Social Media & Text Analysis

lecture 8 - Paraphrase Identification and Logistic Regression

CSE 5539-0010 Ohio State University

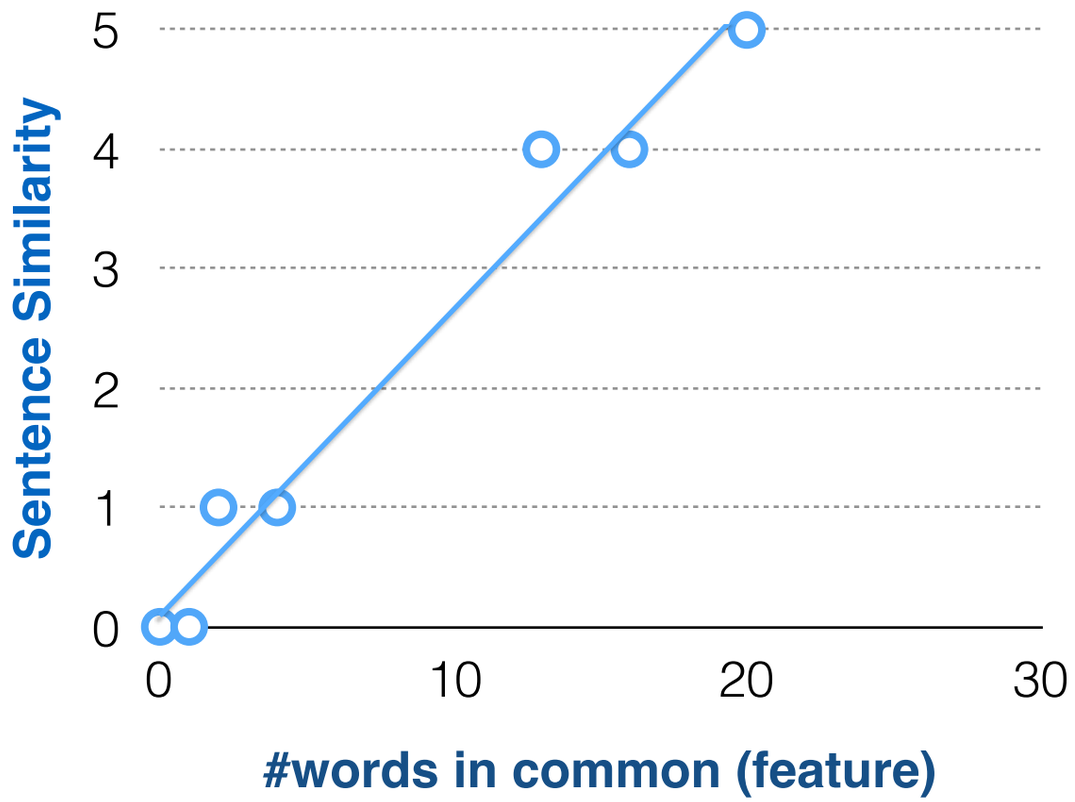
Instructor: Wei Xu

Website: socialmedia-class.org

Many slides adapted from Andrew Ng

(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

- **Hypothesis**:

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

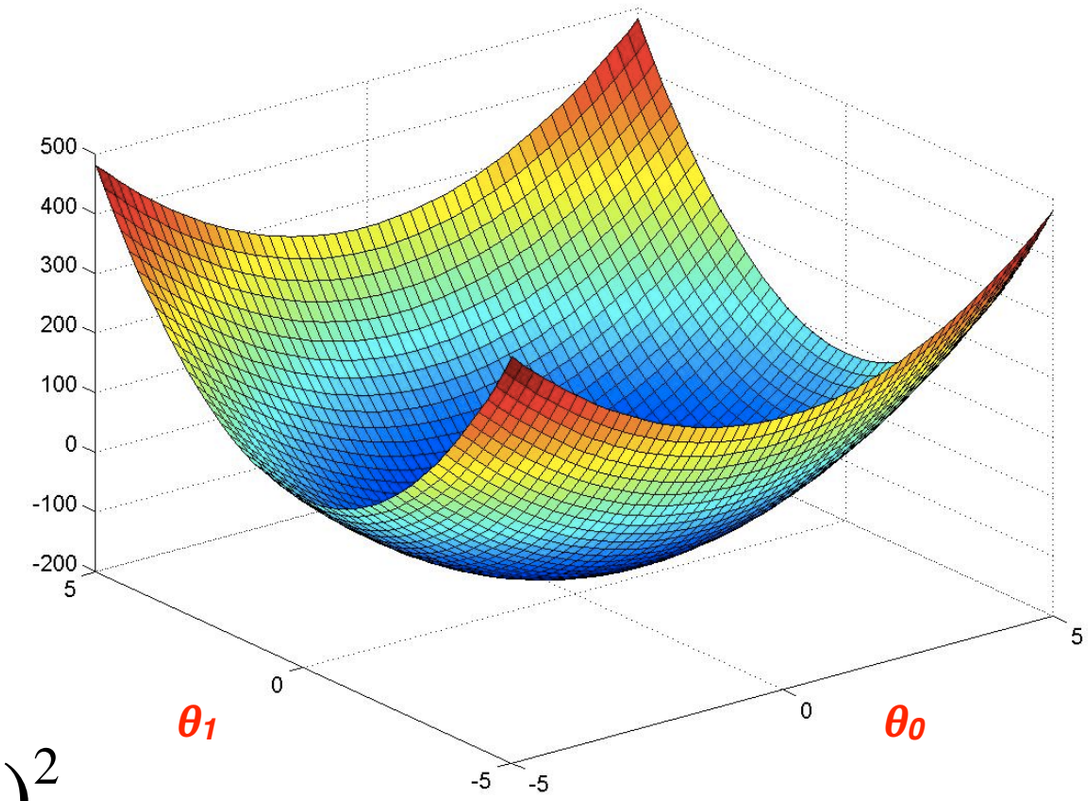
- **Parameters:**

θ_0, θ_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:

Model Representation

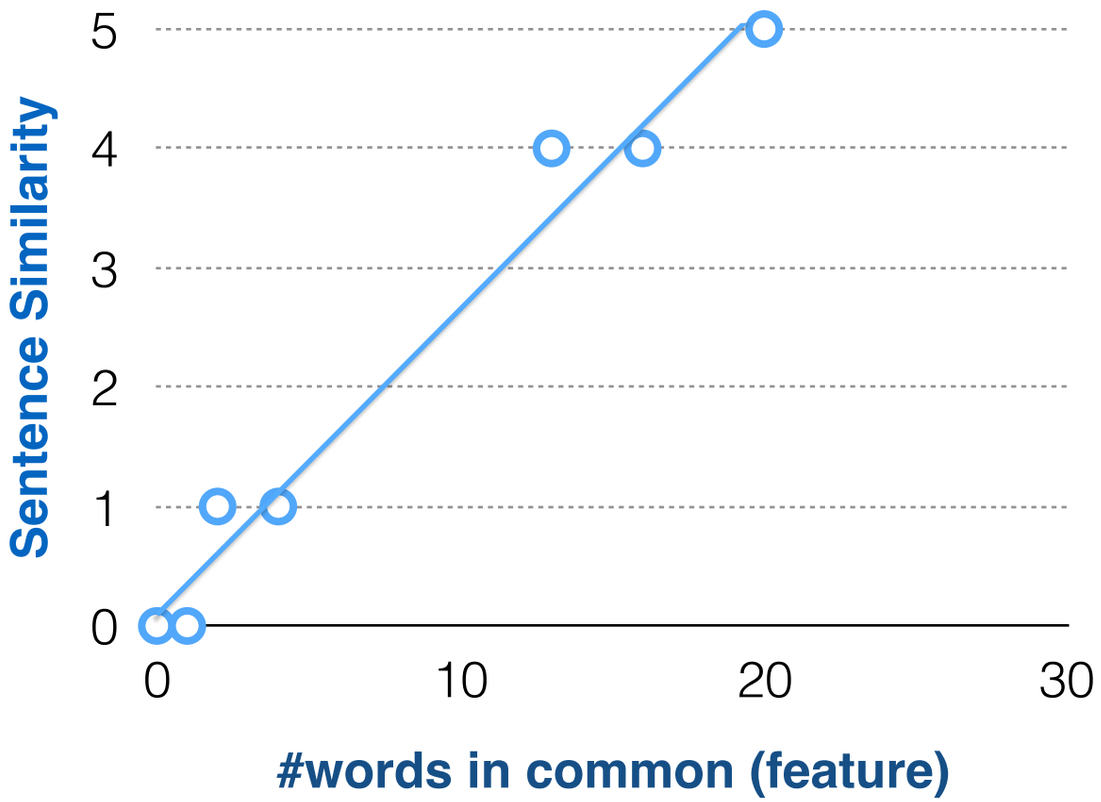
#words in common (*x*)	Sentence Similarity (*y*)
1	0
4	1
13	4
18	5
…	…

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

Source: many following slides are adapted from Andrew Ng

(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 {

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

simultaneous update for j=0 and j=1

}

learning rate

Linear Regression w/ one variable:

Gradient Descent

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

Cost Function

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

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = ?$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = ?$$

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

$$= \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \underline{\theta_1 x^{(i)} - y^{(i)}})^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0^2 + 2\theta_0 \cdot (\theta_1 x^{(i)} - y^{(i)}) + (\theta_1 x^{(i)} - y^{(i)})^2 \right)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{2m} \sum_{i=1}^{m} (2\theta_0 + 2 \cdot (\theta_1 x^{(i)} - y^{(i)}))$$

$$= \frac{1}{m} \sum_{i=1}^{m} (\underbrace{\theta_0 + \theta_1 x^{(i)}}_{H_\theta(x^{(i)})} - y^{(i)})$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1^2 x^{(i)2} + 2 \cdot \theta_1 x^{(i)} (\theta_0 - y^{(i)}) + (\theta_0 - y^{(i)})^2 \right)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{2m} \sum_{i=1}^{m} 2 \cdot x^{(i)2} \theta_1 + 2 \cdot x^{(i)} (\theta_0 - y^{(i)})$$

$$= \frac{1}{m} \sum_{i=1}^{m} (\underbrace{x^{(i)} \theta_1 + \theta_0}_{H_\theta(x^{(i)})} - y^{(i)}) \cdot x^{(i)}$$

(Recap) Linear Regression w/ one variable:

Gradient Descent

repeat until convergence {

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

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

}

simultaneous update θ_0, θ_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 $x_0 = 1$)

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

Mancini gets the boot from Man City

Yes!

WORLD OF JENKS IS ON AT 11

World of Jenks is my favorite show on tv

No!

Wei Xu, Alan Ritter, Chris Callison-Burch, Bill Dolan, Yangfeng Ji. “Extracting Lexically Divergent Paraphrases from Twitter” In TACL (2014)

(Recap) Classification Method:

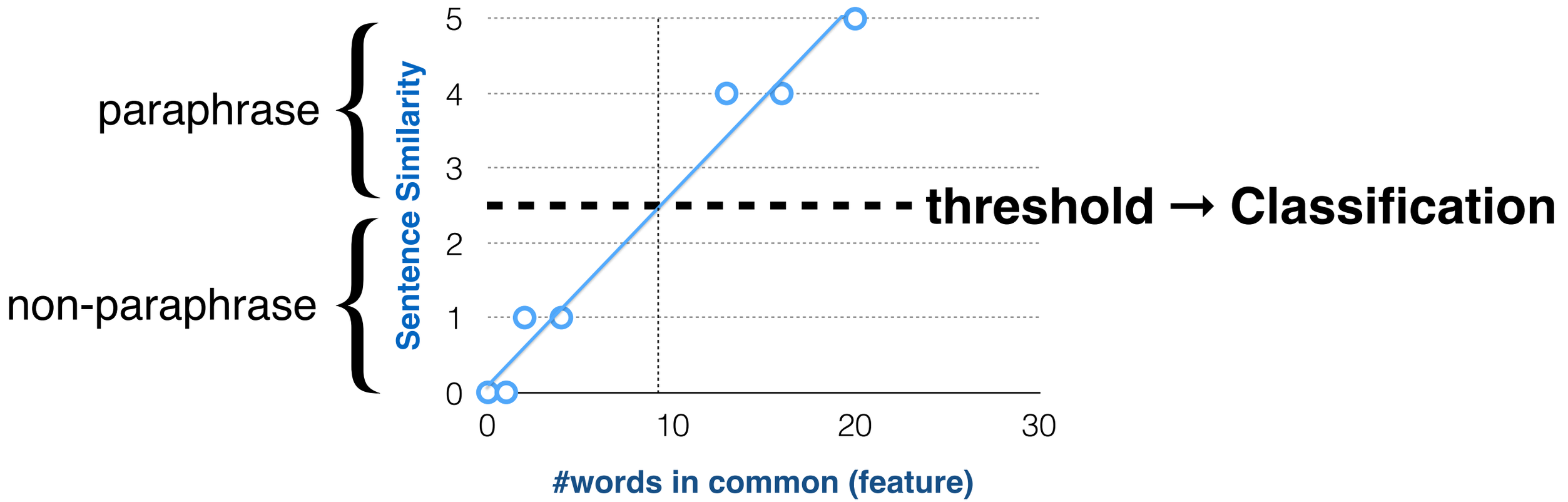
Supervised Machine Learning

- 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 ***$(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$***

- Output:
  - a learned classifier ***$\gamma: x \rightarrow y \in Y$*** (***y = 0*** or ***y = 1***)

(Recap)

Linear Regression

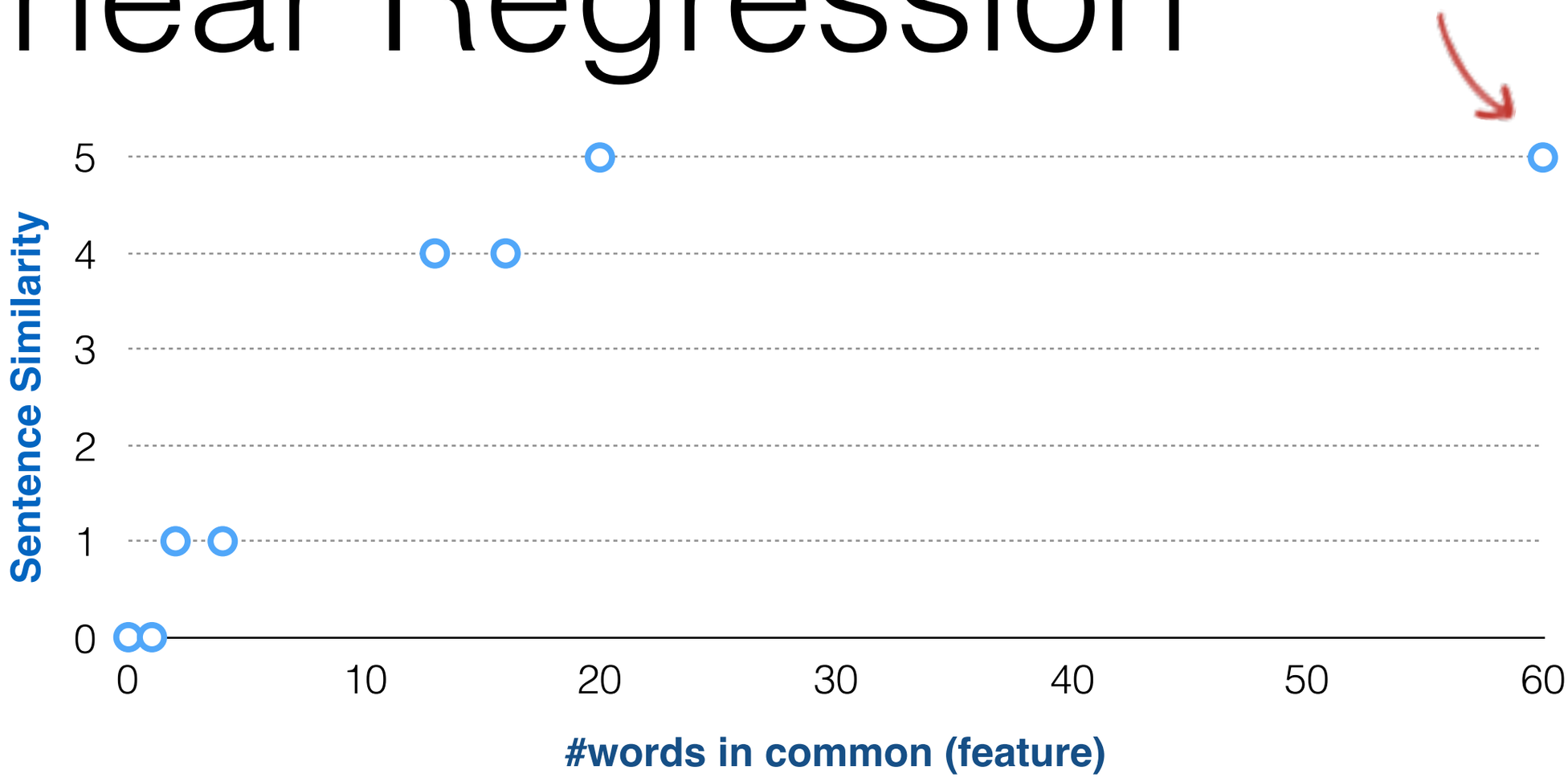


- also supervised learning (learn from annotated data)

- but for **Regression**: predict **real-valued** output
  (Classification: predict discrete-valued output)

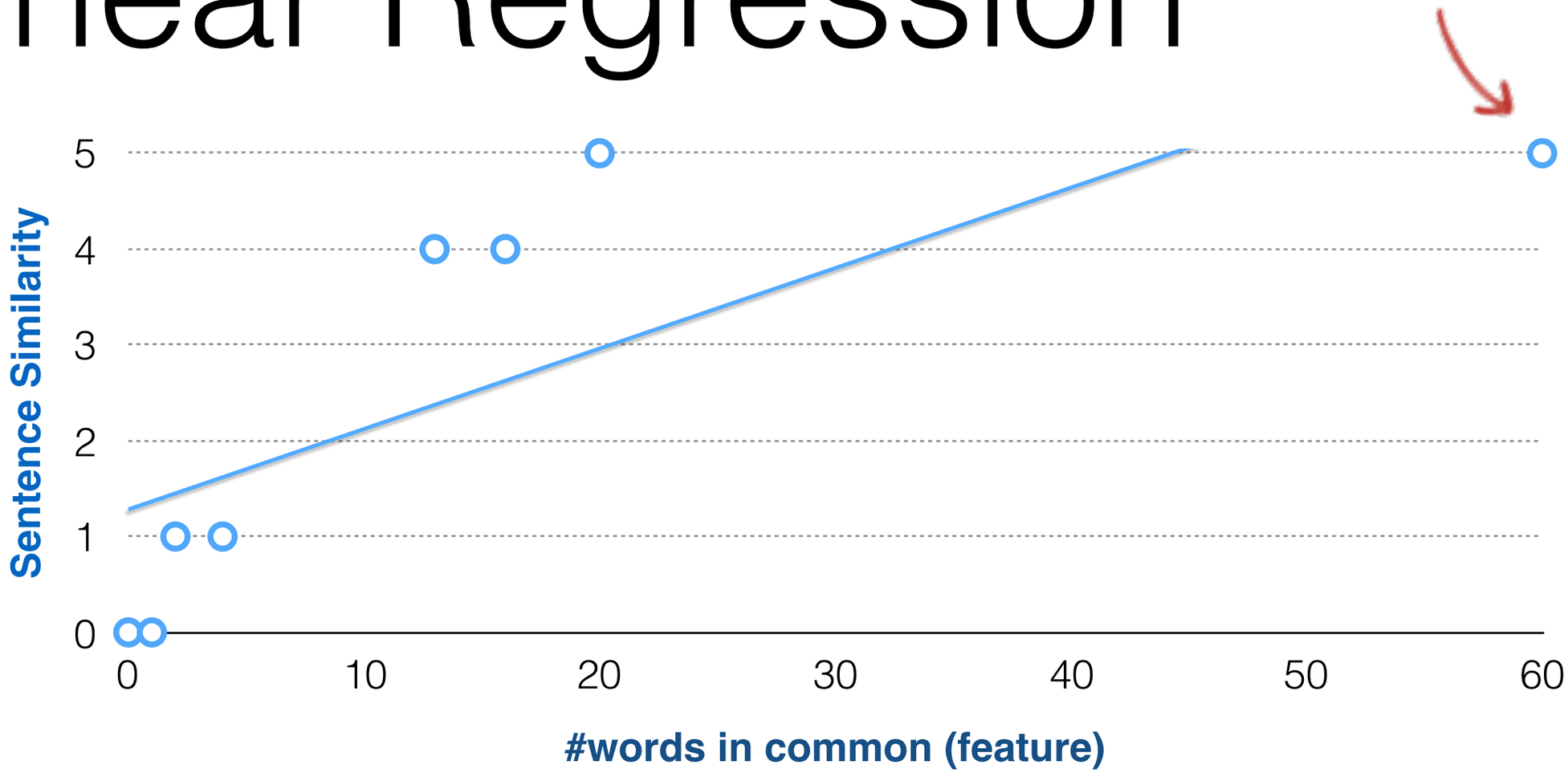
A problem in classification

Linear Regression



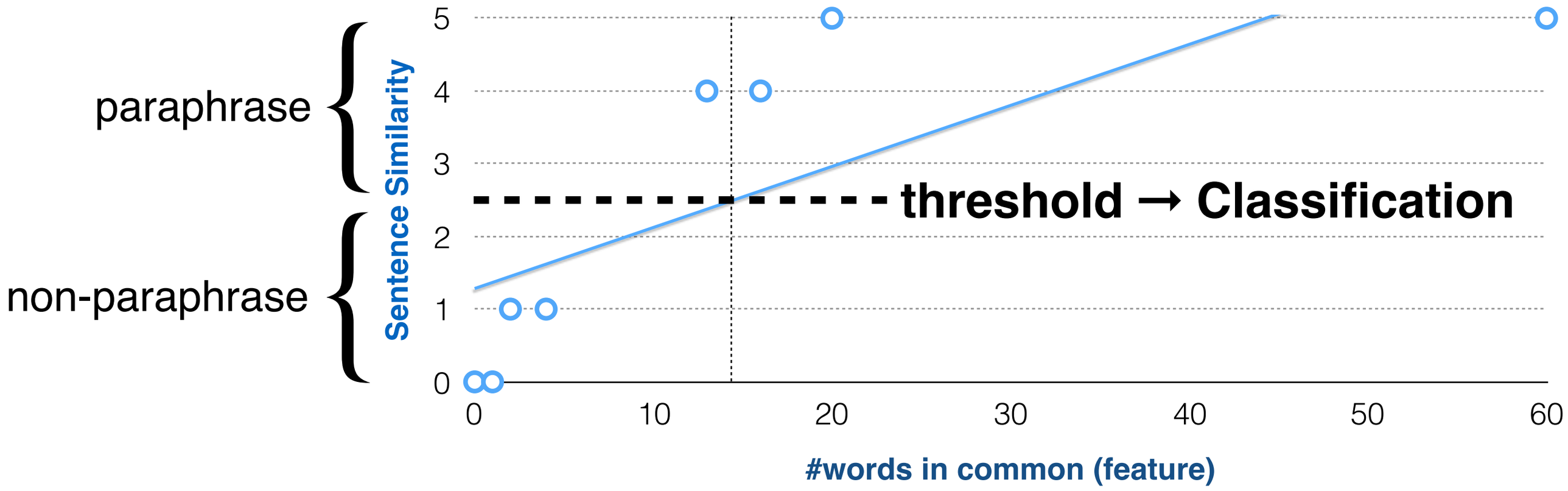
A problem in classification

Linear Regression



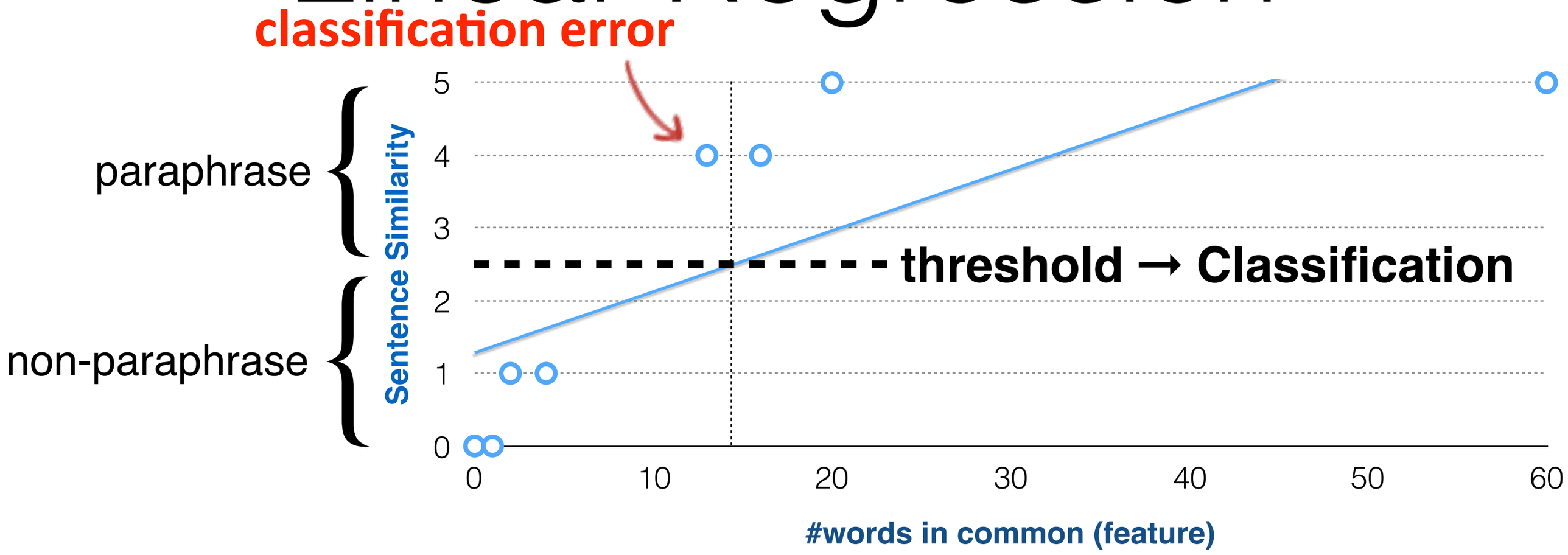
A problem in classification

Linear Regression



A problem in classification

Linear Regression



In practice, do not use linear regression for classification.

(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

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

Hypothesis:

Linear → Logistic Regression

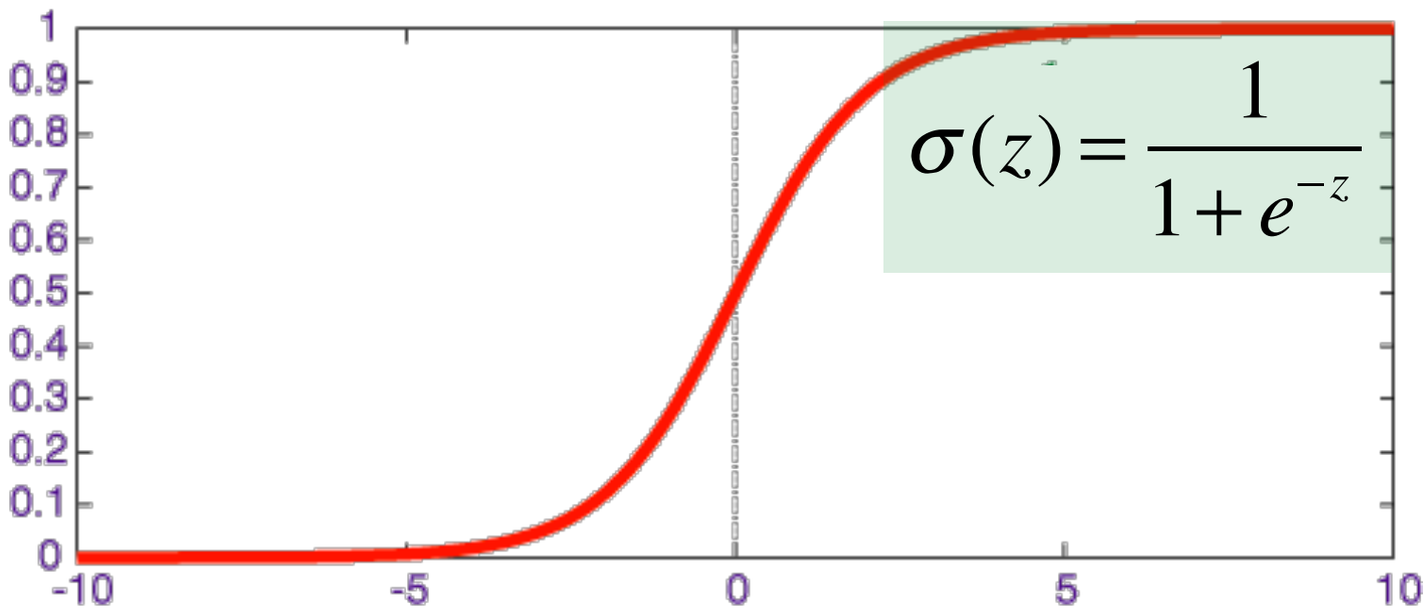
- Linear Regression: $h_\theta(x) = \theta^T x$
- Logistic Regression: $\text{want } 0 \leq h_\theta(x) \leq 1$

Hypothesis:

Linear → Logistic Regression

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

sigmoid (logistic) function



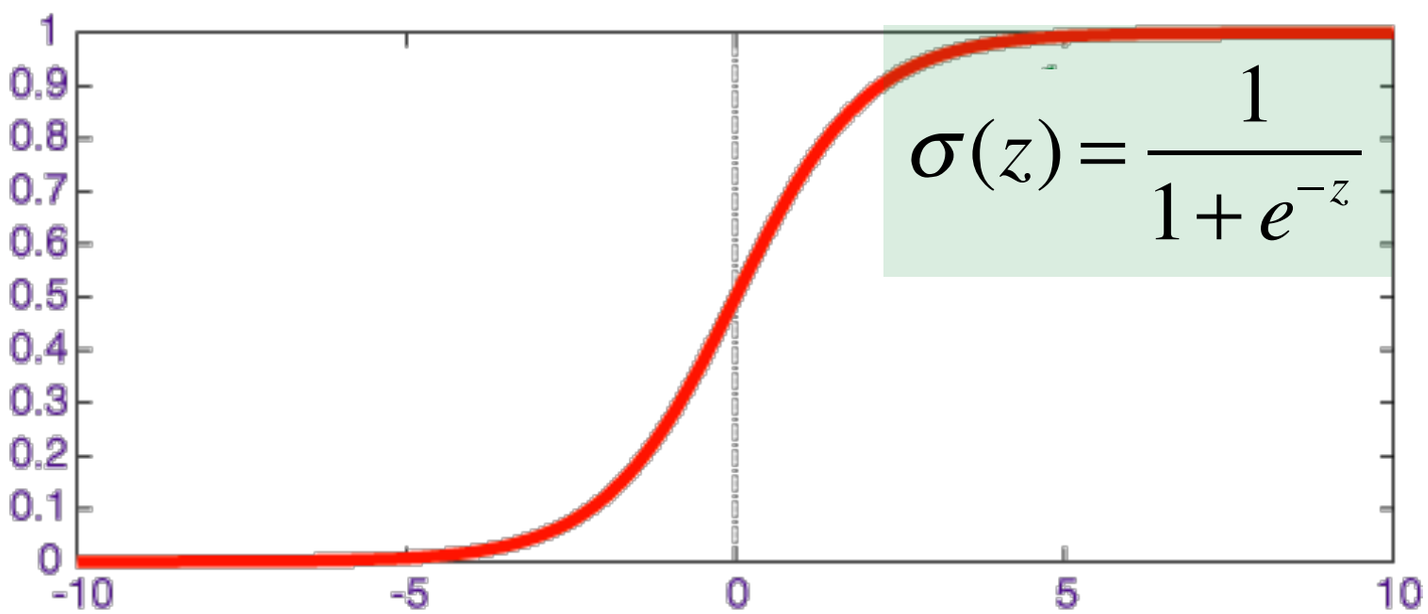
Hypothesis:

Linear → Logistic Regression

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

$$h_\theta(x) = \sigma\left(\theta^T x\right)$$

sigmoid (logistic) function



(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 ***$(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$***

- 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

$$If \quad x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \#\text{words\_in\_common} \end{bmatrix}, \; h_\theta(x) = 0.7$$

70% chance of the sentence pair being paraphrases

Logistic Regression:

Interpretation of Hypothesis

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

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

70% chance of the sentence pair being paraphrases

$$h_\theta(x) = P(y = 1 \mid x; \theta)$$

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

Logistic Regression:

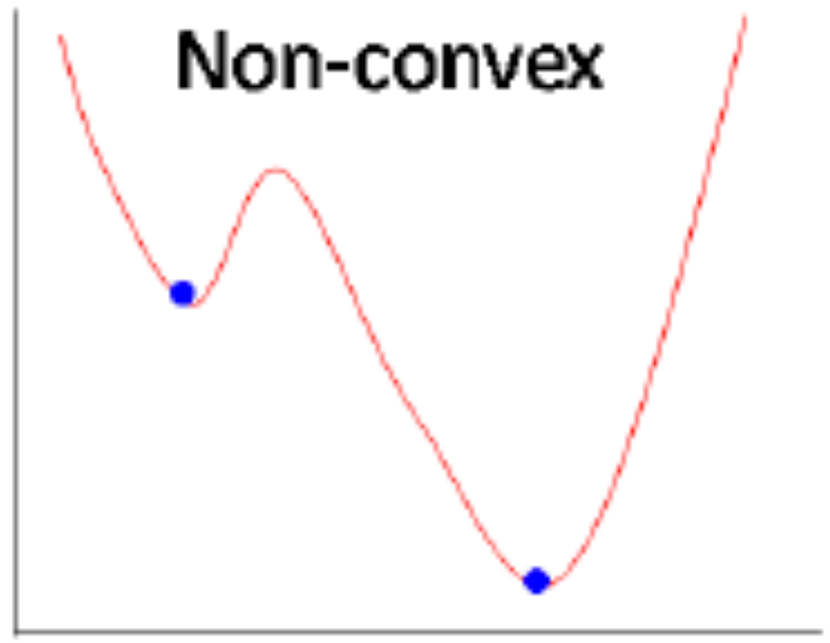
Interpretation of Hypothesis

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

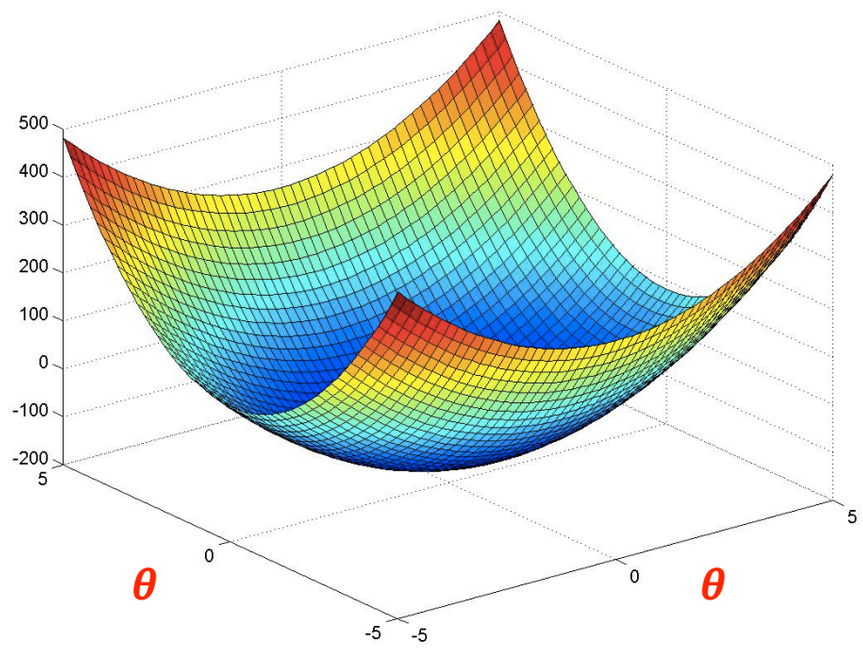
$$h_\theta(x) = P(y = 1 \mid x; \theta)$$

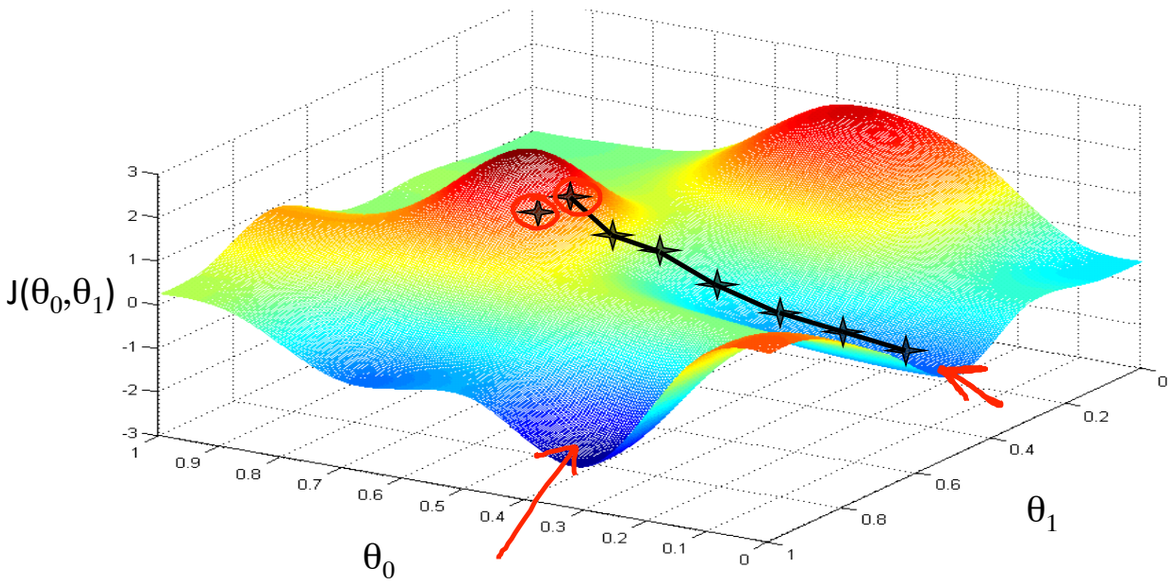
probability that *y = 1*, given x, parameterized by *θ*

we want convex! easy gradient descent!



$J(\theta_1$





Logistic Regression:

Interpretation of Hypothesis

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

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

$$h_\theta(x)=P(y=1 \mid x;\theta)$$

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

Logistic Regression:

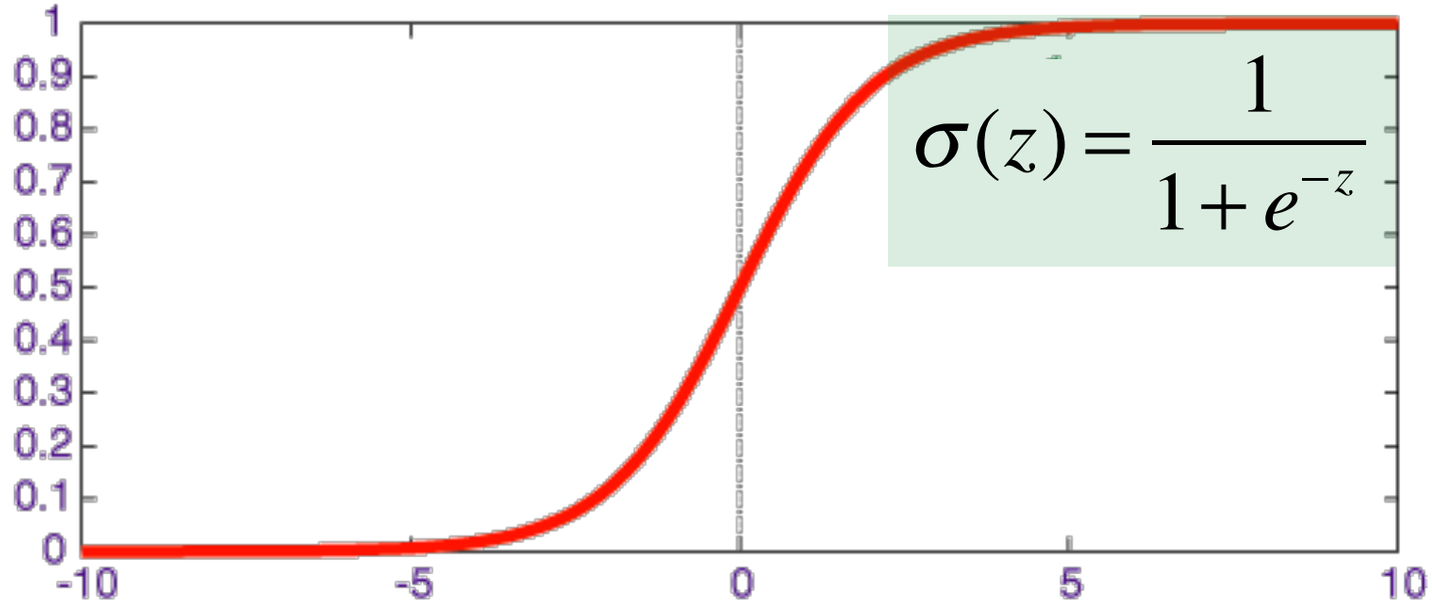
Decision Boundary

- Logistic Regression:

$$h_\theta(x) = \sigma\left(\theta^T x\right)$$

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

sigmoid (logistic) function



Logistic Regression:

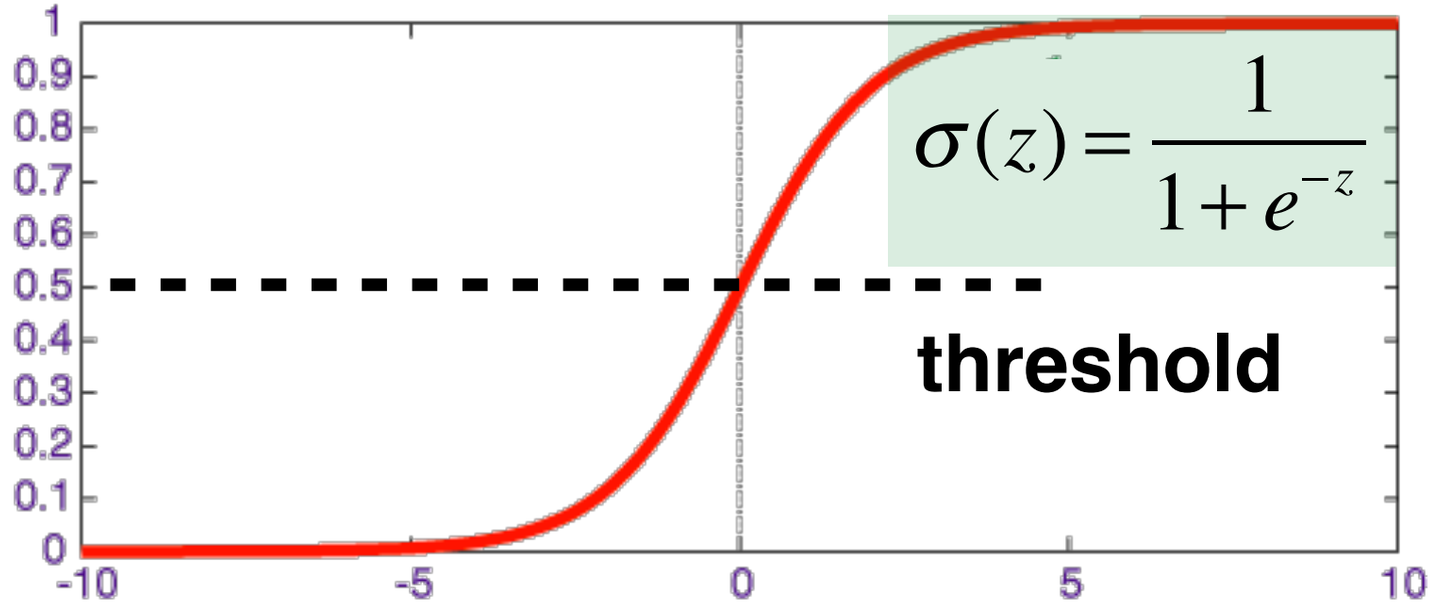
Decision Boundary

- Logistic Regression:

$$h_\theta(x) = \sigma\left(\theta^T x\right)$$

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

sigmoid (logistic) function



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

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

Logistic Regression:

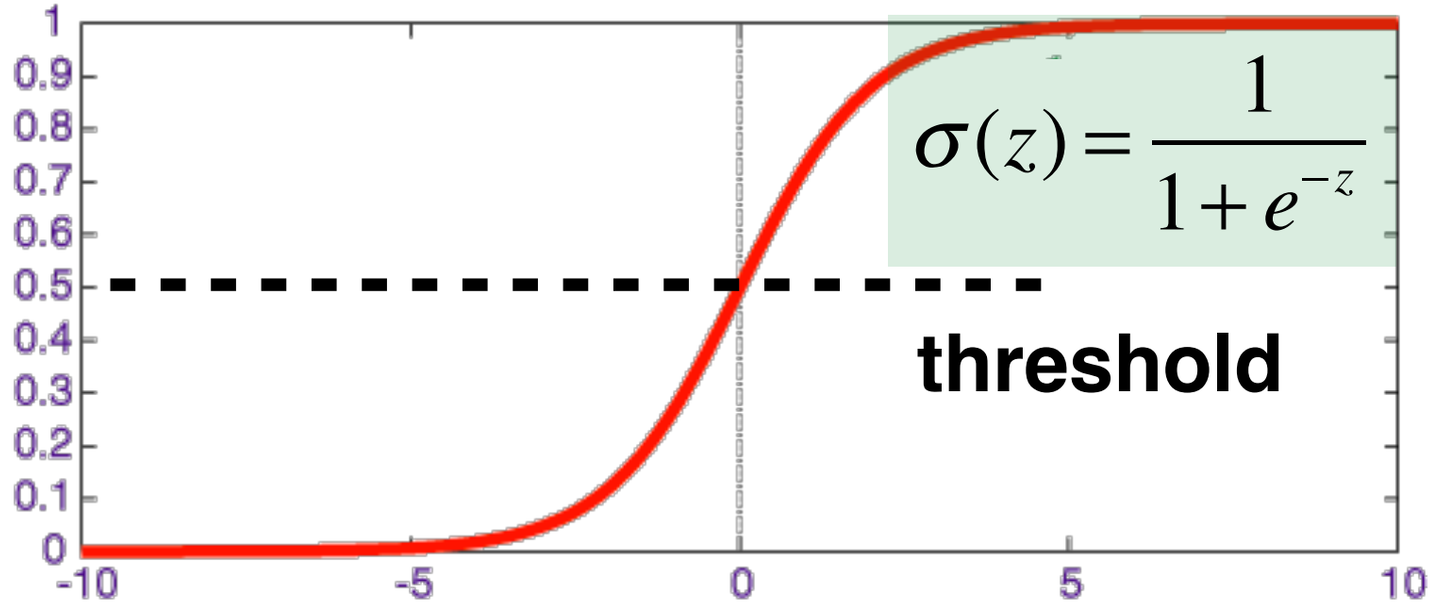
Decision Boundary

- Logistic Regression:

sigmoid (logistic) function

$$h_\theta(x) = \sigma\left(\theta^T x\right)$$

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

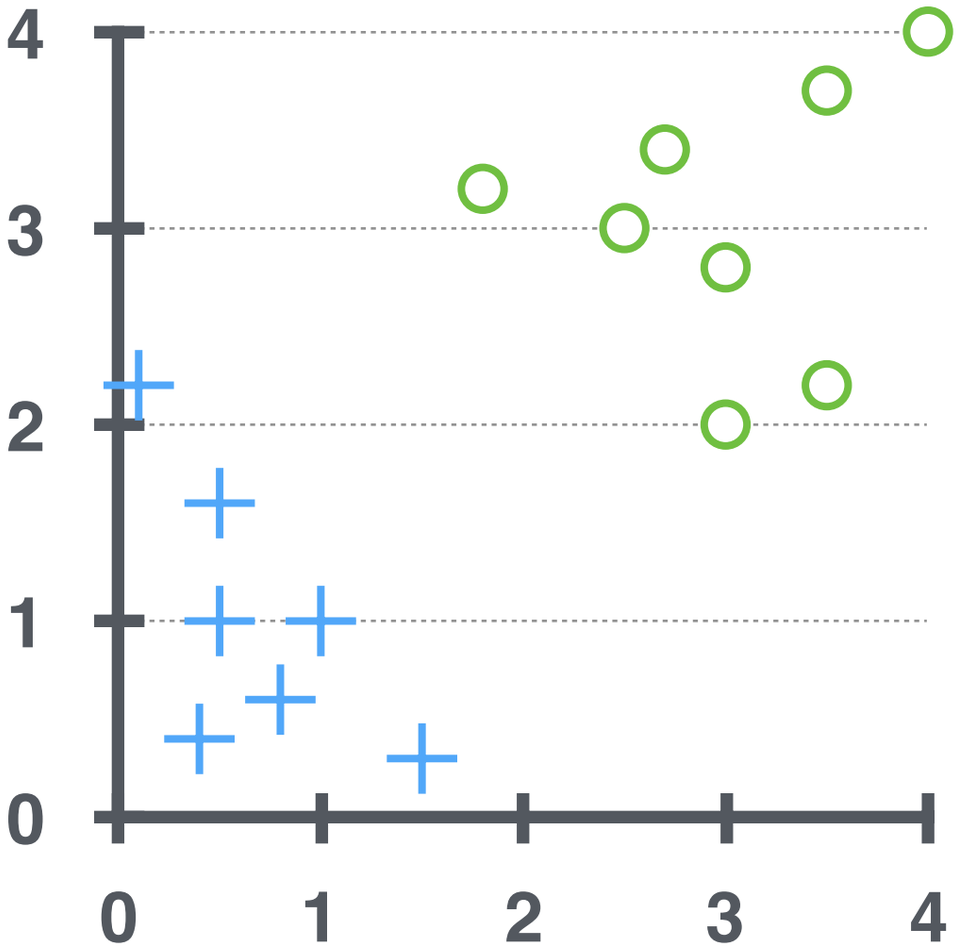


predict ***y = 1*** if ***$h_\theta(x) \geq 0.5$*** ⟵ when ***$\theta^T x \geq 0$***

predict ***y = 0*** if ***$h_\theta(x) < 0.5$*** ⟵ when ***$\theta^T x < 0$***

Logistic Regression:

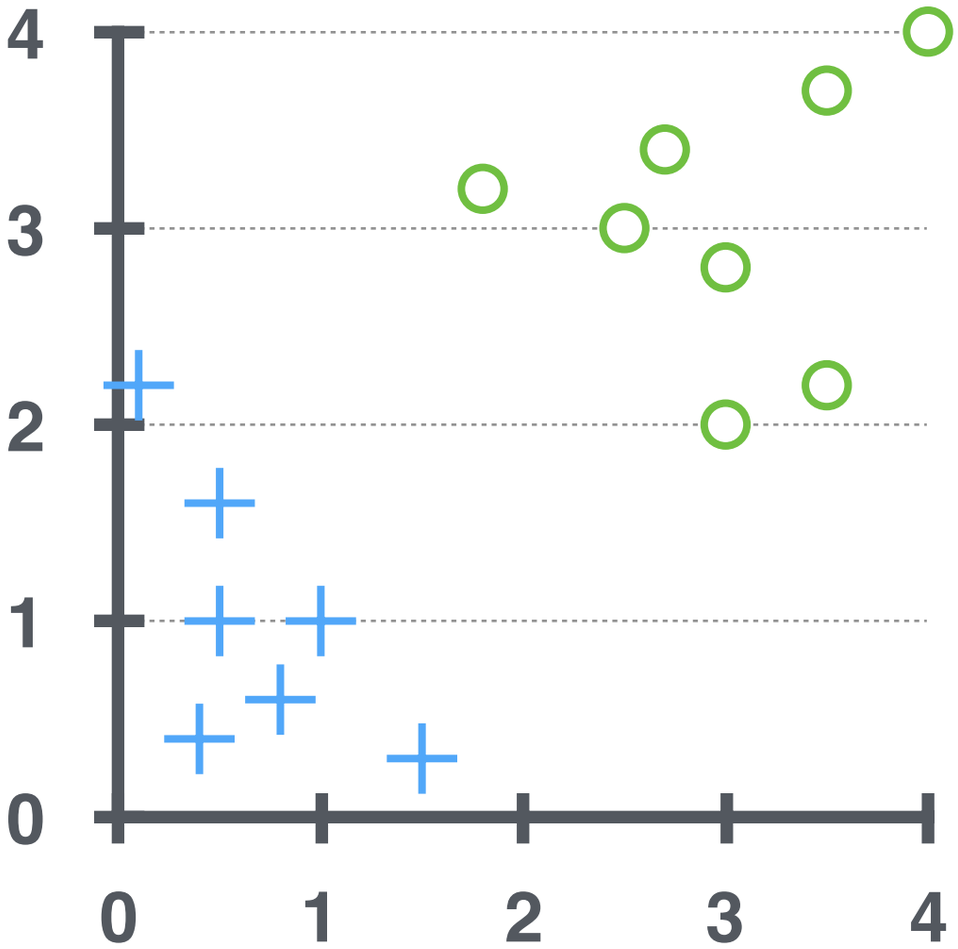
Decision Boundary



Logistic Regression:

Decision Boundary

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



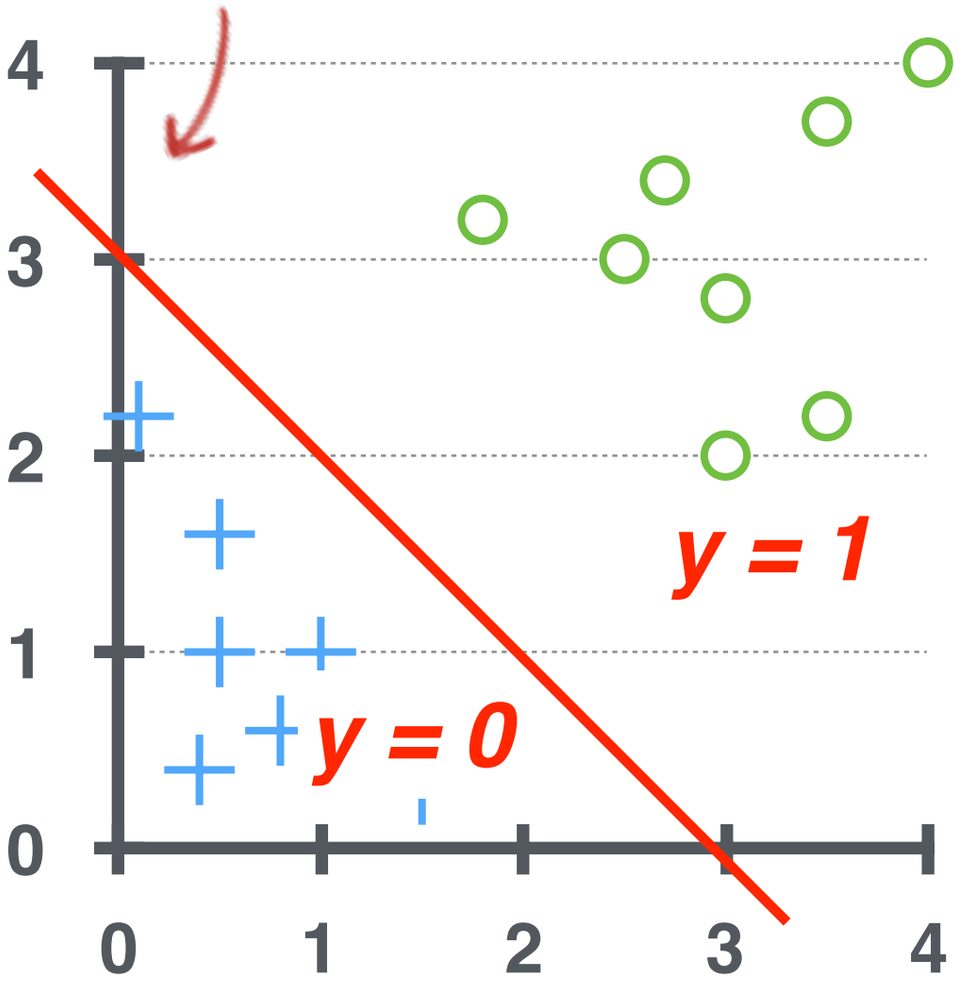
What if $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$?

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

Logistic Regression:

Decision Boundary

decision boundary



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

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

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

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

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

Cost function:

Linear → Logistic Regression

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

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

$$\| \quad \text{Cost}\left(h_\theta(x), y\right)$$

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

this cost function is non-convex for logistic regression

Logistic Regression:

Cost Function

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

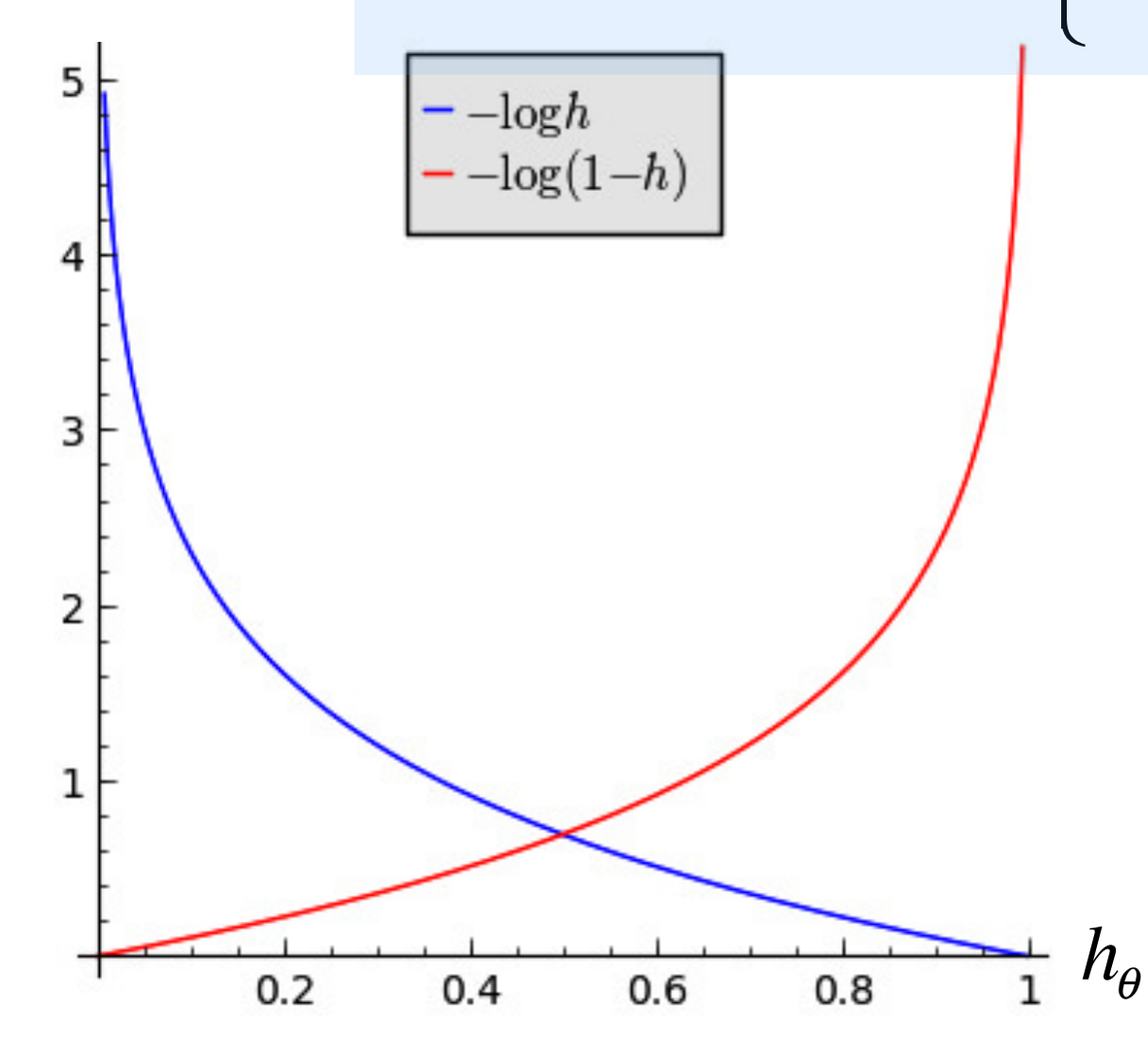
remember that

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

Logistic Regression:

Cost Function

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



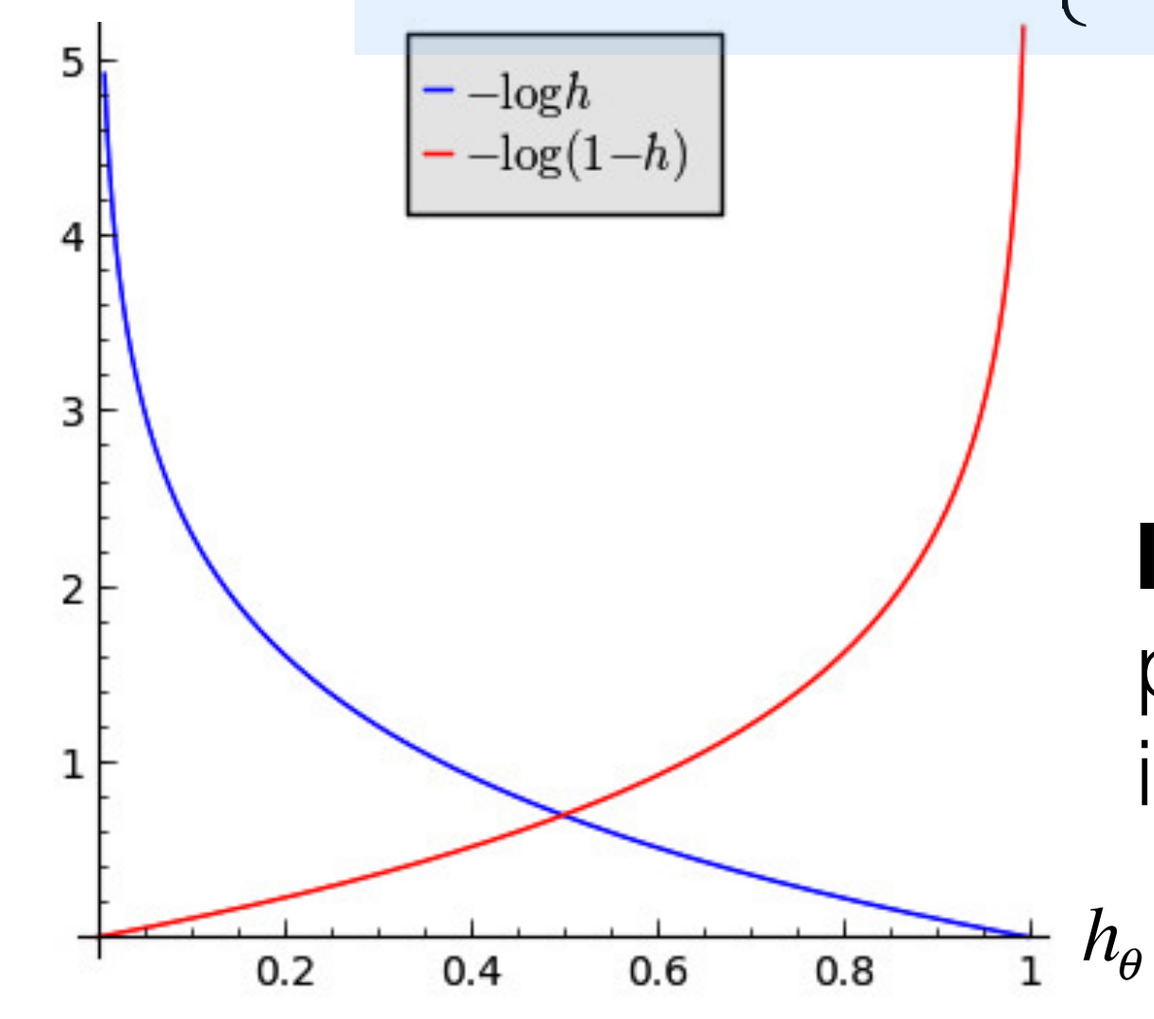
remember that

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

Logistic Regression:

Cost Function

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



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

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

Intuition:

penalize learning algorithm if ***$h_\theta(x) = 0$*** (predict **$P(y=1|x;\theta) = 0$**), but ***y = 1***

Logistic Regression:

Cost Function

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

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

remember that $y = 0$ or 1 always

Logistic Regression:

Cost Function

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

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

remember that $y = 0$ or 1 always

the same

cross entropy loss:

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

Learn more: https://en.wikipedia.org/wiki/Cross_entropy

Logistic Regression

- **Cost Function:**

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

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

- **Goal:**

$$\text{learn parameters } \theta \text{ to } \underset{\theta}{\text{minimize}}\, J(\theta)$$

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

P(y=1|x;θ)

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

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

Logistic Regression:

Gradient Descent

repeat until convergence {

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

}

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 {

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

}

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 {

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

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

}

simultaneous update for all θ_j

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

using different hypothesis from linear regression

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 = the, c_1 = DT)***
  - a training set consists of ***m*** sequences of labeled examples ***$(x_1, y_1), \ldots, (x_m, y_m)$***

    x_1 = <the back door> and y_1= <DT JJ NN>

- Output:
  - a learned classifier to predict label sequences ***γ: x → y***

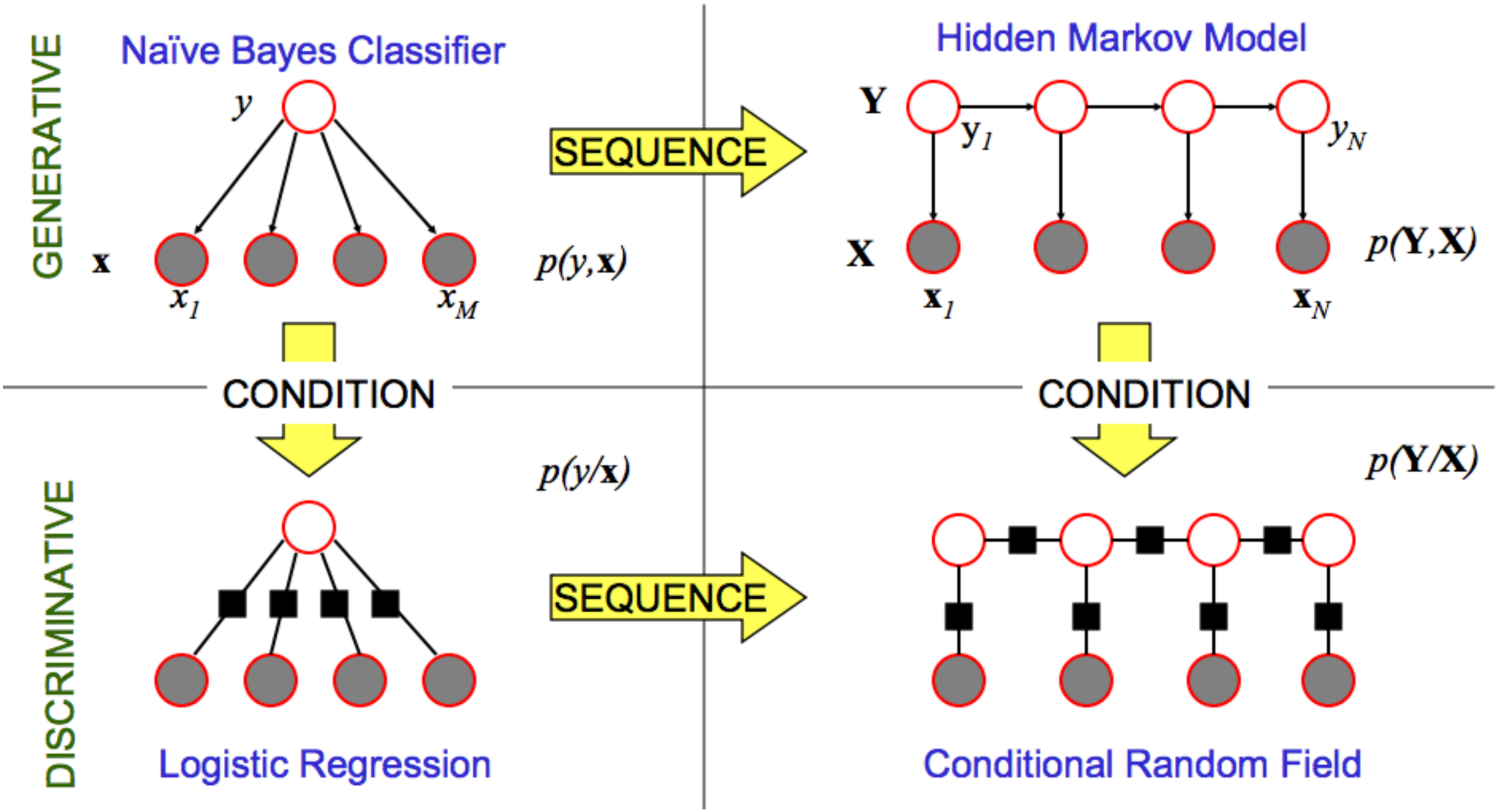
Features for Sequential Tagging

- 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

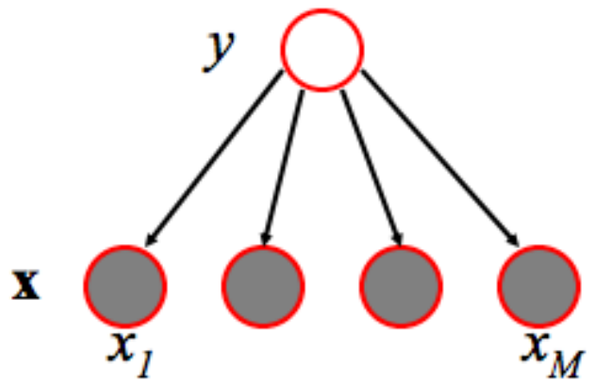


Source: Andrew McCallum, Sargur Srihari

Probabilistic Graphical Models

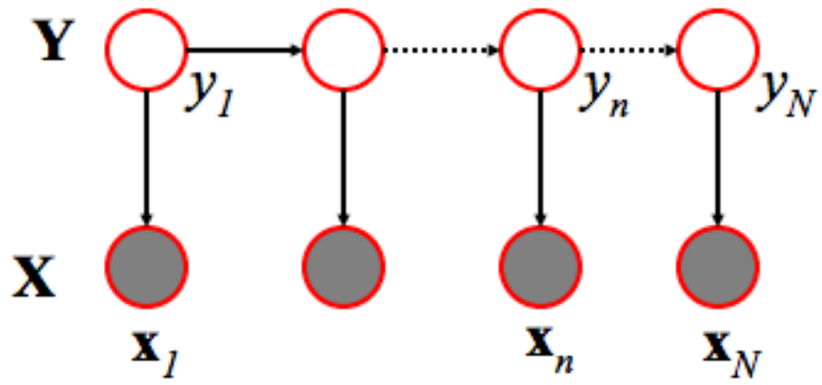
GENERATIVE

Naïve Bayes Classifier



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

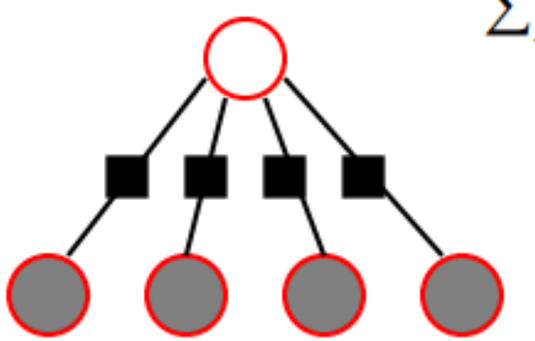
Hidden Markov Model



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

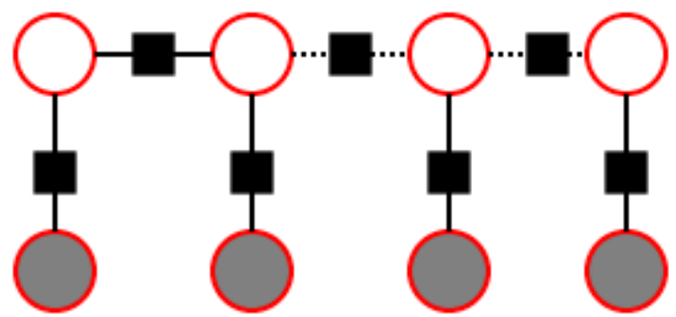
DISCRIMINATIVE

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



Logistic Regression

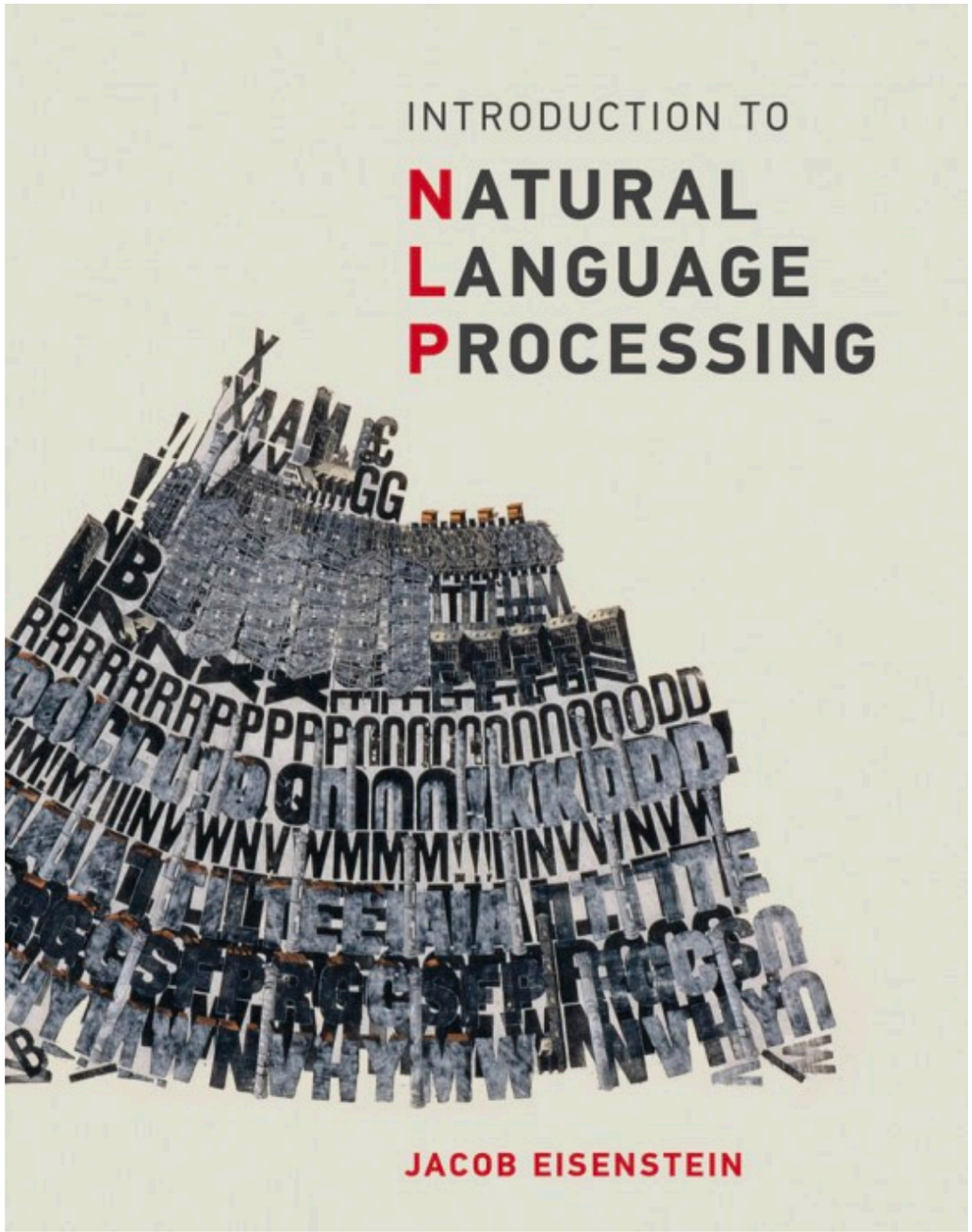
$$p(\mathbf{Y} \mid \mathbf{X}) = \frac{\exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y_n, y_{n-1}, \mathbf{x}_n)\right\}}{\sum_{y'} \exp\left\{\sum_{m=1}^{M} \lambda_m f_m(y_n', y_{n-1}', \mathbf{x}_n)\right\}}$$



Conditional Random Field

Source: Andrew McCallum, Sargur Srihari

New Textbook



October 2019
MIT Press

https://mitpress.mit.edu/books/introduction-natural-language-processing

Follow @cocoweixu

Instructor: Wei Xu

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

Course Website: socialmedia-class.org