Statistical Learning for Text Data Analytics
Language Models

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

Spring 2018

∗Contents are based on materials created by Hongning Wang, Julia Hockenmaier, Dan Jurafsky, Dan Klein, Noah Smith, Slav Petrov, Yejin Choi, and Michael Collins

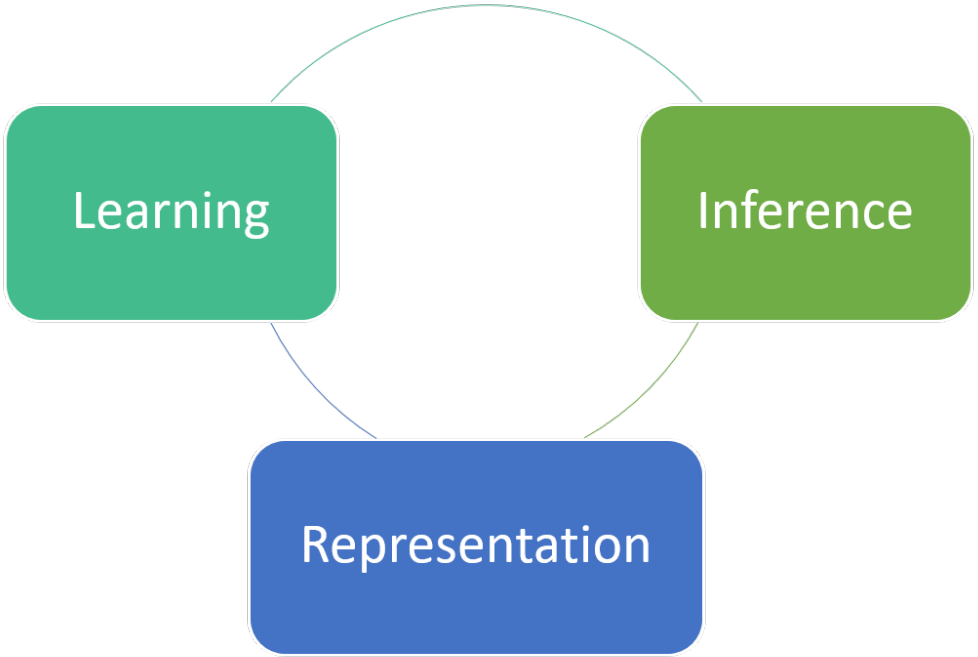
Reference Content

- Noah Smith. CSE 517: Natural Language Processing https://courses.cs.washington.edu/courses/cse517/16wi/
- Julia Hockenmaier. CS447: Natural Language Processing. http://courses.engr.illinois.edu/cs447
- Hongning Wang. CS6501 Text Mining. http://www.cs.virginia.edu/~hw5x/Course/Text-Mining-2015-Spring/_site/
- Dan Jurafsky. cs124/ling180: From Languages to Information. http://web.stanford.edu/class/cs124/
- Dan Klein. CS 288: Statistical Natural Language Processing. https://people.eecs.berkeley.edu/~klein/cs288/sp10/

Reference Content (Cont'd)

- Slav Petrov. Statistical Natural Language Processing. `https://cs.nyu.edu/courses/fall16/CSCI-GA.3033-008/`
- Chris Manning. CS 224N/Ling 237. Natural Language Processing. `https://web.stanford.edu/class/cs224n/`
- Yejin Choi. CSE 517 (Grad) Natural Language Processing. `http://courses.cs.washington.edu/courses/cse517/15wi/`
- Michael Collins. COMS W4705: Natural Language Processing. `www.cs.columbia.edu/~mcollins/courses/nlp2011/`

Course Topics



- Representation: language models, word embeddings, topic models
- Learning: supervised learning, semi-supervised learning, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

NLP applications: tasks introduced in Lecture 1

Overview

Random Variables

- A random variable is some aspect of the world about which we (may) have uncertainty
  - R = Is it raining?
  - T = Is it hot or cold?
  - D = How long will it take to drive to work?
  - W = A word that can be written by human.
- Random variables have domains
  - R in {true, false} (often write as {+r, -r})
  - T in {hot, cold}
  - D in $[0, \infty)$
  - W in vocabulary

Probability Distributions

- Unobserved random variables have distributions
- Must have: $\forall x, P(X = x) \geq 0$ and $\sum_x P(X = x) = 1$

Example (Probability Distributions)

T	$P(T)$
hot	0.7
cold	0.3

W	$P(W)$
sun	0.5
rain	0.3
fog	0.2

We will have more examples when introducing topic models

Joint Distributions

- A joint distribution over a set of random variables: $X_1, X_2, \ldots, X_n$ specifies a real number for each assignment (or outcome):

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \text{ or } P(x_1, x_2, \ldots, x_n)$$

Must obey:

$$P(x_1, x_2, \ldots, x_n) \geq 0 \text{ and } \sum_{x_1, x_2, \ldots, x_n} P(x_1, x_2, \ldots, x_n) = 1$$

Example (Joint Distribution)

T	W	$P(T, W)$
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Events

- An event is a set E of outcomes:
  $P(E) = \sum_{x_1,x_2,\ldots,x_n \in E} P(x_1, x_2, \ldots, x_n)$
- From a joint distribution, we can calculate the probability of any event
  - Probability that its hot AND sunny?
  - Probability that its hot?
  - Probability that its hot OR sunny?

T	W	$P(T, W)$
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

Example (Marginal Distribution)

T	W	$P(T, W)$
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

T	$P(T)$
hot	0.5
cold	0.5

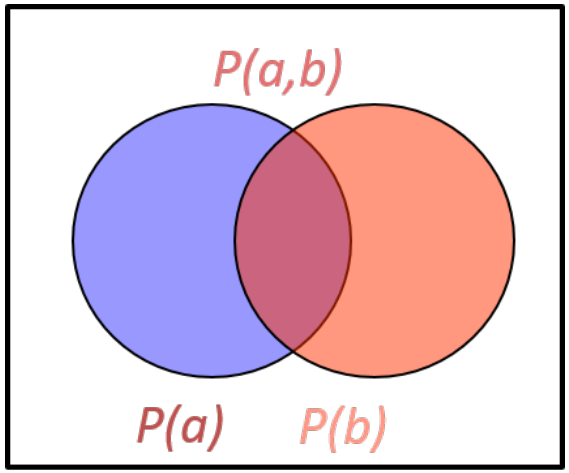
$P(t) = \sum_w P(t, w)$

W	$P(W)$
sun	0.6
rain	0.4

$P(w) = \sum_t P(t, w)$

Conditional Probabilities

- A simple relation between joint and conditional probabilities
- In fact, this is taken as the definition of a conditional probability $P(a|b) = \frac{P(a,b)}{P(b)}$



Example (Conditional Probabilities)

T	W	$P(T,W)$
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

W	$P(W\|hot)$
sun	0.8
rain	0.2

$\frac{P(W,T=hot)}{P(T=hot)}$

W	$P(W\|cold)$
sun	0.4
rain	0.6

$\frac{P(W,T=cold)}{P(T=cold)}$

Product Rule, Chain Rule, and Bayes' Rule

- Sometimes have conditional distributions but want the joint (product rule)
  $P(x|y)P(y) = P(x,y) \Leftrightarrow P(x|y) = \frac{P(x,y)}{P(y)}$
- More generally, can always write any joint distribution as an incremental product of conditional distributions (chain rule)
  $P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_2, x_1)$
  $P(x_1, x_2, \ldots, x_n) = \prod_i P(x_i|x_1, \ldots, x_{i-1})$
- Two ways to factor a joint distribution over two variables:
  $P(x,y) = P(x|y)P(y) = P(y|x)P(x)$
- Dividing, we get Bayes' Rule:
  $P(x|y) = \frac{P(y|x)P(x)}{P(y)}$

Independence and Conditional Independence

- X and Y are independent if $\forall x, y,\ P(x, y) = P(x)P(y)$
- X and Y are conditionally independent given Z if $\forall x, y, z,\ P(x, y|z) = P(x|z)P(y|z)$
- (Conditional) independence is a property of a distribution

Probability Models

Building a probability model consists of two steps:

- Defining the model
- Estimating the models parameters

Models (almost) always make independence assumptions.

- That is, even though X and Y are not actually independent, our model may treat them as independent.
- This reduces the number of model parameters that we need to estimate (e.g. from N^2 to $2N$)

Overview

What is a Statistical Language Model (LM)?

- A model specifying probability distribution over word sequences
  - P("Today is Wednesday") ≈ 0.001
  - P("Today Wednesday is") ≈ 0.0000000000001
  - P("The eigenvalue is positive") ≈ 0.00001
- It can be regarded as a probabilistic mechanism for "generating" text, thus also called a "generative" model

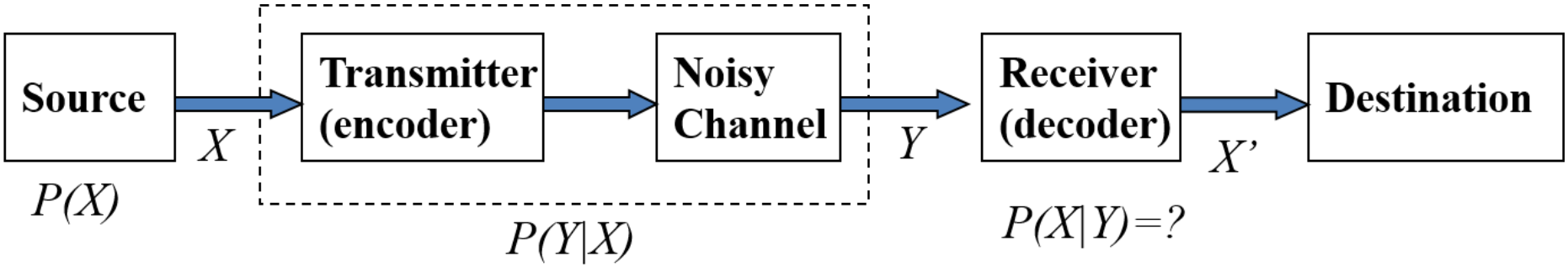
Why is a LM Useful?

- Provide a principled way to quantify the uncertainties associated with natural language
- Allow us to answer questions like:
  - Given that we see "John" and "feels", how likely will we see "happy" as opposed to "habit" as the next word? (speech recognition)
  - Given that we observe "baseball" three times and "game" once in a news article, how likely is it about "sports" v.s. "politics" (text categorization)
  - Given that a user is interested in sports news, how likely would the user use "baseball" in a query? (information retrieval)

Measure the Fluency of Documents

- How likely this document is generated by a given language model
  - If $P_{machine-learning}(d) > P_{health}(d)$, document d belongs to machine learning related topics
  - If $P_{user_a}(d_1) > P_{user_a}(d_2)$, recommend d_1 to $user_a$

Source-Channel Framework [Shannon '48]



$$\hat{X} = \arg\max_X P(X|Y) = \arg\max_X P(Y|X)P(X) \text{ (Bayes Rule)}$$

When X is text, $P(X)$ is a language model

	X	Y
Speech recognition	Word sequence	Speech signal
Machine translation	English sentence	Chinese sentence
OCR Error Correction	Correct word	Erroneous word
Information Retrieval	Document	Query
Summarization	Summary	Document

Language Model for Text

- Goal: Assign useful probabilities $P(X)$ to sentences/documents X
  - Input: many observations of training sentences X
  - Output: system capable of computing $P(X)$
- Probabilities should broadly indicate plausibility of sentences
  - P(I saw a van) $\gg$ P(eyes awe of an)
  - Not grammaticality: P(artichokes intimidate zippers) ≈ 0
  - In principle, "plausible" depends on the domain, context, speaker...

Language Model for Text

- Probability distribution over word sequences (chain rule)
  $P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_1, w_2, \ldots, w_{n-1})$
- Complexity – $O(V^{n^*})$
  - V: vocabulary size
  - n^*: maximum document (or sentence) length
  - We need independence assumptions!

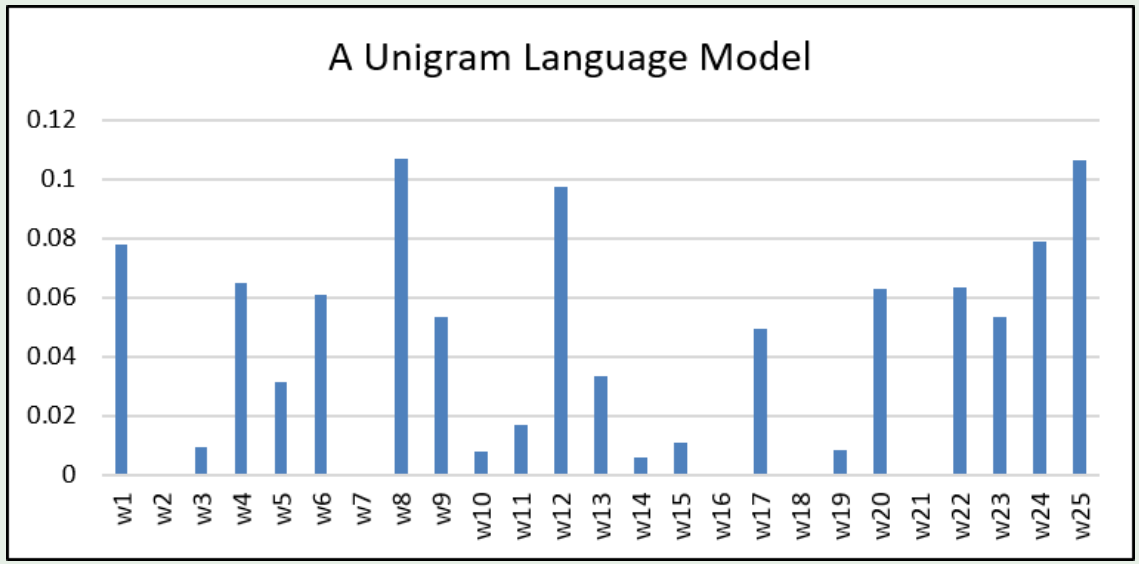
Example

- 475,000 main headwords in Webster's Third New International Dictionary
- Average English sentence length is 14.3 words
- A rough estimate: $O(475,000^{14}) \approx 3.38e^{66}\,TB$

Unigram Language Model

- Generate a piece of text by generating each word independently
  - $P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2)\ldots P(w_n)$
- Essentially a multinomial distribution over the vocabulary
- The simplest and most popular choice!

Example (Unigram Language Model)



More Sophisticated LMs

N-gram language models

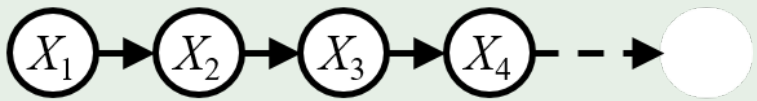
- Assumes each word depends only on the last n-1 words
  - bigram $P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1})$
  - trigram $P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2|w_1)\ldots P(w_n|w_{n-1}, w_{n-2})$
- Such independence assumptions are called Markov assumptions (of order n-1)
  $P(w_i|w_1, \ldots, w_{i-1}) = P(w_i|w_{i-n+1}, \ldots, w_{i-1})$

Markov Models

- Value of X at a given time is called the state
- Parameters: called transition probabilities, specify how the state evolves over time (also, initial state probabilities)
- Stationarity assumption: transition probabilities the same at all times

Example (First-order Markov Chain)

"Markov" generally means that given the present state, the future and the past are independent

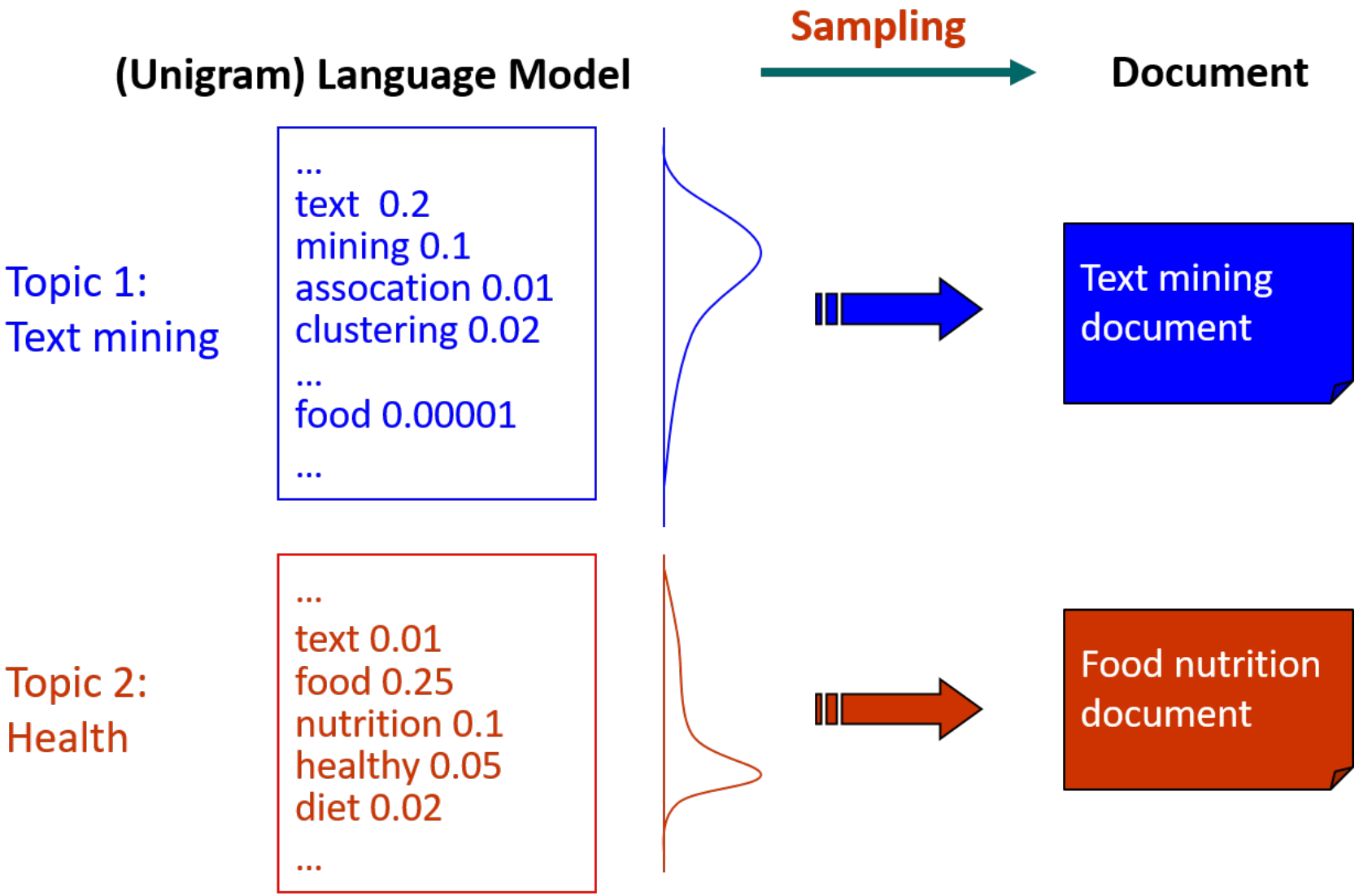


$P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2|X_1)\ldots P(X_n|X_{n-1}) =$
$P(X_1)\prod_{t=1}^{n} P(X_t|X_{t-1})$

Why Just Unigram Models (in most cases)?

- Difficulty in moving toward more complex models
  - They involve more parameters, so need more data to estimate
  - They increase the computational complexity significantly, both in time and space
- Capturing word order or structure may not add so much value for "topical inference"
- But, using more sophisticated models can still be expected to improve performance ...

Generative View of Text Documents

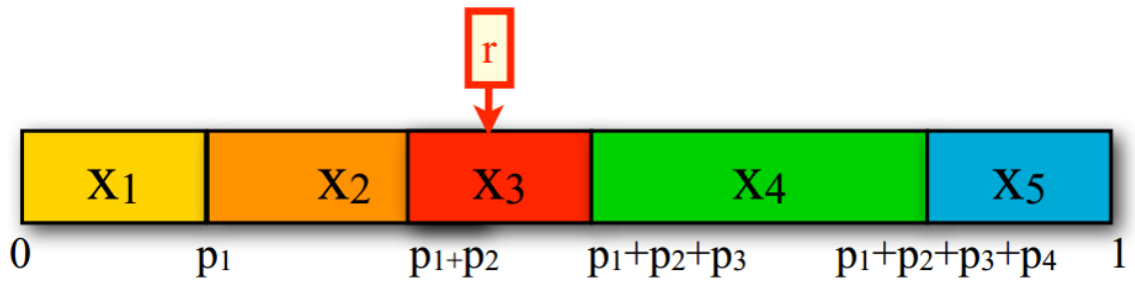


Computer Simulation

Sample from a discrete distribution $P(X)$, assuming n outcomes in the event space X

Algorithm 1 Sample from a distribution $P(X)$

1: **for** $t = 1$ to T **do**
2: Divide the interval $[0, 1]$ into n intervals according to the probabilities of the outcomes
3: Generate a random number r between 0 and 1
4: Return x_i where r falls into $[\sum_0^{i-1} p_i, \sum_0^i p_i]$
5: **end for**



Generating Text from Language Models

Example

P(of) = 3/66	P(her) = 2/66
P(Alice) = 2/66	P(sister) = 2/66
P(was) = 2/66	P(,) = 4/66
P(to) = 2/66	P(') = 4/66

Under a unigram language model:

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

Generating Text from Language Models

Example

P(of) = 3/66	P(her) = 2/66
P(Alice) = 2/66	P(sister) = 2/66
P(was) = 2/66	P(,) = 4/66
P(to) = 2/66	P(') = 4/66

Under a unigram language model:

The same likelihood!

beginning by, very Alice but was and?
reading no tired of to into sitting
sister the, bank, and thought of without
her nothing: having conversations Alice
once do or on she it get the book her had
peeped was conversation it pictures or
sister in, 'what is the use had twice of
a book''pictures or' to

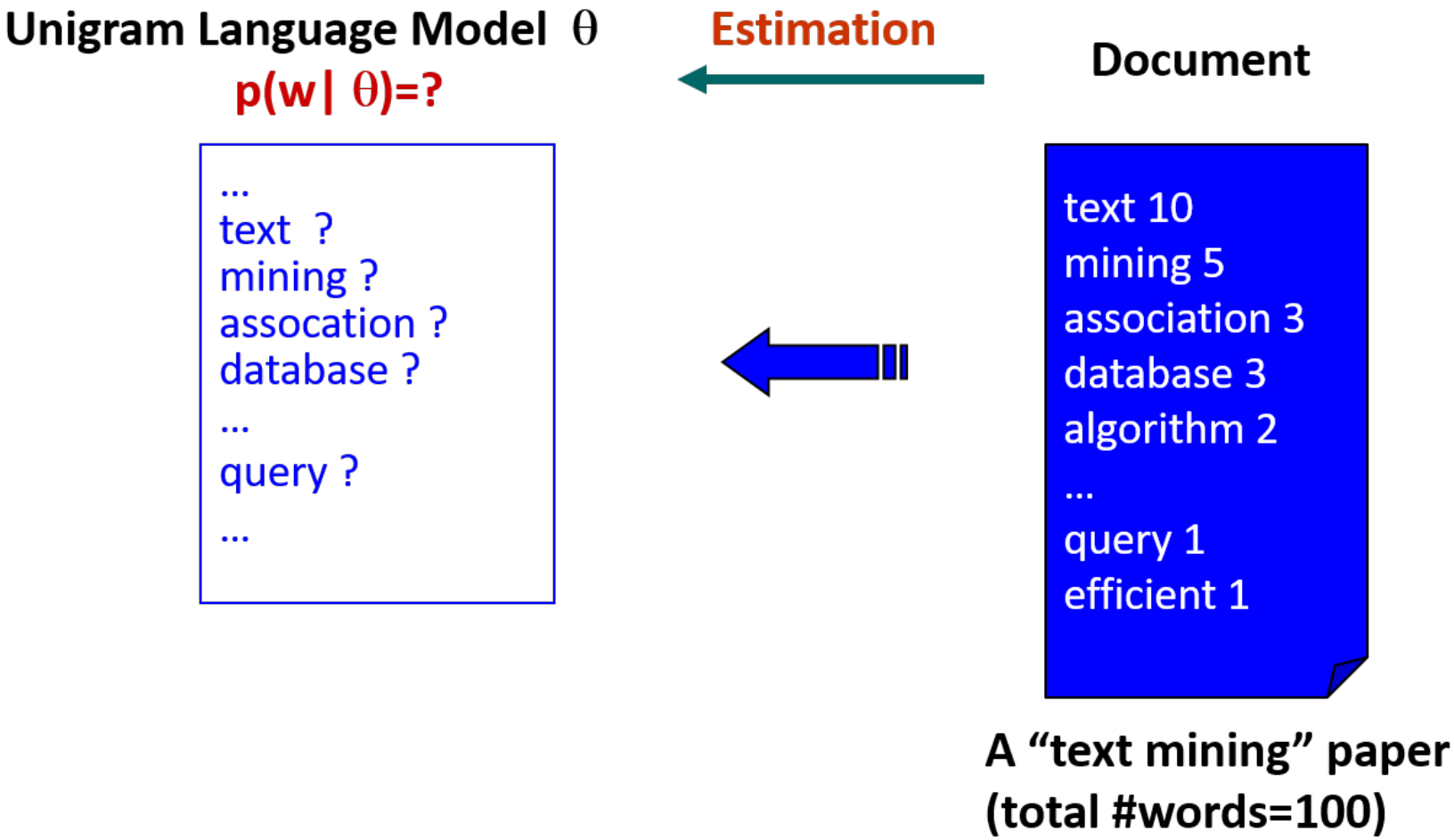
N-gram Language Models Will Help

Example (Generated from language models of New York Times)

- Unigram
  - Months the my and issue of year foreign new exchanges september were recession exchange new endorsed a q acquire to six executives.
- Bigram
  - 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 told M.X. corporation of living on information such as more frequently fishing to keep her.
- Trigram
  - They also point to ninety nine point six billon dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions.

Overview

Estimation of Language Models



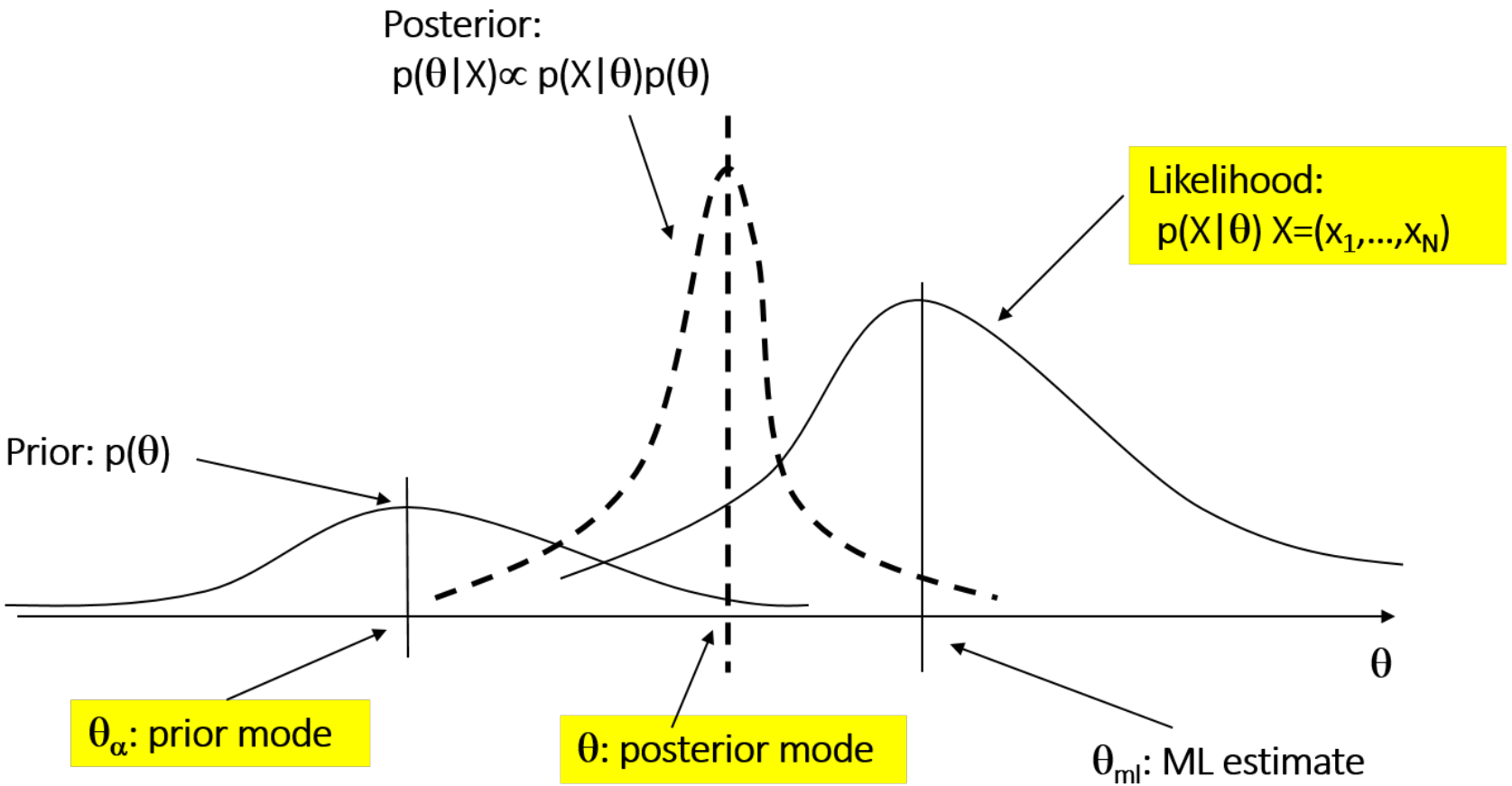
Parameter Estimation

- General setting
  - Given a (hypothesized & probabilistic) model that governs the random experiment
  - The model gives a probability of any data $P(\mathcal{X}|\boldsymbol{\theta})$ that depends on the parameter $\boldsymbol{\theta}$
  - Now, given actual sample data $\mathcal{X} = \mathbf{x}_1, \ldots, \mathbf{x}_n$, what can we say about the value of $\boldsymbol{\theta}$?
- Intuitively, take our best guess of $\boldsymbol{\theta}$
  - "best" means "best explaining/fitting the data"
- Generally an optimization problem

Maximum Likelihood vs. Bayesian

- Maximum likelihood estimation
  - "Best" means "data likelihood reaches maximum"
    $$\hat{\theta} = \arg\max_{\theta} P(\mathcal{X}|\theta)$$
  - Issue: small sample size
- Bayesian estimation
  - "Best" means being consistent with our "prior" knowledge and explaining data well
    $$\hat{\theta} = \arg\max_{\theta} P(\theta|\mathcal{X}) = \arg\max_{\theta} P(\mathcal{X}|\theta)P(\theta)$$
  - A.k.a, maximum a posterior estimation
  - Issue: how to define prior?

Illustration of Bayesian Estimation



Corpora

- A corpus is a collection of text
  - Annotated in some way: supervised learning
  - Sometimes just lots of text without any annotations: unsupervised learning
  - Balanced vs. uniform corpora
- Examples
  - Newswire collections: 500M+ words
  - Brown corpus: 1M words of tagged balanced text
  - Penn Treebank: 1M words of parsed WSJ
  - Canadian Hansards: 10M+ words of aligned French / English sentences
  - The Web: billions of words of who knows what

Unigram Modeling

- Data corpus: a collection of words, $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$
- Model: multinomial distribution $P(\mathcal{W}|\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_V)$, where
  - $\theta_i = P(v_i)$
  - $v_i \in \mathcal{V}$
  - $\mathcal{V}$ is the vocabulary
  - $|\mathcal{V}| = V$
- Count of words in corpus $\mathbf{u} = (u_1, \ldots, u_V)$ where $u_i = c(v_i)$ is the count of v_i shown in $\mathcal{W}$, $\sum_i u_i = N$

Unigram Modeling

- "Bag of words" assumes the words are sampled from a multinomial distribution $\mathbf{u} \sim \text{Multi}(\boldsymbol{\theta})$

$$P(\mathbf{u}|\boldsymbol{\theta}) = \binom{N}{\mathbf{u}} \prod_{i=1}^{V} \theta_i^{u_i} \triangleq \text{Mult}(\mathbf{u}|\boldsymbol{\theta}, N), \textit{where} \binom{N}{\mathbf{u}} = \frac{N!}{\prod_i u_i!}$$

If we focus on a single trial, we have:

$$P(w|\boldsymbol{\theta}) = P(w = v_i) = \prod_{i=1}^{V} \theta_i^{\delta_{w=v_i}} \triangleq \text{Mult}(w|\boldsymbol{\theta})$$

- Maximum likelihood estimator: $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{W}|\boldsymbol{\theta})$

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_{j=1}^{N} P(w_j|\boldsymbol{\theta}) = \prod_{i=1}^{V} P(v_i)^{u_i} = \prod_{i=1}^{V} \theta_i^{u_i}$$

Maximum Likelihood Estimation: $\hat{\theta} = \arg\max_{\theta} P(\mathcal{W}|\theta)$

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_i^V \theta_i^{u_i}$$

(log likelihood)

$$\Rightarrow \log P(\mathcal{W}|\boldsymbol{\theta}) = \sum_i^V u_i \log \theta_i$$

(Lagrange multiplier to make θ be a distribution)

$$\Rightarrow L(\mathcal{W}, \boldsymbol{\theta}) = \log P(\mathcal{W}|\boldsymbol{\theta}) = \sum_i^V u_i \log \theta_i + \lambda(\sum_i \theta_i - 1)$$

(Set partial derivatives to zero)

$$\Rightarrow \frac{\partial L}{\partial \theta_i} = \frac{u_i}{\theta_i} + \lambda$$

Since $\sum_i^V \theta_i = 1$, we have $\lambda = -\sum_i^V u_i$

$$\Rightarrow \theta_i = \frac{u_i}{\sum_i^V u_i} = \frac{u_i}{N} \ (\textit{Maximum Likelihood Estimation}, \textit{MLE})$$

The Problems with Unigram Modeling

- Pros:
  - Easy to understand
  - Cheap
  - Good enough for information retrieval (maybe)
- Cons:
  - "Bag of words" assumption is linguistically inaccurate
    - P(the the the the) $\gg$ P(I want ice cream)
  - Data sparseness; high variance in the estimator
  - "Out of vocabulary" problem

N-gram model

- Markov modeling

$$
\begin{aligned}
&= P(w_1, \ldots, w_N) \\
&= \prod_{i=1}^{N} P(w_i | w_1, \ldots, w_{i-1}) \ (\textit{chain rule}) \\
&= \prod_{i=1}^{N} P(w_i | w_{i-1}, \ldots, w_{i-n+1}) \ (\textit{Markov model})
\end{aligned}
$$

  - (n - 1)th-order Markov assumption $\equiv$ n-gram model
    - Unigram model is the n $= 1$ case
    - For a long time, trigram models (n $= 3$) were widely used
    - 5-gram models (n $= 5$) are not uncommon now in machine translation systems

- Parameter estimation

$$
P(w_i | w_{i-1}, \ldots, w_{i-n+1}) = \frac{c(v^1 = w_i, \ldots, v^n = w_{i-n+1})}{c(v^1 = w_{i-1}, \ldots, v^{n-1} = w_{i-n+1})}
$$

v^j is a unique word v at position j

Estimating N-gram models: A Running Example

Example (Bigram Model)

- Bracket each sentence by special start and end symbols:
  $\langle s \rangle$ Alice was beginning to get very tired ... $\langle s \rangle$
  (We only assign probabilities to strings $\langle s \rangle ... \langle s \rangle$)
- Count the frequency of each n-gram
  c($\langle s \rangle$, Alice) = 1, c(Alice, was) = 1,
- Normalize to get the probability
  $P(w_i|w_{i-1}) = \frac{c(w_i, w_{i-1})}{c(w_{i-1})}$
  $P(was|Alice) = \frac{c(was, Alice)}{c(Alice)}$
- This is called a relative frequency estimate of $P(w_i|w_{i-1})$

The Problems with N-gram Modeling

- The curse of dimensionality: the number of parameters grows exponentially in n
- Pros:
  - Easy to understand
  - Cheap (with modern hardware; Lin and Dyer (2010))
  - Good enough for machine translation, speech recognition, ...
- Cons:
  - Markov assumption is linguistically inaccurate
    - (But not as bad as unigram models!)
  - Data sparseness; high variance in the estimator
    - most n-grams will never be observed, even if they are linguistically plausible
  - "Out of vocabulary" problem

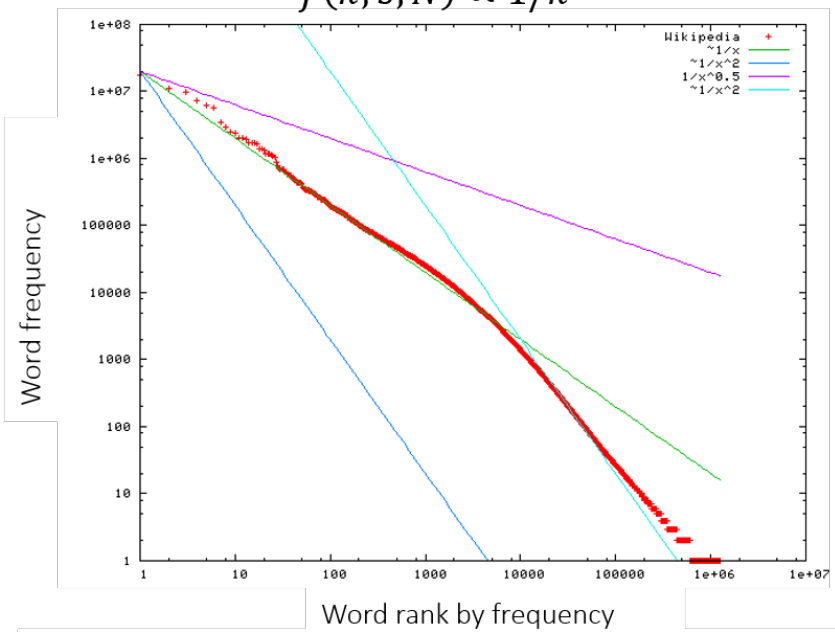
Overview

Problem with MLE: Unseen Events

- We estimated a model on 440K word tokens, but:
  - Only 30,000 unique words occurred
  - Only 0.04% of all possible bigrams occurred
- This means any word/n-gram that does not occur in the training data has zero probability!
- No future documents can contain those unseen words/n-grams

- In natural language:
  - A small number of events (e.g. words) occur with high frequency
  - A large number of events occur with very low frequency
  - Zipfs law: the long tail

$$f(k; s, N) \propto 1/k^s$$



A plot of word frequency in Wikipedia (Nov 27, 2006)

Dealing with Unseen Events

- Relative frequency estimation assigns all probability mass to events in the training corpus
- But we need to reserve some probability mass to events that don't occur in the training data
  - Unseen events = new words, new bigrams
- Important questions:
  - What possible events are there?
  - How much probability mass should they get?

Dealing with Unseen Events

- If we want to assign non-zero probabilities to unseen events
  - Unseen events = new words, new n-grams
  - Discount the probabilities of observed words
- General procedure
  - Reserve some probability mass of words seen in a document/corpus
  - Re-allocate it to unseen words

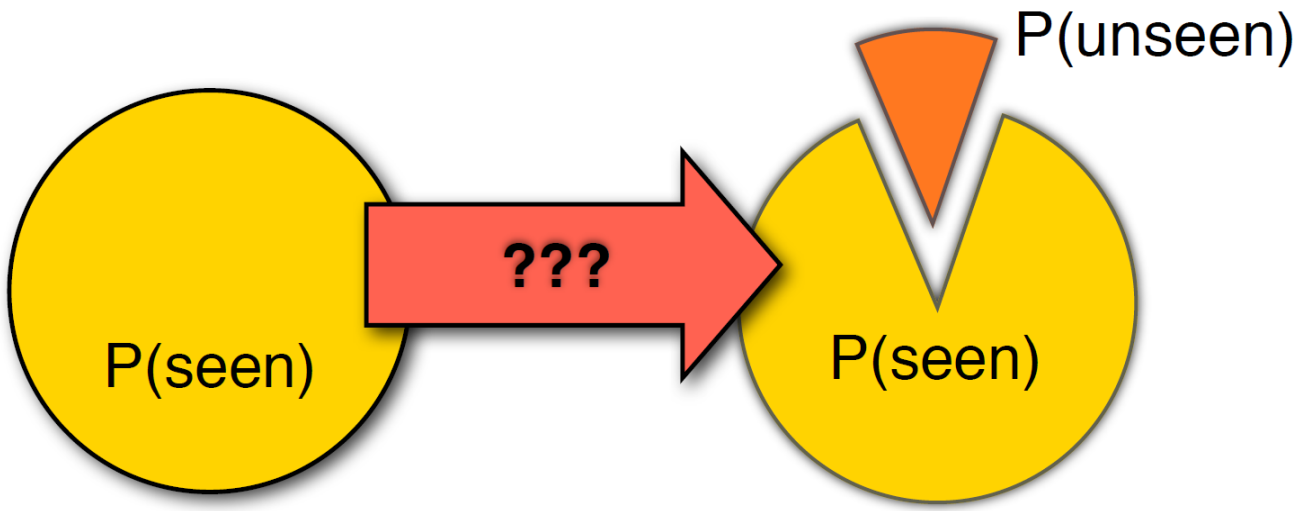
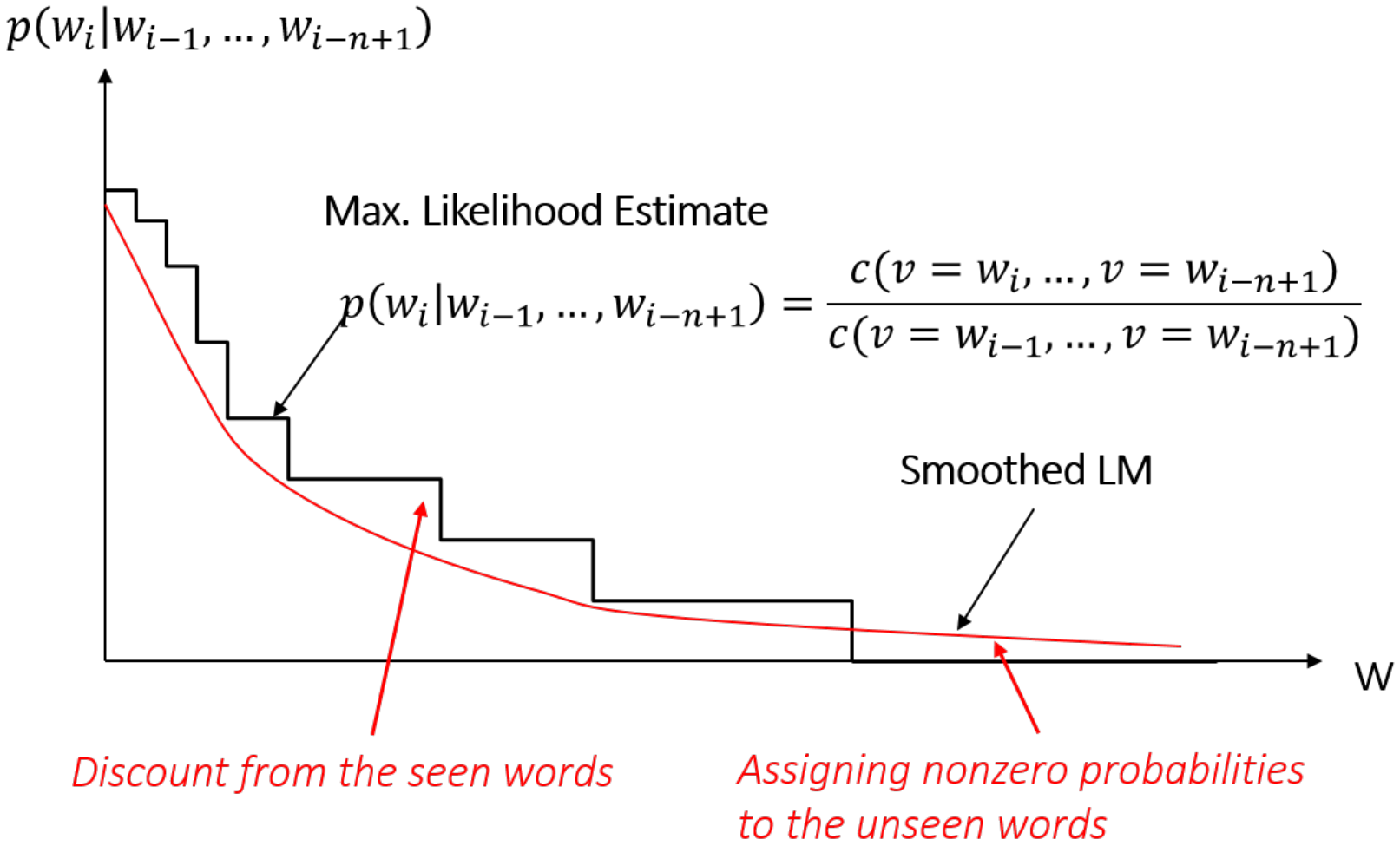


Illustration of N-gram Language Model Smoothing



What Unseen Events May Occur?

- Simple distributions:

$$P(X = x)$$

  (e.g. unigram models)
- Possibility:
  - The outcome x has not occurred during training (i.e. is unknown)
  - We need to reserve mass in $P(X)$ for x
- What outcomes x are possible?
- How much mass should they get?

What Unseen Events May Occur?

- Simple conditional distributions:

$$P(X = x|Y = y)$$

  (e.g. bigram models)
- Possibility:
  - The outcome x has been seen, but not in the context of $Y = y$:
  - We need to reserve mass in $P(X|Y = y)$ for $X = x$
- The conditioning variable y has not been seen:
  - We have no $P(X|Y = y)$ distribution.
  - We need to drop the conditioning context $Y = y$ and use $P(X)$ instead.

What Unseen Events May Occur?

- Complex conditional distributions:

$$P(X = x|Y = y, Z = z)$$

  (e.g. trigram models)
- Possibility:
  - The outcome x has been seen, but not in the context of $(Y = y, Z = z)$:
  - We need to reserve mass in $P(X|Y = y, Z = z)$ for $X = x$
- The joint conditioning event $(Y = y, Z = z)$ has not been seen:
  - We have no $P(X|Y = y, Z = z)$ distribution.
  - We need to drop z and use $P(X|Y = y)$ instead.

Examples

Example

- Training data: The wolf is an endangered species
- Test data: The wallaby is endangered

Unigram	Bigram	Trigram
$P(the)$	$P(the\|\langle s\rangle)$	$P(the\|\langle s\rangle)$
$\times\ P(wallaby)$	$\times\ P(wallaby\|the)$	$\times\ P(wallaby\|the,\langle s\rangle)$
$\times\ P(is)$	$\times\ P(is\|wallaby)$	$\times\ P(is\|wallaby, the)$
$\times\ P(endangered)$	$\times\ P(endangered\|is)$	$\times\ P(endangered\|is, wallaby)$

Examples

Example

- Training data: The wolf is an endangered species
- Test data: The wallaby is endangered

Unigram	Bigram	Trigram
$P(the)$	$P(the\|\langle s\rangle)$	$P(the\|\langle s\rangle)$
$\times$ $P(wallaby)$	$\times$ $P(wallaby\|the)$	$\times$ $P(wallaby\|the,\langle s\rangle)$
$\times$ $P(is)$	$\times$ $P(is\|wallaby)$	$\times$ $P(is\|wallaby, the)$
$\times$ $P(endangered)$	$\times$ $P(endangered\|is)$	$\times$ $P(endangered\|is, wallaby)$

- Case 1:
  - $P(wallaby)$, $P(wallaby|the)$, $P(wallaby|the,\langle s\rangle)$
  - What is the probability of an unknown word (in any context)?

Examples

Example

- Training data: The wolf is an endangered species
- Test data: The wallaby is endangered

Unigram	Bigram	Trigram
$P(the)$	$P(the\|\langle s\rangle)$	$P(the\|\langle s\rangle)$
$\times\ P(wallaby)$	$\times\ P(wallaby\|the)$	$\times\ P(wallaby\|the, \langle s\rangle)$
$\times\ P(is)$	$\times\ P(is\|wallaby)$	$\times\ P(is\|wallaby, the)$
$\times\ P(endangered)$	$\times\ P(endangered\|is)$	$\times\ P(endangered\|is, wallaby)$

- Case 2:
  - $P(endangered|is)$
  - What is the probability of a known word in a known context, if that word hasn't been seen in that context?

Examples

Example

- Training data: The wolf is an endangered species
- Test data: The wallaby is endangered

Unigram	Bigram	Trigram
$P(the)$	$P(the\|\langle s\rangle)$	$P(the\|\langle s\rangle)$
$\times\ P(wallaby)$	$\times\ P(wallaby\|the)$	$\times\ P(wallaby\|the, \langle s\rangle)$
$\times\ P(is)$	$\times\ P(is\|wallaby)$	$\times\ P(is\|wallaby, the)$
$\times\ P(endangered)$	$\times\ P(endangered\|is)$	$\times\ P(endangered\|is, wallaby)$

- Case 3:
  - $P(is|wallaby)$, $P(is|wallaby, the)$, $P(endangered|is, wallaby)$
  - What is the probability of a known word in an unseen context?

Dealing with Unknown Words: The Simple Solution

- Training:
  - Assume a fixed vocabulary (e.g. all words that occur at least twice (or n times) in the corpus)
  - Replace all other words by a token $\langle UNK \rangle$ (or a special OOV)
  - Estimate the model on this corpus
- Testing:
  - Replace all unknown words by $\langle UNK \rangle$
  - Run the model

This requires a large training corpus to work well!
Note: You cannot fairly compare two language models that apply different *UNK* treatments!

Overview

1. Basic Concepts of Probability
2. Language Models
3. Parameter Estimation
   - Maximum Likelihood
   - Unseen Events (Words)
   - Add-one Smoothing
   - Add-K Smoothing and Bayesian Estimation
   - Good-turing Smoothing
   - Interpolation Smoothing
4. Evaluation

Further Reading

- Manning et al. (2008). Introduction to information retrieval. Chapter 12: Language models for information retrieval.
- Jurafsky and Martin (2017). Speech and Language Processing. Chapter 4: N-Grams. `https://web.stanford.edu/~jurafsky/slp3/`
- Chen and Goodman (1996). An empirical study of smoothing techniques for language modeling.
- Collins (2011). Course notes for COMS w4705: Language modeling, 2011. `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/lm.pdf`
- Zhu (2010). Course notes for cs769: Language modeling, 2011. `http://pages.cs.wisc.edu/~jerryzhu/cs769/lm.pdf`

References

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *ACL*, pages 310–318.

Collins, M. (2011). Course notes for coms w4705: Language modeling. Technical report, Columbia University.

Jurafsky, D. and Martin, J. H. (2017). *Speech and Language Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Lin, J. and Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Zhu, X. J. (2010). Course notes for cs769: Language modeling. Technical report, University of Wisconsin-Madison.