CS 6501 Natural Language Processing

Statistical Language Modeling

Yangfeng Ji

September 10, 2018

Department of Computer Science
University of Virginia

UNIVERSITY of VIRGINIA | ENGINEERING

Announcements

- Project 1 is out, due on **Sept. 23**
- Sign up for the group projects
  - You can find the Google spreadsheet link on the course Github page
  - Leave the `project topic` field blank, if you don't have an idea for now
- Slides will be released online *before* class

Overview

1. Basic Probability
2. Language Modeling: Motivating examples
3. Language Modeling: Formulation
4. Parameter Estimation
5. Evaluation
6. Resources

Basic Probability

Quick Review of Probability

- Event space – in this class, usually discrete
  - notations: $\mathcal{X}, \mathcal{Y}$
  - example: $\mathcal{Y} = \in \{\text{positive}, \text{negative}\}$
- Random variables
  - notations: X, Y
  - example: document label
- Typical statement:
  random variable X takes value $x \in \mathcal{X}$ with probability $P(X = x)$, or in shorthand, $P(x)$
- $P(X)$ and $P(x)$

Quick Review of Probability (II)

- Conditional probability $P(Y|X)$
- Joint probability
  $P(X, Y) = P(X)P(Y|X) = P(Y)P(X|Y)$
- Independence $P(X, Y) = P(X) \cdot P(Y)$ if $X \perp\!\!\!\perp Y$

Probability Estimation

Notations

- $P(X)$: true probability of X
- $Q(X)$: estimated probability of X
  - In some literature, it is $\hat{P}(X)$

Probability Estimation

Notations

- $P(X)$: true probability of X
- $Q(X)$: estimated probability of X
  - In some literature, it is $\hat{P}(X)$

Probability estimation

$$P(X = x) \approx Q(X = x) = \frac{c(x)}{N} \tag{1}$$

where N is the number of total experiments and $c(x)$ is the number of experiments with outcome x.

Example

Is my student in the lab?

In lab? (X)	Yes	Yes	No	Yes	No	Yes	Yes	No

- Event space $\mathcal{X} = \{\text{Yes}, \text{No}\}$
- $P(X)$:

$$P(X = \text{Yes}) = \frac{5}{8} = 0.625 \tag{2}$$

Example

Is my student in the lab?

In lab? (X)	Yes	Yes	No	Yes	No	Yes	Yes	No
Weather (Y)	Sunny	Rain	Rain	Sunny	Sunny	Sunny	Rain	Rain

- Event space $(\mathcal{X}, \mathcal{Y})$
  $\{(\text{Yes,Sunny}), (\text{No, Sunny}), (\text{Yes, Rain}), (\text{No, Rain})\}$
- $P(X, Y)$

$$\begin{aligned} P(\text{Yes}|\text{Sunny}) &= \frac{3}{4} = 0.75 \\ P(\text{Yes}|\text{Rain}) &= \frac{2}{4} = 0.5 \end{aligned} \tag{2}$$

Example

Is my student in the lab?

In lab? (X)	Yes	Yes	No	Yes	No	Yes	Yes	No
Weather (Y)	Sunny	Rain	Rain	Sunny	Sunny	Sunny	Rain	Rain
Time (Z)	9AM	2PM	12PM	9AM	9AM	2PM	2PM	11PM

Example

Is my student in the lab?

In lab? (X)	Yes	Yes	No	Yes	No	Yes	Yes	No
Weather (Y)	Sunny	Rain	Rain	Sunny	Sunny	Sunny	Rain	Rain
Time (Z)	9AM	2PM	12PM	9AM	9AM	2PM	2PM	11PM

- Requires more data for $P(X|Y,Z)$
- Even for $P(X)$ and $P(X|Y)$: more data, more reliable estimation

Language Modeling: Motivating examples

Motivating Example (I): Speech recognition

`I saw a van`

vs.

`eyes awe of an`

[Jurafsky, 2018]

Motivating Example (II): Machine translation

Measure the quality of a sentence in machine translation

Chinese	晚饭去哪里吃?
Word by word	`Dinner go where eat?`
Google translate	`Where do you go for dinner?`

A score function $\Psi(x)$ in MT:

$$\Psi(\texttt{Where do you go for dinner?}) > \Psi(\texttt{Dinner go where eat?})$$

Motivating Example (III): Word prediction

How to predict the next word, given a half sentence?

Example

`Bob gave Tina the burger, because she was` ________

$\Psi(x|\texttt{Bob gave Tina the burger, because she was})$

1. `study`
2. `hungry`
3. `sleepy`
4. $\cdots$

A model agnostic to syntactic/semantic information.

How to Model a Sentence?

Use a probability function over a sentence with words $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$

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

- Random variables/vector $\boldsymbol{X}$
- Event space $\mathcal{X}$

Language Modeling: Formulation

The Language Modeling Problem

- Finite vocabulary $\mathcal{V}$

$$\mathcal{V} = \{\textsc{the}, \textsc{a}, \textsc{student}, \textsc{computer}, \textsc{with} \ldots\}$$

[Collins, 2017]

The Language Modeling Problem

- Finite vocabulary $\mathcal{V}$

$$\mathcal{V} = \{\textsc{the}, \textsc{a}, \textsc{student}, \textsc{computer}, \textsc{with} \ldots\}$$

- Event space: infinite set of strings, $\mathcal{V}^+$
  - `the`
  - `a`
  - `a student`
  - `a student with a computer`
  - `...`

[Collins, 2017]

The Language Modeling Problem (II)

We need a probability distribution P that satisfies

$$\sum_{\boldsymbol{x} \in \mathcal{V}^+} P(\boldsymbol{x}) = 1 \tag{2}$$

where

$$P(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in \mathcal{V}^+ \tag{3}$$

Example Sentences

$$
\begin{aligned}
P(\texttt{the}) &= 10^{-12} \\
P(\texttt{a}) &= 10^{-11} \\
P(\texttt{a student}) &= 10^{-13} \\
P(\texttt{a student with a telescope}) &= 10^{-15}
\end{aligned}
$$

Question

How to learn $P(\boldsymbol{X} = \boldsymbol{x})$?

A Naive Method

- We have M training examples/sentences
- For any sentence $\boldsymbol{x}$, $c(\boldsymbol{x})$ is the number of the sentence $\boldsymbol{x}$ in the training set
- A naive estimate

$$P(\boldsymbol{x}) = \frac{c(\boldsymbol{x})}{M} \tag{4}$$

Probabilistic Framework

Reconsider the probabilistic framework:

- A sequence of random variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_N\}$.
- Each random variable X_i can take any value in a finite set $\mathcal{V}$
- Our goal: compute

$$P(\boldsymbol{X} = \boldsymbol{x}) = P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) \quad (5)$$

Conditional Probability

$$
\begin{aligned}
P(X_1 = x_1, &X_2 = x_2, \ldots, X_n = x_n) \\
&= P(X_1 = x_1) \cdot P(X_2 = x_2 | X_1 = x_1) \cdots \\
&\quad P(X_n = x_n | \boldsymbol{X}_{1:n-1} = \boldsymbol{x}_{1:n-1}) \\
&= P(X_1 = x_1) \prod_{i=2}^{N} P(X_i = x_i | \boldsymbol{X}_{1:i-1} = \boldsymbol{x}_{1:i-1})
\end{aligned} \tag{6}
$$

Assumption

The probability of next word only depends a few preceding words

- `a student`
- `a student with a student`

First-order Markov Processes

The probability of X_i only depends on X_{i-1}:

$$P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) = P(X_i = x_i | X_{i-1} = x_{i-1}) \tag{7}$$

Example

$$P(\texttt{computer}|\texttt{a student with a}) = P(\texttt{student}|\texttt{a}) \tag{8}$$

First-order Markov Processes

The probability of X_i only depends on X_{i-1}:

$$P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) = P(X_i = x_i | X_{i-1} = x_{i-1}) \tag{7}$$

Example

$$P(\texttt{computer}|\texttt{a student with a}) = P(\texttt{student}|\texttt{a}) \tag{8}$$

Overall

$$\begin{aligned} P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) \\ = P(X_1 = x_1) \prod_{i=2}^{N} P(X_i = x_i | X_{i-1} = x_{i-1}) \end{aligned} \tag{9}$$

Questions

$$
\begin{aligned}
P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) \\
= P(X_1 = x_1) \prod_{i=2}^{N} P(X_i = x_i | X_{i-1} = x_{i-1})
\end{aligned} \tag{10}
$$

Two questions

- $P(X_1 = x_1)$

Questions

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N) = P(X_1 = x_1) \prod_{i=2}^{N} P(X_i = x_i | X_{i-1} = x_{i-1}) \tag{10}$$

Two questions

- $P(X_1 = x_1)$
- Compare the following two examples

`a student with confidence`

vs.

`a student with a`

The START Token

Examples

- `START a student`
- `START a student with a computer`

Now, $P(X_1 = x_1)$ becomes

$$P(X_1 = x_1 | X_0 = \texttt{START}) \tag{11}$$

Additional benefit: unified mathematical formulation

The STOP Token

Examples

- `START a student with confidence STOP`
- `START a student with a STOP`

Now, add one more to the conditional probability chain

$$P(\texttt{STOP}|X_N = x_N) \tag{12}$$

The STOP Token

Examples

- `START a student with confidence STOP`
- `START a student with a STOP`

Now, add one more to the conditional probability chain

$$P(\texttt{STOP} | X_N = x_N) \tag{12}$$

A way to handle variable length sentences.

Bi-gram Language Models: Example Sentence

$$
\begin{aligned}
P(\texttt{START a student STOP}) = & P(\texttt{a}|\texttt{START}) \\
& \cdot P(\texttt{student}|\texttt{a}) \\
& \cdot P(\texttt{STOP}|\texttt{student})
\end{aligned} \tag{13}
$$

Generic Framework

Bi-gram language model with $x_0 = \texttt{START}$ and $x_N = \texttt{STOP}$

$$P(X_1 = x_1, \ldots, X_N = x_n) = \prod_{i=1}^{N} P(X_i = x_i | X_{i-1} = x_{i-1}) \tag{14}$$

Parameter Estimation

Maximum Likelihood Estimate (MLE)

$$Q(X_i = x_i | X_{i-1} = x_{i-1}) = \frac{c(x_{i-1}, x_i)}{c(x_{i-1})} \tag{15}$$

Example

$$Q(\mathsf{a}|\mathsf{START}) = \frac{c(\mathsf{START}\ \ \mathsf{a})}{c(\mathsf{START})} \tag{16}$$

Probability Table

	$X_i = \texttt{a}$	$X_i = \texttt{man}$	$\cdots$	$X_i = \texttt{STOP}$
$X_{i-1} = \texttt{START}$				
$X_{i-1} = \texttt{a}$				
$X_{i-1} = \texttt{man}$				
$\vdots$				

- Vocab size $|\mathcal{V}| = 10^4$
- Number of parameters $|\mathcal{V}|^2 = 10^8 = 100\text{M}$

Uni-gram LMs

$$P(X_i = x_i) = \frac{c(x_i)}{T} \tag{17}$$

- T: total number of the tokens in the training set
- The simplest language model
- Parameters: $|\mathcal{V}|$

Uni-gram LMs (II)

- Pros
  - Easy to understand
  - Cheap to learn
- Cons
  - Bag-of-words assumption

$$P(\texttt{the the the}) \gg P(\texttt{I want coffee})$$

[Smith, 2018]

N-gram LMs

Language has long-distance dependencies

Example

`Bob gave Tina the burger, because she was ________`

1. `study`
2. `hungry`
3. `sleepy`
4. ···

Tri-gram LMs

$$P(X_i = x_i | \boldsymbol{X}_{i-1,i-2} = \boldsymbol{x}_{i-1,i-2}) = \frac{c(\boldsymbol{x}_{i,i-1,i-2})}{c(\boldsymbol{x}_{i-1,i-2})} \tag{18}$$

- More close to the "*real*" language model
- Widely used models for a long time
- Parameters: $|\mathscr{V}|^3 = 10^{12}$ if $|\mathscr{V}| = 10^4$ (about 50 billion pages on the indexed, searchable Web (Washington Post, 2015).)

Interpolation

$$\lambda_1 \cdot P(X_i) + \lambda_2 \cdot P(X_i|X_{i-1}) + \lambda_3 \cdot P(X_i|\boldsymbol{X}_{i-1,i-2}) \tag{19}$$

- $\lambda_1 + \lambda_2 + \lambda_3 = 1$
- $\{\lambda_j\}_{j=1}^{3}$ are estimated on a development data

Evaluation

Likelihood

- Test data: M sentences

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M$$

- Likelihood

$$\log \prod_{m=1}^{M} P(\boldsymbol{x}_m) = \sum_{m=1}^{M} \log P(\boldsymbol{x}_m)$$

- Factors
  - Number of tokens
  - No intuitive explanation

Perplexity

$$\text{Perplexity} = 2^{-\frac{1}{T}\sum_{m=1}^{M} \log P(\boldsymbol{x}_m)} \tag{20}$$

where T is the total number of words in the test data.

Special Case

- An impossible case

$$Q(X_i = x_i | X_{i-1} = x_{i-1}) = 1 \tag{21}$$

Special Case

- An impossible case

$$Q(X_i = x_i | X_{i-1} = x_{i-1}) = 1 \tag{21}$$

- Perplexity

$$\begin{aligned} \text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M} \log 1} \\ &= 2^0 \\ &= 1 \end{aligned} \tag{22}$$

Special Case (II)

- A trivial case

$$Q(X_i = x_i | X_{i-1} = x_{i-1}) = \frac{1}{|\mathscr{V}|} \tag{23}$$

Special Case (II)

- A trivial case

$$Q(X_i = x_i | X_{i-1} = x_{i-1}) = \frac{1}{|\mathcal{V}|} \tag{23}$$

- Perplexity

$$\begin{aligned}
\text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M} \log \frac{1}{|\mathcal{V}|}} \\
&= 2^{-\frac{1}{T}(T \cdot \log \frac{1}{|\mathcal{V}|})} \\
&= 2^{-\log \frac{1}{|\mathcal{V}|}} \\
&= |\mathcal{V}|
\end{aligned} \tag{24}$$

Typical Values of Perplexity

- $|\mathcal{V}| = 50K$
- A uni-gram model: Perplexity = 955
- A bi-gram model: Perplexity = 137
- A tri-gram model: Perplexity = 74

Lower is better

[Collins, 2017]

A Few Comments on Perplexity

- Perplexity is only an intermediate measure of performance
  - e.g., lower perplexity does not mean better translation (wrt BLEU score)

A Few Comments on Perplexity

- Perplexity is only an intermediate measure of performance
  - e.g., lower perplexity does not mean better translation (wrt BLEU score)
- Perplexity is not directly comparable even on the same test data
  - you need the exactly same input for comparison

Resources

Google N-gram

Google Books Ngram Viewer

Language	#Volumes	#Tokens
English	4,541,627	468,491,999,592
Spanish	854,649	83,967,471,303
French	792,118	102,174,681,393
German	657,991	64,784,628,286
Russian	591,310	67,137,666,353
Italian	305,763	40,288,810,817
Chinese	302,652	26,859,461,025
Hebrew	70,636	8,172,543,728

Table 1: Number of volumes and tokens for each language in our corpus. The total collection contains more than 6% of all books ever published.

[Lin et al., 2012]

KenLM

KenLM: Faster and Smaller Language Model Queries

Kenneth Heafield
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213 USA
heafield@cs.cmu.edu

https://github.com/kpu/kenlm

[Heafield, 2011]

Reference

Collins, M. (2017).
Natural language processing: Lecture notes.

Heafield, K. (2011).
Kenlm: Faster and smaller language model queries.
In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Jurafsky, D. (2018).
Language modeling.

Lin, Y., Michel, J.-B., Aiden, E. L., Orwant, J., Brockman, W., and Petrov, S. (2012).
Syntactic annotations for the google books ngram corpus.
In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174. Association for Computational Linguistics.

Smith, N. A. (2018).
Natural language processing: Lecture notes.