Social Media & Text Analysis

lecture 5 - Paraphrase Identification and Logistic Regression

Follow @cocoweixu

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

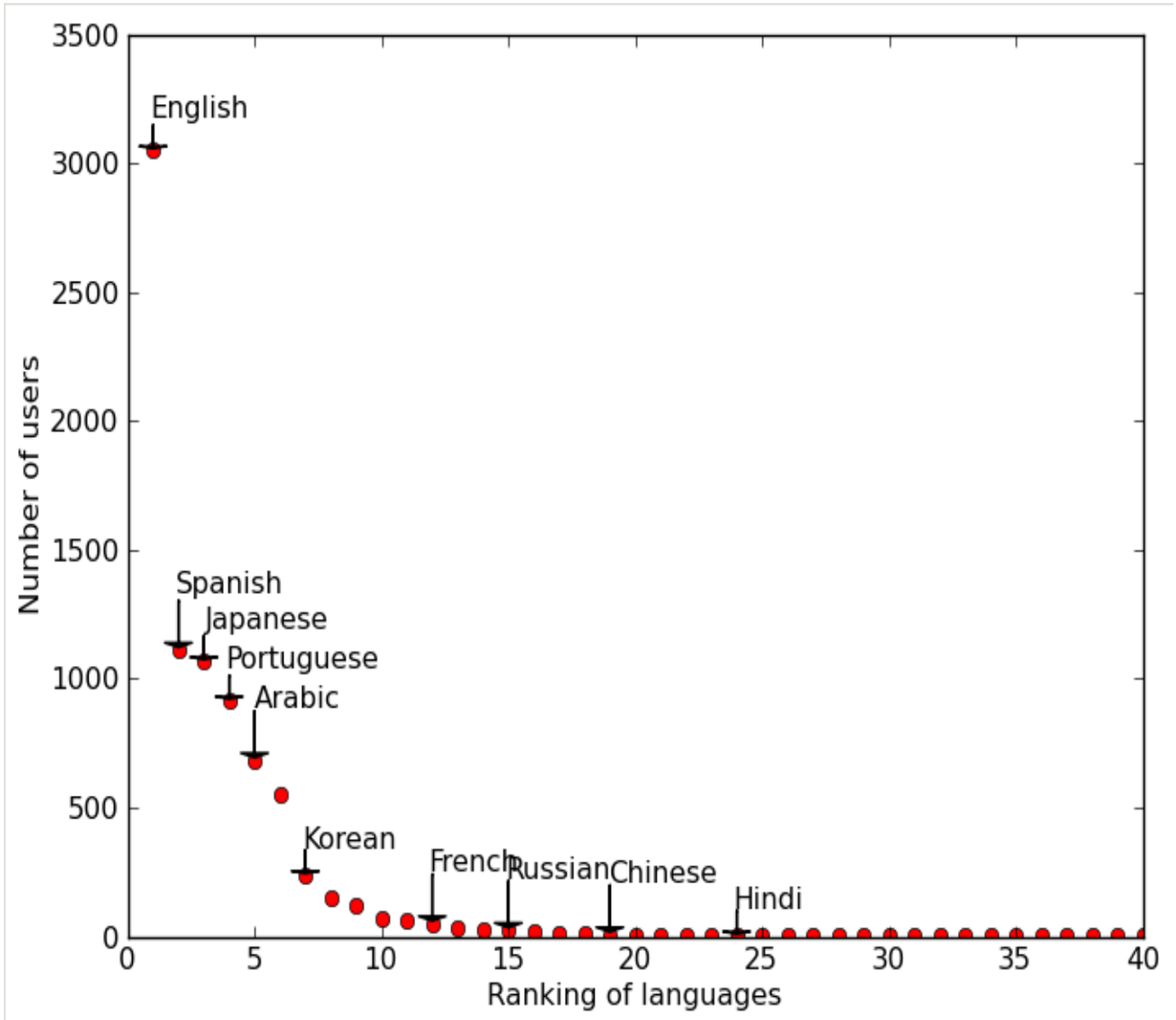
Homework #1

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

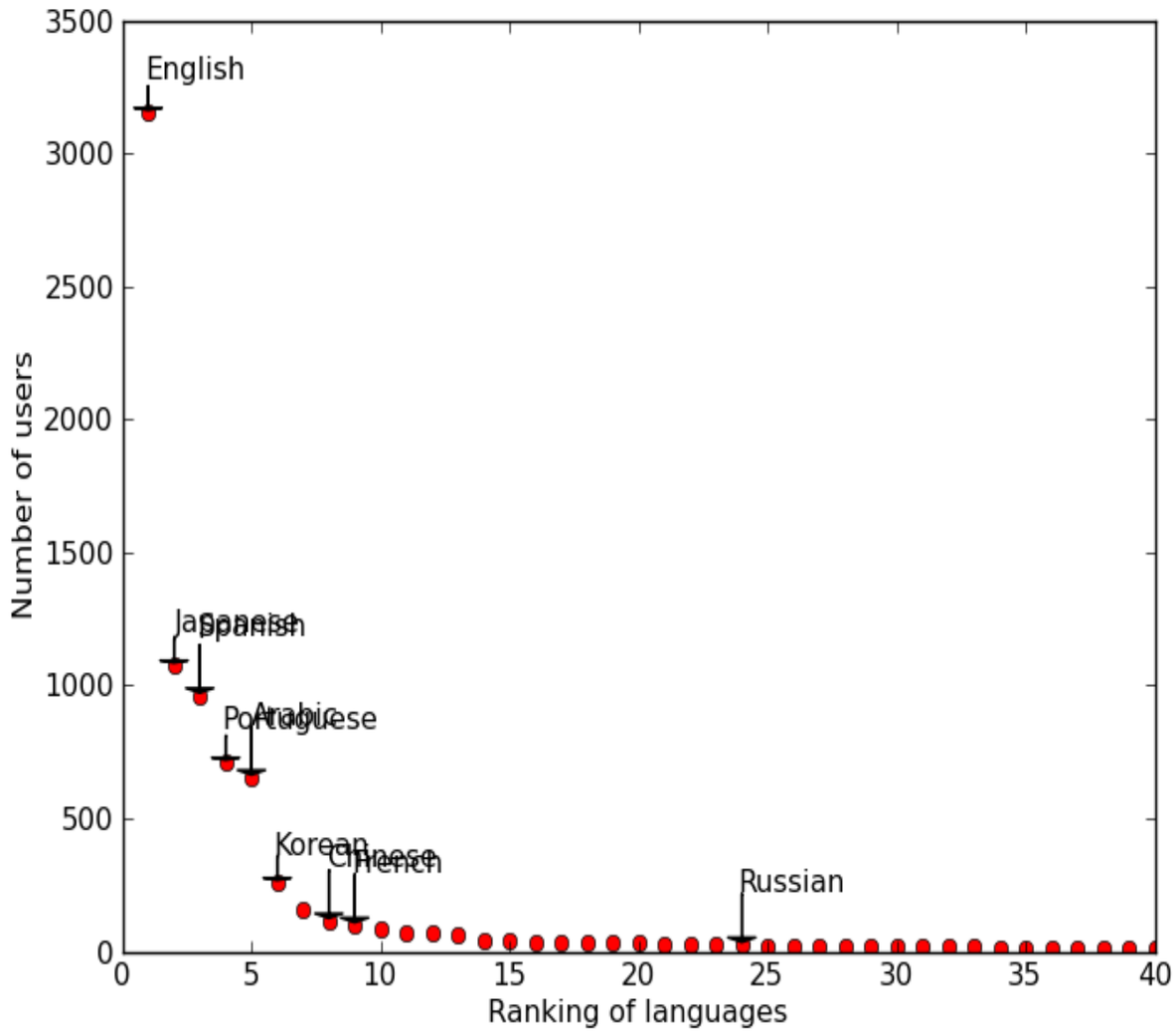
Solution:

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

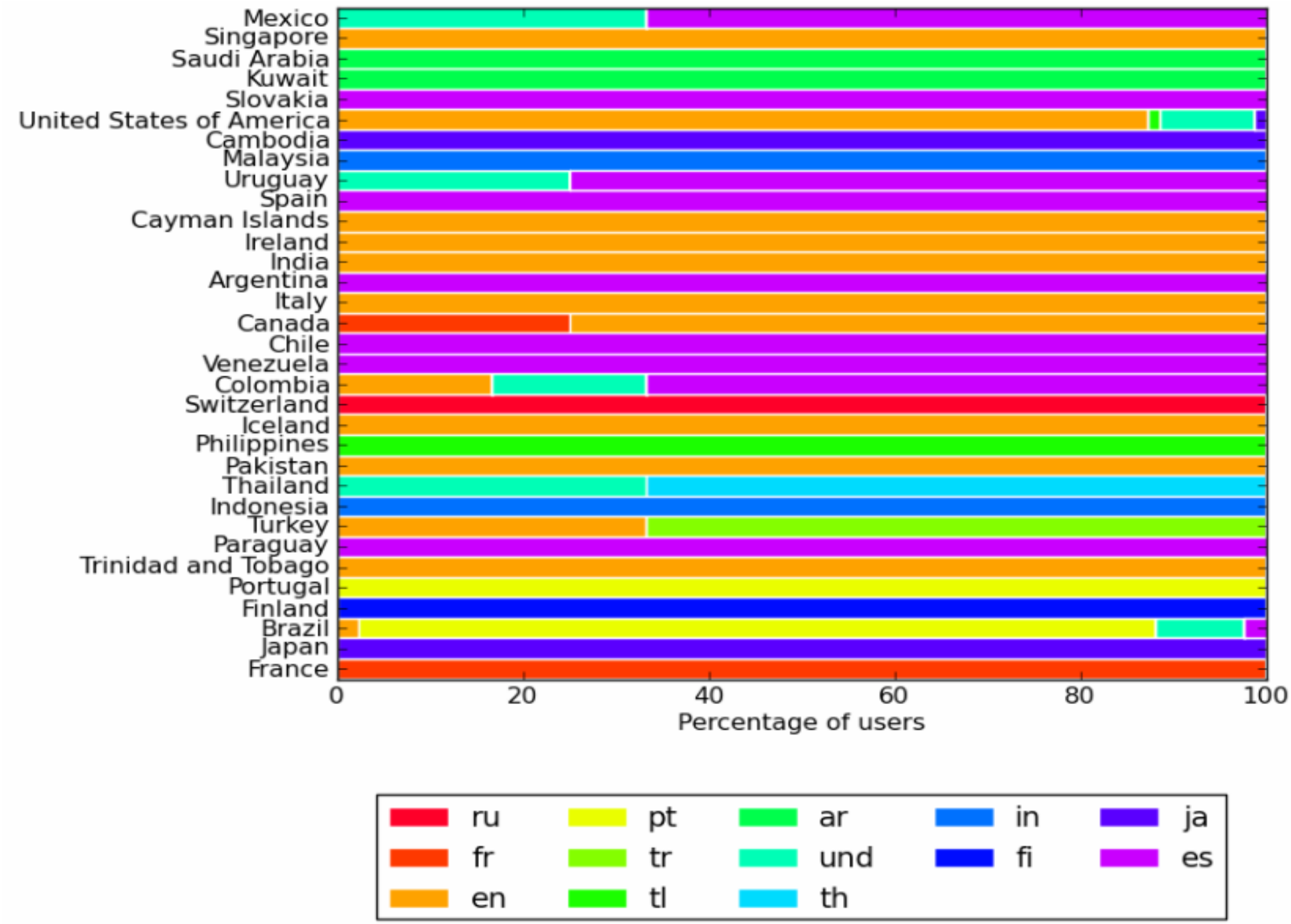


Twitter API



langid.py

Homework #1


Mexico
Singapore
Saudi Arabia
Kuwait
Slovakia
United States of America
Cambodia
Malaysia
Uruguay
Spain
Cayman Islands
Ireland
India
Argentina
Italy
Canada
Chile
Venezuela
Colombia
Switzerland
Iceland
Philippines
Pakistan
Thailand
Indonesia
Turkey
Paraguay
Trinidad and Tobago
Portugal
Finland
Brazil
Japan
France
0
20
40
60
80
100
Percentage of users
ru
fr
en
pt
tr
tl
ar
und
th
in
fi
ja
es

what is Paraphrase?

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

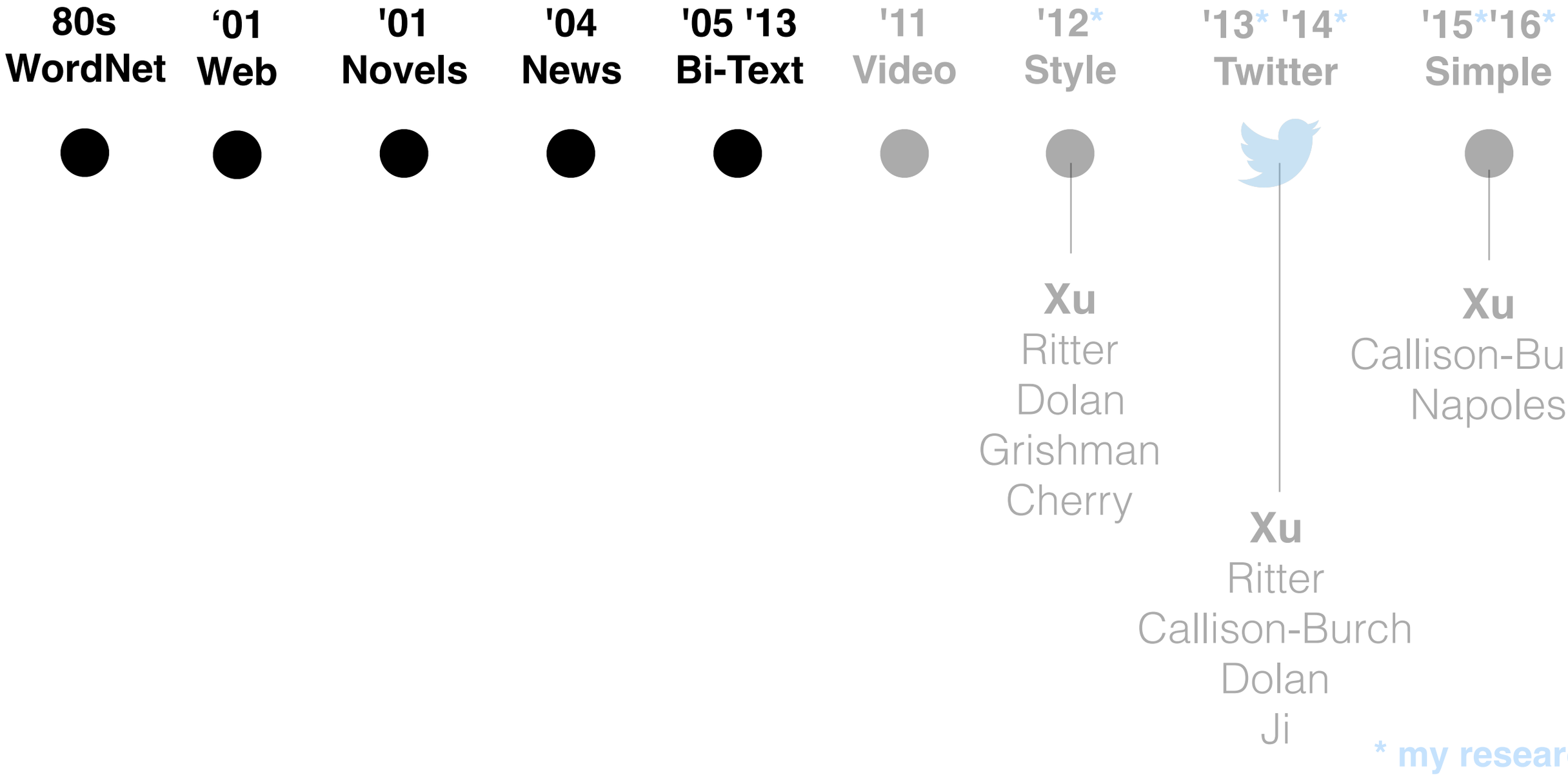
wealthy	**word**	*rich*
the king's speech	**phrase**	*His Majesty's address*
… the forced resignation of the CEO of Boeing, Harry Stonecipher, for …	**sentence**	*… after Boeing Co. Chief Executive Harry Stonecipher was ousted from …*

The Ideal

Translation: "You have a bruised rib."

(Review Last Week)

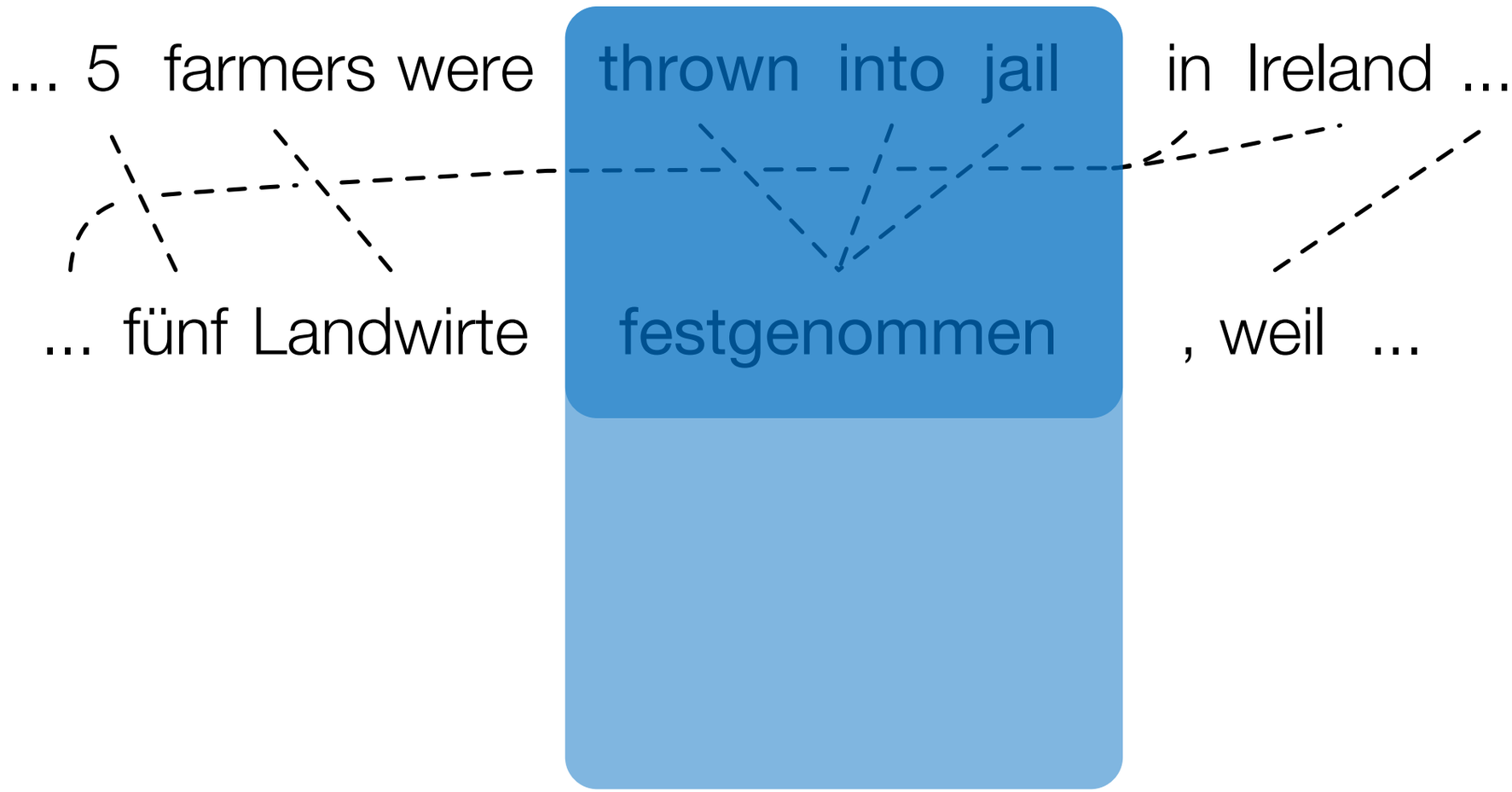
Paraphrase Research



Wei Xu. "Data-driven Approaches for Paraphrasing Across Language Variations" PhD Thesis (2014)

Bilingual Pivoting

word alignment



Source: Chris Callison-Burch

Distributional Similarity

Idea: similar words occur in similar contexts.

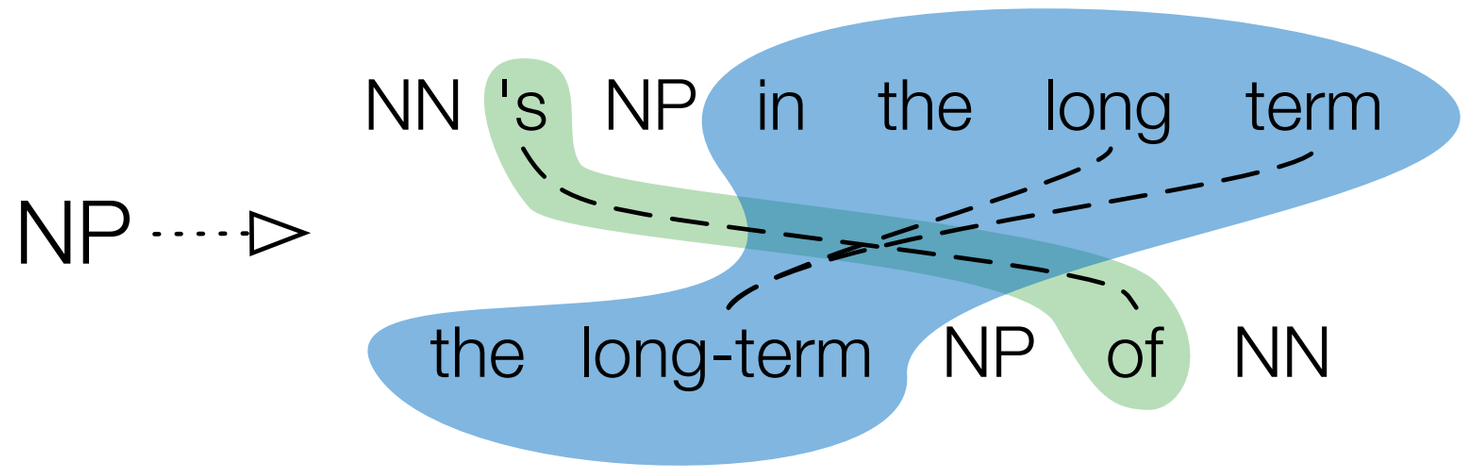
Characterize words by their contexts

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

"Are these paraphrases substitutable?"

Source: Chris Callison-Burch

Paraphrase Similarity

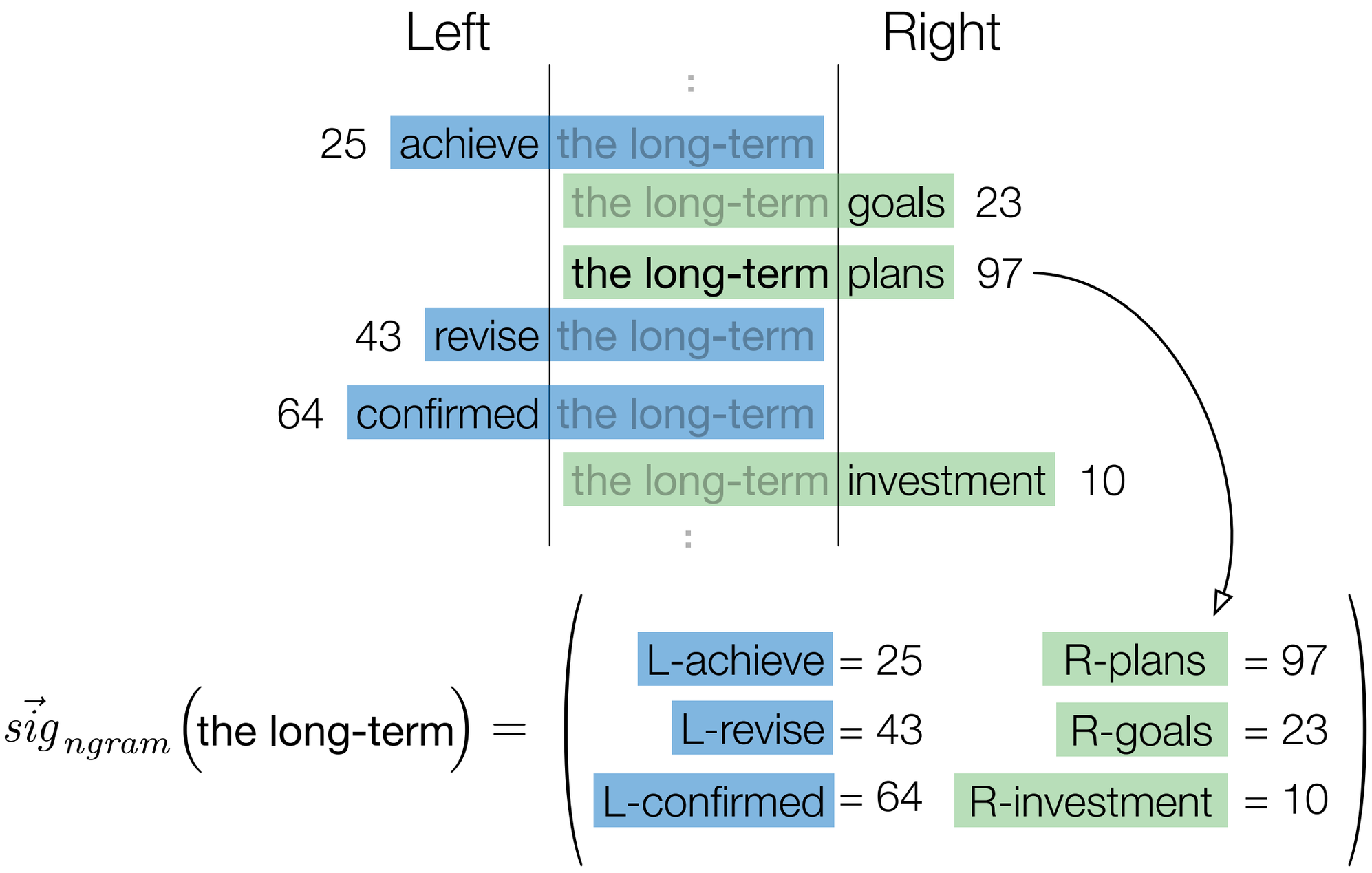


$$sim(\mathbf{r}) = \frac{1}{2}\left(sim\begin{pmatrix}\text{the long-term}\\ \text{in the long term}\end{pmatrix} + sim\begin{pmatrix}\text{'s}\\ \text{of}\end{pmatrix}\right)$$

Source: Chris Callison-Burch

Juri Ganitkevitch, Ben Van Durme and Chris Callison-Burch. Monolingual Distributional Similarity for Text-to-Text Generation. *SEM 2012.

n-gram Context

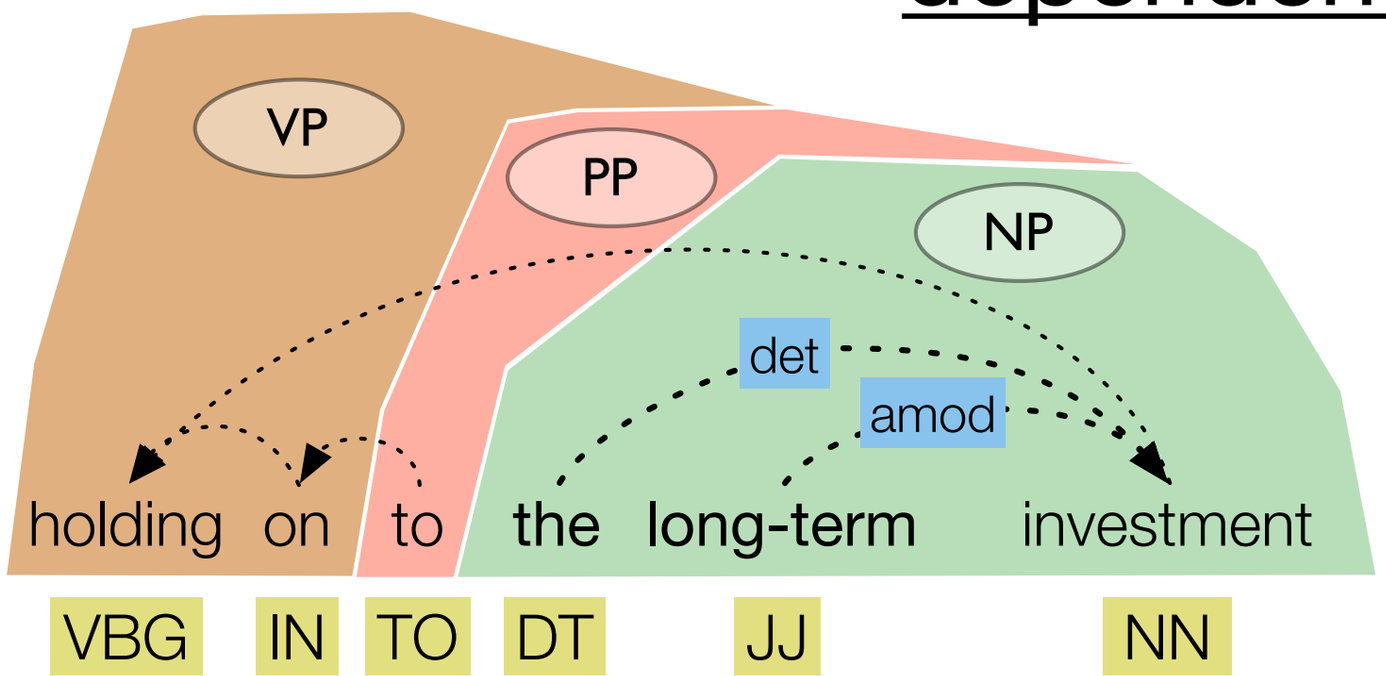


Source: Chris Callison-Burch

Juri Ganitkevitch, Ben Van Durme and Chris Callison-Burch. PPDB: The Paraphrase Database. NAACL 2013.

Syntactic Context

dependency parsing



$$\vec{sig}_{syntax}\left(\text{the long-term}\right) = \left(\begin{array}{l} \text{lex-R-investment} \quad \text{lex-L-on-to} \\ \text{pos-L-IN-TO} \quad \text{pos-L-TO} \quad \text{lex-L-to} \\ \text{dep-det-R-investment} \quad \text{pos-R-NN} \\ \text{dep-amod-R-investment} \\ \text{dep-det-R-NN} \quad \text{dep-amod-R-NN} \\ \text{syn-gov-NP} \quad \text{syn-miss-L-NN} \end{array}\right)$$

Source: Chris Callison-Burch

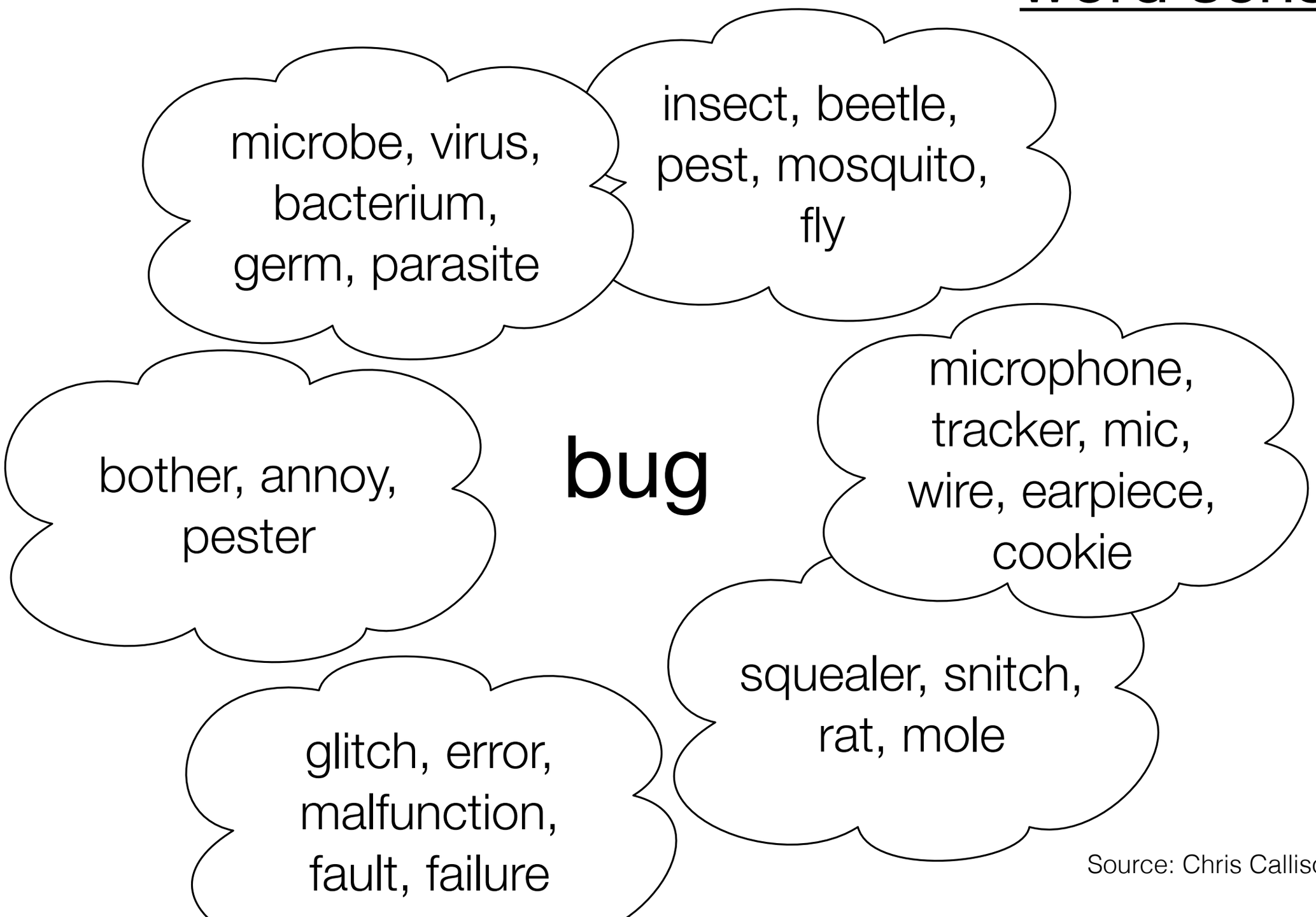
Juri Ganitkevitch, Ben Van Durme and Chris Callison-Burch. PPDB: The Paraphrase Database. NAACL 2013.

Quiz #3

Key Limitations?

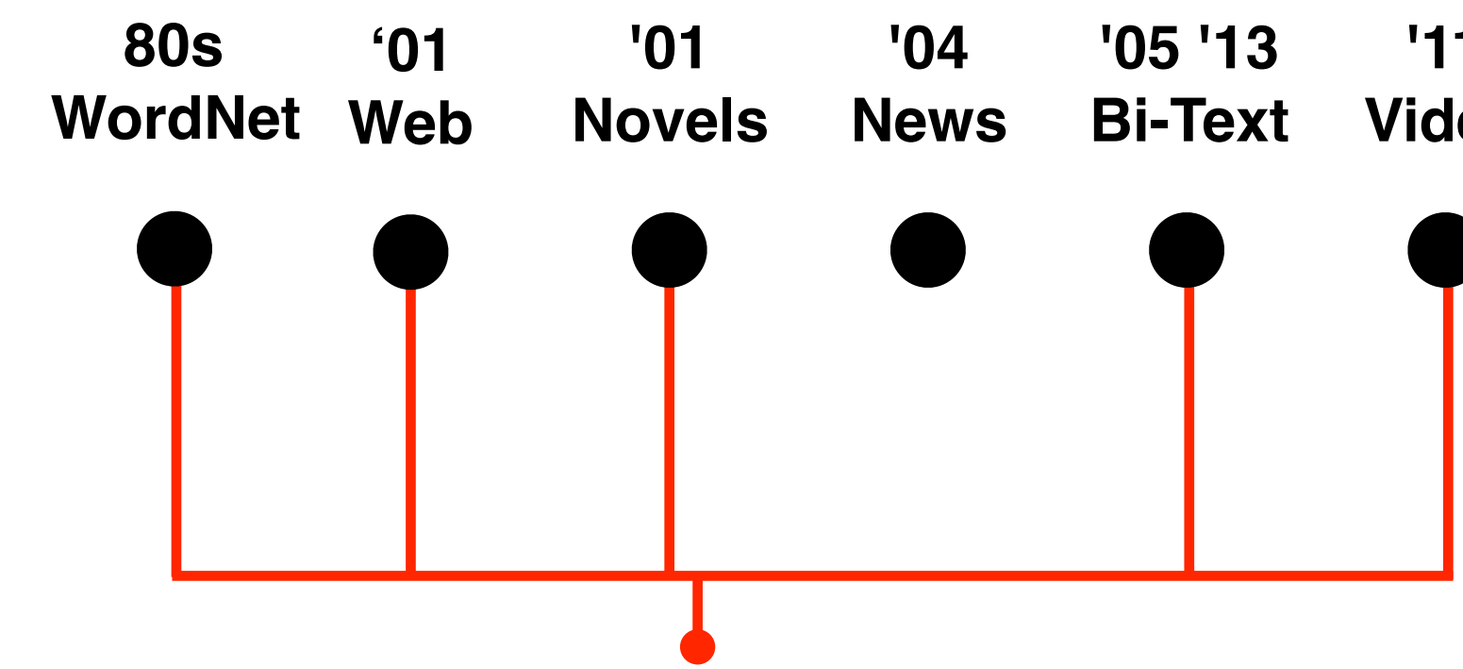
Quiz #3

word sense



Source: Chris Callison-Burch

One Key Limitation

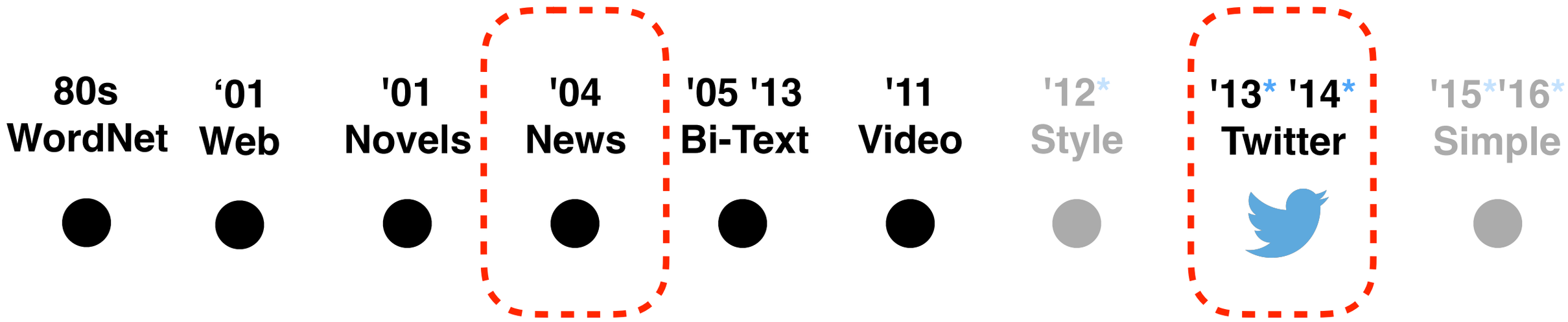


only paraphrases, no non-paraphrases

* my research

Wei Xu. “Data-driven Approaches for Paraphrasing Across Language Variations” PhD Thesis (2014)

Also Non-Paraphrases



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

* my research

Wei Xu. “Data-driven Approaches for Paraphrasing Across Language Variations” PhD Thesis (2014)

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!

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

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

News Paraphrase Corpus

Microsoft Research Paraphrase Corpus

also contains some non-paraphrases

(Dolan, Quirk and Brockett, 2004; Dolan and Brockett, 2005; Brockett and Dolan, 2005)

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

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

Paraphrase Identification:

A Binary Classification Problem

- Input:
  - a sentence pair ***x***
  - a fixed set of binary classes ***Y = {0, 1}***

negative (non-paraphrases)

positive (paraphrases)

- Output:
  - a predicted class ***y*** $\in$ ***Y*** (***y = 0*** or ***y = 1***)

Paraphrase Identification:

A Binary Classification Problem

- Input:
  - a sentence pair ***x***
  - a fixed set of binary classes ***Y = {0, 1}***

- Output:
  - a predicted class ***y*** ∈ ***Y*** (***y = 0*** or ***y = 1***)

Classification Method:

Supervised Machine Learning

- Input:
  - a sentence pair ***x***
  - 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***)

Classification Method:

Supervised Machine Learning

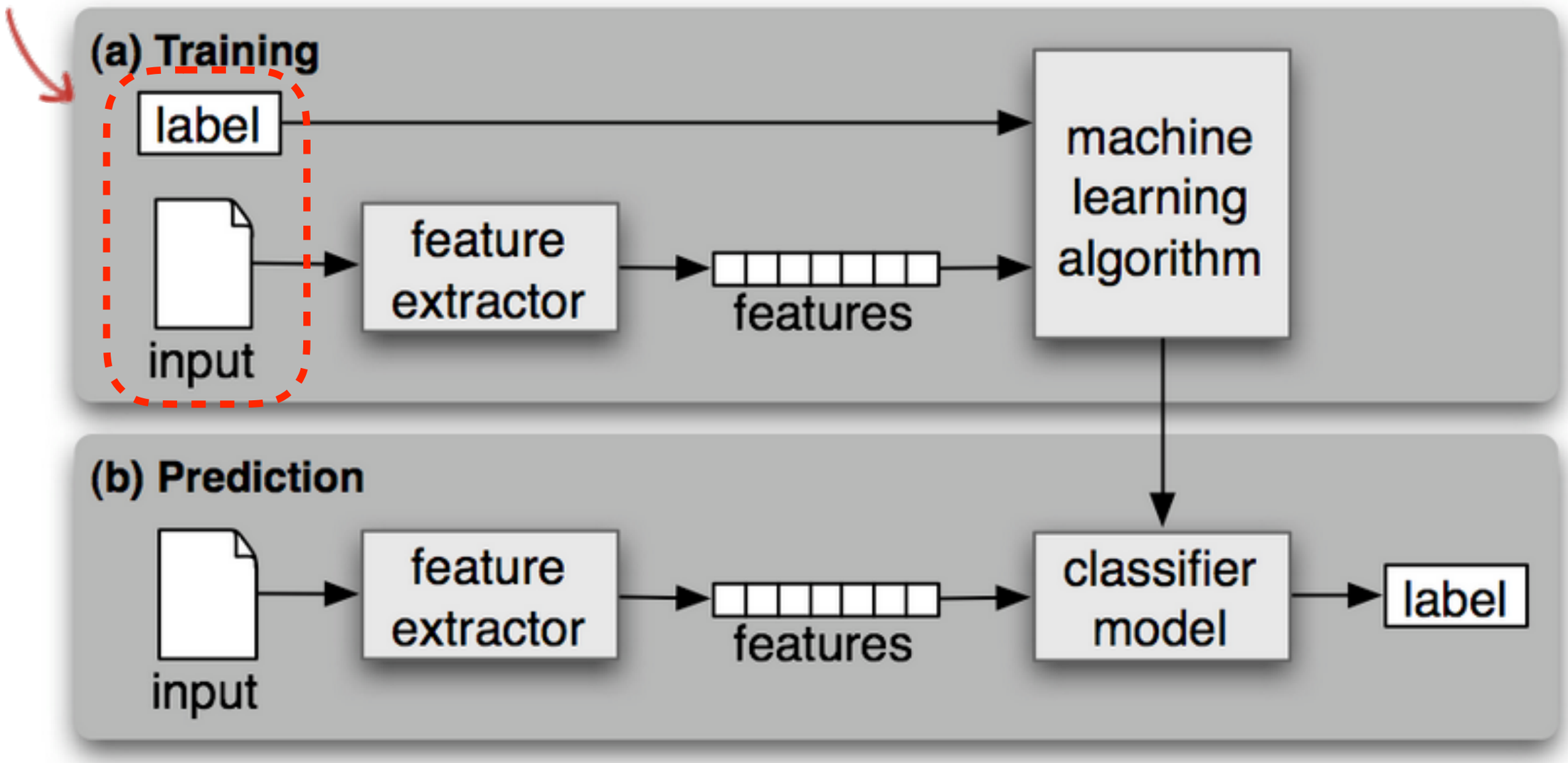
- Input:
  - a sentence pair ***x*** **(represented by features)**
  - a fixed set of binary classes ***Y = {0, 1}***
  - a training set of ***m*** hand-labeled sentence pairs ***$(x_1, y_1), \ldots, (x_m, y_m)$***

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

(Recap Week#3) Classification Method:

Supervised Machine Learning

training set



Source: NLTK Book

(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

$$P(t_1, t_2, ..., t_n \mid y) = P(t_1 \mid y) \cdot P(t_2 \mid y) \cdot ... \cdot P(t_n \mid y)$$

- ***This will cause problems:***
  - *correlated features → double-counted evidence*
  - *while parameters are estimated independently*
  - *hurt classier'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:

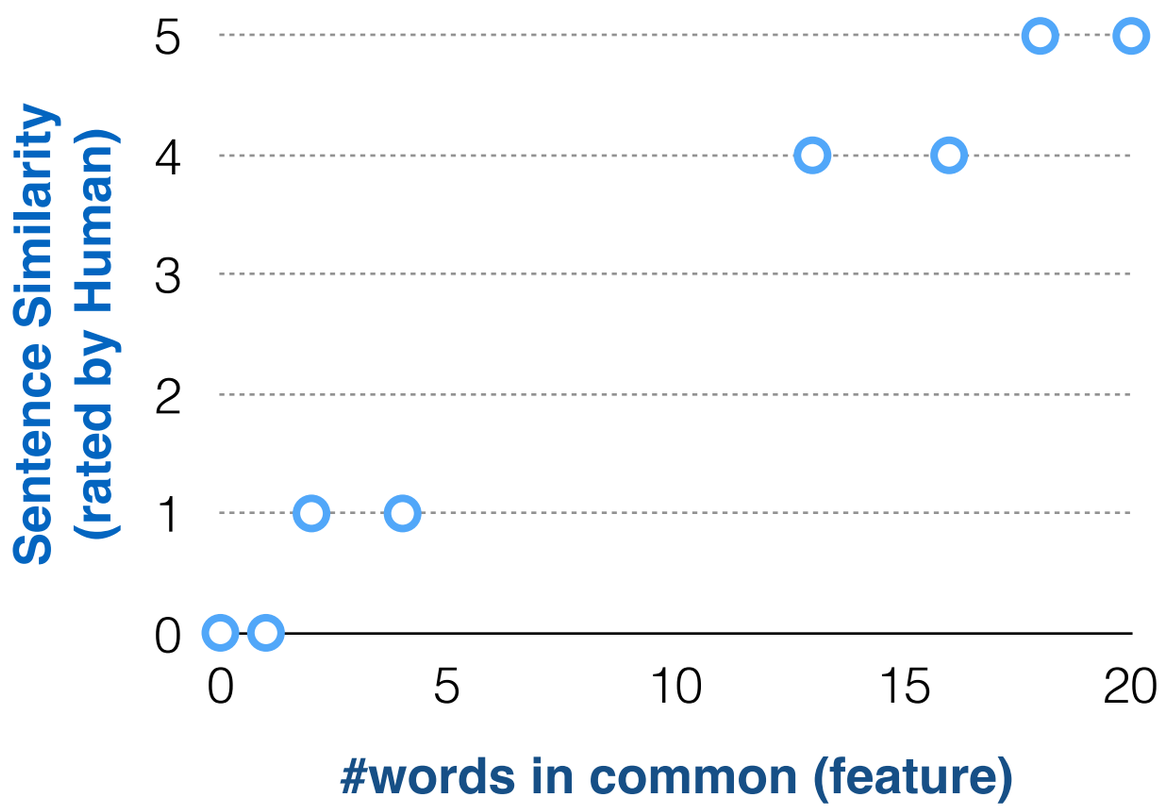
Semantic Textual Similarity

- 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

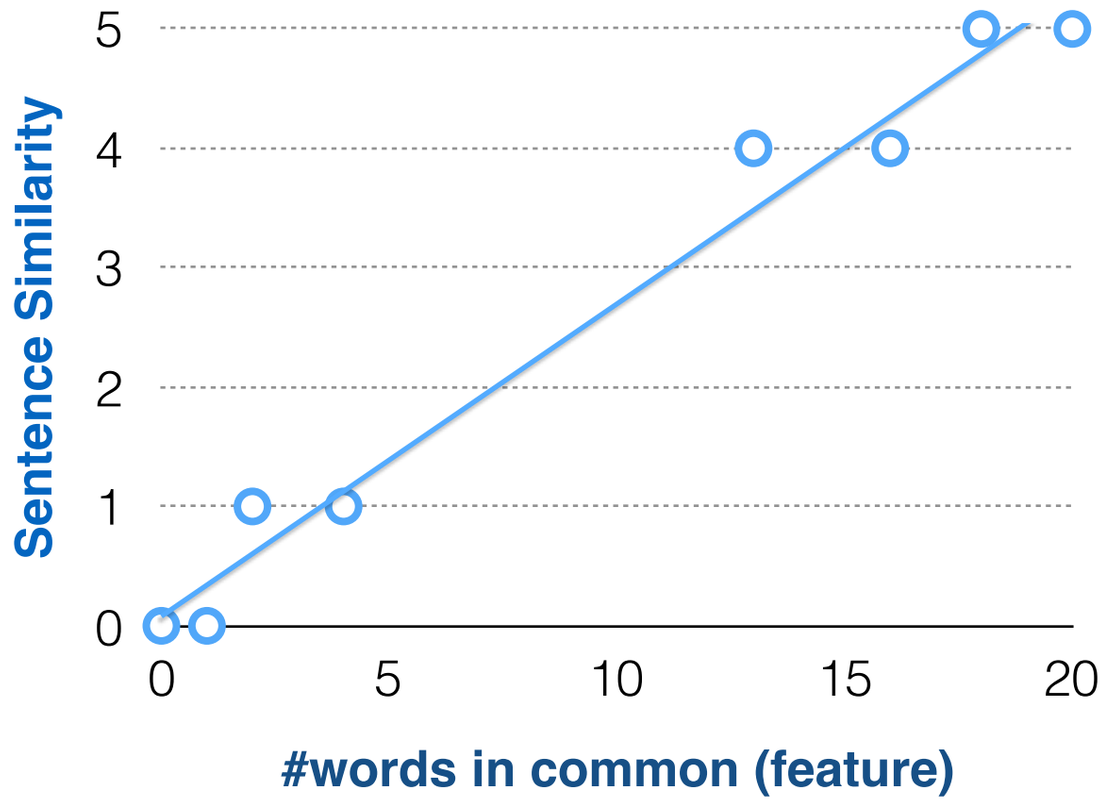
A Simpler Model:

Linear Regression



A Simpler Model:

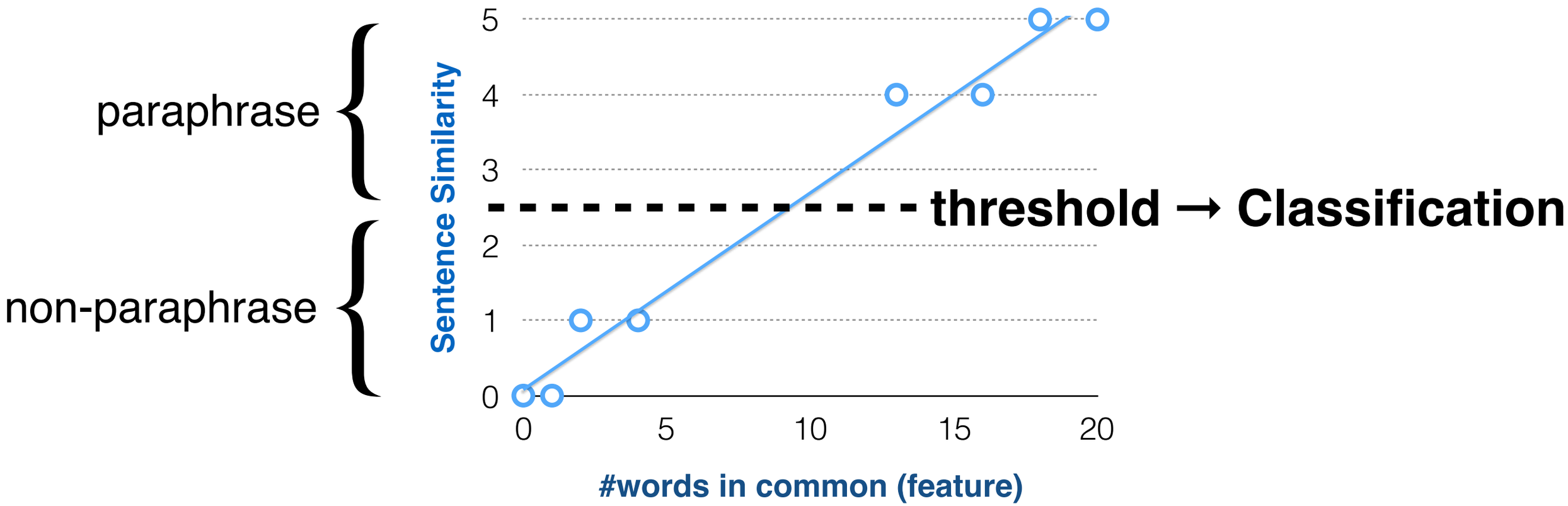
Linear Regression



- 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

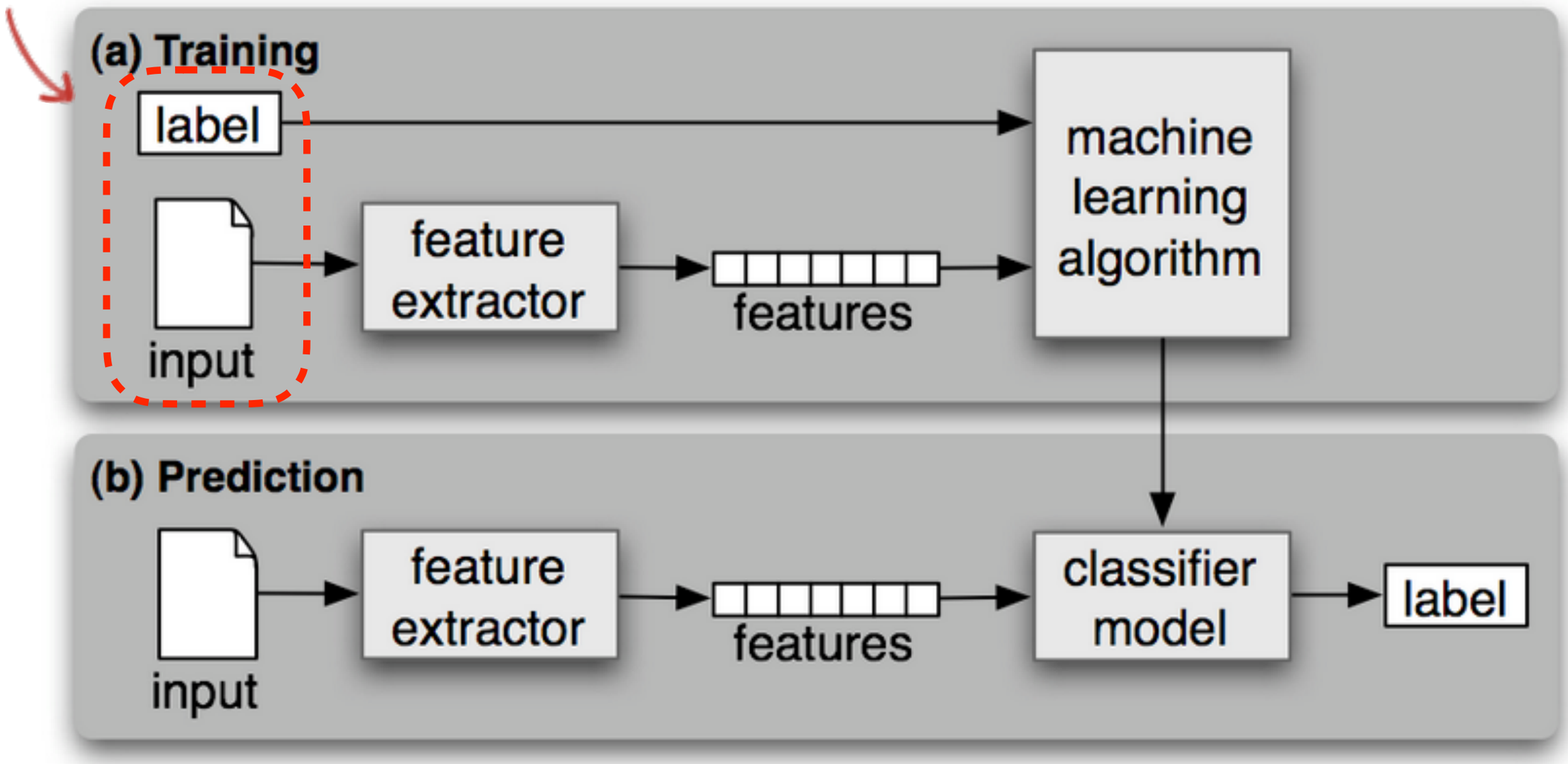
#words in common (*x*)	Sentence Similarity (*y*)
1	0
4	1
13	4
18	5
…	…

- ***m*** hand-labeled sentence pairs ***$(x_1, y_1), \ldots, (x_m, y_m)$***
- ***x***'s: "input" variable / features
- ***y***'s: "output"/"target" variable

(Recap Week#3)

Supervised Machine Learning

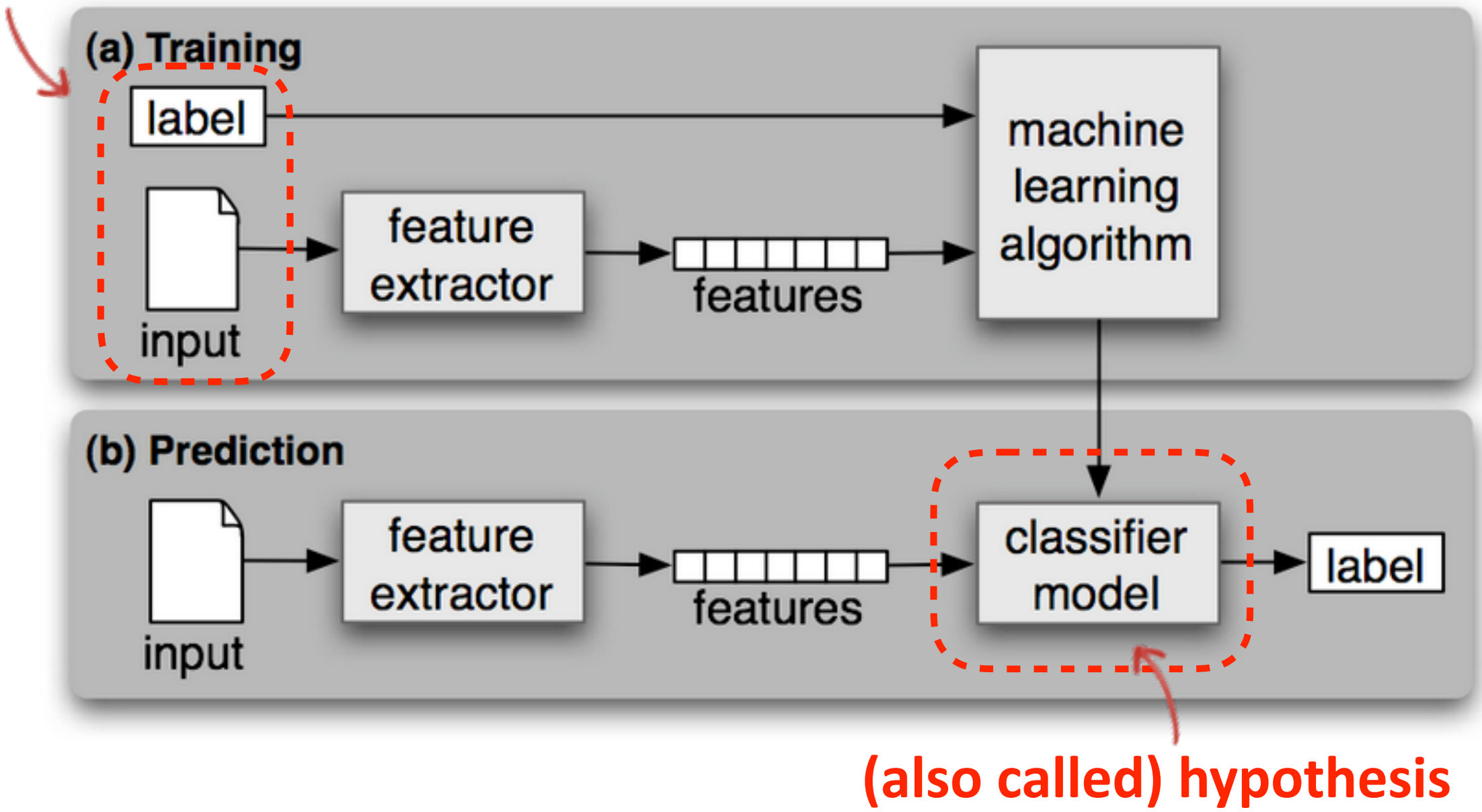
training set



Source: NLTK Book

Supervised Machine Learning

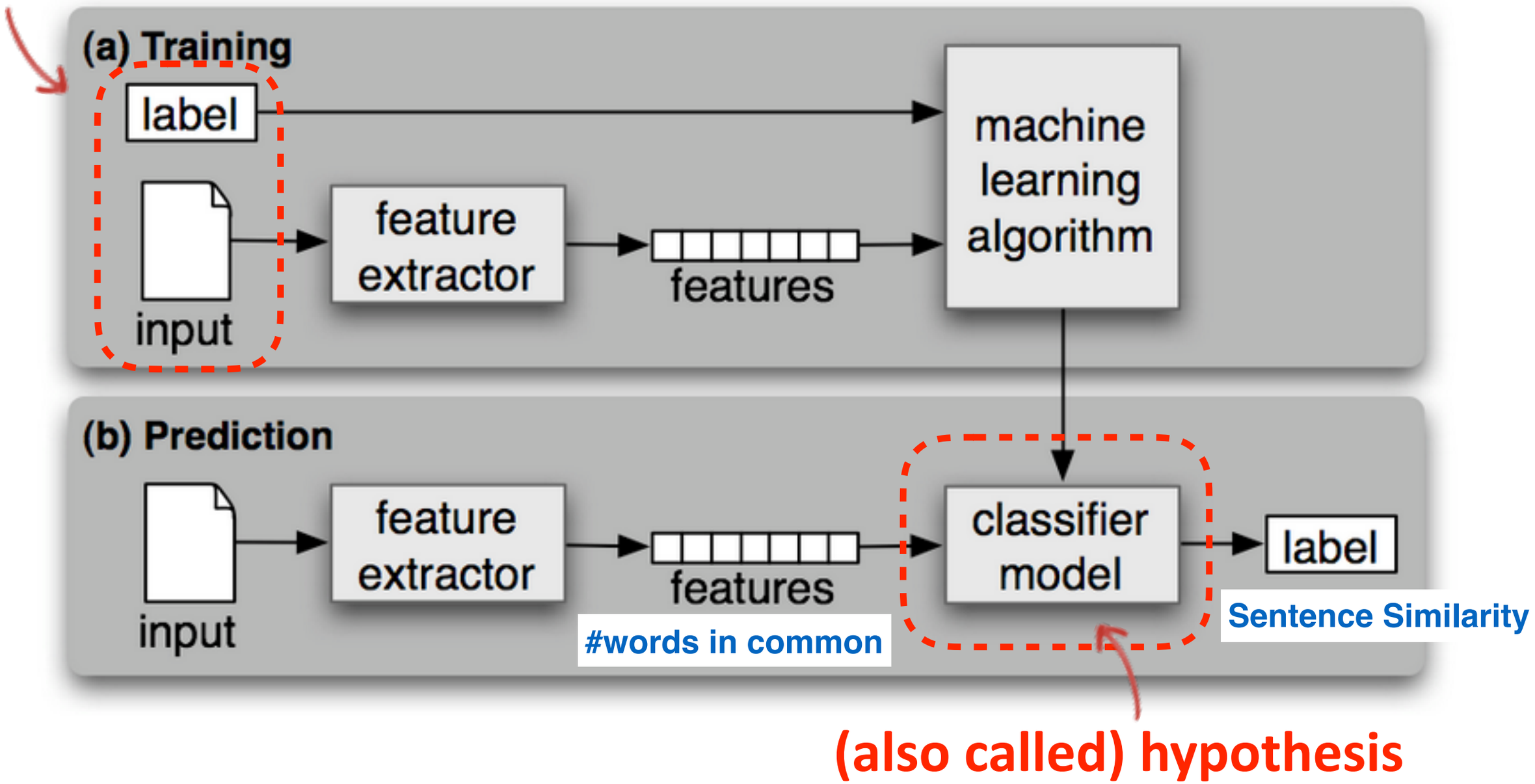
training set



(also called) hypothesis

Supervised Machine Learning

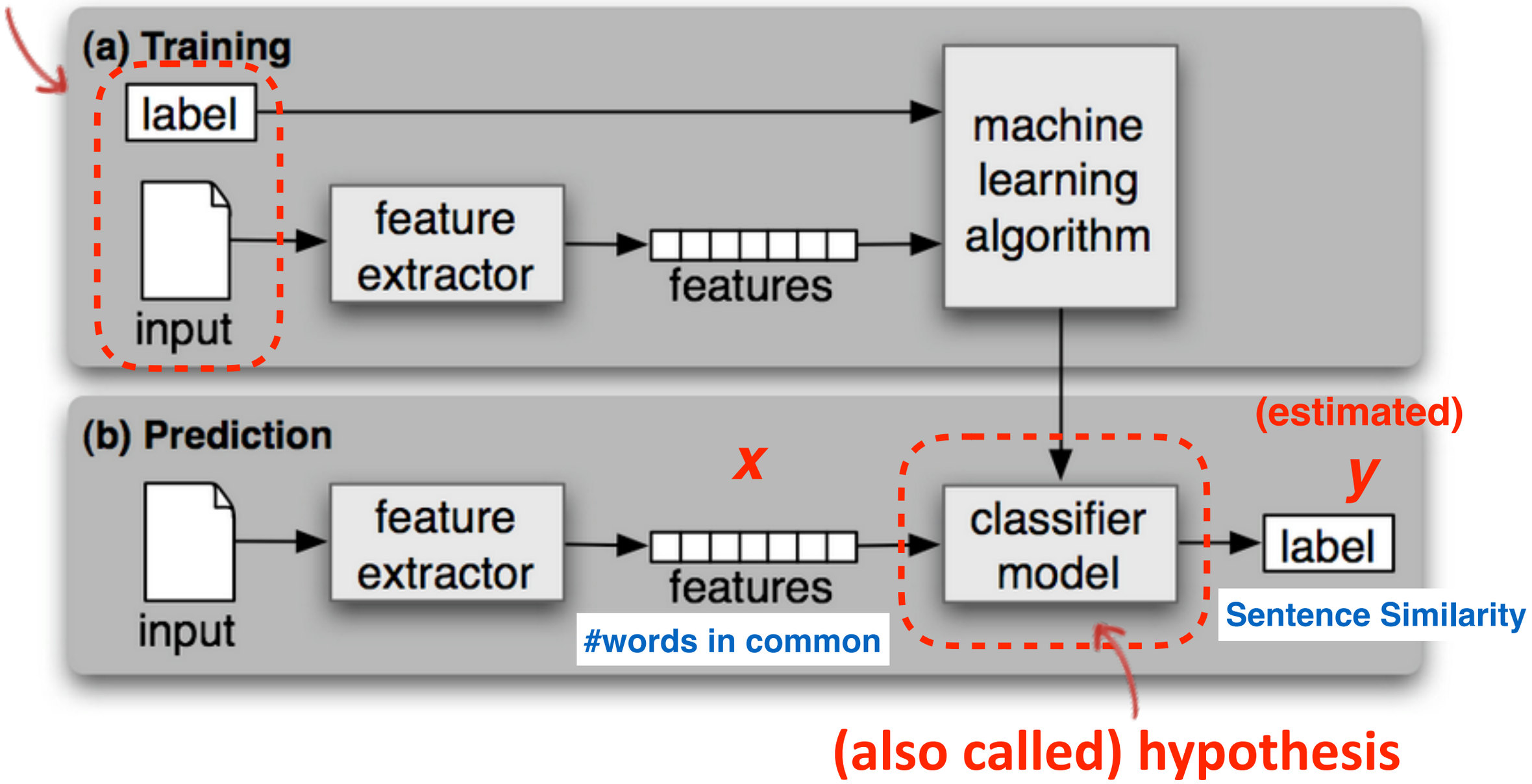
training set



(also called) hypothesis

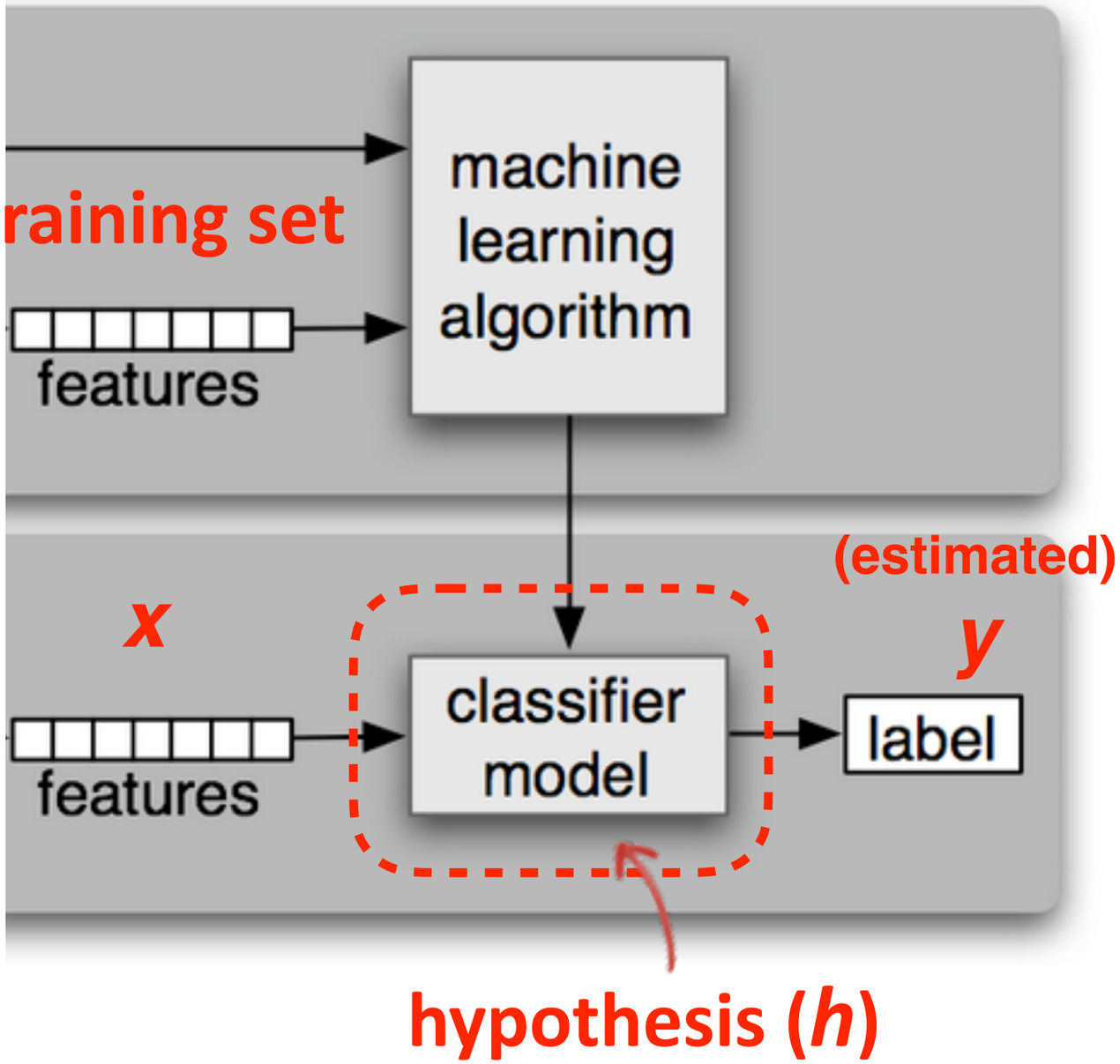
Supervised Machine Learning

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 (*x*)	Sentence Similarity (*y*)
1	0
4	1
13	4
18	5
…	…

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

- ***m*** hand-labeled sentence pairs ***$(x_1, y_1), \ldots, (x_m, y_m)$***
- ***θ***'s: parameters

Source: many following slides are adapted from Andrew Ng

Linear Regression w/ one variable:

Model Representation

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

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

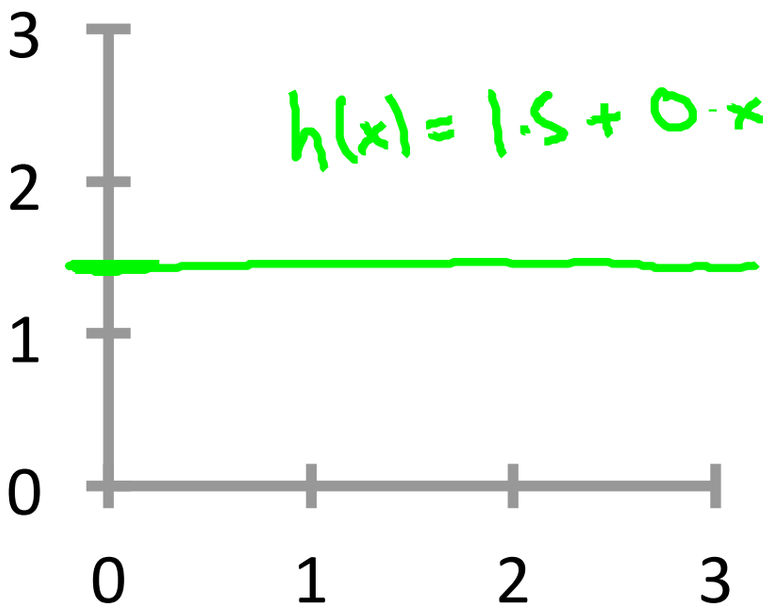
- ***m*** hand-labeled sentence pairs ***$(x_1, y_1), \ldots, (x_m, y_m)$***
- ***θ***'s: parameters

How to choose *θ*?

Linear Regression w/ one variable::

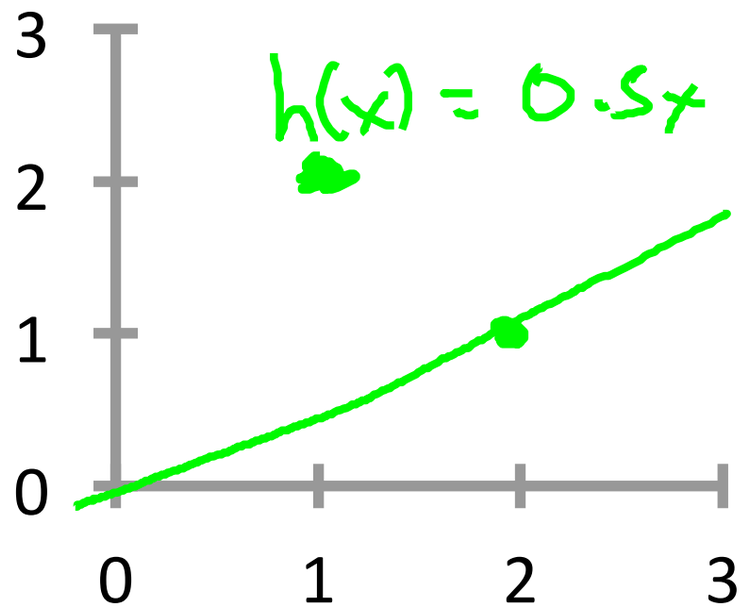
Model Representation

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



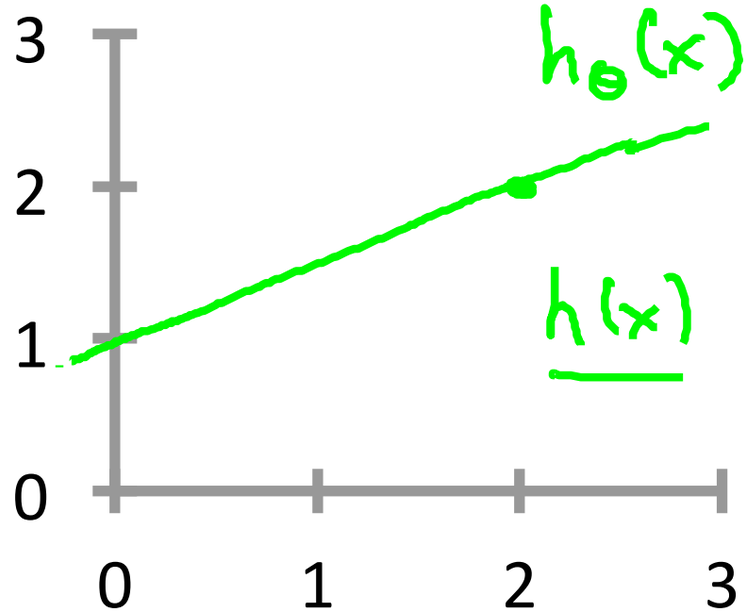
$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

$\theta_1 = 0.5$

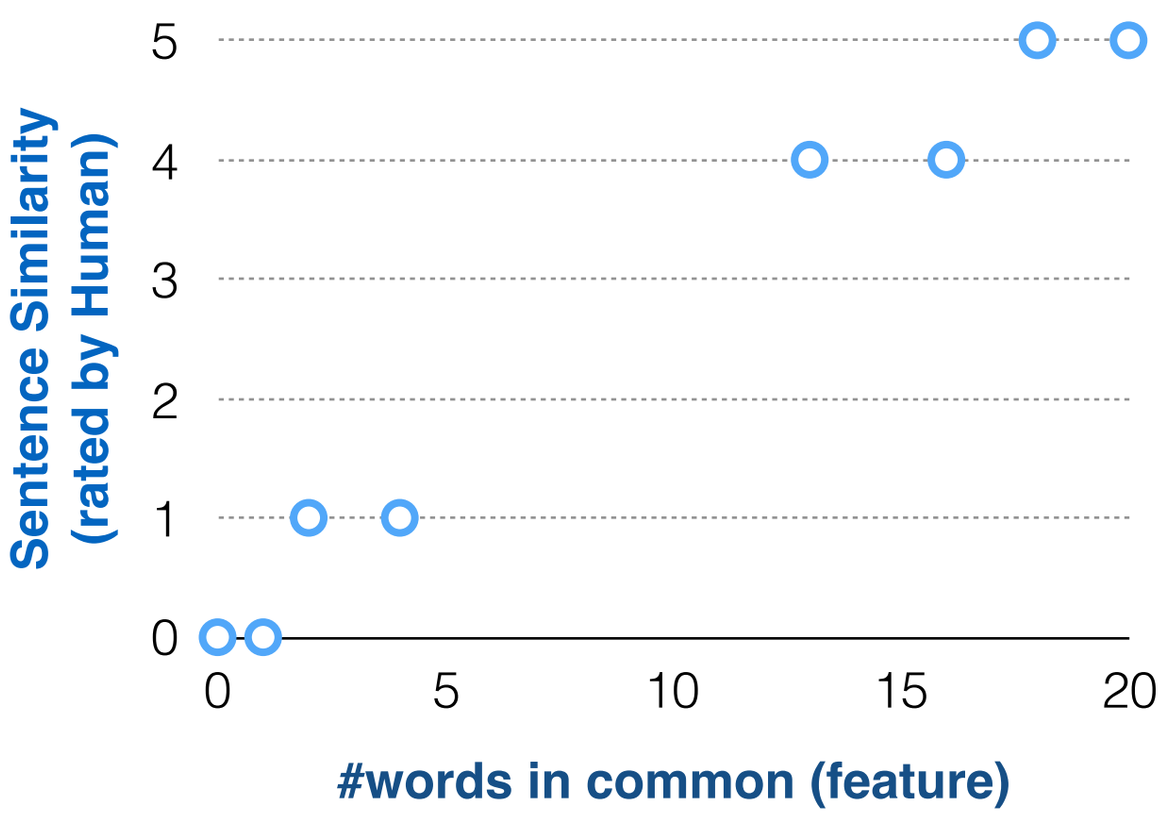


$\theta_0 = 1$

$\theta_1 = 0.5$

Linear Regression w/ one variable:

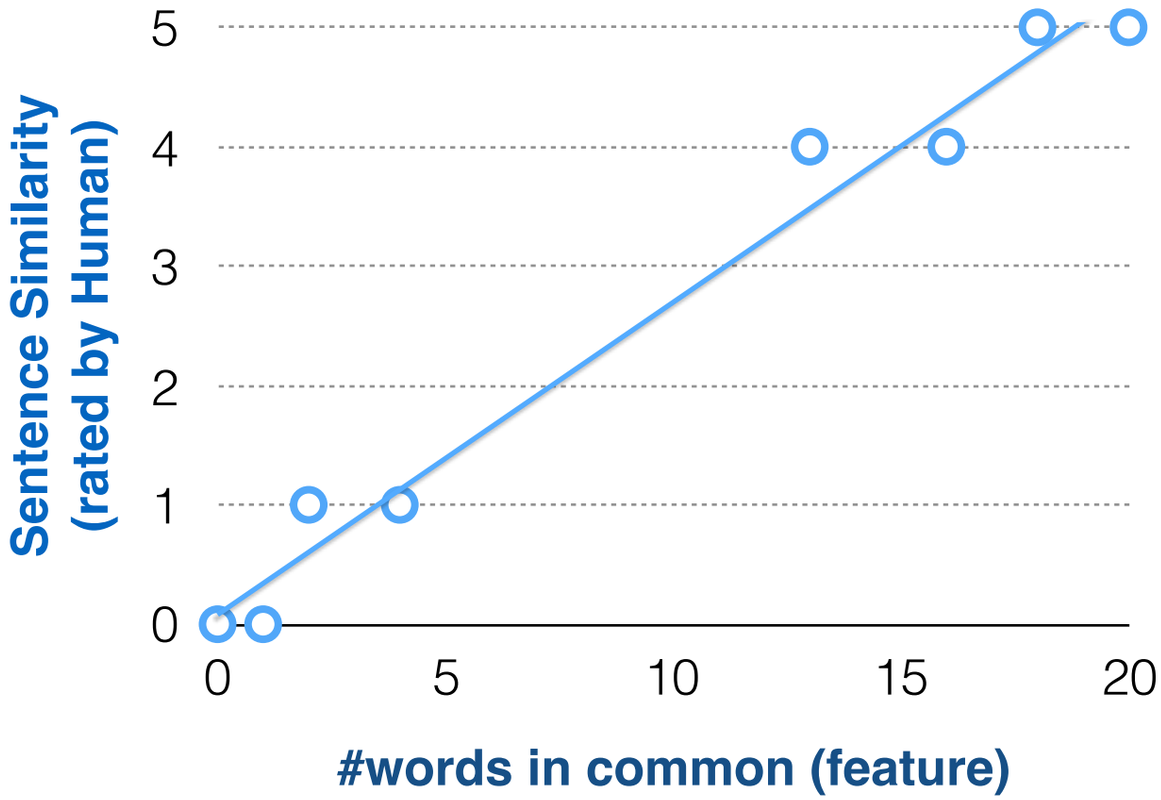
Cost Function



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

Linear Regression w/ one variable:

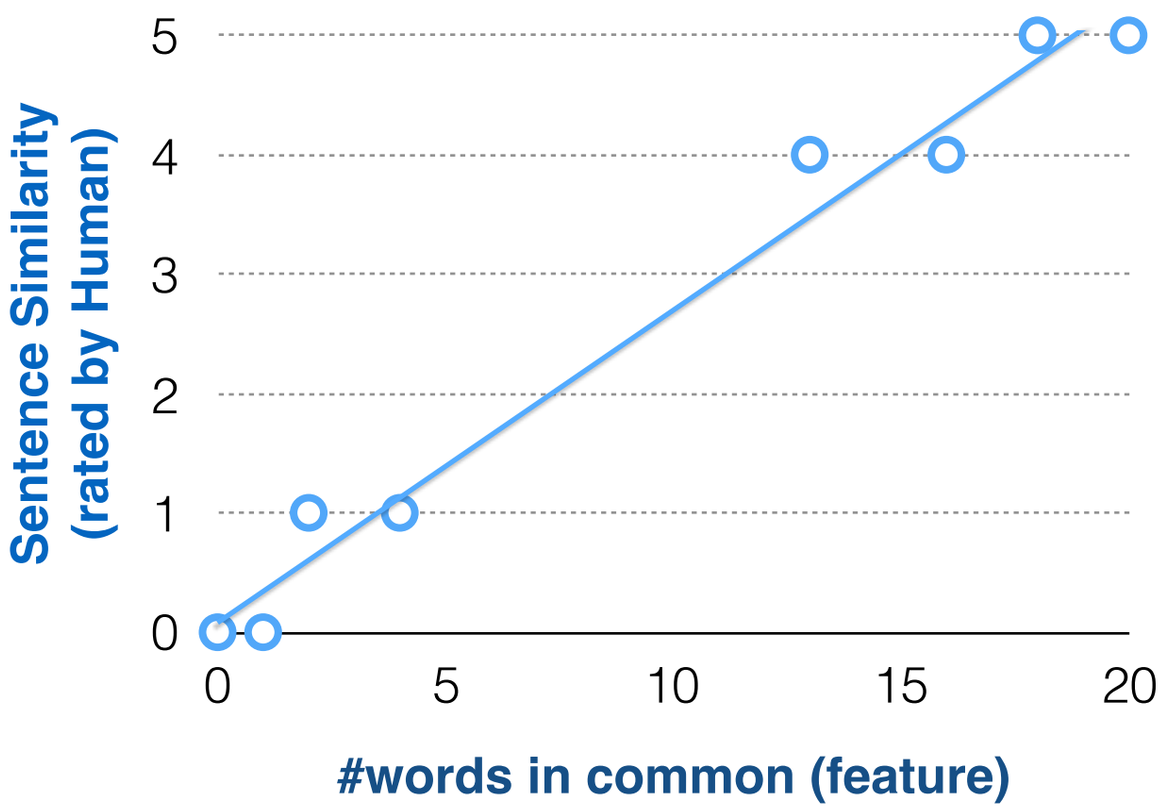
Cost Function



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

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

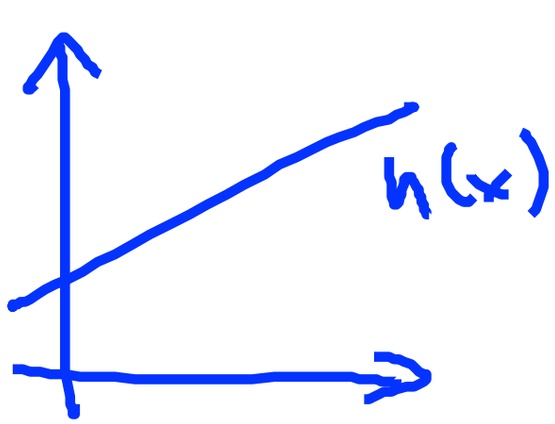
Linear Regression

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

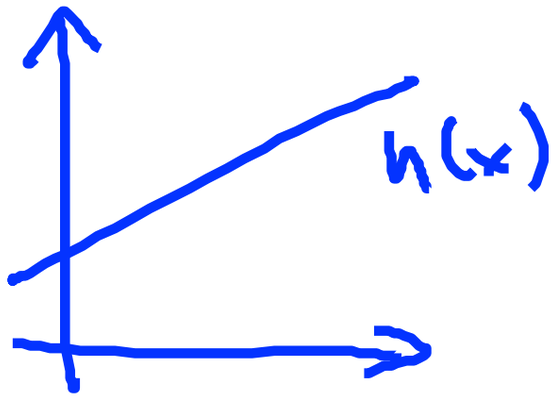
Linear Regression

- **Hypothesis**:

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

- **Parameters:**

θ_0, θ_1



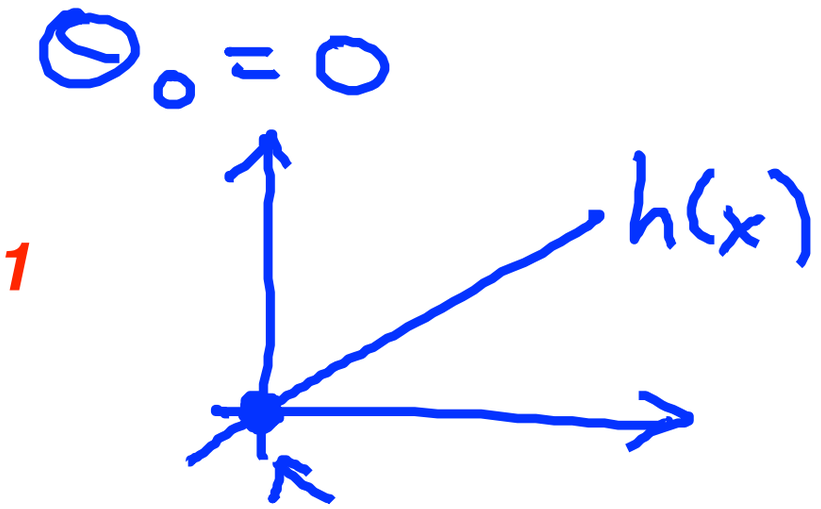
- **Cost Function:**

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

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

Simplified

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

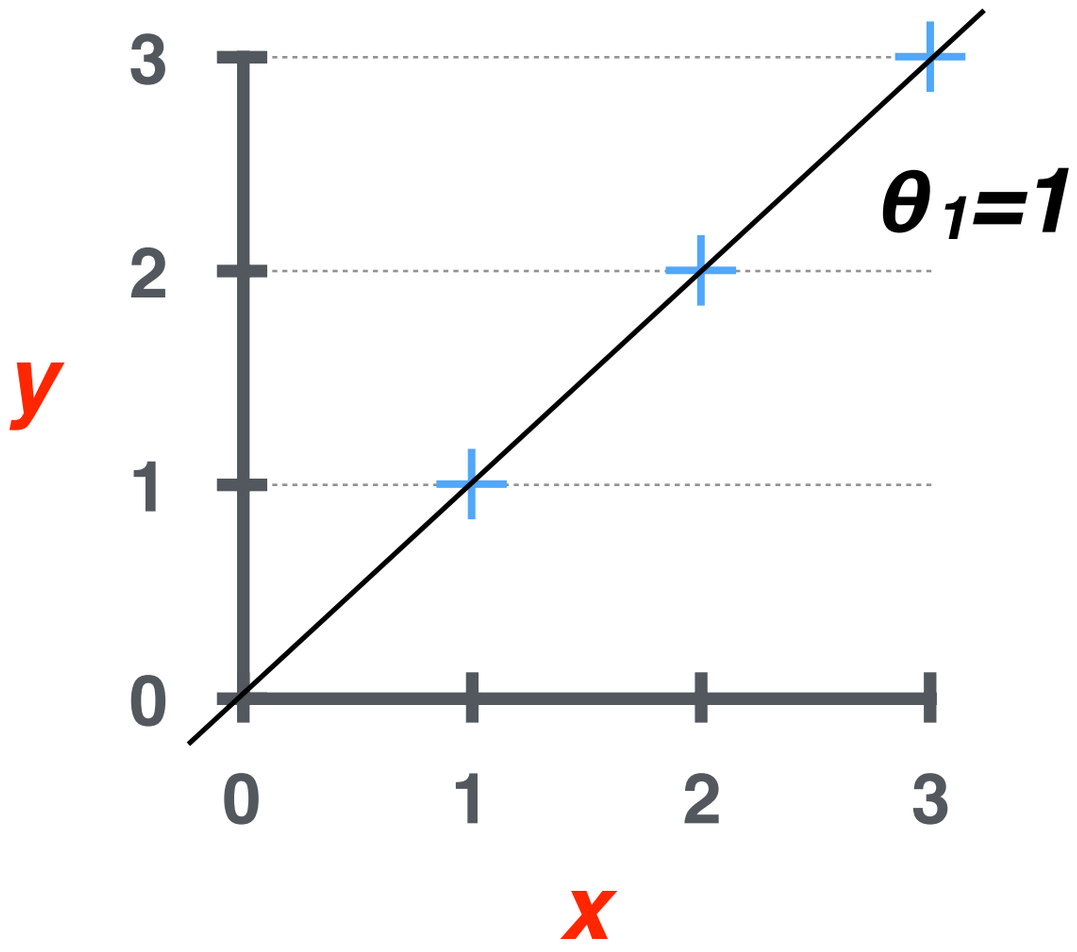


$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

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

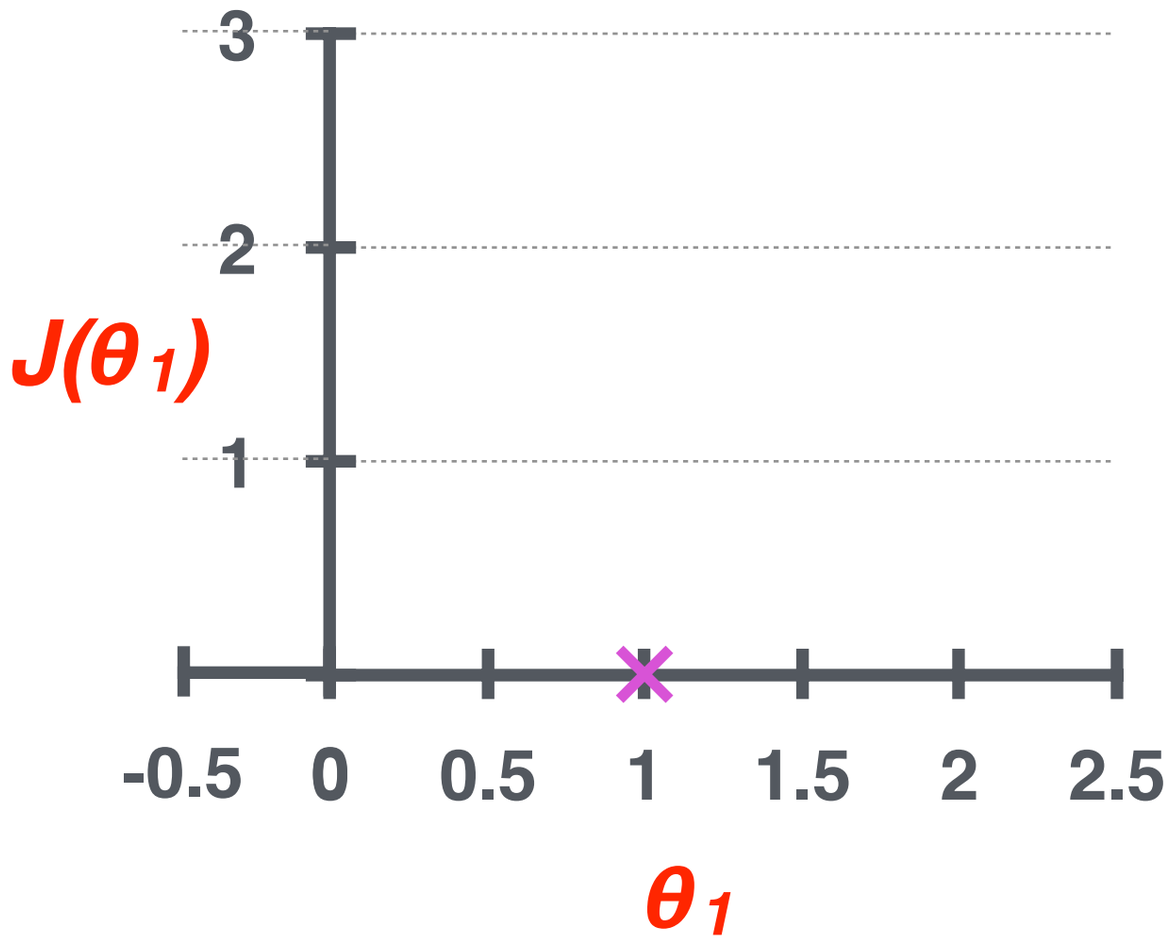
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

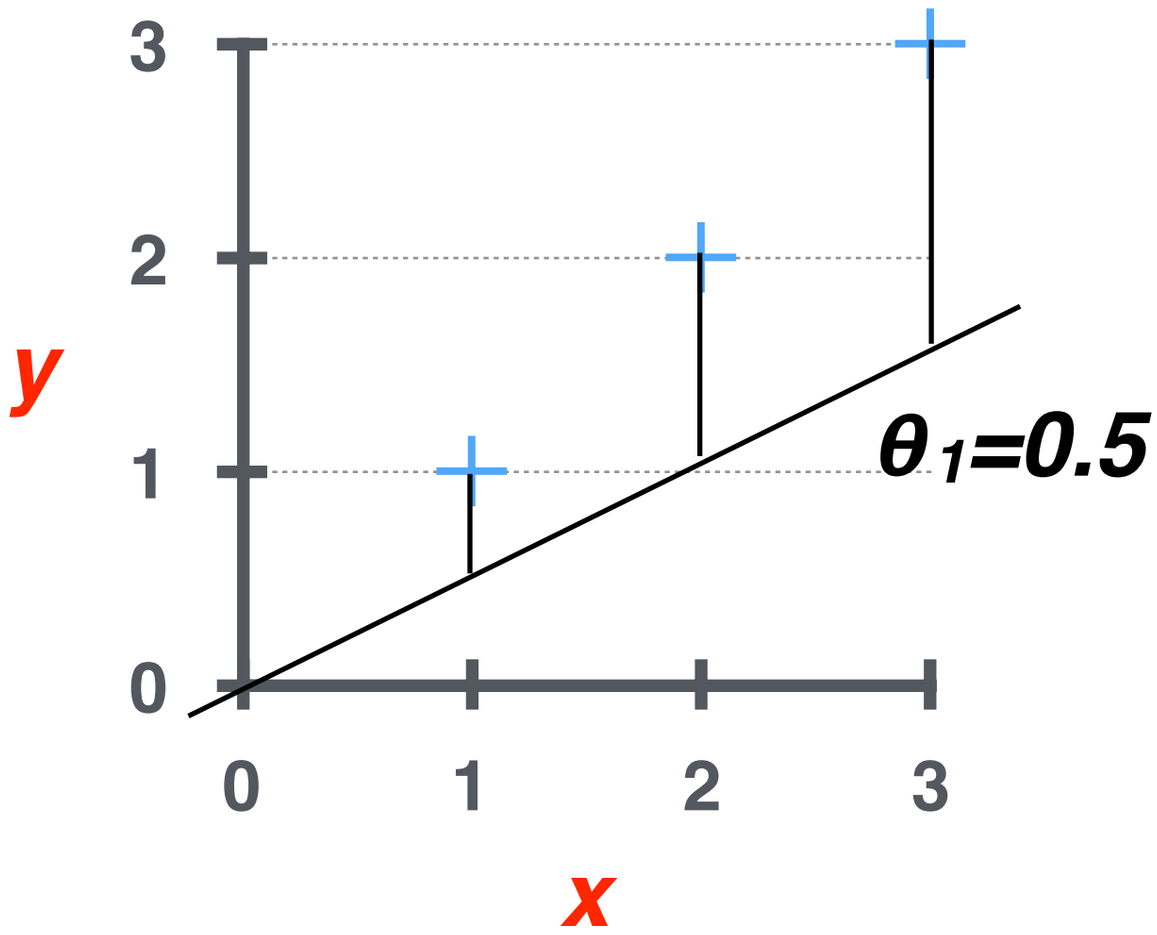
(function of the parameter θ_1)



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

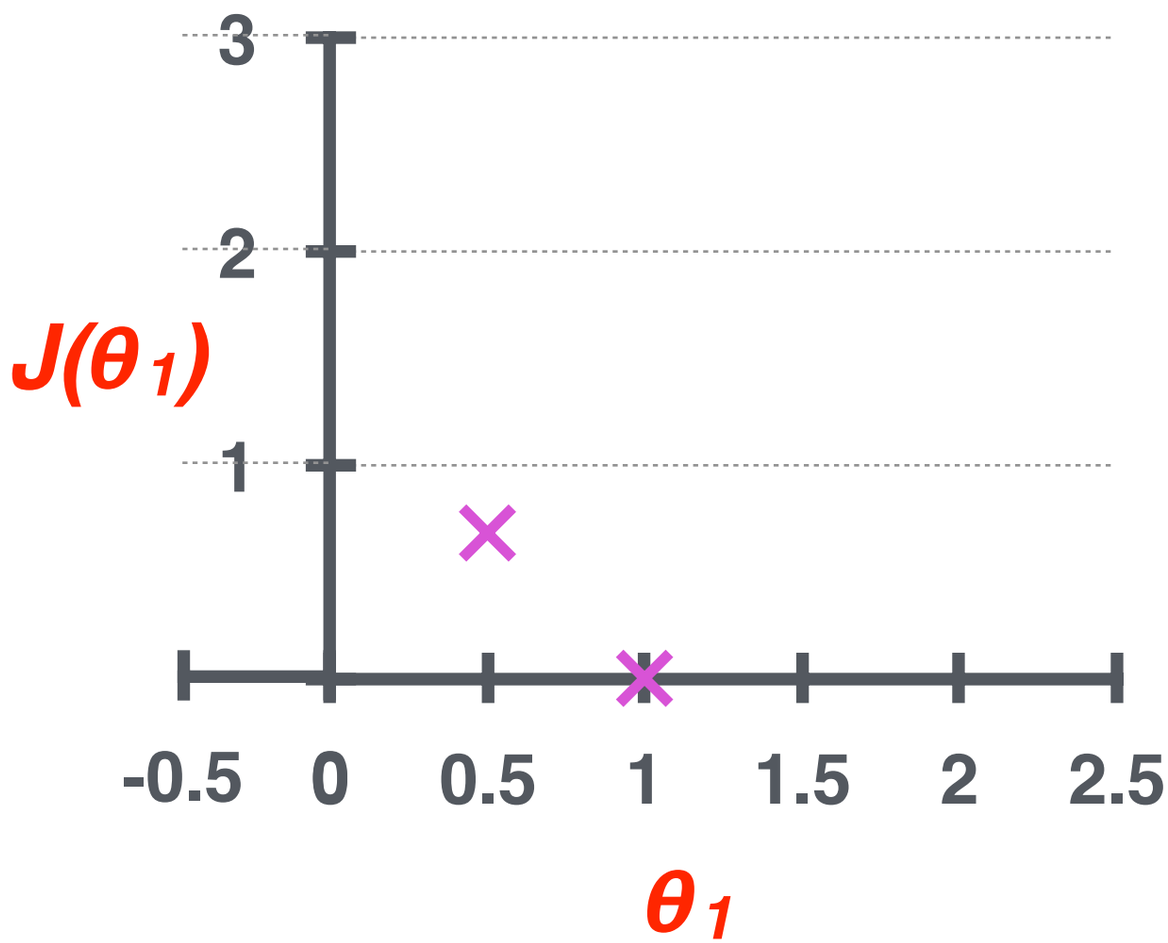
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

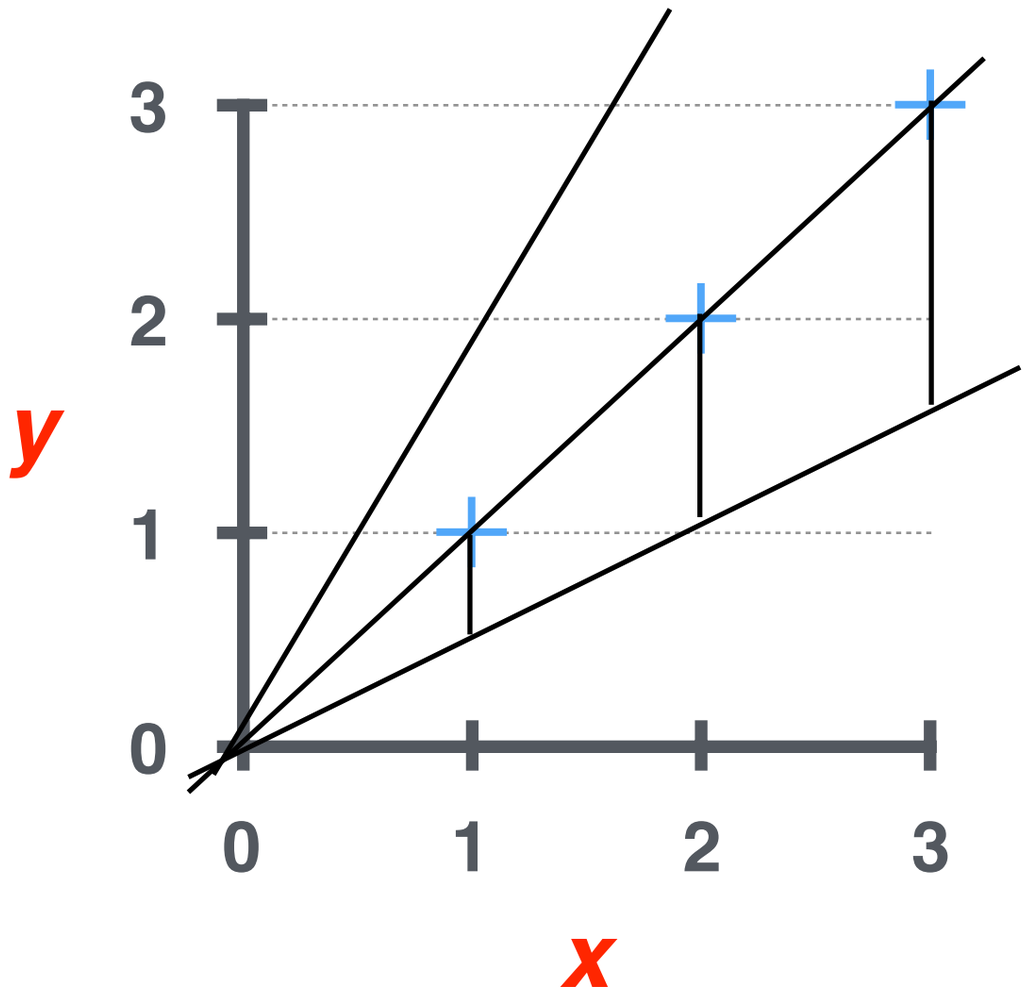
(function of the parameter θ_1)



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

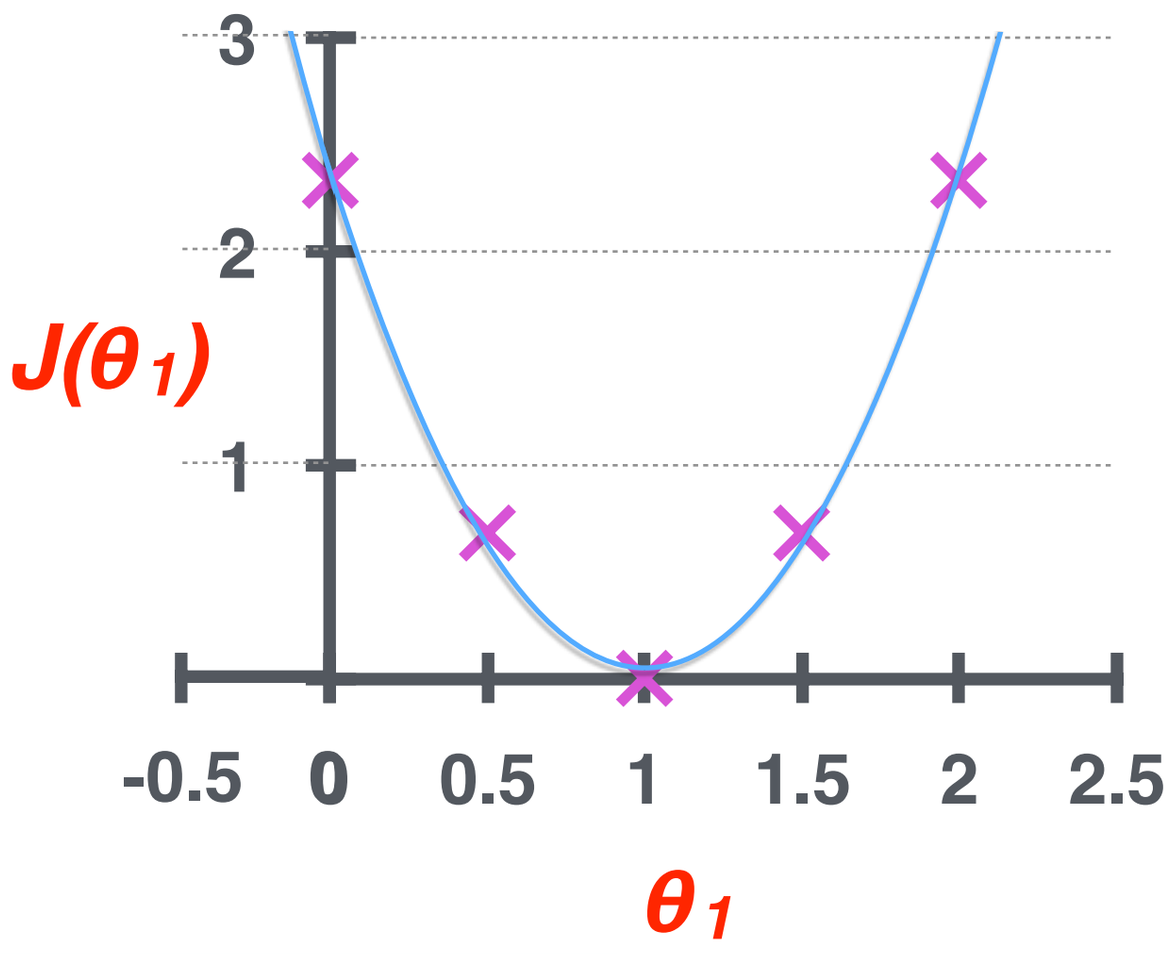
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



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

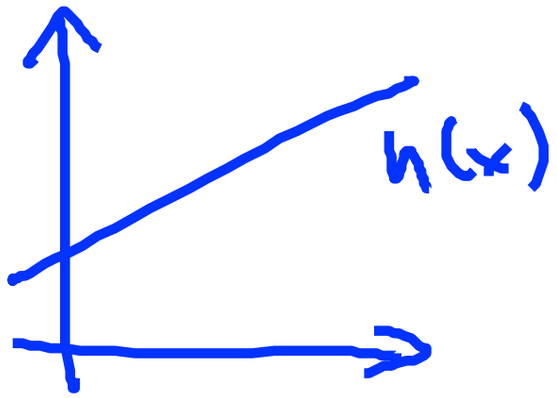
Linear Regression

- **Hypothesis**:

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

- **Parameters:**

θ_0, θ_1



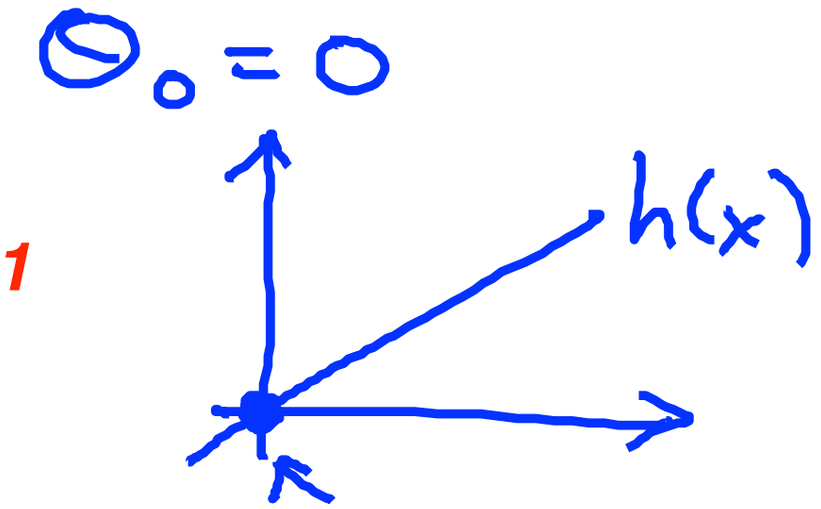
- **Cost Function:**

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

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

Simplified

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



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

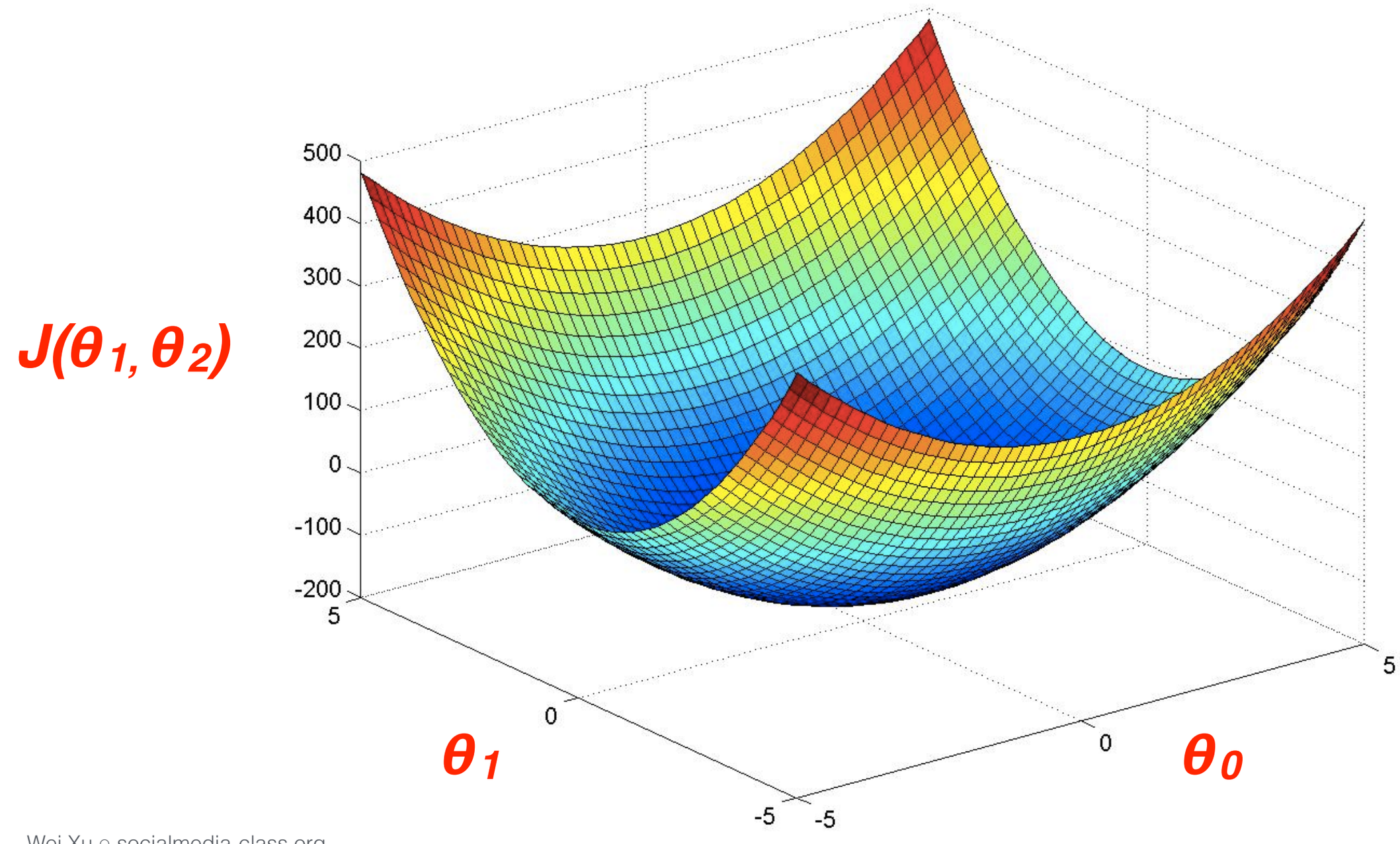
$$\underset{\theta_1}{\text{minimize}}\, J(\theta_1)$$

$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)

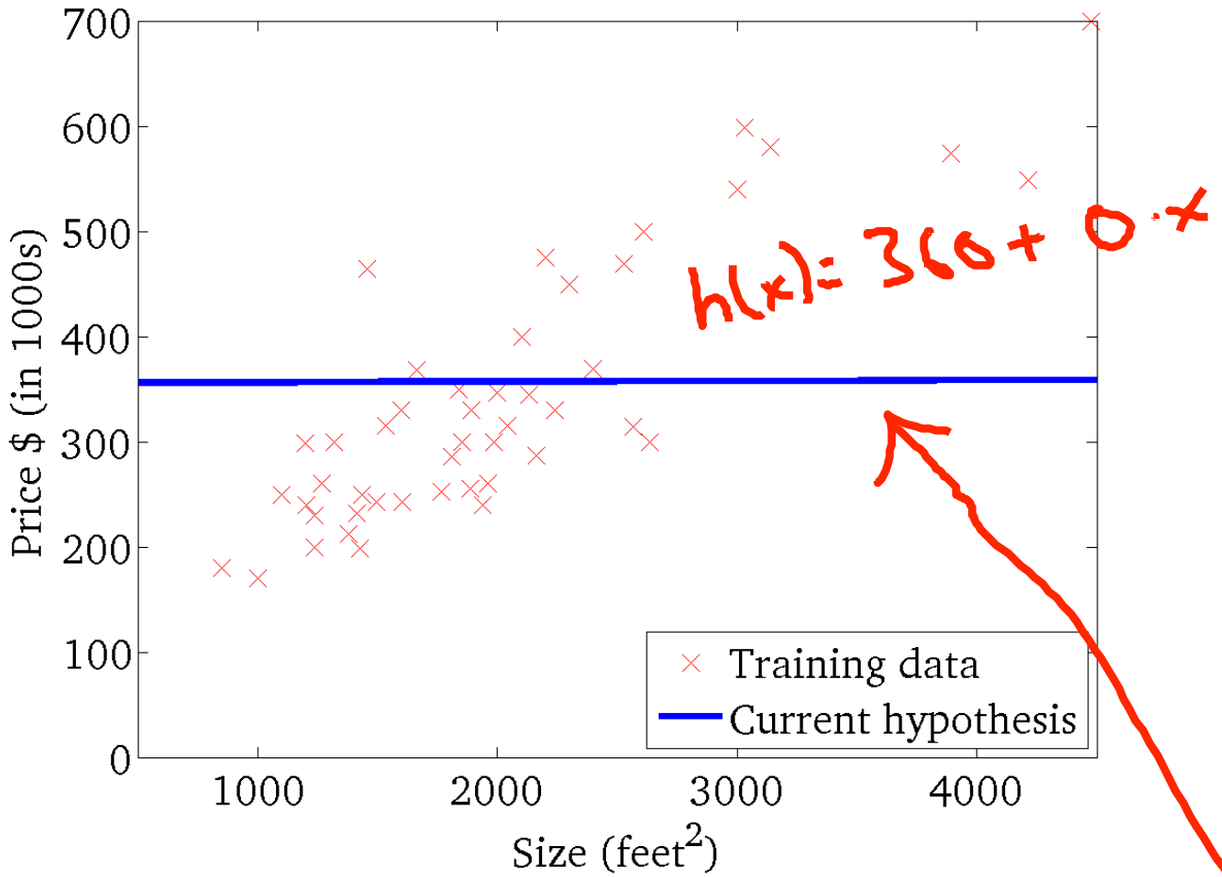


$$h_\theta(x)$$

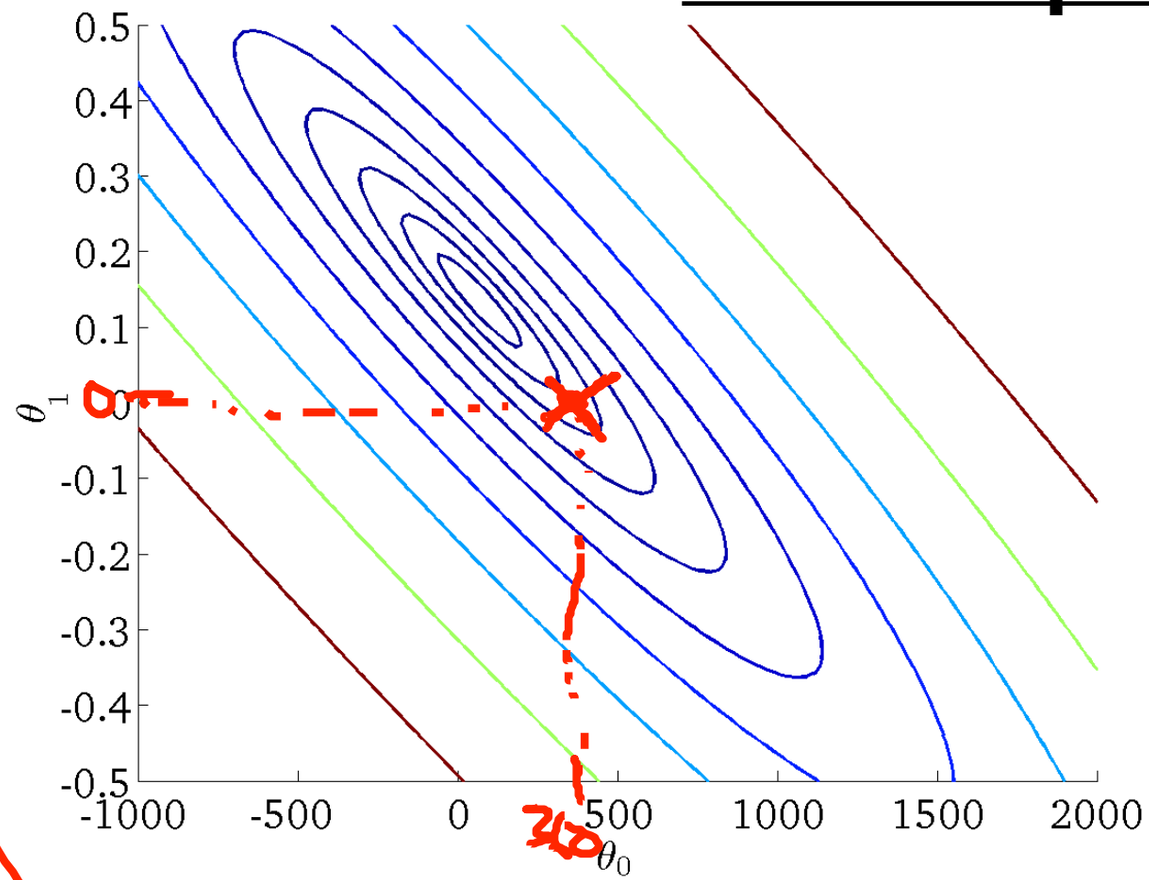
(for fixed θ_0, θ_1, this is a function of x)

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

(function of the parameter θ_0, θ_1)



contour plot



$\theta_0 = 360$

$\theta_1 = 0$

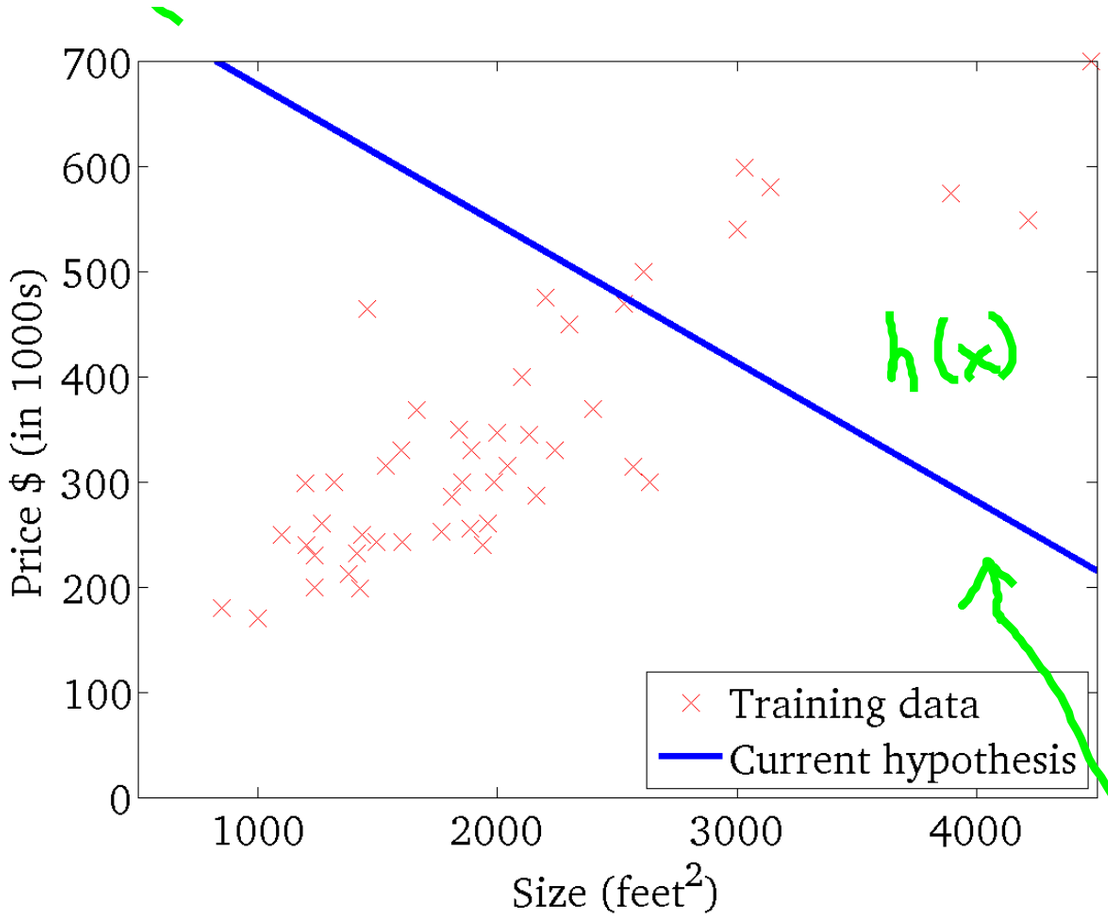
Andrew Ng

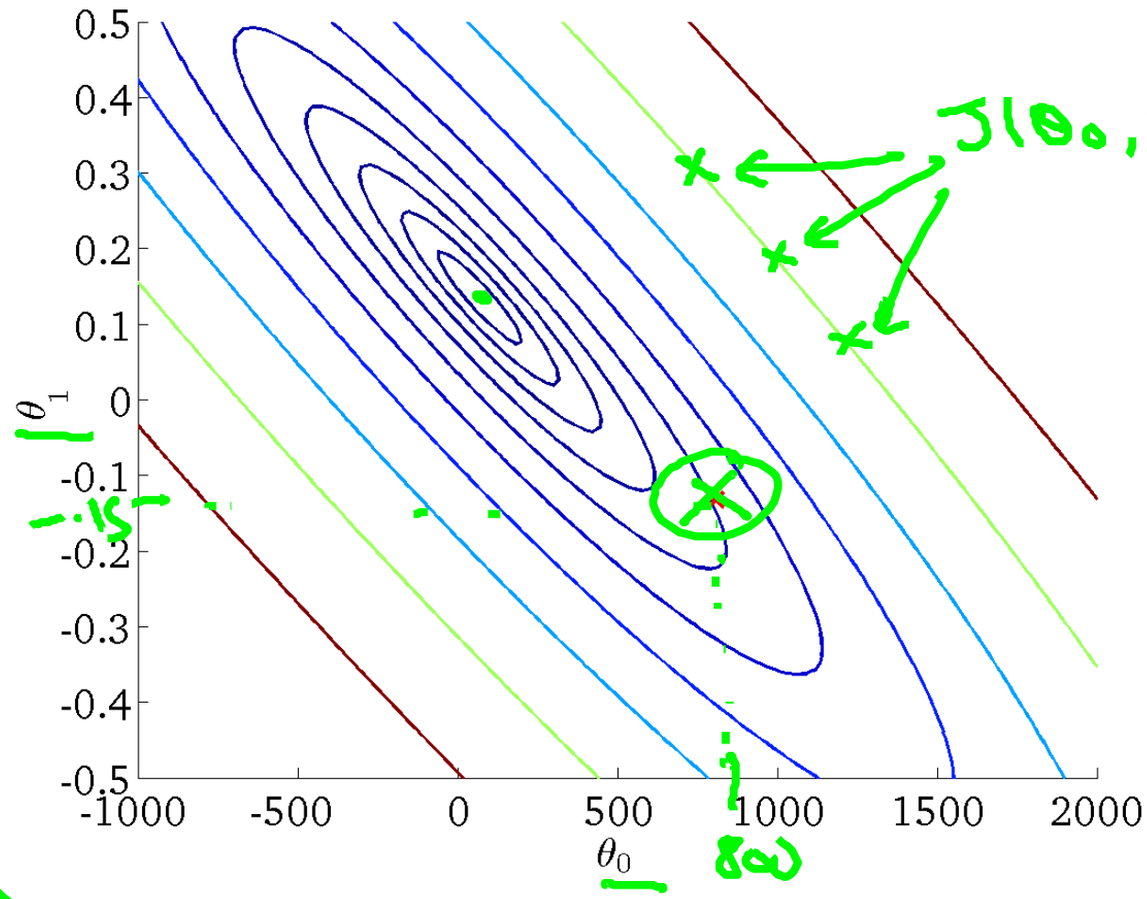
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)





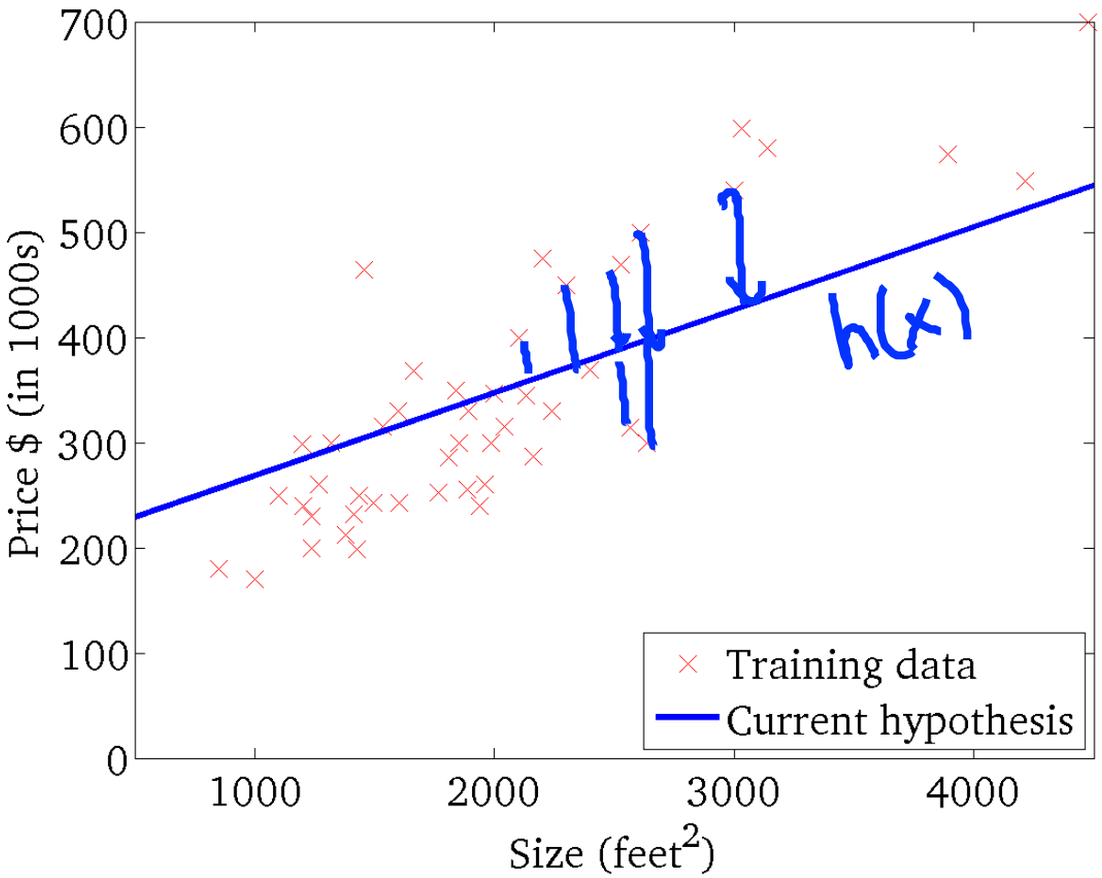
Andrew Ng

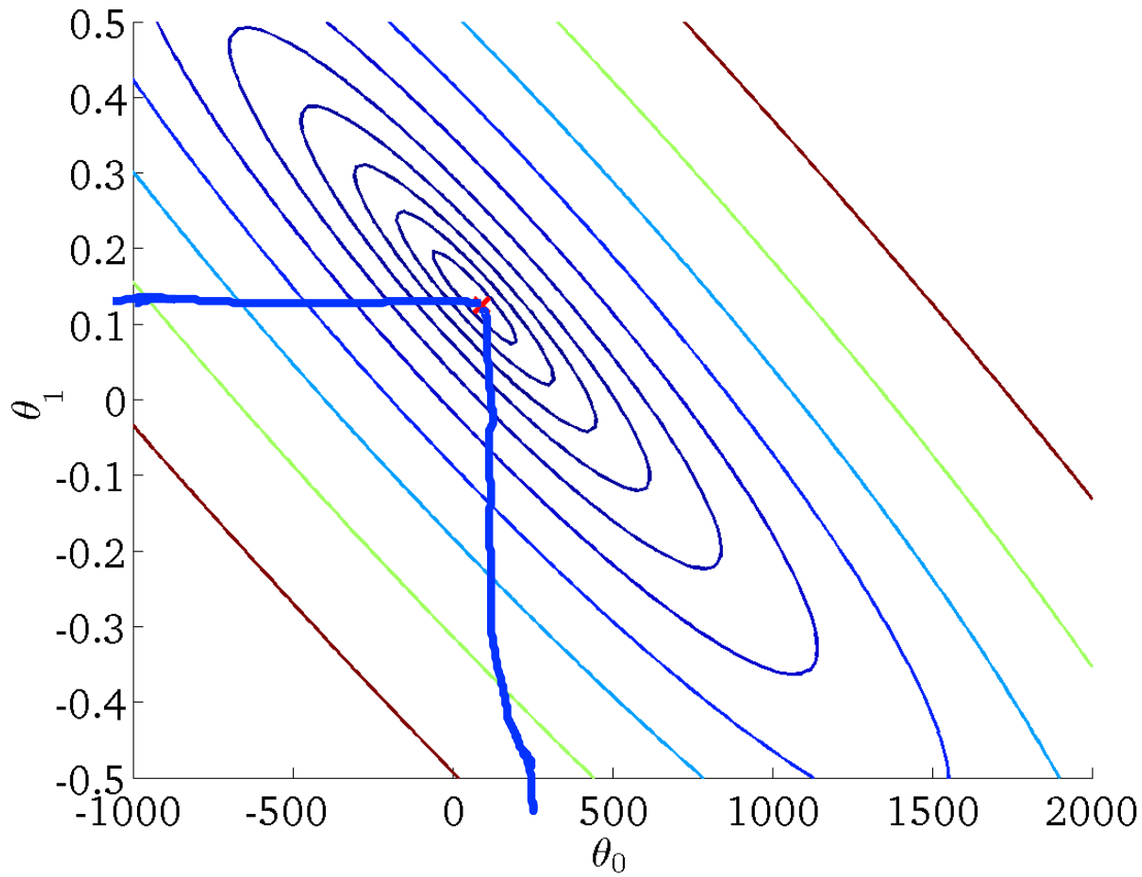
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)





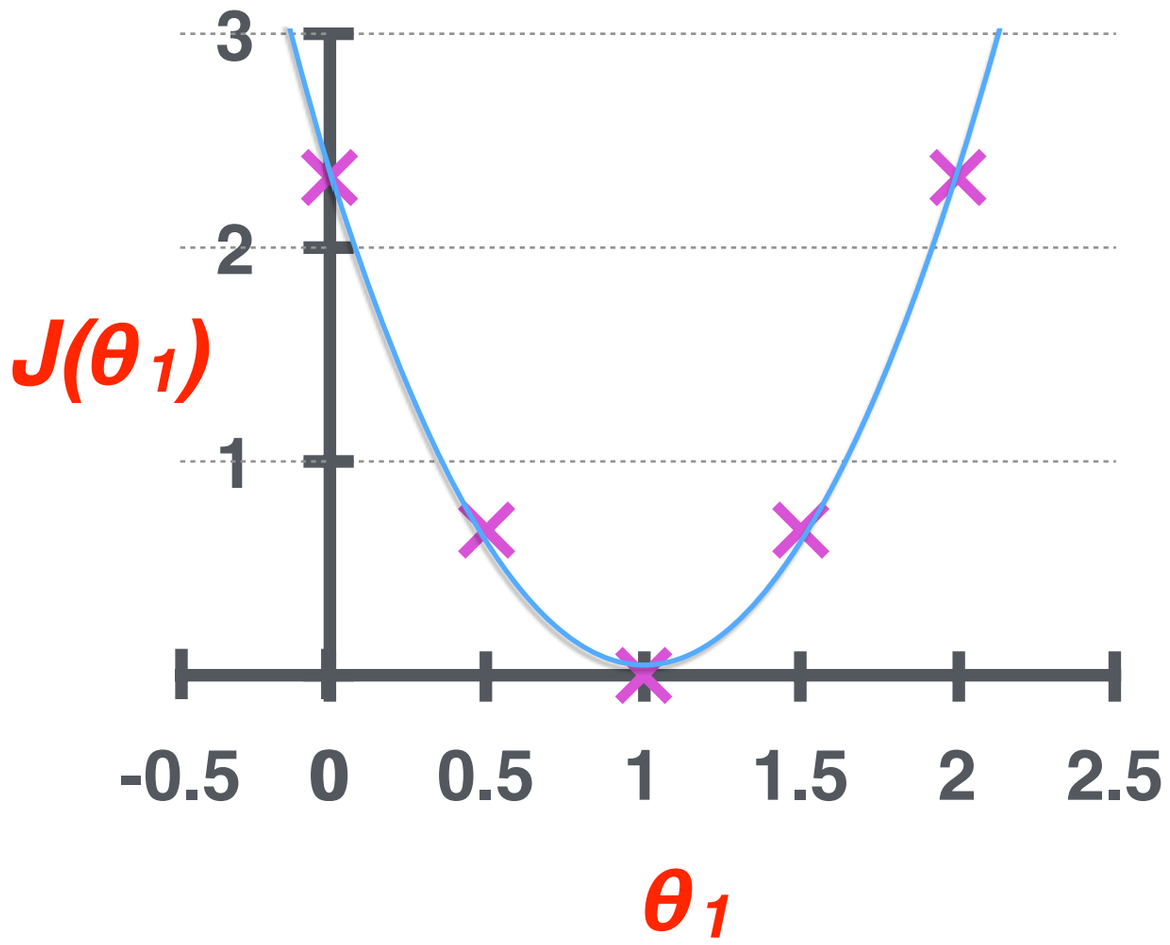
Andrew Ng

Parameter Learning

- Have some function $J(\theta_0, \theta_1)$
- Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

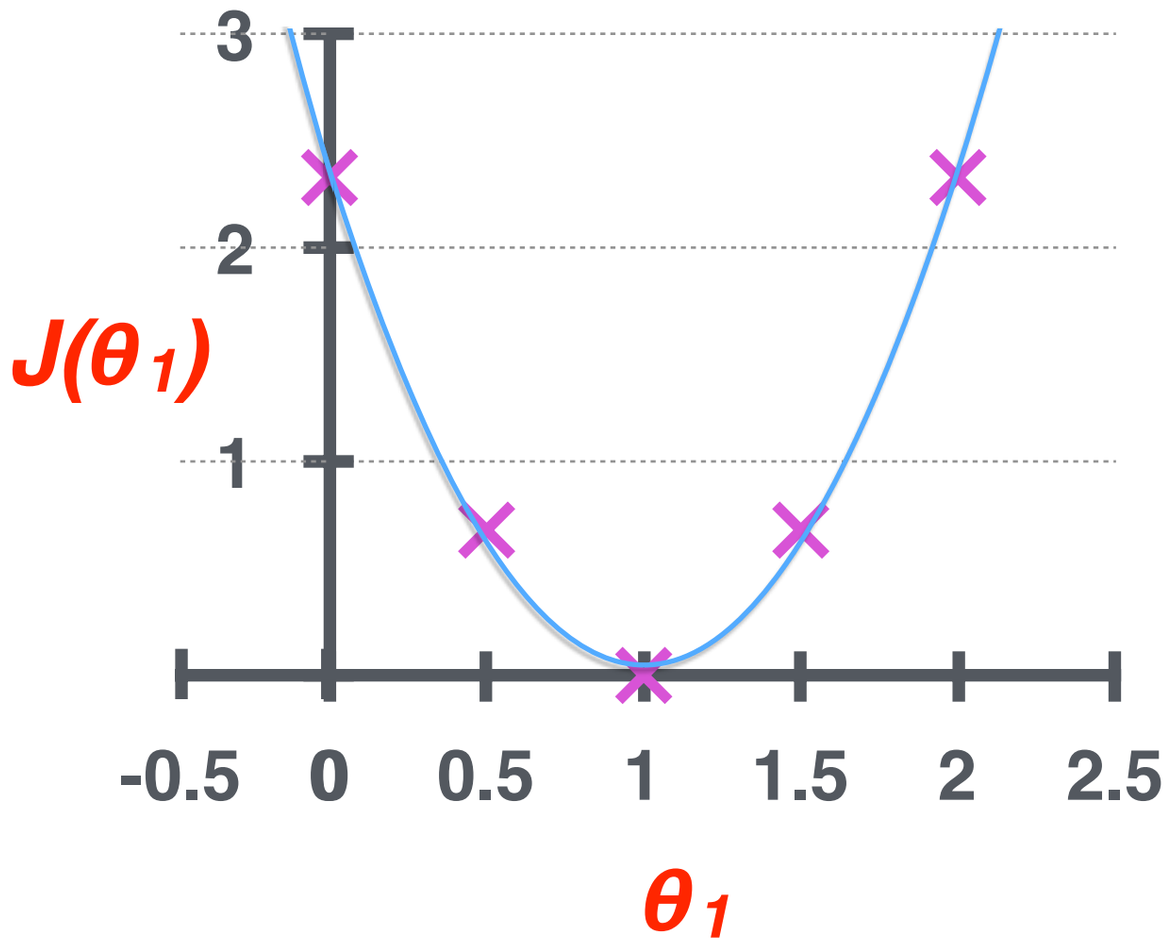
- **Outline:**
  - Start with some θ_0, θ_1
  - Keep changing θ_0, θ_1 to reduce $J(\theta_1, \theta_2)$ until we hopefully end up at a minimum

Gradient Descent



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

Gradient Descent

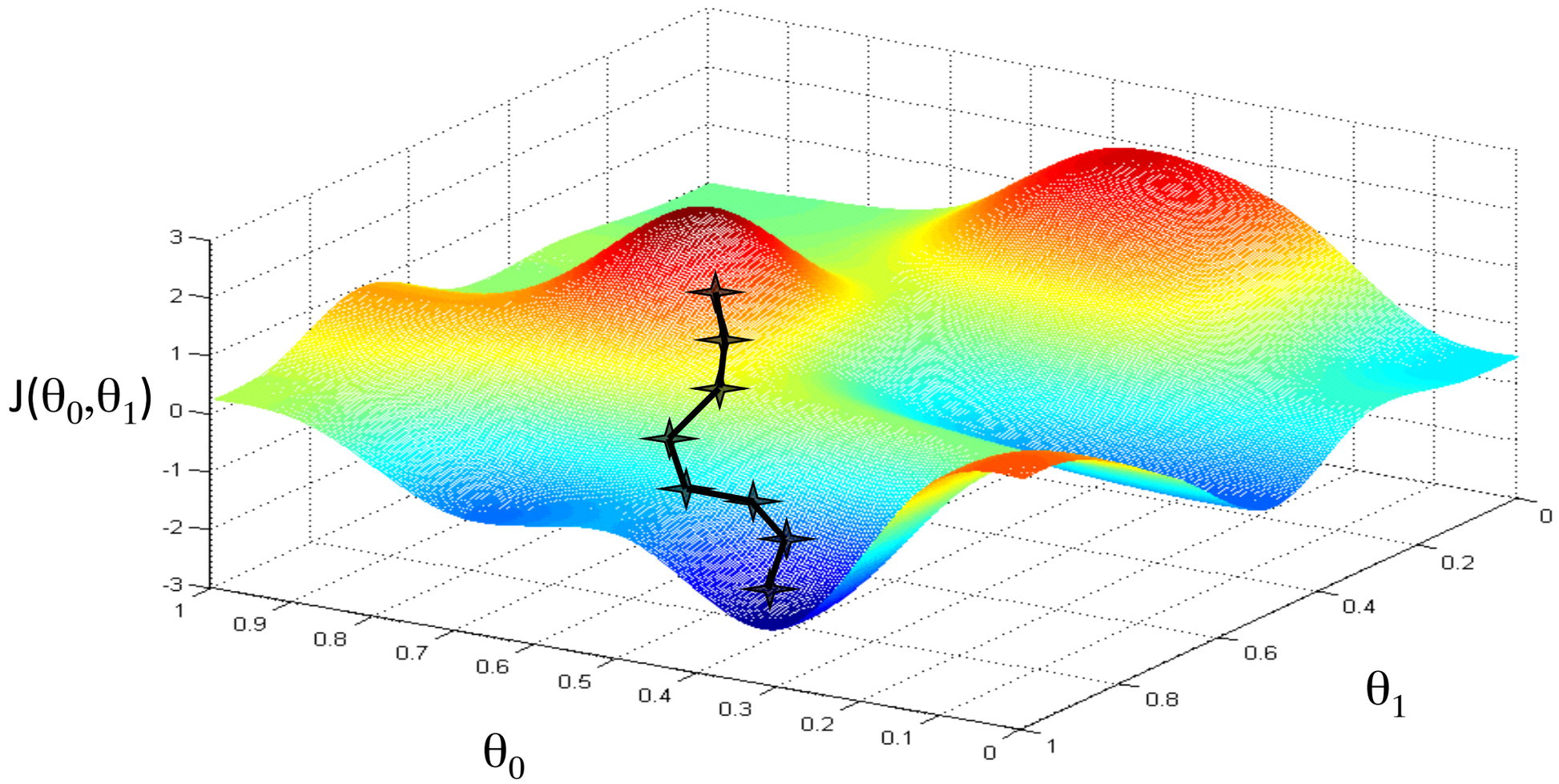


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

learning rate

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

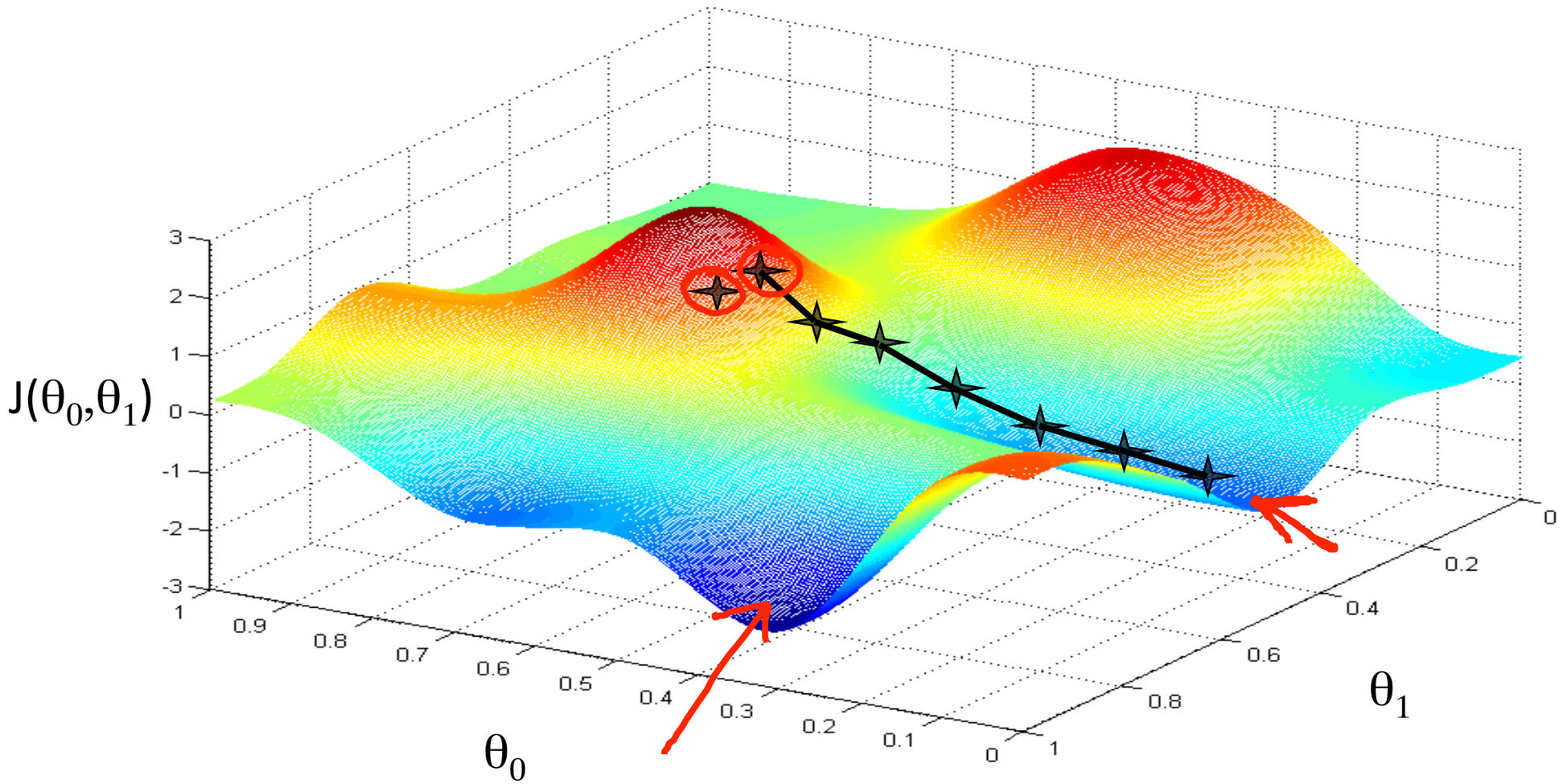
Gradient Descent



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

Andrew Ng

Gradient Descent



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

Andrew Ng

Gradient Descent

repeat until convergence {

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

(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

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

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

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

Linear Regression w/ one variable:

Gradient Descent

repeat until convergence {

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

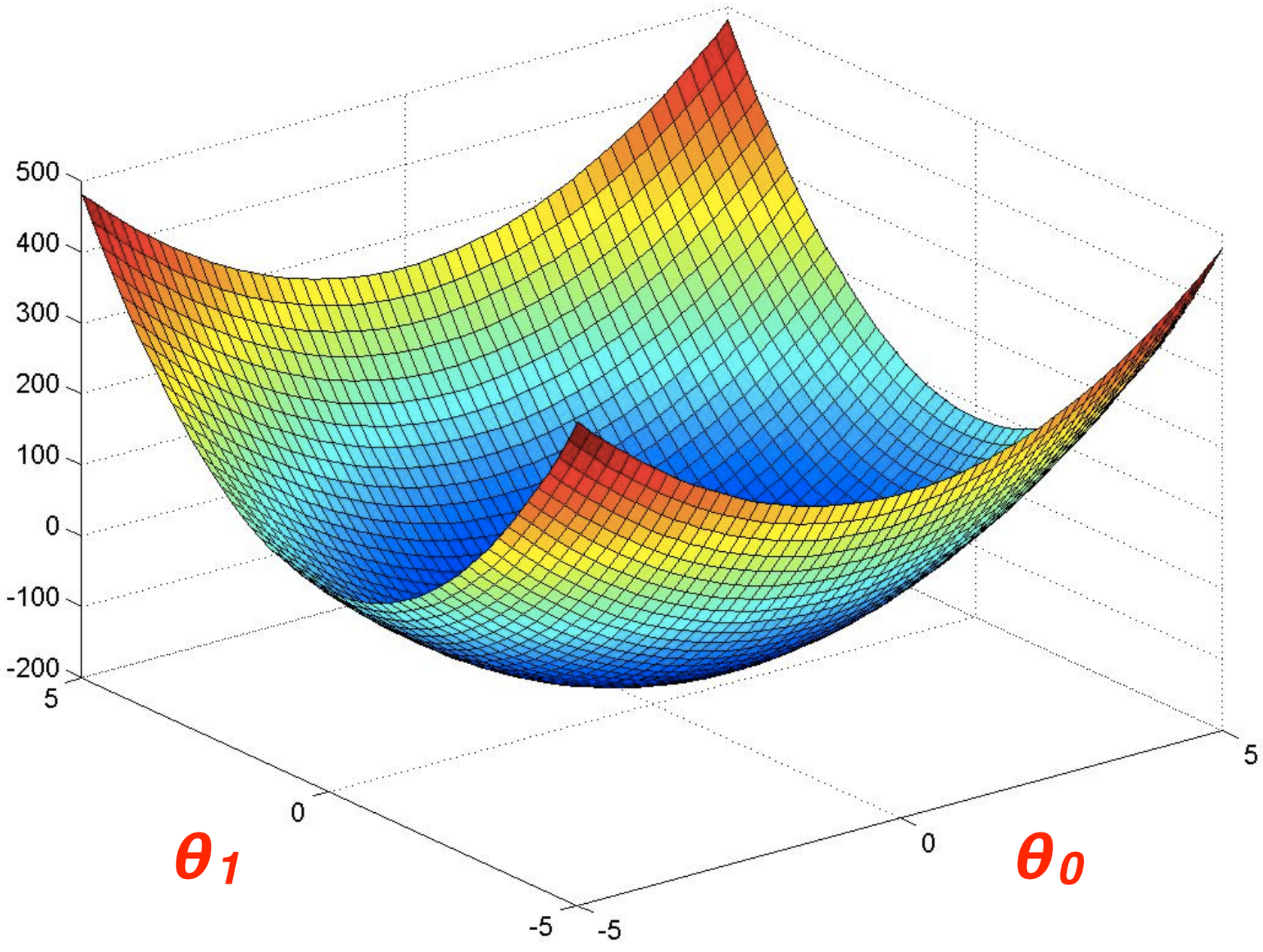
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) \cdot x_i$$

}

simultaneous update θ_0, θ_1

Linear Regression

cost function is convex



$J(\theta_1, \theta_2)$

θ_1

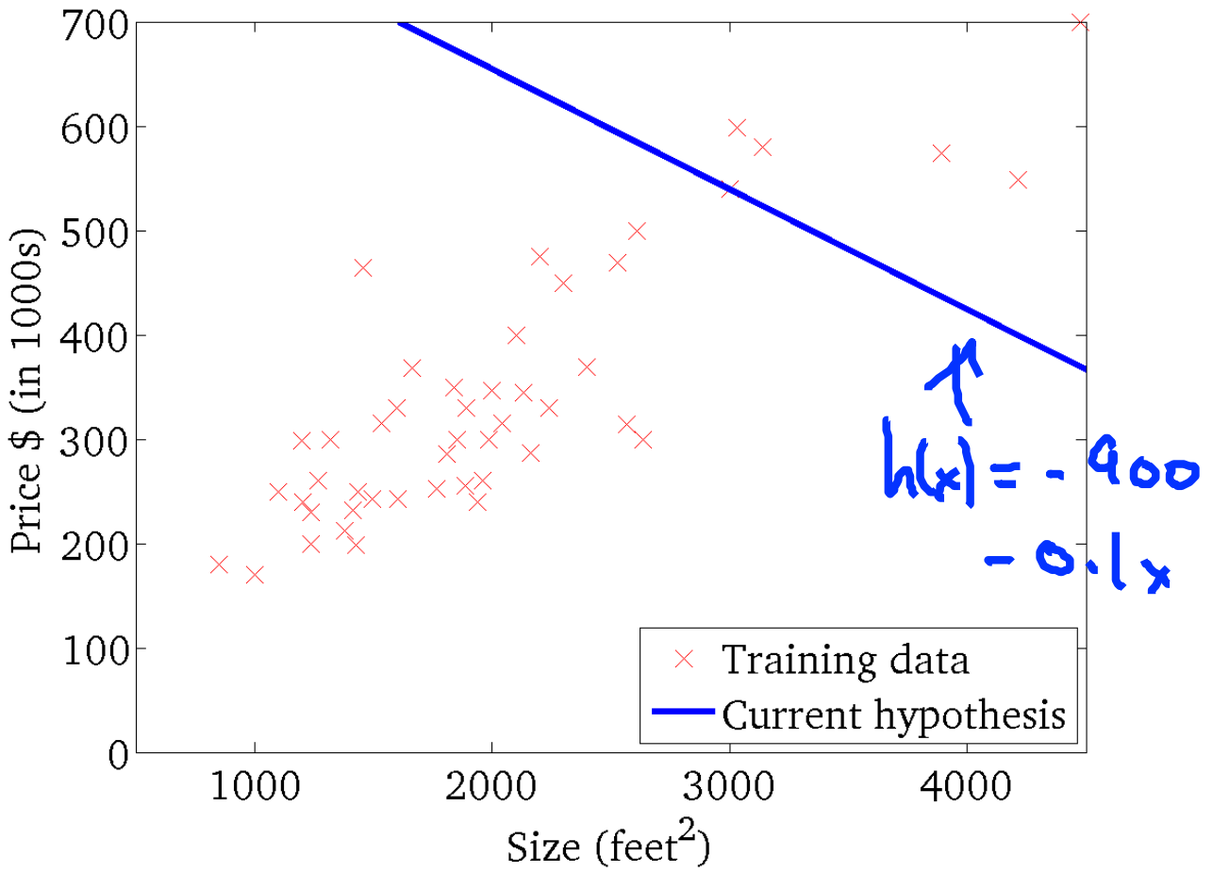
θ_0

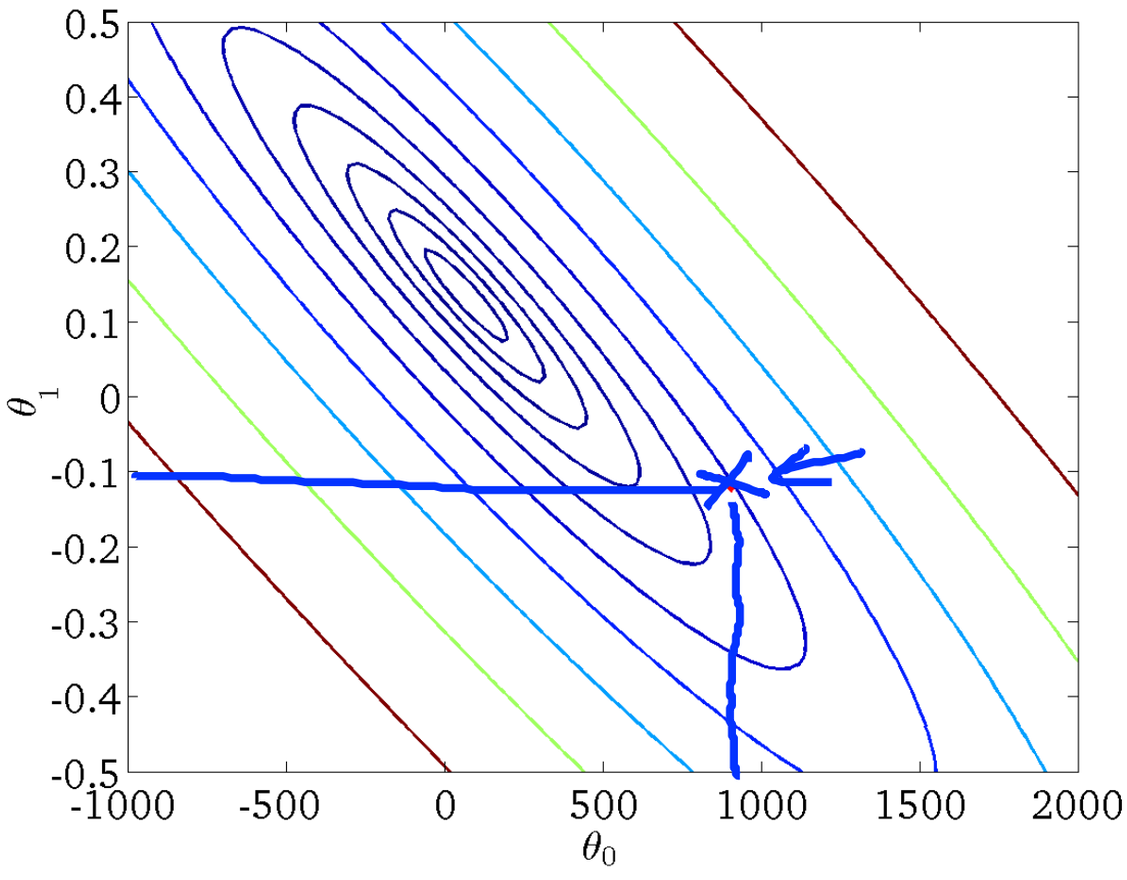
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



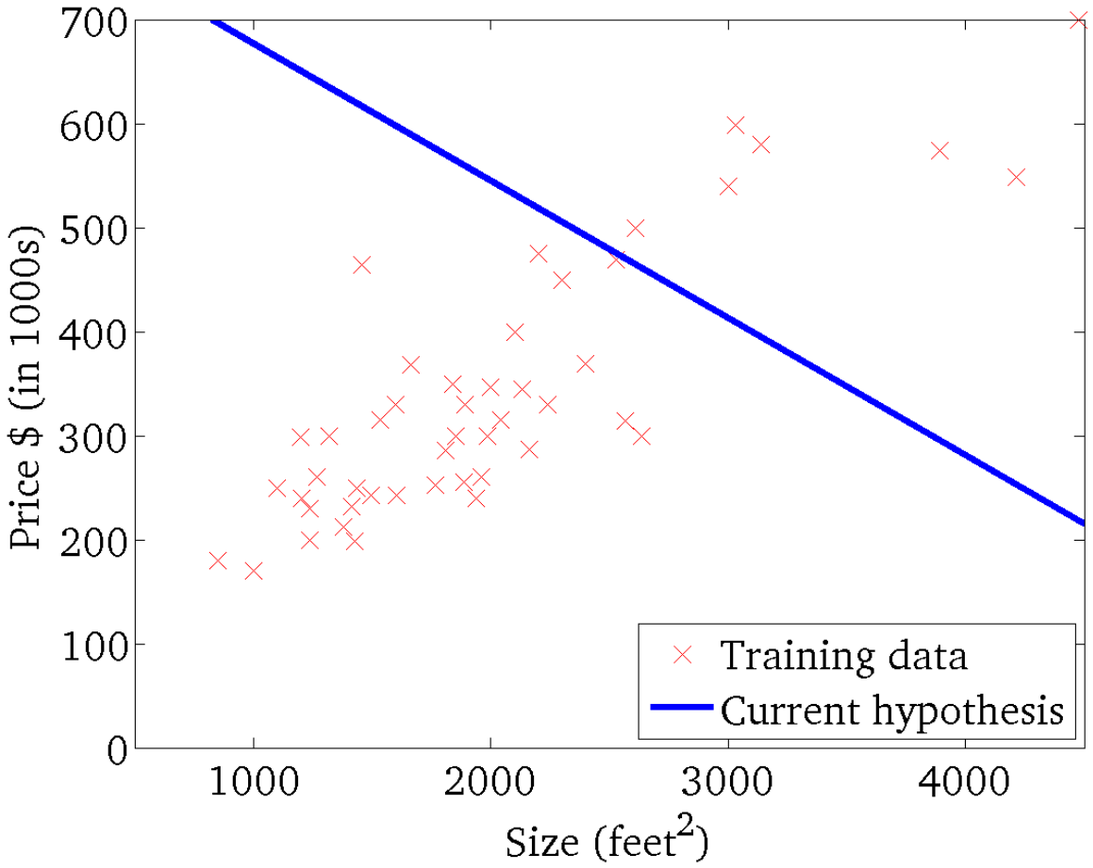


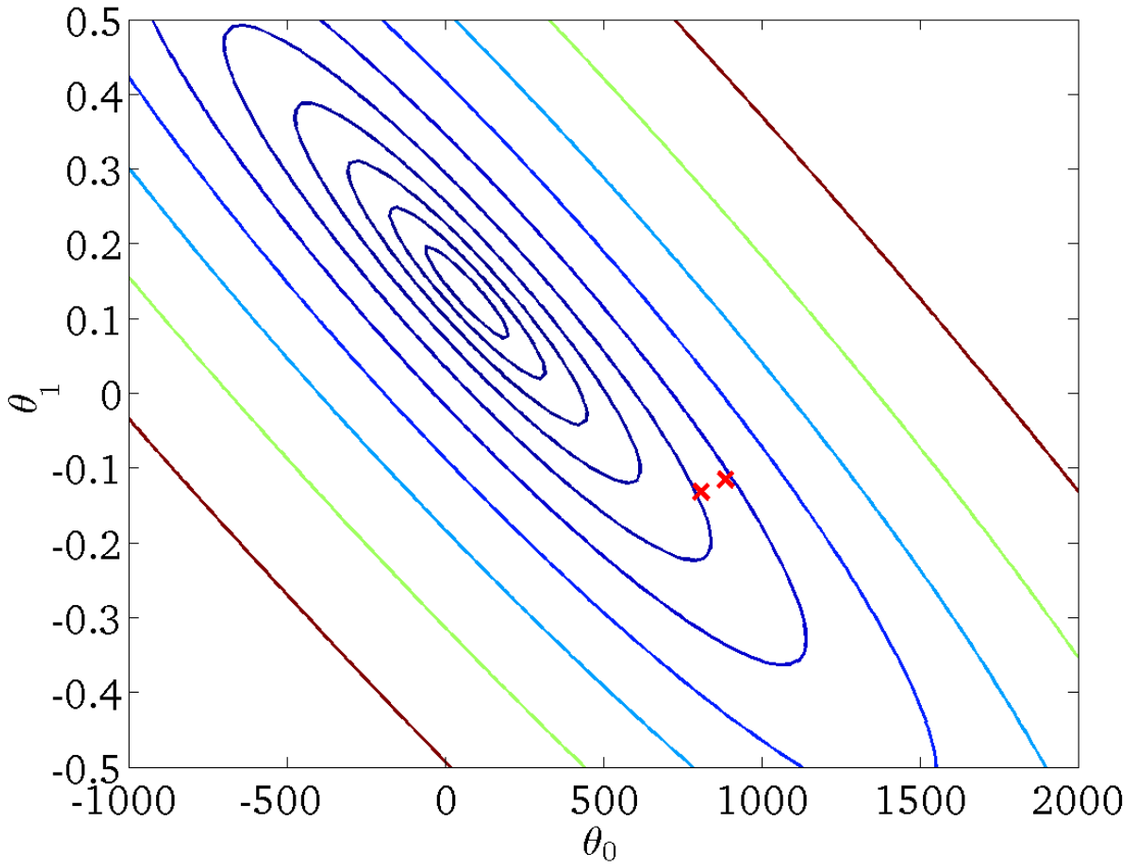
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



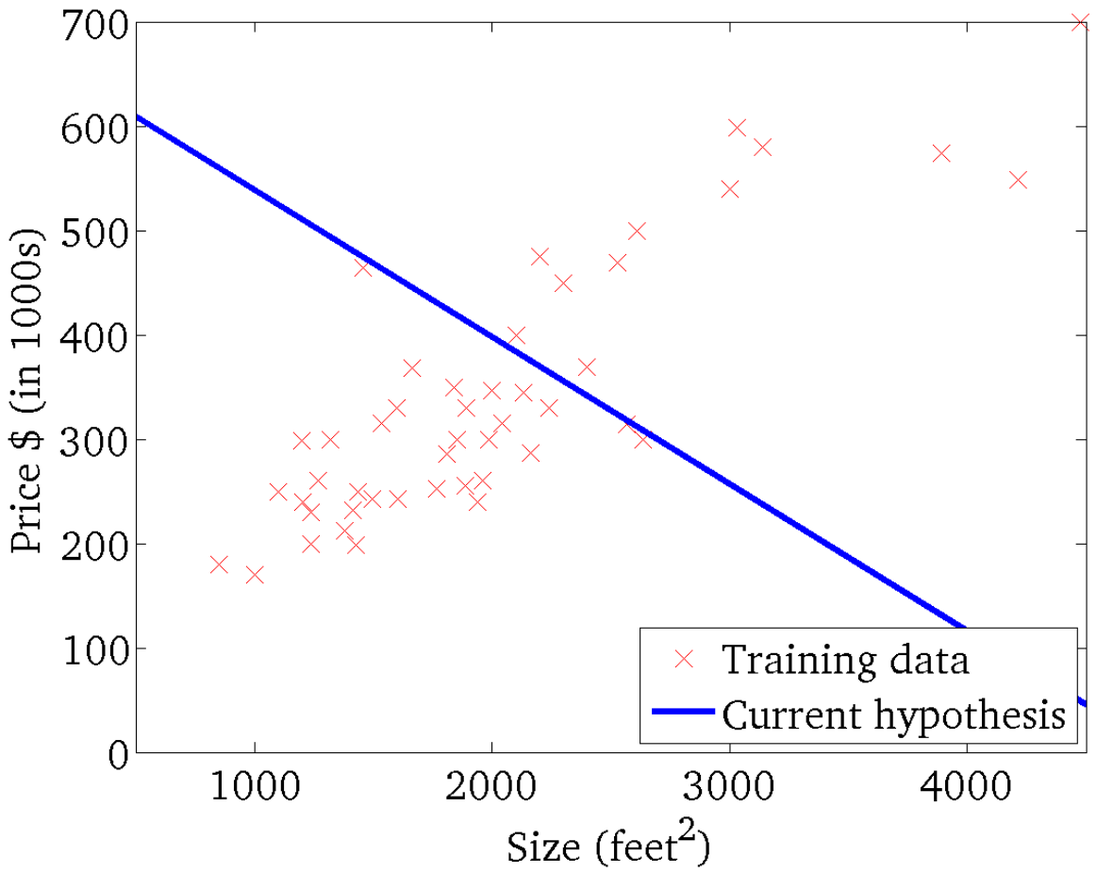


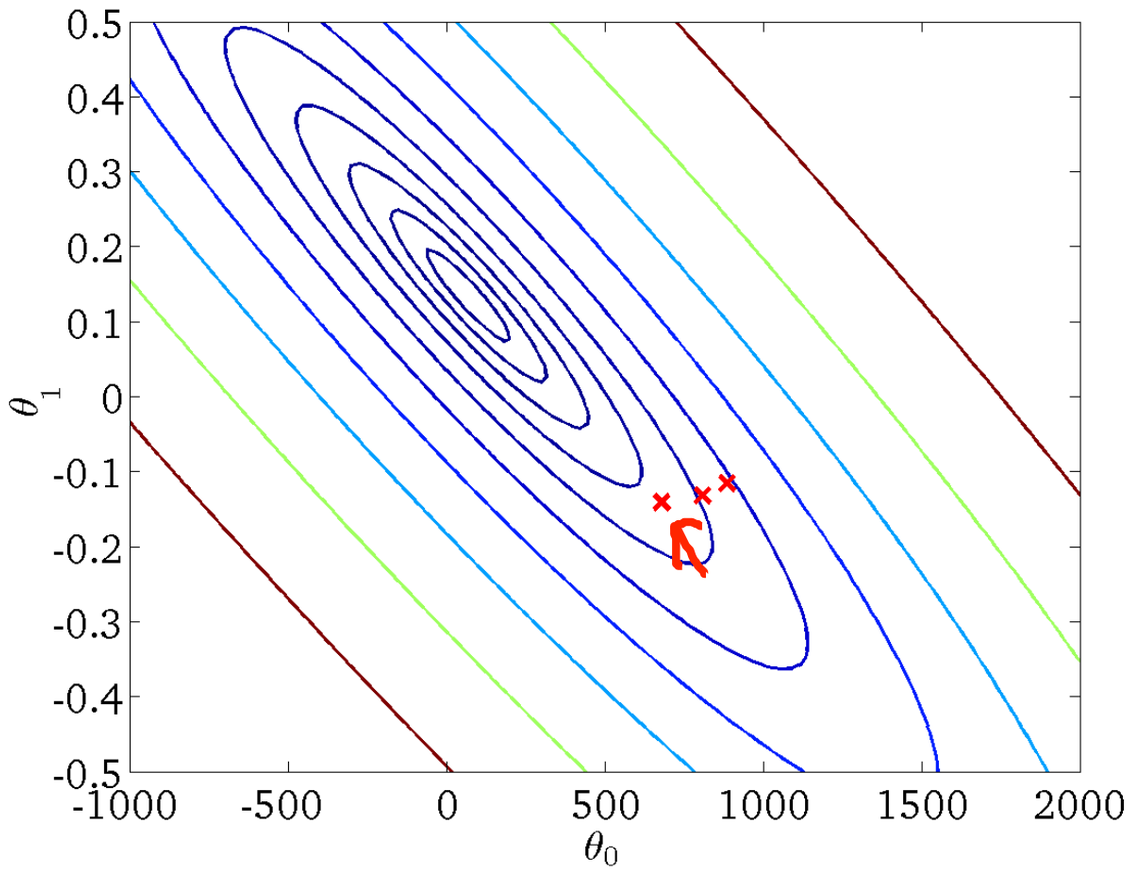
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



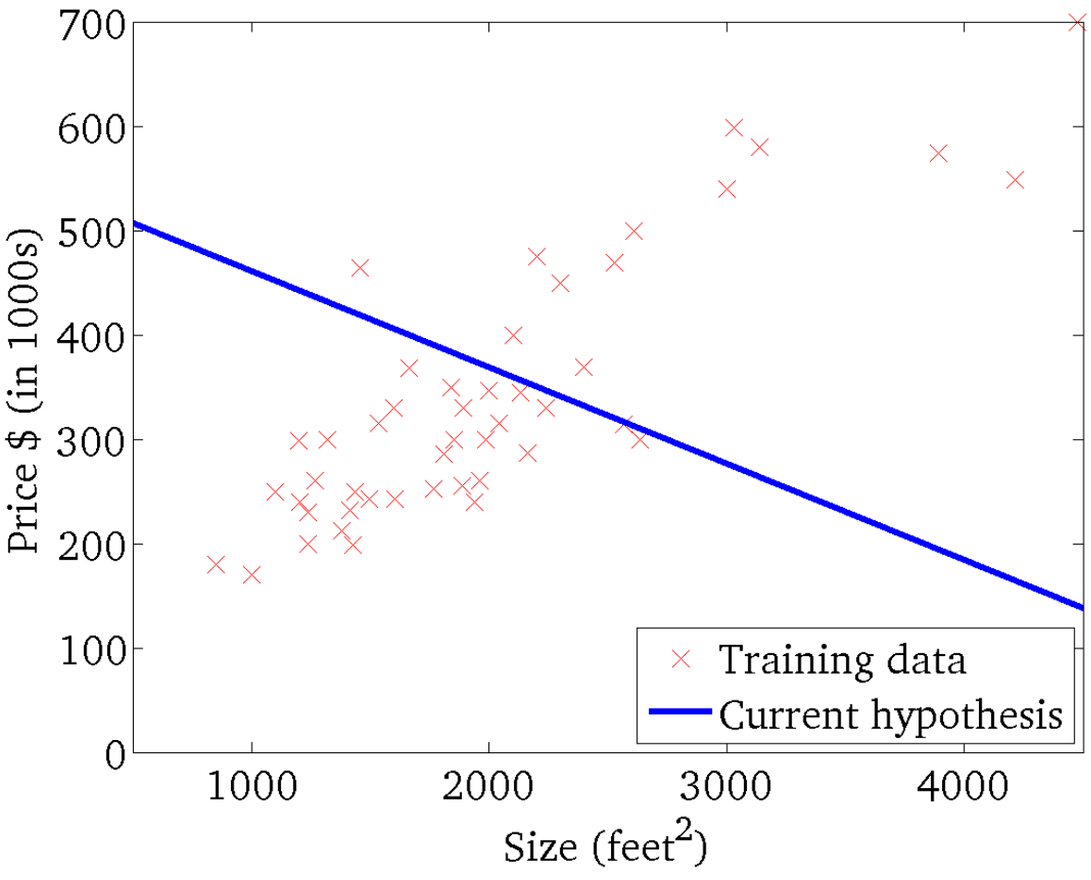


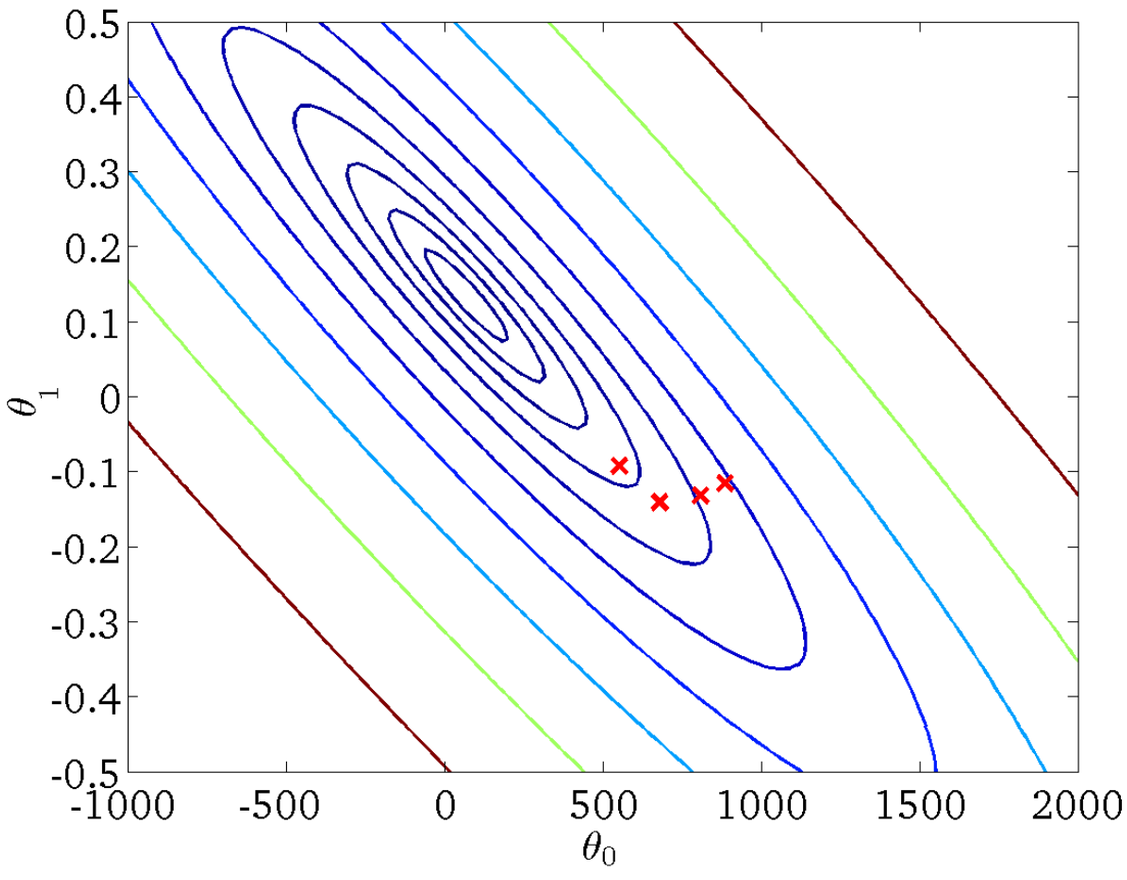
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



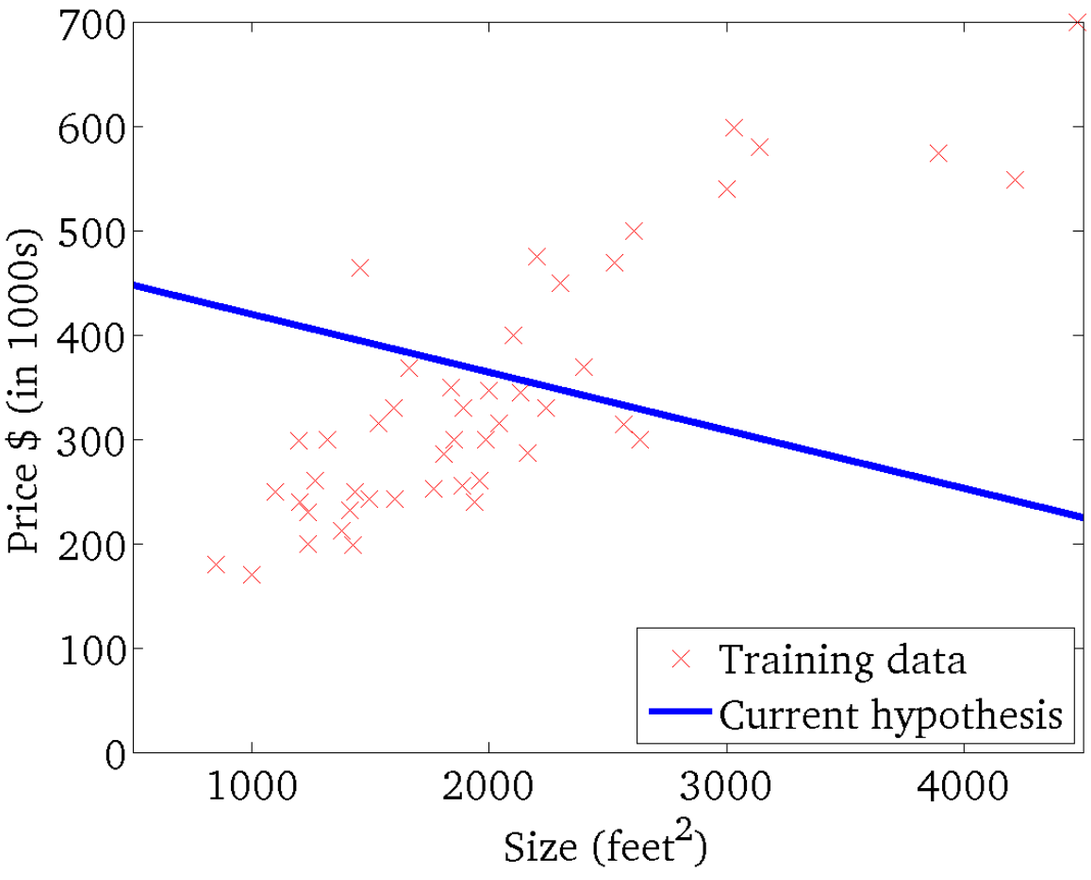


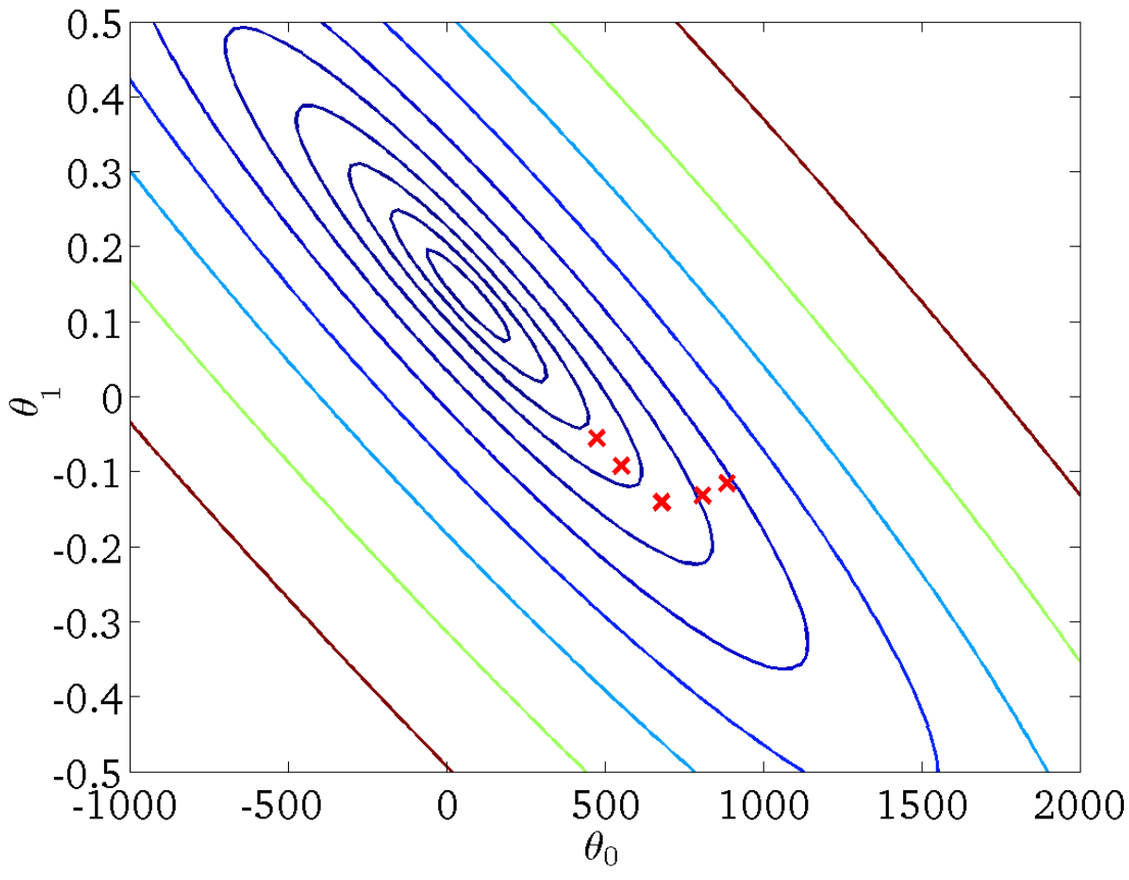
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



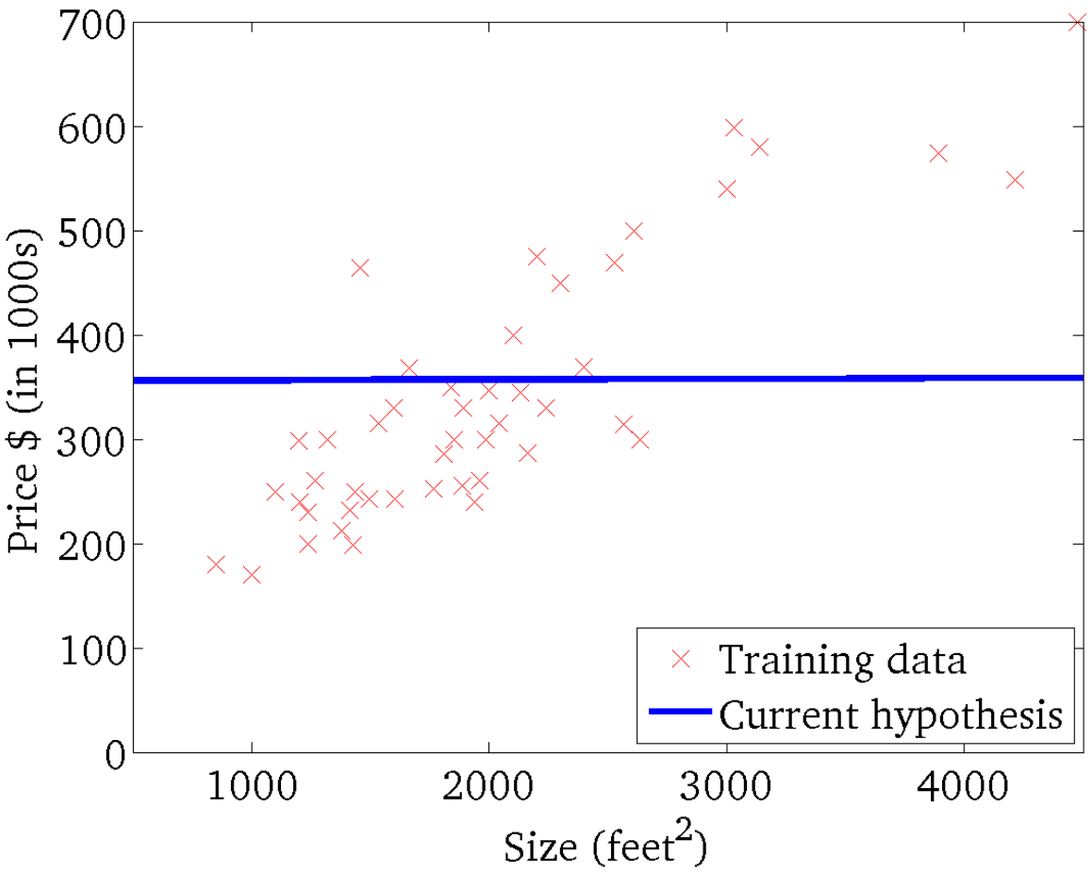


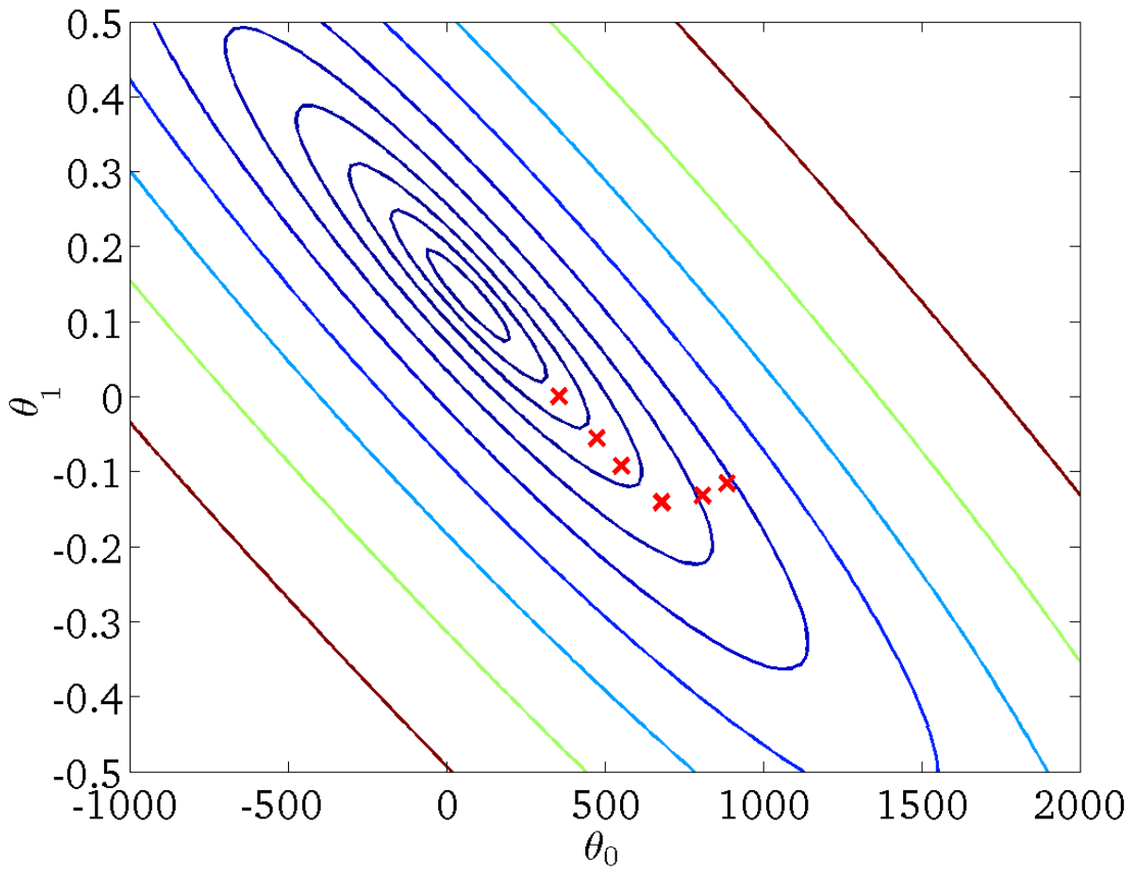
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



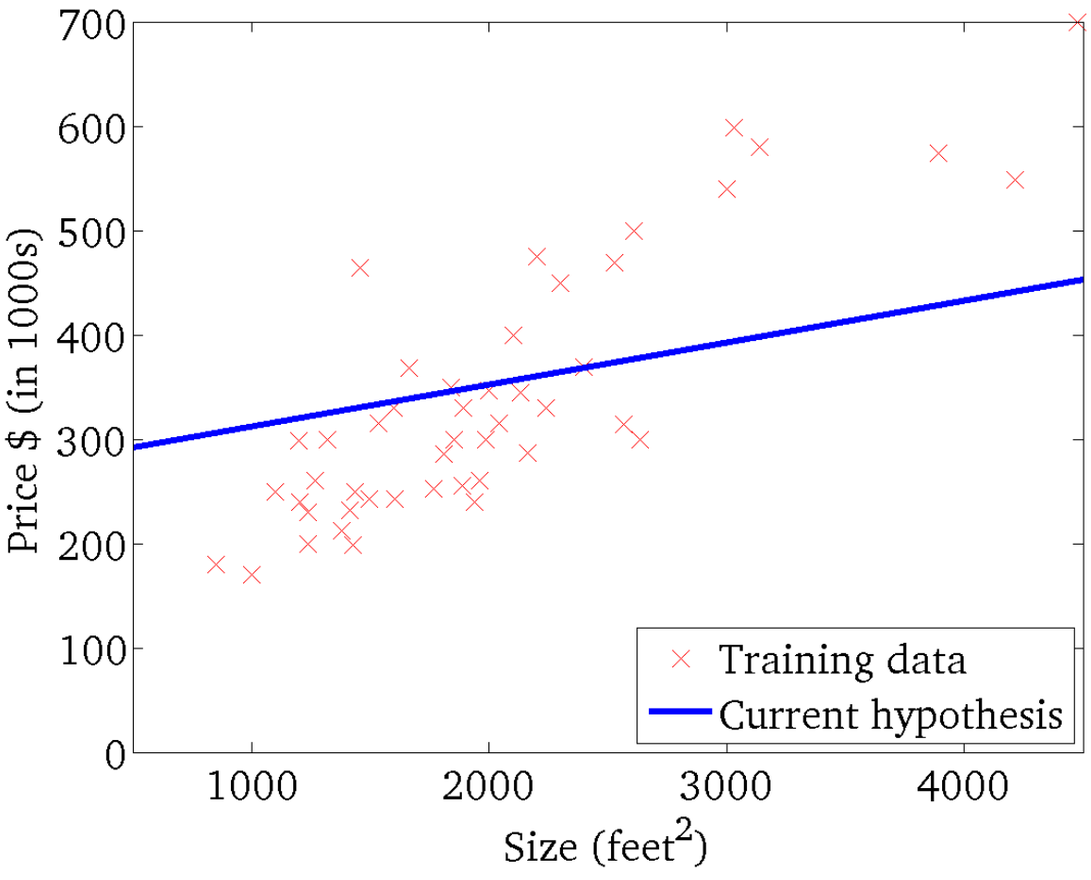


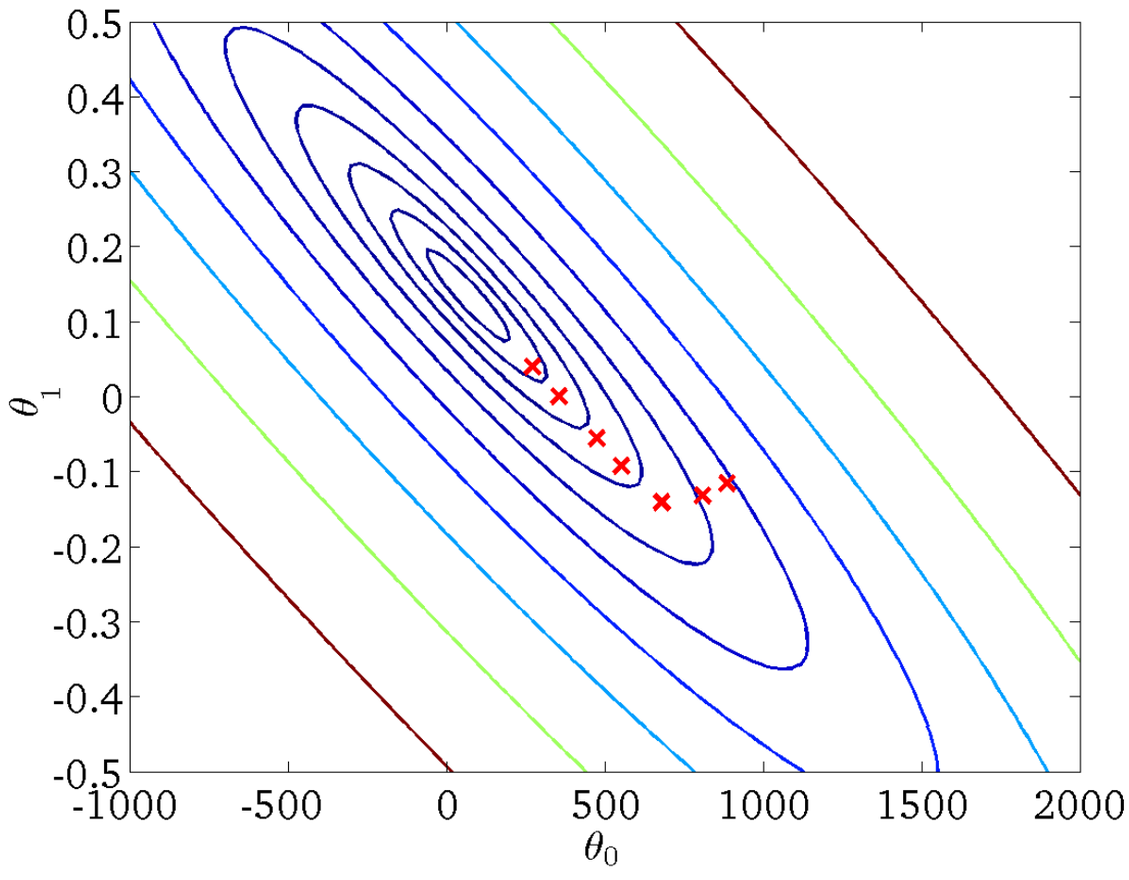
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



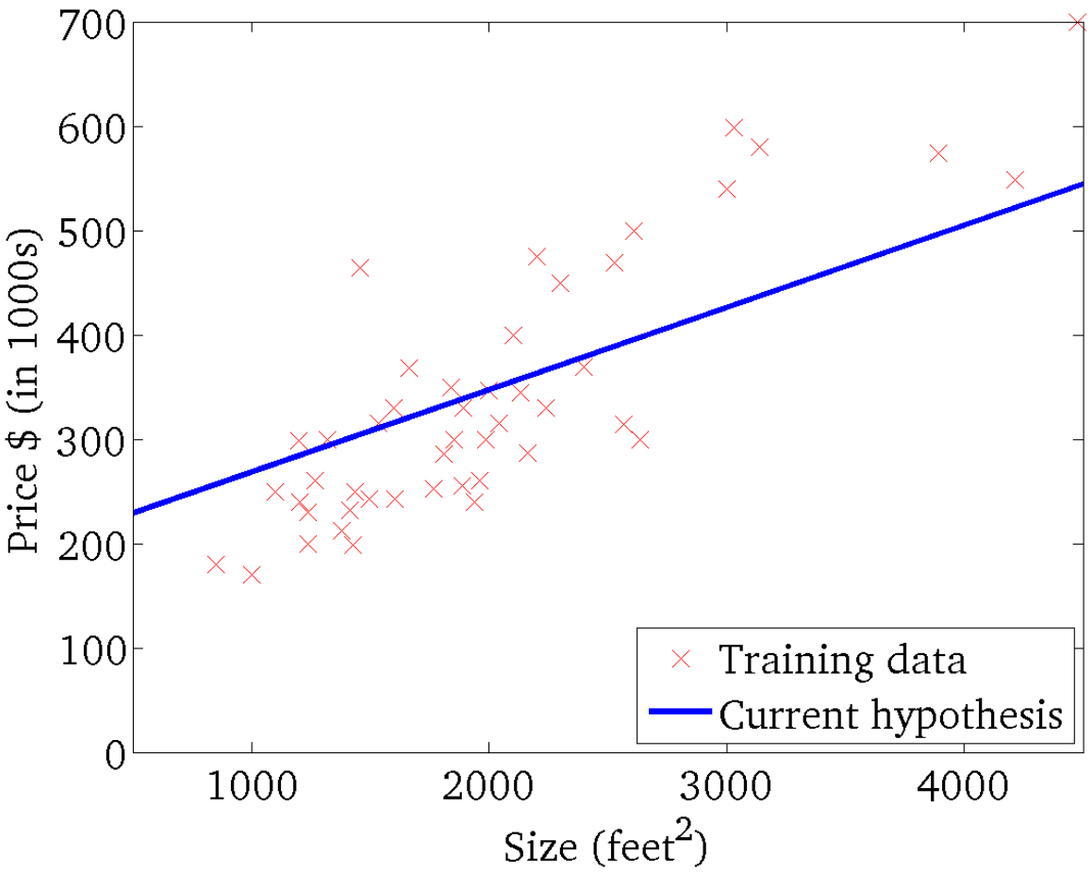


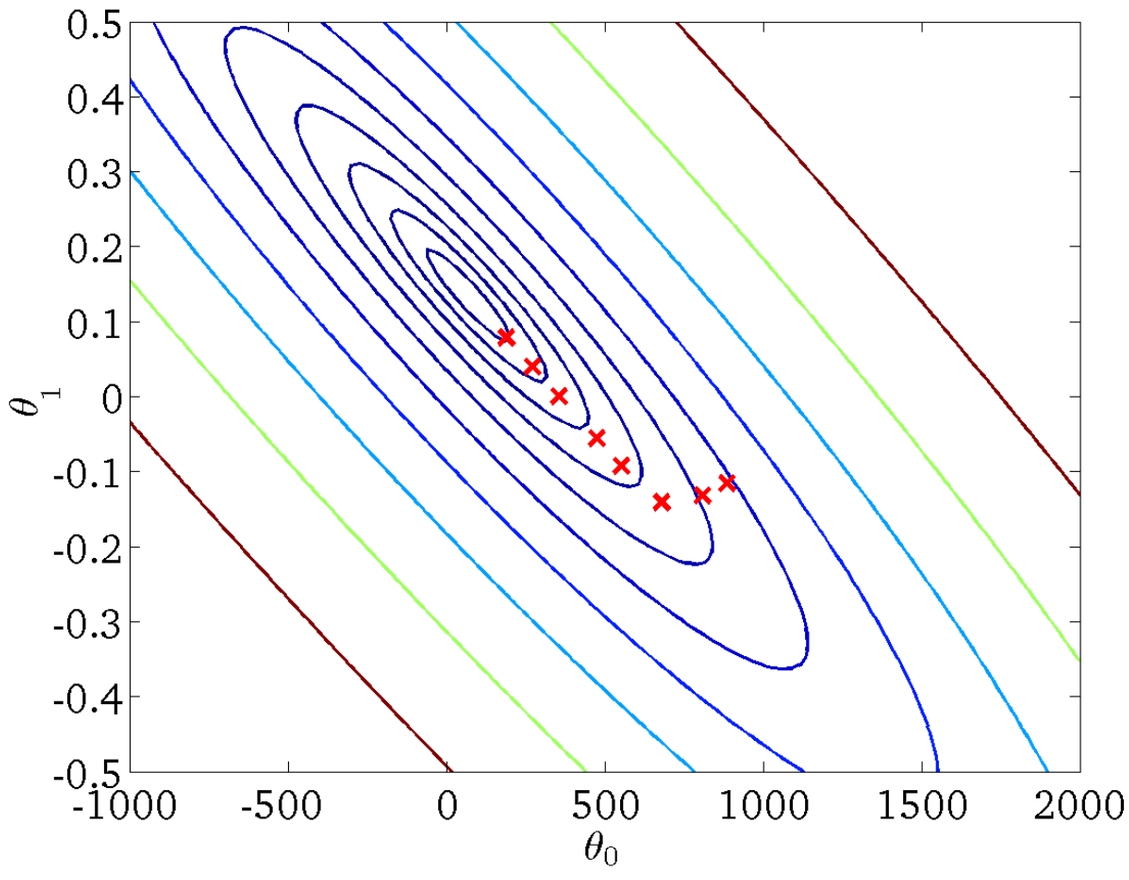
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



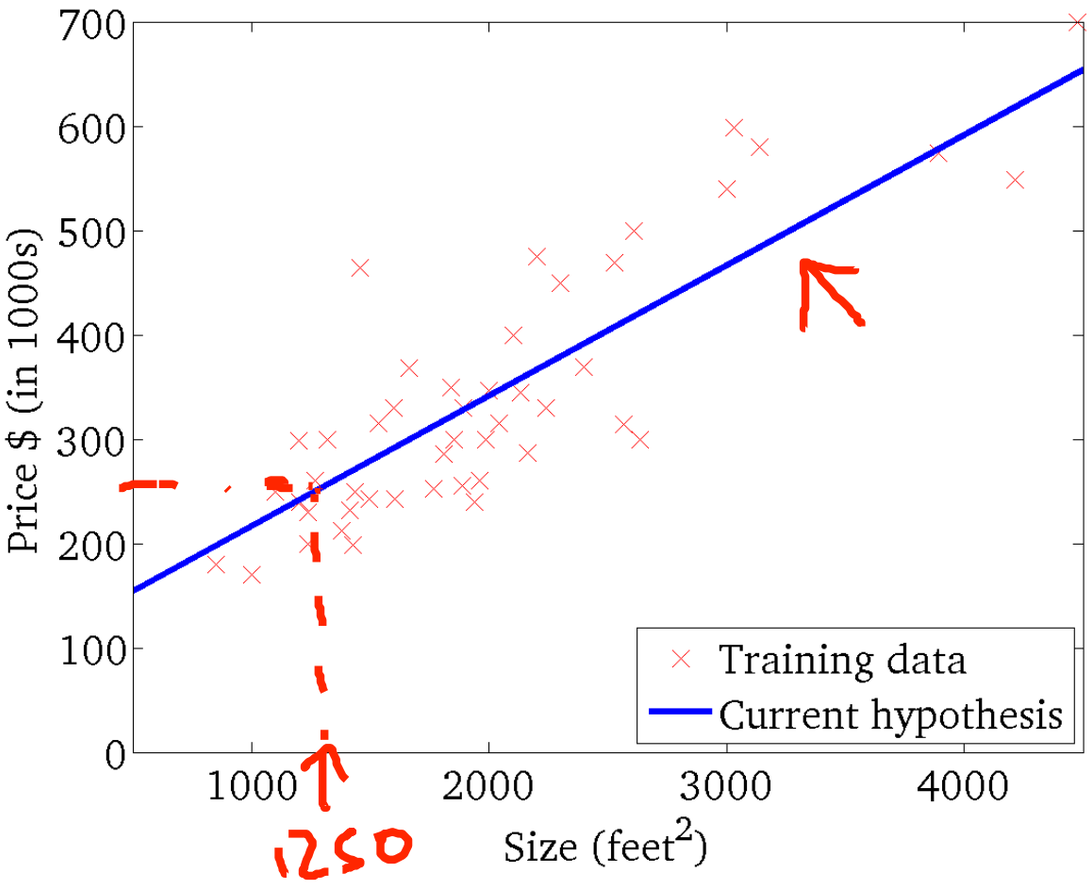


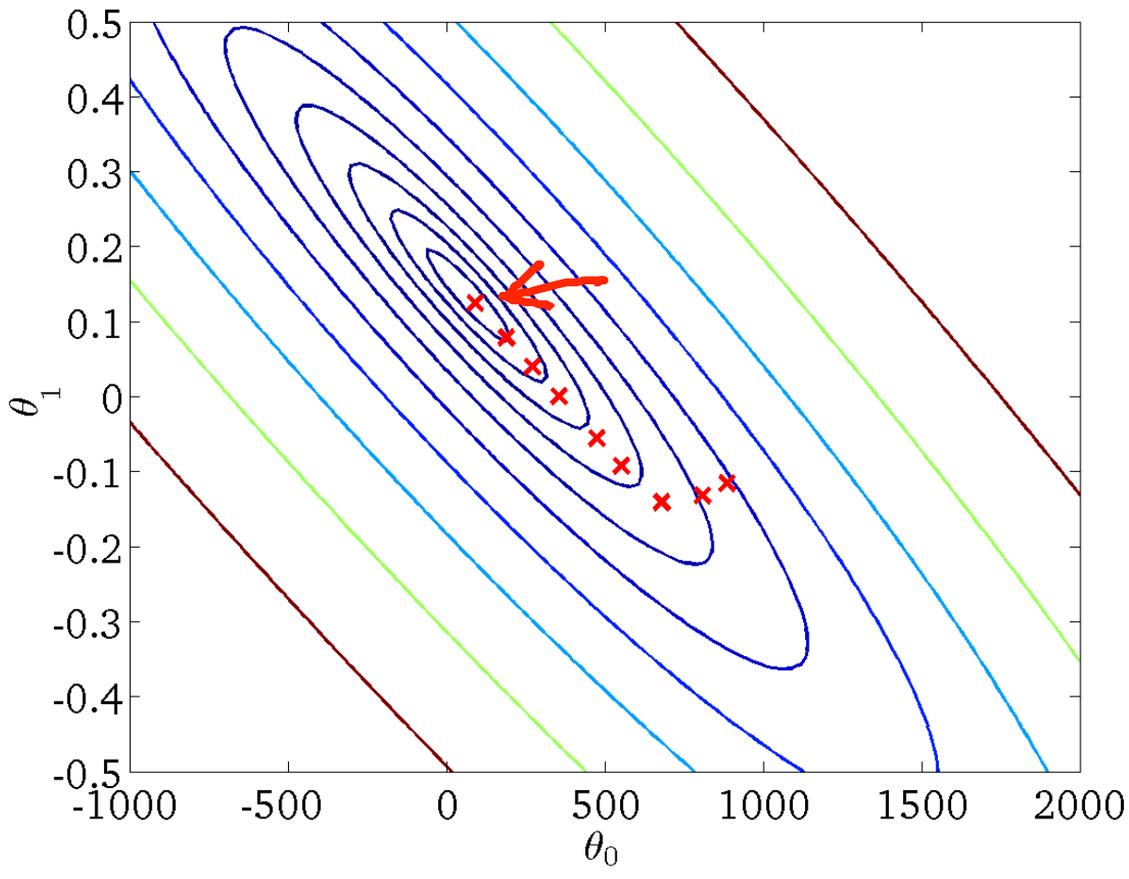
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)

$J(\theta_0, \theta_1)$

(function of the parameter θ_0, θ_1)



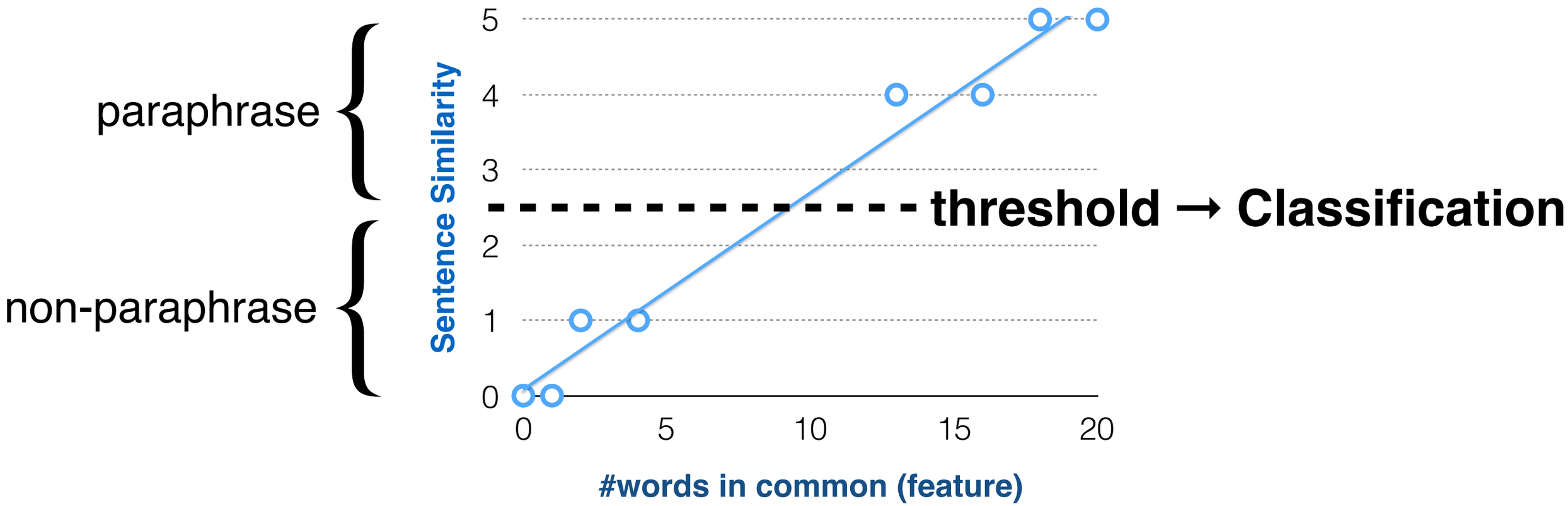


Batch Update

- 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**:

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

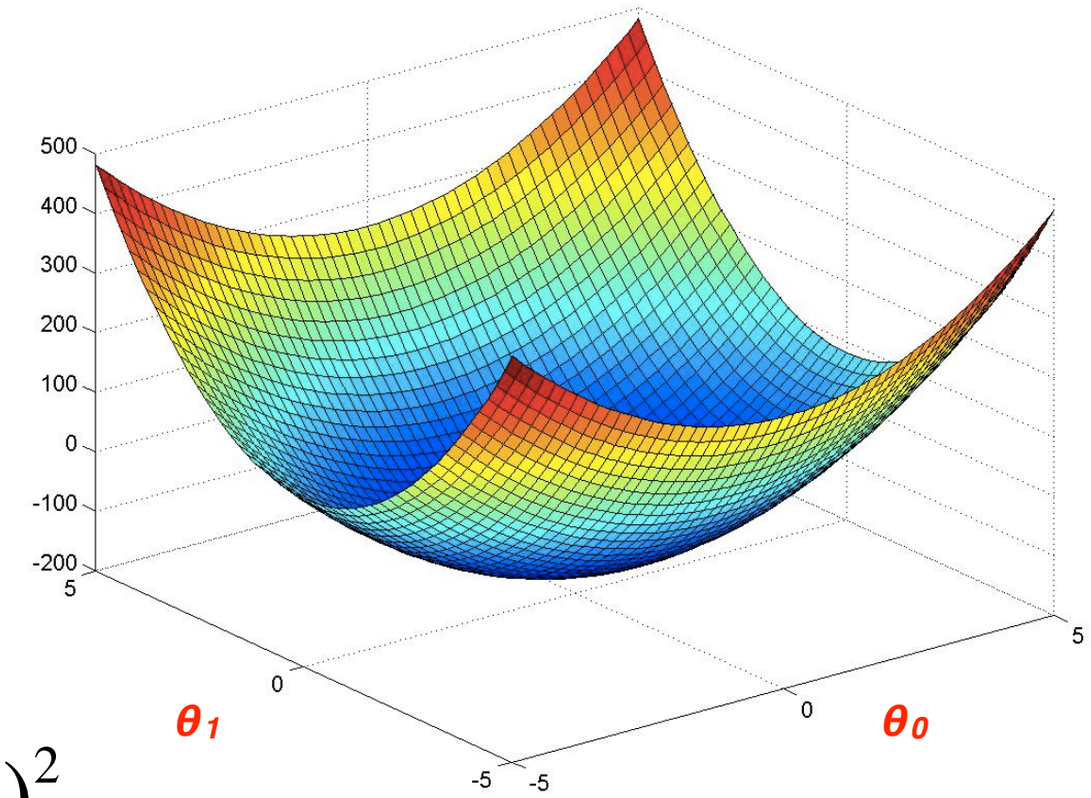
- **Parameters:**

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)

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

NLP Talk on Friday

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

socialmedia-class.org

Next Class:

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

socialmedia-class.org