



# Lecture 5: Word Embeddings

*Instructor: Swabha Swayamdipta*  
*USC CSCI 544 Applied NLP*  
*Sep 10, Fall 2024*



# Lecture Outline

- Quiz 1 Solution
- Recap
  - Logistic Regression
    - Model
    - Loss
    - Optimization
    - Regularization
  - Multinomial Logistic Regression
  - Word Embeddings

# Quiz 1: Solutions (Redacted)

# Announcements

# Announcements

# Announcements

- Start Z
- Today: Group Formation Deadline (deadline: midnight)
  - As of now: we have 48 teams?
  - We might make some adjustments for students who registered late
    - Email us:

# Announcements

- Start Z
- Today: Group Formation Deadline (deadline: midnight)
  - As of now: we have 48 teams?
  - We might make some adjustments for students who registered late
    - Email us:
- Next Week:
  - Quiz 2
  - HW 1 Due

# Announcements

- Start Z
- Today: Group Formation Deadline (deadline: midnight)
  - As of now: we have 48 teams?
  - We might make some adjustments for students who registered late
    - Email us:
- Next Week:
  - Quiz 2
  - HW 1 Due
- All quiz dates were changed slightly to give some space between them

# Lecture Outline

- Quiz 1 Solution
- Recap
  - Logistic Regression
    - Model
    - Loss
    - Optimization
    - Regularization
  - Multinomial Logistic Regression
  - Word Embeddings

Recap:  
Model (Logistic Regression)  
+ Loss + Optimization

# Ingredients of Supervised Machine Learning

## I. Data as pairs $(x^{(i)}, y^{(i)})$ s.t $i \in \{1 \dots N\}$

- $x^{(i)}$  usually represented by a feature vector  $\mathbf{x}^{(i)} = [x_1, x_2, \dots, x_d]$ ,
- e.g. word embeddings

## II. Model

- A classification function that computes  $\hat{y}$ , the estimated class, via  $p(y|x)$
- e.g. logistic regression, naïve Bayes

## III. Loss

- An objective function for learning
- e.g. cross-entropy loss,  $L_{CE}$

## IV. Optimization

- An algorithm for optimizing the objective function
- e.g. stochastic gradient descent

## V. Inference / Evaluation

Learning Phase

# How to get the right $y$ ?

- For each feature  $x_i$ , introduce a weight  $w_i$ , which determines the importance of  $x_i$ 
  - Sometimes we have a bias term,  $b$  or  $w_0$ , which is just another weight not associated to any feature
  - Together, all parameters can be termed as  $\theta = [w; b]$
- We consider the weighted sum of all features and the bias

$$z = \left( \sum_d w_d x_d + b \right)$$

$$= \mathbf{w} \cdot \mathbf{x} + b$$

If high,  $\hat{y} = 1$

If low,  $\hat{y} = 0$

# How to get the right $y$ ?

- For each feature  $x_i$ , introduce a weight  $w_i$ , which determines the importance of  $x_i$ 
  - Sometimes we have a bias term,  $b$  or  $w_0$ , which is just another weight not associated to any feature
  - Together, all parameters can be termed as  $\theta = [w; b]$
- We consider the weighted sum of all features and the bias

$$z = \left( \sum_d w_d x_d + b \right)$$

$$= \mathbf{w} \cdot \mathbf{x} + b$$

If high,  $\hat{y} = 1$

If low,  $\hat{y} = 0$

But how to determine the threshold?

# How to get the right $y$ ?

- For each feature  $x_i$ , introduce a weight  $w_i$ , which determines the importance of  $x_i$ 
  - Sometimes we have a bias term,  $b$  or  $w_0$ , which is just another weight not associated to any feature
  - Together, all parameters can be termed as  $\theta = [w; b]$
- We consider the weighted sum of all features and the bias

$$z = \left( \sum_d w_d x_d + b \right)$$
$$= \mathbf{w} \cdot \mathbf{x} + b$$

If high,  $\hat{y} = 1$

If low,  $\hat{y} = 0$

But how to determine the threshold?

We need probabilistic models!

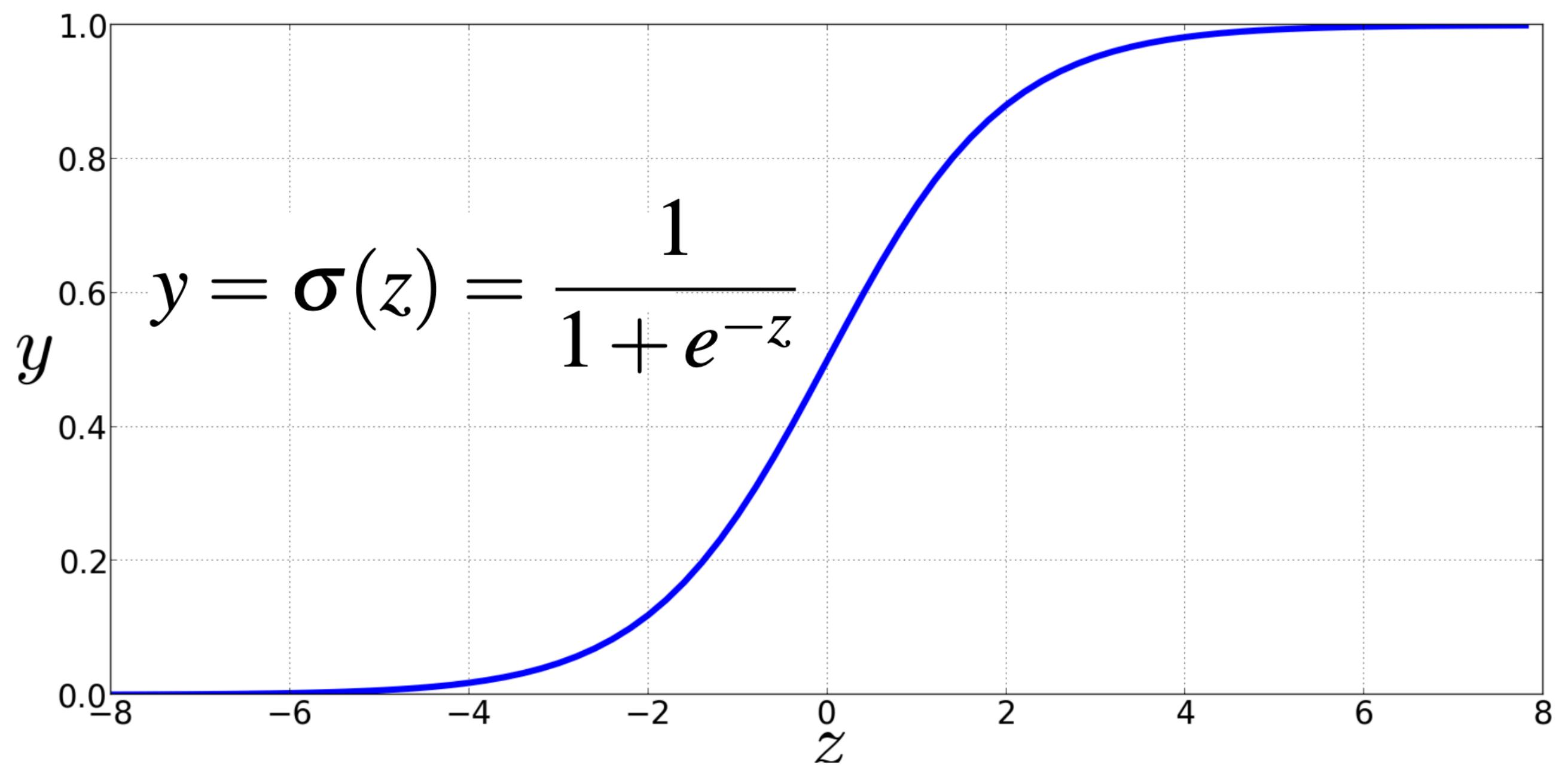
$$P(y = 1 | \mathbf{x}; \theta)$$

$$P(y = 0 | \mathbf{x}; \theta)$$

# Solution: Squish it into the 0-1 range

$$z = \mathbf{w} \cdot \mathbf{x} + b \quad z \in \mathbb{R}$$

- Sigmoid Function,  $\sigma(\cdot)$ 
  - Non-linear!
- Compute  $z$  and then pass it through the sigmoid function
- Treat it as a probability!
- Also, a differentiable function, which makes it a good candidate for optimization (more on this later!)



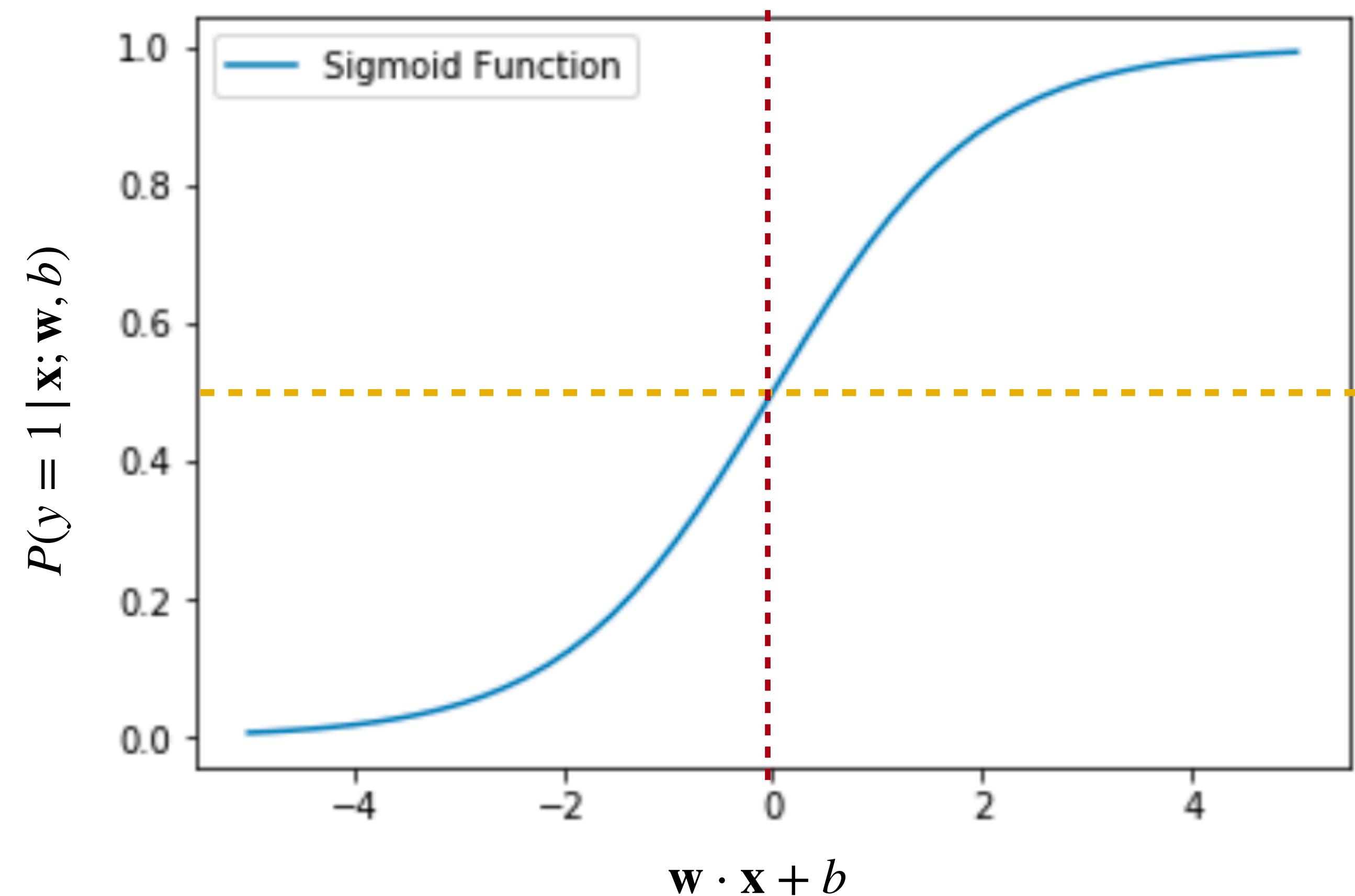
$$P(y = 1 | \mathbf{x}; \theta) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) \quad P(y = 0 | \mathbf{x}; \theta) = \sigma(-(\mathbf{w} \cdot \mathbf{x} + b))$$

# Classification Decision

$$\hat{y} = \begin{cases} 1 & \text{if } p(y = 1 | x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Decision Boundary

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \end{cases}$$



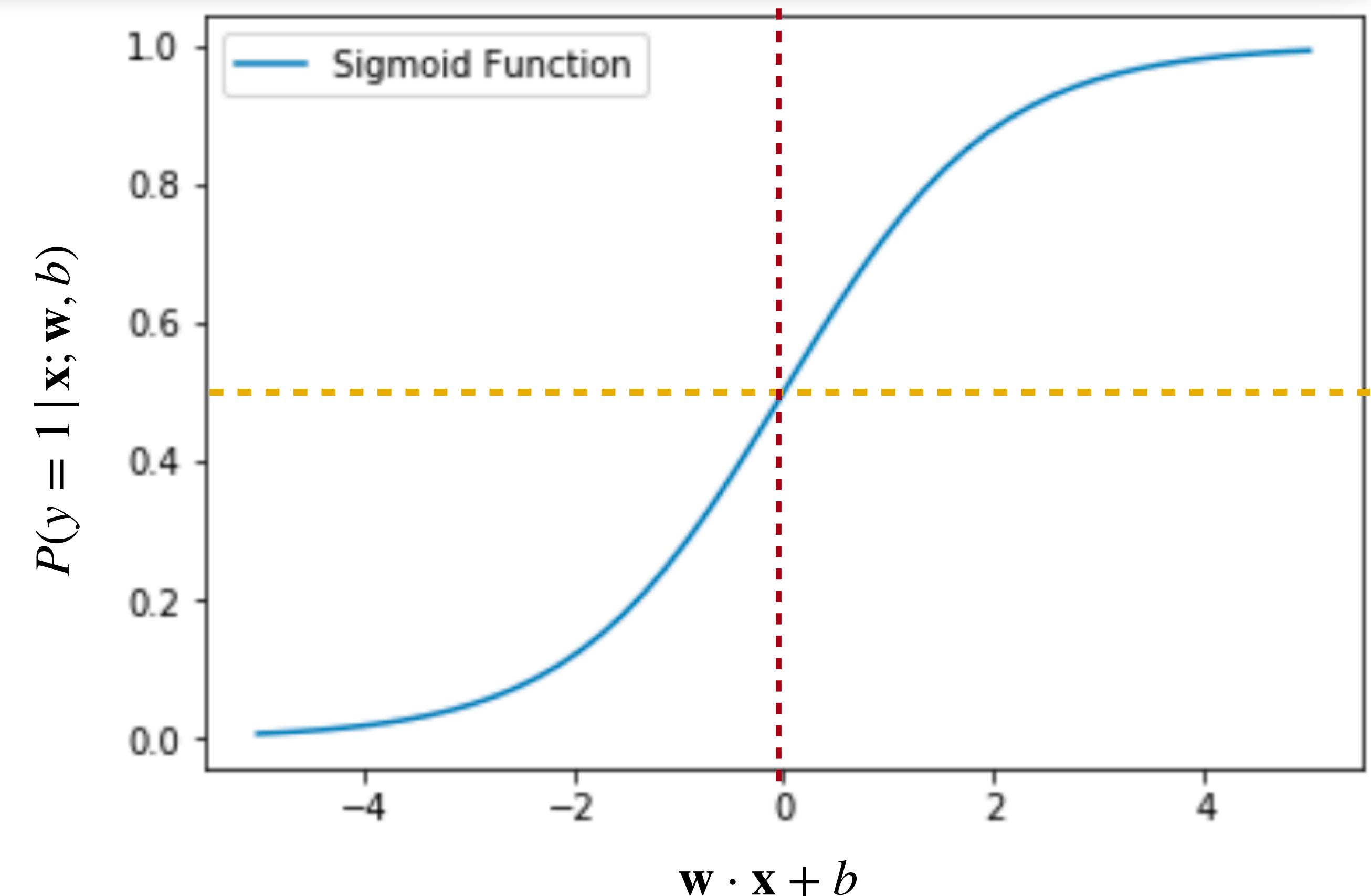
# Classification Decision

Inference under a Logistic Regression Model

$$\hat{y} = \begin{cases} 1 & \text{if } p(y = 1 | x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Decision Boundary

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \end{cases}$$



# Classification Decision

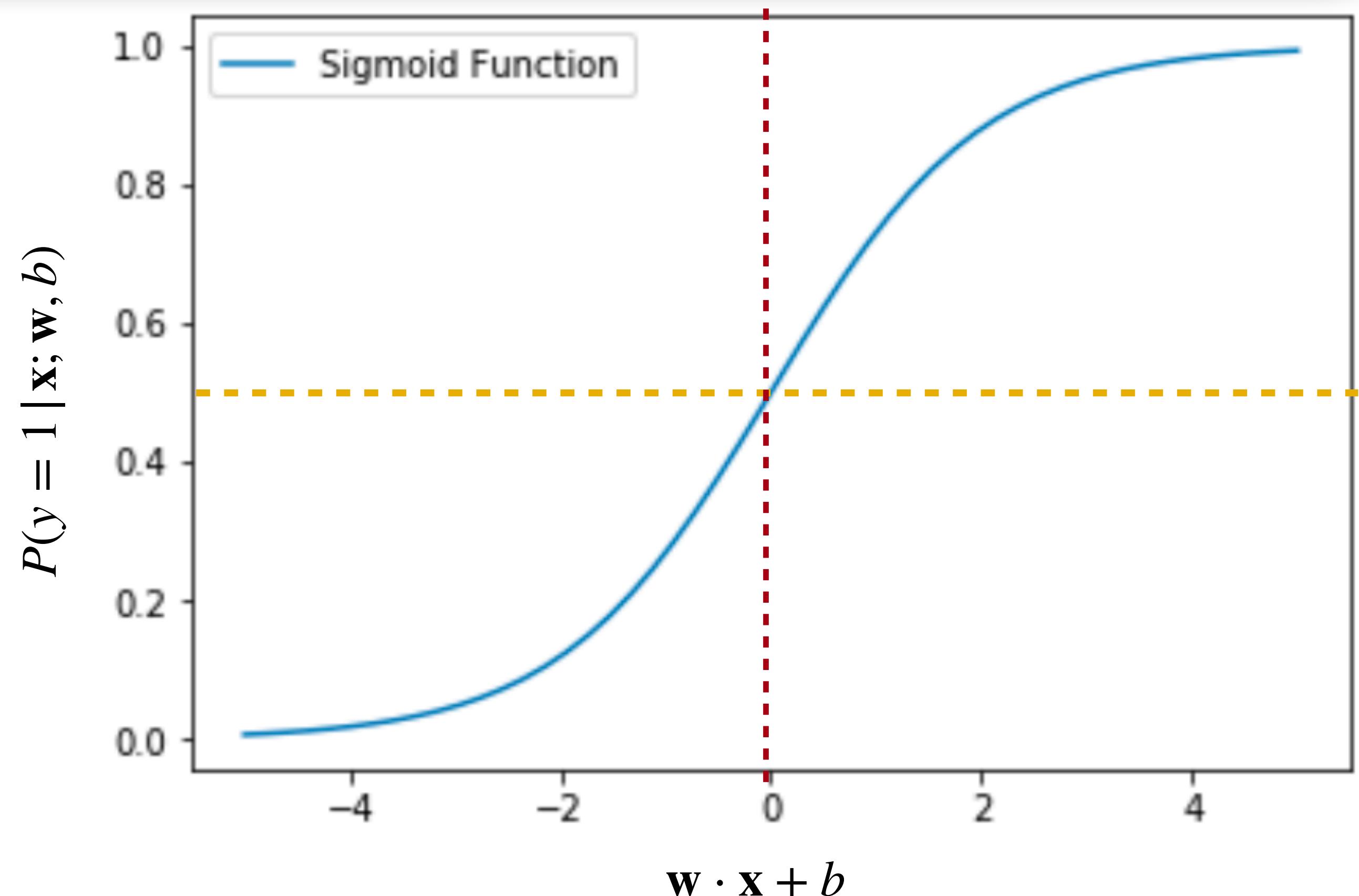
Inference under a Logistic Regression Model

$$\hat{y} = \begin{cases} 1 & \text{if } p(y = 1 | x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Decision Boundary

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \end{cases}$$

Often  $\hat{y}$  is used to indicate probability:



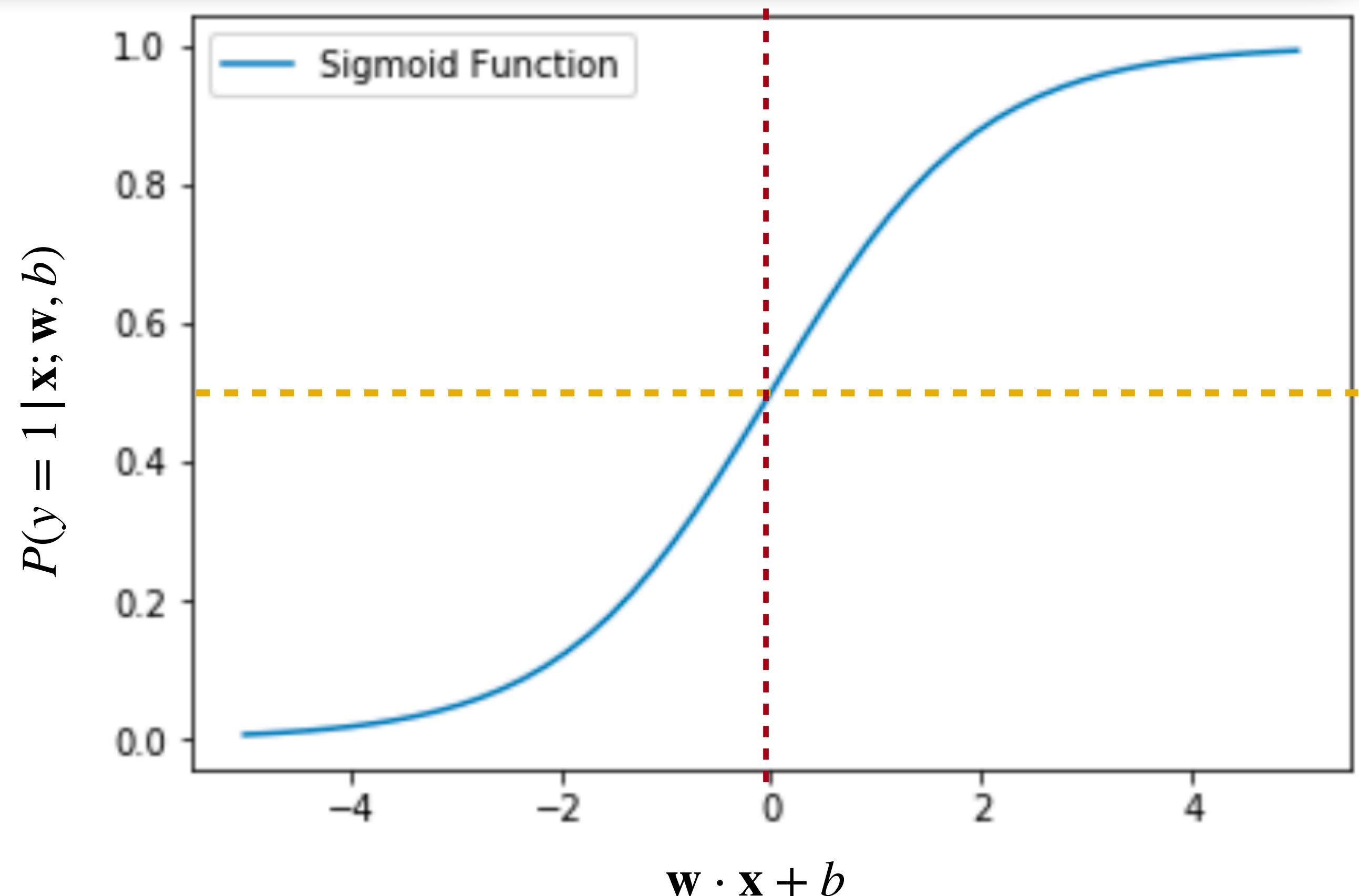
# Classification Decision

Inference under a Logistic Regression Model

$$\hat{y} = \begin{cases} 1 & \text{if } p(y = 1 | x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Decision Boundary

$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \end{cases}$$



Often  $\hat{y}$  is used to indicate probability:

$$\hat{y} = P(y = 1 | \mathbf{x}; \theta) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

# Maximizing conditional likelihood

For a single observation

# Maximizing conditional likelihood

**Goal:** maximize probability of the correct label  $p(y|x)$

For a single observation

# Maximizing conditional likelihood

**Goal:** maximize probability of the correct label  $p(y|x)$

For a single observation

Since there are only 2 discrete ground truth outcomes,  $y$  (0 or 1) we can express the probability  $p(y|\mathbf{x})$  from our classifier (the thing we want to maximize) as

# Maximizing conditional likelihood

**Goal:** maximize probability of the correct label  $p(y | x)$

For a single observation

Since there are only 2 discrete ground truth outcomes,  $y$  (0 or 1) we can express the probability  $p(y | \mathbf{x})$  from our classifier (the thing we want to maximize) as

$$p(y | x) = \hat{y}^y(1 - \hat{y})^{1-y}$$

# Maximizing conditional likelihood

**Goal:** maximize probability of the correct label  $p(y | x)$

For a single observation

Since there are only 2 discrete ground truth outcomes,  $y$  (0 or 1) we can express the probability  $p(y | \mathbf{x})$  from our classifier (the thing we want to maximize) as

$$p(y | x) = \hat{y}^y(1 - \hat{y})^{1-y}$$

Data Likelihood

# Maximizing conditional likelihood

**Goal:** maximize probability of the correct label  $p(y|x)$

For a single observation

Since there are only 2 discrete ground truth outcomes,  $y$  (0 or 1) we can express the probability  $p(y|\mathbf{x})$  from our classifier (the thing we want to maximize) as

$$p(y|x) = \hat{y}^y(1 - \hat{y})^{1-y}$$

Data Likelihood

Estimated probabilities →

		$\hat{y} = 0$	$\hat{y} = .3$	$\hat{y} = .5$	$\hat{y} = .7$	$\hat{y} = 1$
$y = 0$	$y = 0$	1	0.7	0.5	0.3	0
	$y = 1$	0	0.3	0.5	0.7	1

# Minimizing negative log likelihood

**Goal:** maximize probability of the correct label  $p(y | \mathbf{x})$

$$\begin{aligned}\text{Maximize: } \log p(y | x) &= \log(\hat{y}^y(1 - \hat{y})^{1-y}) \\ &= y \log \hat{y} + (1 - y) \log(1 - \hat{y})\end{aligned}$$

Measures how well the training data matches the proposed model distribution and how good the model distribution is

Now flip the sign for something to minimize (we minimize the loss / cost)

$$\begin{aligned}\text{Minimize: } L_{CE}(y, \hat{y}) &= -\log p(y | x) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \\ &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log \sigma(-(\mathbf{w} \cdot \mathbf{x} + b))]\end{aligned}$$

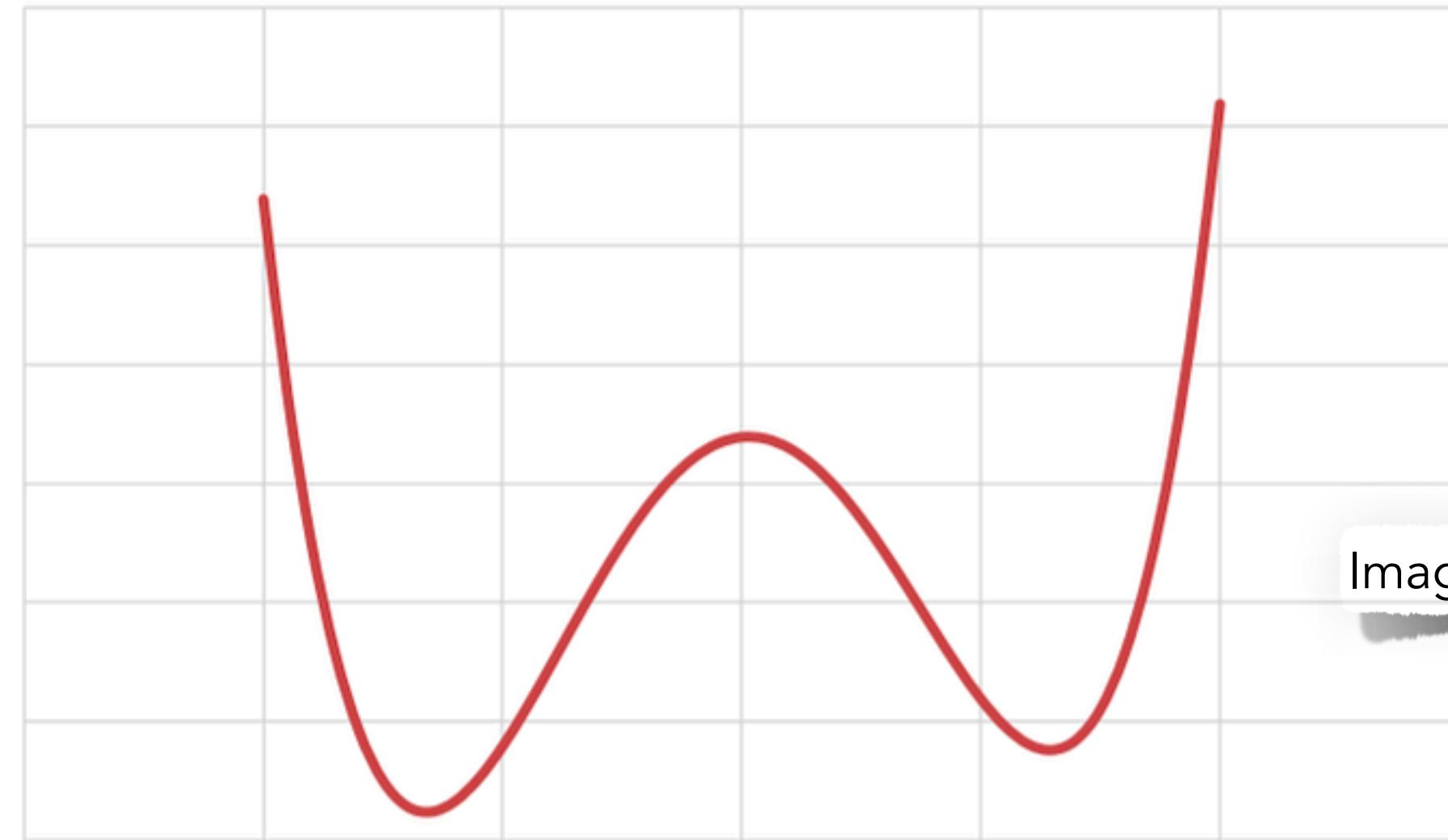
Cross-Entropy  
Loss

# Logistic Regression: Loss



Convex function

- Has only one option for steepest gradient
  - Or one minimum
- Gradient descent starting from any point is guaranteed to find the minimum



Non-convex function

Neural Networks -  
multiple alternatives

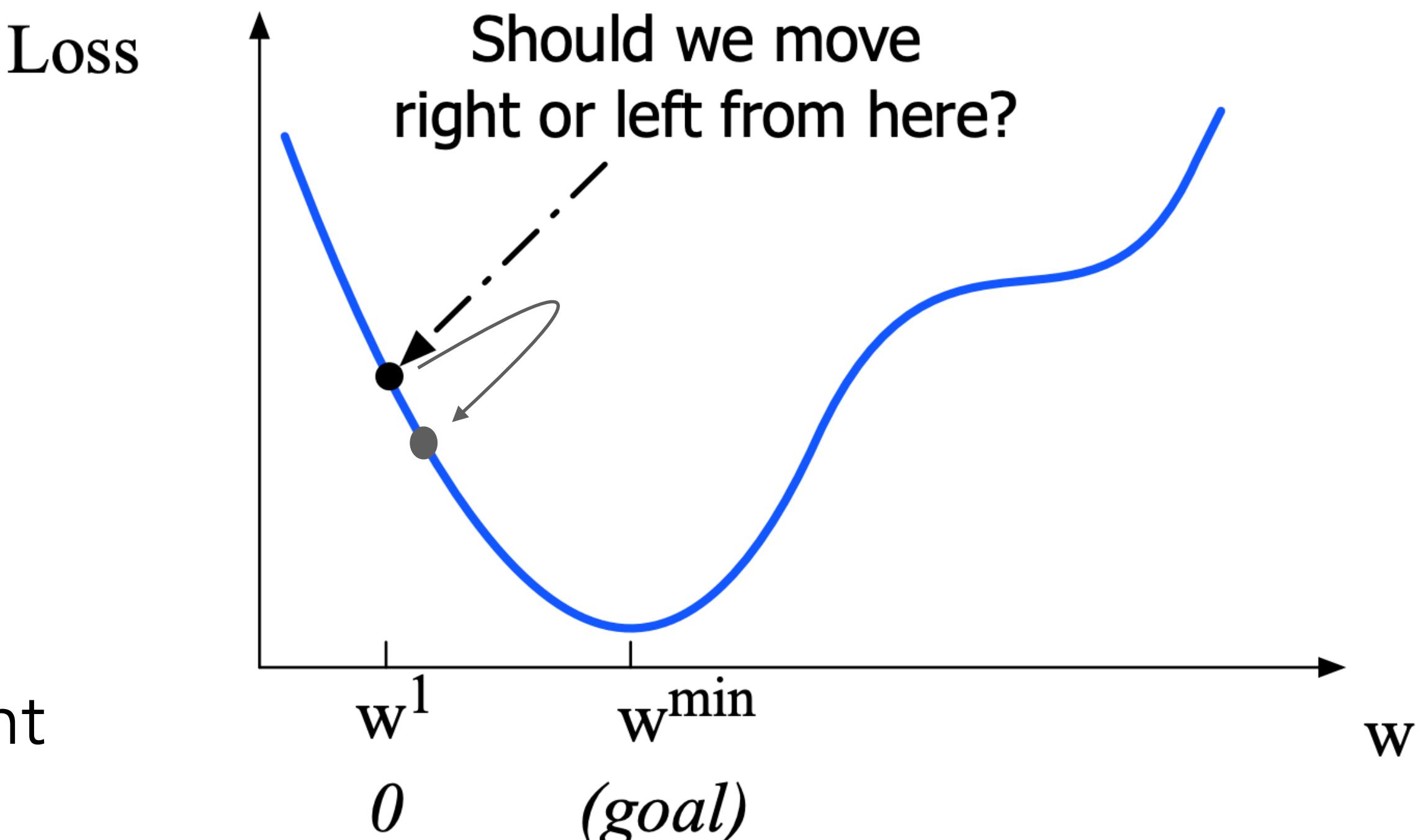
Image Credit: Medium

# Gradients

The **gradient** of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

Find the gradient of the loss function at the current point and move in the **opposite** direction.

Gradient Descent



But by how much?

# Gradient Updates

- Move  $w$  by the value of the gradient  $\frac{\partial}{\partial w} L(f(x; w), y^*)$ , weighted by a learning rate  $\eta$
- Higher learning rate means move  $w$  faster

$\eta$  Too high: the learner will take big steps and overshoot

$$w_{t+1} = w_t - \eta \frac{\partial}{\partial w} L(f(x; w), y^*)$$

$\eta$  Too low: the learner will take too long

If parameter  $\theta$  is a vector of  $d$  dimensions:

The gradient is just such a vector; it expresses the directional components of the sharpest slope along each of the  $d$  dimensions.

# Gradients for Logistic Regression

case: Sentiment Analysis

Recall: the cross-entropy loss for logistic regression

$$L_{CE}(y, \hat{y}) = - [y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log(\sigma(-\mathbf{w} \cdot \mathbf{x} + b))]$$

# Gradients for Logistic Regression

Case: Sentiment Analysis

Recall: the cross-entropy loss for logistic regression

$$L_{CE}(y, \hat{y}) = - [y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log(\sigma(-\mathbf{w} \cdot \mathbf{x} + b))]$$

Derivatives have a closed form solution:

$$\frac{\partial L_{CE}(y, \hat{y})}{\partial w_j} = [\sigma(\mathbf{w} \cdot \mathbf{x} + b) - y]x_j$$

# Pseudocode

```
function STOCHASTIC GRADIENT DESCENT ( $L()$ ,  $f()$ ,  $x$ ,  $y$ ) returns  $\theta$ 
```

```
# where:  $L$  is the loss function
```

```
#  $f$  is a function parameterized by  $\theta$ 
```

```
#  $\mathbf{x}$  is the set of training inputs  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$ 
```

```
#  $y$  is the set of training outputs (labels)  $y^{(1)}, y^{(2)}, \dots, y^{(N)}$ 
```

```
 $\theta \leftarrow 0$  (or randomly initialized)
```

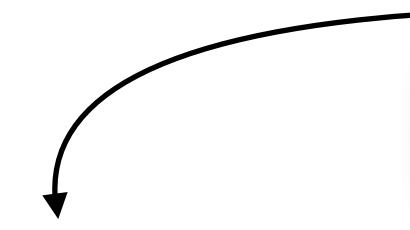
```
repeat till done
```

```
for each training tuple( $x^{(i)}, y^{(i)}$ ): (in random order)
```

1. Compute  $\hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$  # What is our estimated output  $\hat{y}^{(i)}$ ?
2. Compute the loss  $L(\hat{y}^{(i)}, y^{(i)})$  # How far off is  $\hat{y}^{(i)}$  from the true output  $y^{(i)}$  ?
3.  $g \leftarrow \nabla L(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$  # How should we move  $\theta$  to maximize loss?
4.  $\theta \leftarrow \theta - \eta g$  # Go the other way instead

```
return  $\theta$ 
```

Stochastic Gradient Descent



# Mini-Batching

```
function STOCHASTIC GRADIENT DESCENT ( $L()$ ,  $f()$ ,  $x$ ,  $y$ ,  $m$ ) returns  $\theta$ 
    # where:  $L$  is the loss function
    #  $f$  is a function parameterized by  $\theta$ 
    #  $\mathbf{x}$  is the set of training inputs  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$ 
    #  $y$  is the set of training outputs (labels)  $y^{(1)}, y^{(2)}, \dots, y^{(N)}$  and  $m$  is the mini-batch size
     $\theta \leftarrow 0$  (or randomly initialized)
    repeat till done
        for each randomly sampled minibatch of size  $m$ :
            1. for each training tuple  $(\mathbf{x}^{(i)}, y^{(i)})$  in the minibatch: (in random order)
                i. Compute  $\hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$                                 # What is our estimated output  $\hat{y}^{(i)}$ ?
                ii. Compute the loss  $L_{mini} \leftarrow L_{mini} + L(\hat{y}^{(i)}, y^{(i)})$            # How far off is  $\hat{y}^{(i)}$  from the true output  $y^{(i)}$  ?
            2.  $g \leftarrow \frac{1}{m} \nabla L_{mini}(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$                   # How should we move  $\theta$  to maximize loss?
            3.  $\theta \leftarrow \theta - \eta g$                                             # Go the other way instead
        return  $\theta$ 
```

# Mini-Batching

```
function STOCHASTIC GRADIENT DESCENT ( $L()$ ,  $f()$ ,  $x$ ,  $y$ ,  $m$ ) returns  $\theta$ 
    # where:  $L$  is the loss function
    #  $f$  is a function parameterized by  $\theta$ 
    #  $x$  is the set of training inputs  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}$ 
    #  $y$  is the set of training outputs (labels)  $y^{(1)}, y^{(2)}, \dots, y^{(N)}$  and  $m$  is the mini-batch size
     $\theta \leftarrow 0$  (or randomly initialized)
    repeat till done
        for each randomly sampled minibatch of size  $m$ :
            1. for each training tuple  $(\mathbf{x}^{(i)}, y^{(i)})$  in the minibatch: (in random order)
                i. Compute  $\hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$                                 # What is our estimated output  $\hat{y}^{(i)}$ ?
                ii. Compute the loss  $L_{mini} \leftarrow L_{mini} + L(\hat{y}^{(i)}, y^{(i)})$           # How far off is  $\hat{y}^{(i)}$  from the true output  $y^{(i)}$  ?
            2.  $g \leftarrow \frac{1}{m} \nabla L_{mini}(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$                   # How should we move  $\theta$  to maximize loss?
            3.  $\theta \leftarrow \theta - \eta g$                                             # Go the other way instead
    return  $\theta$ 
```

Why is this better than stochastic gradient descent?

# Overfitting

- 4-gram model on tiny data will just memorize the data
  - 100% accuracy on the training set
- But it will be surprised by the novel 4-grams in the test data
  - Low accuracy on test set
- Models that are too powerful can overfit the data
  - Fitting the details of the training data so exactly that the model doesn't generalize well to the test set

How to avoid overfitting?

Regularization in logistic regression

Dropout in neural networks

# Regularization

- A solution for overfitting: Add a regularization term  $R(\theta)$  to the loss function
  - (for now written as maximizing logprob rather than minimizing loss)
- Idea: choose an  $R(\theta)$  that penalizes large weights
  - fitting the data well with lots of big weights not as good as
  - fitting the data a little less well, with small weights

$$\hat{\theta} = \arg \min_{\theta} - \sum_{i=1}^n \log P(y^{(i)} | \mathbf{x}^{(i)}) + \alpha R(\theta)$$

# L2 / Ridge Regularization

- The sum of the squares of the weights
- The name is because this is the (square of the) L2 norm  $\|\theta\|_2^2$ , = Euclidean distance of  $\theta$  to the origin.

$$R(\theta) = \|\theta\|_2^2 = \sum_{j=1}^d \theta_j^2$$

L2 regularized objective function:

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^n \log P(y^{(i)} | x^{(i)}) - \alpha \sum_{j=1}^d \theta_j^2$$

# L1 / Lasso Regularization

- The sum of the (absolute value of the) weights
- Named after the L1 norm  $\|\theta\|_1 = \text{sum of the absolute values of the weights} = \text{Manhattan distance}$

$$R(\theta) = \|\theta\|_1 = \sum_{j=1}^d |\theta_j|$$

L1 regularized objective function:

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^n \log P(y^{(i)} | x^{(i)}) - \alpha \sum_{j=1}^d |\theta_j|$$

# Lecture Outline

- Quiz 1 Solution
- Recap
  - Logistic Regression
    - Model
    - Loss
    - Optimization
    - Regularization
  - Multinomial Logistic Regression
  - Word Embeddings

# Multinomial Logistic Regression



# Multinomial Logistic Regression

# Multinomial Logistic Regression

- Often we need more than 2 classes
  - Positive / negative / neutral sentiment of a document
  - Parts of speech of a word (noun, verb, adjective, adverb, preposition, etc.)
  - Actionable classes for emergency SMSs

# Multinomial Logistic Regression

- Often we need more than 2 classes
  - Positive / negative / neutral sentiment of a document
  - Parts of speech of a word (noun, verb, adjective, adverb, preposition, etc.)
  - Actionable classes for emergency SMSs
- If  $> 2$  classes we use the term “multinomial logistic regression”
  - Typically the term “logistic regression” indicates binary classification

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P(+) | x) + P(- | x) + P(\sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values
  - each  $z_i$  corresponds to weighted sum of features for the  $K$ th class

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values
    - each  $z_i$  corresponds to weighted sum of features for the  $K$ th class
  - Outputs a probability distribution

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values
    - each  $z_i$  corresponds to weighted sum of features for the  $K$ th class
  - Outputs a probability distribution
    - each value in the range  $[0,1]$

# Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P( + | x) + P( - | x) + P( \sim | x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values
    - each  $z_i$  corresponds to weighted sum of features for the  $K$ th class
  - Outputs a probability distribution
    - each value in the range  $[0,1]$
    - all the values summing to 1

Softmax

# The Softmax Function

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

# The Softmax Function

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)}$$

# The Softmax Function

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad 1 \leq i \leq K$$

# The Softmax Function

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad 1 \leq i \leq K$$

The denominator  $\sum_{i=1}^K \exp(z_i)$  is used to normalize all the values into probabilities.

# The Softmax Function

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad 1 \leq i \leq K$$

The denominator  $\sum_{i=1}^K \exp(z_i)$  is used to normalize all the values into probabilities.

$$\text{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right]$$

# Softmax: Example

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\text{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right]$$

# Softmax: Example

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\mathbf{z} = [0.6, 1.1, 1.5, 1.2, 3.2, 1.1]$$

$$\text{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right]$$

# Softmax: Example

Turns a vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  of  $K$  arbitrary values into probabilities

$$\mathbf{z} = [0.6, 1.1, 1.5, 1.2, 3.2, 1.1]$$

$$\text{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right]$$

$$\text{softmax}(\mathbf{z}) = [0.055, 0.090, 0.0067, 0.10, 0.74, 0.010]$$

# Binary versus Multinomial

## Binary Logistic Regression

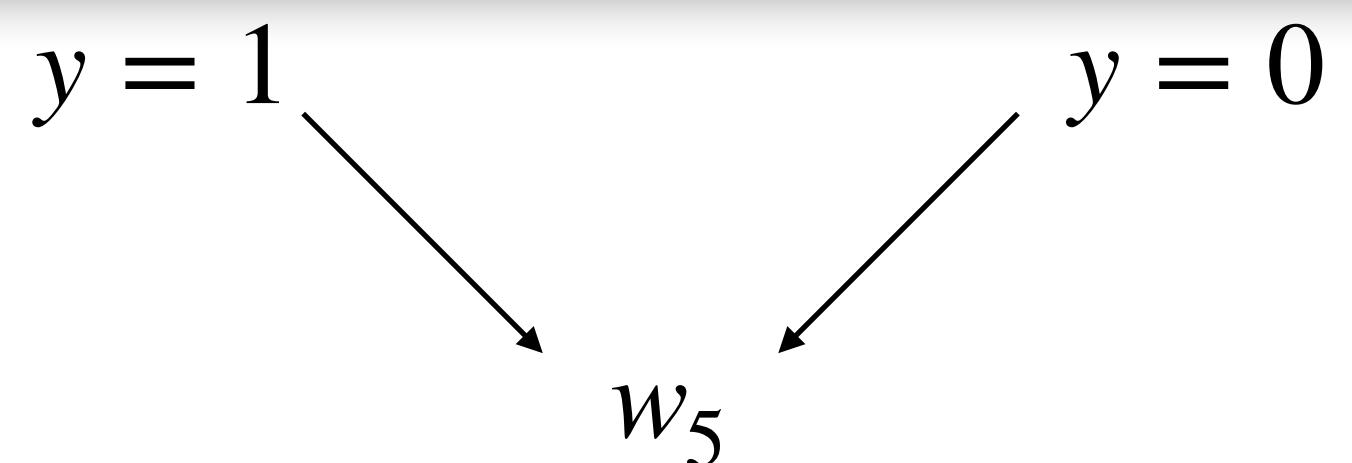
$y = 1$

$y = 0$

Why do we NOT need a different weight for each class in binary logistic regression?

# Binary versus Multinomial

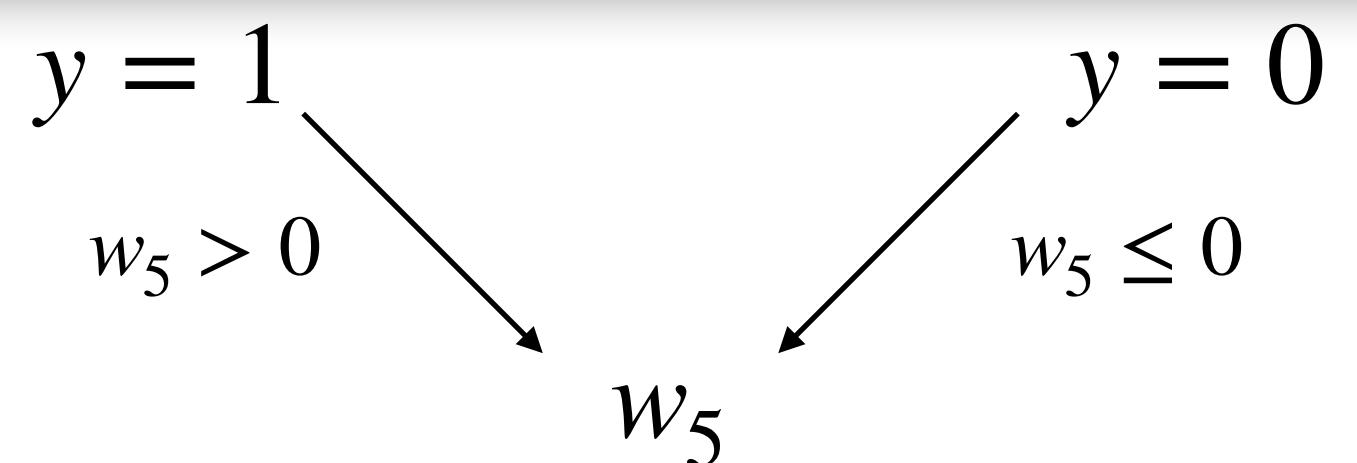
## Binary Logistic Regression



Why do we NOT need a different weight for each class in binary logistic regression?

# Binary versus Multinomial

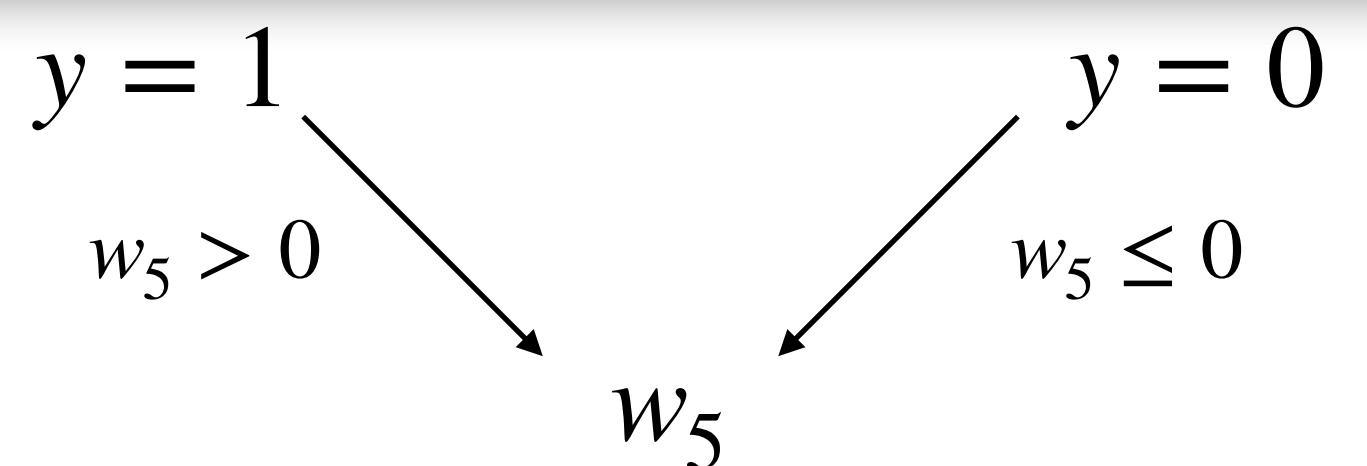
## Binary Logistic Regression



Why do we NOT need a different weight for each class in binary logistic regression?

# Binary versus Multinomial

## Binary Logistic Regression

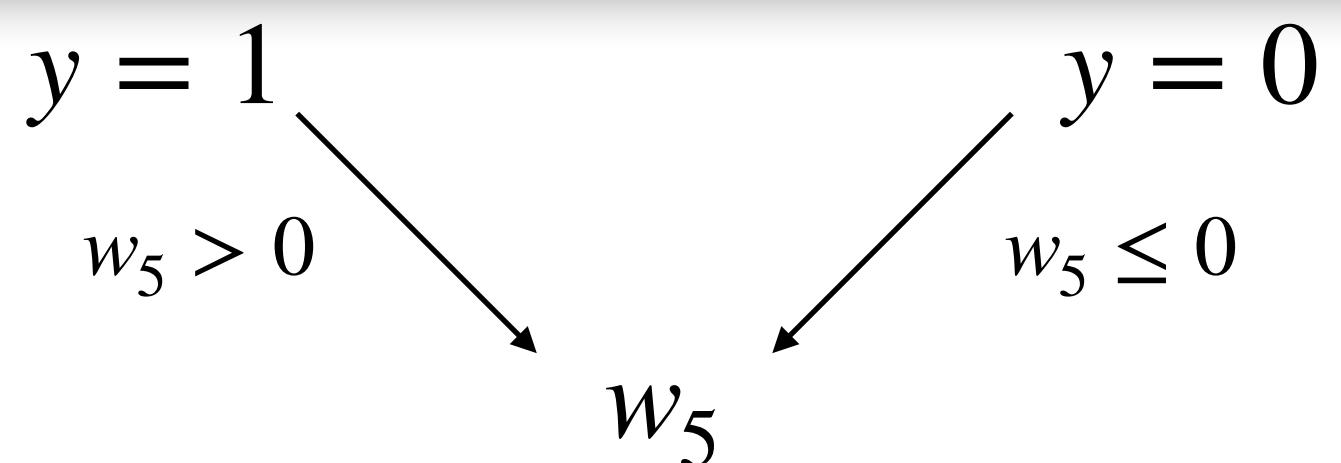


$$x_5 = \begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases} \quad w_5 = 3.0$$

Why do we NOT need a different weight for each class in binary logistic regression?

# Binary versus Multinomial

## Binary Logistic Regression



$$x_5 = \begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases} \quad w_5 = 3.0$$

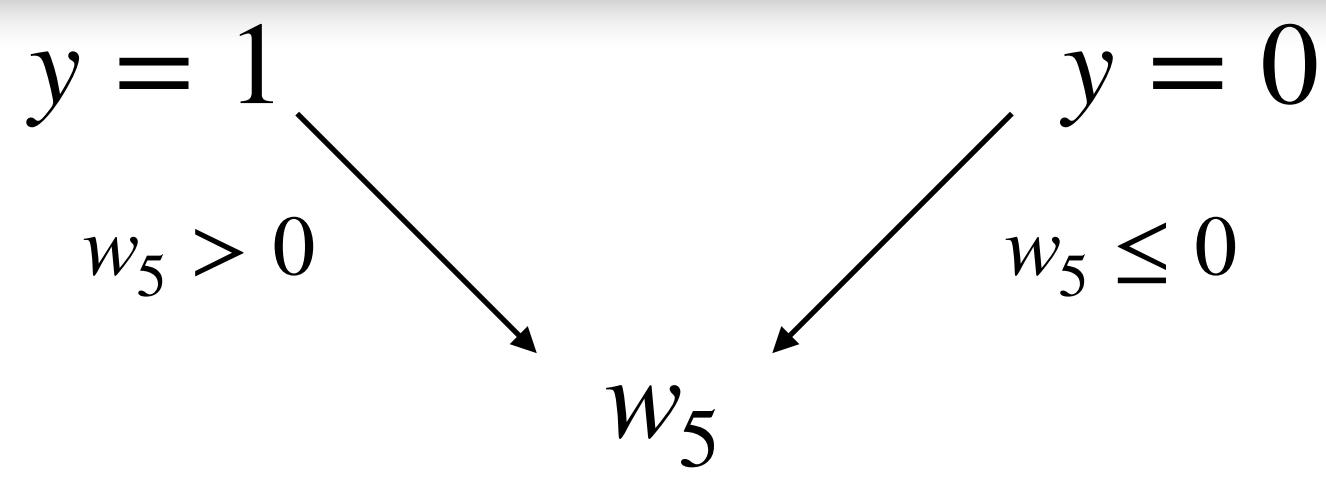
## Multinomial Logistic Regression

$y = +$        $y = \sim$        $y = -$

Why do we NOT need a different weight for each class in binary logistic regression?

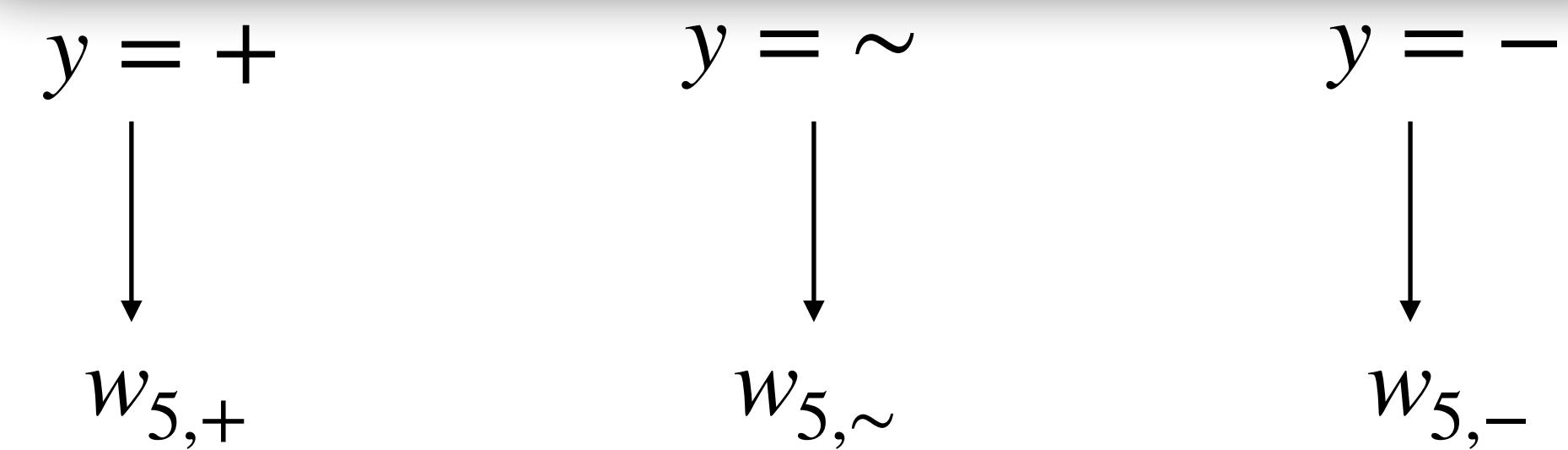
# Binary versus Multinomial

## Binary Logistic Regression



$$x_5 = \begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases} \quad w_5 = 3.0$$

## Multinomial Logistic Regression

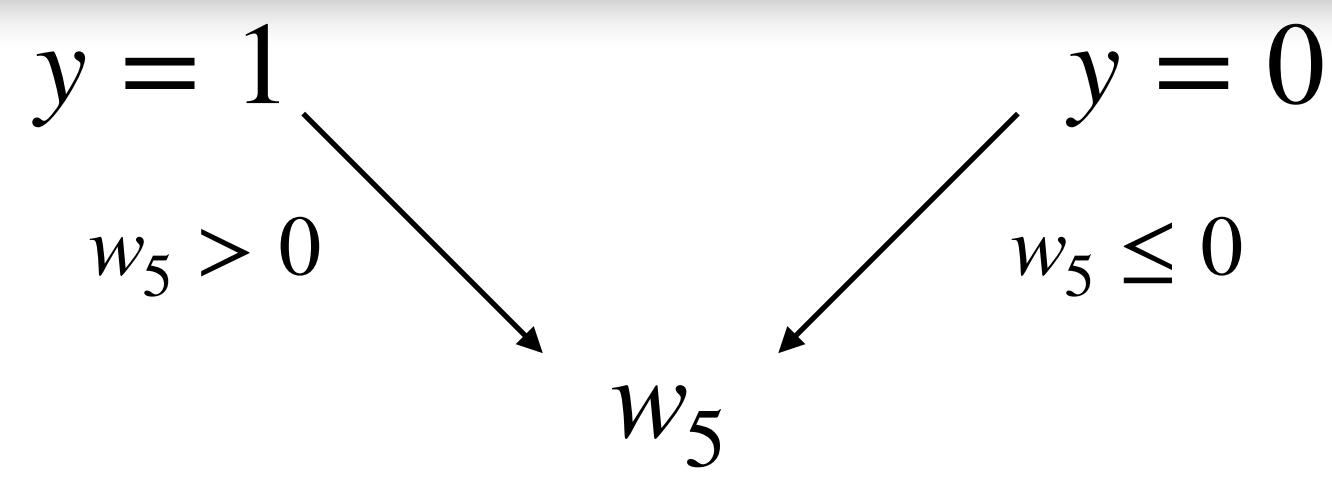


Separate weights for each class

Why do we NOT need a different weight for each class in binary logistic regression?

# Binary versus Multinomial

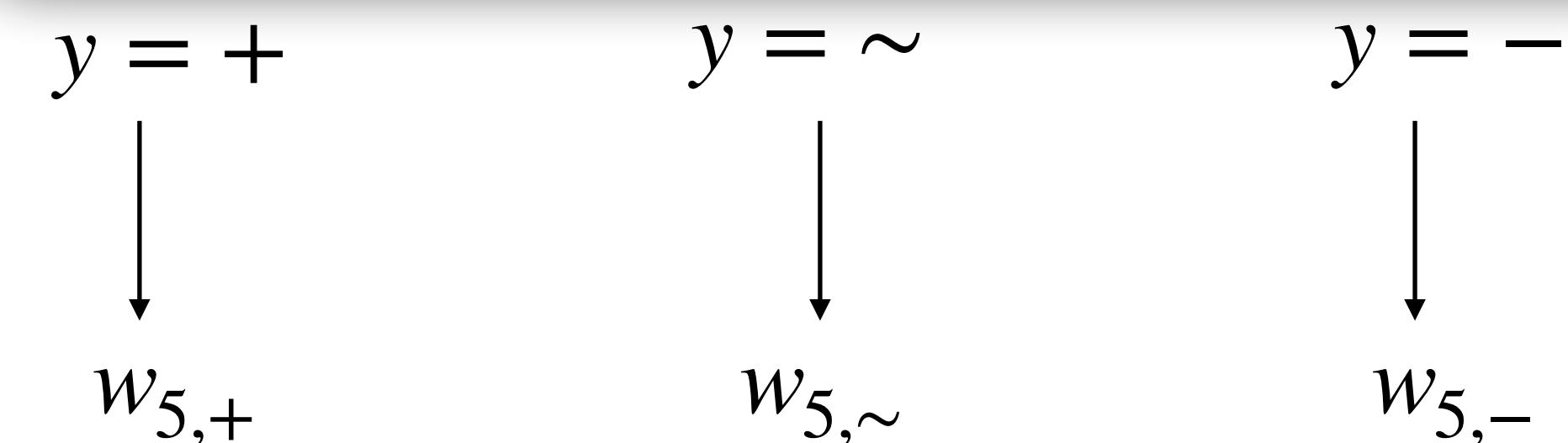
## Binary Logistic Regression



$$x_5 = \begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases} \quad w_5 = 3.0$$

Why do we NOT need a different weight for each class in binary logistic regression?

## Multinomial Logistic Regression



Separate weights for each class

Feature	Definition	$w_{5,+}$	$w_{5,-}$	$w_{5,0}$
$f_5(x)$	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	3.5	3.1	-5.3

# Softmax in multinomial logistic regression

$$P(y = c \mid \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

# Softmax in multinomial logistic regression

$$P(y = c \mid \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$

# Softmax in multinomial logistic regression

$$P(y = c \mid \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$
- But separate weight vectors for each of the  $K$  classes, each of dimension  $d$

# Softmax in multinomial logistic regression

Parameters are now a matrix  $\mathbf{W} \in \mathbb{R}^{d \times K}$  and  $b \in \mathbb{R}^1$

$$P(y = c | \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$
- But **separate weight vectors for each of the  $K$  classes, each of dimension  $d$**

# Softmax in multinomial logistic regression

Parameters are now a matrix  $\mathbf{W} \in \mathbb{R}^{d \times K}$  and  $b \in \mathbb{R}^1$

$$P(y = c | \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$
- But **separate weight vectors for each of the  $K$  classes, each of dimension  $d$**

Multinomial LR Loss:

# Softmax in multinomial logistic regression

Parameters are now a matrix  $\mathbf{W} \in \mathbb{R}^{d \times K}$  and  $b \in \mathbb{R}^1$

$$P(y = c | \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$
- But **separate weight vectors for each of the  $K$  classes, each of dimension  $d$**

Multinomial LR Loss:

$$L_{CE} = -\log P(y = c | \mathbf{x}; \theta) = -(\mathbf{w}_c \cdot \mathbf{x} + b) + \log \left[ \sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b) \right]$$

# Softmax in multinomial logistic regression

Parameters are now a matrix  $\mathbf{W} \in \mathbb{R}^{d \times K}$  and  $b \in \mathbb{R}^1$

$$P(y = c | \mathbf{x}; \theta) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x} + b)}{\sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector  $\mathbf{w}_c$  and input vector  $\mathbf{x}$ , offset by  $b$
- But **separate weight vectors for each of the  $K$  classes, each of dimension  $d$**

Multinomial LR Loss:

$$L_{CE} = -\log P(y = c | \mathbf{x}; \theta) = -(\mathbf{w}_c \cdot \mathbf{x} + b) + \log \left[ \sum_{j=1}^K \exp(\mathbf{w}_j \cdot \mathbf{x} + b) \right]$$

# Case Study: Word Neighborhood Predictor

# Case Study: Word Neighborhood Predictor

- Yet do I fear thy nature. It is too full o'the milk of human kindness To catch the nearest way.
- If it were done when 'tis done, then 'twere well It were done quickly.
- Is this a dagger which I see before me, The handle toward my hand?
- My hands are of ;your color, but I shame To wear a heart so white.
- Knock, knock, knock! Who's there,

7

**THE WALL STREET JOURNAL.**

★★★ \$2.00

TUESDAY, SEPTEMBER 23, 2008 • VOL. CCLII NO. 71

**DOWJONES**  
A NEWS CORPORATION COMPANY

DJIA 11015.69 ▼ 372.75 -3.3% NASDAQ 2178.98 ▼ 4.2% NIKKEI 12090.59 ▲ 1.4% DJ STOXX 50 2745.32 ▼ 2.3% 10-YR TREAS ▼ 16/32, yield 3.82% OIL \$120.92 ▲ \$16.37 GOLD \$903.90 ▲ \$43.30 EURO \$1.4839 YEN 105.19

**What's News—**  
Business & Finance      World-Wide

Volatility spread across financial markets, undermining the dollar and contributing to the biggest-ever one-day jump in oil prices. Investor sentiment turned negative, spurred by the government's proposed \$700 billion bailout plan. Prenzlau last-minute trading in the oil market sent crude surging \$16.37 to \$120.92 a barrel. A1, C2, C14 ■ The Dow industrials fell 372.75 points, or 3.3%, to 11015.69 as the rescue plan worried investors. Financial stocks led the declines. C1 ■ China's benchmark index jumped 2.8% as Beijing took more steps to shore up shares. In Russia, signs of consolidation are emerging. C7, A8 ■ The Fed loosened rules that limited the ability of buyout firms and private investors to take big stakes in banks. A1 ■ The SEC said it will revise newly issued rules to curb short-selling, a move that caught participants off guard and prompted criticism. A9 ■ New York regulators are requiring some sellers of credit-default swaps to be

■ Obama and McCain assessed the bailout plan. Both nominees left their options open about whether they would support or oppose the mammoth proposal if and when it reaches the Senate floor. McCain and Obama agree on several aspects they want changed. Both favor limits on executive pay, increased oversight of the Treasury and more transparency on how the money is spent. A18 ■ Unlike McCain, Obama wants a \$715 billion stimulus package and tax breaks for the middle class. C1 ■ U.S. commanders in Afghanistan say they expect a Taliban offensive this winter, usually a relatively peaceful season. A23 ■ Pakistan's top leaders were to dine at the hotel bombed Saturday but changed the venue, an official said. A23 ■ Gunmen kidnapped Afghanistan's top diplomat to Pakistan after killing his driver. A11 ■ Eleven European tourists and their local guides were abducted in a remote area of Egypt near the Sudanese border. A23

BY TOM LAURICELLA,  
LIZ RAPPAPORT  
AND JOANNA SLATER

Volatility spread across financial markets on Monday—undermining the dollar and contributing to the biggest-ever one-day surge in oil prices—as investors grappled with the depth of the financial crisis and the cost of the government's proposed bailout plan.

In the stock market, investors once again turned negative after two days of gains spurred by the government's plan to remove troubled assets from banks and brokers' books. The Dow Jones Industrial Average dropped 372.75, or 3.3%, Monday to 11015.69, more than wiping out the gain posted Friday on news of the proposed bailout. It marked the first time in the Dow's history that it has moved more than 350 points, four days in a row.

The tumult came as Democratic leaders in Congress and the Bush administration moved closer to agreement on key details of the behemoth package, amid widening hostility from economists and lawmakers toward the plan. The administra-

**Doubts on Rescue Plan Spur Fall in Dollar, Leap for Oil**

**A Day of Volatility**

Stocks retreat sharply as investors flock to safer investments

The dollar weakens due to fears that the U.S. will effectively have to print more money

Funds Get Freer Hand In Buying Bank Stakes

BY PETER LATTMAN AND DAMIAN PALETTA

WASHINGTON—The Federal Reserve, unleashing its latest attempt to inject more cash into the nation's ailing banks, loosened longstanding rules that had limited the ability of buyout firms and private investors to take big stakes in banks.

It marks the latest move by the Fed to rewrite the rulebook in response to the financial crisis. Regulators have grown worried about a shortage of capital at banks, in particular smaller thrifts and regional institutions. The Fed has been crafting this policy for at least two years, and private-equity firms have been aggressively lobbying for more

# Case Study: Word Neighborhood Predictor

- Yet do I fear thy nature. It is too full o'the milk of human kindness To catch the nearest way.
- If it were done when 'tis done, then 'twere well It were done quickly.
- Is this a dagger which I see before me, The handle toward my hand?
- My hands are of ;your color, but I shame To wear a heart so white.
- Knock, knock, knock! Who's there,

Where is it more likely to find the word “doth”?

THE WALL STREET JOURNAL.

TUESDAY, SEPTEMBER 23, 2008 • VOL. CCLII NO. 71      ★★★ \$2.00

**What's News—**

Business & Finance      World-Wide

Volatility spread across financial markets, undermining the dollar and contributing to the biggest-ever one-day jump in oil prices. Investor sentiment turned negative, spurred by the government's proposed \$700 billion bailout plan. Prenzlau last-minute trading in the oil market sent crude surging \$16.37 to \$120.92 a barrel. A1, C2, C14 ■ The Dow industrials fell 372.75 points, or 3.3%, to 11015.69 as the rescue plan worried investors. Financial stocks led the declines. C1 ■ China's benchmark index jumped 2.8% as Beijing took more steps to shore up shares. In Russia, signs of consolidation are emerging. C7, A8 ■ The Fed loosened rules that limited the ability of buyout firms and private investors to take big stakes in banks. A1 ■ The SEC said it will revise newly issued rules to curb short-selling, a move that caught participants off guard and prompted criticism. A9 ■ New York regulators are requiring some sellers of credit-default swaps to be

■ Obama and McCain assessed the bailout plan. Both nominees left their options open about whether they would support or oppose the mammoth proposal if and when it reaches the Senate floor. McCain and Obama agree on several aspects they want changed. Both favor limits on executive pay, increased oversight of the Treasury and more transparency on how the money is spent. A18 ■ Unlike McCain, Obama wants a \$715 billion stimulus package and tax breaks for the middle class. A1 ■ U.S. commanders in Afghanistan say they expect a Taliban offensive this winter, usually a relatively peaceful season. A23 ■ Pakistan's top leaders were to dine at the hotel bombed Saturday but changed the venue, an official said. A23 ■ Gunmen kidnapped Afghanistan's top diplomat to Pakistan after killing his driver. A23 ■ Eleven European tourists and their local guides were abducted in a remote area of Egypt near the Sudanese border. A23

BY TOM LAURICELLA,  
LIZ RAPPAPORT  
AND JOANNA SLATER

Volatility spread across financial markets on Monday—undermining the dollar and contributing to the biggest-ever one-day surge in oil prices—as investors grappled with the depth of the financial crisis and the cost of the government's proposed bailout plan.

In the stock market, investors once again turned negative after two days of gains spurred by the government's plan to remove troubled assets from banks and brokers' books. The Dow Jones Industrial Average dropped 372.75, or 3.3%, Monday to 11015.69, more than wiping out the gain posted Friday on news of the proposed bailout. It marked the first time in the Dow's history that it has moved more than 350 points, four days in a row.

The tumult came as Democratic leaders in Congress and the Bush administration moved closer to agreement on key details of the behemoth package, amid widening hostility from economists and lawmakers toward the plan. The administra-

**Doubts on Rescue Plan Spur Fall in Dollar, Leap for Oil**

**A Day of Volatility**

Stocks retreat sharply as investors flock to safer investments

The dollar weakens due to fears that the U.S. will effectively have to print more money

Funds Get Freer Hand In Buying Bank Stakes

BY PETER LATTMAN AND DAMIAN PALETTA

WASHINGTON—The Federal Reserve, unleashing its latest attempt to inject more cash into the nation's ailing banks, loosened longstanding rules that had limited the ability of buyout firms and private investors to take big stakes in banks.

It marks the latest move by the Fed to rewrite the rulebook in response to the financial crisis. Regulators have grown worried about a shortage of capital at banks, in particular smaller thrifts and regional institutions. The Fed has been crafting this policy for at least two years, and private-equity firms have been aggressively lobbying for more

# Case Study: Word Neighborhood Predictor

What words are likely to co-occur with the word “garnish”?

# Case Study: Word Neighborhood Predictor

What words are likely to co-occur with the word "garnish"?



## Ingredients

12 Large Eggs, Hard-boiled And Peeled  
(See Note)

1/2 c. Mayonnaise

2 tbsp. Prepared Yellow Mustard

1/4 tsp. Kosher Salt

[See Nutritional Information ▾](#)

## Directions

Slice eggs in half lengthwise. Pop out yolks and place them in a medium-sized mixing bowl. Mash yolks with a fork and then add mayonnaise, mustard, and salt. Mix until smooth. To achieve maximum smoothness, blitz the yolk mixture a few times with an immersion blender. If you are planning to use a pastry bag and piping tip to add the yolk mixture to the egg whites, it helps to get the yolk mixture nice and smooth.

Divide the yolk mixture evenly between the egg white halves. Garnish as desired. Serve immediately or refrigerate until ready to serve.

# Case Study: Word Neighborhood Predictor

What words are likely to co-occur with the word “garnish”?



## Ingredients

12 Large Eggs, Hard-boiled And Peeled  
(See Note)

1/2 c. Mayonnaise

2 tbsp. Prepared Yellow Mustard

1/4 tsp. Kosher Salt

[See Nutritional Information ▾](#)

## Directions

Slice eggs in half lengthwise. Pop out yolks and place them in a medium-sized mixing bowl. Mash yolks with a fork and then add mayonnaise, mustard, and salt. Mix until smooth. To achieve maximum smoothness, blitz the yolk mixture a few times with an immersion blender. If you are planning to use a pastry bag and piping tip to add the yolk mixture to the egg whites, it helps to get the yolk mixture nice and smooth.

Divide the yolk mixture evenly between the egg white halves. Garnish as desired. Serve immediately or refrigerate until ready to serve.

Next Class: This is the text classification task we will use logistic regression for to learn word embeddings!

# Text Classification with a Language Model

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!
- However questions and answers (remember one of  $K$  classes) need to be part of the same sequence

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!
- However questions and answers (remember one of  $K$  classes) need to be part of the same sequence
- At inference time, we only care about the probability of the answers, given the history

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!
- However questions and answers (remember one of  $K$  classes) need to be part of the same sequence
- At inference time, we only care about the probability of the answers, given the history
  - Probability of all other tokens does not matter

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!
- However questions and answers (remember one of  $K$  classes) need to be part of the same sequence
- At inference time, we only care about the probability of the answers, given the history
  - Probability of all other tokens does not matter
  - Renormalize probabilities of the  $K$  possible answers so they add to 1

# Text Classification with a Language Model

- Can use an  $n$ -gram language model for text classification!
- However questions and answers (remember one of  $K$  classes) need to be part of the same sequence
- At inference time, we only care about the probability of the answers, given the history
  - Probability of all other tokens does not matter
  - Renormalize probabilities of the  $K$  possible answers so they add to 1
    - Use the softmax function!

# Lecture Outline

- Quiz 1 Solution
- Recap
  - Logistic Regression
    - Model
    - Loss
    - Optimization
    - Regularization
  - Multinomial Logistic Regression
  - Word Embeddings

**Words, words, words**

~~Words, words, words~~

Types, types, types

# What do words mean?

# What do words mean?

## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)



### objective

/əb'jektiv/

*adjective*

1. (of a person or their judgment) not influenced by personal feelings or opinions in considering and representing facts.

"historians try to be objective and impartial"

Similar:

impartial

unbiased

unprejudiced

nonpartisan

disinterested



2. **GRAMMAR**

relating to or denoting a case of nouns and pronouns used as the object of a transitive verb or a preposition.

*noun*

1. a thing aimed at or sought; a goal.

"the system has achieved its objective"

Similar:

aim

intention

purpose

target

goal

intent

object

end



2. **GRAMMAR**

the objective case.

# What do words mean?

A **sense** or “concept” is the meaning component of a word

## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)



ob·jec·tive

/əb'jektiv/

*adjective*

1. (of a person or their judgment) not influenced by personal feelings or opinions in considering and representing facts.

"historians try to be objective and impartial"

Similar:

impartial

unbiased

unprejudiced

nonpartisan

disinterested



2. **GRAMMAR**

relating to or denoting a case of nouns and pronouns used as the object of a transitive verb or a preposition.

*noun*

1. a thing aimed at or sought; a goal.

"the system has achieved its objective"

Similar:

aim

intention

purpose

target

goal

intent

object

end



2. **GRAMMAR**

the objective case.

Sense

# What do words mean?

A **sense** or “concept” is the meaning component of a word

## Lemmas

- Canonical form
- For example,  
break, breaks, broke, broken and  
breaking all share the lemma  
“break”

## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)



ob·jec·tive

/əb'jektiv/

Lemma

adjective

1. (of a person or their judgment) not influenced by personal feelings or opinions in considering and representing facts.  
"historians try to be objective and impartial"

Similar:

impartial

unbiased

unprejudiced

nonpartisan

disinterested



2. GRAMMAR

relating to or denoting a case of nouns and pronouns used as the object of a transitive verb or a preposition.

noun

1. a thing aimed at or sought; a goal.

"the system has achieved its objective"

Similar:

aim

intention

purpose

target

goal

intent

object

end



2. GRAMMAR

the objective case.

Sense

# What do words mean?

A **sense** or “concept” is the meaning component of a word

## Lemmas

- Canonical form
- For example,  
break, breaks, broke, broken and  
breaking all share the lemma  
“break”
- Different from “stem”

Can be polysemous (have multiple senses)

## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)



ob·jec·tive

/əb'jektiv/

Lemma

adjective

1. (of a person or their judgment) not influenced by personal feelings or opinions in considering and representing facts.  
"historians try to be objective and impartial"

Similar:

impartial

unbiased

unprejudiced

nonpartisan

disinterested



2. GRAMMAR

relating to or denoting a case of nouns and pronouns used as the object of a transitive verb or a preposition.

noun

1. a thing aimed at or sought; a goal.

"the system has achieved its objective"

Similar:

aim

intention

purpose

target

goal

intent

object

end



2. GRAMMAR

the objective case.

Sense

# Words as Vectors

$\text{man} \rightarrow$	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
$\text{woman} \rightarrow$	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
$\text{king} \rightarrow$	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
$\text{queen} \rightarrow$	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9



# Words as Vectors

In NLP, we commonly represent word types with vectors!

*man* → 

0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
-----	------	-----	-----	------	------	------

*woman* → 

0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
-----	-----	-----	------	-----	------	------

*king* → 

0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
-----	------	-----	-----	-----	------	------

*queen* → 

0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9
-----	------	-----	------	-----	------	------



# Words as Vectors

Why?

In NLP, we commonly represent word types with vectors!

*man* → 

0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
-----	------	-----	-----	------	------	------

*woman* → 

0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
-----	-----	-----	------	-----	------	------

*king* → 

0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
-----	------	-----	-----	-----	------	------

*queen* → 

0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9
-----	------	-----	------	-----	------	------

 Word       Word embedding

# Words as Vectors

Why?

In NLP, we commonly represent word types with vectors!

- Very useful in capturing similarity between words, and other forms of lexical semantics (e.g. synonymy, hypernyms, antonymy)

<i>man</i> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Word                          Word embedding

# Words as Vectors

Why?

In NLP, we commonly represent word types with vectors!

- Very useful in capturing similarity between words, and other forms of lexical semantics (e.g. synonymy, hypernyms, antonymy)
- Computing the similarity between two words (or phrases, or documents) is extremely useful for many NLP tasks

<i>man</i> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Word    Word embedding

# Words as Vectors

Why?

In NLP, we commonly represent word types with vectors!

- Very useful in capturing similarity between words, and other forms of lexical semantics (e.g. synonymy, hypernyms, antonymy)
- Computing the similarity between two words (or phrases, or documents) is extremely useful for many NLP tasks
  - Q: How **tall** is Mount Everest?

$\text{man} \rightarrow$	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
$\text{woman} \rightarrow$	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
$\text{king} \rightarrow$	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
$\text{queen} \rightarrow$	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Word                                  Word embedding

# Words as Vectors

Why?

In NLP, we commonly represent word types with vectors!

- Very useful in capturing similarity between words, and other forms of lexical semantics (e.g. synonymy, hypernyms, antonymy)
- Computing the similarity between two words (or phrases, or documents) is extremely useful for many NLP tasks
  - Q: How **tall** is Mount Everest?
  - A: The official **height** of Mount Everest is 29029 ft

<i>man</i> →	0.6	-0.2	0.8	0.9	-0.1	-0.9	-0.7
<i>woman</i> →	0.7	0.3	0.9	-0.7	0.1	-0.5	-0.4
<i>king</i> →	0.5	-0.4	0.7	0.8	0.9	-0.7	-0.6
<i>queen</i> →	0.8	-0.1	0.8	-0.9	0.8	-0.5	-0.9

Word                                  Word embedding

- Similarity for plagiarism detection
- Word similarity can lead to sentence and document similarity

enough scale for companies to make profit from it. In order to be competitive with new technologies, the challenge of today's large companies is to create new business within their business (Garvin & Levesque, 2006). Furthermore, the two researchers emphasize a switch from downsizing and cost cutting to the creation, development and assistance of innovative new businesses. For existing companies the implementation of corporate entrepreneurship, in order to develop innovative businesses, is risky. Are the three types of entrepreneurship linked together over time? How long does it take to change behavior of the firm as a whole? If the five attributes are created, do all grow together equally, or do some grow faster and earlier than others? How do the importance and intensities of the attributes differ both absolutely and relatively in each type? These are the questions that a longitudinal study such as this can attempt to answer to shed light on the nature of organizations' adjustments to hostile environments. According to Garvin and Levesque (2006) implementing new ventures face several barriers, and can only be successful if a blend of old and new organizational traits is done. To achieve a blend of old and new, an organization needs to rely on employee innovative behavior in order to succeed in dynamic business environments (Yuan & Woodman, 2010).

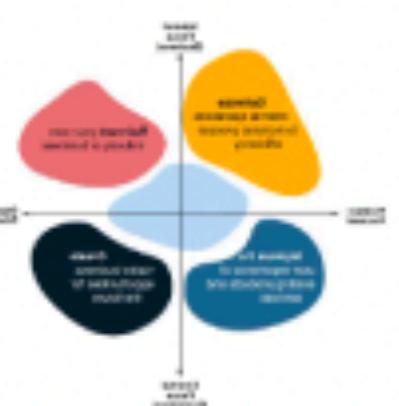


Figure 1 - Corporate entrepreneurship model - innovation-leveraging.com

The downside is potentially very damaging to a startup's lifespan: if a startup lands a pilot or POC with the corporation running the accelerator, they have very little bargaining power or time to find other partners to test their solution with. The transition from manufacturing economy to service economy has led to a shift in business agenda from

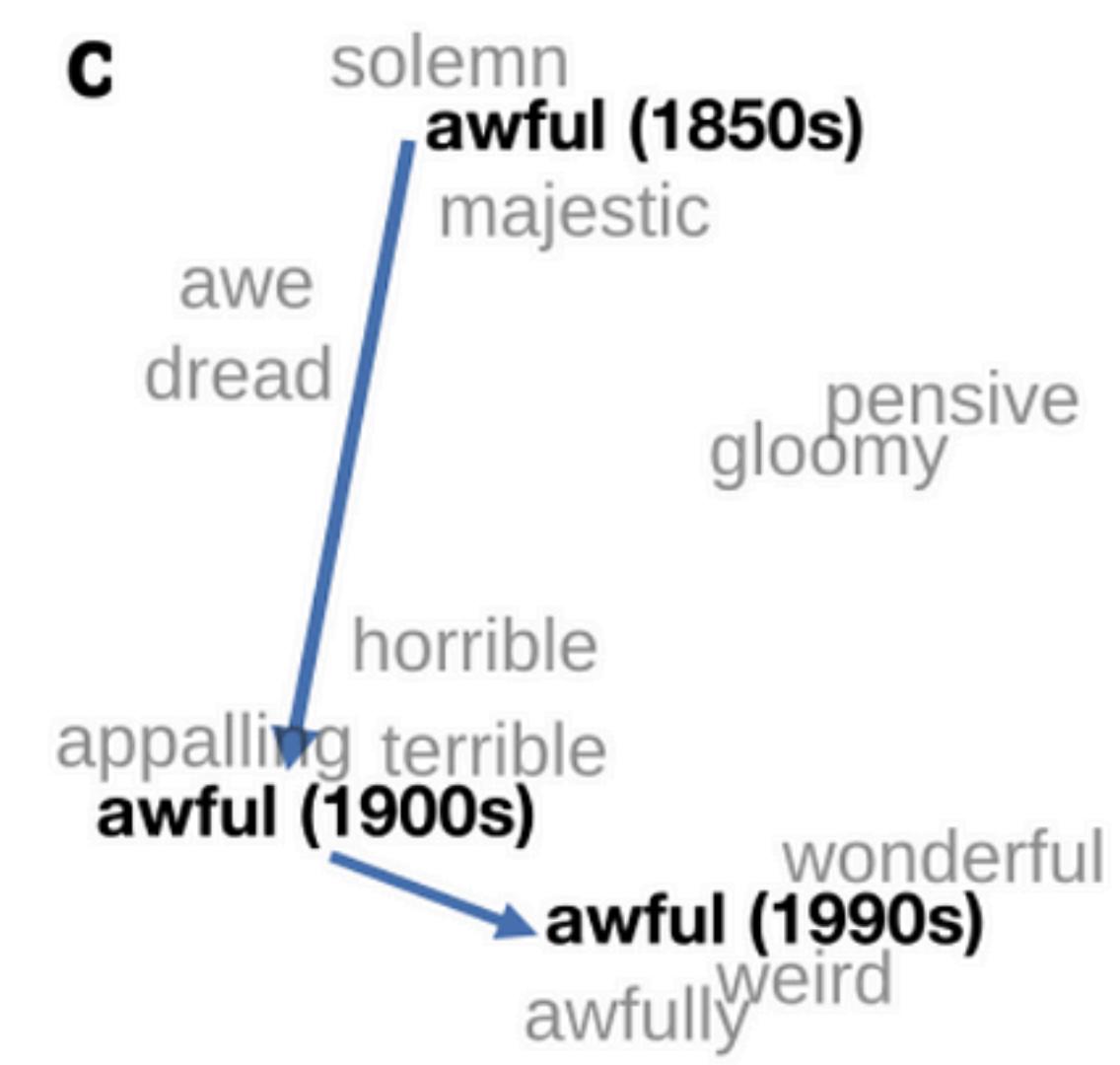
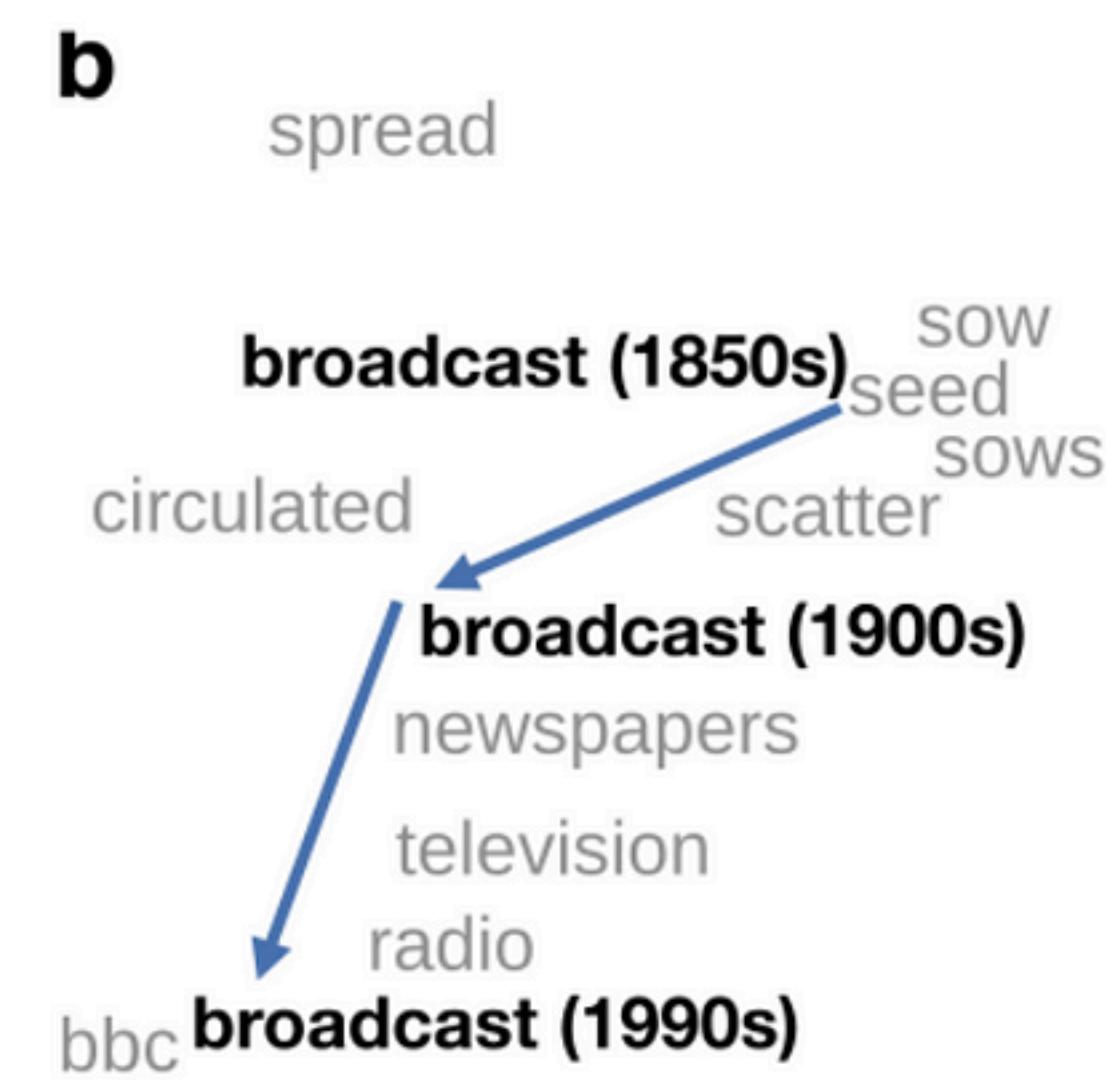
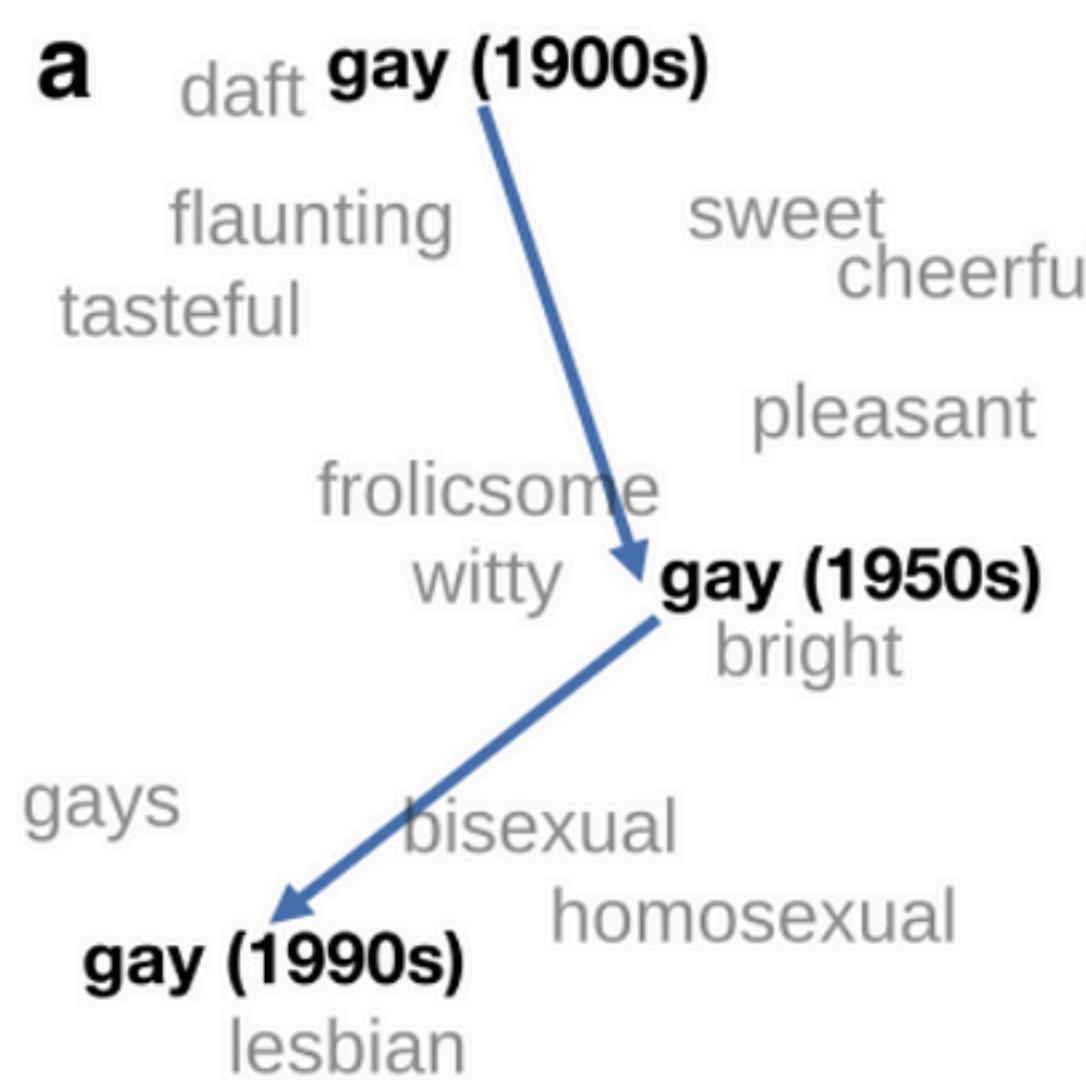
1

Original source  
[onlinelibrary.wiley.com/stor...](http://onlinelibrary.wiley.com/stor...)

...all grow together equally, or do some grow faster and earlier than others? These are the questions that a longitudinal study such as this can attempt to answer to shed light on the nature of organizations' adjustments to hostile environments. Of the many ways to adjust, two stand out at

- Visualizing semantic change over time

- New words: dank, cheugy, rizz, shook, situationship, simp, delulu, cottagecore



~30 million books, 1850-1990, Google Books data

# Cosine Similarity for Word Similarity

Cosine similarity of two vectors

$$\cos(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|}$$

# Cosine Similarity for Word Similarity

Cosine similarity of two vectors

$$\begin{aligned}\cos(\vec{v}, \vec{w}) &= \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} \\ &= \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}\end{aligned}$$

# Cosine Similarity for Word Similarity

Cosine similarity of two vectors

$$\begin{aligned}\cos(\vec{v}, \vec{w}) &= \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} \\ &= \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}\end{aligned}$$

Based on the definition of the dot product between two vectors  $\vec{a}$  and  $\vec{b}$

$$\vec{v} \cdot \vec{w} = |\vec{v}| |\vec{w}| \cos \theta$$

# Cosine Similarity for Word Similarity

Cosine similarity of two vectors

$$\begin{aligned}\cos(\vec{v}, \vec{w}) &= \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} \\ &= \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}\end{aligned}$$

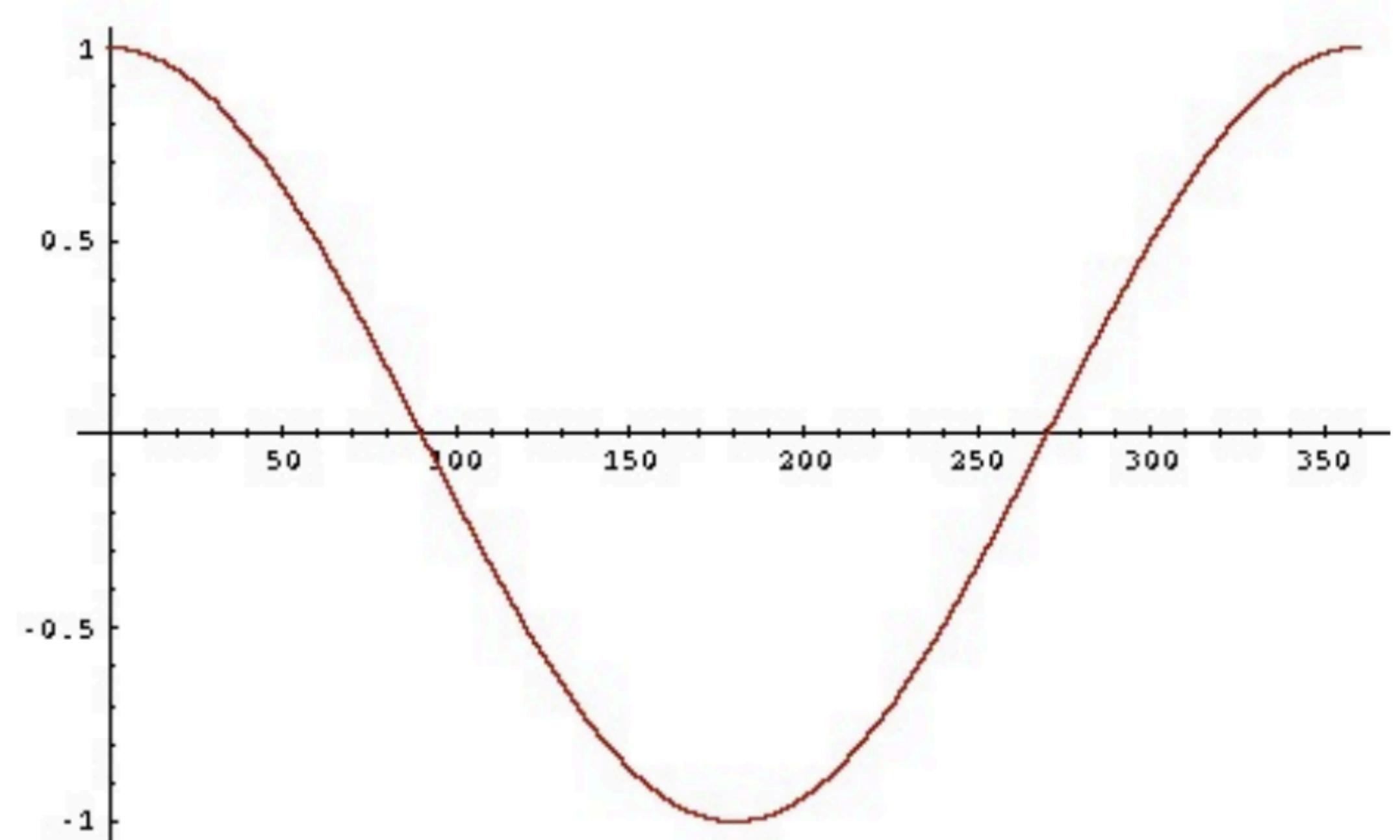
Based on the definition of the dot product between two vectors  $\vec{a}$  and  $\vec{b}$

$$\vec{v} \cdot \vec{w} = |\vec{v}| |\vec{w}| \cos \theta$$

$$\cos \theta = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|}$$

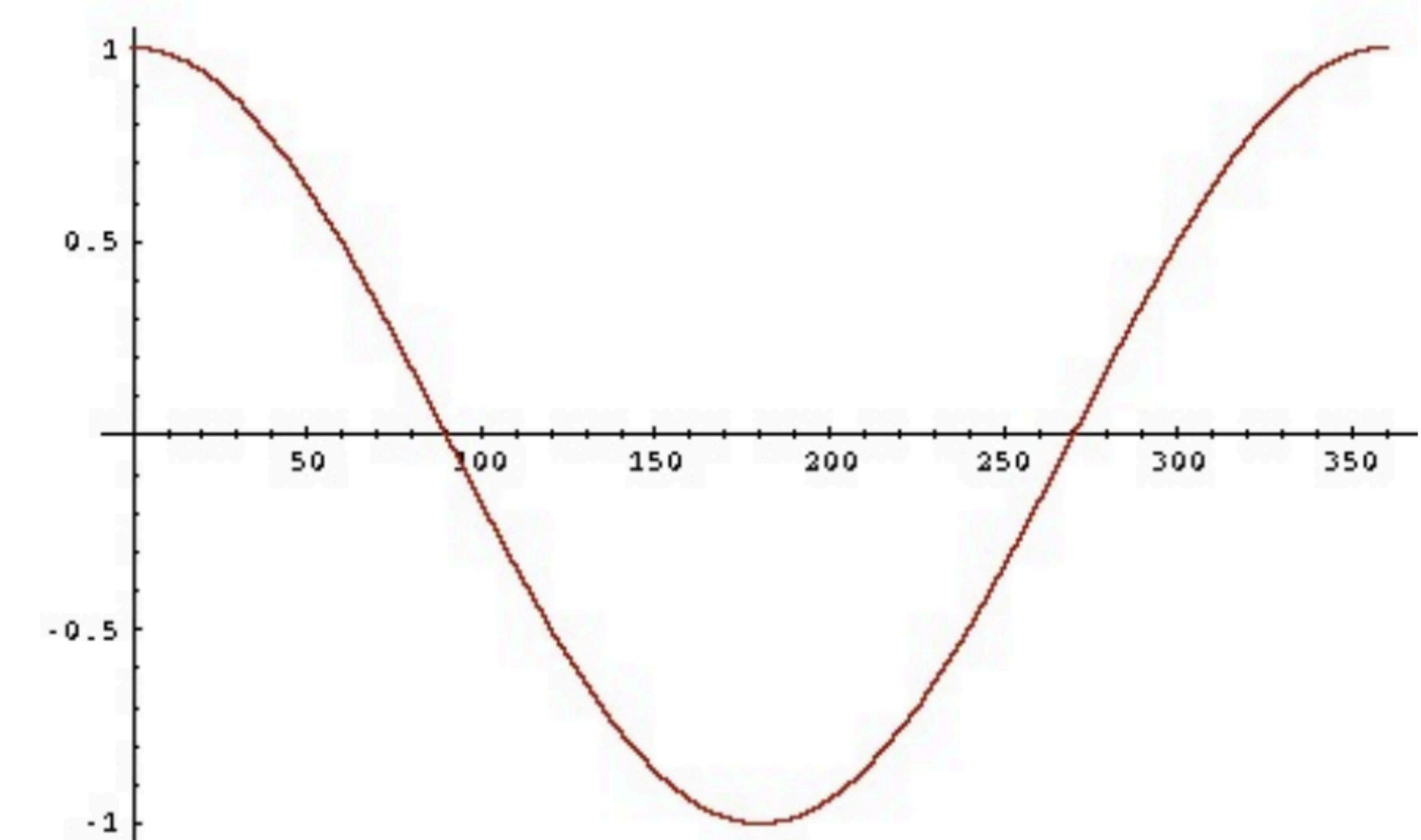
# Cosine as a similarity metric

- 1: vectors point in opposite directions
- +1: vectors point in same directions
- 0: vectors are orthogonal



# Cosine as a similarity metric

- 1: vectors point in opposite directions
- +1: vectors point in same directions
- 0: vectors are orthogonal



Greater the cosine, more similar the words

# *n*-grams and Semantics

# *n*-grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters

# *n*-grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters
- Just strings - atomic symbols!

# $n$ -grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters
- Just strings - atomic symbols!
  - As  $n$  increases, we get strings that co-occur

# *n*-grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters
- Just strings - atomic symbols!
  - As  $n$  increases, we get strings that co-occur
- $n$ -grams do not represent meaning well
  - Do not tell us that the word “rancor” is close in meaning to the word “hatred”
  - Or that “Rise” and “Fall” have opposite meanings
  - Let alone more complex is-a or part-of relations

# *n*-grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters
- Just strings - atomic symbols!
  - As  $n$  increases, we get strings that co-occur
- $n$ -grams do not represent meaning well
  - Do not tell us that the word “rancor” is close in meaning to the word “hatred”
  - Or that “Rise” and “Fall” have opposite meanings
  - Let alone more complex is-a or part-of relations
- **Discrete** representations of meaning!

# *n*-grams and Semantics

- Were feature representations! And so were Bag-of-Words
  - $x$ 's in machine learning, which are associated with parameters
- Just strings - atomic symbols!
  - As  $n$  increases, we get strings that co-occur
- $n$ -grams do not represent meaning well
  - Do not tell us that the word "rancor" is close in meaning to the word "hatred"
  - Or that "Rise" and "Fall" have opposite meanings
  - Let alone more complex is-a or part-of relations
- **Discrete** representations of meaning!
- Later: feature representations which are continuous

# *n*-grams as One-hot Vectors

Unigram Vectors: Represent each word as a vector of zeros with a single 1 identifying the index of the word

# *n*-grams as One-hot Vectors

## vocabulary

i

hate

love

the

movie

film

Unigram Vectors: Represent each word as a vector of zeros with a single 1 identifying the index of the word

# *n*-grams as One-hot Vectors

## vocabulary

i

movie = < 0,0,0,0,1,0 >

One hot vector

hate

film = < 0,0,0,0,0,1 >

love

the

movie

film

Unigram Vectors: Represent each word as a vector of zeros with a single 1 identifying the index of the word

# *n*-grams as One-hot Vectors

## vocabulary

i

movie = &lt; 0,0,0,0,1,0 &gt;

One hot vector

hate

film = &lt; 0,0,0,0,0,1 &gt;

love

the

movie

film

Unigram Vectors: Represent each word as a vector of zeros with a single 1 identifying the index of the word

Dot product is zero! These vectors are orthogonal

# *n*-grams as One-hot Vectors

## vocabulary

i

movie = &lt; 0,0,0,0,1,0 &gt;

One hot vector

hate

film = &lt; 0,0,0,0,0,1 &gt;

love

the

movie

film

Unigram Vectors: Represent each word as a vector of zeros with a single 1 identifying the index of the word

How can we compute a vector representation such that the dot product correlates with word similarity?

Dot product is zero! These vectors are orthogonal



# Visualizing Embeddings

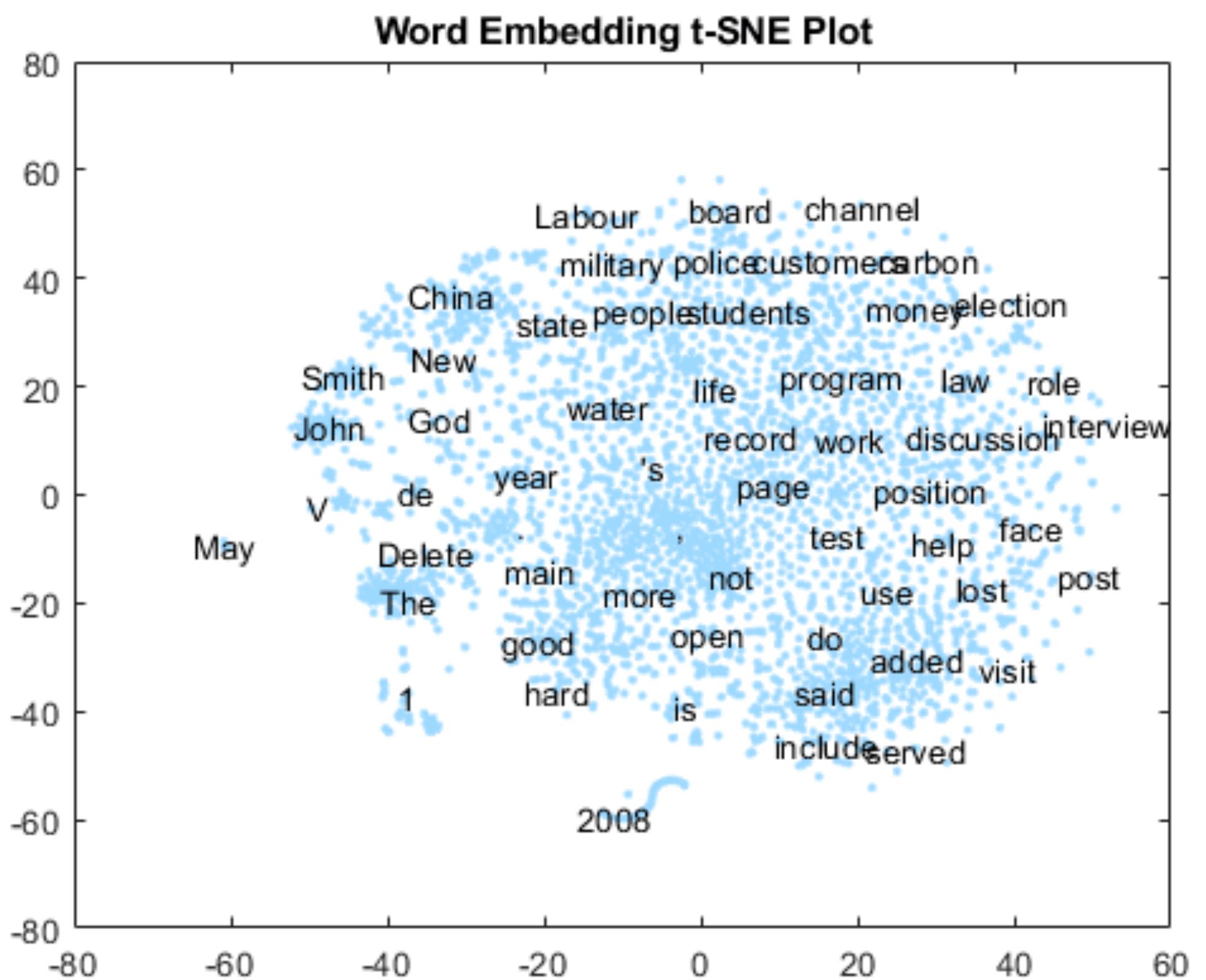
# Visualizing Embeddings

Project high-dimensional embeddings down into 2 dimensions

# Visualizing Embeddings

Project high-dimensional embeddings down into 2 dimensions

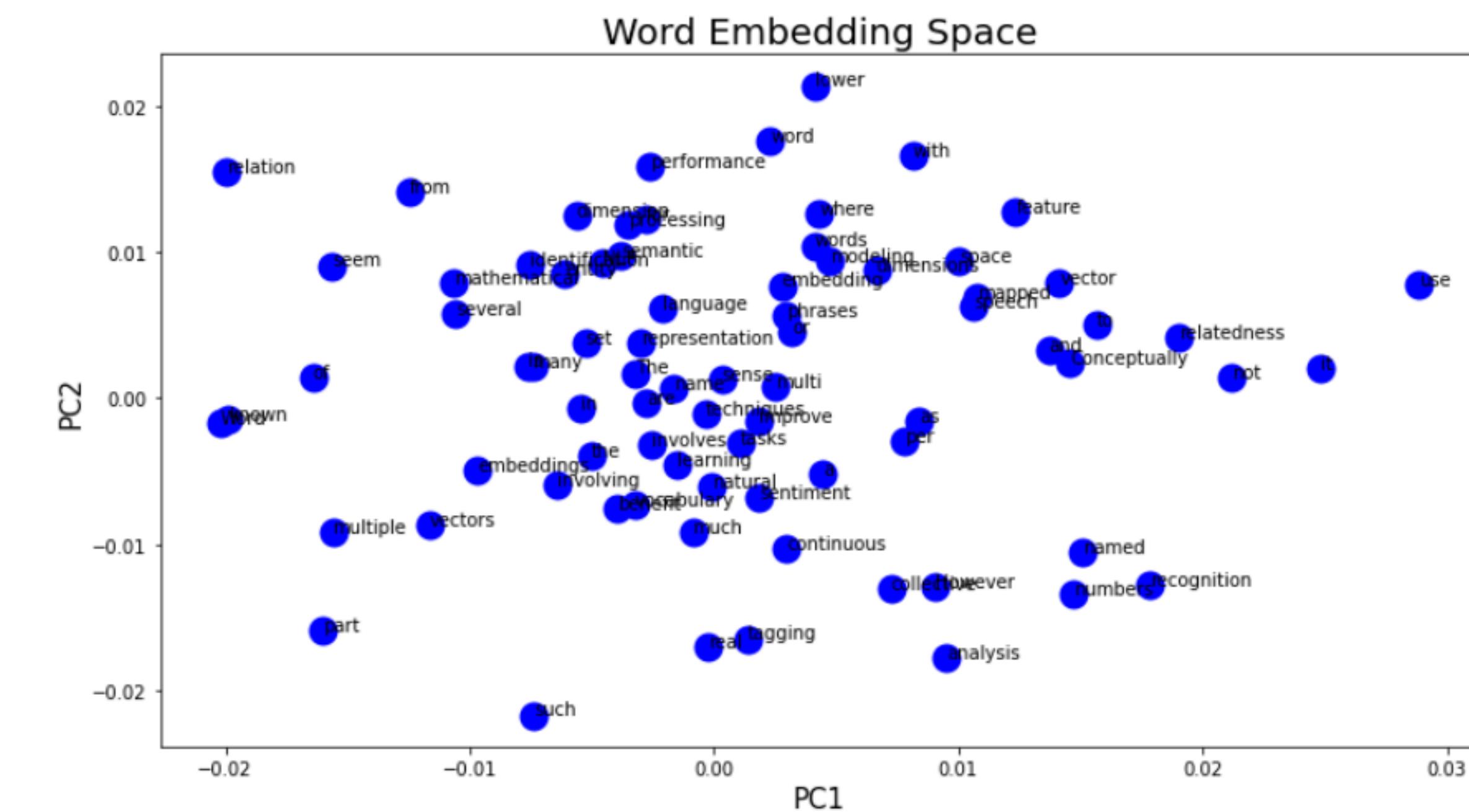
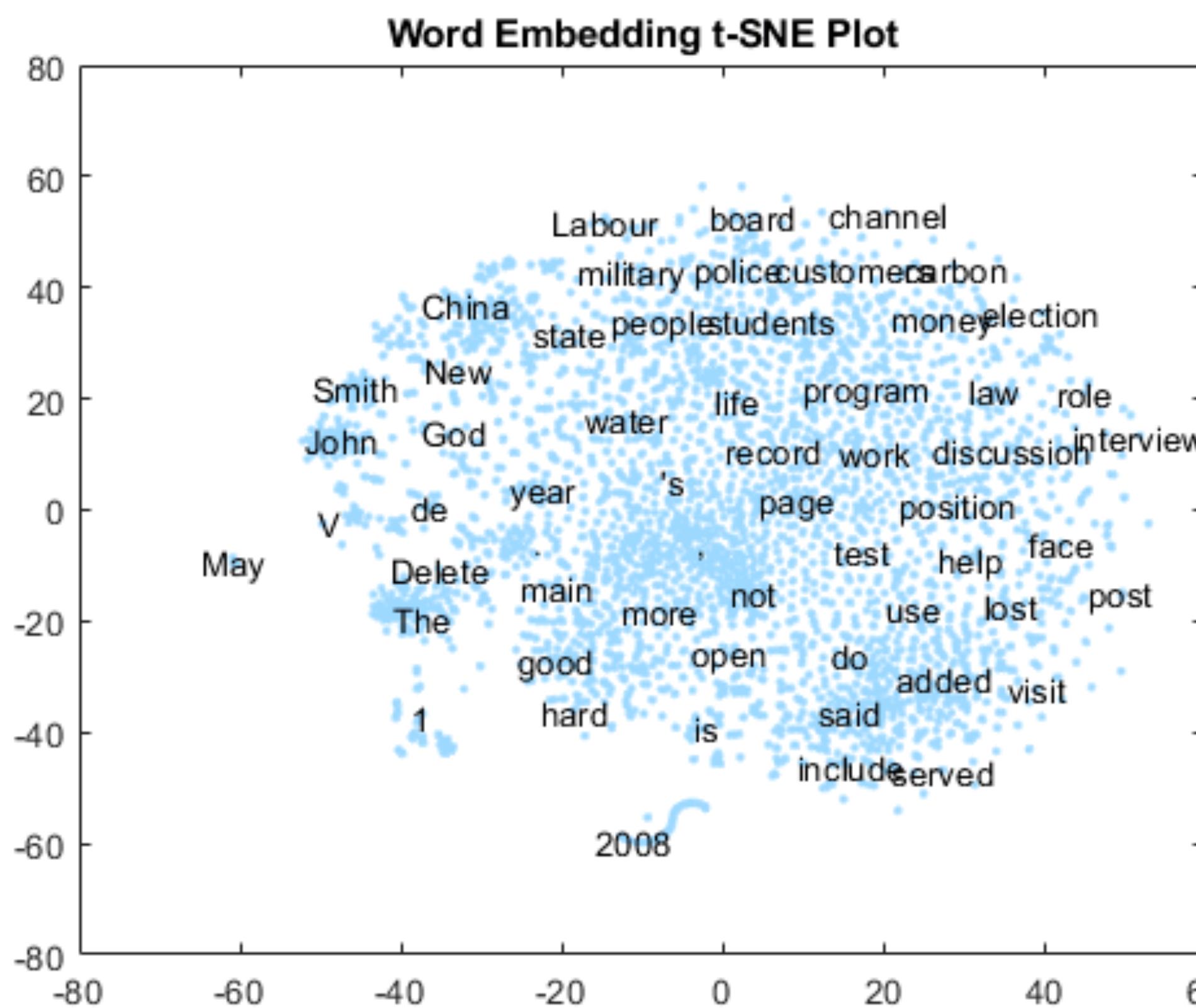
- Most common projection method: t-SNE



# Visualizing Embeddings

Project high-dimensional embeddings down into 2 dimensions

- Most common projection method: t-SNE
- Also: Principal Component Analysis (PCA)



# Analogy Relations

# Analogy Relations

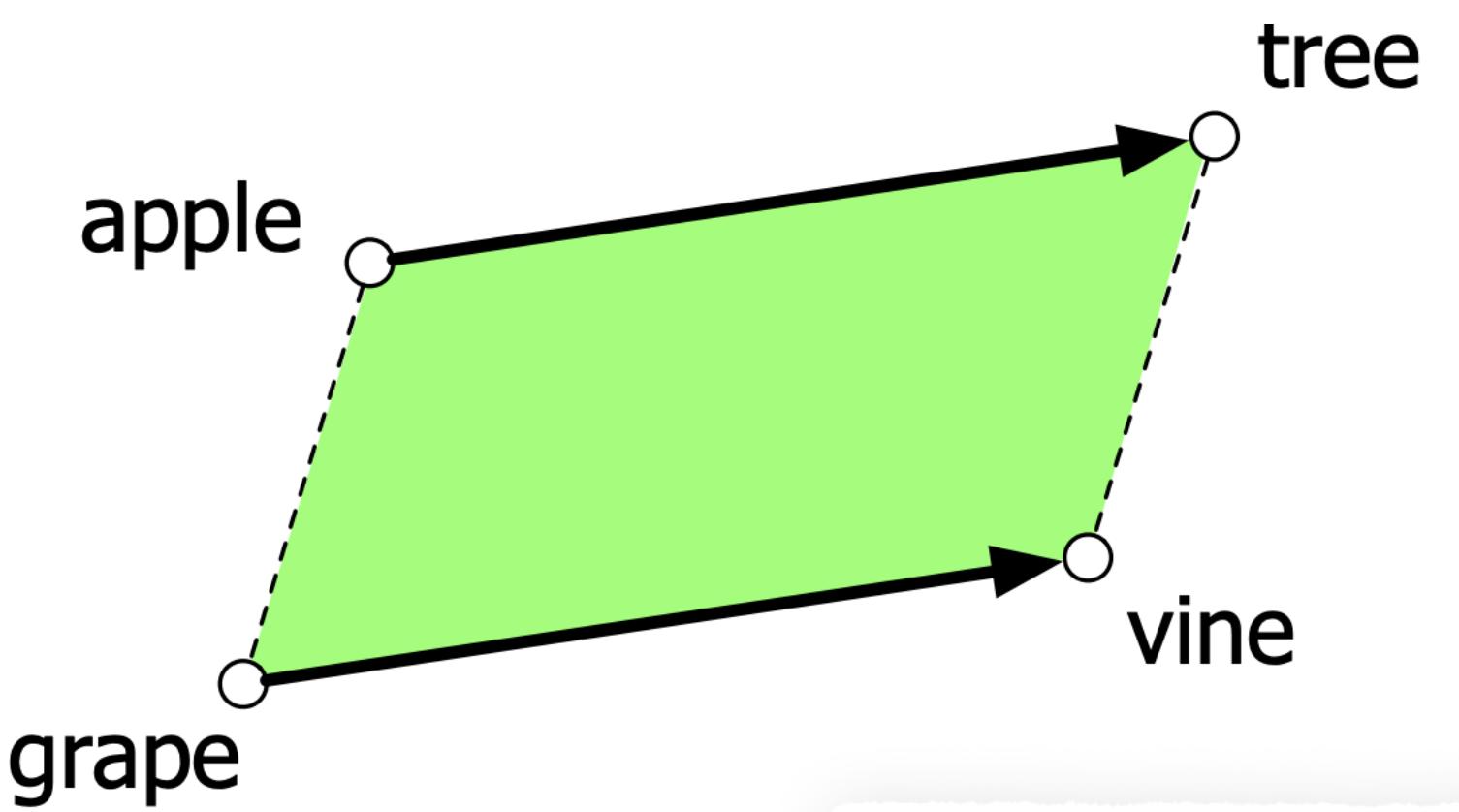
- The classic parallelogram model of analogical reasoning

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

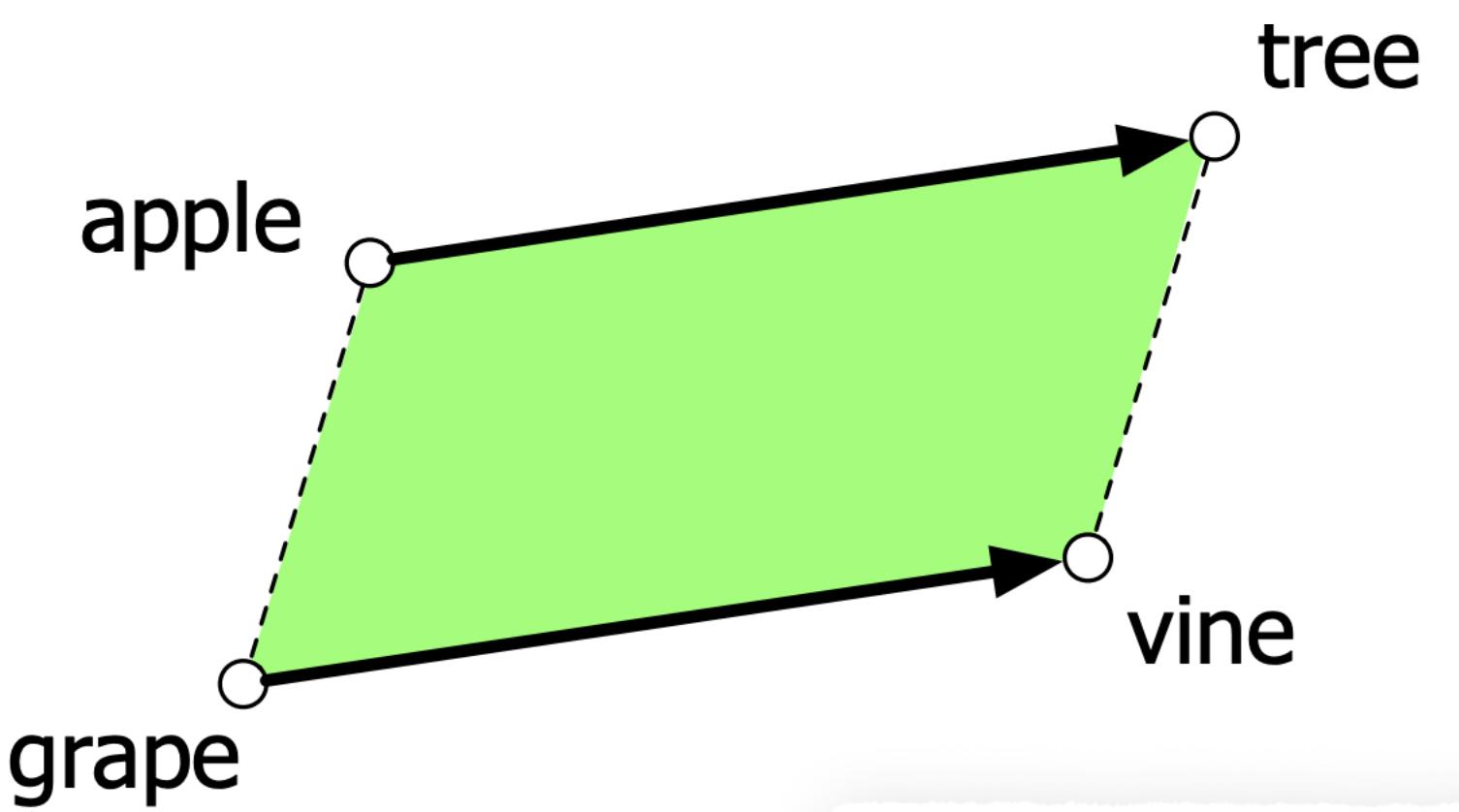


Rumelhart and Abrahamson, 1973

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

Add  $(\mathbf{w}_{apple} - \mathbf{w}_{tree})$  to  $\mathbf{w}_{grape}$  ...



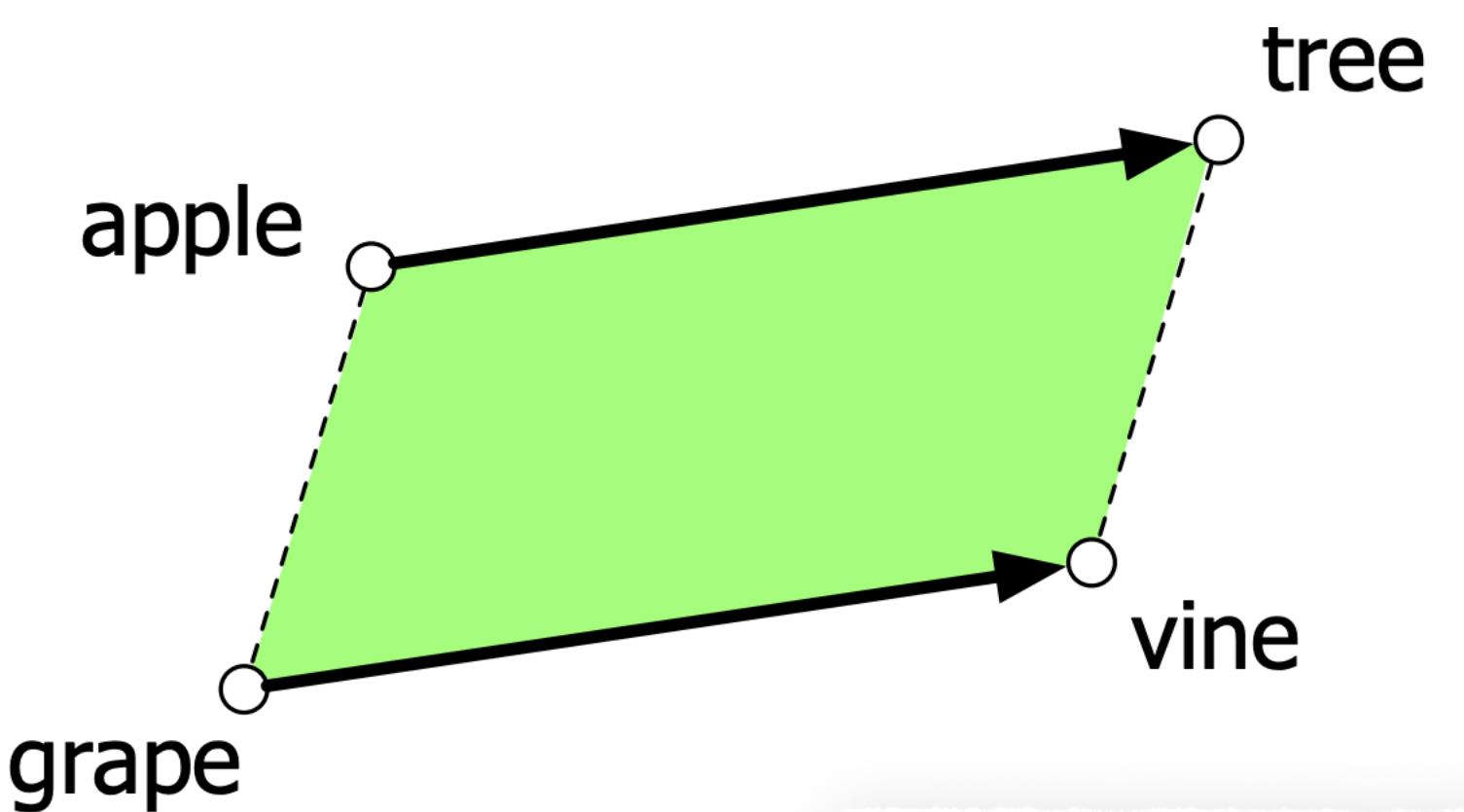
Rumelhart and Abrahamson, 1973

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

Add  $(\mathbf{w}_{apple} - \mathbf{w}_{tree})$  to  $\mathbf{w}_{grape}$  ...

Should result in  $\mathbf{w}_{vine}$



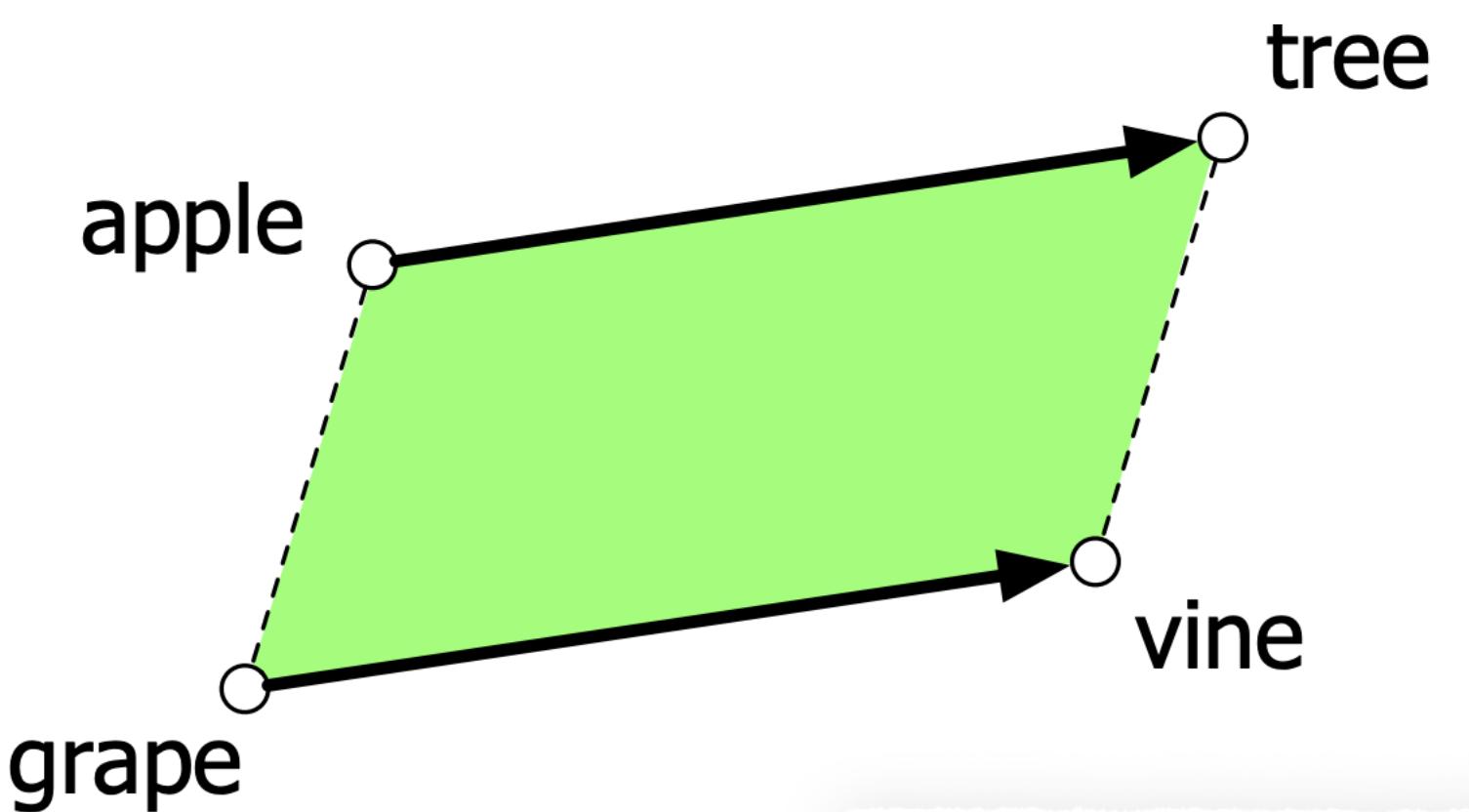
Rumelhart and Abrahamson, 1973

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

Add  $(\mathbf{w}_{apple} - \mathbf{w}_{tree})$  to  $\mathbf{w}_{grape}$  ...

Should result in  $\mathbf{w}_{vine}$



Rumelhart and Abrahamson, 1973

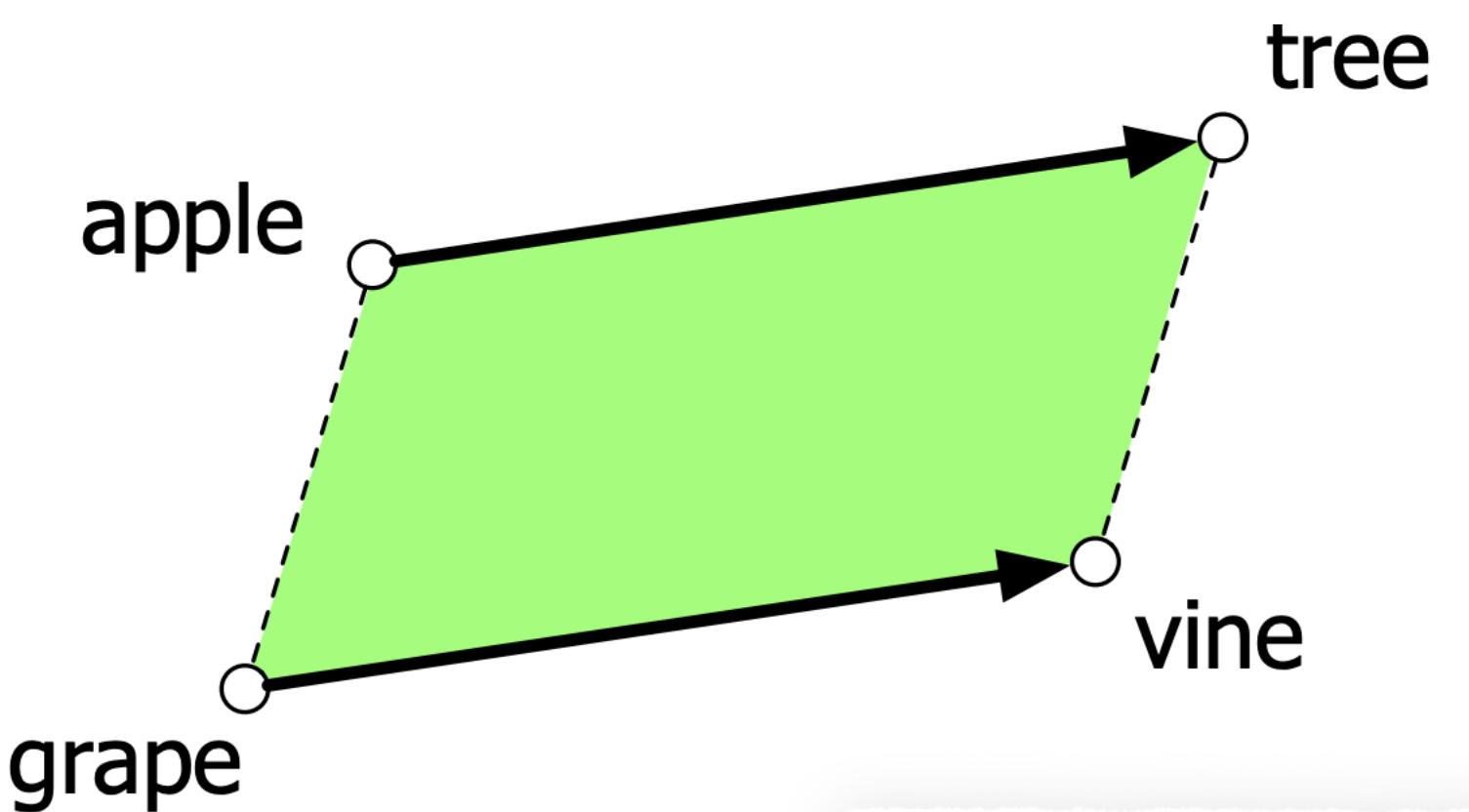
For a problem  $a : a^* :: b : b^*$ , the parallelogram method is:

# Analogy Relations

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

Add  $(\mathbf{w}_{apple} - \mathbf{w}_{tree})$  to  $\mathbf{w}_{grape}$  ...

Should result in  $\mathbf{w}_{vine}$



Rumelhart and Abrahamson, 1973

For a problem  $a : a^* :: b : b^*$ , the parallelogram method is:

$$\hat{b}^* = \arg \max_{\mathbf{w}} sim(\mathbf{w}, \mathbf{b} - \mathbf{a} + \mathbf{a}^*)$$

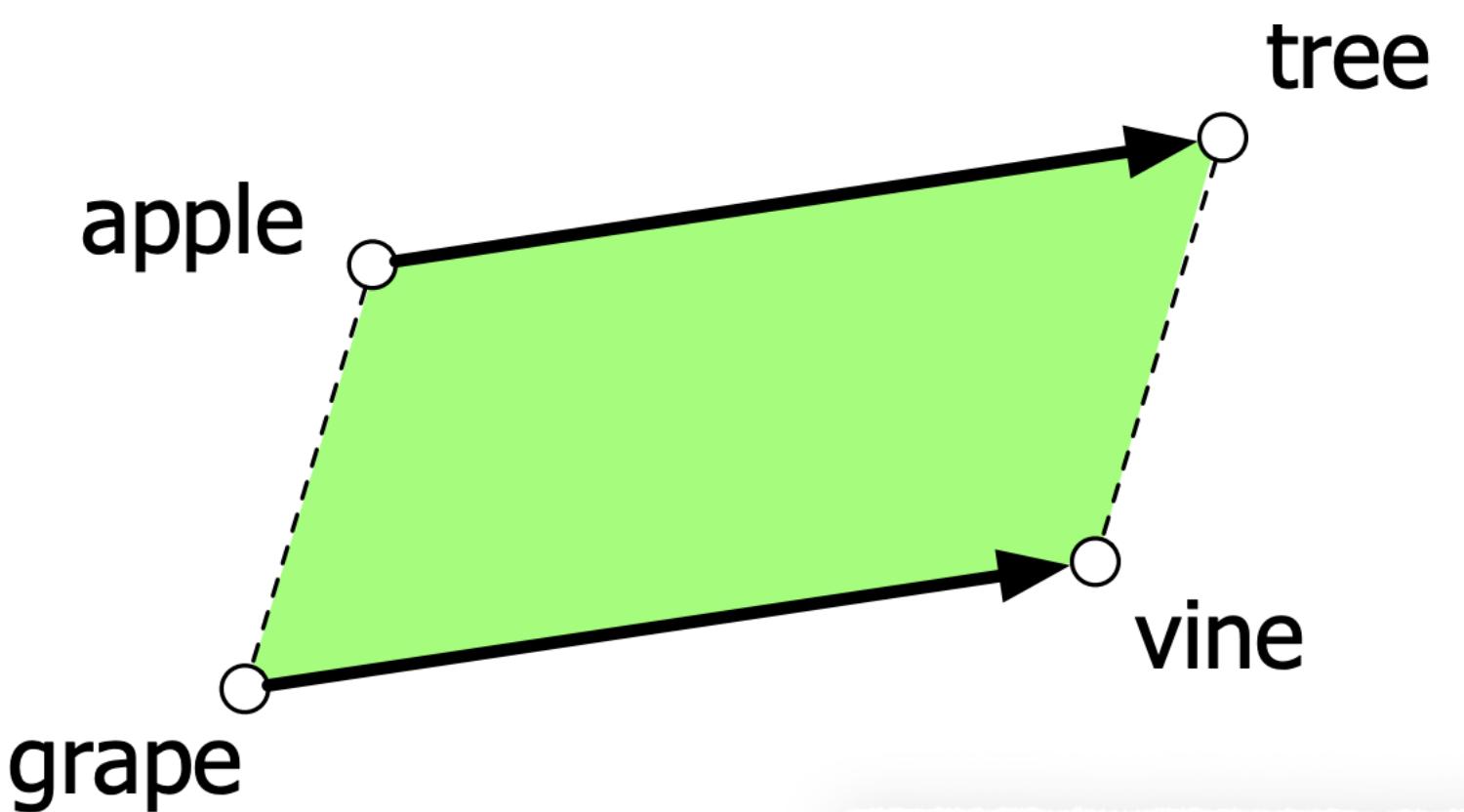
# Analogy Relations

Maximize similarity = minimize distance

- The classic parallelogram model of analogical reasoning
- Word analogy problem:
  - “Apple is to tree as grape is to ...”

Add  $(\mathbf{w}_{apple} - \mathbf{w}_{tree})$  to  $\mathbf{w}_{grape}$  ...

Should result in  $\mathbf{w}_{vine}$

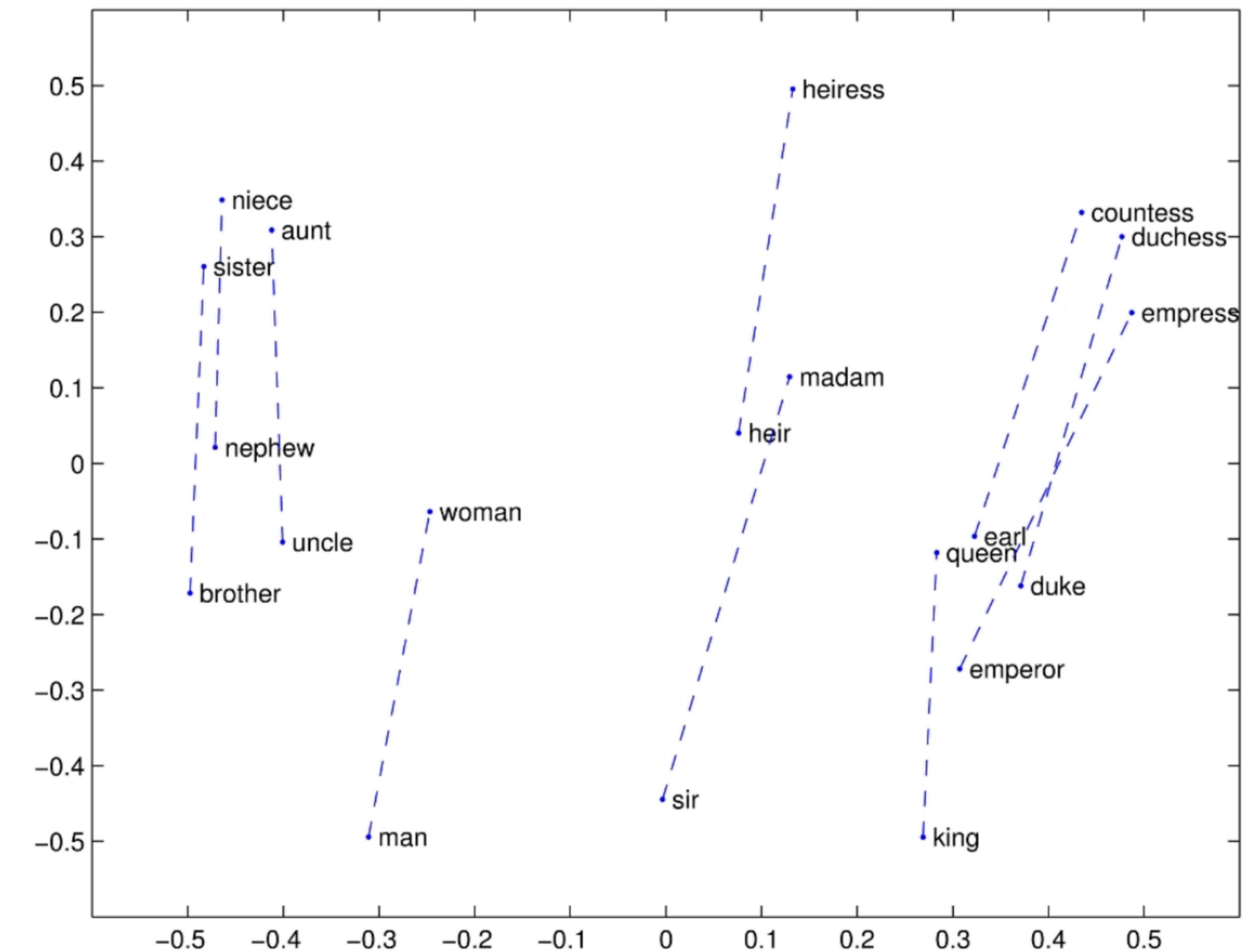


Rumelhart and Abrahamson, 1973

For a problem  $a : a^* :: b : b^*$ , the parallelogram method is:

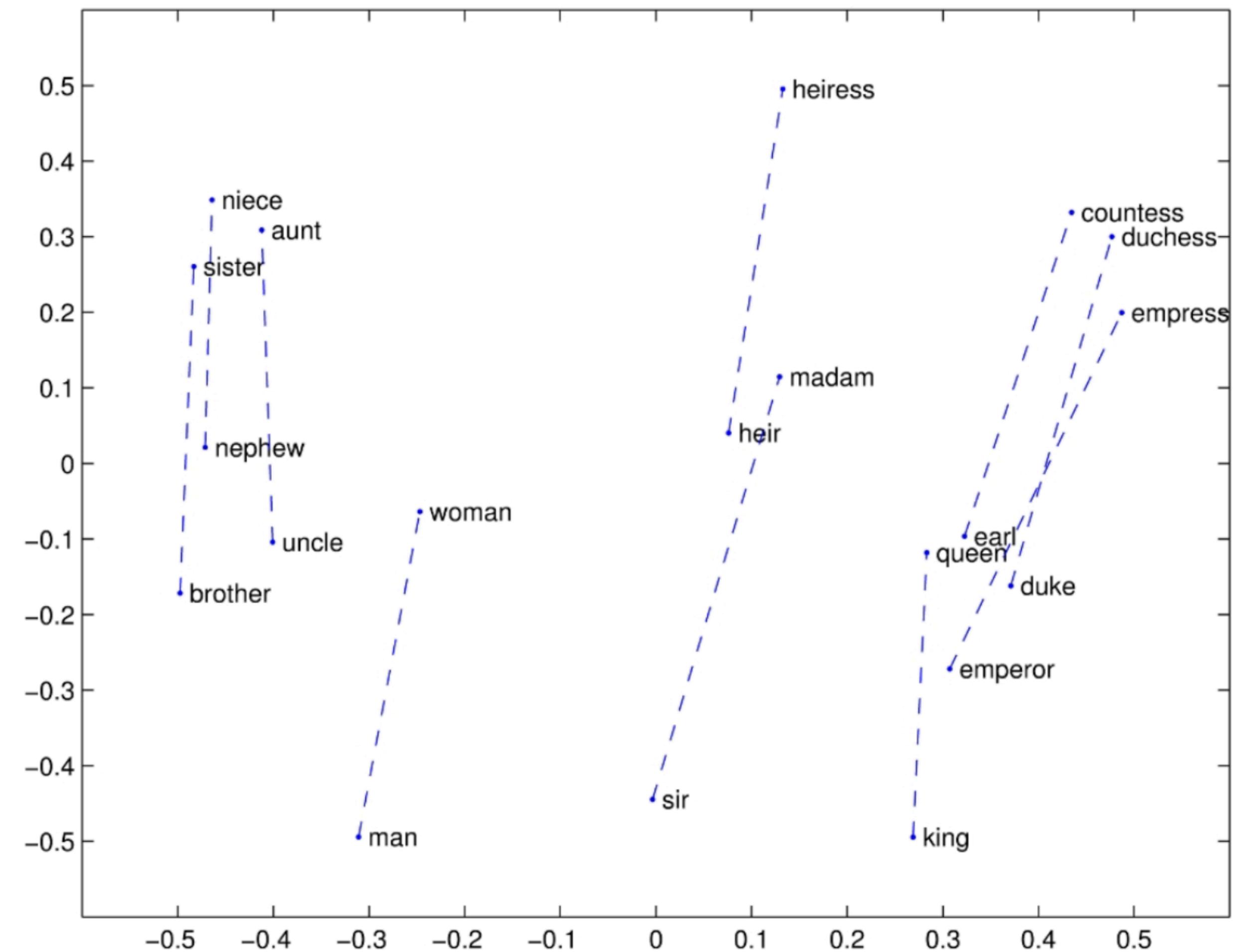
$$\hat{b}^* = \arg \max_{\mathbf{w}} sim(\mathbf{w}, \mathbf{b} - \mathbf{a} + \mathbf{a}^*)$$

# Analogy Relations: GloVe



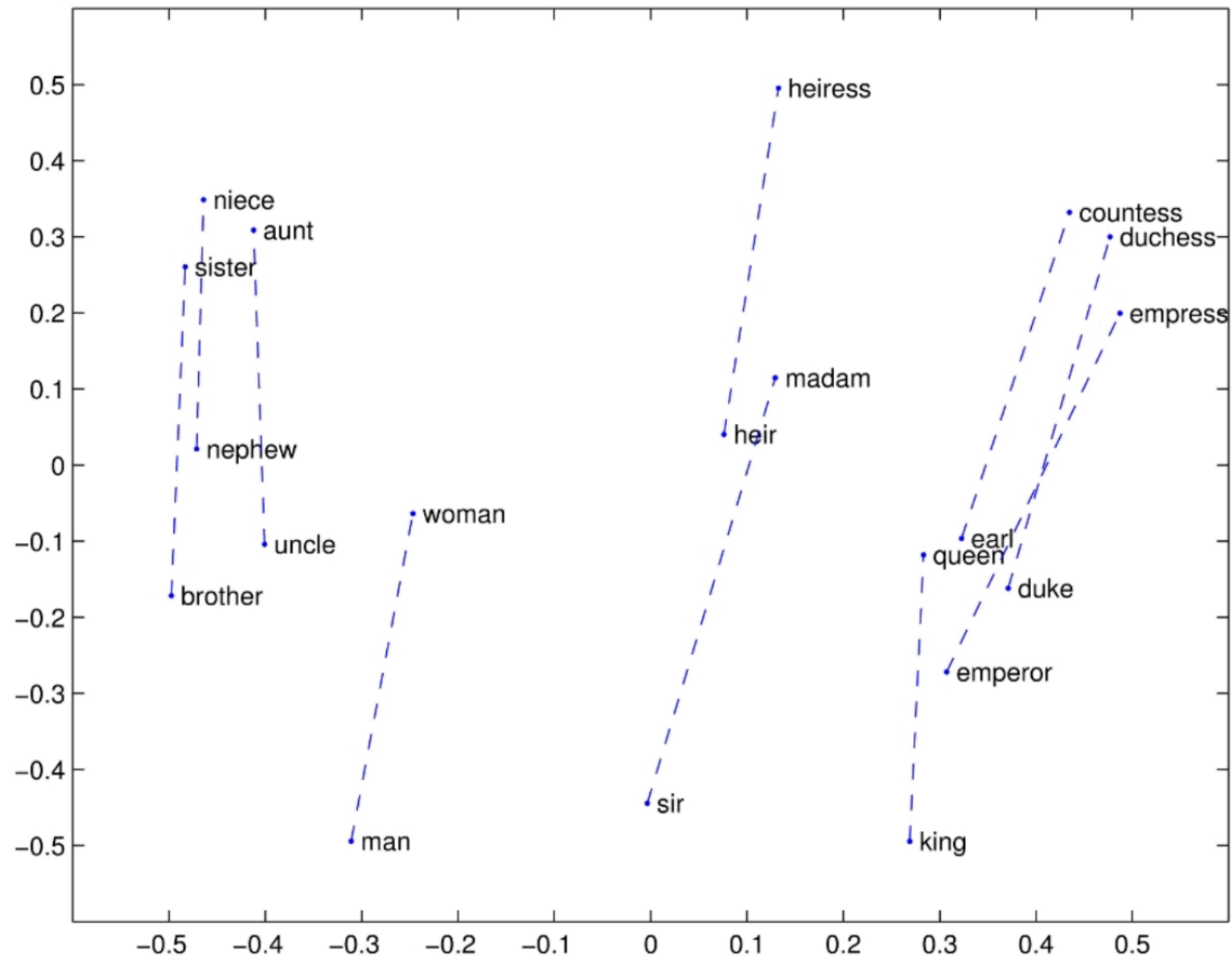
# Analogy Relations: GloVe

- Relational properties of the GloVe vector space, shown by projecting vectors onto two dimensions
- $\mathbf{w}_{king} - \mathbf{w}_{man} + \mathbf{w}_{woman}$  is similar to  $\mathbf{w}_{queen}$



# Analogy Relations: GloVe

- Relational properties of the GloVe vector space, shown by projecting vectors onto two dimensions
- $\mathbf{w}_{king} - \mathbf{w}_{man} + \mathbf{w}_{woman}$  is similar to  $\mathbf{w}_{queen}$
- Caveats: Only works for frequent words, small distances and certain relations (relating countries to capitals, or parts of speech), but not others



# Analogy Relations: GloVe

- Relational properties of the GloVe vector space, shown by projecting vectors onto two dimensions
- $\mathbf{w}_{king} - \mathbf{w}_{man} + \mathbf{w}_{woman}$  is similar to  $\mathbf{w}_{queen}$
- Caveats: Only works for frequent words, small distances and certain relations (relating countries to capitals, or parts of speech), but not others
  - Understanding analogy is an open area of research

