CSCI 3907/6907

Fall 2019

Lecture 3

Probability Review and Language Modeling

Part 1: Probability Review

Why Probability?

- A mathematical framework that allows us describe and analyze random phenomena in the world around us

- In NLP, can model what is intuitive for human
  - Draw inferences based on examining only a part of the whole
  - Make decisions

https://www.probabilitycourse.com/

Probability

- Flip a fair coin that have equal chance of landing head or tail. **What is the probability of getting a head?**

  – What are the possible outcomes?

  H (head) or T (tail) → 2 equally likely outcomes

  – Of those, how many different possibilities that meet my constraints?

  Only one –> H (head)

  Answer → **½**

Probability

- Roll a die that have equal chances to turn up as {1,2,3,4,5,6}. **What is the probability of getting 1?**

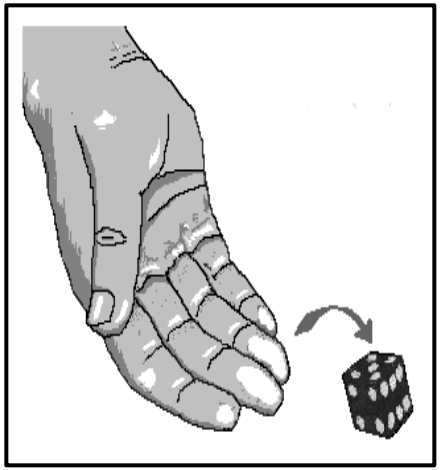
  – What are the possible outcomes?

  1,2,3,4,5,6 → 6 equally likely outcomes

  – Of those, how many different possibilities that meet my constraints?

  Only one (1)

  Answer → 1/6



Probabilistic Model

To define a probabilistic model, one must define:

What are the sample space for our previous examples?

1. A sample space, Ω: the set of all possible outcomes of the experiment.

2. A probability law. A probability law assigns a probability, P(A), to every event, A. $0 <= P(A) <= 1$

   – Must satisfy the Axioms of Probability.

   An event → a collection of outcomes
   (e.g. all even numbers when rolling a die **or** all heads)

Probabilistic Axioms

Probability Axioms

1. **(Nonnegativity)** $\mathbf{P}(A) \geq 0$, for every event A.
2. **(Additivity)** If A and B are two disjoint events, then the probability of their union satisfies

$$\mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B).$$

   Furthermore, if the sample space has an infinite number of elements and $A_1, A_2, \ldots$ is a sequence of disjoint events, then the probability of their union satisfies

$$\mathbf{P}(A_1 \cup A_2 \cup \cdots) = \mathbf{P}(A_1) + \mathbf{P}(A_2) + \cdots$$

3. **(Normalization)** The probability of the entire sample space Ω is equal to 1, that is, $\mathbf{P}(\Omega) = 1$.

Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Probabilistic Axioms – Example

Flipping a coin twice: →

Sample Space: {HH,HT,TH,TT} equally likely events 1/4

- **Nonnegativity** → all are larger than zeros
- ***Normalization*** → P(HH)+P(HT)+P(TH)+P(TT)=1
- ***Additivity*** → P{at least 1 head} = P{exactly 1 head}+P{exactly 2 heads} =P{TH,HT}+P{HH}=2/4+1/4=3/4

Sample Spaces - Types

- Discrete → add all the corresponding outcomes (countable space)

- Continuous → use integration instead of summation (uncountable space)
  - Example: $\Omega = [0,1]$, where all outcomes are equally likely.

Discrete Models

The probability of an event is the sum of the probabilities of its elements:

$$\mathbf{P}\big(\{s_1, s_2, \ldots, s_n\}\big) = \mathbf{P}\big(\{s_1\}\big) + \mathbf{P}\big(\{s_2\}\big) + \cdots + \mathbf{P}\big(\{s_n\}\big).$$

Discrete Uniform Probability Law

If the sample space consists of n possible outcomes which are equally likely (i.e., all single-element events have the same probability), then the probability of any event A is given by

$$\mathbf{P}(A) = \frac{\text{Number of elements of } A}{n}.$$

Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Discrete Models - Example

- Rolling a dice once

$$\Omega = \{1,2,3,4,5,6\}$$

  - P(1 or 6) = P(1) + P(6) = 2/6
  - P(1 and 3) = 0 → not possible in this trial
  - P(even number) =P(2)+P(4)+P(6)=3/6

Conditional Probability

The probability of an event given some partial information

Examples:

- In an experiment involving two successive rolls of a die, you are told that the sum of the two rolls is 9. How likely is it that the first roll was a 6?
  P(the first roll was a 6 | the sum of the two rolls is 9)

- How likely is it that a person has a disease given that a medical test was negative? *P(a person has a disease| a medical test was negative)*

Conditional Probability - Properties

- If A and B are two events in a sample space S, then the conditional probability of A given B is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \text{ when } P(B) > 0.$$

- In the case where all outcomes are finitely many and equally likely, we have

$$\mathbf{P}(A \mid B) = \frac{\text{number of elements of } A \cap B}{\text{number of elements of } B}.$$

https://www.probabilitycourse.com/

Example 1

- Roll a die with equally likely outcomes {1,2,3,4,5,6}.

  A={1,3,5} → the event that the outcome is an odd number
  B={1,2,3} → the event that the outcome is less than or equal to 3

  What is P(A)? P(A|B)?

  P(A)=3/6

  P(A|B)=elements in both A and B/elements in B

  P(A|B)=2/3

Example 2

- If there's an aircraft in a certain region, the radar system generates an alarm with probability 0.99
- If there's no aircraft in the region, the radar system generates an alarm with probability 0.1
- The probability of an aircraft being in the region is 0.05

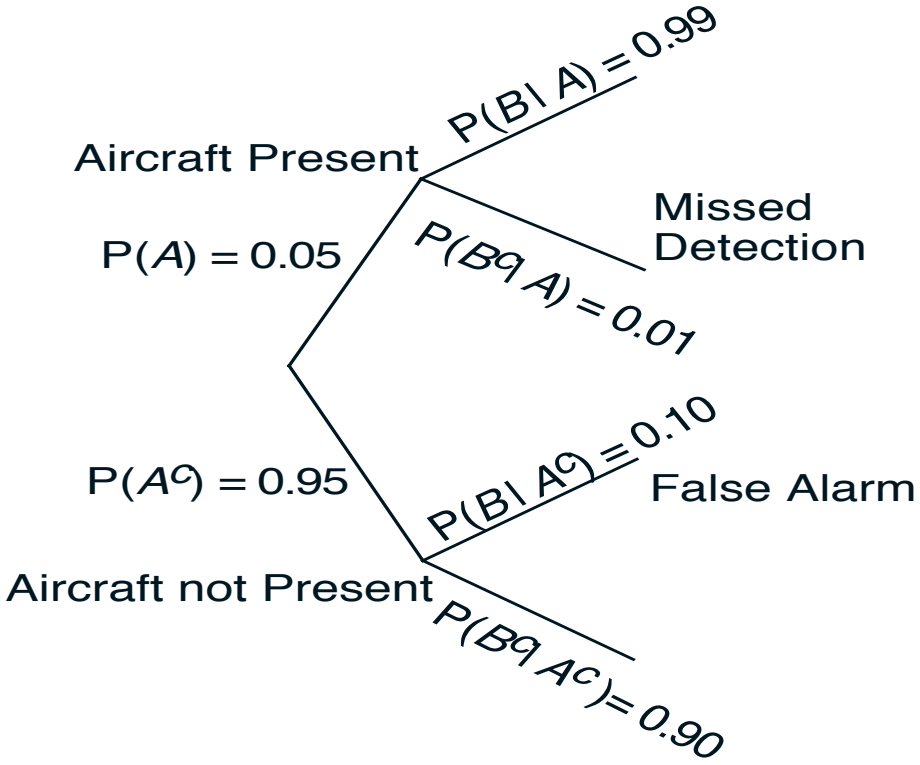
What is the probability that an aircraft is in the region, and the radar system does not generate an alarm?

A → aircraft present (in the region)

B → the radar system generates an alarm

P(radar doesn't generate alarm| aircraft present) = 0.01

P(aircraft present and then radar does not generate an alarm) = ?



Introduction to Probability. Bertsekas & Tsitsiklis, 2008

The Sequential Method ***OR*** The Chain Rule for Conditional Probability

a. We set up the tree so that an event of interest is associated with a leaf. We view the occurrence of the event as a sequence of steps, namely, the traversals of the branches along the path from the root to the leaf.

b. We record the conditional probabilities associated with the branches of the tree.

c. We obtain the probability of a leaf by multiplying the probabilities recorded along the corresponding path of the tree.

Multiplication Rule

Assuming that all of the conditioning events have positive probability, we have

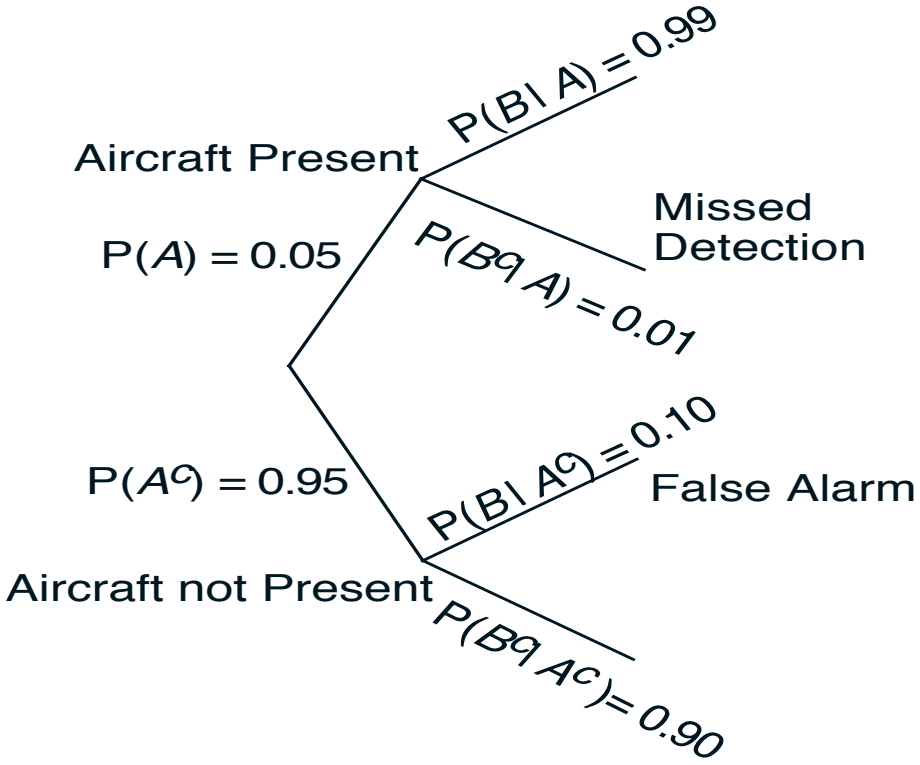
$$\mathbf{P}\left(\cap_{i=1}^{n} A_i\right) = \mathbf{P}(A_1)\mathbf{P}(A_2 \mid A_1)\mathbf{P}(A_3 \mid A_1 \cap A_2)\cdots\mathbf{P}\left(A_n \mid \cap_{i=1}^{n-1} A_i\right).$$

Introduction to Probability. Bertsekas & Tsitsiklis, 2008

A → aircraft present (in the region)

B → the radar system generates an alarm

P(aircraft present and then radar does not generate an alarm) =
0.05*0.01=0.0005



Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Independence

A → event that it rains tomorrow $P(A)=1/3$

B → event that a fair coin lands heads up $P(B)=1/2$

What is P(A|B)?

Does not carry information about each other

$$P(A|B)=P(A)=1/3$$

Independence

Independence

- Two events A and B are said to independent if

$$\mathbf{P}(A \cap B) = \mathbf{P}(A)\mathbf{P}(B).$$

  If in addition, $\mathbf{P}(B) > 0$, independence is equivalent to the condition

$$\mathbf{P}(A \mid B) = \mathbf{P}(A).$$

- If A and B are independent, so are A and B^c.
- Two events A and B are said to be conditionally independent, given another event C with $\mathbf{P}(C) > 0$, if

$$\mathbf{P}(A \cap B \mid C) = \mathbf{P}(A \mid C)\mathbf{P}(B \mid C).$$

  If in addition, $\mathbf{P}(B \cap C) > 0$, conditional independence is equivalent to the condition

$$\mathbf{P}(A \mid B \cap C) = \mathbf{P}(A \mid C).$$

- Independence does not imply conditional independence, and vice versa.

Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Independence - Example

- Flip a coin repeatedly until you observe the first tails at which point you stop. X → the total number of coin tosses

What is P(X=5)

Equivalent to P(HHHHT)

P(HHHHT) = P(H)P(H)P(H)P(H)P(T) = ½*½*½*½*½

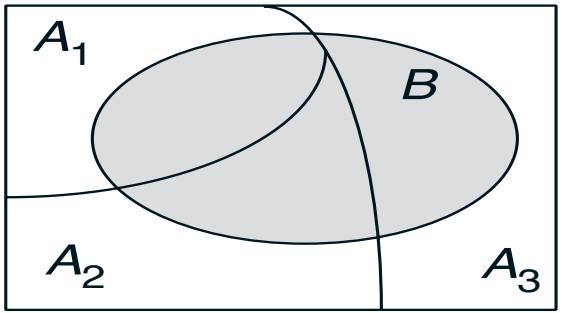
https://www.probabilitycourse.com/

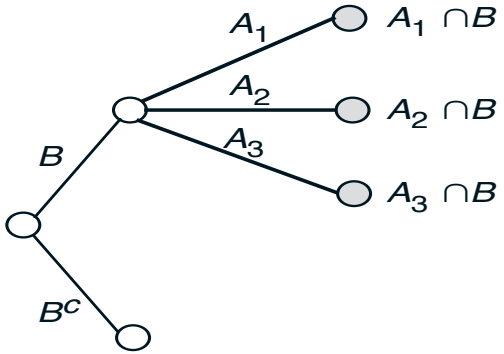
Total Probability Theorem

Total Probability Theorem

Let $A_1, \ldots, A_n$ be disjoint events that form a partition of the sample space (each possible outcome is included in one and only one of the events $A_1, \ldots, A_n$) and assume that $\mathbf{P}(A_i) > 0$, for all $i = 1, \ldots, n$. Then, for any event B, we have

$$\begin{aligned}\mathbf{P}(B) &= \mathbf{P}(A_1 \cap B) + \cdots + \mathbf{P}(A_n \cap B) \\ &= \mathbf{P}(A_1)\mathbf{P}(B \mid A_1) + \cdots + \mathbf{P}(A_n)\mathbf{P}(B \mid A_n).\end{aligned}$$





Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Example

We roll a fair four-sided die. If the result is 1 or 2, we roll once more but otherwise, we stop.

What is the probability that the sum total of our rolls is at least 4?

A → the sum total of our rolls is at least 4 (constraint)

1. 4 then stop	P(A)=1/4
2. 3 then stop	P(A)=0
3. 2 then roll again 1,2,3,4	P(A)=1/4*3/4=3/16
4. 1 then roll again 1,2,3,4	P(A)=1/4*2/4=2/16=1/8

P(the sum total of our rolls is at least 4)=1/4+3/16+1/8=9/16

Bayes' Rule

Bayes' Rule

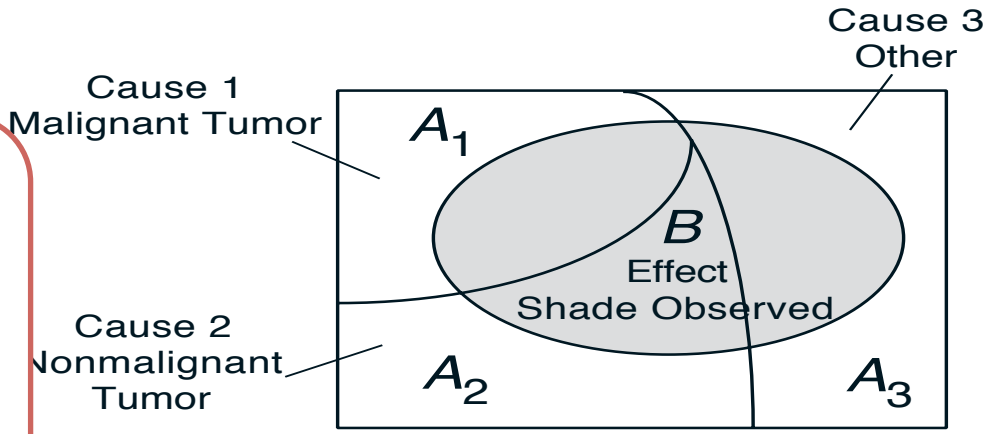
Let $A_1, A_2, \ldots, A_n$ be disjoint events that form a partition of the sample space, and assume that $\mathbf{P}(A_i) > 0$, for all i. Then, for any event B such that $\mathbf{P}(B) > 0$, we have

$$\begin{aligned}\mathbf{P}(A_i \,|\, B) &= \frac{\mathbf{P}(A_i)\mathbf{P}(B \,|\, A_i)}{\mathbf{P}(B)} \\ &= \frac{\mathbf{P}(A_i)\mathbf{P}(B \,|\, A_i)}{\mathbf{P}(A_1)\mathbf{P}(B \,|\, A_1) + \cdots + \mathbf{P}(A_n)\mathbf{P}(B \,|\, A_n)}.\end{aligned}$$

Example

- A shade is observed in a person's X-ray.
- We know the probabilities $P(A_i)$ and $P(B|A_i)$, i =1,2,3
- What is the likelihood that the shade is caused by a malignant tumor?

So we know P(Malignant Tumor), P(shade|malignant Tumor) but we want P(malignant tumor|shade)



Introduction to Probability. Bertsekas & Tsitsiklis, 2008

Example

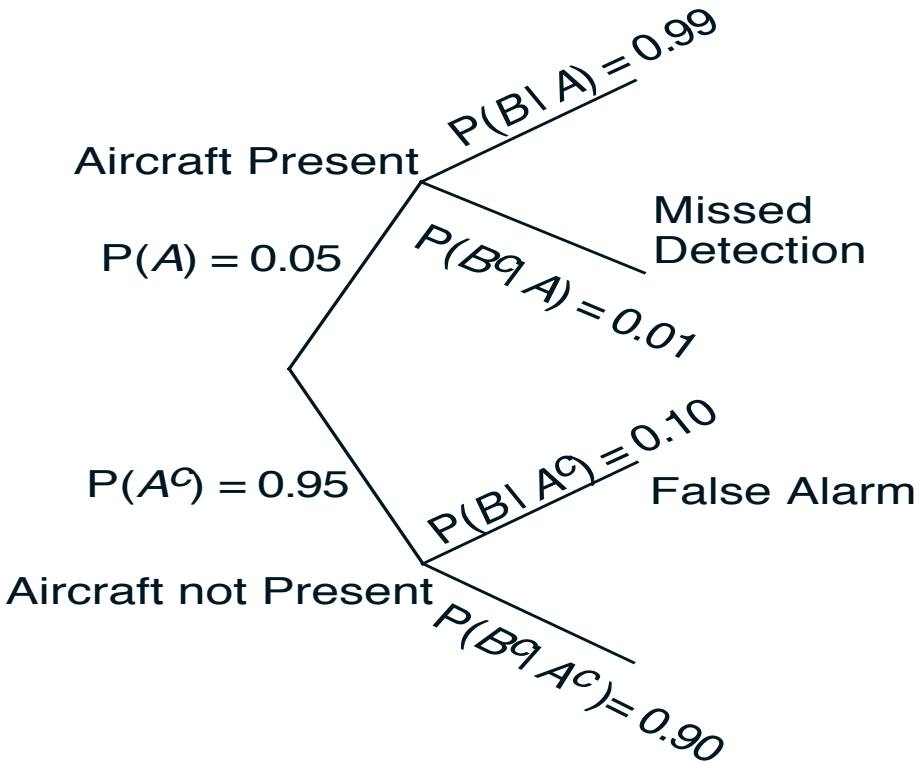
- If there's an aircraft in a certain region, the radar system generates an alarm with probability 0.99.
- If there's no aircraft in the region, the radar system generates an alarm with probability 0.1.
- The probability of an aircraft being in the region is 0.05.

What is the probability that an aircraft is in the region, Given that the radar system generated an alarm?

A → aircraft present (in the region)

B → the radar system generates an alarm

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



P(aircraft present | radar generated an alarm)=
P(A|B)=(0.99*0.05)/(0.99*0.05+0.10*0.95)

Counting

The Counting Principle

Consider a process that consists of r stages. Suppose that:

(a) There are n_1 possible results for the first stage.

(b) For every possible result of the first stage, there are n_2 possible results at the second stage.

(c) More generally, for all possible results of the first $i-1$ stages, there are n_i possible results at the ith stage.

Then, the total number of possible results of the r-stage process is

$$n_1 \cdot n_2 \cdots n_r.$$

Example

A telephone number is a 7-digit sequence, but the first digit has to be different from 0 or 1. How many distinct telephone numbers are there?

8*10*10*10*10*10*10=8,000,000

Random Variables

A random variable (r.v.) is a real-–valued function of the outcome of an experiment.

A random variable is discrete if the set of values it can take is finite, or countably infinite.

Examples:

A) If the experiment is rolling a die twice:

- The sum of the two rolls
- The number of 3's rolled
- (Value of the second roll)^3

B) If the experiment is transmitting a message:

- The number of bits that are transmitted incorrectly
- The time the message takes to be transmitted on the network

Probability Mass Function (PMF)

- Each r.v., X, has an associated PMF, defined as follows, for each value x that X can take:

$$p_X(x) = P(\{X = x\})$$

- **To compute the PMF of a random variable X:**

  For each possible value x of X:

  Collect all the possible outcomes in the event {X = x}
  Sum their probabilities to obtain $p_X(x)$

Example

Experiment: two independent 4-sided die rolls

Random Variable (X): Maximum value

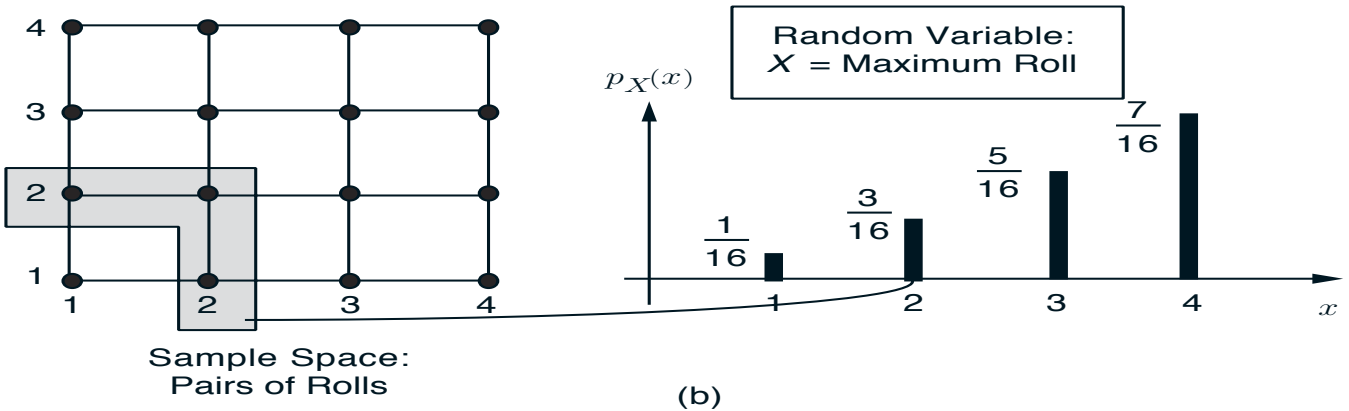
Ω= { (1,1), (1,2), (2,1), (1,3), (3,1), (1,4), (4,1), (2,2), (2,3), (3,2), (2,4), (4,2), (3,3), (3,4), (4,3), (4,4)}

X=1 → ***(1,1) (1,2) (2,1)***

X=2 → ***(1,2) (2,1) (2,2) (2,3) (3,2) (2,4) (4,2)***

X=3 → ***(1,3) (3,1) (2,3) (3,2) (3,3) (3,4) (4,3)***

X=4 → ***(1,4) (4,1) (2,4) (4,2) (3,4) (4,3) (4,4)***



Online Resources

- Introduction to Probability by Dimitri P. Bertsekas and John N. Tsitsiklis (2nd edition)

- https://www.probabilitycourse.com/

- https://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/amsbook.mac.pdf

Part 2: Language Model

Handwriting Recognition

- Assume a note is given to a bank teller, which the teller reads as
  "I have a gub." (cf. Woody Allen)

What is actually meant by **gub**?

  - gun, gum, Gus, and gull are words
  - gun has a higher probability in the context of a bank

Real Word Spelling Errors

- They are leaving in about fifteen minuets to go to her house.
- The study was conducted mainly be John Black.
- I need to notified the bank.
- He is trying to fine out.

How did you figure out what was actually meant?

Probabilistic Language Models

- A Language model (LM) assigns probabilities for next letters or words
  - and by extension, probabilities for entire sentences.

- Goals:
  - compute the probability of a sentence or sequence of words
    $P(W) = P(w_1,w_2,w_3,w_4,w_5...w_n)$

  - Or the of probability of an upcoming word *(related task)*
    $P(w_5|w_1,w_2,w_3,w_4)$

- A model that computes either of these:
  $P(W)$ or $P(w_n|w_1,w_2...w_{n-1})$ is called a **language model**.

Applications of Language Models

- Machine Translation: P(**high** winds tonight) > P(**large** winds tonight)
- Spell Correction: *The office is about fifteen **minuets** from my house*
  P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
- Speech Recognition: P(I saw a van) > P(eyes awe of an)
- + summarization, question-answering … etc

How to compute P(W)

- How to compute this joint probability:
  - P(its, water, is, so, transparent, that)

- Intuition: let's rely on the Chain Rule of Probability

$$P(x_1,x_2,x_3,\ldots,x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1,\ldots,x_{n-1})$$

Example

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

P("its water is so transparent") =

P(its) × P(water|its) × P(is|its water)

× P(so|its water is) × P(transparent|its water is so)

How to estimate these probabilities

- Could we just count and divide?

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

  - Too many possible sentences!
  - We'll never see enough data for estimating these.

Markov Assumption

Andrei Markov

- Simplifying assumption:

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

- Or maybe

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

Markov Assumption

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

- In other words, we approximate each component in the product

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

Simplest case: Unigram model

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

- Also called 1-gram
- Words in the sentence are independent of each other

P(“its water is so transparent”) =

P(its) × P(water) × P(is) × P(so) × P(transparent)

Bigram Model (2-gram)

- Condition on the previous word:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

- Each word is conditionally independent of past context given previous word.

P("its water is so transparent") =

P(its) × P(water| its) × P(is|water)
× P(so|is) × P(transparent|so)

N-gram models

- We can extend to trigrams (3-grams), 4-grams, 5-grams
- In general this is an insufficient model of language
  - because language has **long-distance dependencies**:

"The **computer**(s) which I had just put into the machine room on the fifth floor is (are) **crashing**."

- But we can often get away with N-gram models

Estimating N-Gram Probabilities

How Do We Estimate Probabilities

- The **Maximum Likelihood Estimate (MLE)**

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Example

Beginning and end of sentence markers added to properly estimate bigram probabilities

I am Sam
Sam I am
I do not like green eggs and ham

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

An example

Compute the following:

P(I|<s>)
P(Sam|am)

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

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

$$P(\texttt{I} \mid \texttt{<s>}) = \tfrac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \tfrac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \tfrac{2}{3} = .67$$
$$P(\texttt{</s>} \mid \texttt{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \tfrac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \tfrac{1}{3} = .33$$

Berkeley Restaurant Project

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Raw bigram counts

- Out of 9222 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Raw Bigram Probabilities

Unigram Counts:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Normalized Bigram counts:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Example

P(i|<s>)=0.25
P(food|english)=0.5

P(english|want)=0.0011
P(</s>|food=0.68

P(<s> I want english food </s>) =
P(i|<s>)
× P(want|I)
× P(english|want)
× P(food|english)
× P(</s>|food)
= .000031

What Kinds of Linguistic Knowledge?

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

Practical Issues

- More common to use trigrams – sometimes larger if there is sufficient training data
- We do everything in log space
  - Avoid underflow
  - (also adding is faster than multiplying)

$$\log(p_1 \times p_2 \times p_3 \times p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Evaluation

Evaluation: How good is our model?

- Does our language model prefer good sentences to bad ones?
  - Assign higher probability to “real” or “frequently observed” sentences
    - Than “ungrammatical” or “rarely observed” sentences?

- Types of Evaluation:
  - **Extrinsic Evaluation** → integrate in any application and measure how much the application improves
  - **Intrinsic Evaluation** → the use of an evaluation metric that measures the quality of a model independent of any application

Evaluation: How good is our model?

- We train parameters of our model on a **training set**.

- We test the model's performance on data we haven't seen.
  - A **test set** is an unseen dataset that is different from our training set, totally unused.
  - An **evaluation metric** tells us how well our model does on the test set.

Training on the test set

- We can't allow test sentences into the training set
- We will assign it an artificially high probability when we set it in the test set
- "Training on the test set"
  - Bad science!
  - And violates the honor code

Extrinsic Evaluation of N-gram Models

- Best evaluation for comparing models A and B
  - Put each model in a task
    - spelling corrector, speech recognizer, MT system
  - Run the task, get an accuracy for A and for B
    - How many misspelled words corrected properly
    - How many words translated correctly
  - Compare accuracy for A and B

Difficulty of Extrinsic Evaluation

- Extrinsic evaluation
  - Time-consuming; can take days or weeks

- So
  - Sometimes use **intrinsic** evaluation: **perplexity**
  - Bad approximation
    - unless the test data looks **just** like the training data
    - So **generally only useful in pilot experiments**
  - But is helpful to think about.

Evaluation – Perplexity

Intuition of Perplexity

- The Shannon Game:
  - How well can we predict the next word?

    I always order pizza with cheese and ____

    The 33rd President of the US was ____

    I saw a ____

  - Unigrams are terrible at this game. **(Why?)**

mushrooms 0.1

pepperoni 0.1

anchovies 0.01

….

fried rice 0.0001

….

and 1e-100

- A better model of a text
  - is one which assigns a higher probability to the word that actually occurs

Perplexity

Perplexity is the inverse probability of the test set, normalized by the number of words:

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Applying the Chain rule:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

Minimizing perplexity is the same as maximizing probability

Perplexity as branching factor

- Let's suppose a sentence consisting of random digits

- What is the perplexity of this sentence according to a model that assign P=1/10 to each digit?

- If zero is 10 times more frequent than other numbers, the weighted branching factor is smaller

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \left(\frac{1}{10}^{N}\right)^{-\frac{1}{N}} \\
&= \frac{1}{10}^{-1} \\
&= 10
\end{aligned}
$$

Lower perplexity = better model

- Wall Street Journal:
  - Training: 38 million words
  - Test: 1.5 million words

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

Text Generation

- Sample words according to their bigram probabilities:

```
<s> I
    I want
      want to
           to eat
              eat Chinese
                  Chinese food
                          food  </s>
I want to eat Chinese food
```

Approximating Shakespeare

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

Shakespeare as corpus

- N=884,647 tokens, V=29,066

- Shakespeare produced 300,000 bigram types out of V^2= 844 million possible bigrams.
  - So 99.96% of the possible bigrams were never seen (have zero entries in the table)

- Quadrigrams worse: What's coming out looks like Shakespeare because it ***is*** Shakespeare

The Wall Street Journal - Corpus

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

The Perils of Overfitting

- N-grams only work well for word prediction if the test corpus looks like the training corpus

  - In real life, it often doesn't
  - We need to train robust models that generalize.
  - One kind of generalization: Zeros!
    - Things that don't ever occur in the training set
      - But occur in the test set

Zero Counts (Sparseness)

- **Training set:**

  … denied the allegations
  … denied the reports
  … denied the claims
  … denied the request

  P(“offer” | denied the) = 0

- **Test set**

  … denied the offer
  … denied the loan

The Problem of Sparsness

- Bigrams with zero probability
  - mean that we will assign 0 probability to the test set!

- And hence we cannot compute perplexity (can't divide by 0)!

Smoothing (discounting)

Smoothing (Discounting)

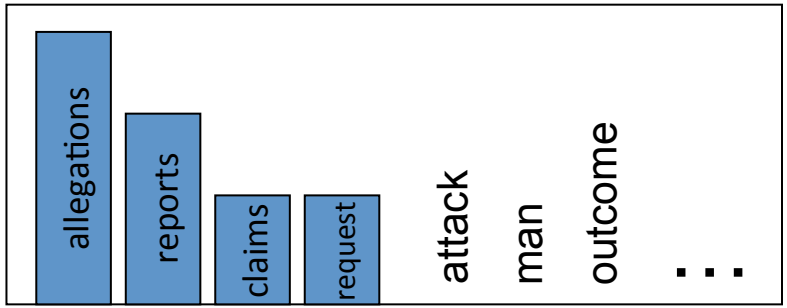
- Deal with words that are in our vocabulary but appear in a test set in an unseen context

- **Different ways:**
  - Add-1 smoothing
  - Backoff and interpolation
  - Good Turing Estimation
  - Kneser-Ney smoothing

The intuition of smoothing (from Dan Klein)

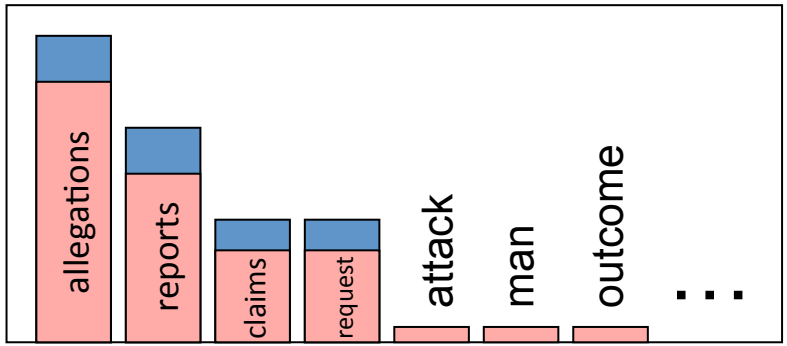
- When we have sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



- Steal probability mass to generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



Add-one Smoothing

- Also called Laplace smoothing
- Pretend we saw each word one more time than we did
- Just add one to all the counts!

MLE Estimate	Add-One Smoothing
$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$	$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)+1}{c(w_{i-1})+V}$

Example: Berkeley Restaurant Corpus

Laplace smoothed bigram counts:

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Laplace-smoothed bigrams

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Reconstituted counts

$$c^*(w_{n-1}w_n) = \frac{[C(w_{n-1}w_n)+1] \times C(w_{n-1})}{C(w_{n-1})+V}$$

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

Compare with raw bigram counts

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

	i	**want**	**to**	**eat**	**chinese**	**food**	**lunch**	**spend**
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

Add-1 estimation is a blunt instrument

- So add-1 isn't used for N-grams:
  - We'll see better methods

- But add-1 is used to smooth other NLP models
  - For text classification
  - In domains where the number of zeros isn't so huge.

Backoff and Interpolation

- Sometimes it helps to use **less** context
  - Condition on less context for contexts you haven't learned much about

- **Backoff:**
  - use trigram if you have good evidence,
  - otherwise bigram, otherwise unigram

- **Interpolation:**
  - mix unigram, bigram, trigram
- Interpolation works better

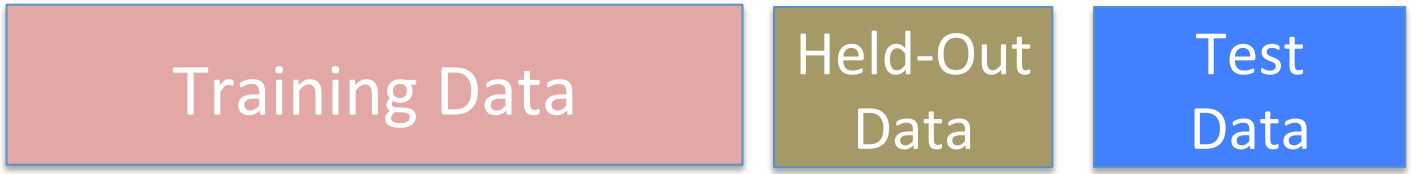
Linear Interpolation

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1 P(w_n|w_{n-2}w_{n-1}) \\ &\quad +\lambda_2 P(w_n|w_{n-1}) \\ &\quad +\lambda_3 P(w_n)\end{aligned}$$

Where $\sum_i \lambda_i = 1$

How to set the lambdas?

- Use a **held-out** corpus



- Choose λs to maximize the probability of held-out data:
  - Fix the N-gram probabilities (on the training data)
  - Then search for λs that give largest probability to held-out set

Good-Turing Smoothing

- Intuition: use the count of things we've **seen once** to estimate the count of things we've **never seen**

- N_c : Number of n-grams that occur c times
  - In other words, the frequency of frequency c
  - For example, for estimating bigram counts:
    - N_0 is the number of bigrams that occur 0 times in the training set
    - N_1 is the number of bigrams that occur once in the training set
    - Etc.

Example

- Sam I am I am Sam I do not eat

I → 3

Sam → 2

am → 2

do → 1

not → 1

eat → 1

$N_1 = 3$
$N_2 = 2$
$N_3 = 1$

Good Turing Smoothing Intuition

- You are fishing, and caught:
  - 10 crab, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel= 18 fish

- **How likely is it that next species is trout?**
  1/18
- **How likely is it that next species is new (i.e. bass)?**
  3/18 (because N1=3)
- **How likely is it that next species is trout?**
  Must be less than 1/18

How to estimate?

Good-Turing Smoothing

$$P^*_{GT}(\text{things with zero frequency}) = \frac{N_1}{N}$$

- re-estimate the count c $c^* = (c+1)\frac{N_{c+1}}{N_c}$

- Unseen (e.g. bass or catfish): **(MLE)** p=0 **P_{GT}** =3/18
- Seen once (e.g. trout): **(MLE)** 1/18 **(GT)** C^* = 2*N2/N1=2*1/3=2/3
  so **P_{GT}** (trout)=(2/3) / 18

Good-Turing Smoothing

- P_{GT}(things with zero frequncy)=N1/N
- $$c^* = (c+1)\frac{N_{c+1}}{N_c}$$
- re-estimate the count c

c (MLE)	N_c	c^* (GT)
0	74,671,100,000	0.0000270
1	2,018,046	0.446
2	449,721	1.26
3	188,933	2.24
4	105,668	3.24
5	68,379	4.22
6	48,190	5.19
7	35,709	6.21
8	27,710	7.24
9	22,280	8.25

Practical Issues

- In practice, the counts are only re-estimated for n-grams with low counts. Larger counts are considered reliable
- We can set a reliability threshold k
  - $c^* = c$, if $c > k$
  - The correct equation for calculating c^* when a threshold k is introduced is as follows:

$$c^* = \frac{(c+1)\frac{N_{c+1}}{N_c} - c\frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}}, \quad \text{for } 1 \leq c \leq k.$$

Advanced Topics

Absolute discounting

- Suppose we wanted to subtract a little from a count of 4 to save probability mass for the zeros
- How much to subtract ?
- It sure looks like c* = (c - .75)

Bigram count in training	Average bigram count in heldout set
0	.0000270
1	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21
9	8.26

Absolute Discounting Interpolation

- Save ourselves some time and just subtract 0.75 (or some d)!

discounted bigram

Interpolation weight

$$P_{\text{AbsoluteDiscounting}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1}) P(w_i)$$

unigram

- But should we really just use the regular unigram P(w)?

Kneser-Ney Smoothing

- Better estimate for probabilities of lower-order unigrams.
  - Shannon game: *I can't see without my reading* ***(glasses or Francisco)***?
  - "Francisco" is more common than "glasses"
  - … but "Francisco" always follows "San"
  - The unigram is useful exactly when we haven't seen this bigram

Kneser-Ney Smoothing – Cont.

- Instead of P(w): "How likely is w"

- $P_{continuation}(w)$: "How likely is w to appear as a novel continuation?
  - For each word, count the number of bigram types it completes
  - Every bigram type was a novel continuation the first time it was seen

$$P_{CONTINUATION}(w) \propto \left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|$$

San Francisco (common)
^Francisco (rare)

old glasses
the glasses
pretty glasses
....

Kneser-Ney Smoothing – Cont.

- How many times does w appear as a novel continuation:

$$P_{CONTINUATION}(w) \propto \left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|$$

- Normalized by the total number of word bigram types:

$$\left|\{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\}\right|$$

$$P_{CONTINUATION}(w) = \frac{\left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|}{\left|\{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\}\right|}$$

- A frequent word (Francisco) occurring in only one context (San) will have a low continuation probability

Kneser-Ney Smoothing – Cont.

$$P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1}) P_{CONTINUATION}(w_i)$$

λ is a normalizing constant; the probability mass we've discounted

$$\lambda(w_{i-1}) = \frac{d}{c(w_{i-1})} \left| \{w : c(w_{i-1}, w) > 0\} \right|$$

the normalized discount

The number of word types that can follow w_{i-1}
= # of word types we discounted
= # of times we applied normalized discount

Kneser-Ney Smoothing - Example

- They put on their clothes
- The paper is put onto the porch
- Adam put the car in the garage
- The girl put the keys on the garage

Find P_{KN}(the|put).

d=0.75

$$P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1}) P_{CONTINUATION}(w_i)$$

First term=max(2-0.75,0)/4=1.25/4=0.3125
Lambda = (0.75/4)*3=0.5625
$P_{Continuation}$= 3/27=0.11

$$\lambda(w_{i-1}) = \frac{d}{c(w_{i-1})} \left| \{w : c(w_{i-1}, w) > 0\} \right|$$

P_{KN}(the|put)=0.3125+(0.5625*0.11)=0.374375

$$P_{CONTINUATION}(w) = \frac{\left| \{w_{i-1} : c(w_{i-1}, w) > 0\} \right|}{\left| \{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\} \right|}$$

Unknown words

- Often, some words in the test set are not in the training set vocabulary
  - **Out Of Vocabulary** = OOV words
- Create an unknown word token <UNK>
  - Training of <UNK> probabilities
    - Create a fixed lexicon L of size V
    - At text normalization phase, any training word not in L changed to <UNK>
    - Now we train its probabilities like a normal word
  - At decoding time
    - If text input: Use UNK probabilities for any word not in training

Practical Notes

Language Modeling Toolkits

- SRILM
  - http://www.speech.sri.com/projects/srilm/

- KenLM
  - https://kheafield.com/code/kenlm/

Google N-Gram Release, August 2006

AUG
3

All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects,

...

That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Google N-Gram Release

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

Smoothing for Web-scale N-grams

- "Stupid backoff" (Brants *et al*. 2007)
- No discounting, just use relative frequencies
- 0.4 → was set heuristically

$$S(w_i \mid w_{i-k+1}^{i-1}) = \begin{cases} \dfrac{\text{count}(w_{i-k+1}^{i})}{\text{count}(w_{i-k+1}^{i-1})} & \text{if} \quad \text{count}(w_{i-k+1}^{i}) > 0 \\ 0.4 S(w_i \mid w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

$$S(w_i) = \frac{\text{count}(w_i)}{N}$$

Smoothing for Web-scale N-grams - Example

*I have an **umbrella***

What is the probability of umbrella given the previous context?

Do we have ***I have an umbrella*** in the corpus?	Yes.	***MLE***
	No.	**Continue**
Do we have ***have an umbrella*** in the corpus?	Yes.	0.4 * ***MLE***
	No.	**Continue**
Do we have ***an umbrella*** in the corpus?	Yes.	0.4 * 0.4 * ***MLE***
	No.	**Continue**
Do we have ***umbrella*** in the corpus?	Yes	0.4*0.4*0.4*0.4***MLE***

N-gram Smoothing Summary

- Add-1 smoothing:
  - OK for text categorization, not for language modeling
- The most commonly used method:
  - Extended Interpolated Kneser-Ney
- For very large N-grams like the Web:
  - Stupid backoff

Questions?