Natural Language Processing
CSE 325/425

Sihong Xie

Lecture 2:

- Language models
- MLE and Laplacian

Sample space, events, random variable

- NLP is about using probability to model languages, and using optimization to find the right model.
- The universal set Ω contains all objects that can occur in a specific context.
  - Discrete: Ω = All words in a corpus (vocabulary)
  - Continuous: Ω =Sentimental score in [-1, 1]
- Event: subsets of Ω ($A \subseteq \Omega$)
  - A document: bag of words (subset of the vocabulary)
  - Positive sentimental score [0, 1]
- Random variable $X : \Omega \rightarrow \text{set of numbers}$
  - e.g., map a word to an integer index.
  - The sentimental score is a number already.
- Probability distribution of a random variable $P(X = 1) = P(\omega \in \Omega : X(\omega) = 1)$
- Probability of an event $A \subseteq \Omega$:

$$P(A) \rightarrow [0, 1]$$

|V| words in the vocabulary V

m words

Random variable

P(a doc) = P($w_1, w_2, \ldots, w_m$)

m words in Doc

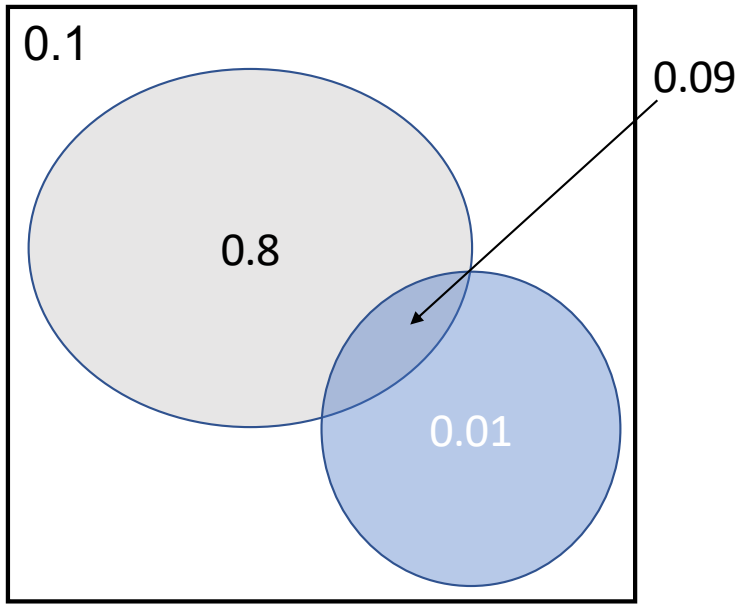
Joint Distribution

$|V|^m$ Comb

Recipe for making a joint distribution of m Rvs

1. List all possible combinations of values of the RVs.
2. Assign valid probability to each combination.

A	B	P(A, B)	
T	T	0.09	=P(A∩B)
T	F	0.8	=P(A∩B^c)
F	T	0.01	=P(B∩A^c)
F	F	0.1	=P(B^c∩A^c)



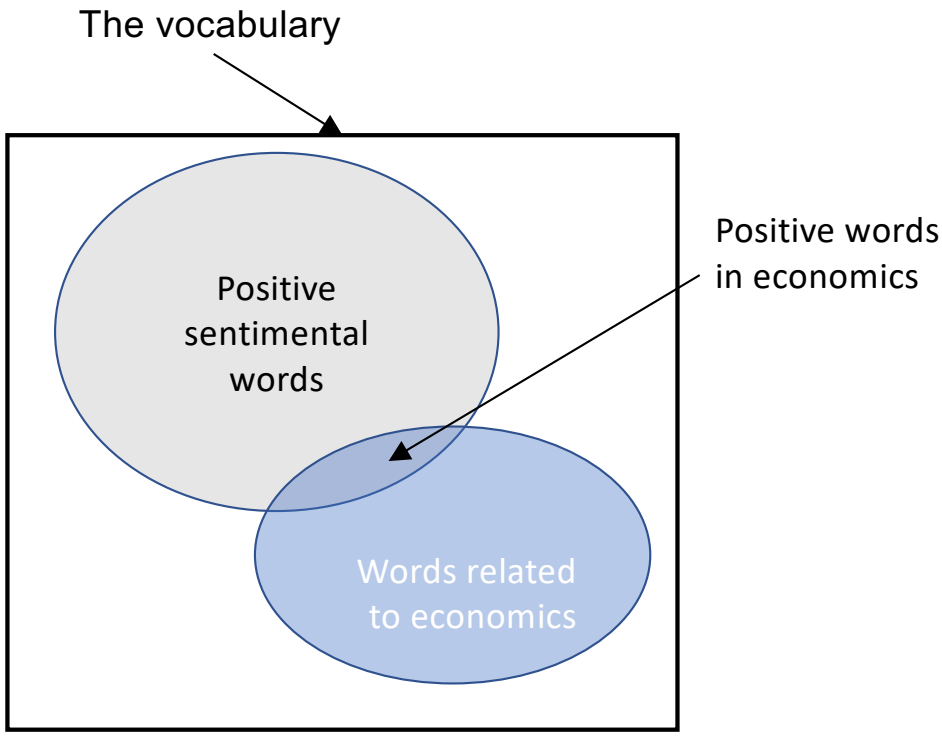
Venn Diagram

$$P(A) = P(A \cap B) + P(A \cap B^c) \qquad P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Axioms

The Axioms of Probability

- $P(\Omega) = 1$
- $0 \leq P(A)$ $A \subseteq \Omega$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$



Useful theorems

Complement of event A

Theorems:

- $P(A) + P(A^c) = 1$

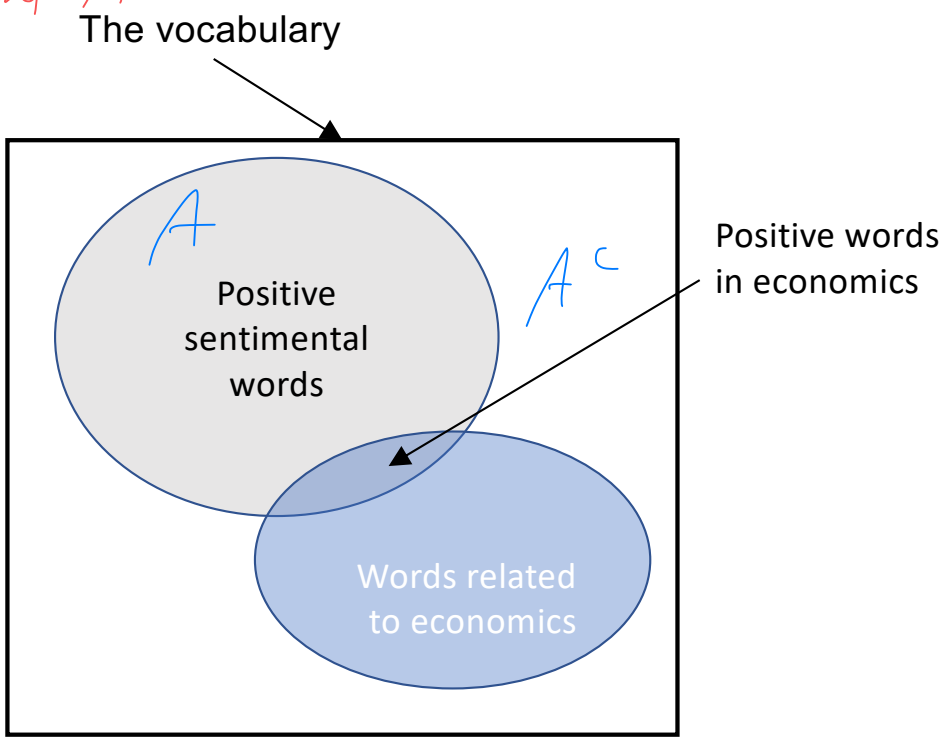
- Total Prob $P(A) = P(A \cap B) + P(A \cap B^c)$

- Conditional: $P(A|B) = \dfrac{P(A \cap B)}{P(B)}$

Definition

$$P(A \cap B) = P(A)P(B|A)$$

$$= P(B)P(A|B)$$



Useful theorems

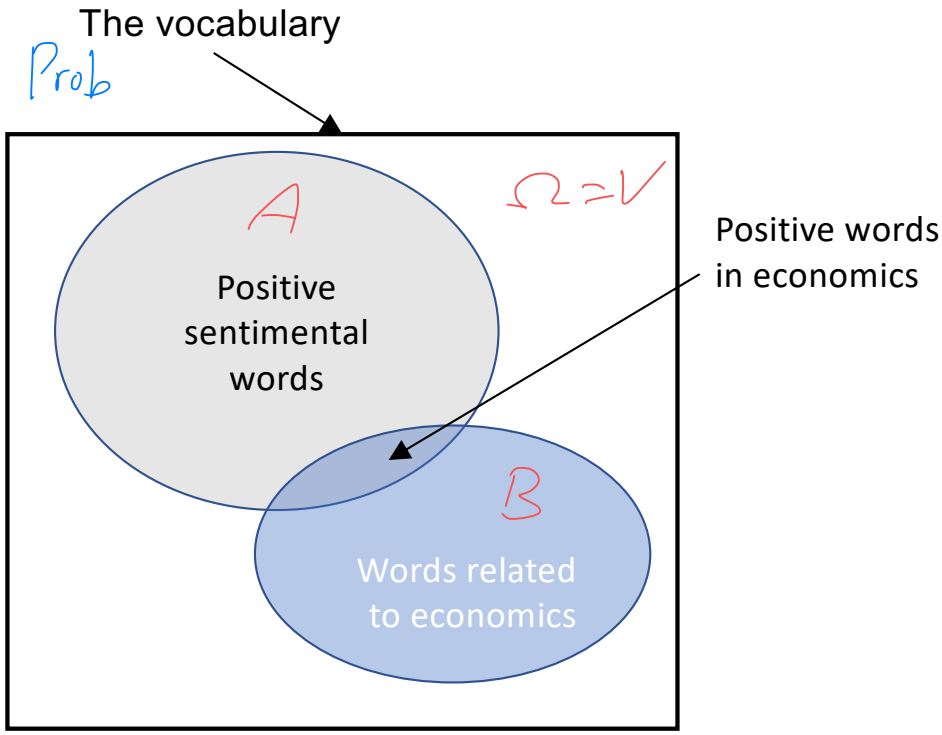
By definition of conditional Prob

Bayes Rule:

- $$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$$

By definition of Joint Prob.

- Prior: $P(B)$
- Likehood : $P(A|B)$
- Posterior: $P(B|A)$
- Marginal: $P(A)$



Useful theorems

Bayes Rule:

- $$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}$$

$$\Downarrow$$

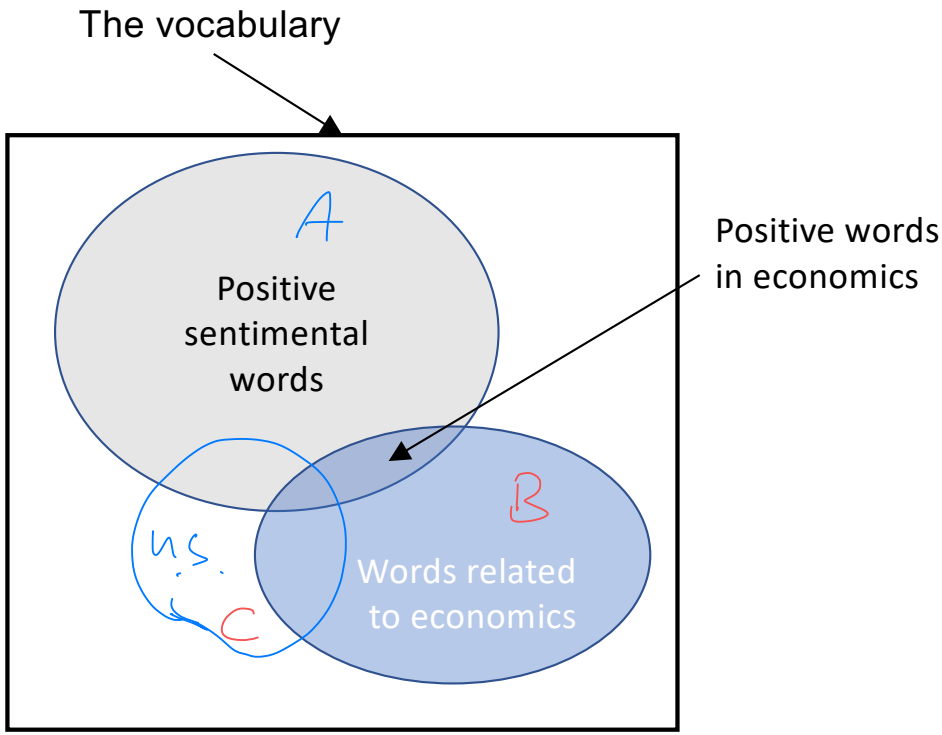
$$P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B^c)P(B^c)}$$

$B \cap B^c = \emptyset$

$$P(B|A \cap C) = \frac{P(A|B \cap C)P(B \cap C)}{P(A \cap C)}$$

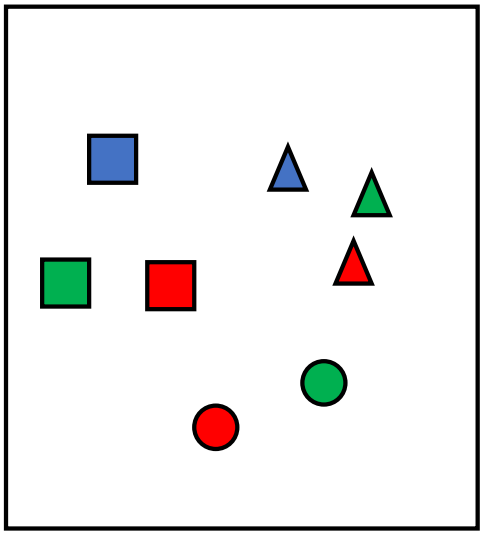
$$= \frac{P(C|B \cap A)\,P(A \cap B)}{P(A \cap C)}$$

$\neq P(A \cap C|B)$



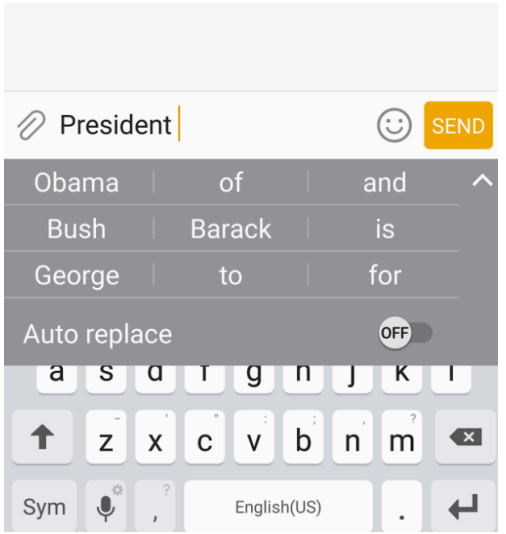
More exercises

Sample space Ω



- Joint probability
  - P(red □) = 1/8
- Conditional probability
  - P(red | □) = 1/3
  - P(□ | red) = 1/3 1/4
- Total probability
  - P(red) = P(red | □) x P(□) + P(red | △) x P(△)+ P(red | ○) x P(○)
    1/3 3/8 1/3 3/8 1/2 2/8
- Bayes rule
  - P(red | □) = P(red□) / P(□)
    1/3 1/8 3/8

Language models



Language models

- Define the probaility of a sequence of words (e.g., a sentence).
- $P(w_1, \ldots, w_n)$
- Can be used for spell-checking: *P("the book") >> P("book the")*
- Can be used to predict the next word: *P("Obama"|"President") >> P("book"|"President")*
- Many different ways to interprete this probability:
  - Uni-gram: $P(w_1)P(w_2)\ldots P(w_n)$
  - Bi-gram: $P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1})$
  - Tri-gram:
- Where do these probabilities come from?

won't be able to find long-range dependency

Language models

Language models training

- Training corpus.
- Uni-gram: relative frequency of each word $$P(w) = \frac{\text{Count of } w}{\text{Count of the total tokens}}$$
- Bi-gram: conditional probability of a word given another word $$P(w|v) = \frac{\text{Count of } (vw)}{\text{Count of } v}$$
- Example
  - Estimate P(Sam) and P(I|do)

The corpus = {
D1=[<s> I am Sam </s>] = length 5
D2=[<s> Sam I am </s>] = length 5
D3=[<s> I do not like green eggs and ham </s>] = length 10
}

$P(Sam) = \frac{2}{20} = 1/10$

$P(I|do) = \frac{0}{\quad} = 0$

$P(am|I) = \frac{2}{3}$

Language models

Common issues in language models

- Time/space complexity: n-gram has $O(|V|^n)$ complexity.
- Sample complexity: how many data are needed to estimate a quantity.
  - Data sparsity: corpus is not enough to reliably estimate the probabilities.
  - Recall the large-number-theorem.
  - Flipping a coin once and estimate *P(head)*

|V| : size of the vocab V

P(Sam|I)

Laplacian smoothing

- Missing words and phrases (in training corpus)
  - Laplacian smoothing for unigram

$$P_{\text{Laplacian}}(w) = \frac{(\text{Count of } w) + 1}{(\text{Count of the total tokens}) + |V|}$$

HW: $\sum_{w \in V} P_{Lap}(w) = 1$

$0 \leq \quad \leq 1$

← Size of the vocab V

  - Verify that the smoothed estimation is a distribution over the vocabulary.
  - Laplacian smoothing for bi-gram

$$P_{\text{Laplacian}}(w_2|w_1) = \frac{(\text{Count of } w_1 w_2) + 1}{(\text{Count of } w_1) + |V|}$$

  - Use the following corpus to make Laplacian estimation.

The corpus = {
D1=[<s> I am Sam </s>]
D2=[<s> Sam I am </s>]
D3=[<s> I do not like green eggs and ham </s>]

Yon ∈ V }

$P_{Lap}(\text{"Yon"}) = \frac{0+1}{20+|V|}$