CS 181: Natural Language Processing

Lecture 5: Probability & N-Grams

Kim Bruce
Pomona College
Spring 2008

Disclaimer: Slide contents borrowed from many sources on web!

Probability

- Run experiments (trials)
- Observe set of all possible outcomes
  Sample space, Ω:
  - 3 coin flips: {TTT, TTH, THT, THH, HTT, HTH, HHH, HHT}
  - Part of speech of word: dogs: {N Pl, V 3Sg}
- Compute probability of basic events, use to compute probability of actual events of interest

Events

- Event, A, is set of basic outcomes - *subset of sample space, Ω*
  - At least 2 heads: A = {HHH, HHT, HTH, THH}
  - dogs is noun: A = {N Pl}
- A = Ω is certain event,
  A = ∅ is impossible event
- Notation: $\overline{A}$ = Ω - A
- Event space, F, is $\mathcal{P}(\Omega)$, collection of all subsets of sample space, Ω.

Probability

- Probability function, P, assigns probability mass to events in event space, F, where
  - P: F → [0,1]
  - P(Ω) = 1
  - Countable additivity: For disjoint events A_j in F, $P(\cup_j A_j) = \cup_j P(A_j)$
  - Consequences: $P(\overline{A}) = 1 - P(A)$,
    $\Sigma_{a\in\Omega} P(\{a\}) = 1$,
    $P(\emptyset) = 0$,
    $A \subseteq B$ implies $P(A) \le P(B)$

Estimating Probability

- Repeat experiment many times, say N.
- Count number of basic outcomes that are members of A, say C.
- As N increases, C/N should approach a constant value, best estimate for P(A).
- E.g., Coin is tossed 3 times, what is probability of getting at least 2 heads.
  - Try it 1000 times, record when at least two heads, say C times.
  - Estimate P(at least 2 heads) = C/1000

Using Distributions

- If "fair" coin, then probability of head should be .5
- Uniform distribution:
  - Each basic outcome is equally likely
  - P(HHH) = P(HHT) = ... = P(TTT)
  - Thus, P(at least two heads) = 4/8 = .5

Joint & Conditional Probabilities

- The joint probability of A and B both happening, $P(A \cap B)$, is also written $P(A,B)$.
- The conditional probability of A, given B, is $P(A|B) = P(A,B) / P(B)$.
  - Hence $P(A,B) = P(A|B) * P(B)$
- Bayes rule: $P(A|B)*P(B) = P(B|A)*P(A)$
  - Hence $P(A|B) = (P(B|A)*P(A)) / P(B)$
  - Can calculate one conditional if know other.

Independence

- Can we compute $P(A,B)$ from $P(A)$ & $P(B)$?
- $P(A,B) = P(B|A)*P(A)$
- Now, $P(B|A) = P(B)$ iff probability of B is unaffected by whether or not A is true.
- Def: Two events A and B are independent iff $P(A,B) = P(A)*P(B)$, and otherwise dependent.

Independence

- If events are independent, then need much less data to be saved,
- Though we'll leverage info on dependency to disambiguate data.

Chain Rule

- Let $A_i^j = A_i, A_{i+1}, \ldots, A_j$
- $P(A_1, A_2, A_3, \ldots, A_n) = P(A_1^n) =$
  $= P(A_n \mid A_1^{n-1})*P(A_1^{n-1})$
  $= P(A_n \mid A_1^{n-1})*P(A_{n-1} \mid A_1^{n-2})*P(A_1^{n-2})$
  ...
  $= P(A_n \mid A_1^{n-1})*P(A_{n-1} \mid A_1^{n-2})*\ldots*P(A_1)$
- *Simplifies if all independent!*

Using Chain Rule

- What is likelihood of 3 heads in 3 tosses:
  - Based on counts: 1 / 8
  - Chain rule:
    - $P(H_1H_2H_3) = P(H_3|H_1H_2)*P(H_2|H_1)*P(H_1)$
      $= P(H_3)*P(H_2)*P(H_1)$
      $= 1/2 * 1/2 * 1/2 = 1/8$

Bayesian Decision Theory

- Can choose which model best:
- $P(model_1|data) = \frac{P(data|model_1) \times P(model_1)}{P(data)}$
- $P(model_2|data) = \frac{P(data|model_2) \times P(model_2)}{P(data)}$
- Usually ignore denominator in comparisons

Using Bayes ...

- Ex: P(French | glacier, melange) vs
  P(English | glacier, melange)
- Ex: Test authorship or identity of text
  - P(Hamlet | "hand", "death")
  - P(Oliver | "hand", "death")

Chain Rule problems

- $P(A_1^n) = P(A_n \mid A_1^{n-1}) * P(A_{n-1} \mid A_1^{n-2}) * \ldots * P(A_1)$
- *History-based model*
- Calculating last few based on training data fine, but eventually get little or no data:
  - $P(A_1) = 2536/158796$, $P(A_2 \mid A_1) = 128/2536$, ... $P(A_4 \mid A_1, A_2, A_3) = 0/8$
  - Results in $P(A_1^n)=0$, yet not likely

Problems w/Training Data

- Estimate each of the probabilities from training data, but get unique sequences!
- Some words - let alone whole phrases - may not even appear in training data.
- Give up accuracy
  - Rather than computing probability of a word given its entire history, approximate the history by a limited number, n, of preceding words.
  - Called nth-order Markov assumption

Estimating Probabilities

- Trigram model results in simplification:
  - $P(A_1^n) = P(A_n \mid A_{n-2}, A_{n-1}) *$
    $P(A_{n-1} \mid A_{n-3}, A_{n-2}) * \ldots * P(A_2 \mid A_1) * P(A_1)$
  - *Can get much better estimates from data!*
- Use maximum likelihood estimation (MLE)
  - $P(C|A,B) = \frac{P(A,B,C)}{\sum_d P(A,B,d)}$
  - Approx for seqs: $P(w_3|w_1 w_2) \approx \frac{C(w_1 w_2 w_3)}{C(w_1 w_2)}$

Examples

- P("horses") = P("h")*P("o"| "h")
  *P("r"| "ho")*P("s"| "or")
  * ... *P("s"| "se")
- If want word "horses" then often add "<s>" at beginning and </s> at end.

N-gram Models

- Used in speech recognition, OCR, context-sensitive spelling correction.
- Appallingly simple from linguistic POV
- Relations can be arbitrarily distant
  - *The **man** on the sidewalk, without pausing to look at what was happening down the street, and quite oblivious to the situation that was about to befall him, confidently **strode** into the center of the road.*
- But not usually ...

N-gram Models

- Collins (1997): if treat noun phrases as a unit, 74% of dependencies in WSJ part of Penn Treebank are with an adjacent word.
- 95% with word less than 5 words away

Evaluating

- Best is to test in application
- Predict using "perplexity" on training data
- $PP(W) = P(w_1 \ldots w_n)^{-\frac{1}{n}}$

$$= \sqrt[n]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-2}w_{i-1})}}$$

- Smaller perplexity -- more predictable

Perplexity

- Strings of digits:
  - all 0's - perplexity = 1
  - {0,1} same freq, perplexity = 2
  - {0,...,9} all same freq, perplexity = 10
  - 0 occurs 10x more often, perplexity = 5.5
  - WSJ words, preplexity = 109
- Perplexity related to information theoretic entropy

Using N-grams for classification

- Separate documents into training and testing
- Tokenize into words
- Count occurrences of each word in each document.
- Estimate P(w|c) by ratio of counts
- For each test document

Classifying Documents

- Given some text, estimate which class it came from.
  - E.g., P(Hamlet | "ghost", "walks")
- Use Bayesian:
  - P(Hamlet | "ghost", "walks") = P("ghost", "walks" | Hamlet) * P(Hamlet)

More Problems

- Sparsity of data
  - Even common words don't occur very often
    - In a million words:
      - "kick" occurs about 10 times
      - "wrist" occurs about 5 times
      - Even common 3 word phrases are unlikely to appear!
    - How to cope with missing data?

It's bad!

count	2-grams	3-grams
1	8,045,024	53,737,350
2	2,065,469	9,229,958
3	970,434	3,654,791
> 4	3,413,290	8,728,789
> 0	14,494,217	75,349,888
possible	6.8×10^{10}	1.7×10^{16}

Taken from data set w/ 261,741 words
365,000,000 words training!

Too Many Zeroes

- 6.8×10^{10} possible bigrams,
  but only 3.65×10^{8} words in training set.
- Trigrams worse!
- Can't get data set large enough to get them all -- even those that could occur.
- Solution:
  - Redistribute probability to *save* some for those that haven't been encountered.

Any Questions?