Language Models

Data Science: Jordan Boyd-Graber
University of Maryland
SLIDES ADAPTED FROM PHILIP KOEHN

Language models

- **Language models** answer the question: How likely is a string of English words good English?
- Autocomplete on phones and websearch
- Creating English-looking documents
- Very common in machine translation systems
  - Help with reordering / style

$$p_{\text{lm}}(\text{the house is small}) > p_{\text{lm}}(\text{small the is house})$$

  - Help with word choice

$$p_{\text{lm}}(\text{I am going home}) > p_{\text{lm}}(\text{I am going house})$$

  - Use **conditional probabilities**

N-Gram Language Models

- Given: a string of English words $W = w_1, w_2, w_3, ..., w_n$
- Question: what is $p(W)$?
- Sparse data: Many good English sentences will not have been seen before

$\longrightarrow$ Decomposing $p(W)$ using the chain rule:

$$p(w_1, w_2, w_3, ..., w_n) = \\ p(w_1)\, p(w_2|w_1)\, p(w_3|w_1, w_2) \ldots p(w_n|w_1, w_2, ...w_{n-1})$$

(not much gained yet, $p(w_n|w_1, w_2, ...w_{n-1})$ is equally sparse)

Markov Chain

- **Markov independence assumption**:
  - only previous history matters
  - limited memory: only last k words are included in history (older words less relevant)
  - $\longrightarrow$ ***k*th order Markov model**
- For instance 2-gram language model:

$$p(w_1, w_2, w_3, ..., w_n) \simeq p(w_1)\, p(w_2|w_1)\, p(w_3|w_2) ... p(w_n|w_{n-1})$$

- What is conditioned on, here w_{i-1} is called the **history**
- How do we estimate these probabilities?

How do we estimate a probability?

- Suppose we want to estimate $P(w_n = \text{"}home\text{''} | h = \mathrm{go})$.

How do we estimate a probability?

- Suppose we want to estimate $P(w_n = \text{“}home\text{''} | h = \texttt{go})$.

home	**home**	big	with	to
big	with	to	and	money
and	**home**	big	and	**home**
money	**home**	and	big	to

How do we estimate a probability?

- Suppose we want to estimate $P(w_n = \text{``}home\text{''} | h = \texttt{go})$.

home	**home**	big	with	to
big	with	to	and	money
and	**home**	big	and	**home**
money	**home**	and	big	to

- Maximum likelihood (ML) estimate of the probability is:

$$\hat{\theta}_i = \frac{n_i}{\sum_k n_k} \tag{1}$$

Example: 3-Gram

- Counts for trigrams and estimated word probabilities

the red (total: 225)

word	c.	prob.
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

  - 225 trigrams in the Europarl corpus start with **the red**
  - 123 of them end with **cross**

$\longrightarrow$ maximum likelihood probability is $\frac{123}{225} = 0.547$.

Example: 3-Gram

- Counts for trigrams and estimated word probabilities

the red (total: 225)

word	c.	prob.
cross	123	0.547
tape	31	0.138
army	9	0.040
card	7	0.031
,	5	0.022

  - 225 trigrams in the Europarl corpus start with **the red**
  - 123 of them end with **cross**

  $\longrightarrow$ maximum likelihood probability is $\frac{123}{225} = 0.547$.

- Is this reasonable?

The problem with maximum likelihood estimates: Zeros

- If there were no occurrences of "bageling" in a history go, we'd get a zero estimate:

$$\hat{P}(\text{“bageling”} \mid \mathrm{go}) = \frac{T_{\mathrm{go},\text{“bageling”}}}{\sum_{w' \in V} T_{\mathrm{go},w'}} = 0$$

- $\rightarrow$ We will get $P(\mathrm{go}|d) = 0$ for any sentence that contains go bageling!
- Zero probabilities cannot be conditioned away.

Add-One Smoothing

- Equivalent to assuming a **uniform** prior over all possible distributions over the next word (you'll learn why later)
- But there are many more unseen n-grams than seen n-grams
- Example: Europarl 2-bigrams:
  - $86,700$ distinct words
  - $86,700^2 = 7,516,890,000$ possible bigrams
  - but only about $30,000,000$ words (and bigrams) in corpus