Introduction to Artificial Intelligence

Lecture 11: Machine learning for natural language processing

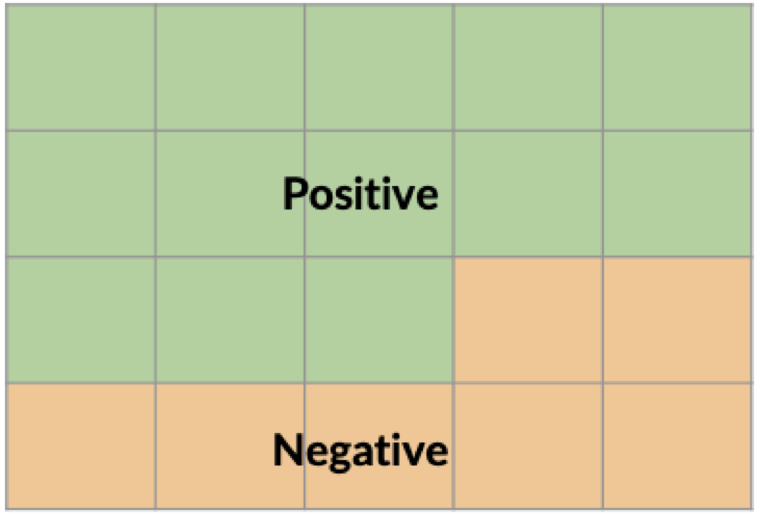
October 9, 2025

Lecture plan

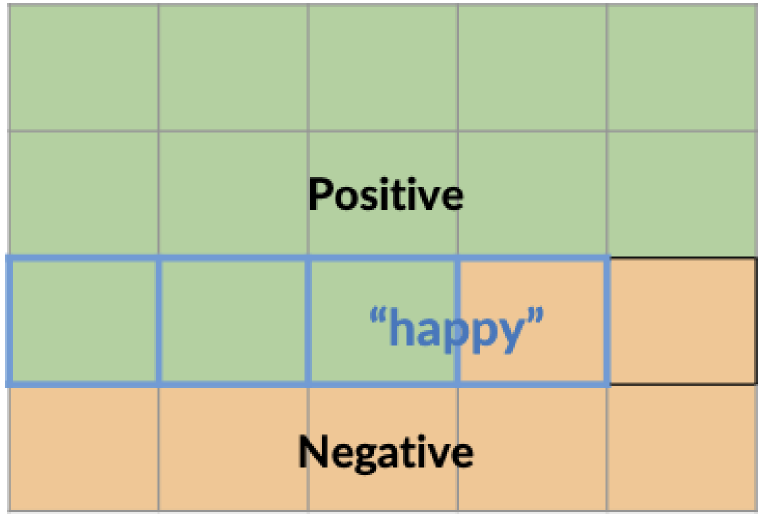
- Using machine learning for natural language processing
  - **Naïve Bayes classifier**
  - Logistic regression

Probabilities

Corpus of tweets

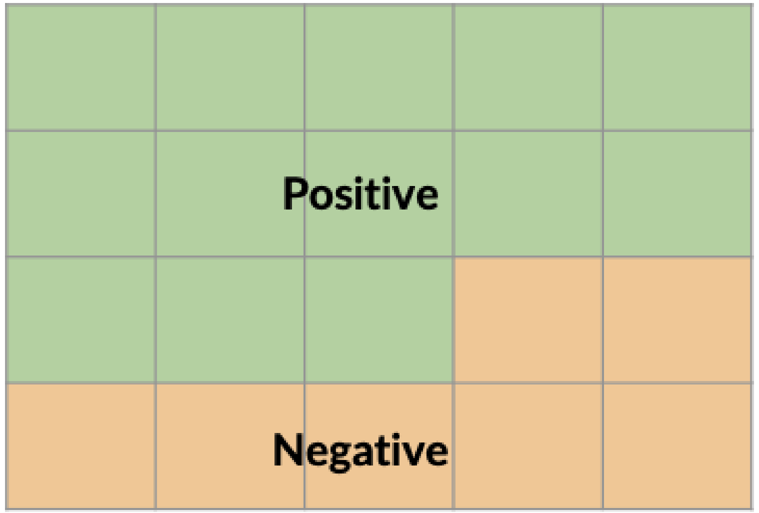


Tweets containing word "happy"



Probabilities

Corpus of tweets



A -> Positive tweet

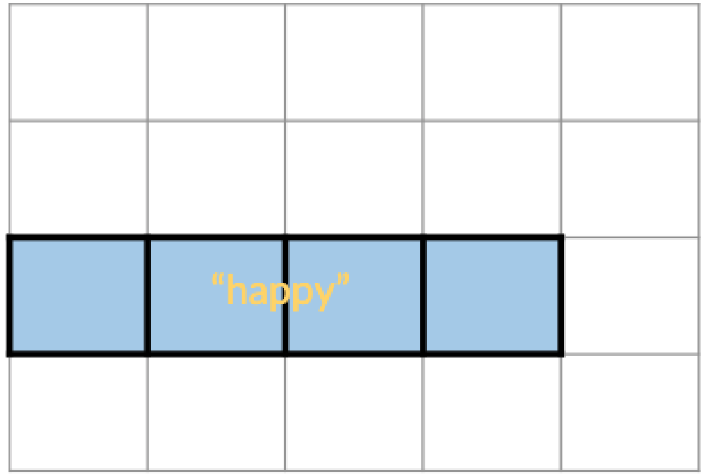
$$\Pr(A) = \Pr(\text{Positive}) = \frac{N_{pos}}{N}$$

$$\Pr(A) = \frac{N_{pos}}{N} = \frac{13}{20} = 0.65$$

$$\Pr(\text{Negative}) = 1 - \Pr(\text{Positive}) = 0.35$$

Probabilities of the intersection

Tweets containing the word “happy”



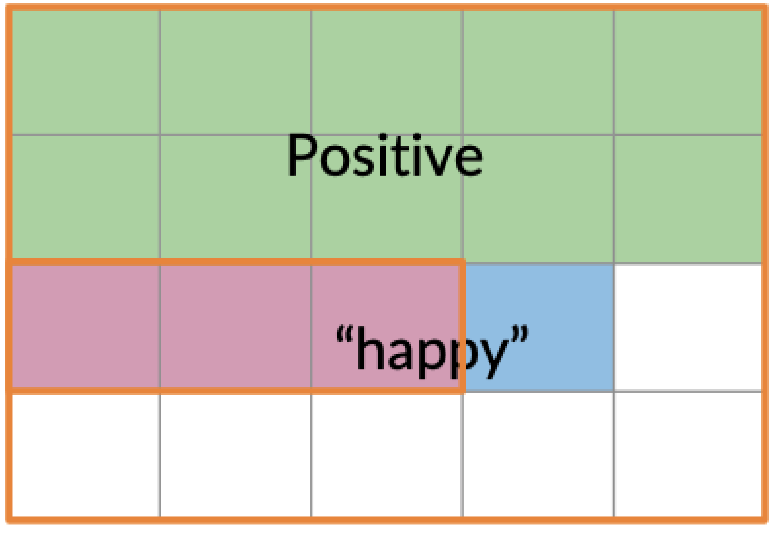
B -> tweet contains “happy”

$$\Pr(\mathrm{B}) = \Pr(\mathrm{Happy}) = \frac{N_{happy}}{N}$$

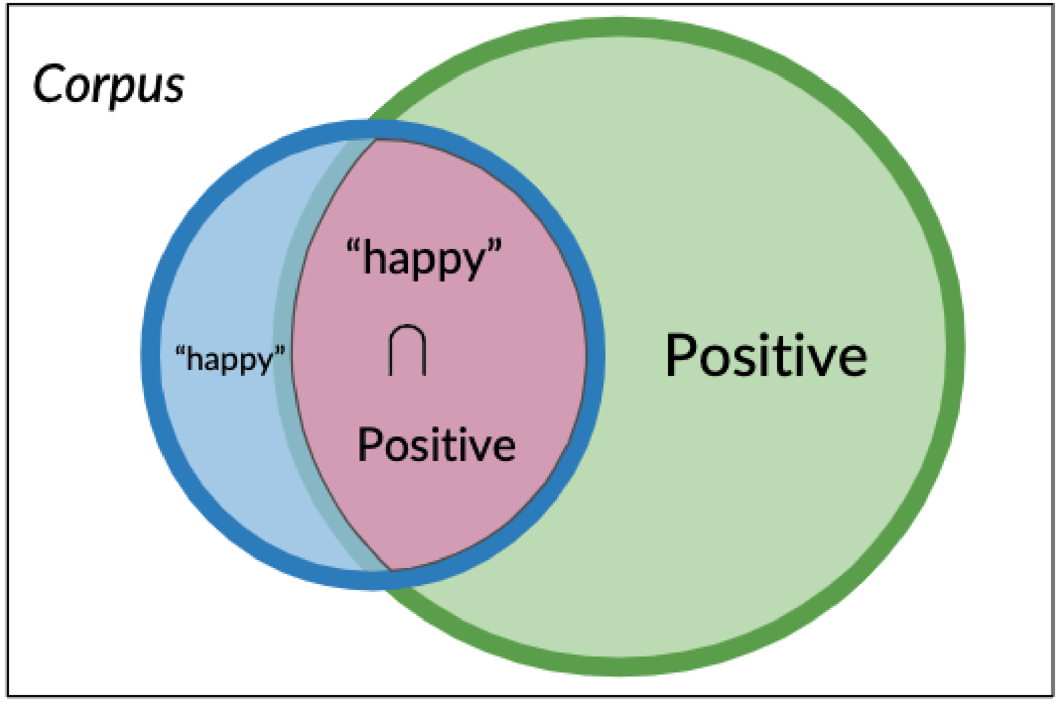
$$\Pr(\mathrm{B}) = \frac{4}{20} = 0.2$$

Conditional probabilities

- Probability of the intersection

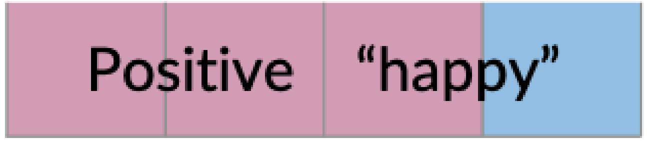


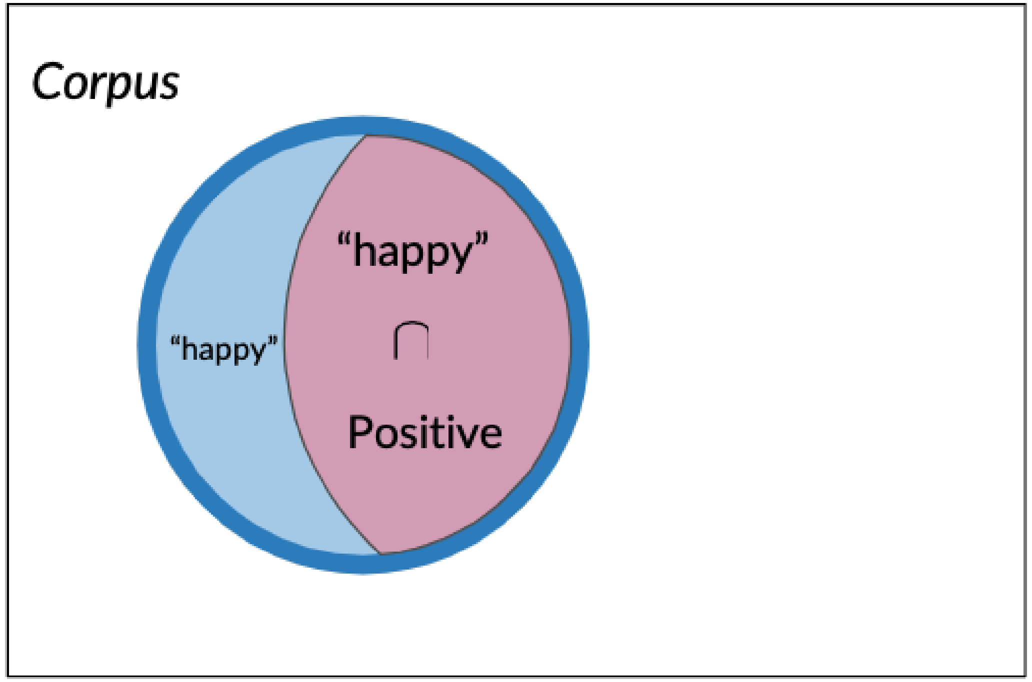
$$\Pr(A \cap B) = \Pr(A, B) = \frac{3}{20} = 0.15$$



Conditional probabilities

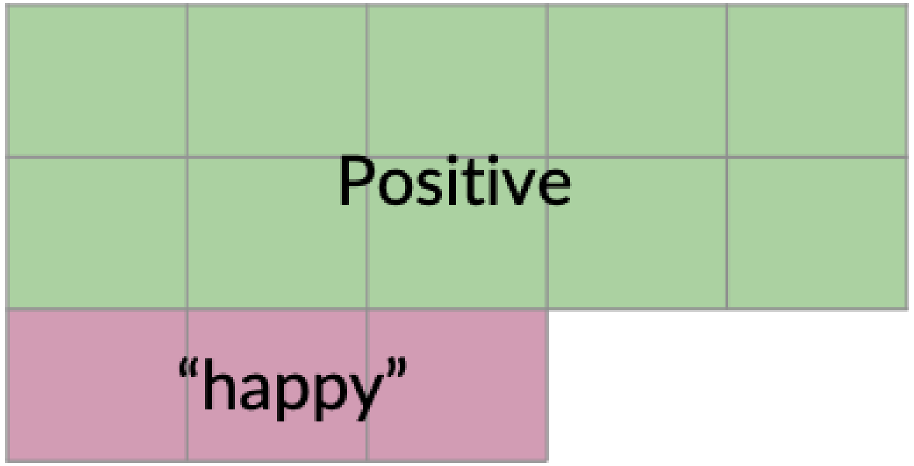
- $\Pr(A|B) =$ P(Positive | "happy")
- $\Pr(A|B) = \frac{3}{4} = 0.75$

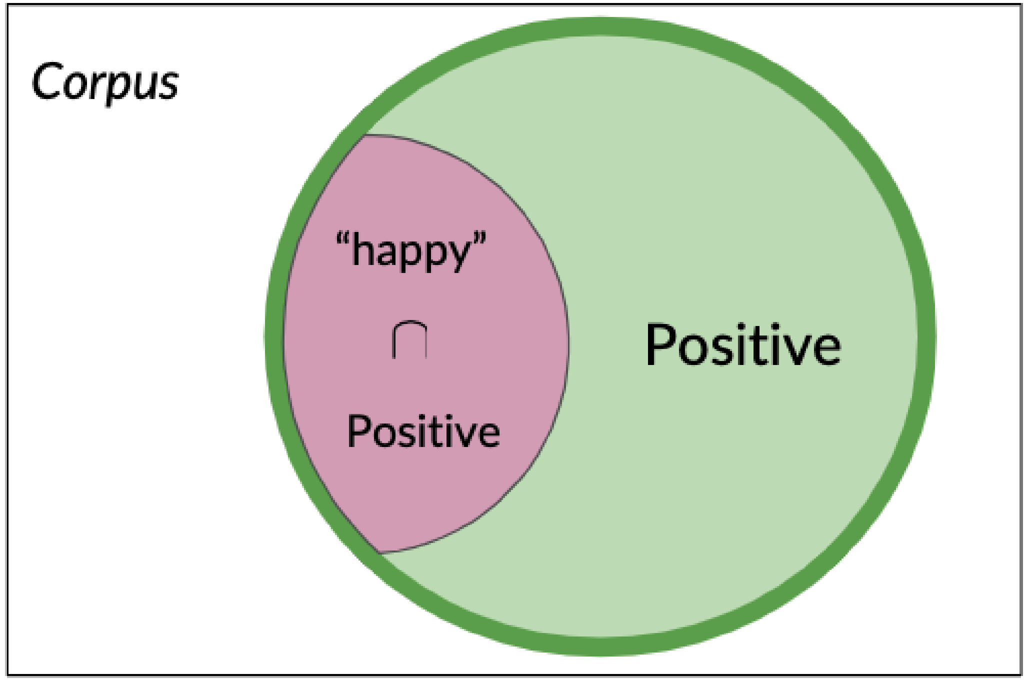




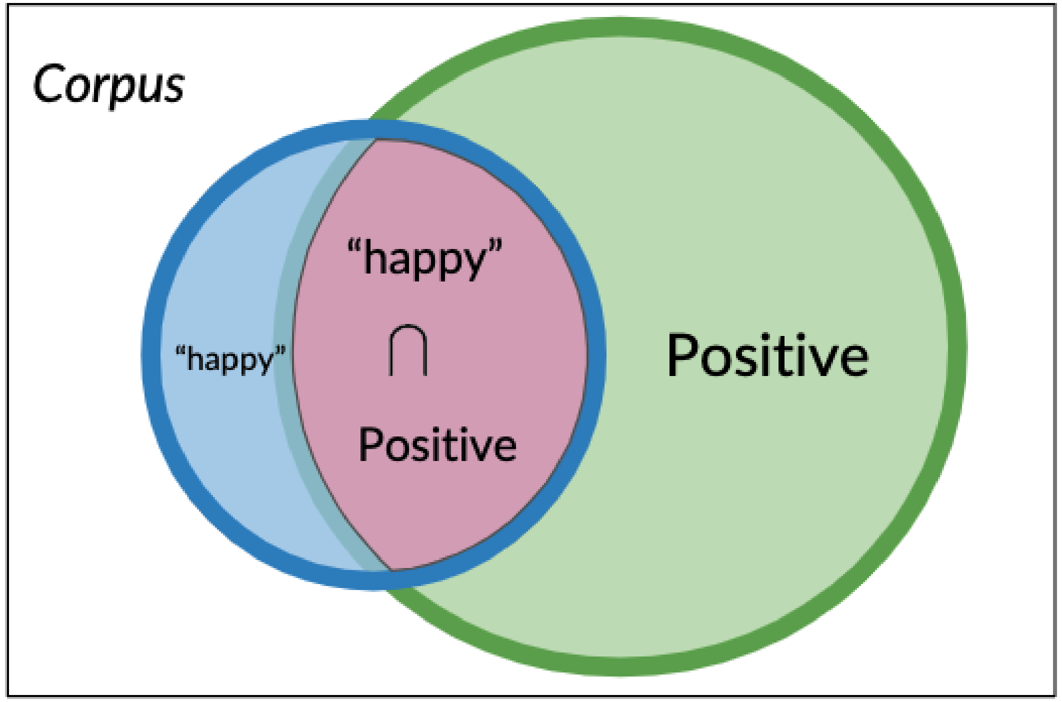
Switching the conditional probabilities

- $\Pr(B|A) =$ P("happy" | Positive)
- $\Pr(B|A) = \frac{3}{13} = 0.231$





Bayes' rule: Combining both sides together



$$\Pr(\text{Positive} \mid \text{“happy”})$$

$$= \frac{\Pr(\text{Positive} \cap \text{“happy”})}{\Pr(\text{“happy”})}$$

Bayes' rule

- Quiz: what is Pr("happy" | Positive)?

Sentiment analysis example

- Positive tweets
  - I am happy because I am learning NLP
  - I am happy, not sad
- Negative tweets
  - I am sad, I am not learning NLP
  - I am sad, not happy

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2

Naïve Bayes for sentiment analysis

- Normalize the occurrences
- Pr(word | class)

word	Pos	Neg
I	3/13	3/13
am	3/13	3/13
happy	2/13	1/13
because	1/13	0/13
learning	1/13	1/13
NLP	1/13	1/13
sad	1/13	2/13
not	1/13	2/13

Naïve Bayes for sentiment analysis

- Assumes conditional independence

$$\mathbf{Pr}(A_1, A_2, \dots, A_n | y) = \mathbf{Pr}(A_1 | y) \cdot \mathbf{Pr}(A_2 | y) \cdot \dots \cdot \mathbf{Pr}(A_n | y)$$

- Tweet: I am happy; I am learning

$$\prod_{i=1}^{6} \frac{\Pr(w_i \mid pos)}{\Pr(w_i \mid neg)} = \frac{2}{1} > 1$$

Laplacian smoothing

- Estimate the conditional probability given one hypothesis:

$$\Pr(A_i|y) = \frac{count(\text{“}A_i\text{”}, y)}{count(w, y)}$$

- Use the add one smoothing

$$\Pr(A_i|y) = \frac{count(\text{“}A_i\text{”}, y) + 1}{count(w, y) + U_{class}}$$

Where U_{class} is the number of unique words in class

Naïve Bayes inference

- Choose the most likely hypothesis given the list of words
  - Hypothesis y is Positive, Neural, or Negative
  - Use Bayes rule: get *likelihood* and *prior*

$$\arg\max_y \Pr(y \mid A_1, A_2, \dots, A_n) = \arg\max_y \frac{\Pr(A_1, A_2, \dots, A_n | y) \cdot \Pr(y)}{\Pr(A_1, \dots, A_n)}$$

- Compare

$$\frac{\Pr(\mathrm{pos})}{\Pr(neg)} \prod_{i=1}^{n} \frac{\Pr(w_i | pos)}{\Pr(w_i | neg)} > 1$$

- Quiz: write the inference rule in terms of log likelihood?

Training

1. Get or annotate a dataset with positive and negative tweets

2. Preprocess the tweets

3. Compute frequency of each word in every class

4. Compute conditional probability $P(w \mid pos)$, $P(w \mid neg)$ and prior

5. Compute log likelihood

Applications of Naïve Bayes

- Spam filtering: $\frac{\Pr(spam \mid email)}{\Pr(non\text{-}spam \mid email)}$

- Information retrieval: Pr(doc_k | query) ≈ Π Pr(query_i | doc_k)

- Word disambiguation: Pr(river | text) / Pr(money | text)

Lecture plan

- Using machine learning for natural language processing
  - Naïve Bayes classifier
  - **Logistic regression classifier**

Vocabulary

- Tweets: [tweet_1, tweet_2, …, tweet_m]
  - I am happy because I am learning NLP
  - I am sad, not happy

- V=[I, am, happy, because, learning, NLP, …,sad, not]

Feature extraction

- Directly encode the tweet in the vocabulary
  - I am happy because I am learning NLP
  - V=[I, am, happy, because, learning, NLP, …,sad, not]

- Representation: [1, 1, 1, 1, 1, 1, 1, 1, 0, 0]

- Large representation that grows with vocabulary size -> higher training and prediction time

Word frequency in classes

- Positive tweets
  - I am happy because I am learning NLP
  - I am happy, not sad
- Negative tweets
  - I am sad, I am not learning NLP
  - I am sad, not happy

- **Map every tweet to pos, neg frequencies**
  - Clean unimportant information from tweets

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2

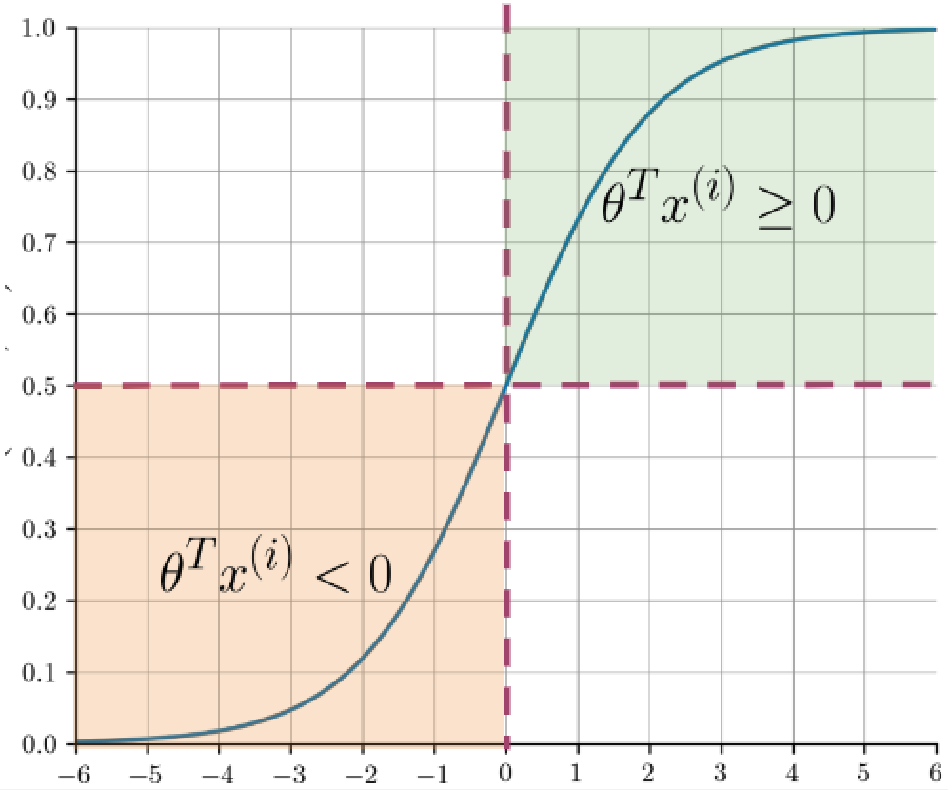
Preprocessing

- Remove stop words and punctuation
  - And, is, are, at, has, for, a
  - , . : ! " '
  - URLs

- Lowercasing: Great / GREAT -> great

Logistic regression

- Zero-one loss: Loss value is zero if predicted label is correct, is one otherwise
- Logistic loss provides an approximation of the zero-one loss
  - Stanford logistic function: $\frac{1}{1+e^{-v}}$
  - $v = \theta^\top x^{(i)}$



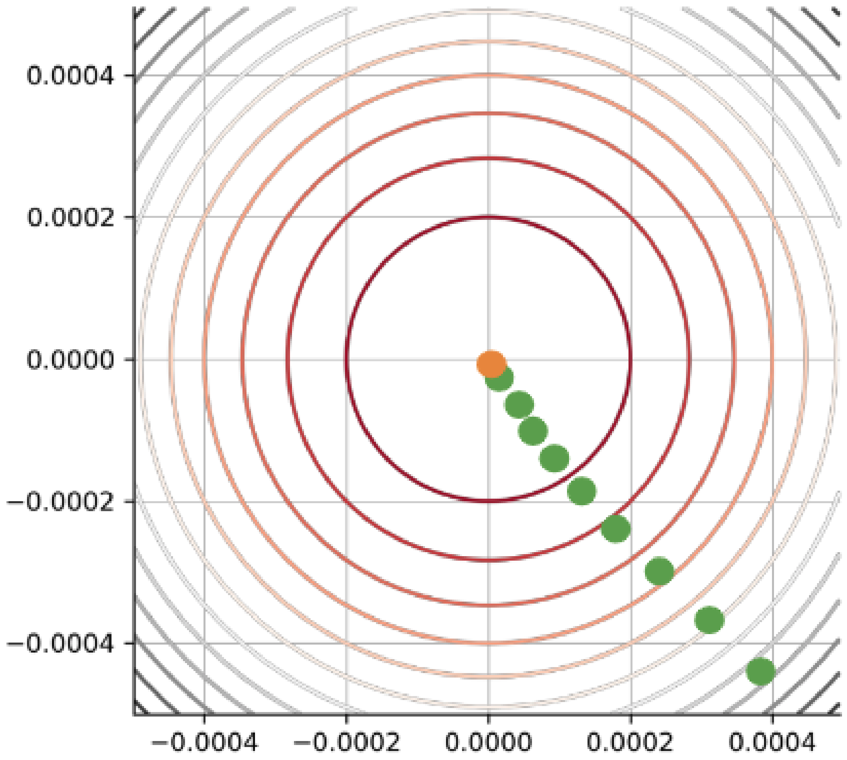
Sigmoid function

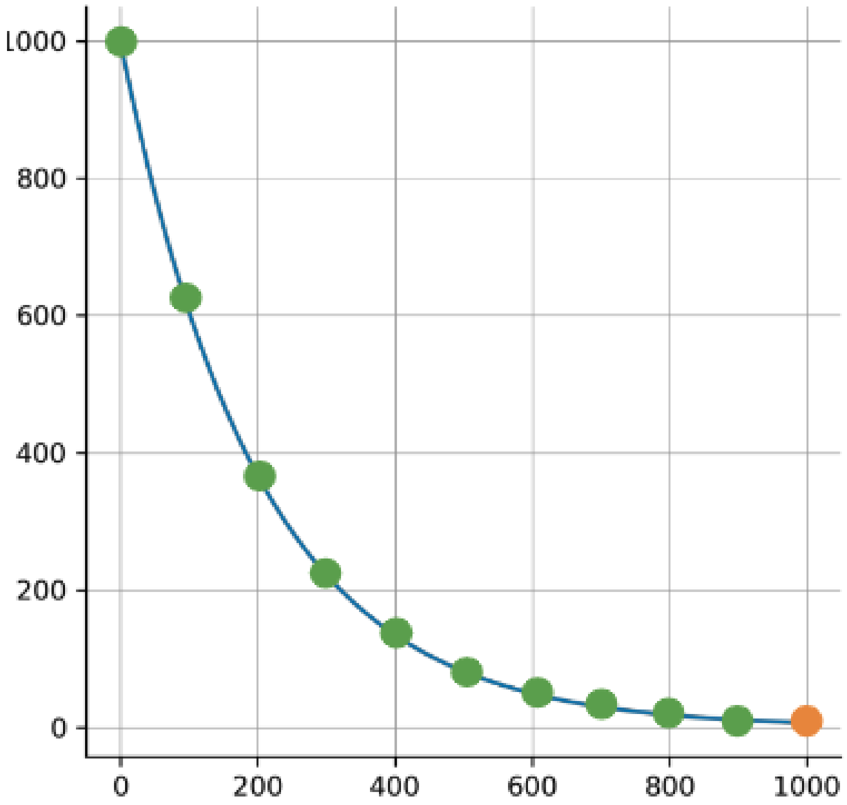
Prediction

- If $v_i > 0$, $\hat{y}_i = +1$ or positive
- If $v_i \leq 0$, $\widehat{y_i} = -1$ or negatie

Training

- Use stochastic gradient descent





- Quiz: how does testing work?

Cost/objective function for logistic regression

- Suppose the labels are either $+1$ or -1
- The log-loss of one sample x_i, y_i is

$$\log(1 + \exp(-y_i \cdot v_i))$$

- Averaged training loss over a dataset of size n

$$\frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \cdot v_i))$$

- Strong disagreement = high cost
- Strong agreement = low cost