

Social Media & Text Analysis

lecture 5 - Paraphrase Identification
and Logistic Regression



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

Homework #1

Percentage of tweets LangID tagged by Twitter: 0.8388

Total number of unique language tags provided by Twitter: 39 different languages

Total number of unique language tags provided by langid.py: 89 different languages

Percentage of langid.py and Twitter's API agreed: 0.794825941822

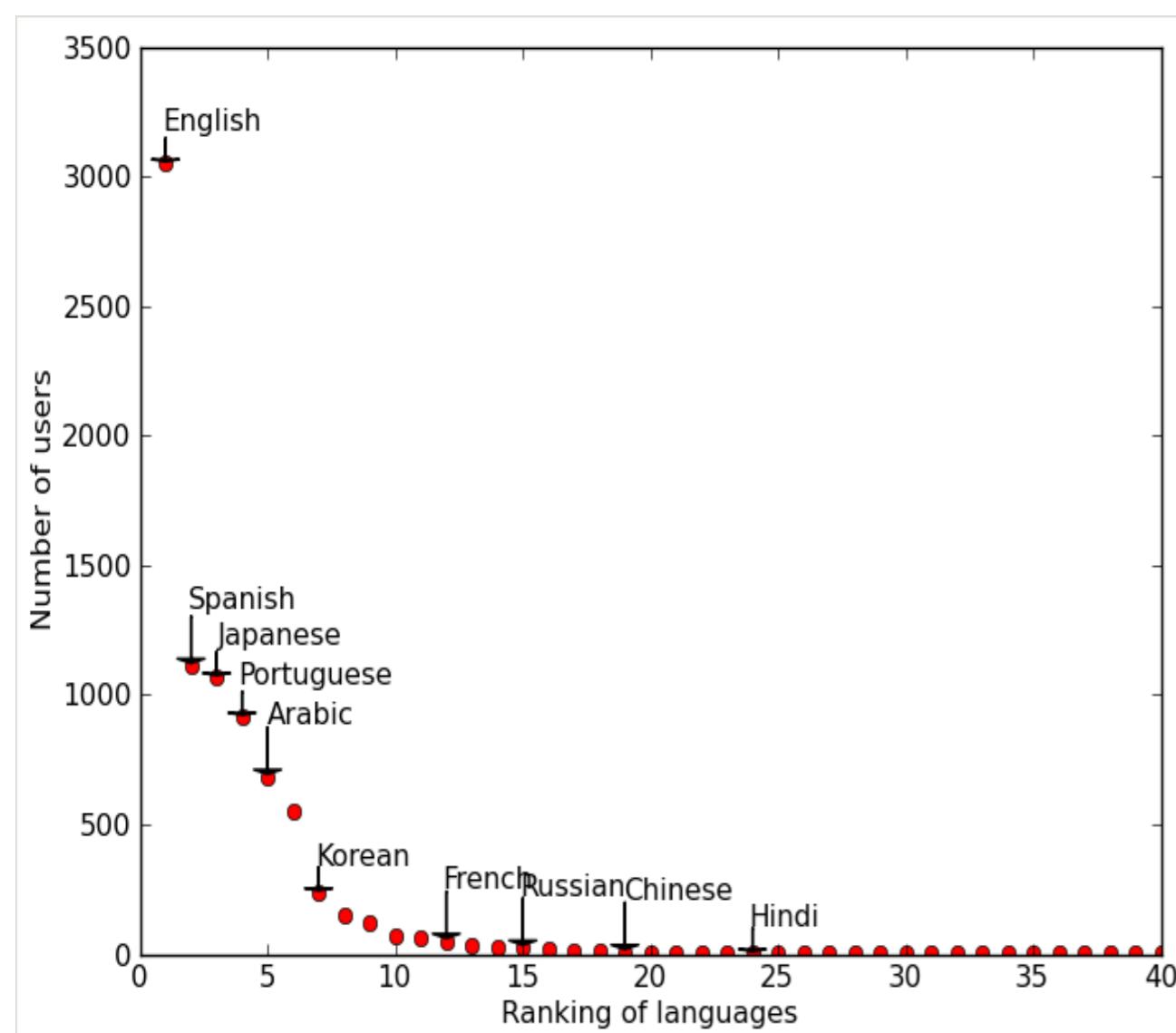
Homework #1

3-3. what kind of tweets/languages do they disagree?

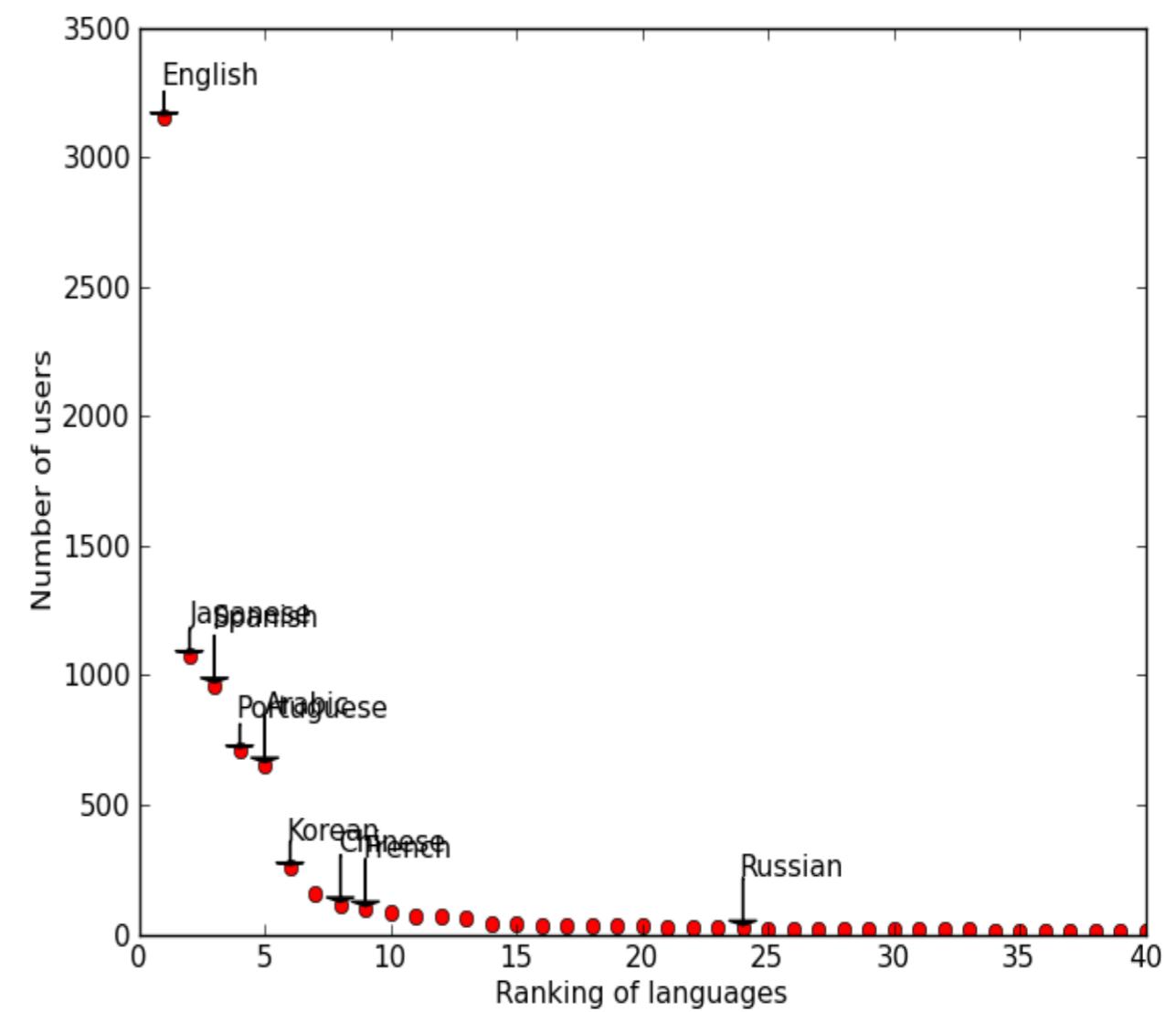
Solution:

By comparing language id from langid.py and Twitter's API, there are 1721 language mismatched tweets among 8,388 LangID tagged tweets. The main reason of disagreement is that they use different language identifier codes, BCP 47 and ISO 639-1. If we can map two different language codes into a common language code, the percentage of agreement will be increased. Furthermore, the results show that a large portion of undefined languages from Twitter's API can be identified by langid.py. To conclude, langid.py has a more strong capability to capture the type of language used in tweets. Types of 1721 disagreed tweets are stored in **disagree_lang_list** list variable in the code. The below table illustrates the example of disagreed tweets.

Homework #1

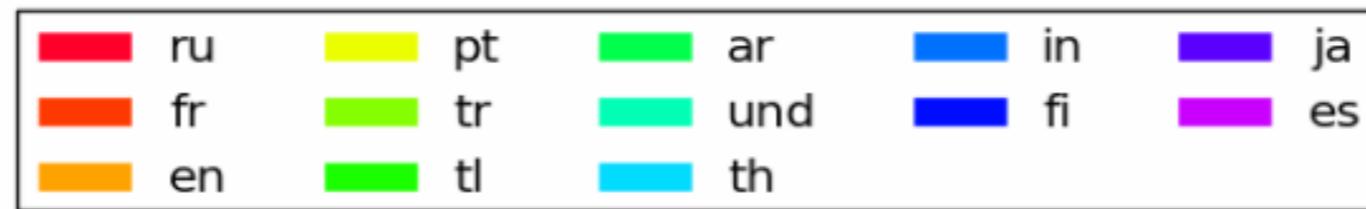
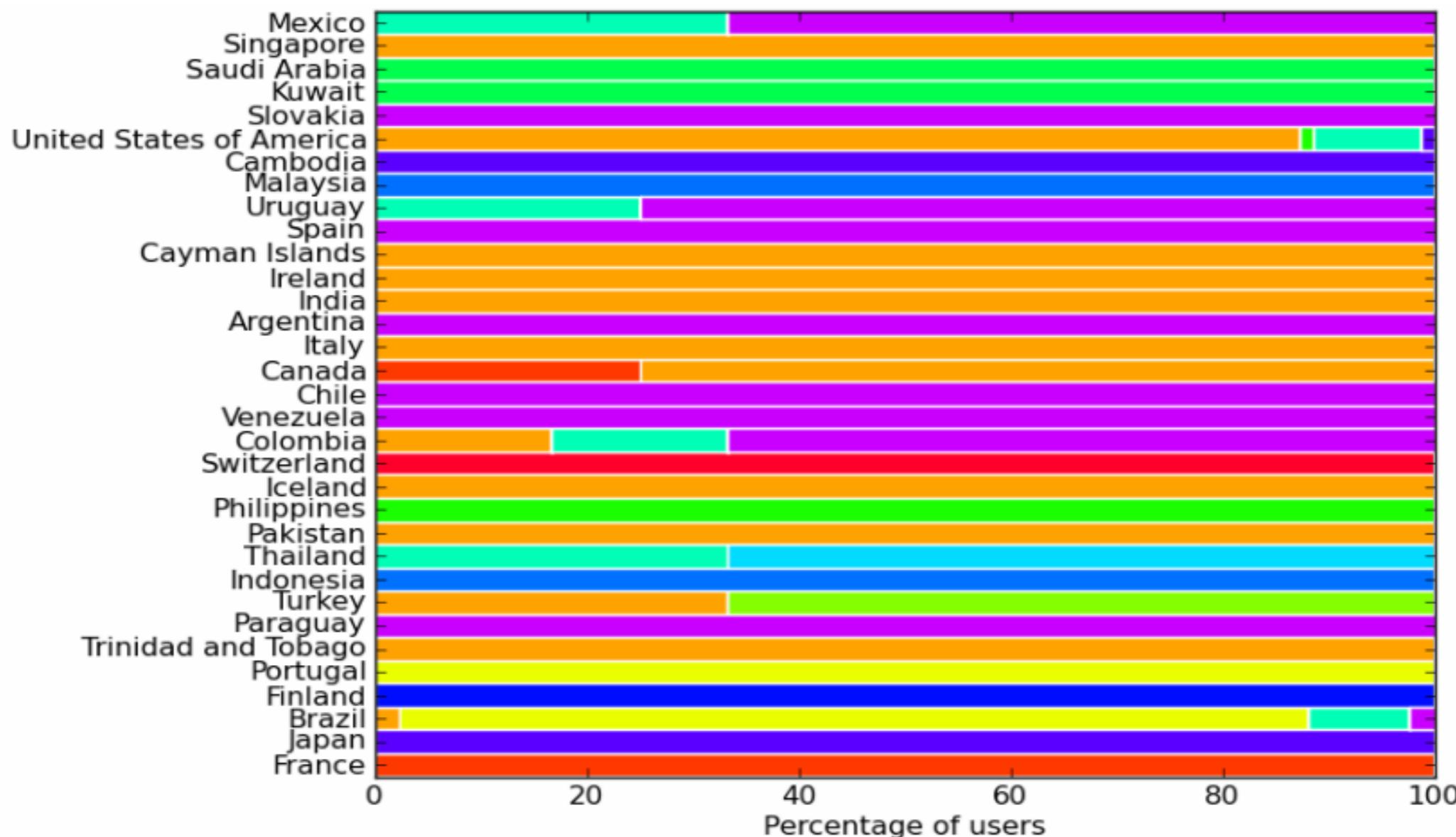


Twitter API



langid.py

Homework #1



what is Paraphrase?

“sentences or phrases that convey approximately the same meaning using different words” — (Bhagat & Hovy, 2012)

wealthy

word

rich

the king's speech

phrase

His Majesty's address

*... the forced resignation
of the CEO of Boeing,
Harry Stonecipher, for ...*

sentence

*... after Boeing Co. Chief
Executive Harry Stonecipher
was ousted from ...*

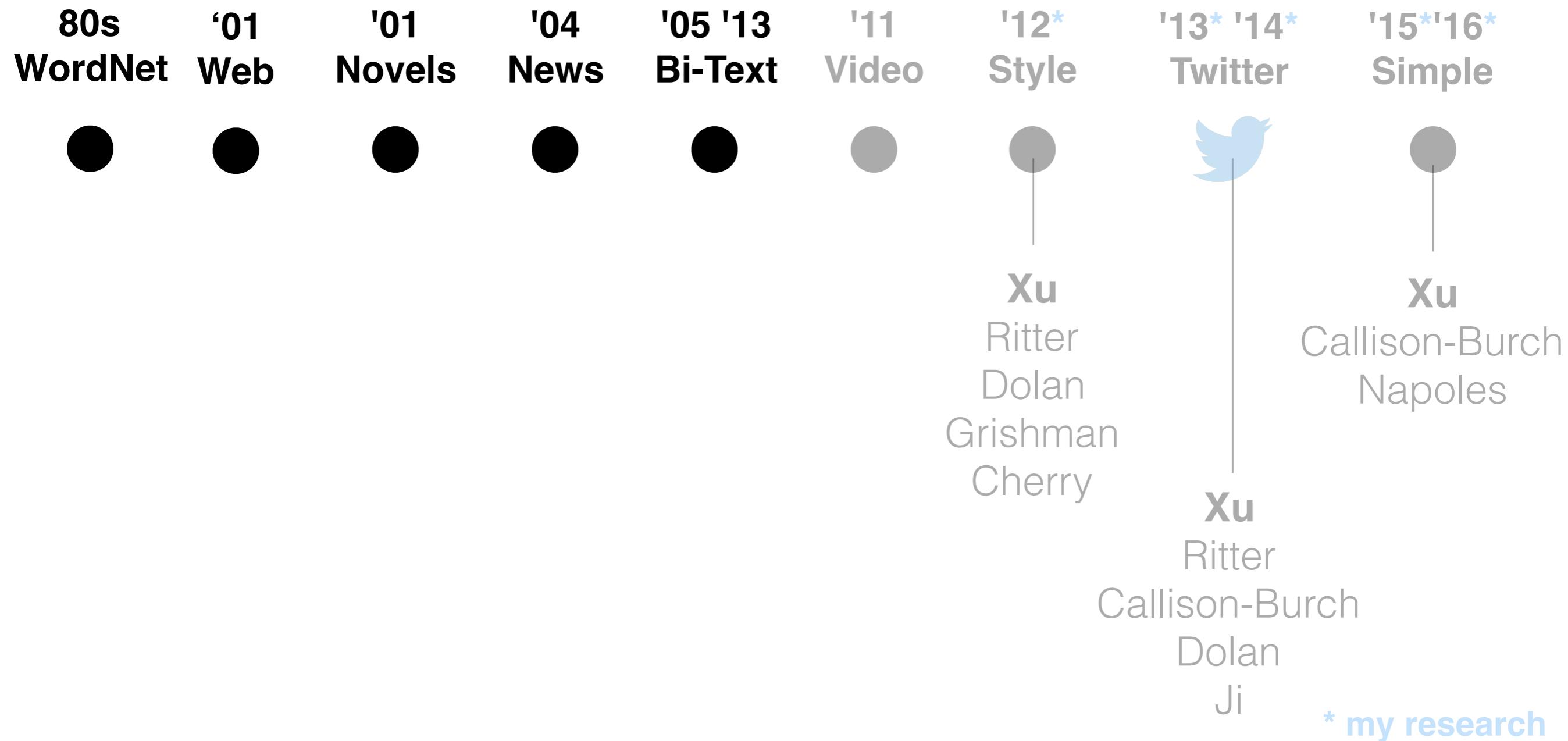
The Ideal



Translation: "You have a bruised rib."

(Review Last Week)

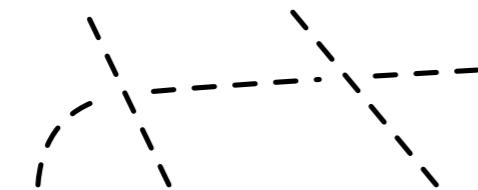
Paraphrase Research



Bilingual Pivoting

word alignment

... 5 farmers were thrown into jail in Ireland ...



... fünf Landwirte

thrown into jail

festgenommen

, weil ...

Distributional Similarity

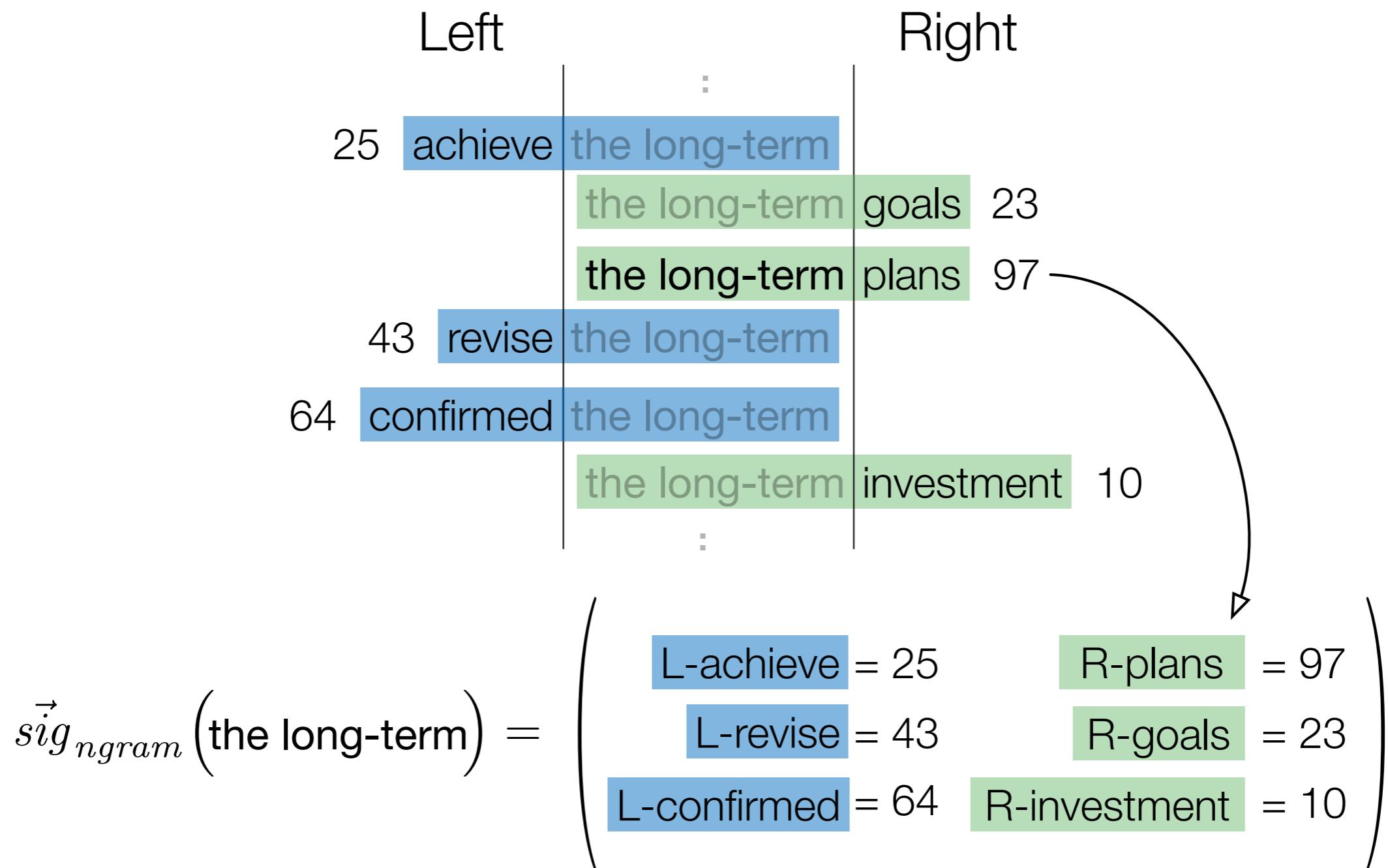
Idea: similar words occur in similar contexts.

Characterize words by their contexts

Contexts represented by co-occurrence vectors, similarity quantified by **cosine**

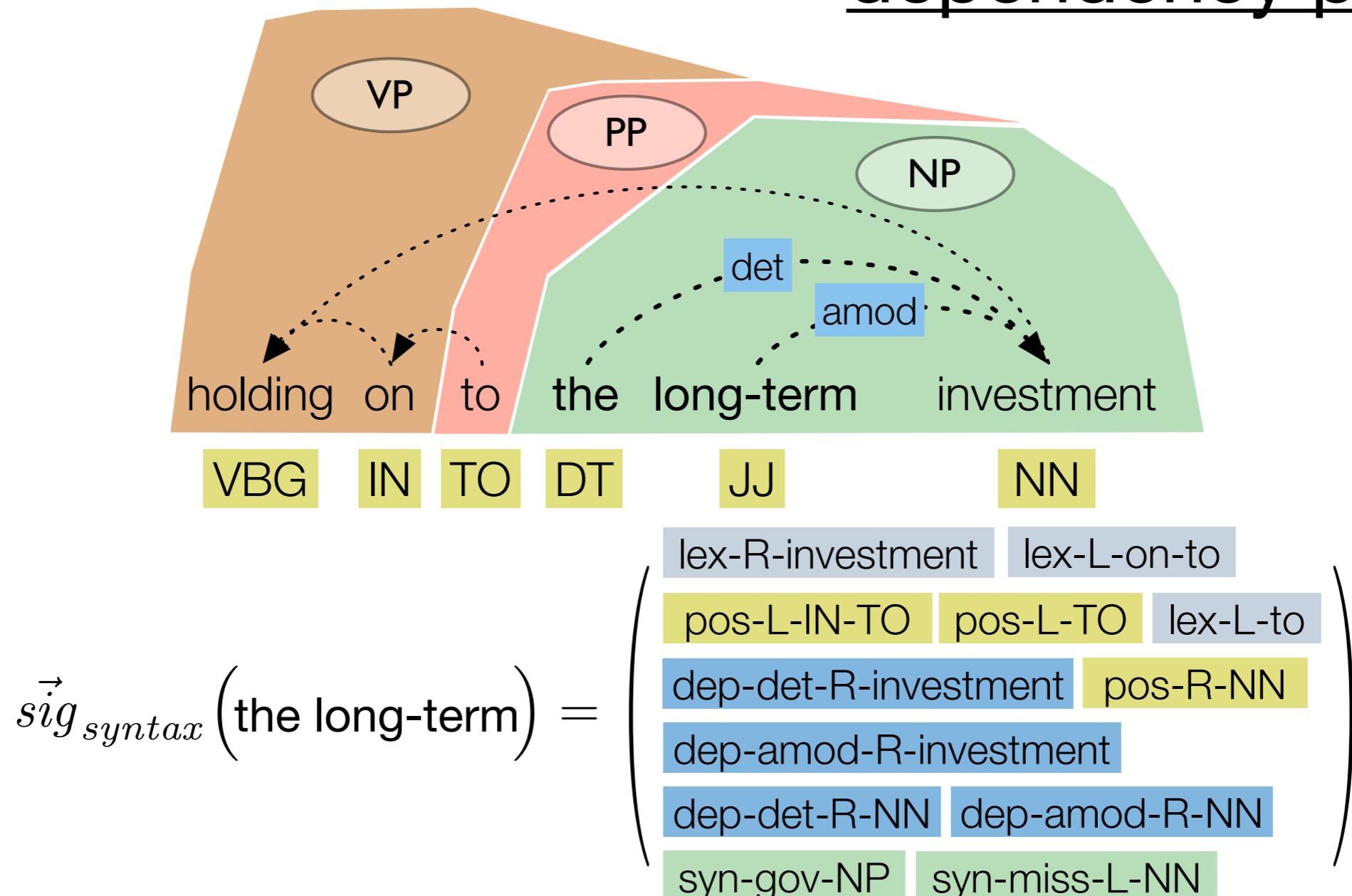
“Are these paraphrases substitutable?”

n -gram Context



Syntactic Context

dependency parsing



Quiz #3

Key Limitations?

Quiz #3

word sense

bug

microbe, virus,
bacterium,
germ, parasite

insect, beetle,
pest, mosquito,
fly

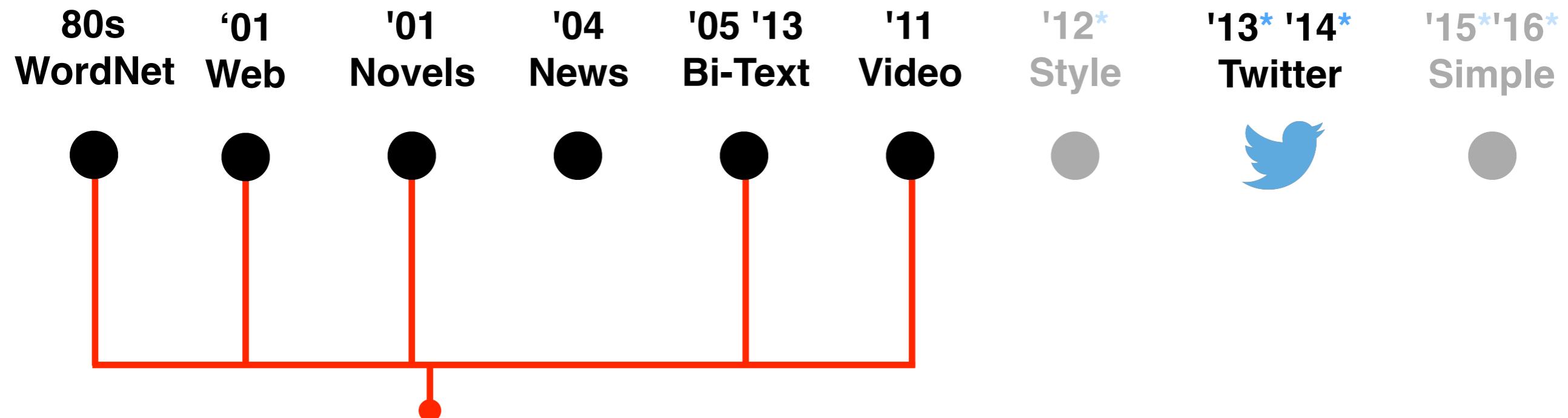
bother, annoy,
pester

microphone,
tracker, mic,
wire, earpiece,
cookie

glitch, error,
malfunction,
fault, failure

squealer, snitch,
rat, mole

One Key Limitation



only paraphrases, no non-paraphrases

* my research

Paraphrase Identification

obtain sentential paraphrases automatically

Mancini has been sacked by Manchester City

Yes!

Mancini gets the boot from Man City

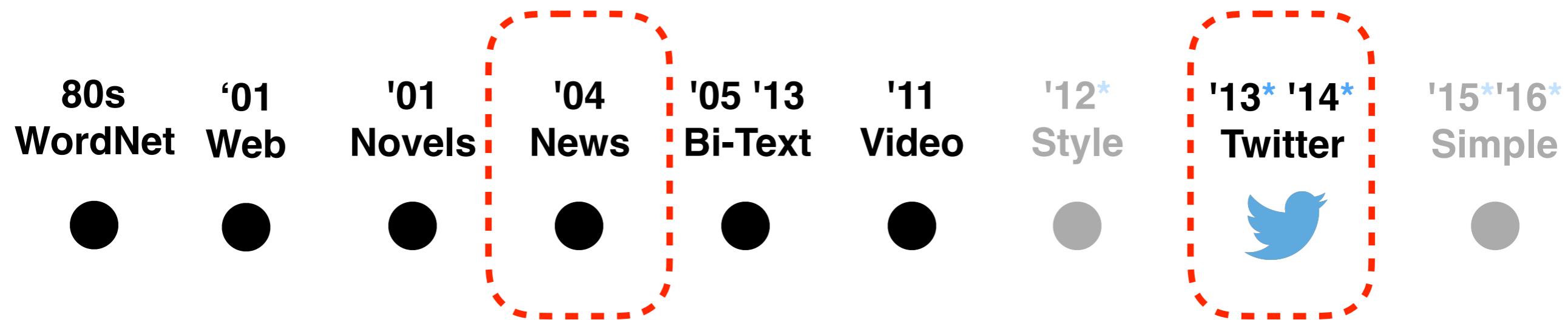
WORLD OF JENKS IS ON AT 11

No!

World of Jenks is my favorite show on tv

(meaningful) non-paraphrases are needed to train classifiers!

Also Non-Paraphrases



(meaningful) non-paraphrases are needed to train classifiers!

* my research

News Paraphrase Corpus



Microsoft Research Paraphrase Corpus
also contains some non-paraphrases

Twitter Paraphrase Corpus



Rep. Stacey Newman @staceynewman · 5h

So sad to hear today of former WH Press Sec **James Brady's passing**.
@bradybuzz & family will carry on his legacy of #gunsense.



Jim Sciutto @jimsciutto · 4h

Breaking: Fmr. WH Press Sec. **James Brady** has died at 73, crusader for gun control after wounded in '81 Reagan assassination attempt



NBC News @NBCNews · 2h

James Brady, President Reagan's press secretary shot in 1981 assassination attempt, dead at 73 nbcnews.to/WX1Btq pic.twitter.com/1ZtuEakRd9



also contains a lot of non-paraphrases

Paraphrase Identification:

A Binary Classification Problem

- Input:
 - a sentence pair \mathbf{x}
 - a fixed set of binary classes $Y = \{0, 1\}$
 - Output:
 - a predicted class $y \in Y$ ($y = 0$ or $y = 1$)
- negative (non-paraphrases)**
- positive (paraphrases)**
-
- ```
graph TD; Input[Input] --> X[x]; Input --> Y["Y = {0, 1}"]; Y --> Neg[negative
(non-paraphrases)]; Y --> Pos[positive
(paraphrases)];
```

Paraphrase Identification:

# A Binary Classification Problem

- Input:
  - a sentence pair  $\mathbf{x}$
  - a fixed set of binary classes  $\mathbf{Y} = \{0, 1\}$
- Output:
  - a predicted class  $y \in \mathbf{Y}$  ( $y = 0$  or  $y = 1$ )

Classification Method:

# Supervised Machine Learning

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

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 Week #3) Classification Method: Supervised Machine Learning

- **Naïve Bayes**
- Logistic Regression
- Support Vector Machines (SVM)
- ...

(Recap Week#3)

# Naïve Bayes

- ***Cons:***

features  $t_i$  are assumed independent given the class  $y$

$$P(t_1, t_2, \dots, t_n | y) = P(t_1 | y) \cdot P(t_2 | y) \cdot \dots \cdot P(t_n | y)$$

- ***This will cause problems:***

- correlated features → double-counted evidence
- while parameters are estimated independently
- hurt classifier's accuracy

# Classification Method: Supervised Machine Learning

- Naïve Bayes
- **Logistic Regression**
- Support Vector Machines (SVM)
- ...

# 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).

# Before Logistic Regression

**Let's start with  
something simpler!**

# Paraphrase Identification: Simplified Features

- We use only one feature:
  - number of words that two sentence shared in common

A very related problem of Paraphrase Identification:  
**Semantic Textual Similarity**

- How similar (close in meaning) two sentences are?

5: completely equivalent in meaning

4: mostly equivalent, but some unimportant details differ

3: roughly equivalent, some important information differs/missing

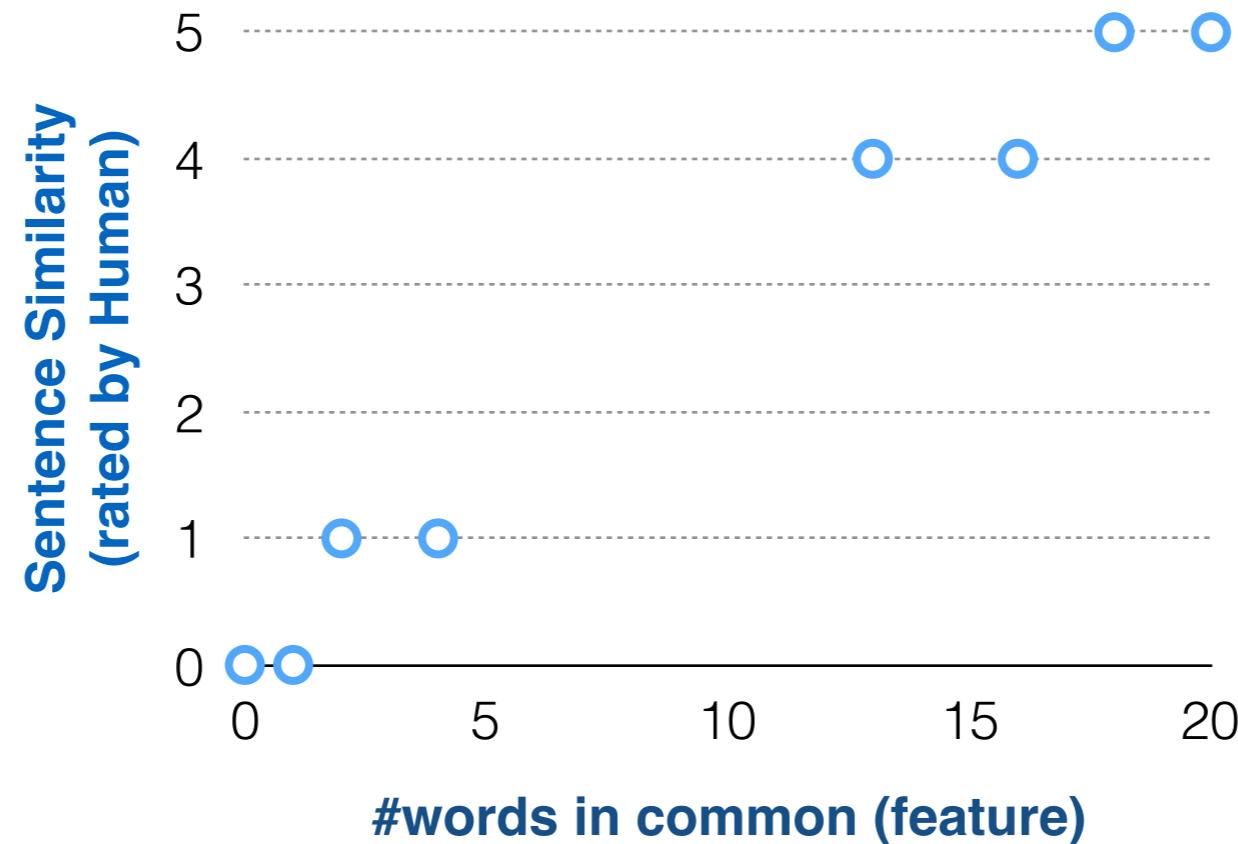
2: not equivalent, but share some details

1: not equivalent, but are on the same topic

0: completely dissimilar

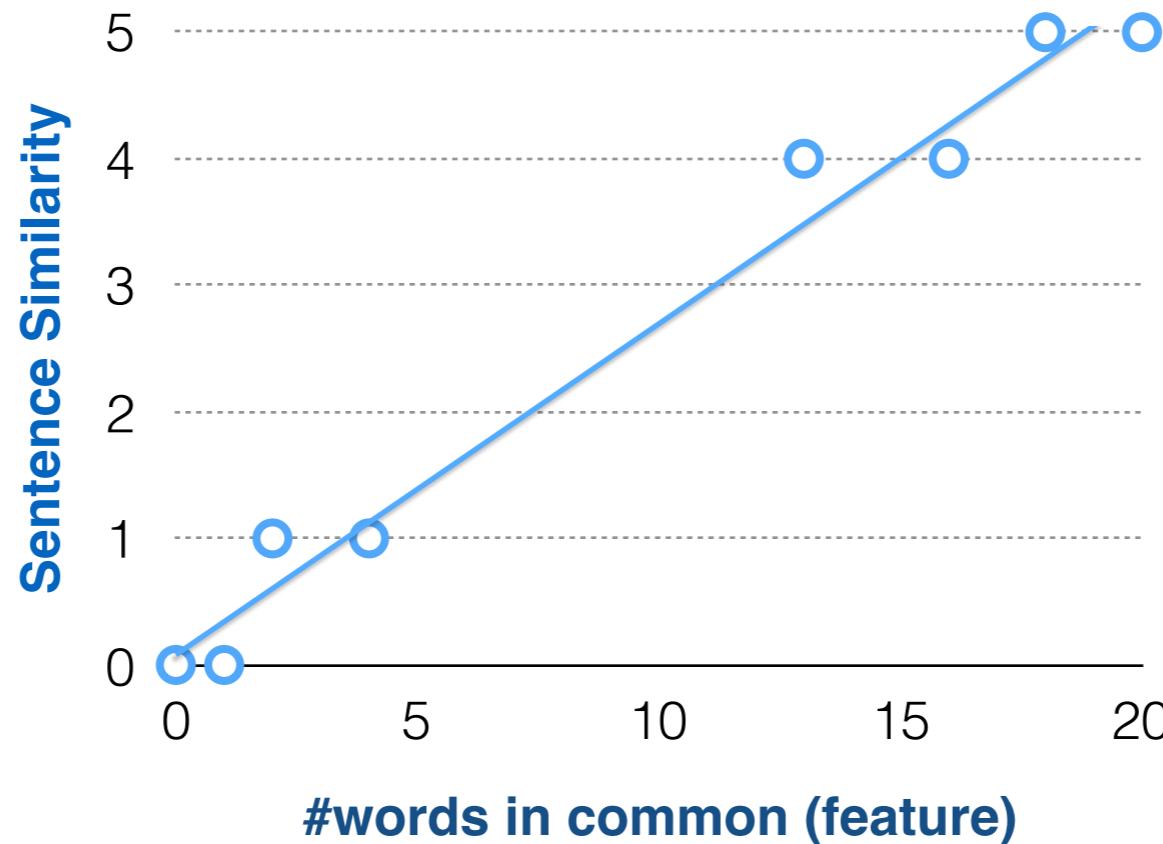
A Simpler Model:

# Linear Regression



A Simpler Model:

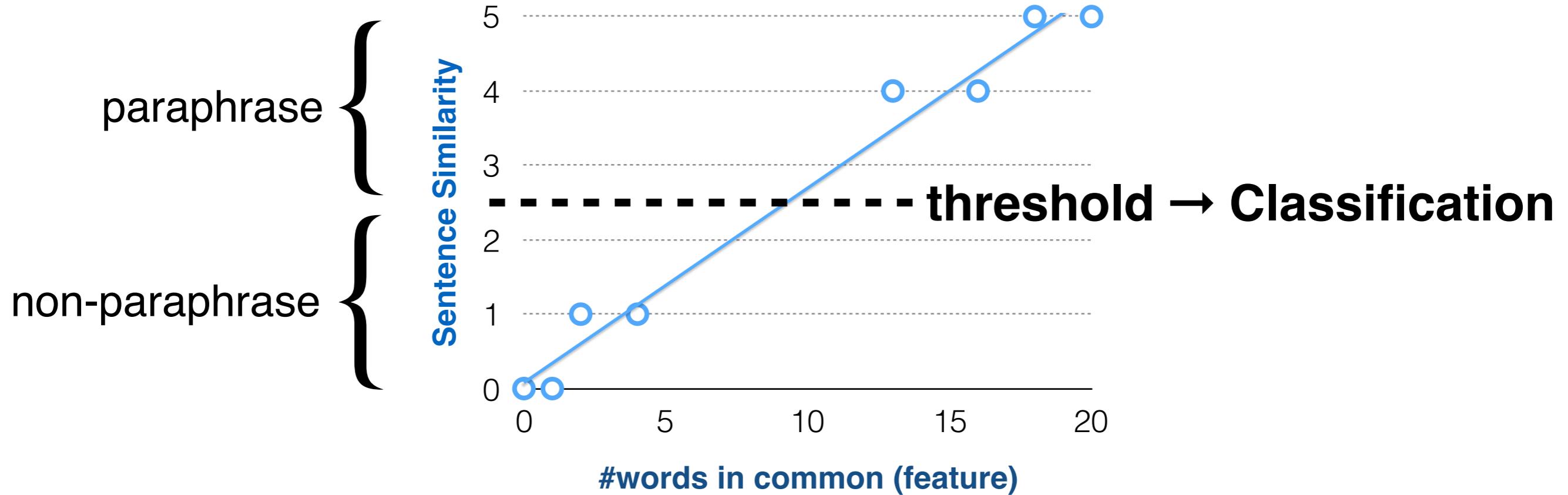
# Linear Regression



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

A Simpler Model:

# Linear Regression



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

# Training Set

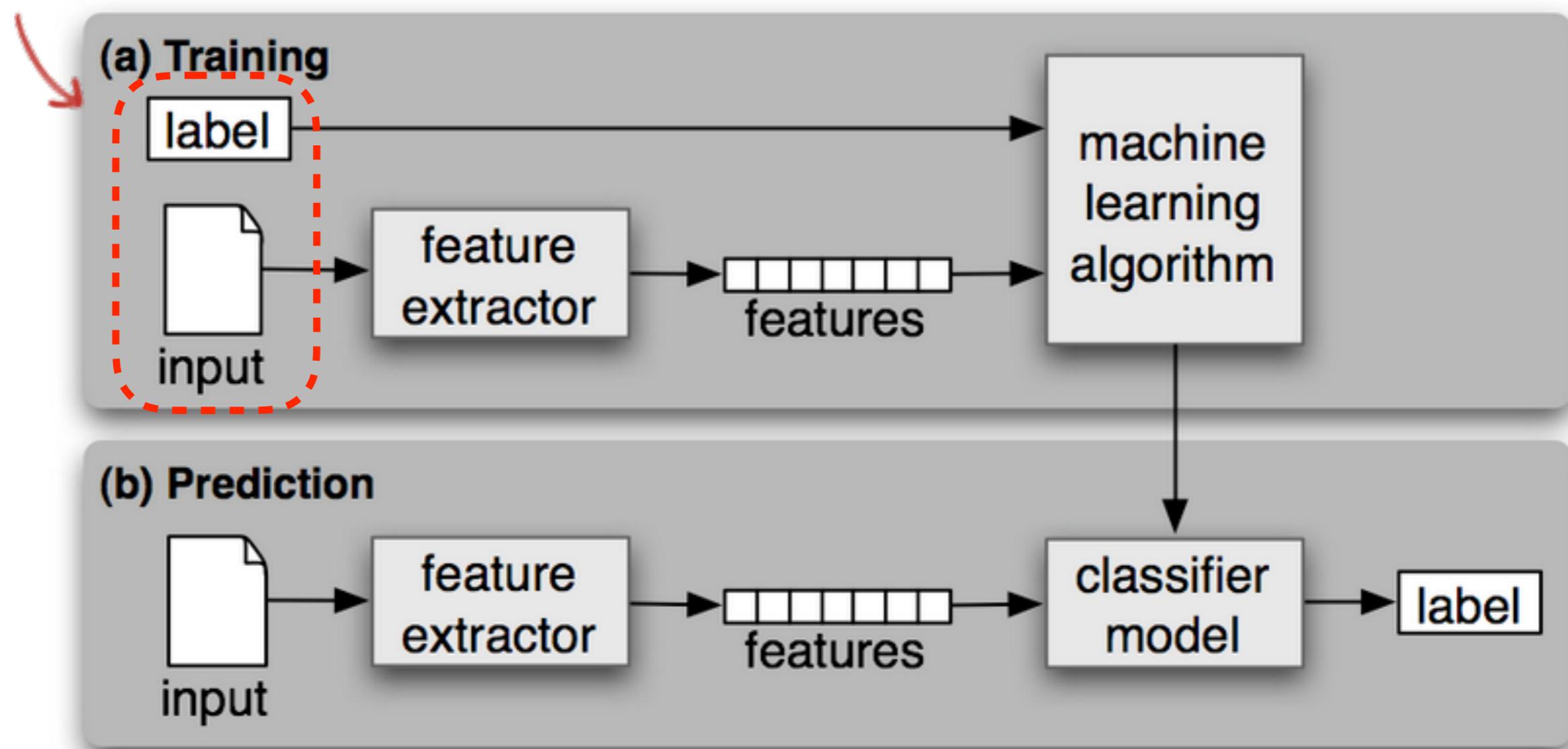
| #words in common<br>( $x$ ) | Sentence Similarity<br>( $y$ ) |
|-----------------------------|--------------------------------|
| 1                           | 0                              |
| 4                           | 1                              |
| 13                          | 4                              |
| 18                          | 5                              |
| ...                         | ...                            |

- $m$  hand-labeled sentence pairs  $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- $x$ 's: “input” variable / features
- $y$ 's: “output”/“target” variable

(Recap Week#3)

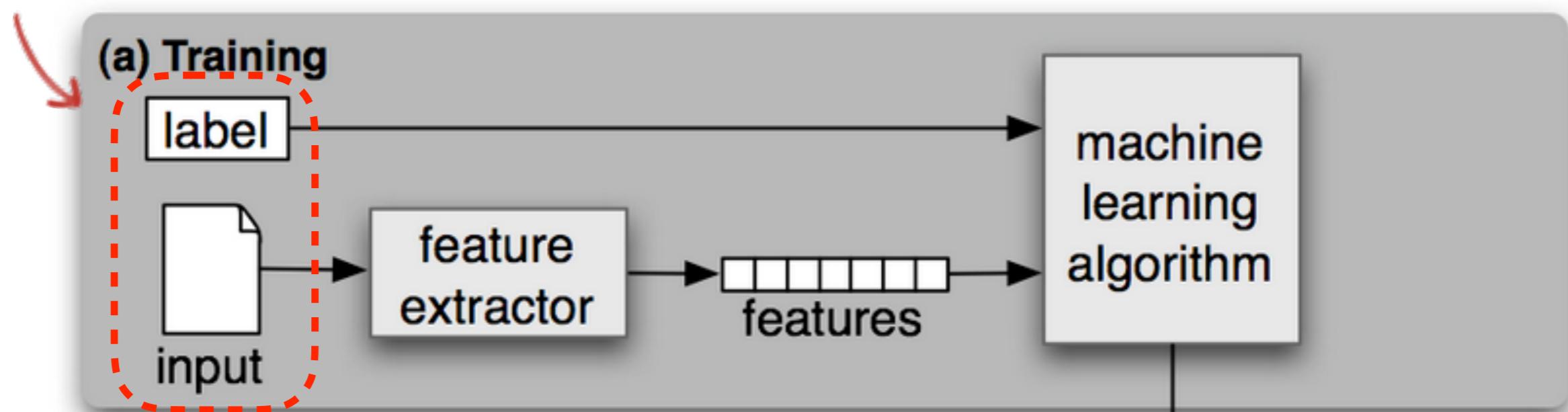
# Supervised Machine Learning

## training set

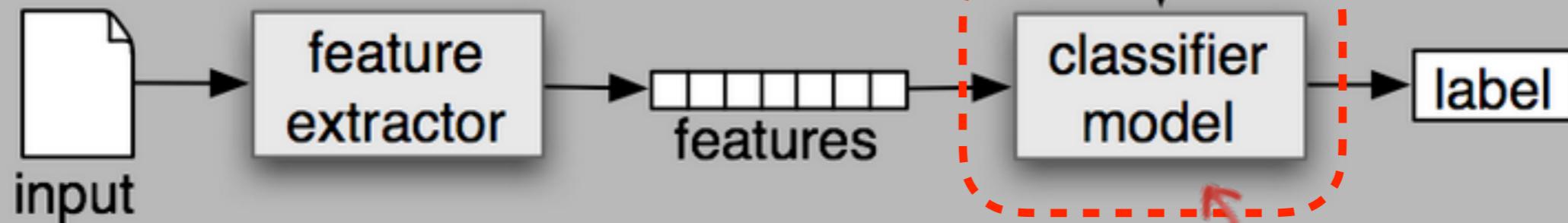


# Supervised Machine Learning

**training set**



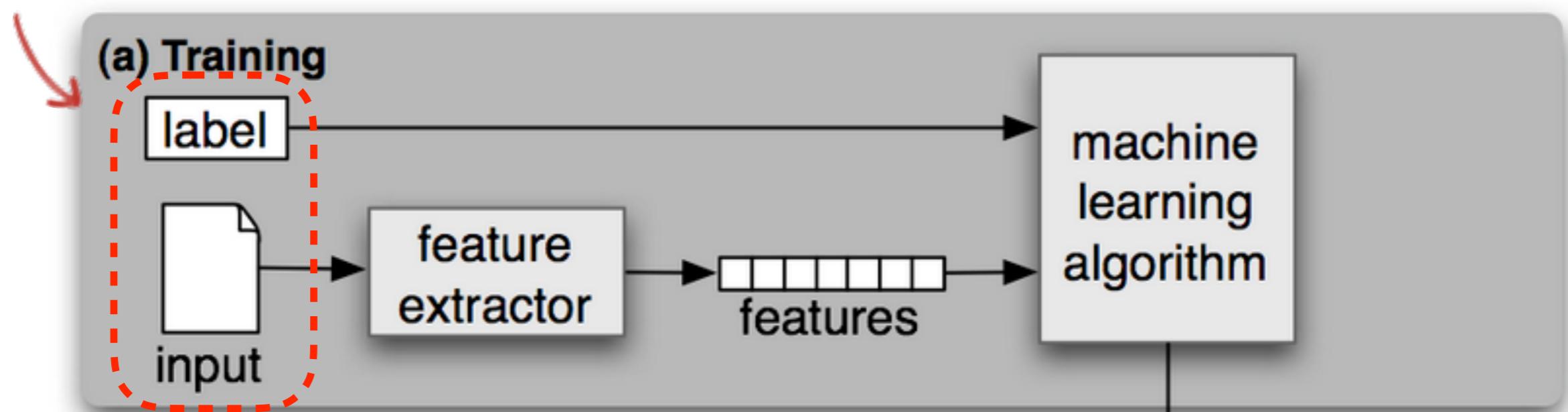
(b) Prediction



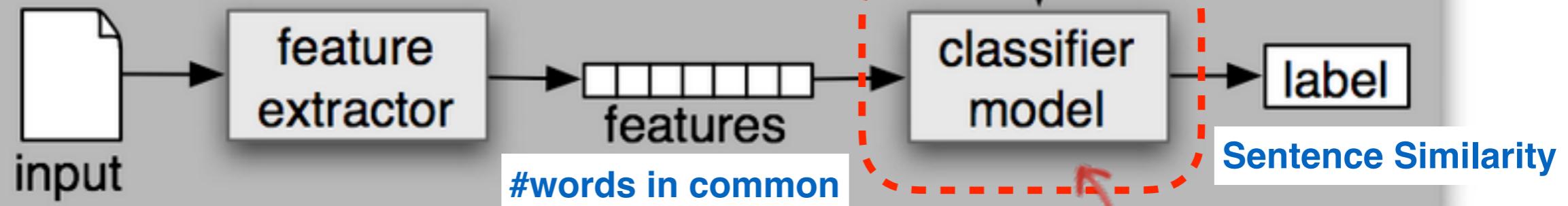
**(also called) hypothesis**

# Supervised Machine Learning

training set



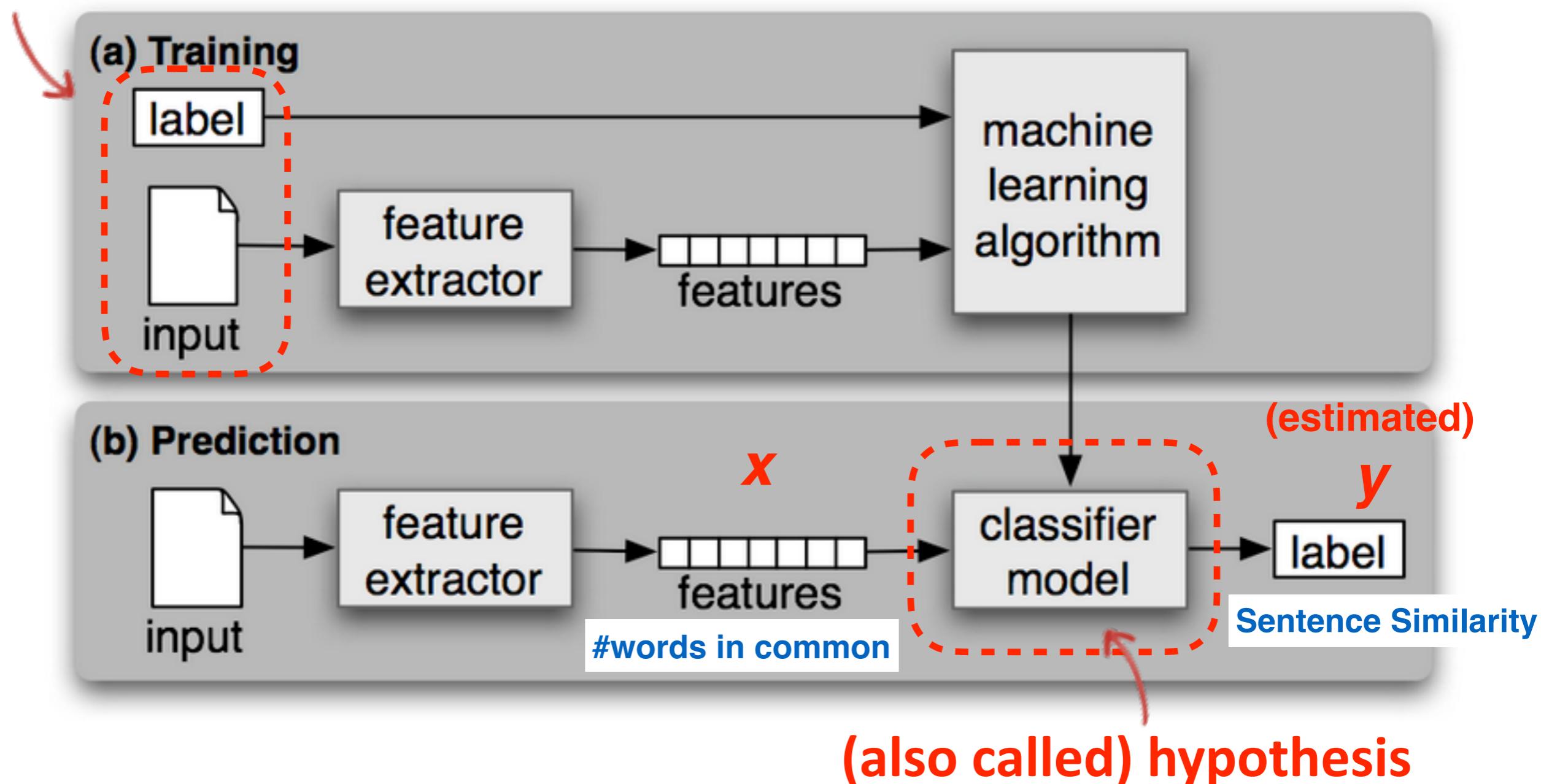
(b) Prediction



(also called) hypothesis

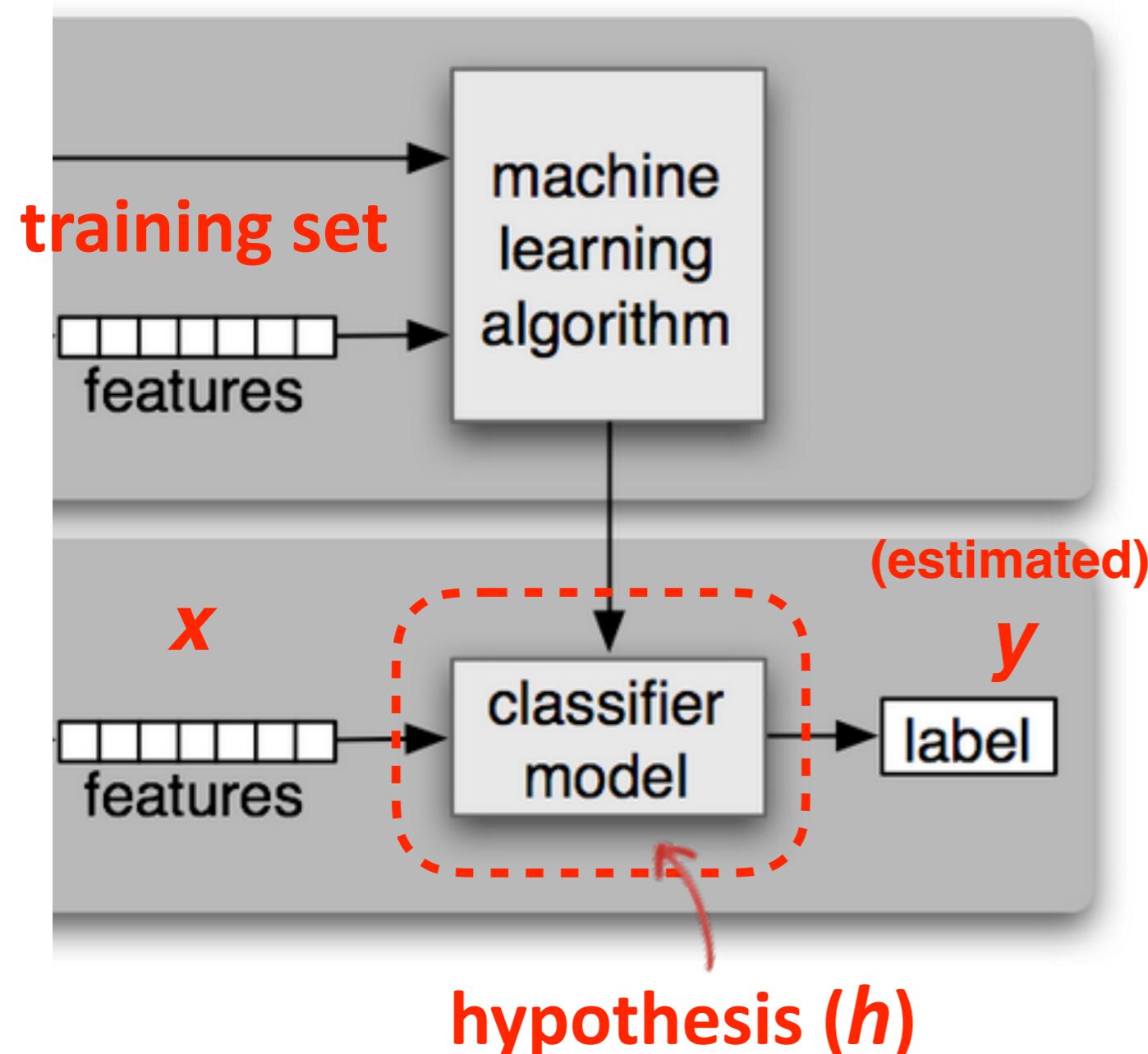
# Supervised Machine Learning

training set



# Linear Regression: Model Representation

- How to represent  $h$  ?



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

Linear Regression  
w/ one variable

# Linear Regression w/ one variable: Model Representation

| #words in common<br>( $x$ ) | Sentence Similarity<br>( $y$ ) |
|-----------------------------|--------------------------------|
| 1                           | 0                              |
| 4                           | 1                              |
| 13                          | 4                              |
| 18                          | 5                              |
| ...                         | ...                            |

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

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

# Linear Regression w/ one variable: Model Representation

| #words in common<br>( $x$ ) | Sentence Similarity<br>( $y$ ) |
|-----------------------------|--------------------------------|
| 1                           | 0                              |
| 4                           | 1                              |
| 13                          | 4                              |
| 18                          | 5                              |
| ...                         | ...                            |

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

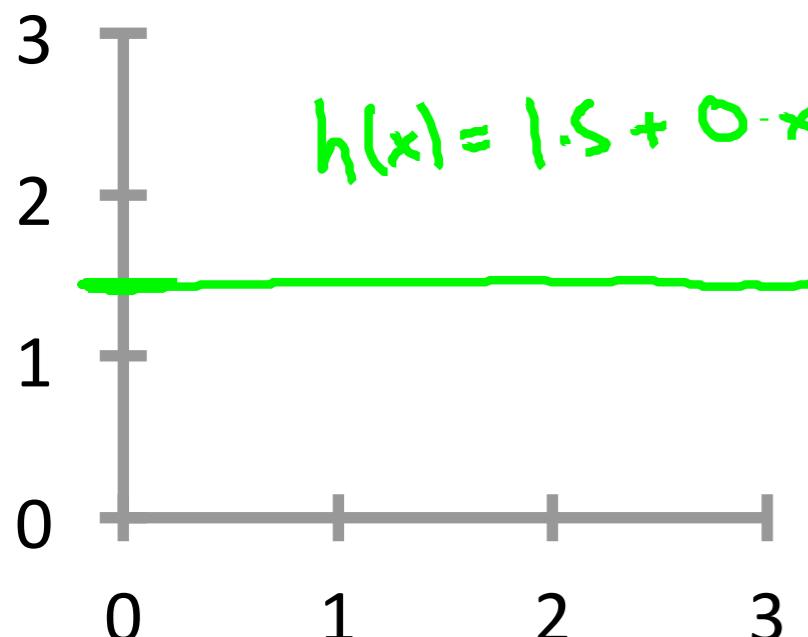
- $m$  hand-labeled sentence pairs  $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- $\theta$ 's: parameters



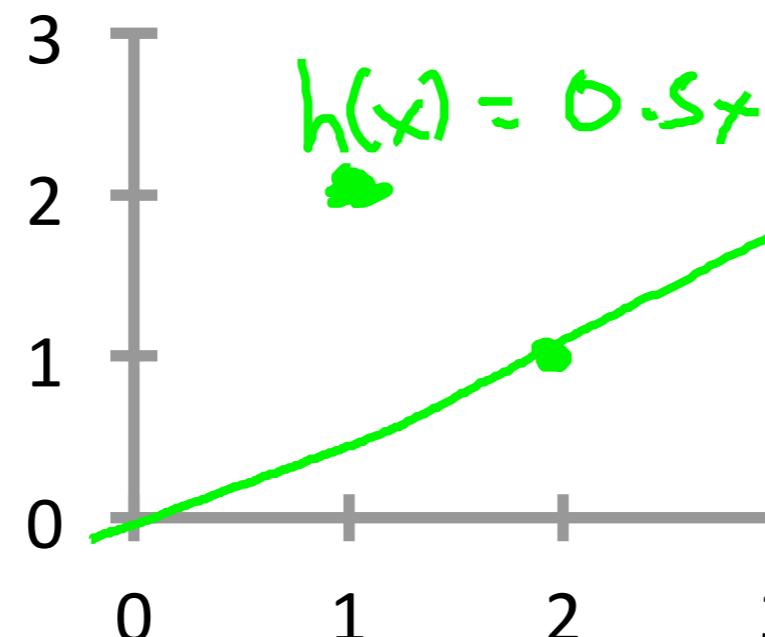
How to choose  $\theta$ ?

# Linear Regression w/ one variable:: Model Representation

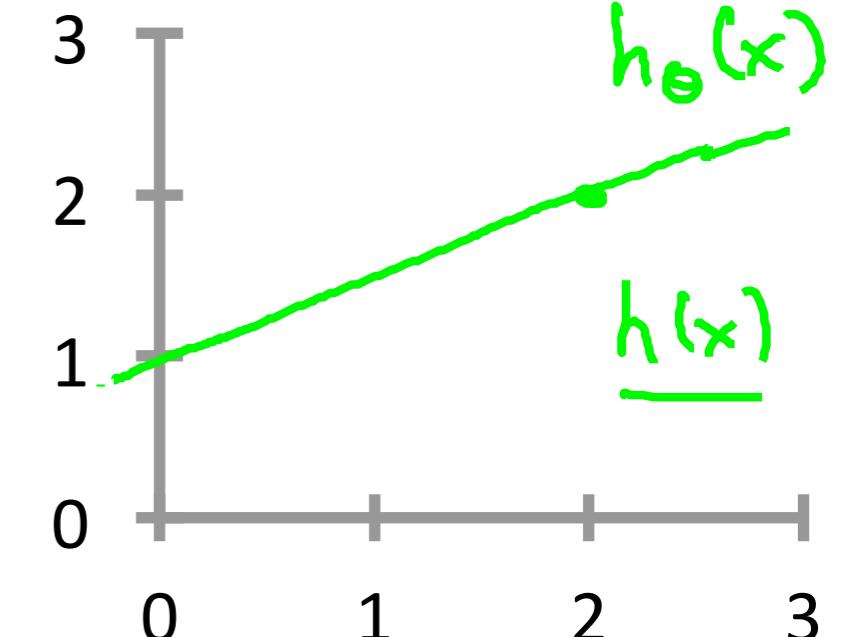
$$\underline{h_{\theta}(x) = \theta_0 + \theta_1 x}$$



$$\begin{aligned} \rightarrow \theta_0 &= 1.5 \\ \rightarrow \theta_1 &= 0 \end{aligned}$$

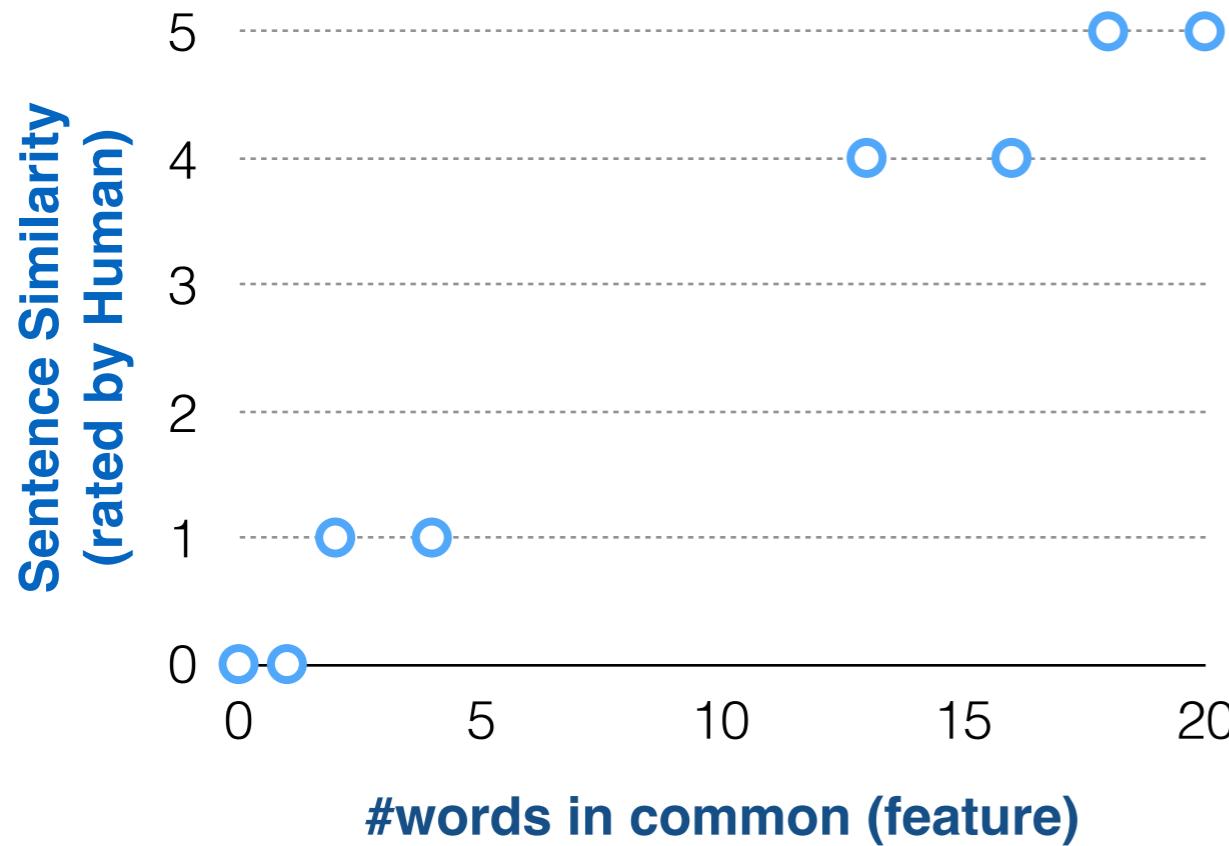


$$\begin{aligned} \rightarrow \theta_0 &= 0 \\ \rightarrow \theta_1 &= 0.5 \end{aligned}$$



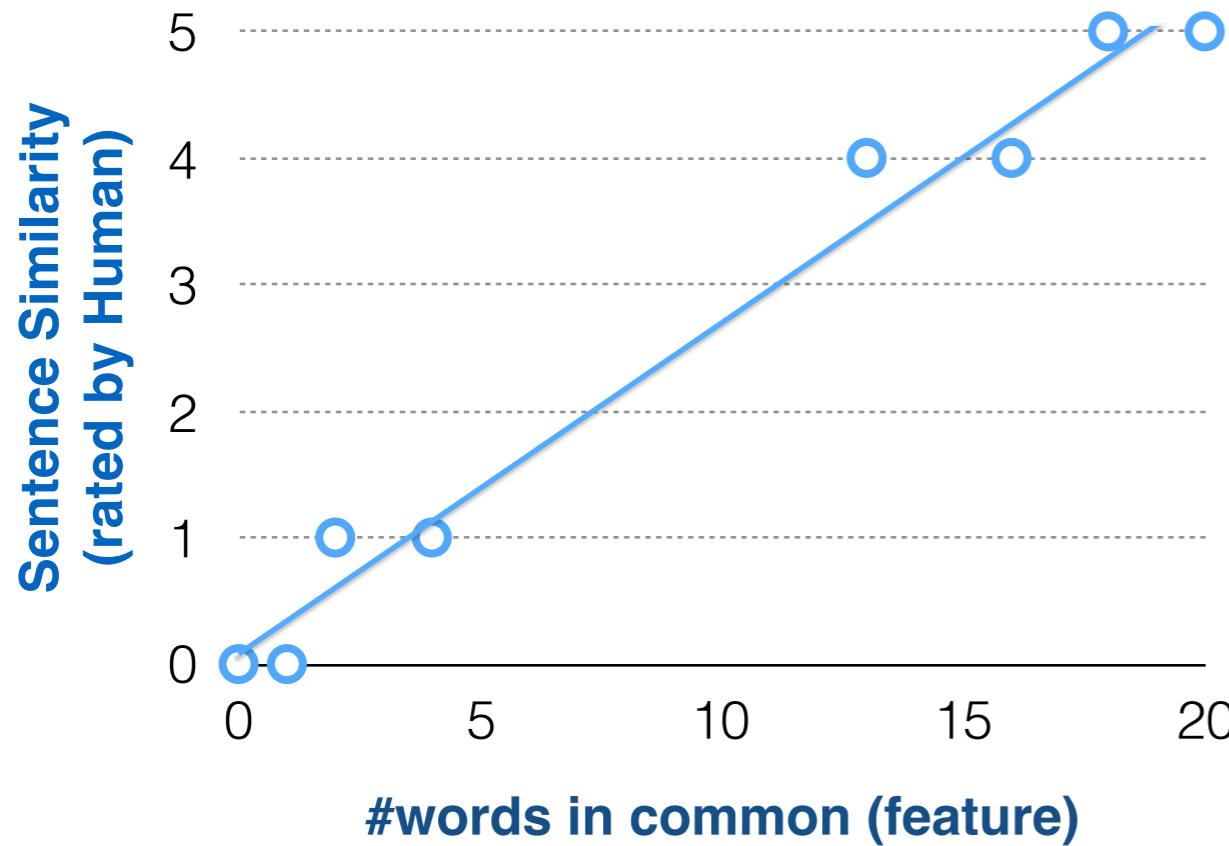
$$\begin{aligned} \rightarrow \theta_0 &= 1 \\ \rightarrow \theta_1 &= 0.5 \end{aligned}$$

# Linear Regression w/ one variable: Cost Function



- **Idea:** choose  $\theta_0, \theta_1$  so that  $h_{\theta}(x)$  is close to  $y$  for training examples  $(x, y)$

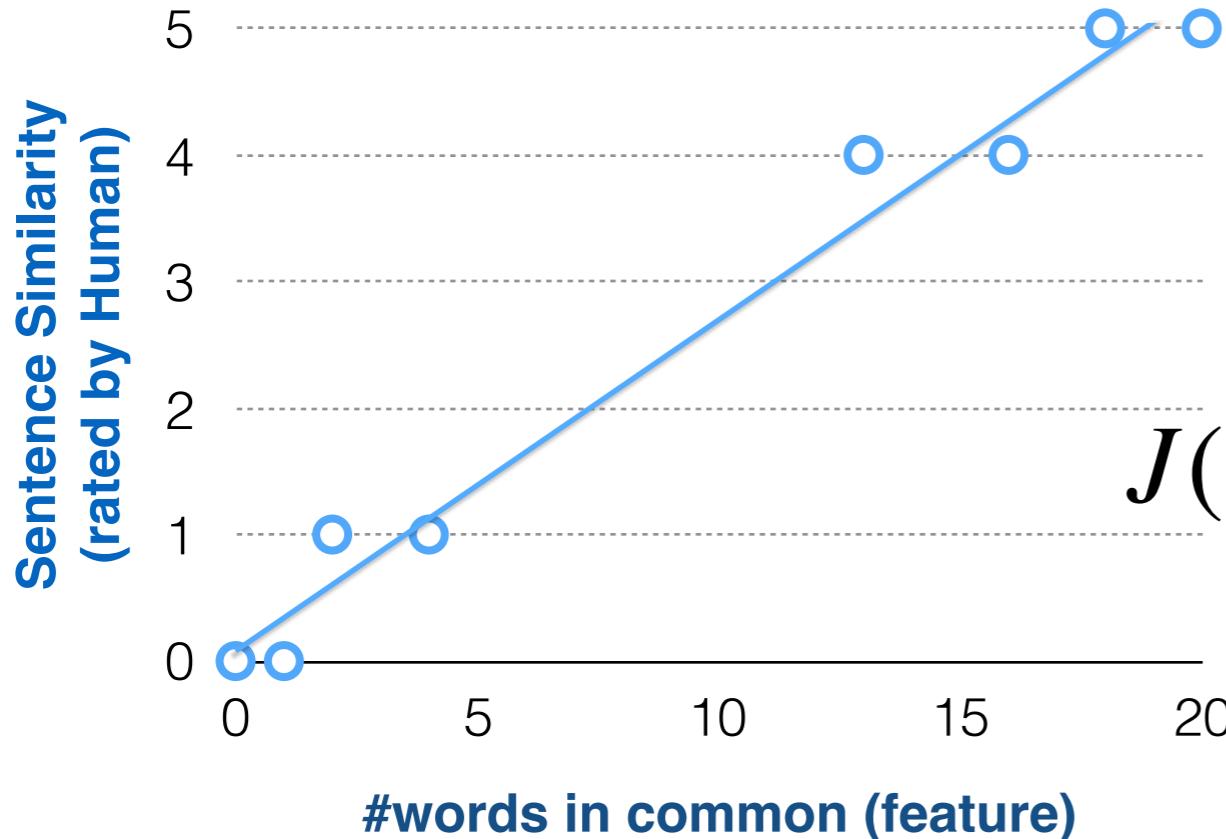
# Linear Regression w/ one variable: Cost Function



- **Idea:** choose  $\theta_0, \theta_1$  so that  $h_{\theta}(x)$  is close to  $y$  for training examples  $(x, y)$

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  $\theta_0, \theta_1$  so that  $h_\theta(x)$  is close to  $y$  for training examples  $(x, y)$

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

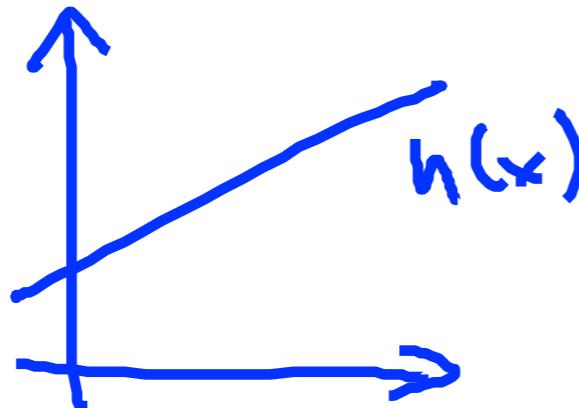
# Linear Regression

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

# Linear Regression

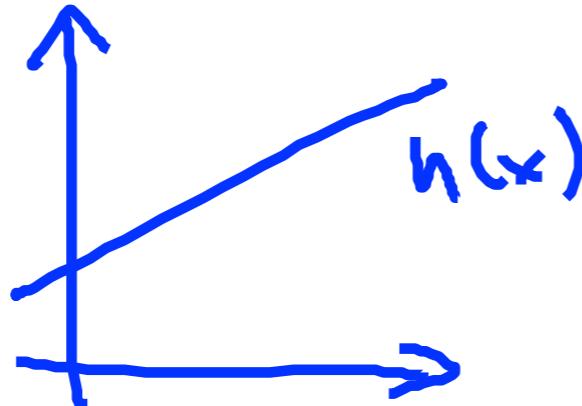
## Simplified

- Hypothesis:

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

- Parameters:

$$\theta_0, \theta_1$$

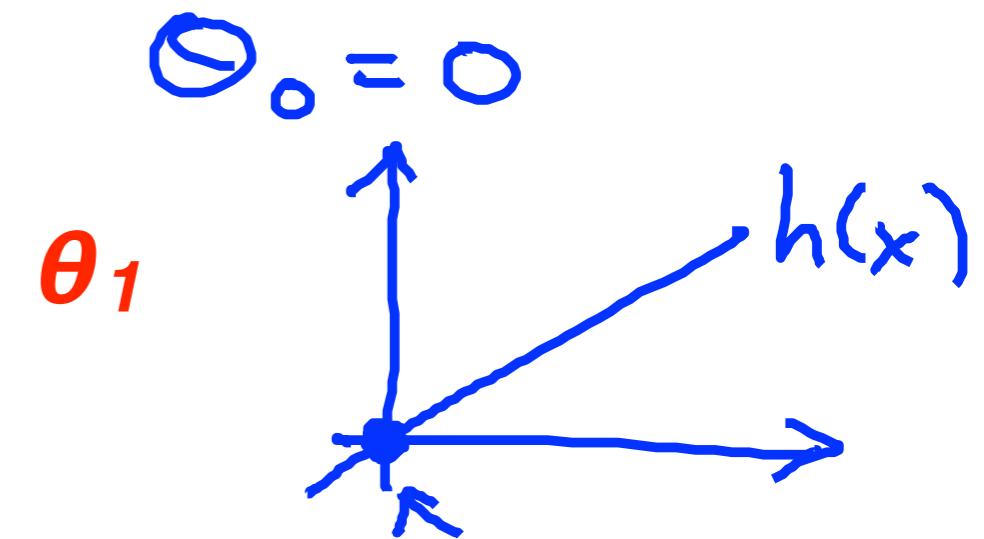


- Cost Function:

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

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

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

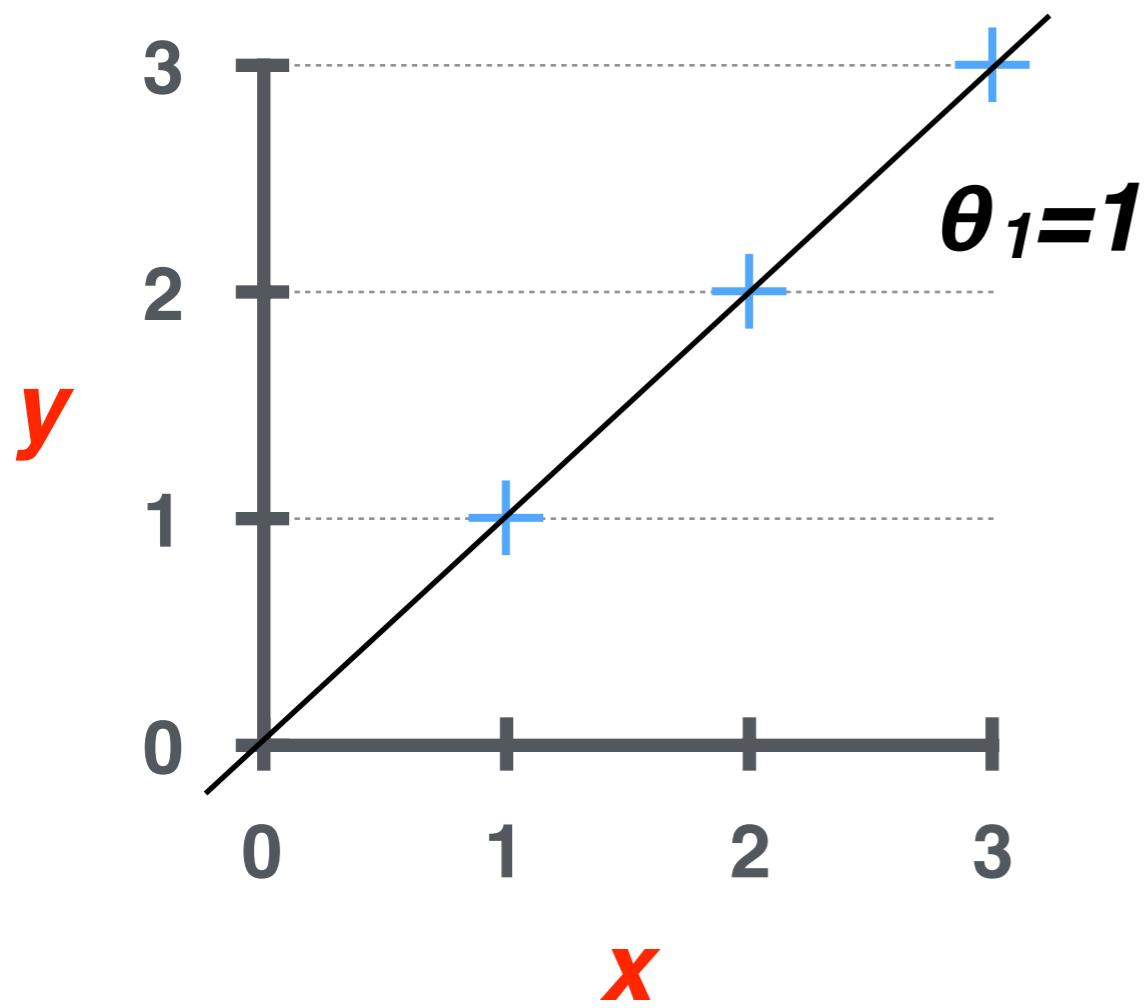


$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

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

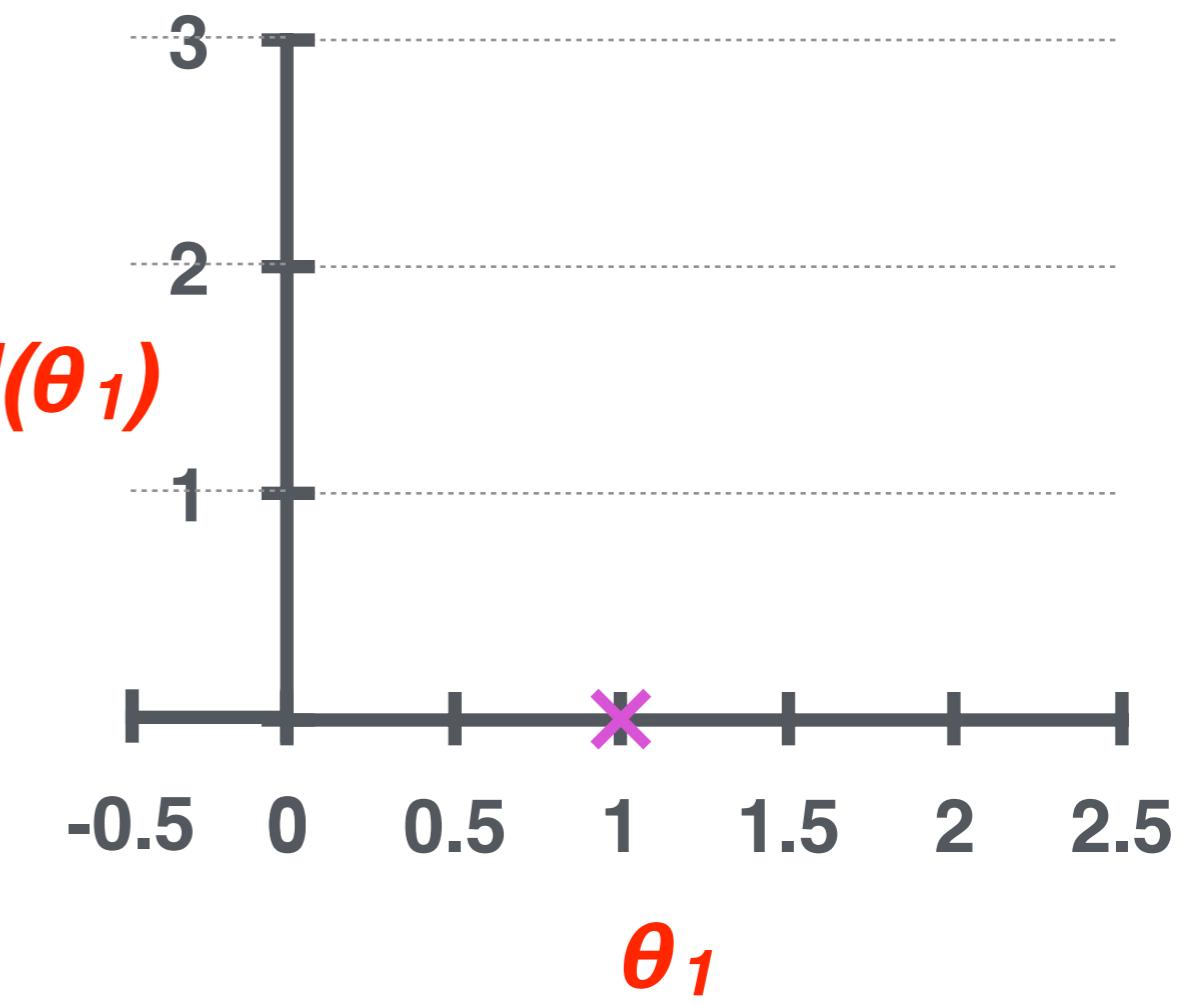
$h_{\theta}(x)$

(for fixed  $\theta_1$ , this is a function of  $x$ )



$J(\theta_1)$

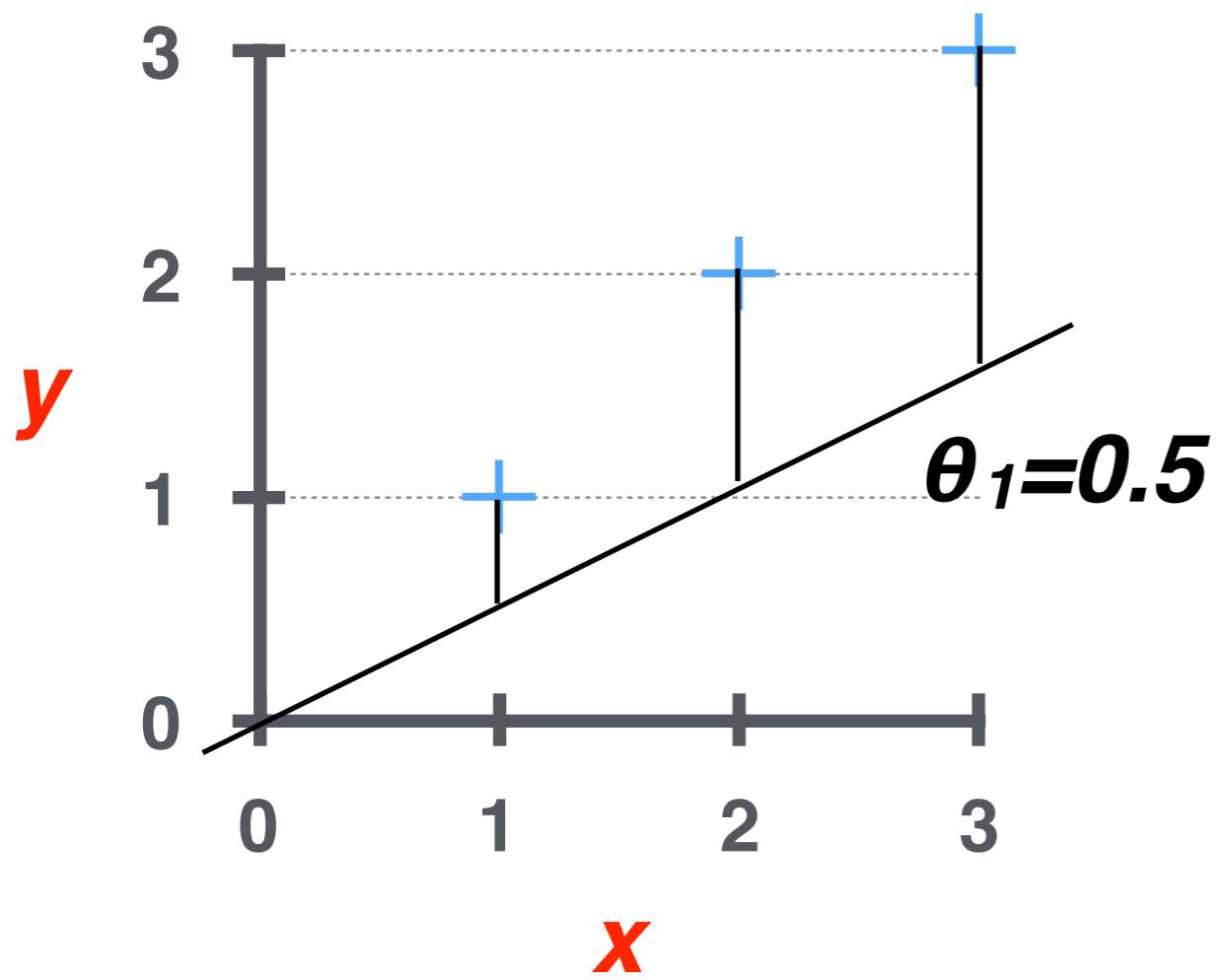
(function of the parameter  $\theta_1$ )



$$J(1) = \frac{1}{2 \times 3} [(1-1)^2 + (2-2)^2 + (3-3)^2] = 0$$

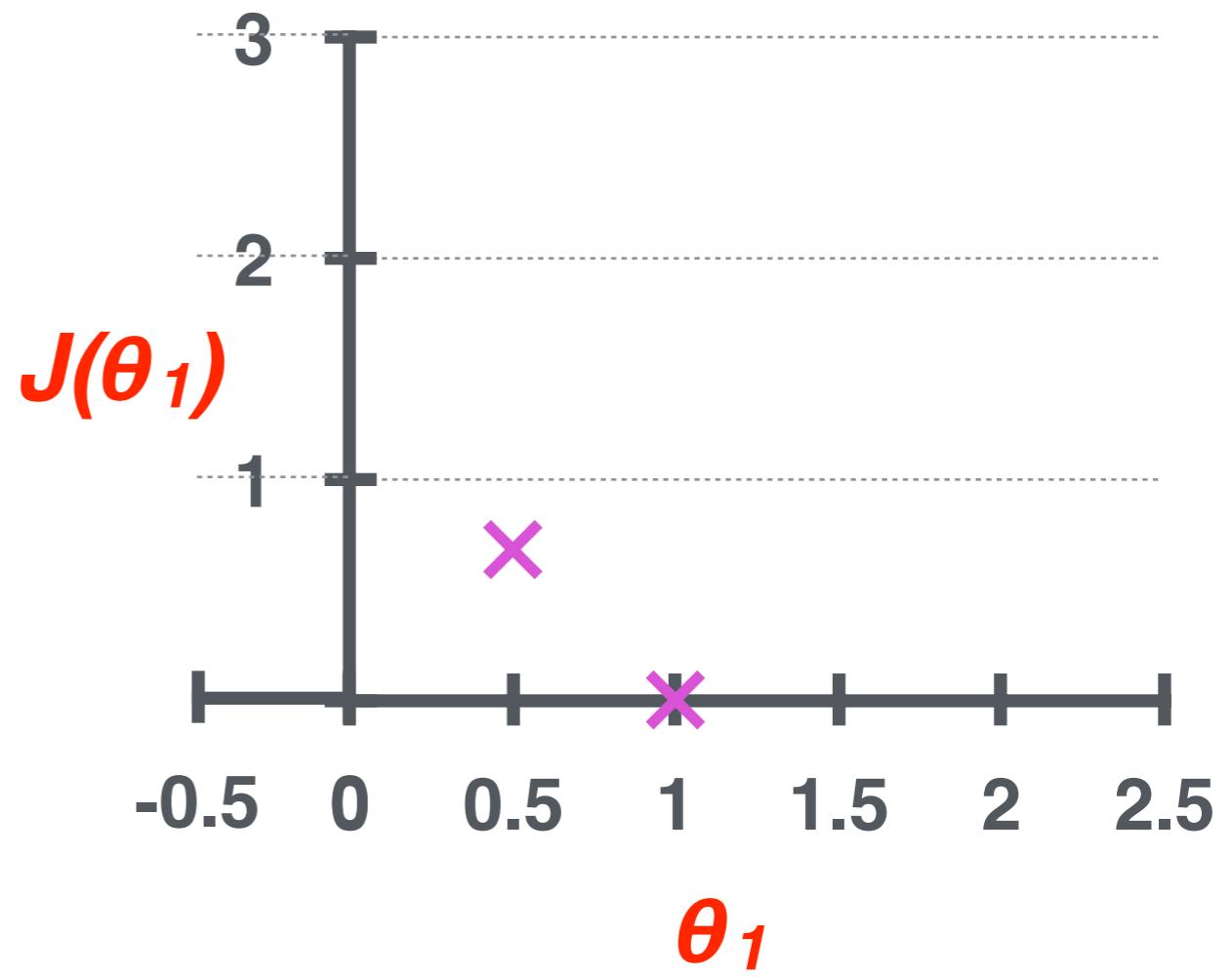
$h_{\theta}(x)$

(for fixed  $\theta_1$ , this is a function of  $x$ )



$J(\theta_1)$

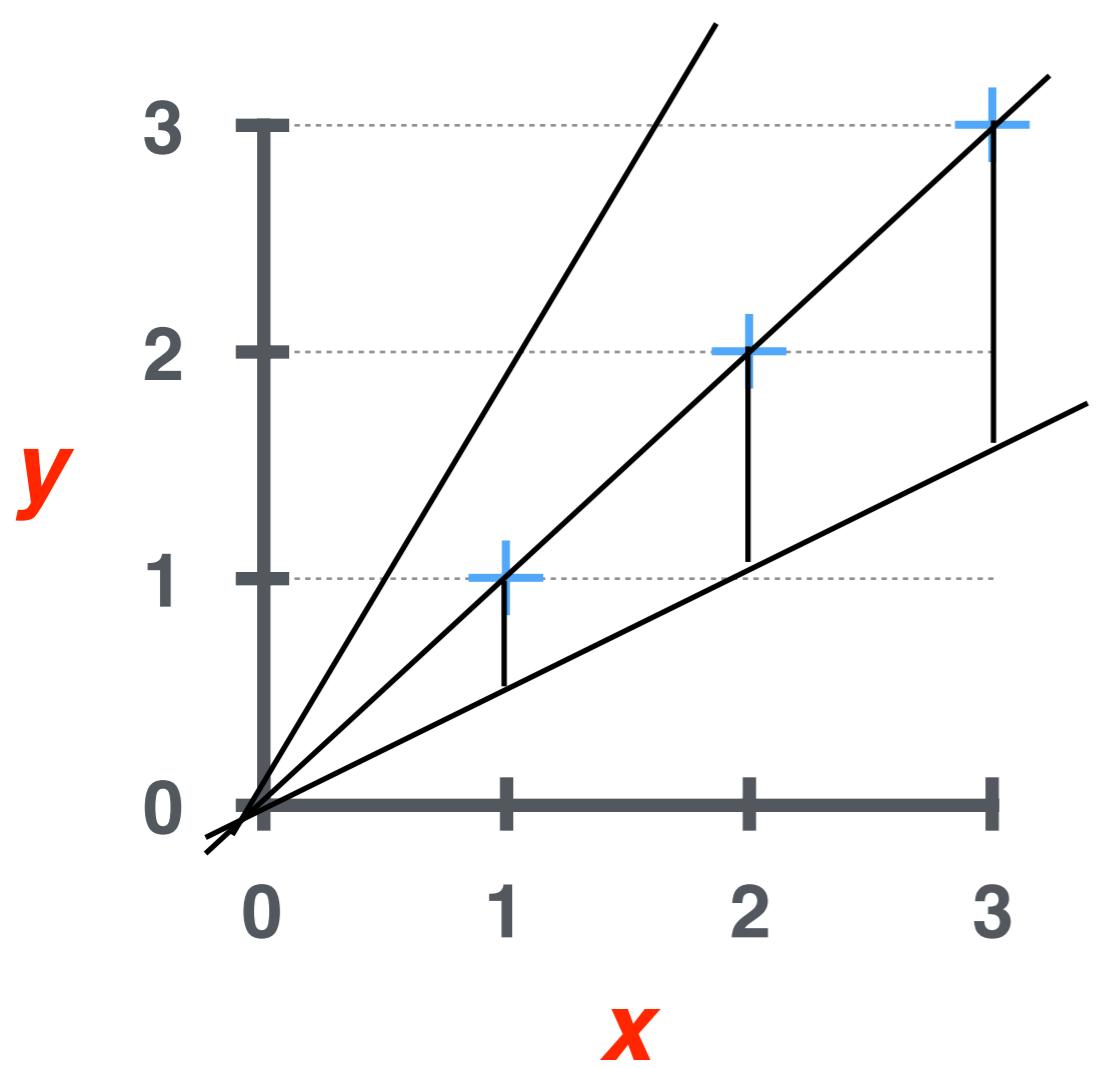
(function of the parameter  $\theta_1$ )



$$J(1) = \frac{1}{2 \times 3} [(0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2] = 0.68$$

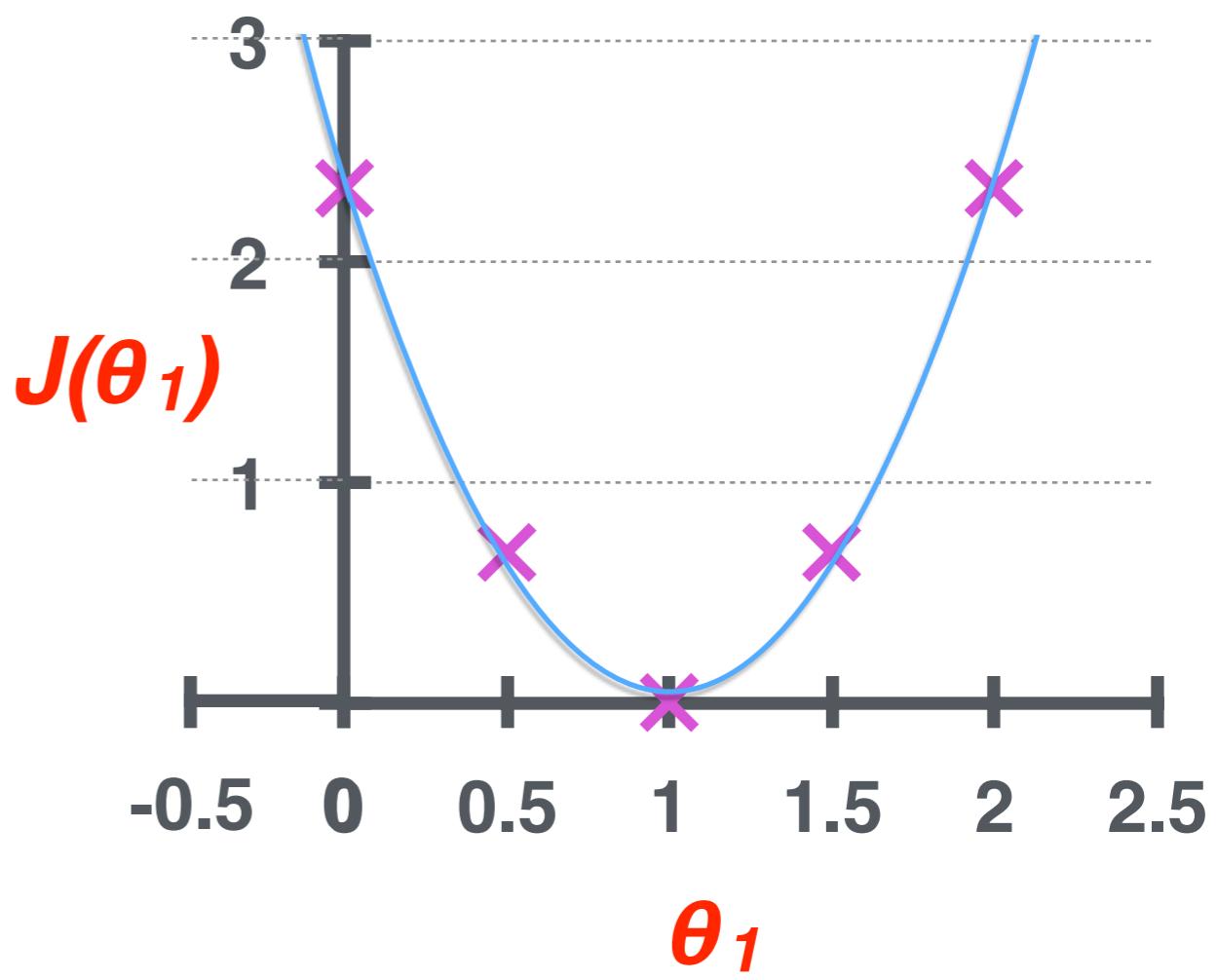
$h_{\theta}(x)$

(for fixed  $\theta_1$ , this is a function of  $x$ )



$J(\theta_1)$

(function of the parameter  $\theta_1$ )



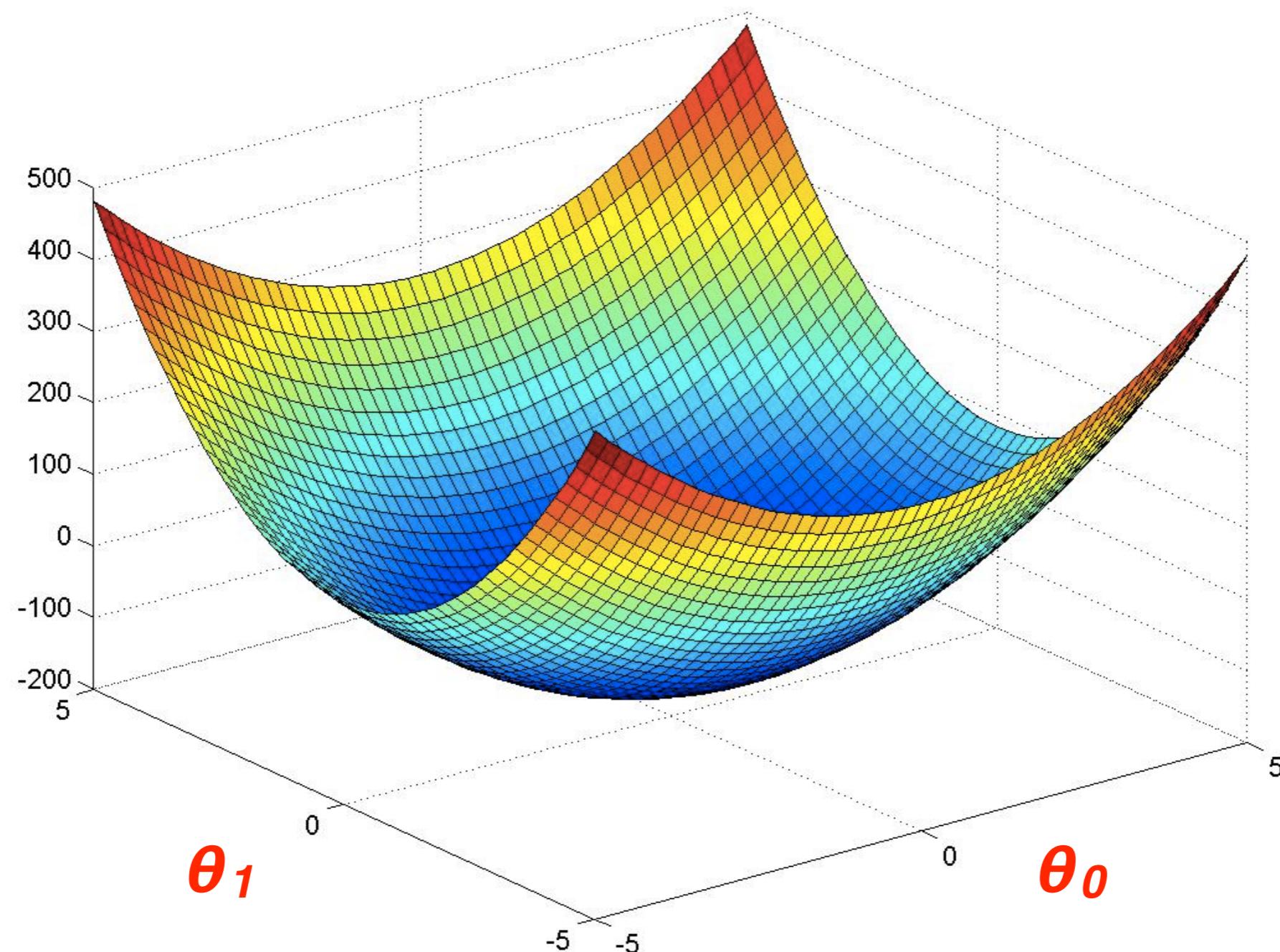
minimize  $J(\theta_1)$   
 $\theta_1$

$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )

$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )

$$J(\theta_1, \theta_2)$$


# Linear Regression

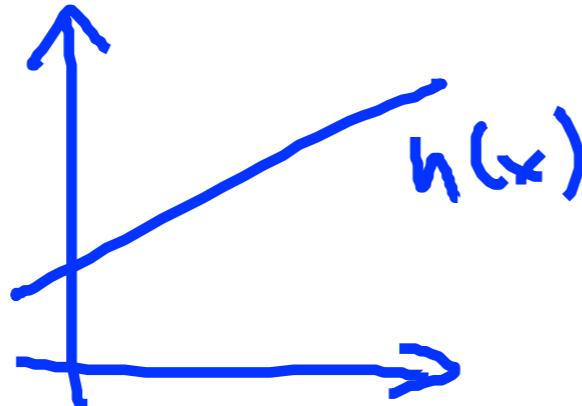
## Simplified

- Hypothesis:

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

- Parameters:

$$\theta_0, \theta_1$$

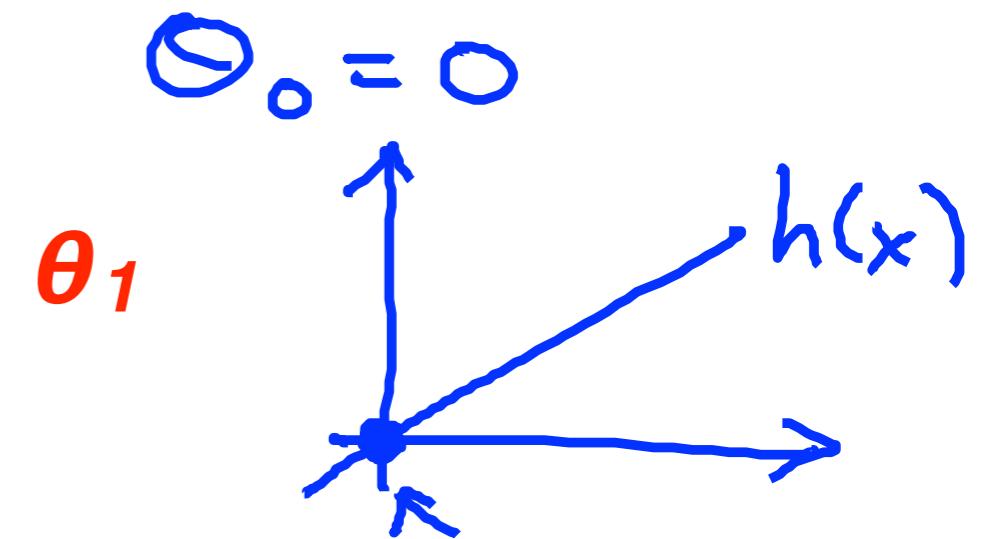


- Cost Function:

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

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

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



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

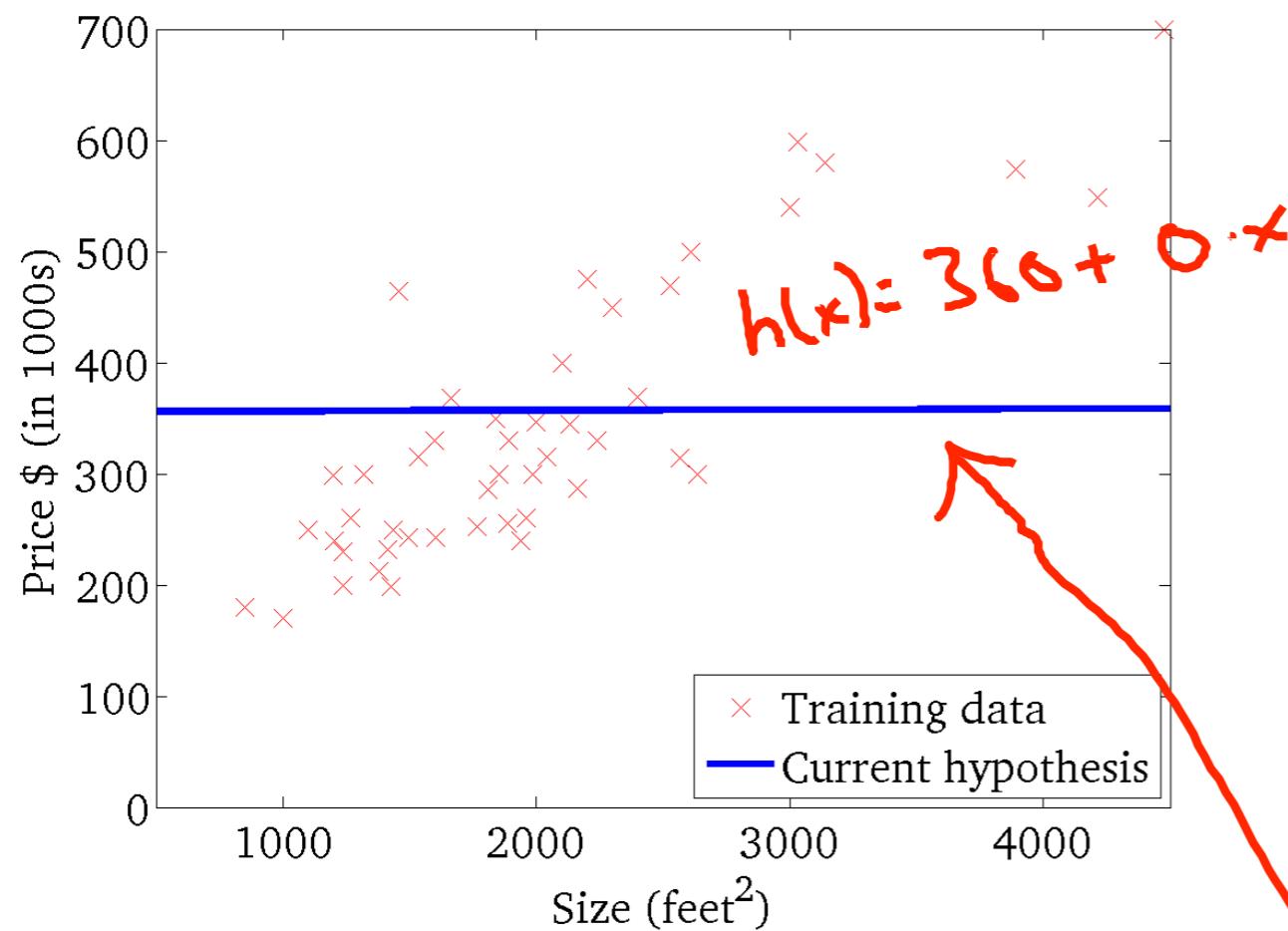
$$\underset{\theta_1}{\text{minimize}} J(\theta_1)$$

$$h_{\theta}(x)$$

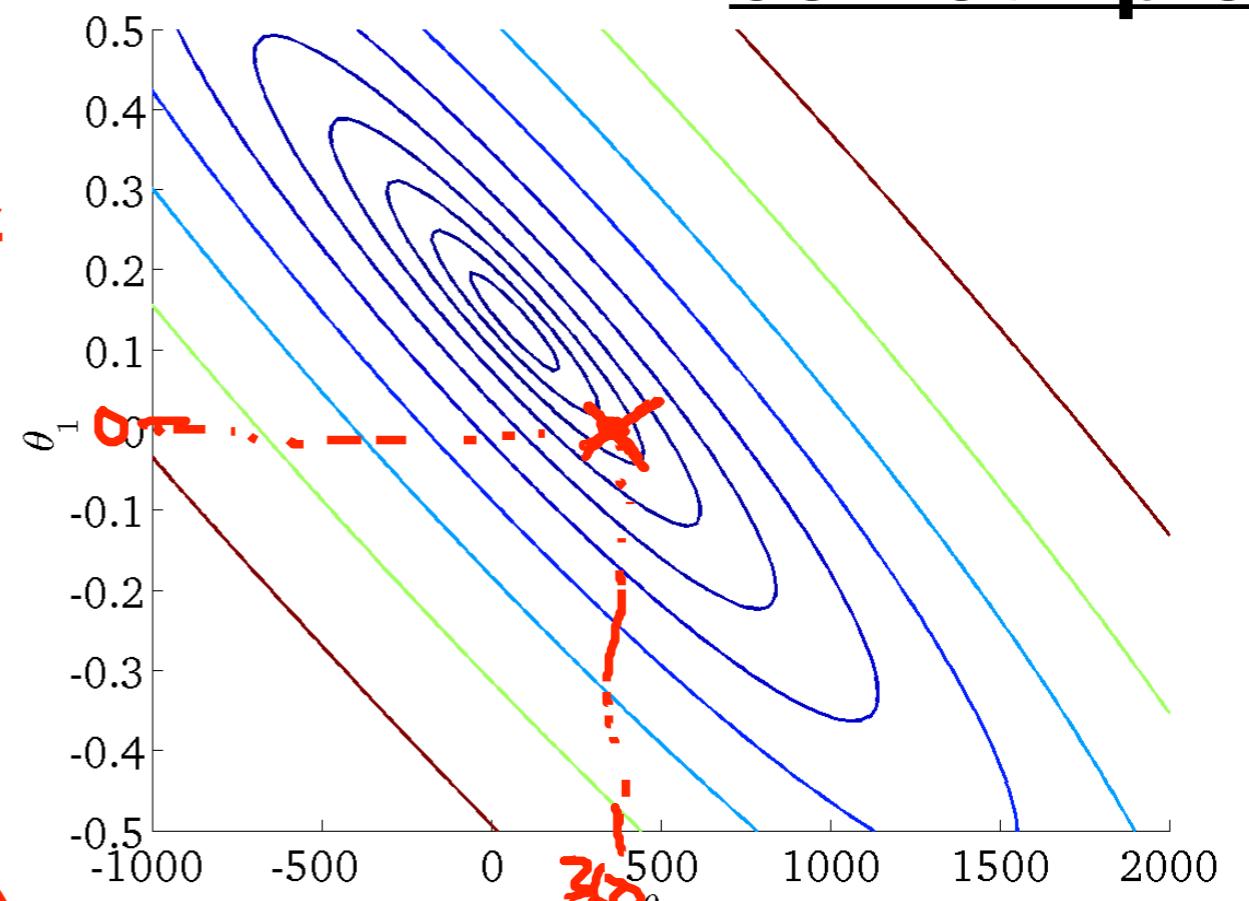
(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )

$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



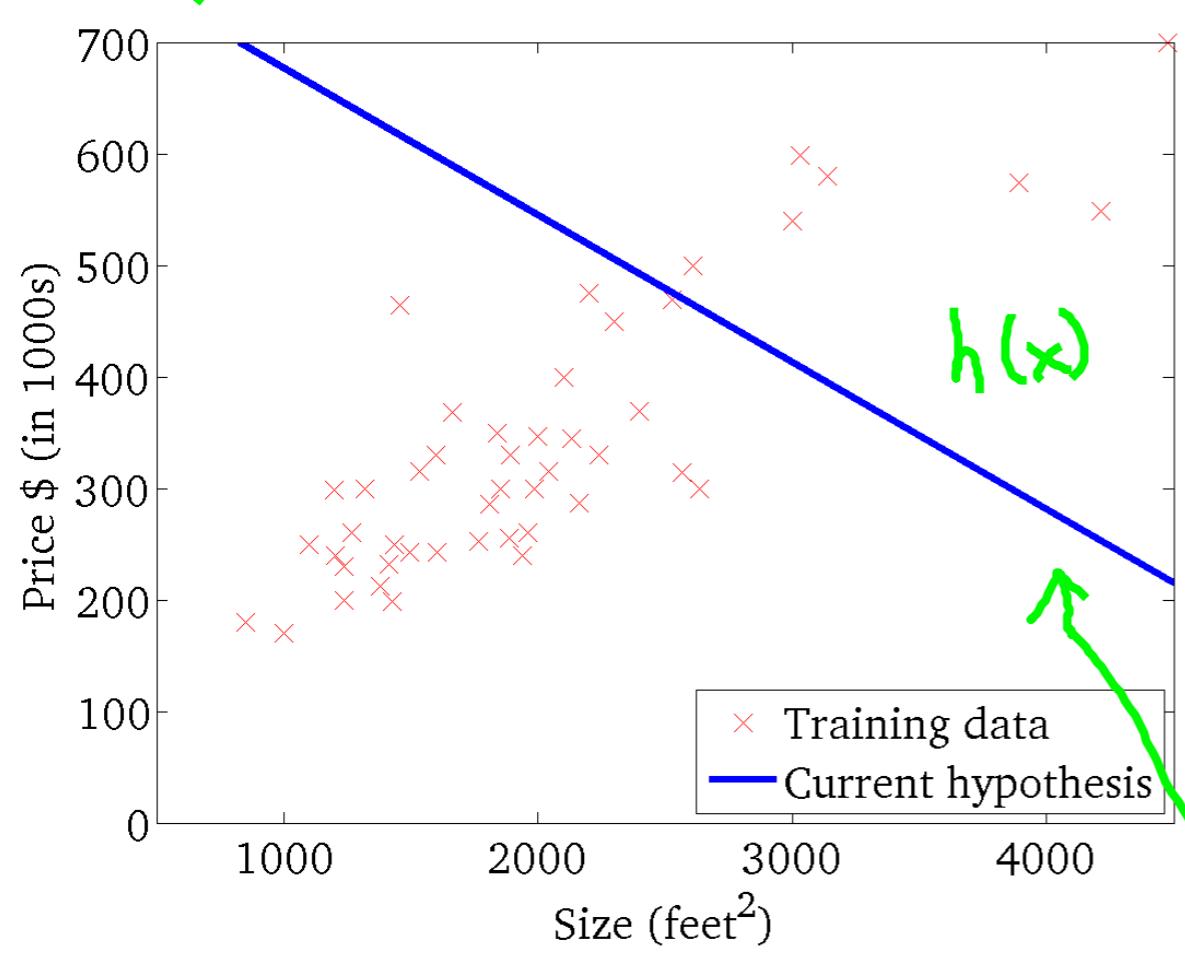
**contour plot**



$$\begin{cases} \theta_0 = 360 \\ \theta_1 \approx 0 \end{cases}$$

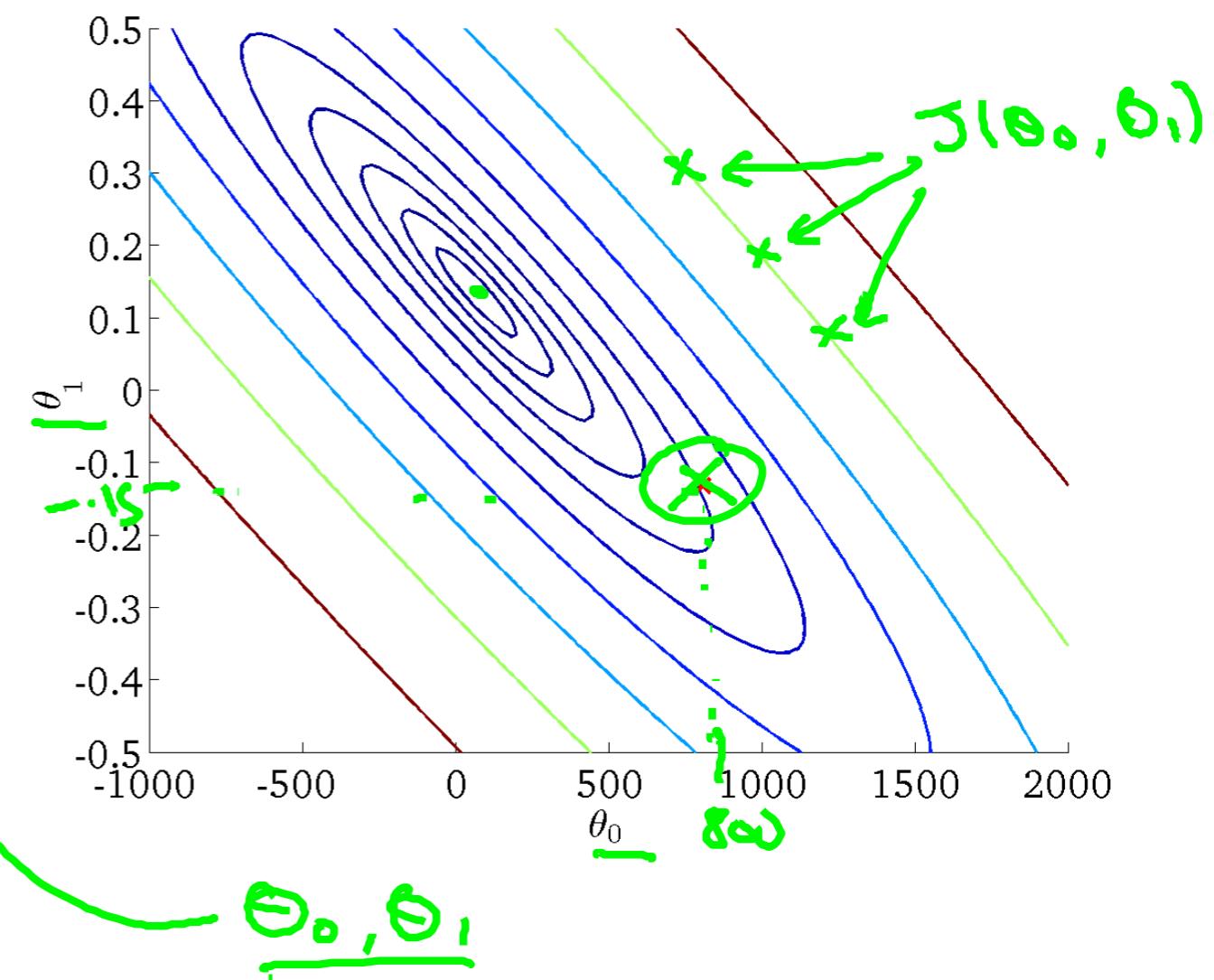
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



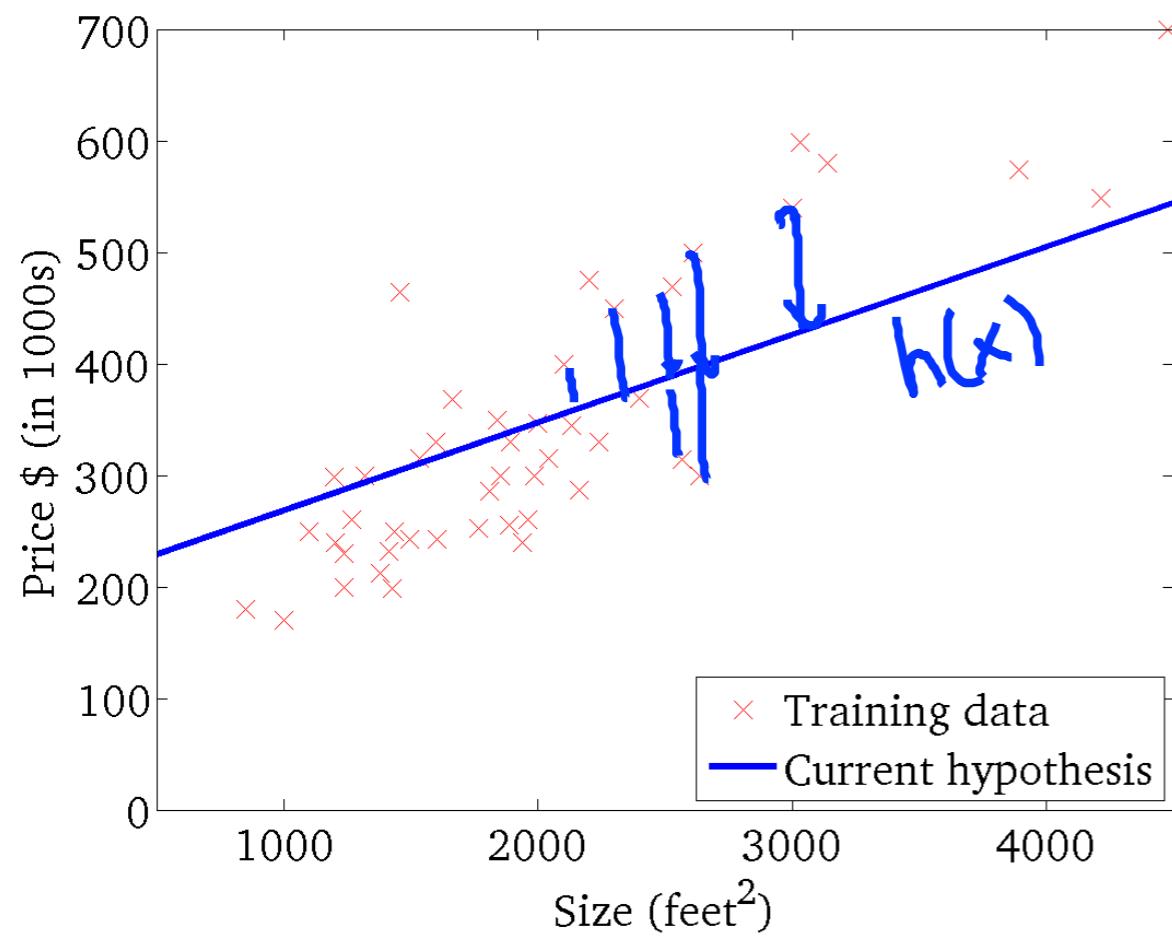
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



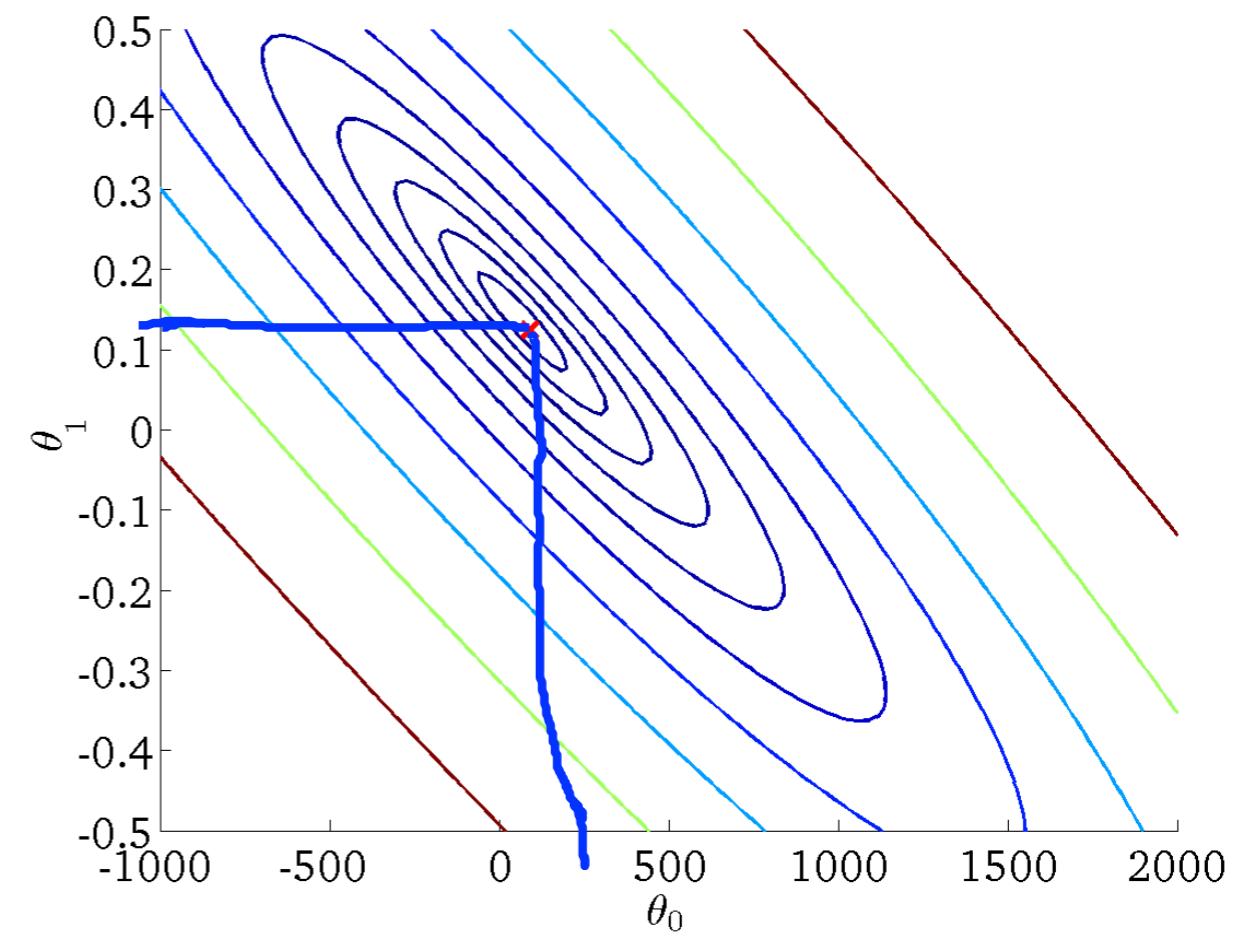
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

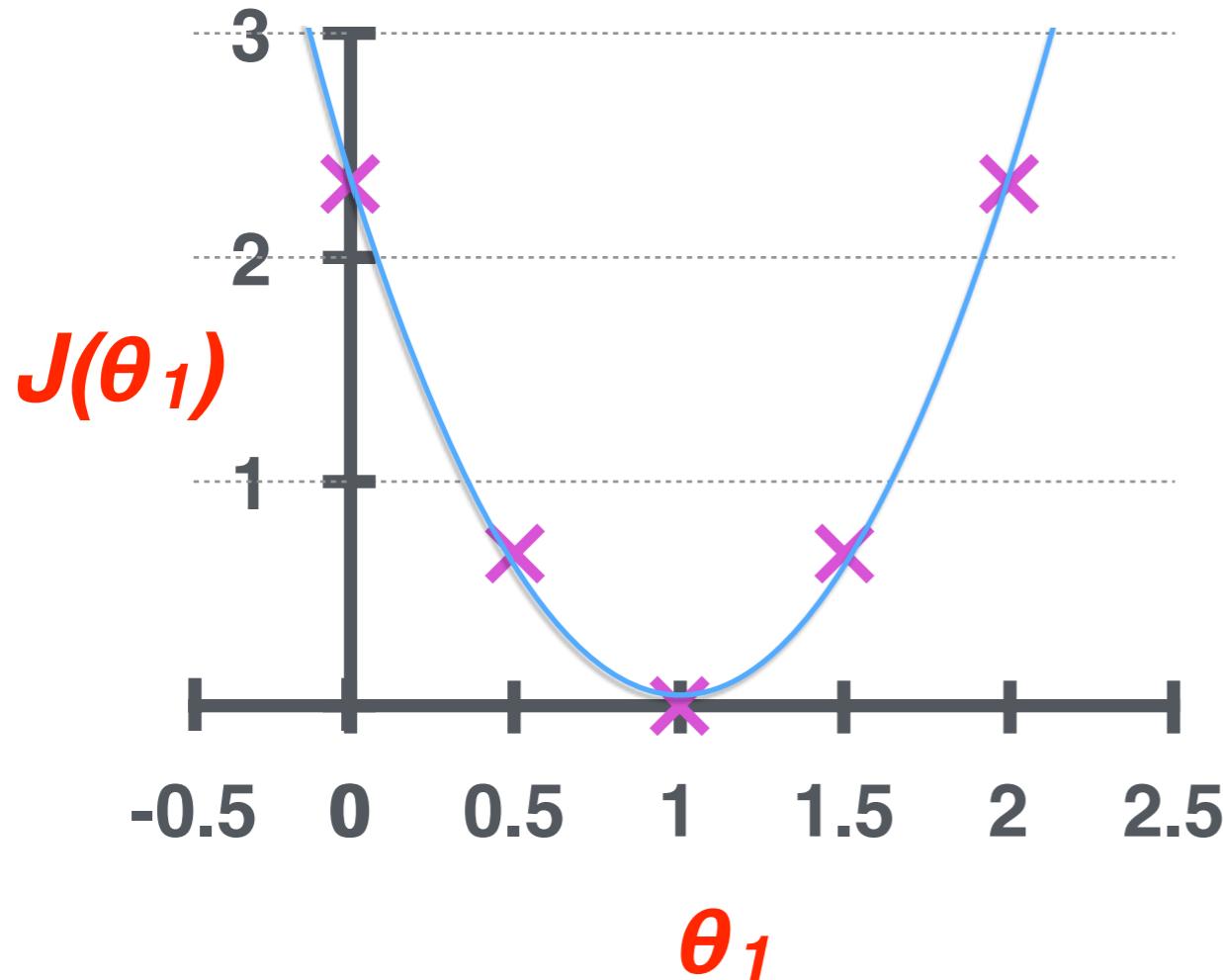
(function of the parameter  $\theta_0, \theta_1$ )



# Parameter Learning

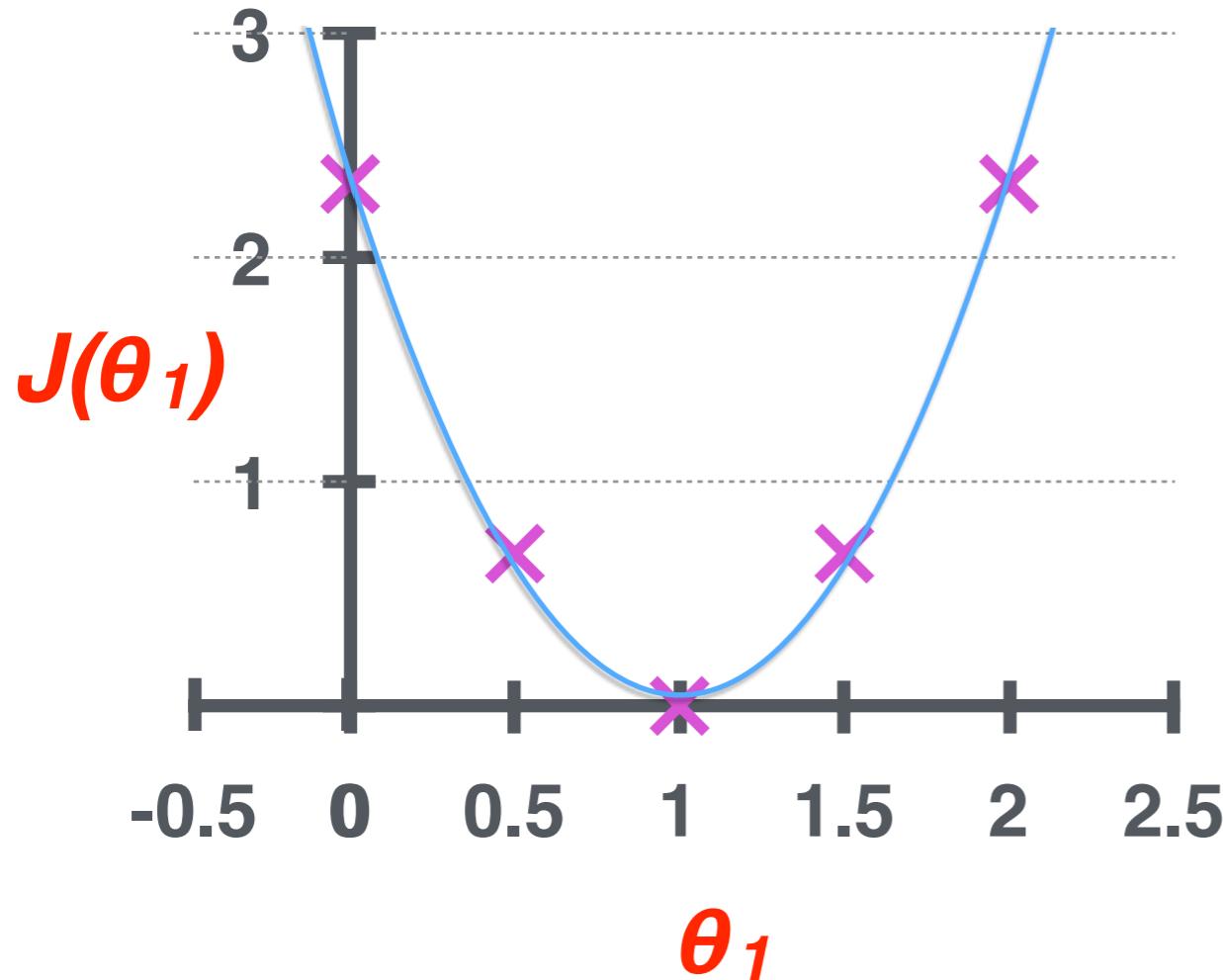
- Have some function  $J(\theta_0, \theta_1)$
- Want  $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$
- **Outline:**
  - Start with some  $\theta_0, \theta_1$
  - Keep changing  $\theta_0, \theta_1$  to reduce  $J(\theta_1, \theta_2)$  until we hopefully end up at a minimum

# Gradient Descent



minimize  $J(\theta_1)$   
 $\theta_1$

# Gradient Descent

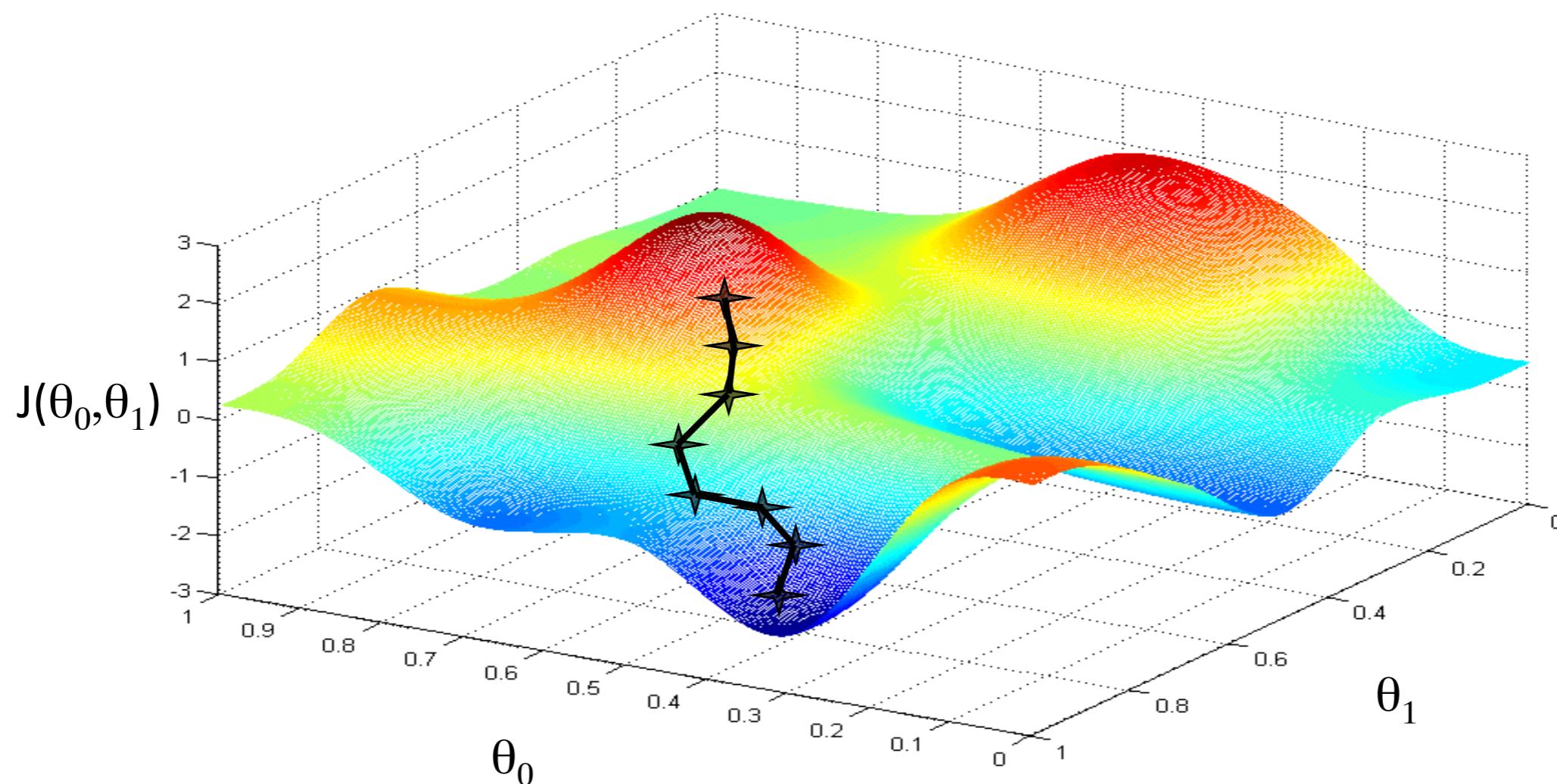


$$\theta_1 \doteq \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

learning rate

minimize  $J(\theta_1)$

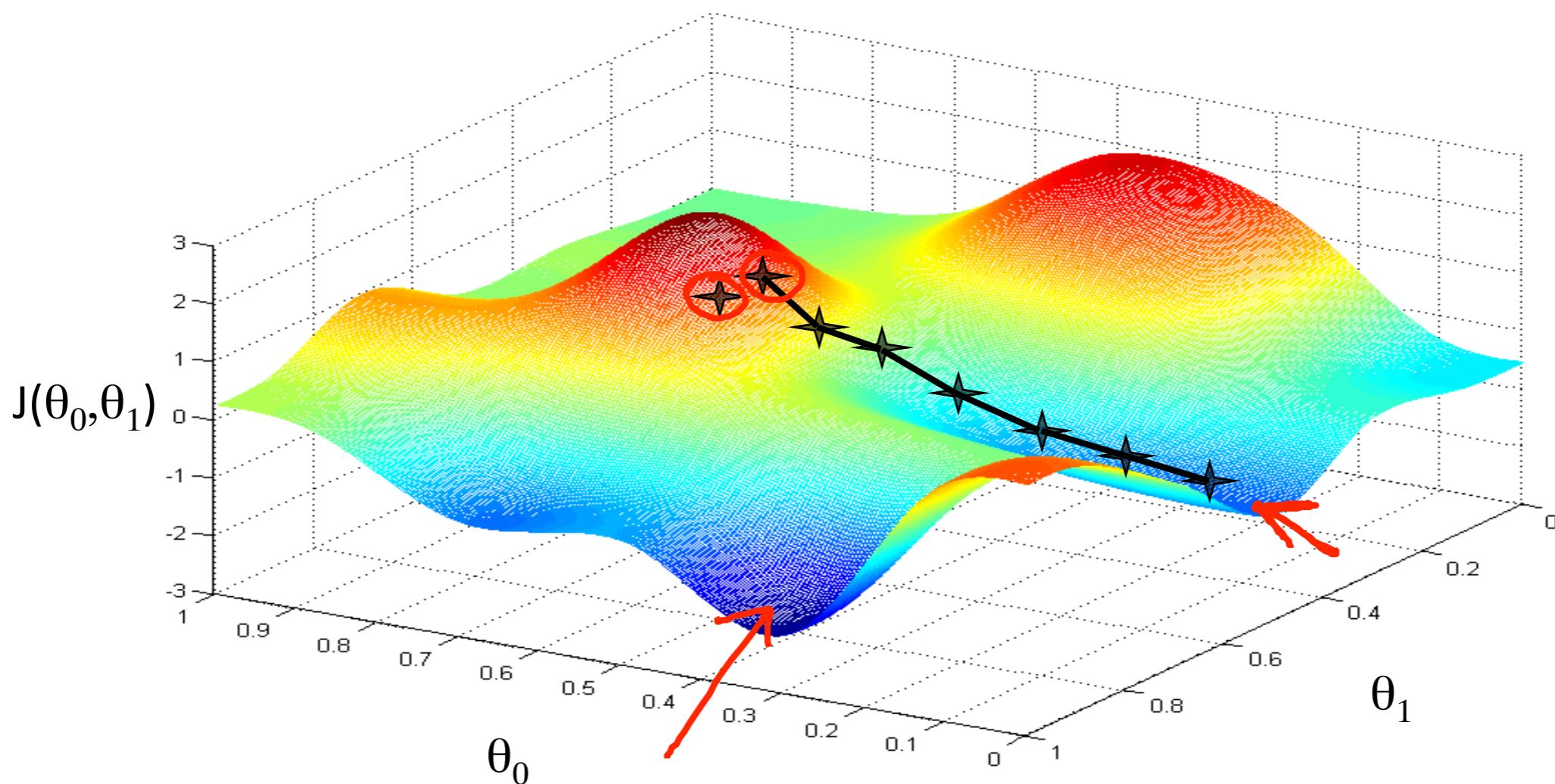
# Gradient Descent



minimize  $J(\theta_0, \theta_1)$   
 $\theta_0, \theta_1$

Andrew Ng

# Gradient Descent



minimize  $J(\theta_0, \theta_1)$   
 $\theta_0, \theta_1$

Andrew Ng

# 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)

}

Linear Regression w/ one variable:

# Gradient Descent

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

## Cost Function



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

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

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

Linear Regression w/ one variable:

# Gradient Descent

repeat until convergence {

$$\theta_0 \leftarrow \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})$$

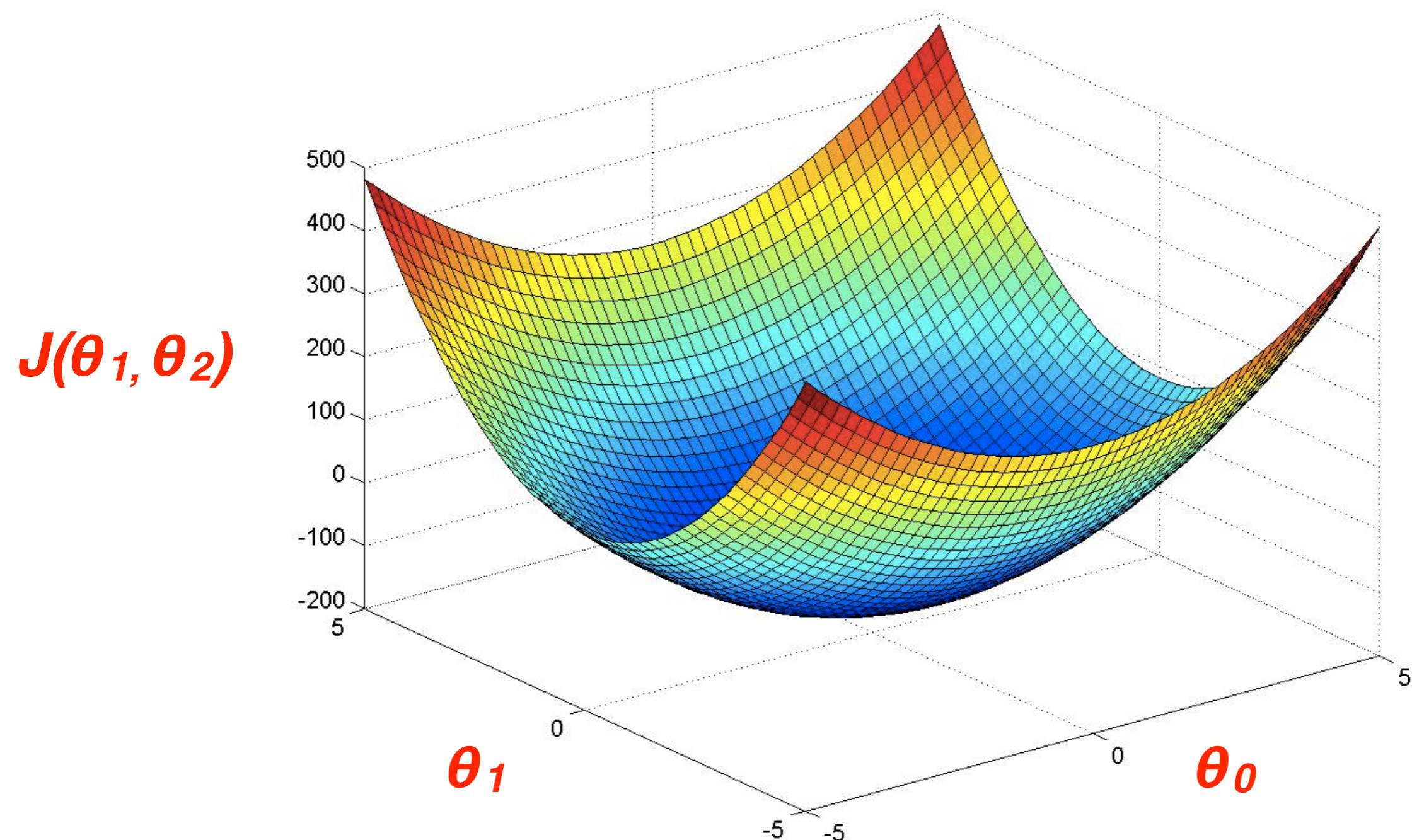
simultaneous update  $\theta_0, \theta_1$

$$\theta_1 \leftarrow \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

}

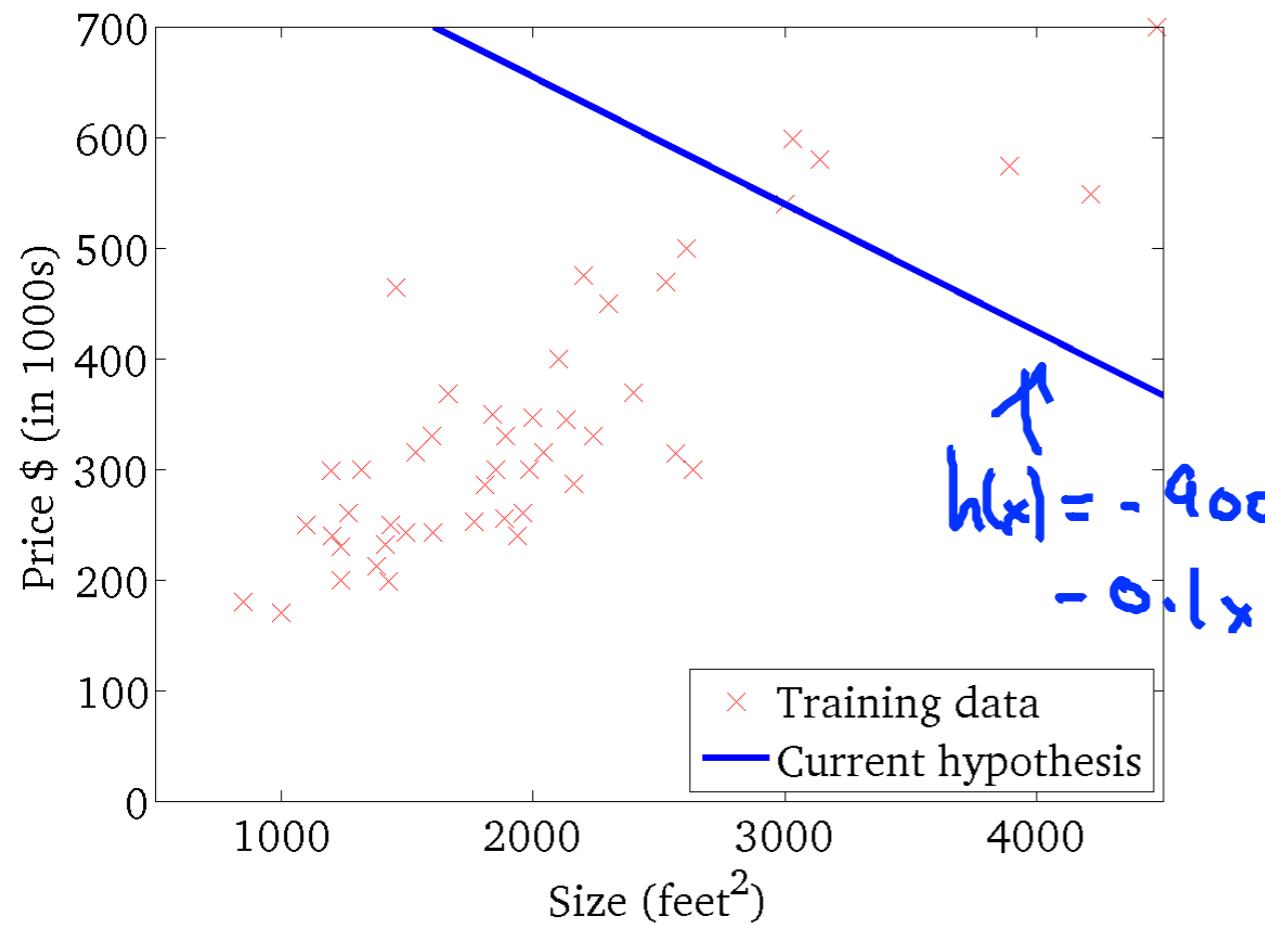
# Linear Regression

**cost function is convex**



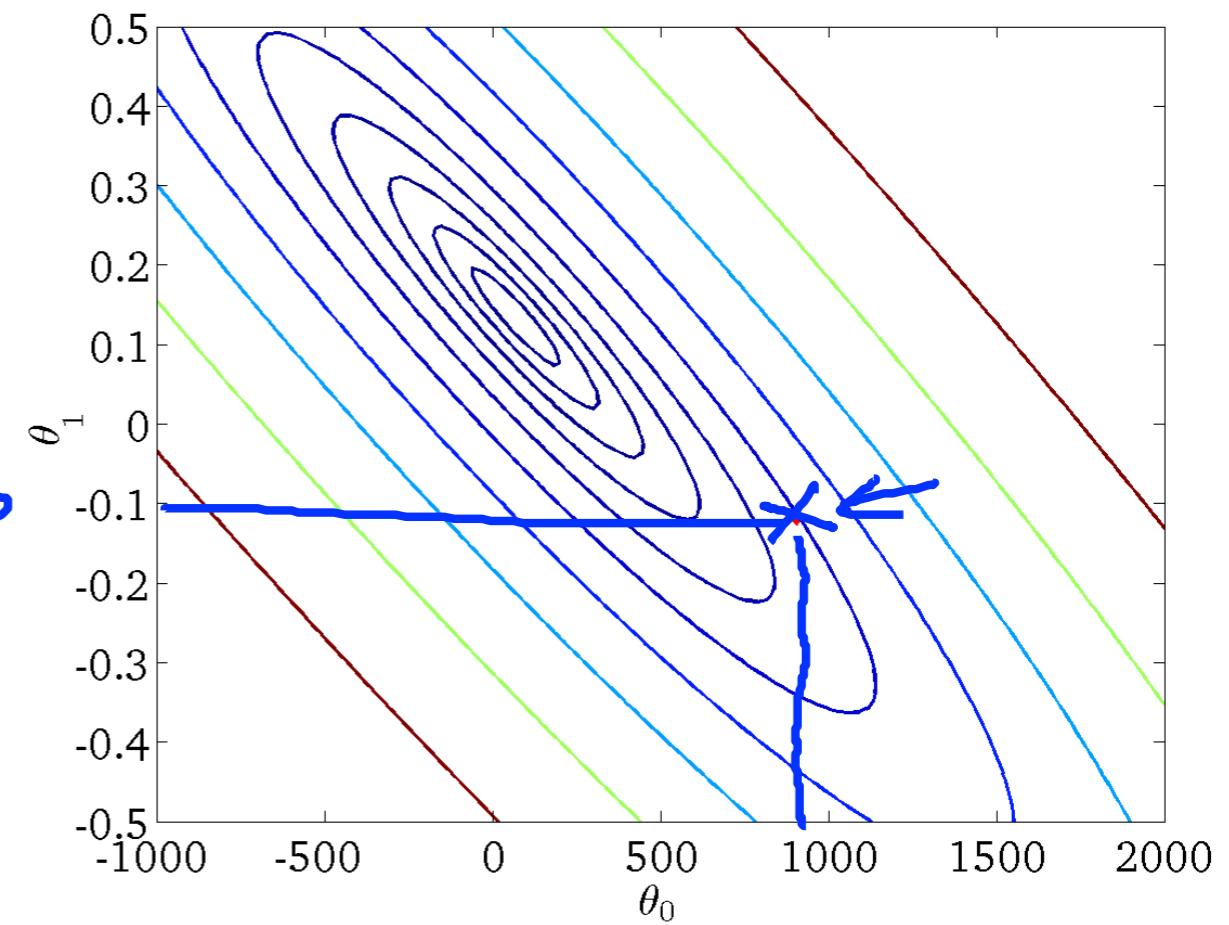
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



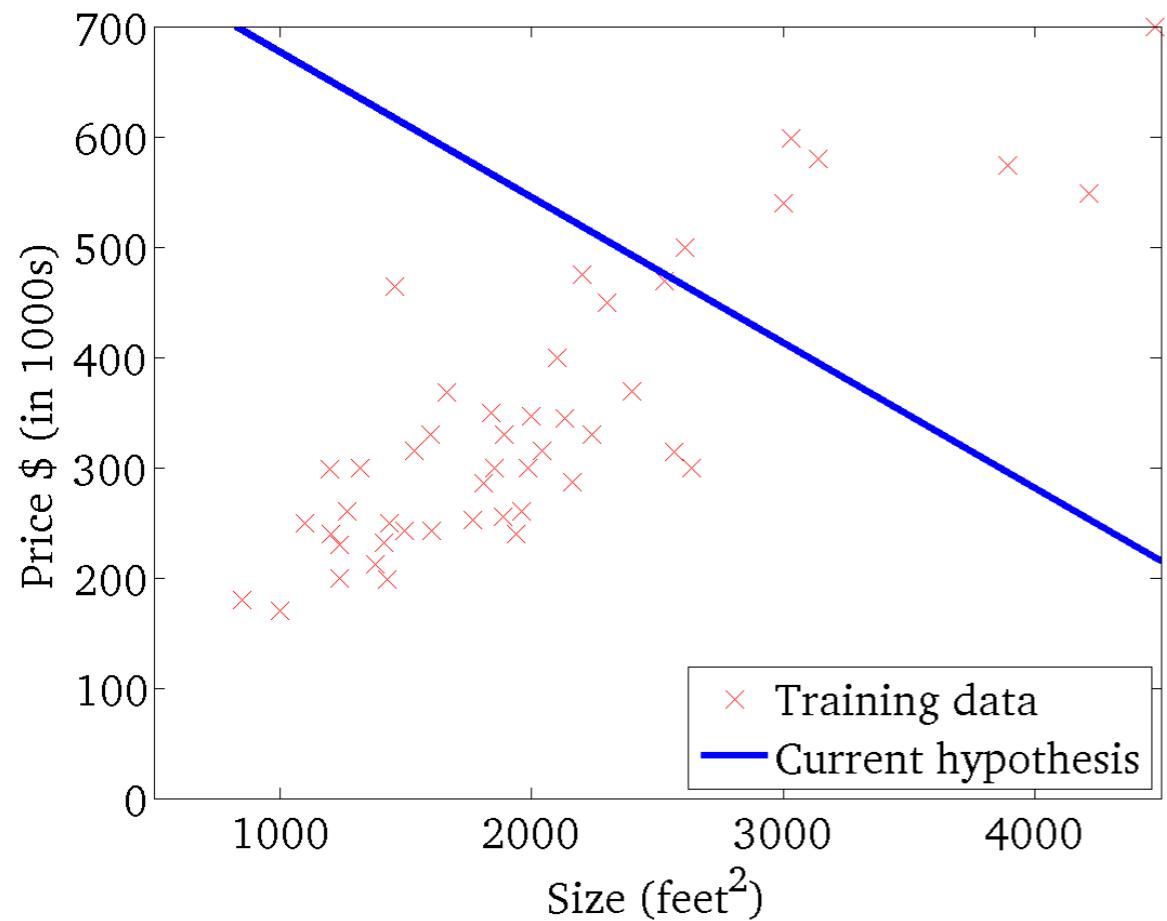
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



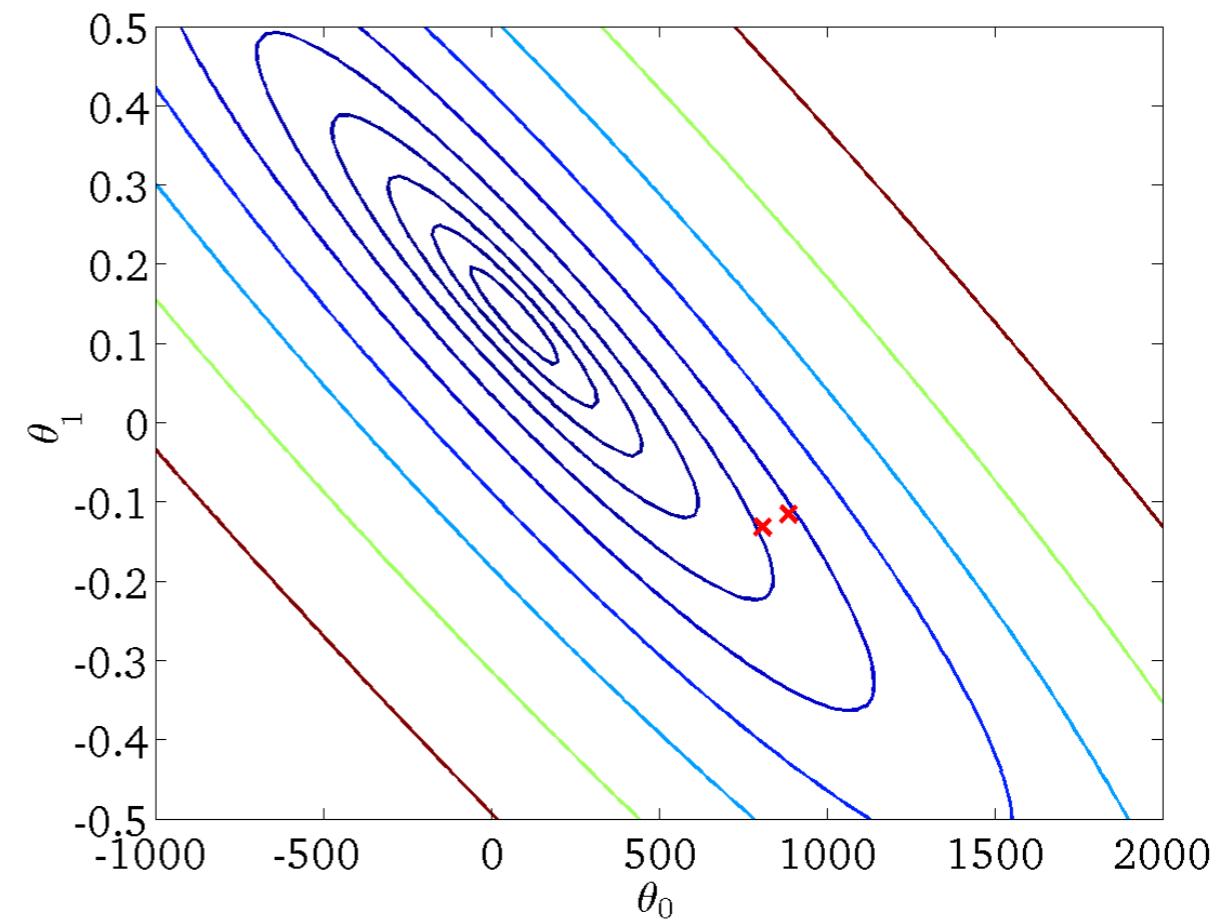
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



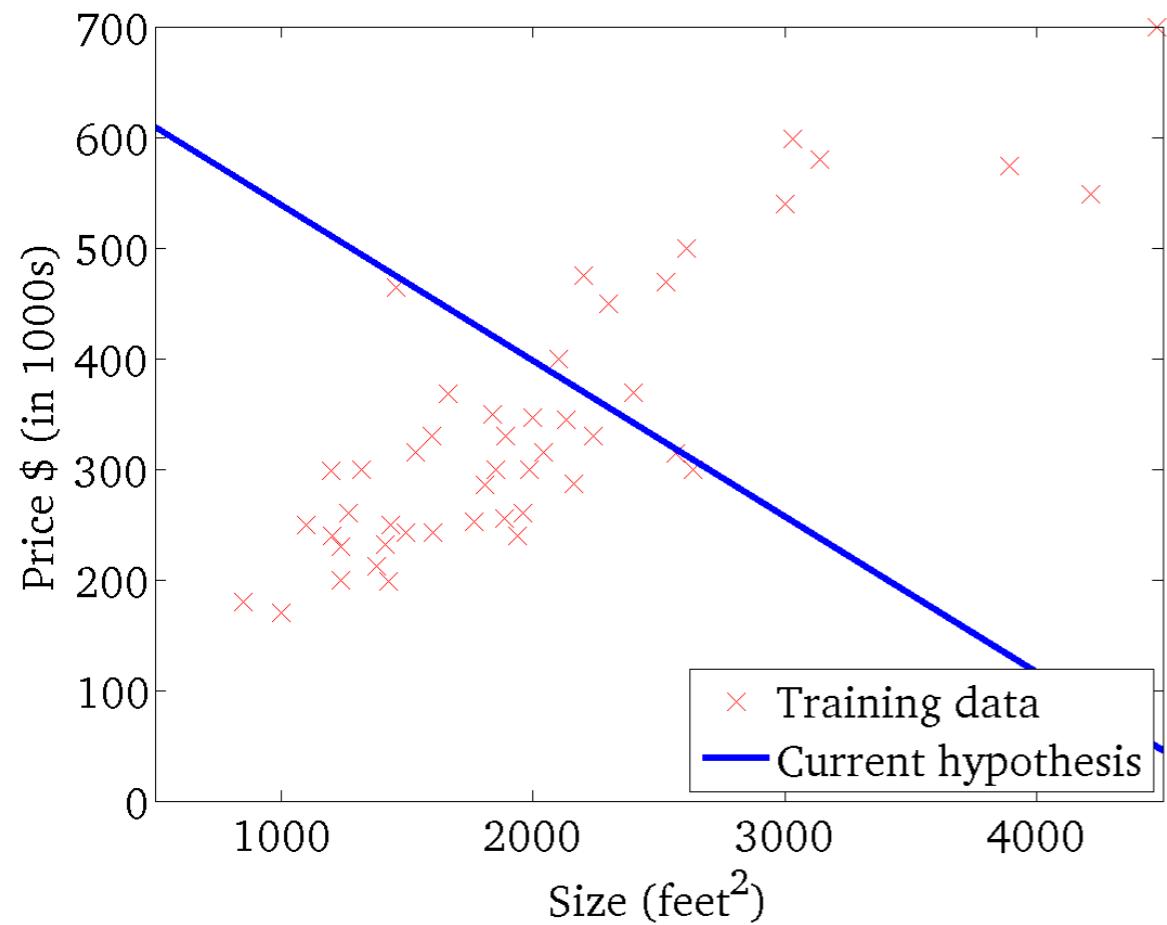
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



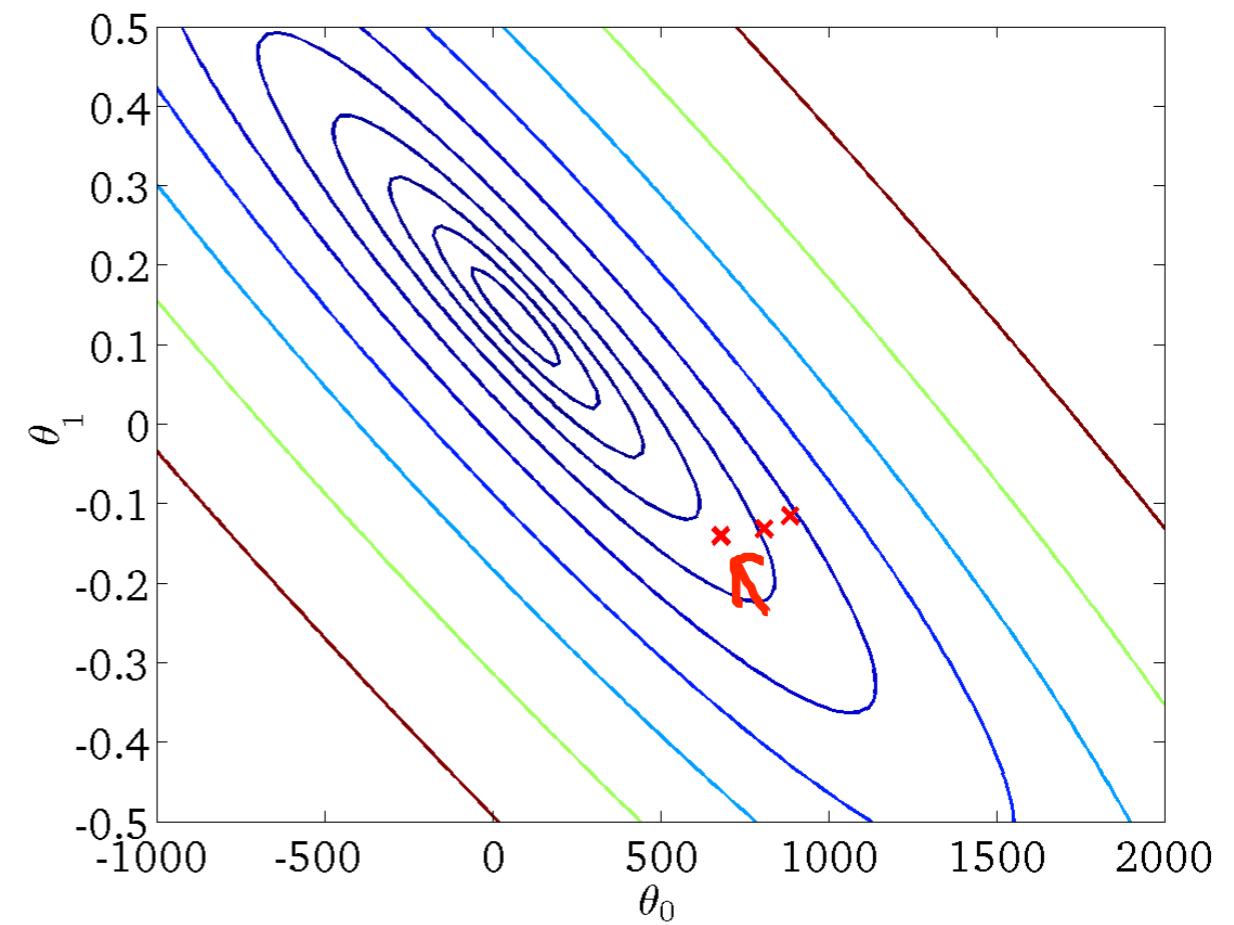
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



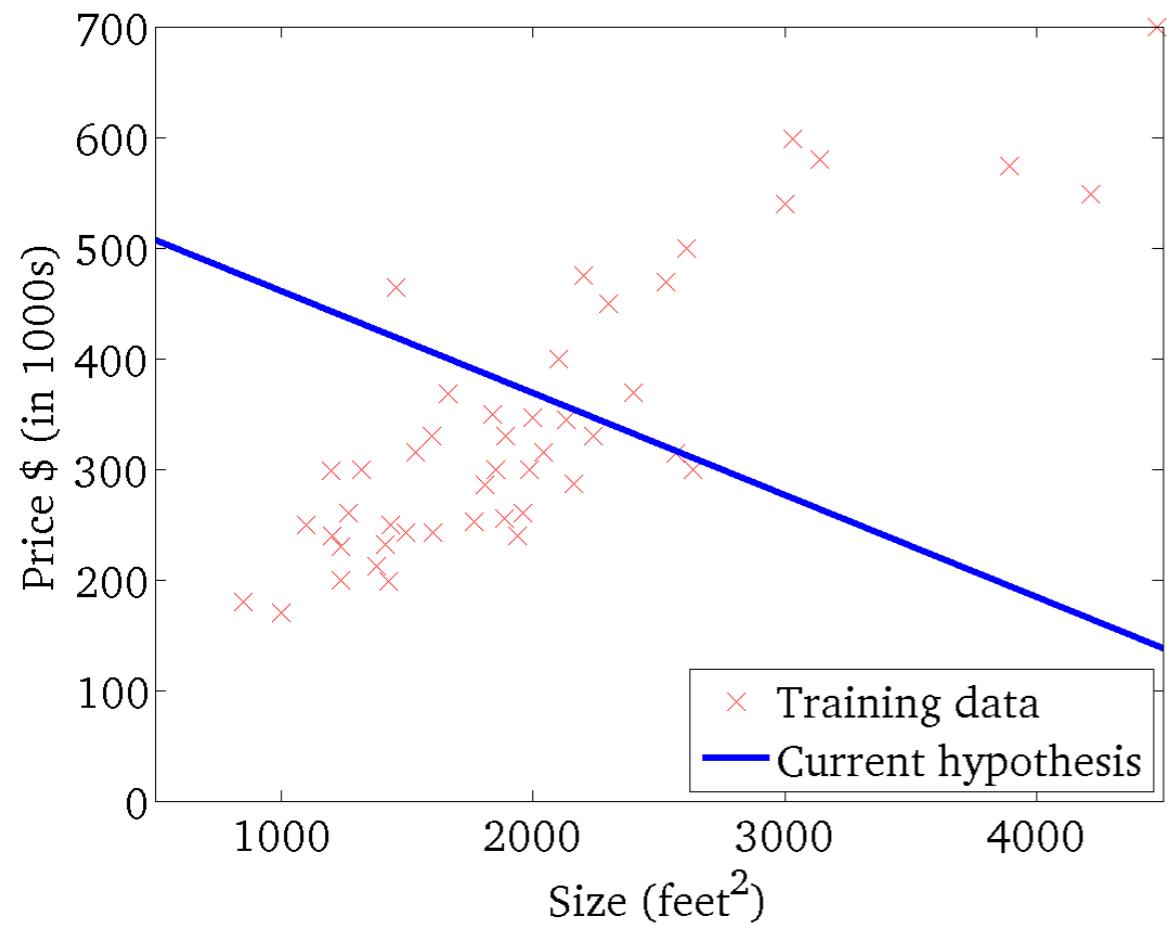
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



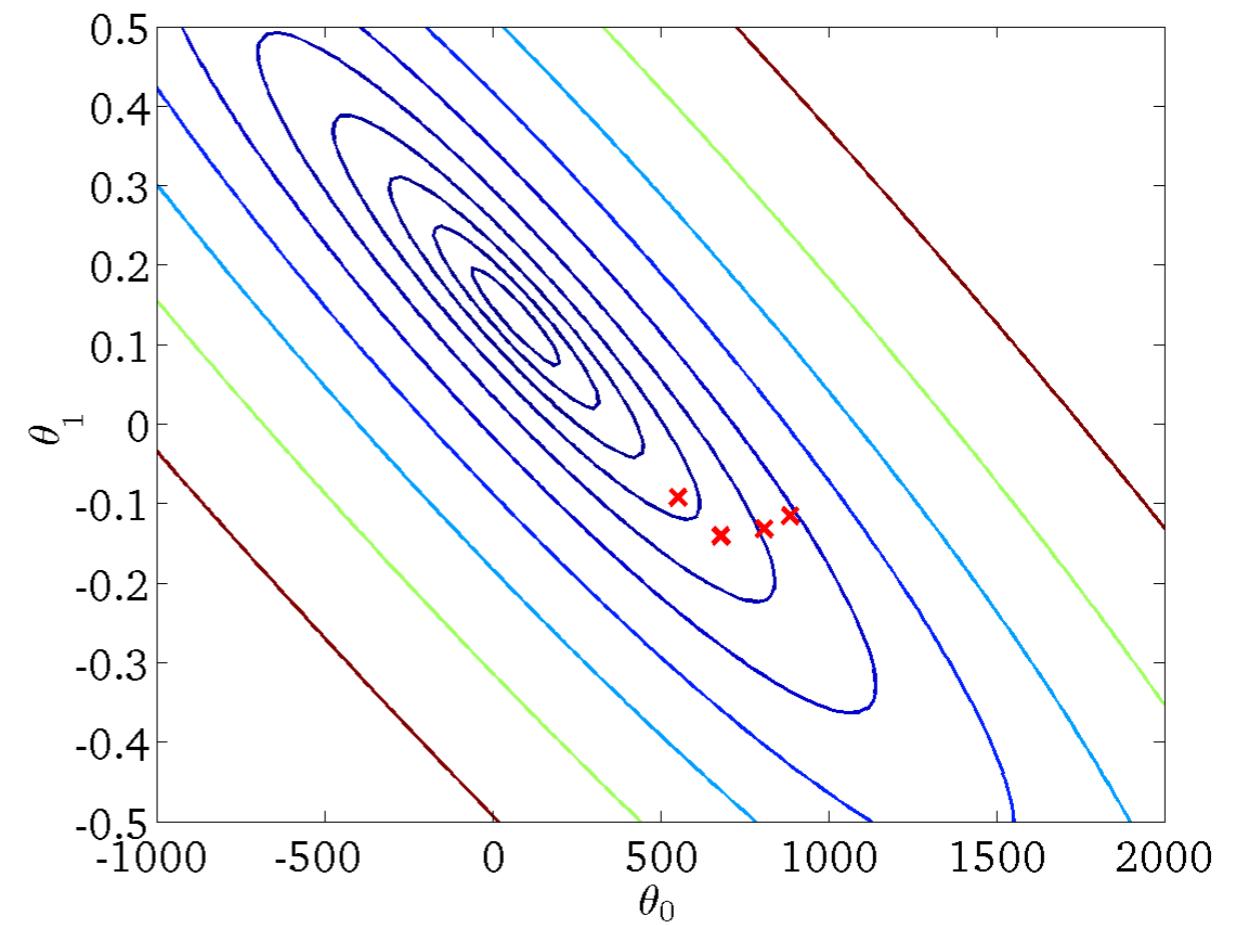
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



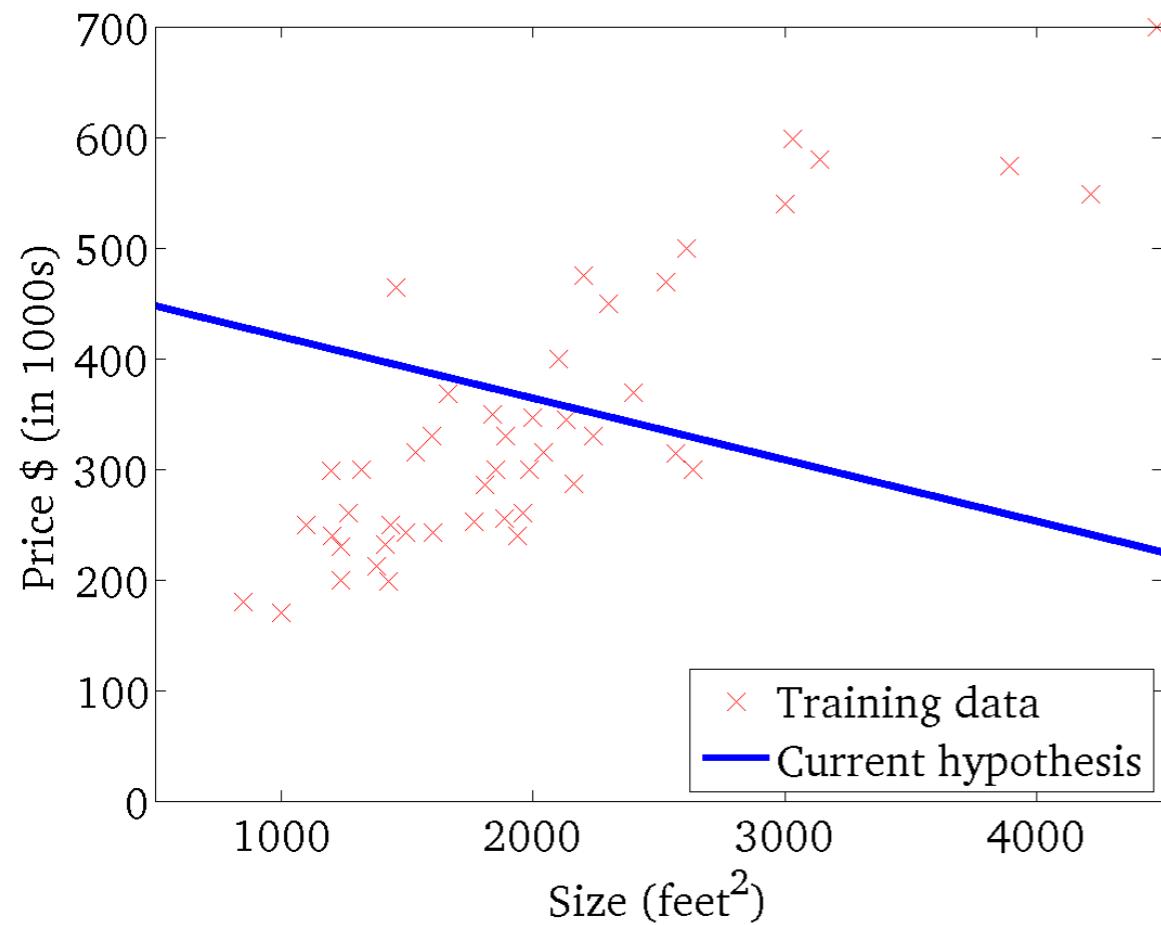
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



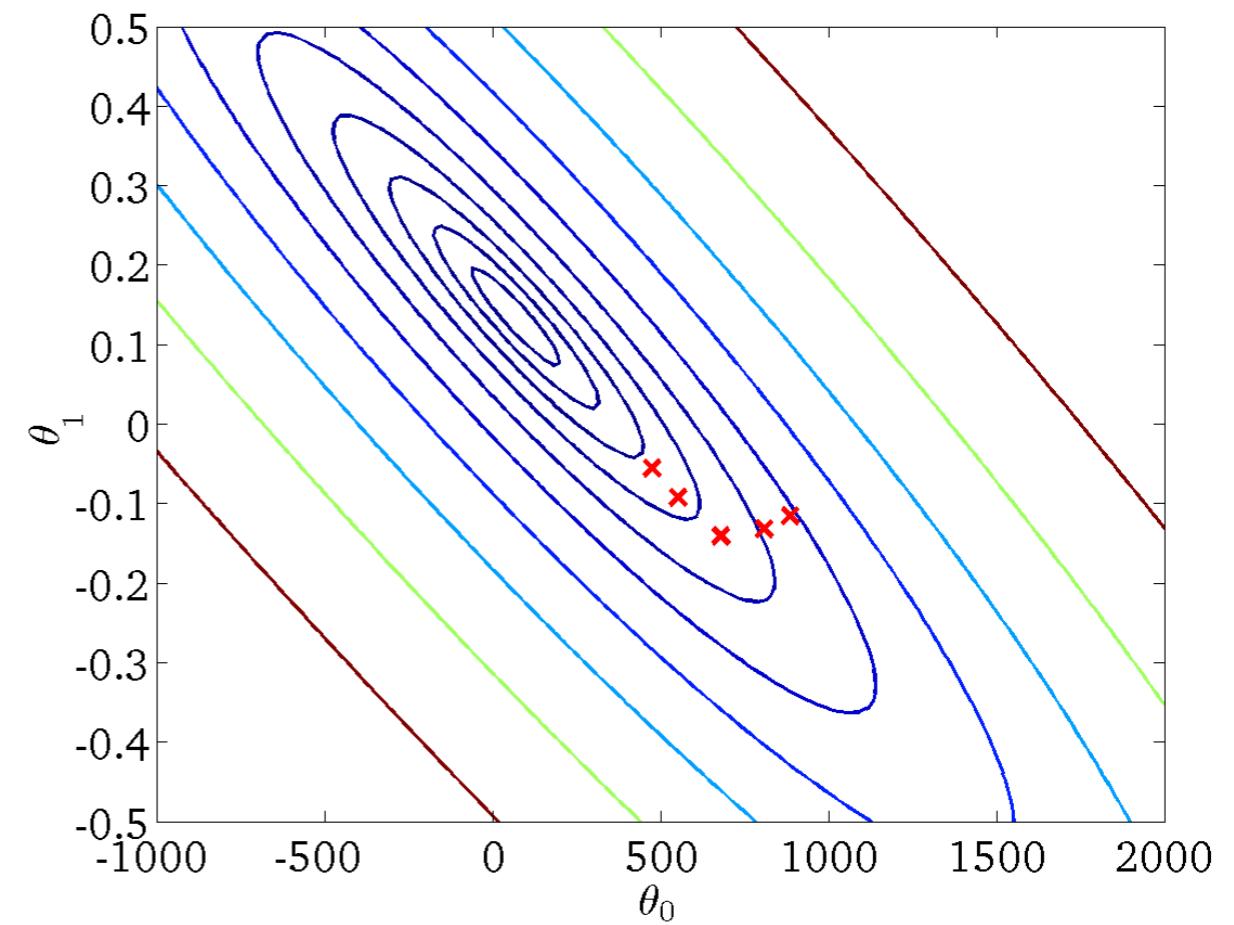
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



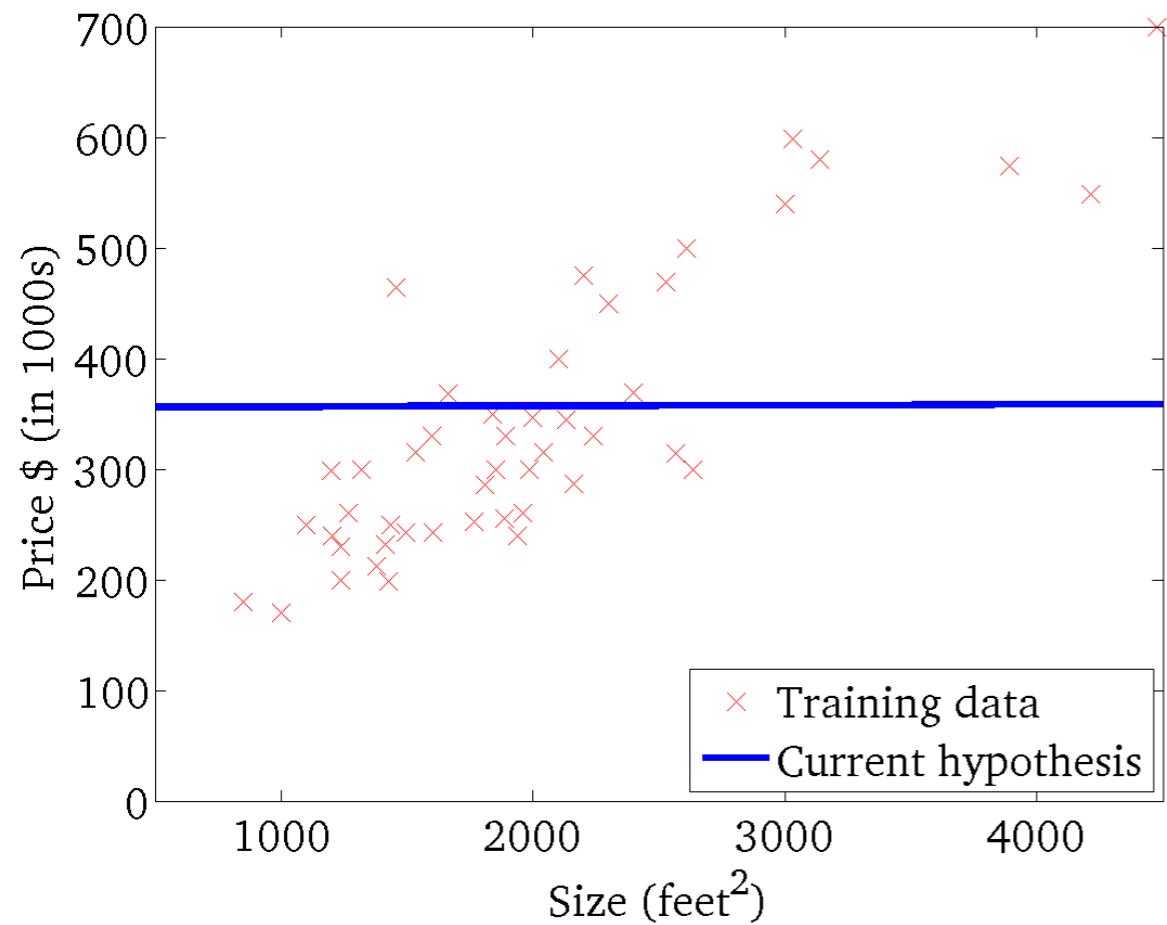
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



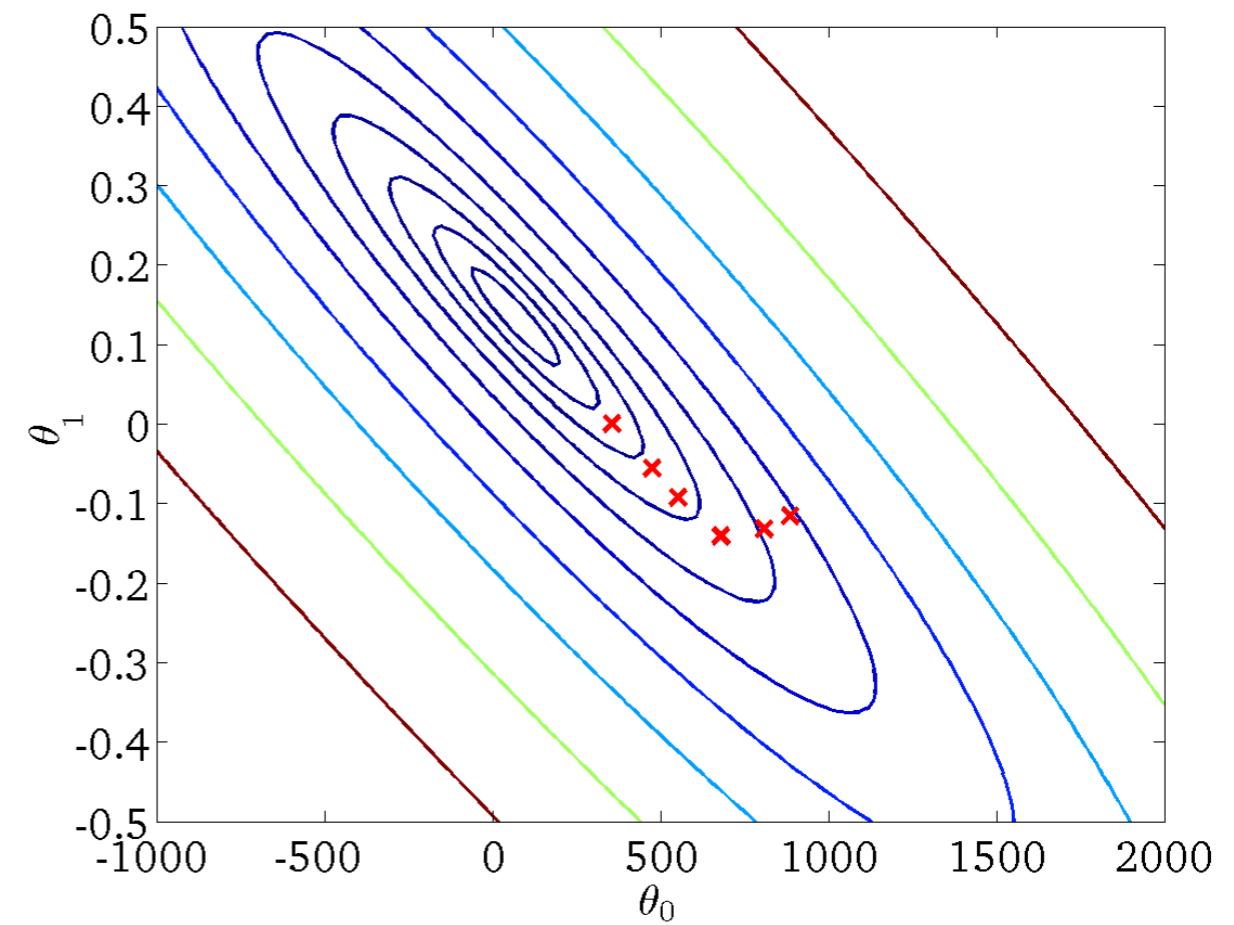
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



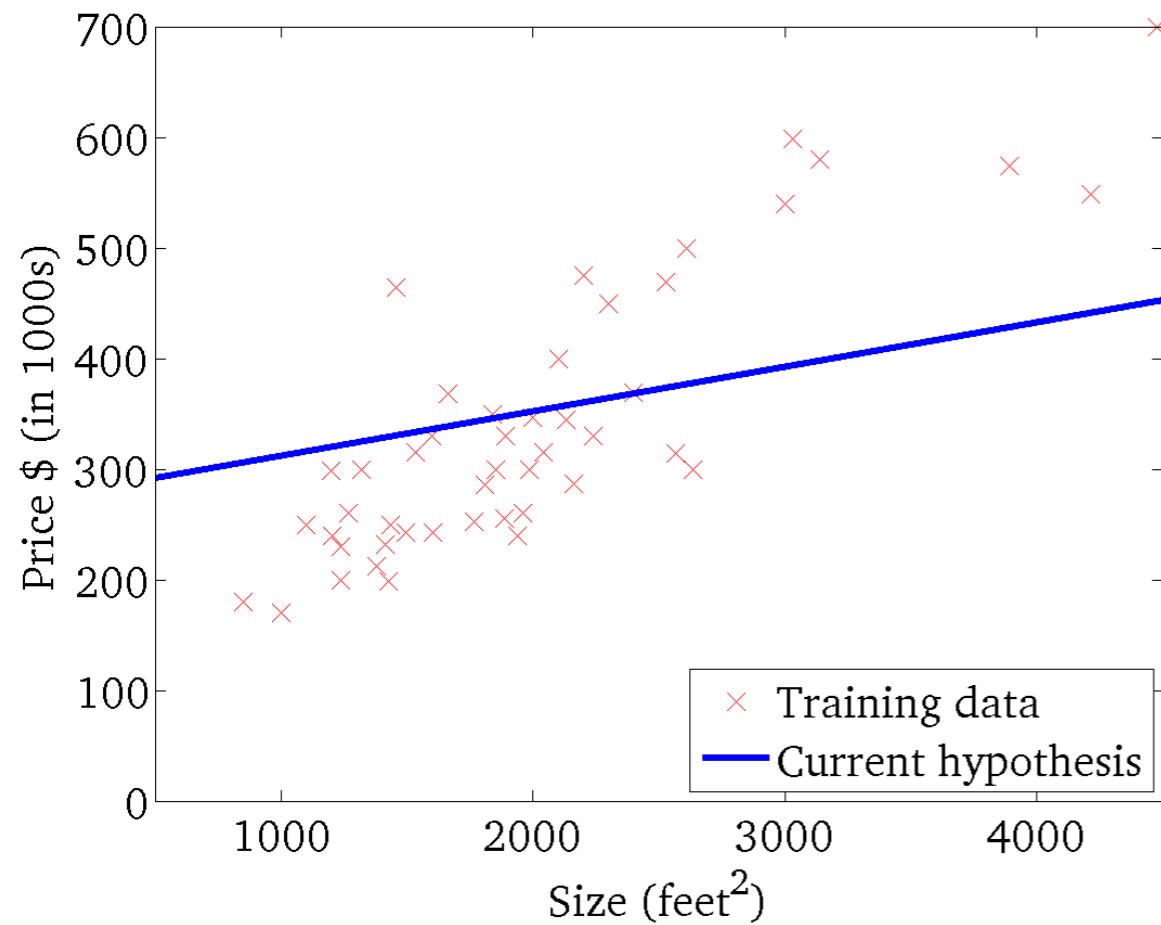
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



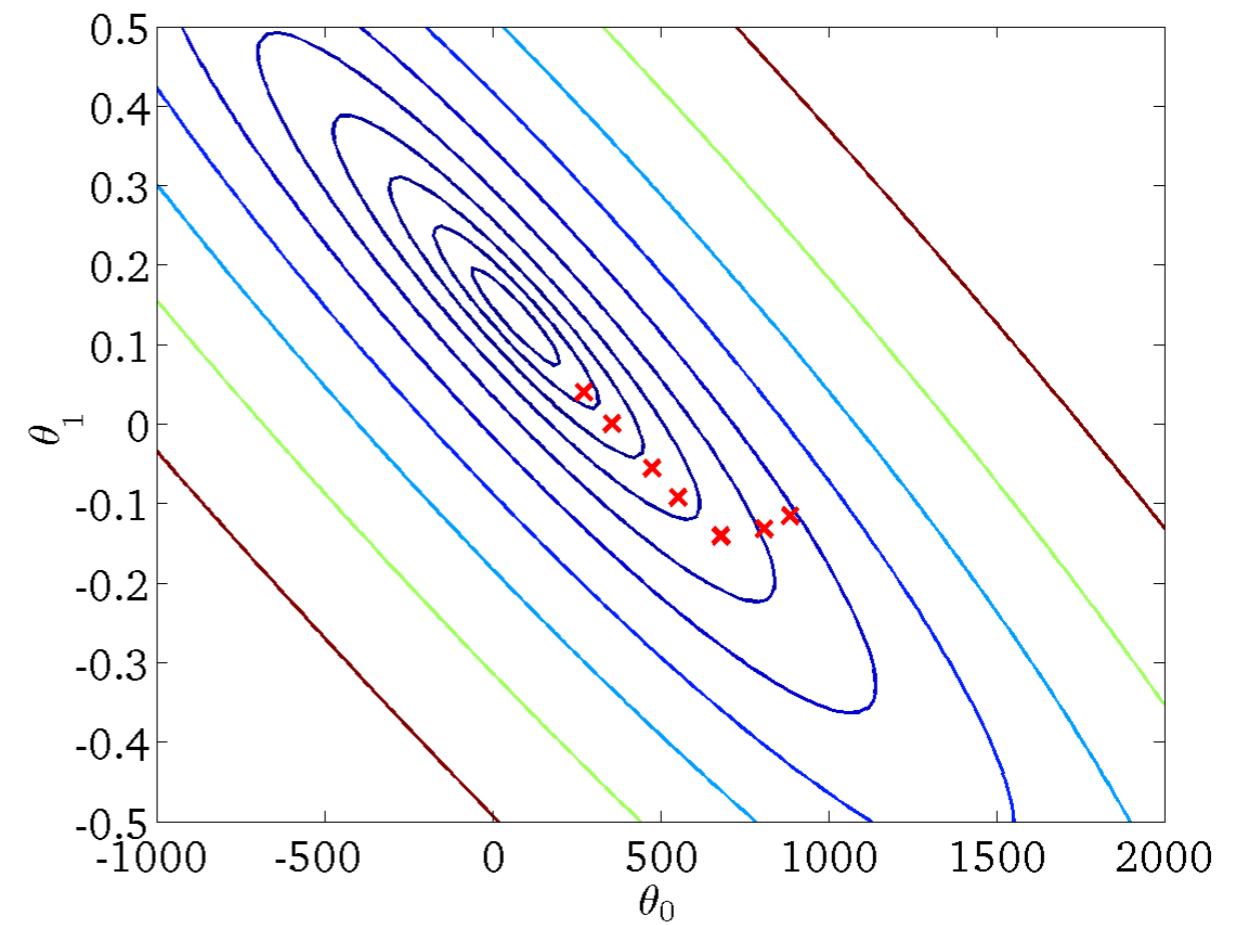
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



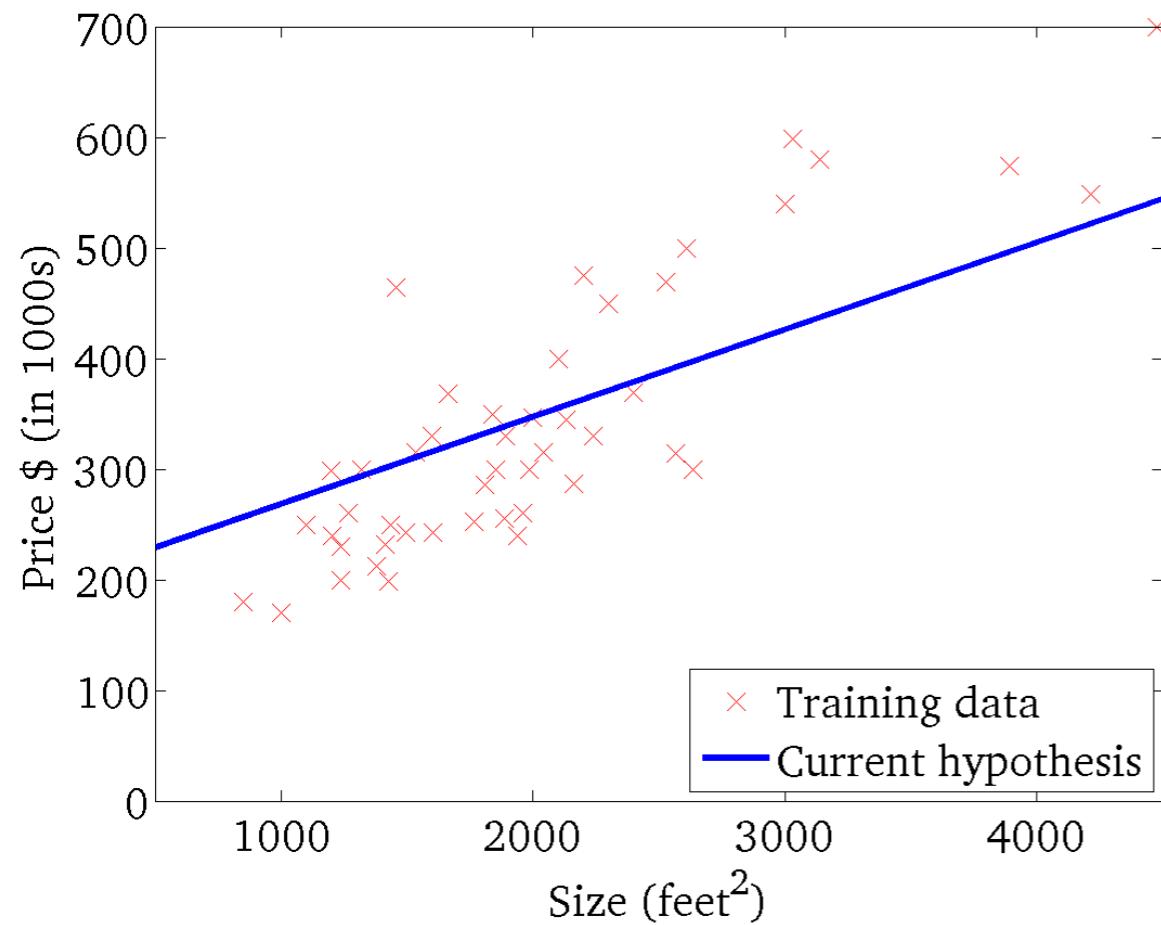
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



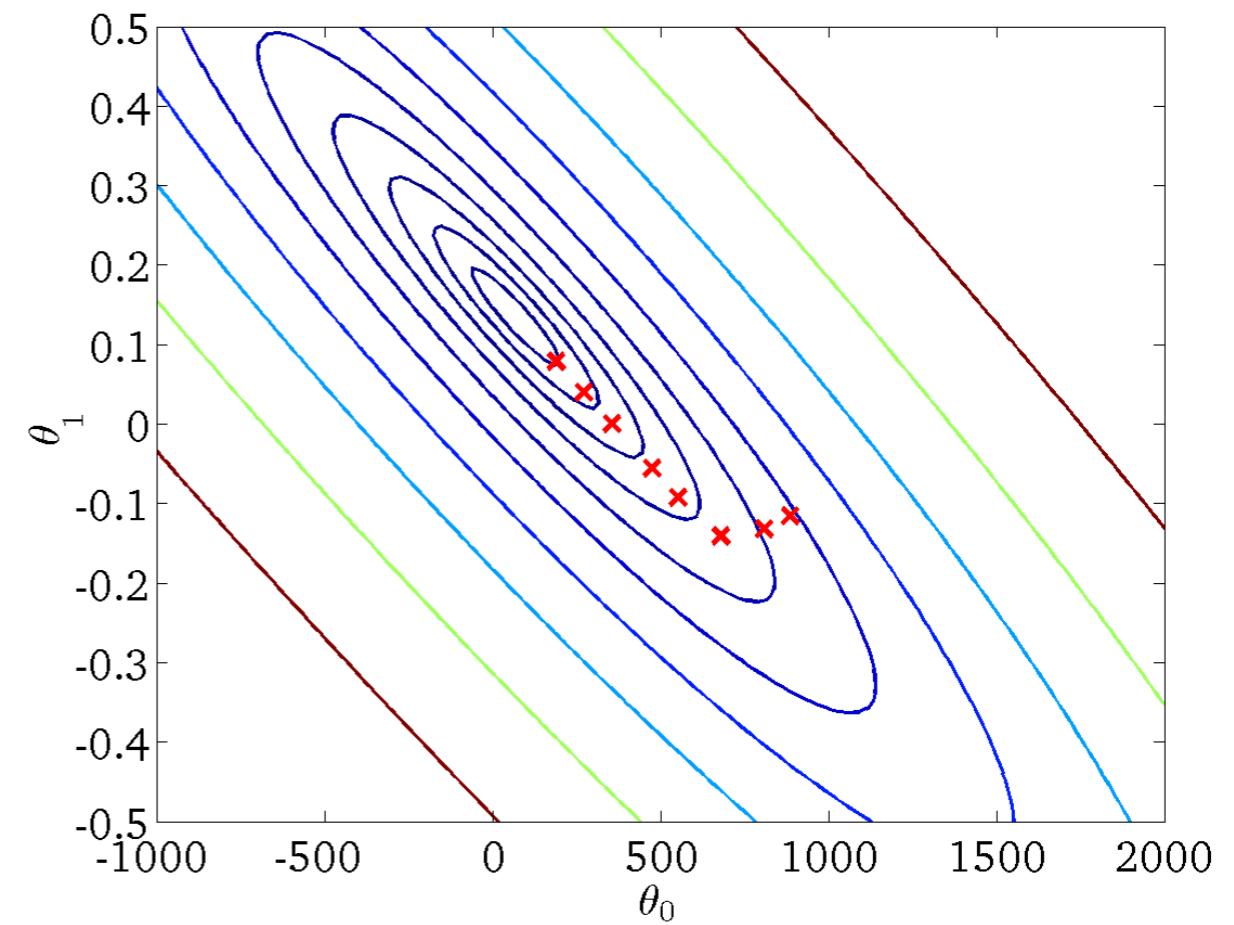
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



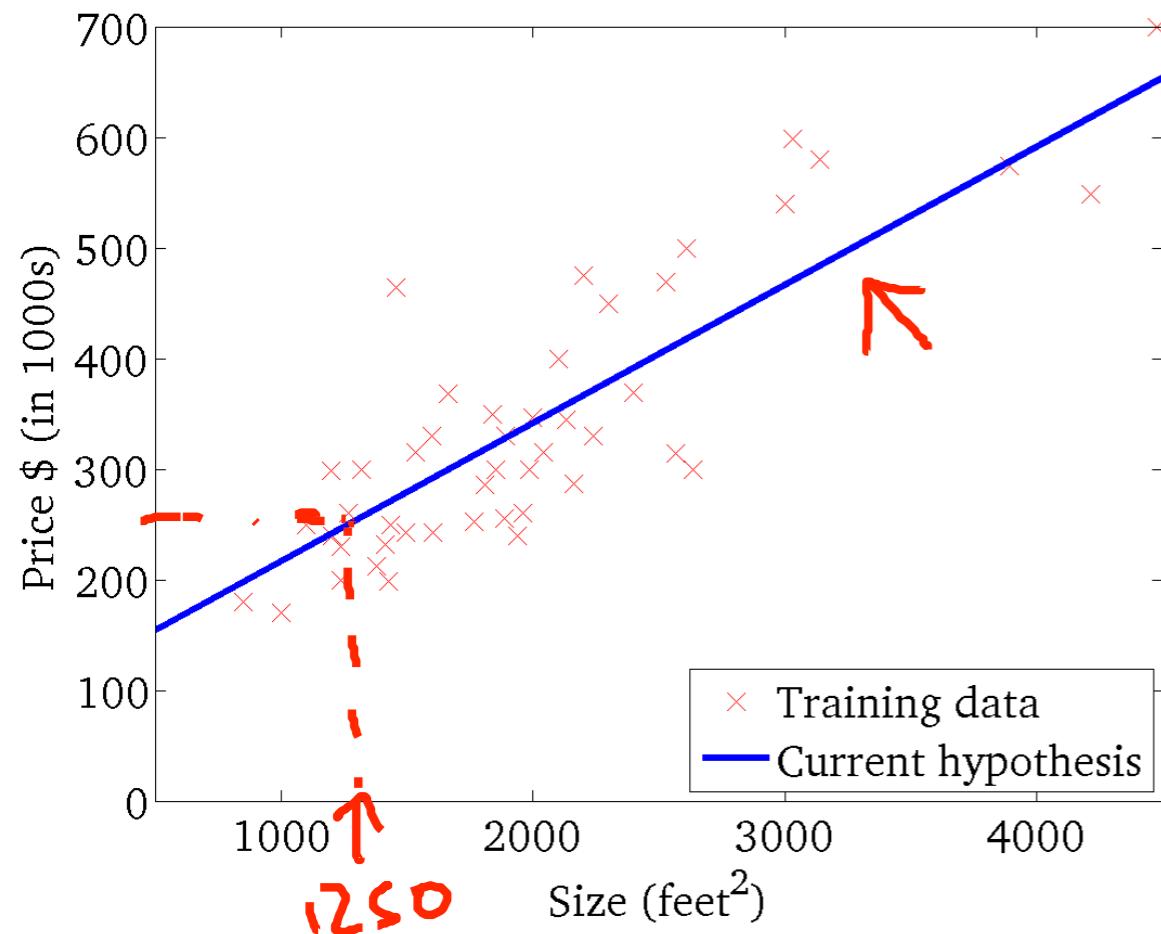
$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )



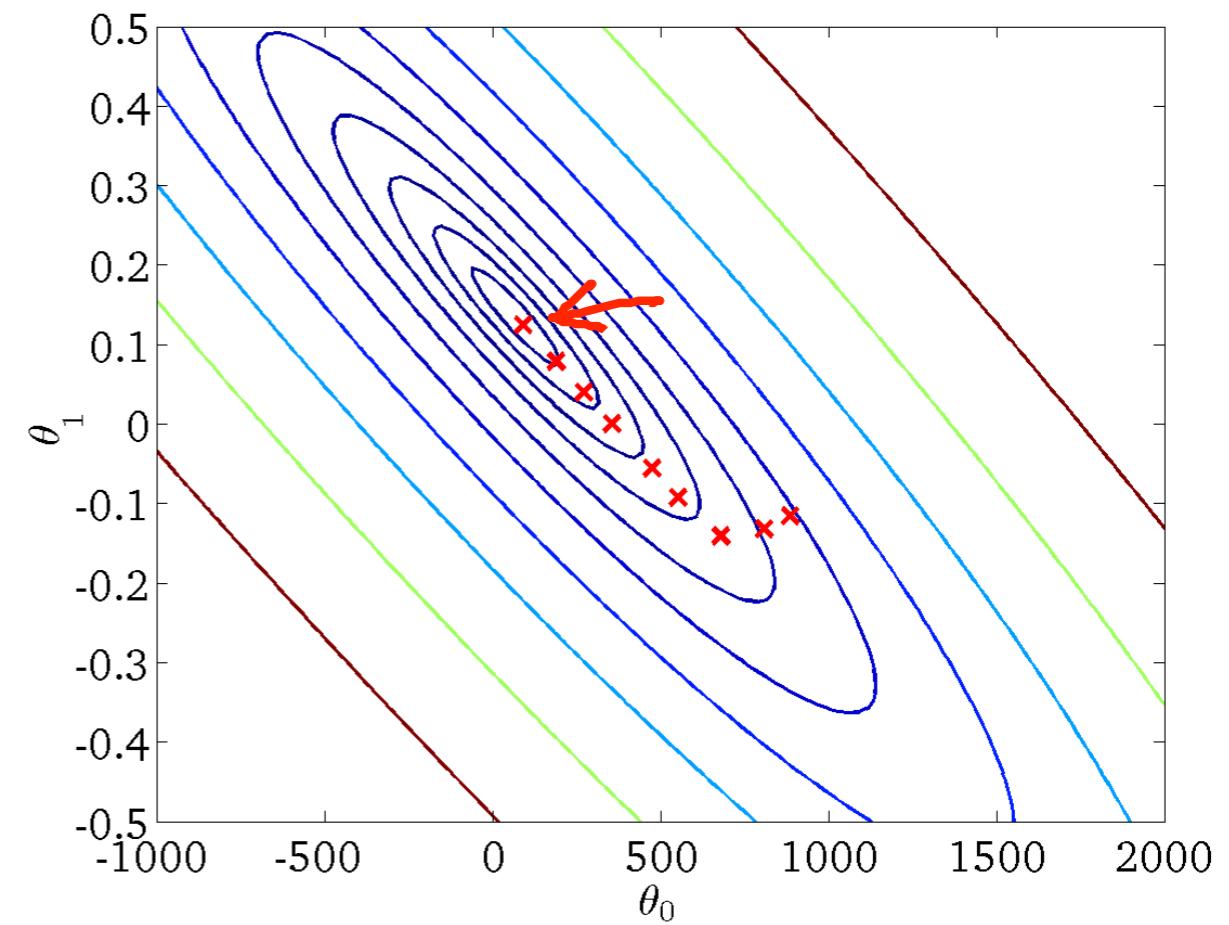
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

(function of the parameter  $\theta_0, \theta_1$ )

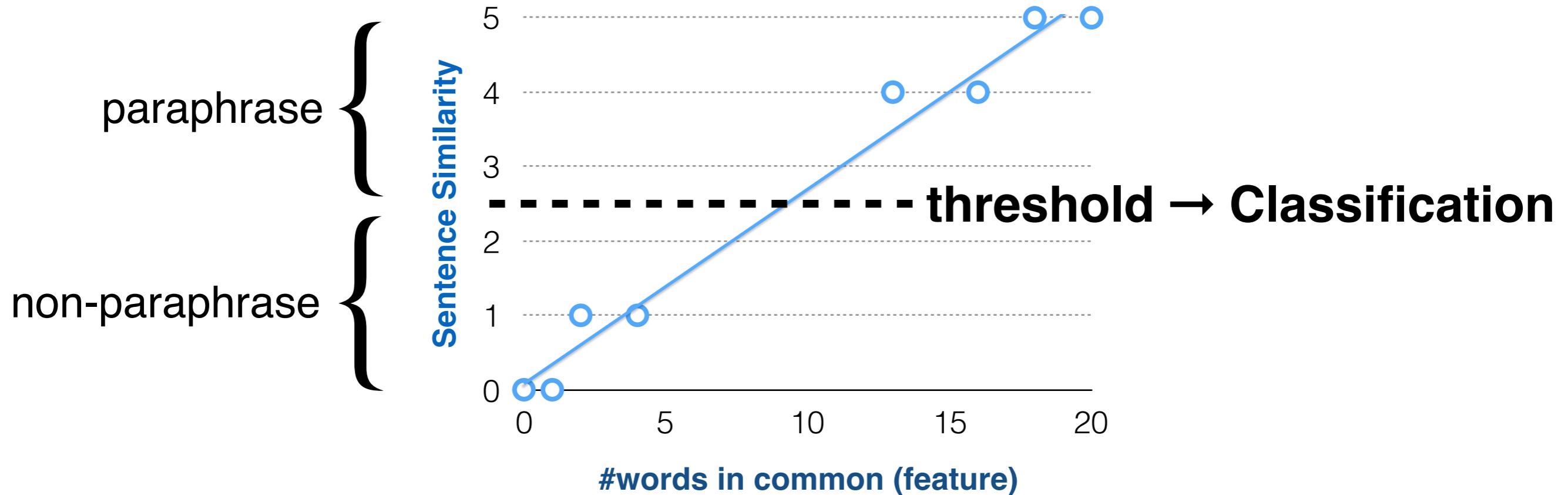


# Batch Update

- Each step of gradient descent uses all the training examples

(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:**

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

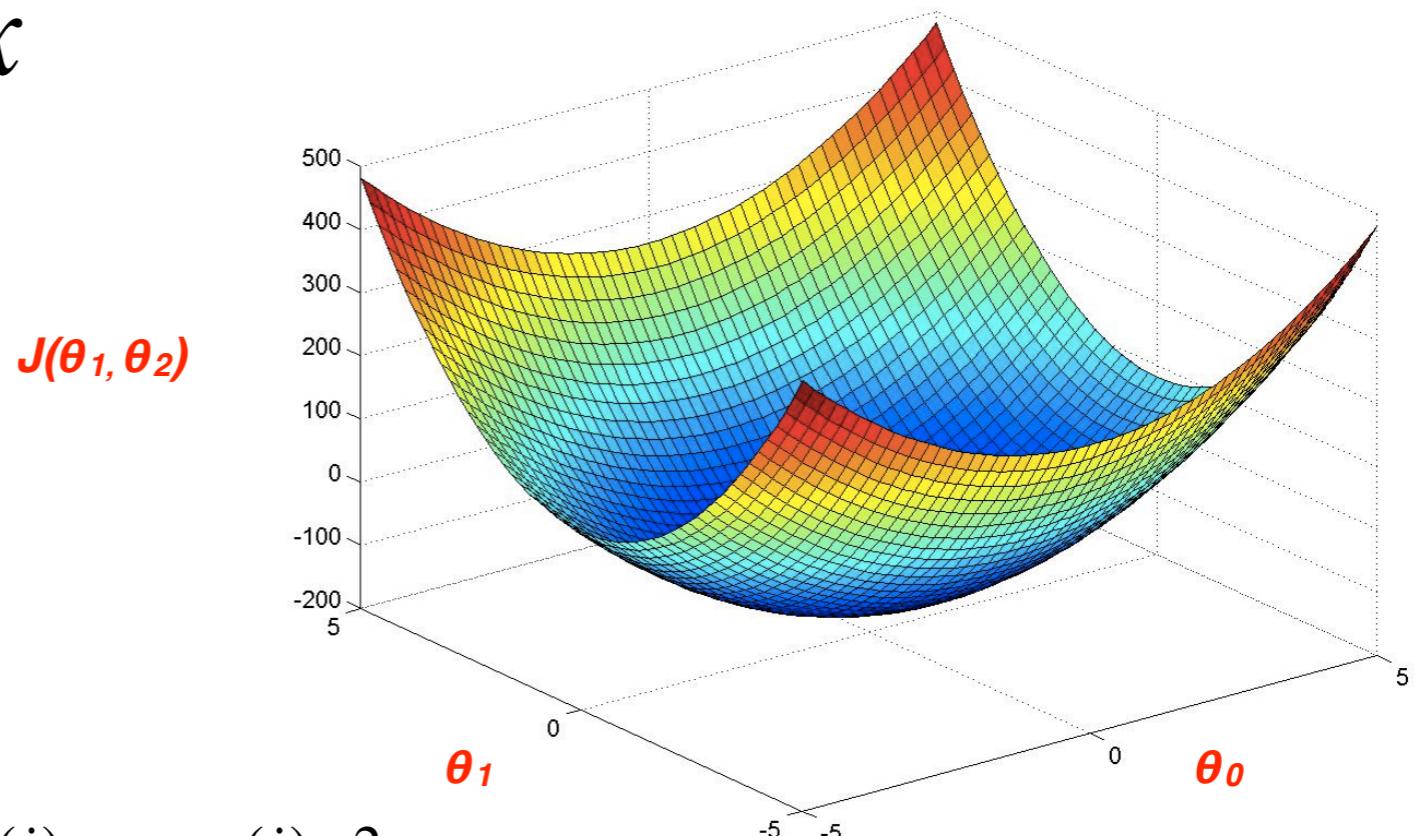
- **Parameters:**

$$\theta_0, \theta_1$$

- **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)

# 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)

# NLP Talk on Friday



- **Jeniya Tabassum (OSU)**
- **Friday**, September 30, 3:00 pm, Dreese 480
- Resolving Time Expressions in Twitter

[socialmedia-class.org](http://socialmedia-class.org)

# Next Class:

- Logistic Regression
- Homework #2 will release
- David Palzer: NLP@CMU

[socialmedia-class.org](http://socialmedia-class.org)