CS 6501 Natural Language Processing

Statistical Language Modeling

Yangfeng Ji

September 17, 2019

Department of Computer Science
University of Virginia

Overview

1. Building Better LR Models
2. Applications of Language Models
3. N-gram Language Models
4. Smoothing
5. Evaluation

Building Better LR Models

Problems with Large Feature Set

- Overlap among features

Example

{FAST, SLOW, SUPER, SUPER FAST, SUPER SLOW}

- Learning in high dimensional space — overfitting

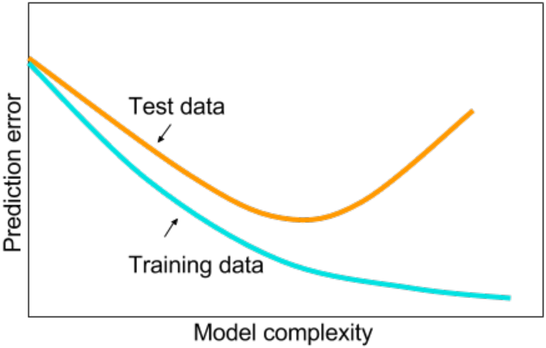
Example

62.4% on dev data

vs.

$\sim$ 100% on training data

Overfitting



[?]

L_2 Regularization

Add an additional constraint on $\{\boldsymbol{w}_y\}$

$$\ell_{L_2}(\{\boldsymbol{w}_y\}) = -\sum_{n=1}^{N} \log P(y^{(n)}|\boldsymbol{x}^{(n)}; \boldsymbol{w}) + \frac{\lambda}{2}\sum_{y} \|\boldsymbol{w}_y\|_2^2 \quad (1)$$

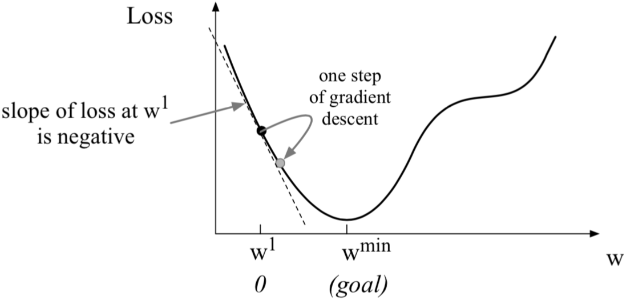
The gradient of the new loss function

$$\frac{\partial \ell_{L_2}(\{\boldsymbol{w}_y\})}{\partial \boldsymbol{w}_y} = -\sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{w}_y} \log P(y^{(n)}|\boldsymbol{x}^{(n)}; \boldsymbol{w}) + \lambda \boldsymbol{w}_y \quad (2)$$

[?, Sec. 2.4.1]

L_2 Regularization (Cont.)

L_2 Regularization introduces a trade-off between the likelihood function and the norm of $\{w_y\}$



Applications of Language Models

Machine Translation

A Statistical Approach to Machine Translation

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek,
John D. Lafferty, Robert L. Mercer, and Paul S. Roossin

IBM
Thomas J. Watson Research Center
Yorktown Heights, NY

In this paper, we present a statistical approach to machine translation. We describe the application of our approach to translation from French to English and give preliminary results.

$$P(f|e) = \frac{P(f)P(e|f)}{P(e)} \propto \underbrace{P(f)}_{\text{language model}} \cdot \underbrace{P(e|f)}_{\text{translation model}} \tag{3}$$

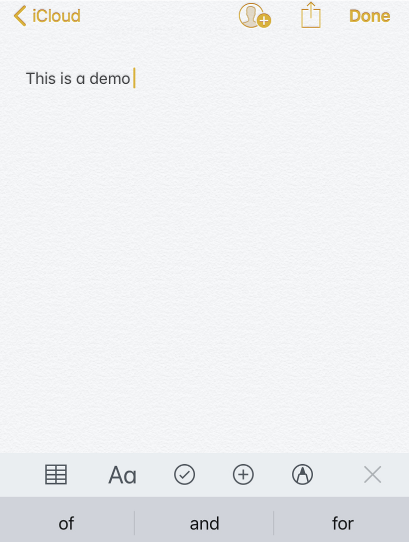
[Brown et al., 1990]

Speech Recognition

$$P(\text{I saw a van}) \gg P(\text{eyes awe of an}) \tag{4}$$

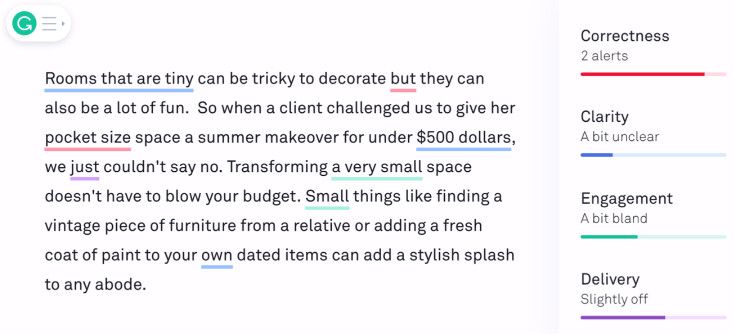
[Jurafsky and Martin, 2019]

Word Prediction in Input Methods



Writing Assistant

Grammarly:



Applications

- Discriminative tasks: evaluating the quality of texts
  - Speech recognition
  - Machine translation
  - Document summarization
  - ...
- Generative tasks: predicting the next word given a context
  - Word prediction
  - Text generation
  - ...

N-gram Language Models

Problem Definition

Given a vocab $\mathcal{V}$ that contains all the possible word types, then the prediction of x_n can be formulated as

$$P(x_n \mid x_1, \ldots, x_{n-1}) \tag{5}$$

Categorical distribution on $\mathcal{V}$

Joint Probability and Chain Rule

Without the independence assumption, any joint probability of two random variable can be decomposed as

$$\begin{aligned} P(X_1, X_2, \cdots, X_k) &= P(X_1)P(X_2, \cdots, X_k \mid X_1) \\ &= P(X_1)P(X_2 \mid X_1)P(X_3, \cdots, X_k \mid X_2, X_1) \\ &= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2, X_1) \cdots \\ &\quad P(X_k \mid X_1, \cdots, X_{k-1}) \end{aligned} \tag{6}$$

Parameter Estimation

Maximum likelihood estimation

$$p(x_n \mid x_1, \ldots, x_{n-1}) = \frac{\#(x_1, x_2, \ldots, x_n)}{\#(x_1, x_2, \ldots, x_{n-1})} \tag{7}$$

For example, with the sentence "`the dog barks`"

$$p(\text{barks} \mid \text{the dog}) = \frac{\#(\text{the dog barks})}{\#(\text{the dog})} \tag{8}$$

[Collins, 2017]

Challenge of Parameter Estimation

With the sentence "`the dog barks at the dumbwaiter where the thief is hiding`"

$$p(\text{hiding} \mid \text{the dog} \cdots \text{ is}) = \frac{\#(\text{the dog} \cdots \text{ is hiding})}{\#(\text{the dog} \cdots \text{ is})} \quad (9)$$

Google "the dog barks at the dumbwaiter where the thief is hiding"

All | Images | Videos | Shopping | News | More | Settings | Tools

1 result (0.41 seconds)

At Whom the Dog Barks | The New Yorker
https://www.newyorker.com › magazine › 1990/12/03 › at-whom-the-dog...
Dec 3, 1990 - Everybody is called together to meet with the police; **the dog barks at the dumbwaiter where the thief is hiding**, but she's ignored. Later Eliza ...

Google "the dog barks at the dumbwaiter where the thief is"

All | Images | Videos | Shopping | Maps | More | Settings | Tools

1 result (0.40 seconds)

At Whom the Dog Barks | The New Yorker
https://www.newyorker.com › magazine › 1990/12/03 › at-whom-the-dog...
Dec 3, 1990 - Everybody is called together to meet with the police; **the dog barks at the dumbwaiter where the thief is** hiding, but she's ignored. Later Eliza ...

Simplification: Uni-gram

Assume all words are independent with each other

$$P(x_n \mid x_1, \ldots, x_{n-1}) \approx P(x_n) \tag{10}$$

For example

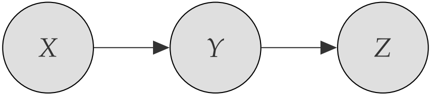
$$p(\text{barks} \mid \text{the dog}) \approx p(\text{barks}) \tag{11}$$

Comments

- It has extremely limited prediction power
- Number of parameters: $V = |\mathcal{V}|$

Markov Property

First-order Markov property: given



$$P(Z \mid X, Y) = P(Z \mid Y) \tag{12}$$

It simplifies the conditional probability

$$p(x_n \mid x_1, \ldots, x_{n-1}) \approx p(x_n \mid x_{n-1}) \tag{13}$$

and also the joint probability

$$\begin{aligned} p(x_1, \ldots, x_n) \quad &\approx \quad p(x_n \mid x_{n-1}) \cdot p(x_{n-1} \mid x_{n-2}) \cdots \\ & \quad\;\; p(x_2 \mid x_1) \cdot P(x_1) \end{aligned} \tag{14}$$

Bi-gram Models

$$
\begin{aligned}
p(x_1, \ldots, x_n) \quad &\approx \quad p(x_n \mid x_{n-1}) \cdot p(x_{n-1} \mid x_{n-2}) \cdots \\
& \quad\quad p(x_2 \mid x_1) \cdot P(x_1) \qquad (15)
\end{aligned}
$$

For example "`the dog barks`"

$$
\begin{aligned}
p(\text{the dog barks}) \quad &= \quad p(\text{the}) \cdot p(\text{dog} \mid \text{the}) \\
& \quad\quad p(\text{barks} \mid \text{dog})
\end{aligned}
$$

Special Tokens

$$
\begin{aligned}
p(\text{the dog barks}) \quad &= \quad p(\text{the}) \cdot p(\text{dog} \mid \text{the}) \\
&\quad\;\; p(\text{barks} \mid \text{dog})
\end{aligned}
$$

The model needs

- a special token ($\square$) to distinguish $p(\text{the})$ from the marginal distribution of word `the`
- another special token ($\blacksquare$) to indicate the end of a sentence

Factorization with special tokens:

$$
\begin{aligned}
p(\square \text{ the dog barks } \blacksquare) \quad &= \quad p(\text{the} \mid \square) \cdot p(\text{dog} \mid \text{the}) \\
&\quad\;\; p(\text{barks} \mid \text{dog}) \cdot p(\blacksquare \mid \text{barks})
\end{aligned}
$$

Example: Parameter Estimation

Example sentences

- `□ I am Sam ■`
- `□ Sam I am ■`
- `□ I do not like green eggs and ham ■`

Some of the probabilities:

$$p(\text{I} \mid \square) = \frac{2}{3} \quad p(\blacksquare \mid \text{Sam}) = \frac{1}{2} \quad p(\text{do} \mid \text{I}) = \frac{1}{3}$$

[Jurafsky and Martin, 2019]

Issues with a Fixed Vocabulary

- $p(x_n \mid x_{n-1})$ is defined a fixed vocabulary, for normalization purpose

$$p(x_n \mid x_{n-1}) = \frac{\#(x_{n-1}, x_n)}{\sum_{x' \in \mathcal{V}} \#(x_{n-1}, x')} \tag{16}$$

- Issues with a fixed vocabulary
  - Unknown words: word w_i is not in the vocabulary
  - Zero probability: word combination $w_i w_j$ never appears in the training set

Unknown Words

Replace all words that are not in the vocab with a special token UNK.

For example

- Original text: "`the dog barks at the dumbwaiter where the thief is hiding`"
- After preprocessing: "`the dog barks at the` UNK `where the thief is hiding`"

Question

Can we simply ignore the unknown words?

Smoothing

High-order Markov Models

A motivating example:

`The printer on the 5th floor of Rice hall crashed`

N-gram Language Models

- Uni-gram: $p(x_n)$
- Bi-gram: $p(x_n \mid x_{n-1})$
- Tri-gram: $p(x_n \mid x_{n-1}, x_{n-2})$
- 4-gram: $p(x_n \mid x_{n-1}, x_{n-2}, x_{n-3})$
- 5-gram: $p(x_n \mid x_{n-1}, x_{n-2}, x_{n-3}, x_{n-4})$

Discounting

It is the same method used in parameter estimation of naive Bayes classifiers

$$p(x_n \mid x_{n-1}) = \frac{\#(x_{n-1}, x_n) + \alpha}{\#(x_{n-1}) + \alpha V} \tag{17}$$

where $\alpha > 0$ is a hyper-parameter.

Linear Interpolation

Estimate the following three models with MLE:

- Uni-gram: $p(x_n)$
- Bi-gram: $p(x_n \mid x_{n-1})$
- Tri-gram: $p(x_n \mid x_{n-1}, x_{n-2})$

Then, the new probability of x_n given x_{n-2} and x_{n-1} is

$$\begin{aligned} p_{LI}(x_n \mid x_{n-1}, x_{n-2}) &= \lambda_1 p(x_n) + \lambda_2 p(x_n \mid x_{n-1}) \\ &\quad + \lambda_3 p(x_n \mid x_{n-1}, x_{n-2}) \end{aligned} \tag{18}$$

$\{\lambda_i\}$ are learned with a held-out corpus (a development set).

Evaluation

Sentence Evaluation (I)

Evaluation with joint probabilities

$$p(\text{I love black coffee}) \text{ vs. } p(\text{black coffee pleases me}) \quad (19)$$

Direct comparison between the probabilities will tell us which sentence is more *fluent*.

Sentence Evaluation (II)

Limitation of comparing joint probabilities directly

$$p(\text{I love black coffee}) \text{ vs. } p(\text{I like black coffee very much}) \tag{20}$$

Due to the *length difference*, the second probability may always be smaller than the first.

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

$$\text{Perplexity} = 2^{-\frac{1}{T}\sum_{m=1}^{M}\log p(\boldsymbol{x}_m)} \qquad (21)$$

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

Special Case

- An impossible case

$$p(x_n|x_{n-1}) = 1 \tag{22}$$

- Perplexity

$$\begin{aligned} \text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M}\log 1} \\ &= 2^{0} \\ &= 1 \end{aligned} \tag{23}$$

Special Case (II)

- A trivial case

$$p(x_n|x_{n-1}) = \frac{1}{|\mathcal{V}|} \tag{24}$$

- Perplexity

$$\begin{aligned}\text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M}\log\frac{1}{|\mathcal{V}|}} \\ &= 2^{-\frac{1}{T}(T\cdot\log\frac{1}{|\mathcal{V}|})} \\ &= 2^{-\log\frac{1}{|\mathcal{V}|}} \\ &= |\mathcal{V}|\end{aligned} \tag{25}$$

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 an intrinsic evaluation measurement
- is not necessarily correlated with the performance of
  - e.g., lower perplexity does not mean better translation (wrt BLEU score)
- is not directly comparable even on the same test data
  - you need the exactly same input for comparison

Reference

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990).
A statistical approach to machine translation.
Computational linguistics, 16(2):79–85.

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

Jurafsky, D. and Martin, J. (2019).
Speech and language processing.