Statistical Language Models
Jurafsky & Martin, Ch 6, Appendix A.1, and additional material

CSE 597: Natural Language Processing

Outline

1. Formalization: random variables and the Markov assumption
2. Quantifying uncertainty
3. Estimating the parameters of a statistical LM
4. Jurafsky & Martin slides
    a. Berkeley Restaurant Corpus: Slides 17-21
    b. Evaluation & Perplexity; Visualization: Slides 28-37

Statistical Language Models

CSE 597: Natural Language Processing

Formalizing Statistical Language Models

Probabilistic Language Modeling

Goal: model the probability of a specific sequence of words:

$$P(W) = P(w_1, w_2, w_3, w_4, w_5, \ldots w_n)$$

Or: model the probability of a word, given previous words

$$P(w_5|w_1, w_2, w_3, w_4)$$

A **language model** computes either of these by assuming

- W is a random variable ranging over sequences of words in English
- Each $w_i \in W$ is a value of W, or an event of a word occurring

Random Variables

- Variables in probability theory are called **random variables**
  - Uppercase names for the variables, e.g., $P(A=true)$
  - Lowercase names for the values, e.g., $P(a)$ is an abbreviation for A=true
- A random variable is a function from a domain of possible worlds Ω (or sample space) to a range of values
  - Functions map values from the input domain to the output range
  - Again: a random variable is a function

Markov Assumption

- Russian statistician Andrei Markov
- Each state depends on a **fixed finite number** of prior states
- Future is **conditionally independent** of the past
- A **Markov chain** is a Bayesian network that incorporates time (temporal sequences of states)

Random Variables Indexed over Time

- Assume: fixed, constant, discrete time steps t
- Notation: $X_{a:b} = X_a, X_{a+1}, \ldots, X_{b-1}, X_b$
- Markov assumption: random variable X_t depends on bounded subset of $X_{0:t-1}$

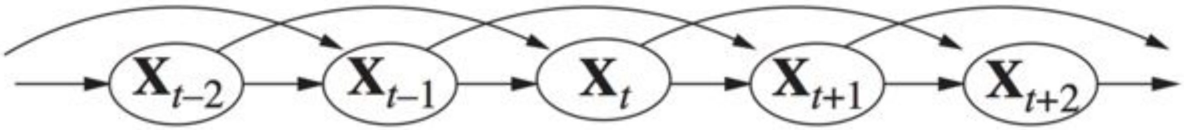
First-order Markov Process

$$P(X_0, \ldots X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2}, \ldots X_n)$$

- Bayesian network over time
  - Random variables . . . X_{t-2}, X_{t-1}, X_t, X_{t+1}, X_{t+2} . . .
  - Directed edges for conditional independence
- Each state X_t is conditioned on the preceding state X_{t-1}

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$$

Second Order Markov Process



$$P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-2}, X_{t-1})$$

- Each time step X_t is conditioned on the two preceding states X_{t-2}, X_{t-1}

Formalization of a Markov Chain (Statistical LM)

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

- Q is the random variable for words $\boldsymbol{w}$ at times $\boldsymbol{t}$
- A is the probability matrix of **conditional** probabilities $P(w_{t+1}|w_t)$
- π is the **prior** probabilities of words $\boldsymbol{w}$, i.e., $P(q_1) = \boldsymbol{w}$

Statistical Language Models

CSE 597: Natural Language Processing

Quantifying Uncertainty

Probabilities of Elementary Events

- The sample space Ω consists of an exhaustive set of mutually exclusive possibilities
  - Example: two words in a row,
- Every $\omega_i \in \Omega$ is assigned a probability (elementary event in the sample space of possible worlds): $P(\omega_i)$
  - $0 \leq P(\omega_i) \leq 1$
- Assuming Ω is finite ($w_1, \ldots, w_n$) we require
  - $P(\Omega) = \Sigma_{\omega\_i} P(\omega_i) = 1$
  - Because Ω is an exhaustive set of mutually exclusive possibilities

Prior versus Conditional Probabilities

- Prior probability: probability of an event from the sample space, with no conditioning evidence
  - ***P***(roll of 2 dice sums to 11) = $P((5, 6)) + P((6,5)) = 1/36 + 1/36 = 1/18$
  - ***P***(w_1, w_2, w_3, w_4, w_5) = ***P***("Students like to try to")
- Conditional (or posterior) probability of an event conditioned on the occurrence of an earlier event
  - $P(Die_2=6|Die_1=5) = 1/6$

Product Rule of Conditional Probabilities

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

$$P(Die_2 = 6|Die_1 = 5) = \frac{P(Die_2 = 6 \wedge Die_1 = 5)}{P(Die_1 = 5)}$$

$$P(a \wedge b) = P(a|b)P(b)$$

Independence

- Random variables X and Y are independent iff:

$$P(X,Y) = P(X)P(Y)$$
$$P(X|Y) = P(X)$$
$$P(Y|X) = P(Y)$$

- Taking any independence into account is essential for efficient probabilistic reasoning
- Unfortunately, complete independence is rare
- Fortunately, assuming conditional independence works well in practice

Conditional Independence

- Random variables X and Y are **conditionally** independent given Z iff

$$P(X|Y,Z) = P(X|Z)$$
$$P(Y|X,Z) = P(Y|Z)$$
$$P(X \wedge Y|Z) = P(X|Z)P(Y|Z)$$

Chain Rule of Probabilities

- Generalizes the product rule:

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

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

- To any number of variables in the joint probability distribution

$$P(A,B,C,D) = P(A)\ P(B|A)\ P(C|A,B)\ P(D|A,B,C)$$

$$P(X_1, X_2, \ldots, X_n) = P(X_1)\ P(X_2|X_1)\ P(X_3|X_1, X_2) \ldots\ P(X_n|X_1, X_2, \ldots X_{n-1})$$

Chain Rule Applied to Language Modeling

$$P(w_1, w_2, \ldots, w_n) = P(w_1) \prod_i P(w_i | w_{i-1}, w_{i-2}, \ldots, w_{i-1})$$

- P("Students like to try to") = P(Students,like, to, try,to)

  = P(Students) P(like|Students) P(to|Students, like) . . .

Markov Rule Applied to Chain Rule for LM

$$P(w_1, w_2, \ldots, w_n) = P(w_1) \prod_i P(w_i | w_{i-1}, w_{i-2}, \ldots, w_{i-1})$$

- Can be approximated by a tri-gram language model

$$P(w_1, w_2, \ldots, w_n) = \prod_i P(w_i | w_{i-1}, w_{i-2})$$

- Or a bi-gram language model. Why not a **unigram** model?

Connecting the Formalization to Probability

- Q represents the length n word sequences
- A represents the probabilities $P(w_i|w_{i-1})$
  - For a bigram markov chain LM
  - What would be needed for a trigram LM?
- π represents the probabilities $P(w_1)$

$$Q = q_1 q_2 \ldots q_N$$

$$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$$

$$\pi = \pi_1, \pi_2, \ldots, \pi_N$$

Language Has Long Distance Dependencies

- Number agreement between grammatical subject and verb, for example:

The **computers** which I just bought for the machine room on the 5th floor **have** crashed.
The **computer** which I just bought for the machine room on the 5th floor **has** crashed.

- Statistical language modeling cannot handle LDDs
- A statistical LM still works well enough: It is easier to get good estimates for a simpler wrong model (fewer parameters, e.g., bigram probabilities) than a more complicated more correct model

Statistical Language Models

CSE 597: Natural Language Processing

Estimating the Parameters of a Bigram Statistical LM

Building a Statistical Language Model

- Collect a large corpus of text (e.g., webscale) $\boldsymbol{C}$
- All the observed word sequences are the data for the two parameters of the model: $\boldsymbol{A}$ and $\boldsymbol{\pi}$

Building a Statistical Language Model

1. Create V(ocabulary) from $\boldsymbol{C}$
   a. a list of the unique words in the corpus
   b. Add a <s> (start) and </> (end) tokens to every sentence, and add <s> and </s> to V
2. $\boldsymbol{\pi}$ applies only to <s>: $P(\text{<s>}_1) = 1$
3. Unigram frequencies: For every $\boldsymbol{v}_i$ in V, compute count($\boldsymbol{v}_i$)
4. For every sequence of two words $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, compute count($\boldsymbol{v}_i$, $\boldsymbol{v}_j$)
5. $P(\boldsymbol{v}_j \mid \boldsymbol{v}_i) = \text{count}(\boldsymbol{v}_i, \boldsymbol{v}_j) / \text{count}(\boldsymbol{v}_i)$

Example

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

Unigram counts
<s>(3), </s>(3), am(2), and(1), do(1), eggs(1), green(1), ham(1), I(3), like(1), not(1), Sam(2)

Bigrams > once
<s>,I(2), I,am(2),

P(I|<s>)	2/3
P(Sam|<s>)	1/3
P(am|I)	2/3
P(do|I)	1/3
P(Sam|am)	1/2
P(</s>|am)	1/2
P(I|Sam)	1/2