y_n | y_{n-1}) p(x_n | y_n)$$

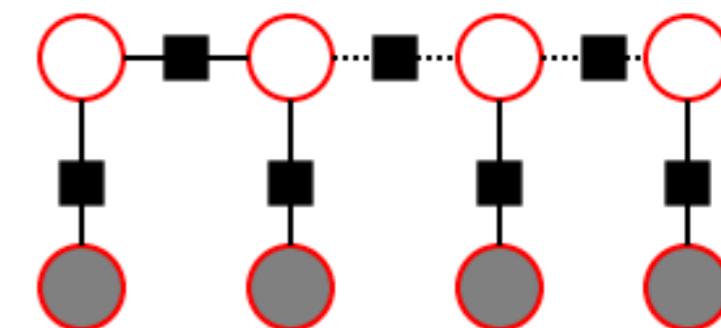
DISCRIMINATIVE



Logistic Regression

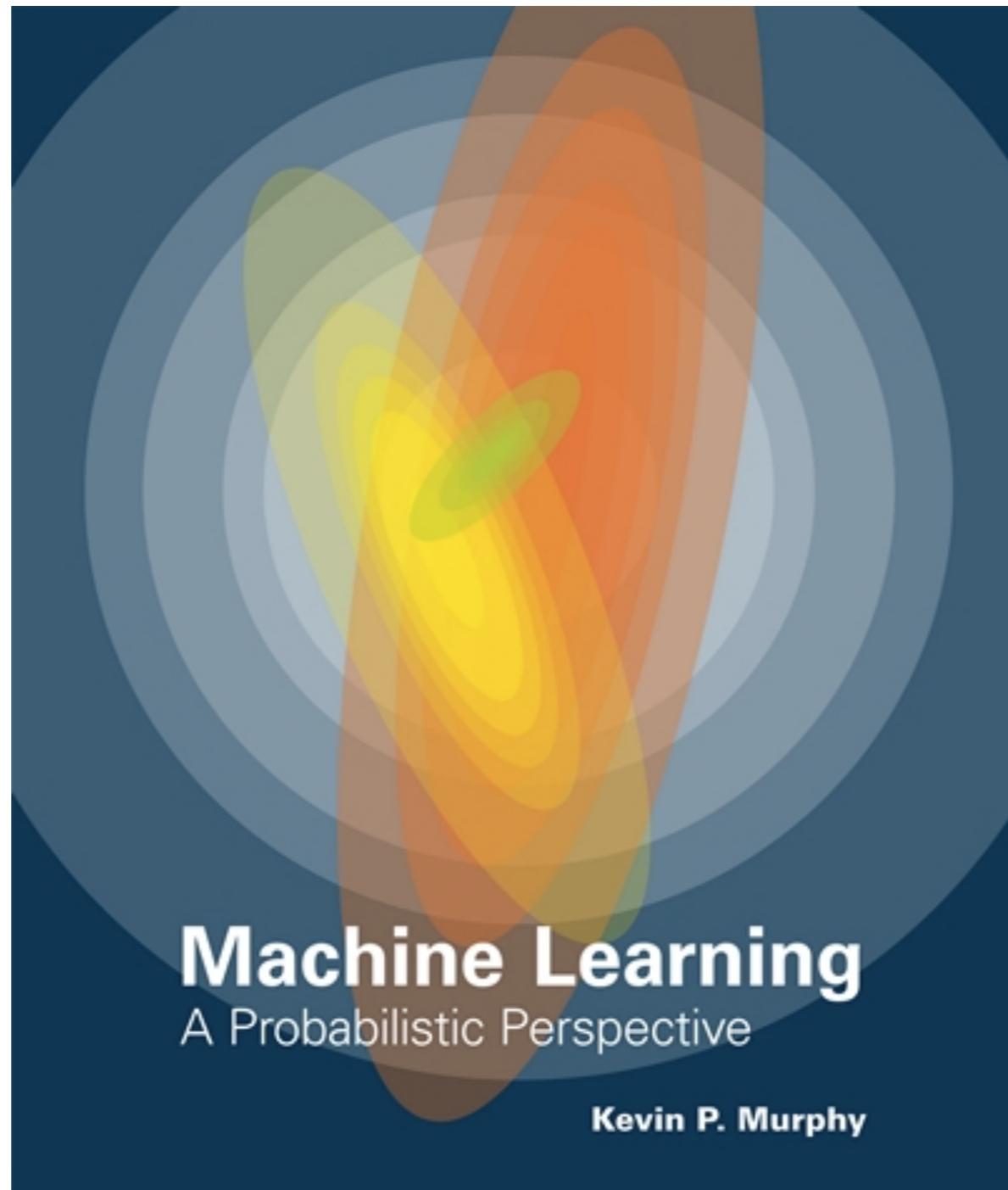
$$p(y | \mathbf{x}) = \frac{\exp \left\{ \sum_{m=1}^M \lambda_m f_m(y, \mathbf{x}) \right\}}{\sum_{y'} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y', \mathbf{x}) \right\}}$$

$$p(\mathbf{Y} | \mathbf{X}) = \frac{\exp \left\{ \sum_{m=1}^M \lambda_m f_m(y_n, y_{n-1}, \mathbf{x}_n) \right\}}{\sum_{y'} \exp \left\{ \sum_{m=1}^M \lambda_m f_m(y'_n, y'_{n-1}, \mathbf{x}_n) \right\}}$$



Conditional Random Field

Probabilistic Graphical Models





Instructor: Wei Xu

<http://web.cse.ohio-state.edu/~weixu/>

Course Website: socialmedia-class.org