S&DS 265 / 565

Introductory Machine Learning

Language Models

October 22

Yale

Welcome back!

For today:

- Where have we been? Where are we going?
- Language models

But first...

- Assignment 2, midterm grades are in the works
- Assignment 3 due this Thurs, October 24
- Assignment 4 out at same time (language models, word embeddings)

Invitation: Panel discussion

Class on Tuesday, December 3:

We will have a panel discussion on

Societal issues for AI and Machine Learning

If you would like to volunteer to participate, please email us:

```
To: sds265@yale.edu
Subject: iML panel
```

Participants will receive a thank you gift! :-)

Where we've been

Week	Dates	Topics	Demos & Tutorials	Lecture Slides	Readings and Notes	Assignments & Exams
1	Aug 31	Course overview		Thu: Course overview		
2	Sept 5, 7	Python and background concepts	CO Python elements CO Covid trends	Tue: Python elements Thu: Pandas and linear regression	Data8 Chapters 3, 4, 5	Quiz 1 CO Assn 1 out
3	Sept 12, 14	Linear regression and classification	CO Covid trends (revisited) CO Classification examples	Tue: Regression concepts Thu: Classification	ISL Sections 3.1, 3.2, 3.5 Notes on regression ISL Sections 4.3, 4.4 Notes on classification	
4	Sept 19, 21	Stochastic gradient descent	CO SGD examples	Tue: Classification (continued) Thu: Stochastic gradient descent	ISL Section 6.2.2 ISL Section 10.7.2	Assn 1 in CO Assn 2 out
5	Sept 26, 28	Bias and variance, cross-validation	CO Bias-variance tradeoff CO Covid trends (revisited) CO California housing	Tue: Bias and variance Thu: Cross-validation	ISL Section 2.2 ISL Section 5.1	Quiz 2
6	Oct 3, 5	Tree-based methods and principal components	CO Trees and forests Visualizing trees CO PCA examples	Tue: Trees and Forests Thu: PCA	ISL Sections 8.1, 8.2 ISL Section 12.2	Assn 2 in CO Assn 3 out
7	Oct 10, 12	PCA and dimension reduction	CO PCA revisited CO Used for dimension reduction CO Word embeddings	Tue: PCA and word embeddings Thu: Embeddings and review	ISL Section 12.2	Quiz 3
8	Oct 17	Midterm exam (in class)			On Canvas: Practice midterms / Sample solns Midterm / Sample soln	

Where we're going

			embeddings			
8	Oct 17	Midterm exam (in class)			On Canvas: Practice midterms / Sample solns Midterm / Sample soln	
9	Oct 24, 26	Language models, topic models	GPT-3 demo Bayesian inference Topic models	Tue: Language models Thu: Topic models	OpenAI: Better language models Notes on Bayesian inference	Assn 3 in Assn 4 out
10	Oct 31, Nov 2	Introduction to neural networks	Sanity check Minimal neural network Regression examples	Tue: Neural networks Thu: Neural networks	ISL Sections 10.1, 10.2	Quiz 4
11	Nov 7, 9	Reinforcement learning	Q-learning	Tue: Reinforcement learning Thu: Deep reinforcement learning		Assn 4 in Assn 5 out
12	Nov 14, 16	Deep neural networks	Tensorflow playground Autoencoder examples	Tue: Reinforcement learning Thu: Deep reinforcement learning	ISL Section 10.7 Notes on backpropagation	Quiz 5
13	Nov 21, 23	No class, Thanksgiving break				
14	Nov 28, 30	Transformers and ChatGPT	ChatGPT demo	Tue: Transformers Thu: Human feedback and rewards		Assn 5 in
15	Dec 5, 7	Societal issues for machine learning		Tue: Panel discussion Thu: Course wrap up		Quiz 6
16	Fri, Dec 15, 2pm, Room TBA	Final exam			Registrar: Final exam schedule Practice final	

For Today

- Language models
- Concepts...no new methods

Language models

- A language model is a way of *generating* any sequence of words

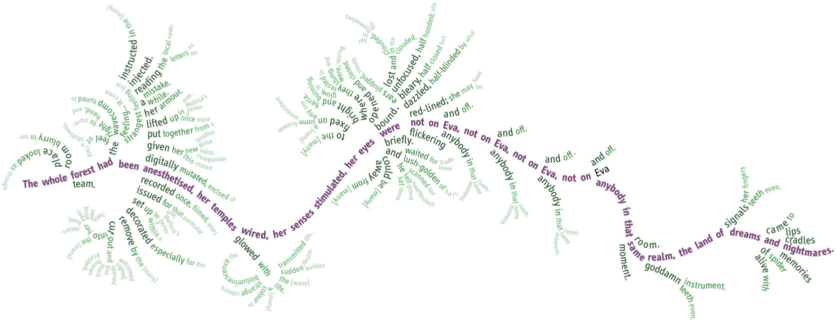
$$
\begin{aligned}
P(\text{“the whole forest had been anesthetized”}) =& \\
P(\text{“the”}) \times P(\text{“whole”} \mid \text{“the”})& \\
\times P(\text{“forest”} \mid \text{“the whole”})& \\
\times P(\text{“had”} \mid \text{“the whole forest”})& \\
\times P(\text{“been”} \mid \text{“the whole forest had”})& \\
\times P(\text{“anesthetized”} \mid \text{“the whole forest had been”})&
\end{aligned}
$$

Remixing Noon

Text generated from Channel Skin by Jeff Noon

"The whole forest had been anesthetised, her temples wired, her senses stimulated, her eyes were not on Eva, not on Eva, not on Eva, not on anybody in that same realm, the land of dreams and nightmares."

Viability: 0.000000326%



https://chatbotslife.com/notes-on-remixing-noon-generative-text-and-markov-chains-84ff4ec23937

Text generation

- Words generated one-by-one
- A word is chosen by sampling from a probability distribution
- Result is purely synthetic text—generative AI

Uses of language models

- Speech recognition
- Machine translation
- Text compression
- Texting
- Email completion
- Image captioning
- Mind reading from fMRI

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Dasher: LMs for assistive devices

Language models enable new modes of text input:

`https://www.youtube.com/watch?v=quw_Kci4fUg`

`https://youtu.be/QxFEUk3J89Q?t=72`

Dasher for text input, David Mackay, University of Cambridge

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

- The number of *histories* grows as V^{n-1}. Number of parameters in model grows as V^n, where V is number of words in vocabulary.

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \ldots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2 \mid w_1) \ldots p(w_n \mid w_1, \ldots, w_{n-1})$$

- The number of *histories* grows as V^{n-1}. Number of parameters in model grows as V^n, where V is number of words in vocabulary.
- What are some ways of reducing the number of parameters?

One approach: Grouping histories

- Let $g(w_1, \ldots, w_n)$ be the group assigned to the history

One approach: Grouping histories

- Let $g(w_1, \ldots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} \mid w_1, \ldots, w_n) = p(w_{n+1} \mid g(w_1, \ldots, w_n))$$

One approach: Grouping histories

- Let $g(w_1, \ldots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} \mid w_1, \ldots, w_n) = p(w_{n+1} \mid g(w_1, \ldots, w_n))$$

- Number of parameters: $O(V \cdot \text{number of groups})$

One approach: Grouping histories

- Let $g(w_1, \ldots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} \mid w_1, \ldots, w_n) = p(w_{n+1} \mid g(w_1, \ldots, w_n))$$

- Number of parameters: $O(V \cdot \text{number of groups})$
- What are some example groupings?

Grouping histories

- Unigrams: $g(w_1, \ldots, w_n) = \emptyset$.

The notation $O(\cdot)$ is called "Big Oh" and means "no greater than a constant times", so that $O(f(n))$ means a sequence that is bounded by $C \cdot f(n)$ for large enough n.

Grouping histories

- Unigrams: $g(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \ldots, w_n) = w_n$.

The notation $O(\cdot)$ is called “Big Oh” and means “no greater than a constant times”, so that $O(f(n))$ means a sequence that is bounded by $C \cdot f(n)$ for large enough n.

Grouping histories

- Unigrams: $g(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \ldots, w_n) = w_n$.
- Trigrams: $g(w_1, \ldots, w_n) = (w_{n-1}, w_n)$.

The notation $O(\cdot)$ is called "Big Oh" and means "no greater than a constant times", so that $O(f(n))$ means a sequence that is bounded by $C \cdot f(n)$ for large enough n.

Grouping histories

- Unigrams: $g(w_1, \ldots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \ldots, w_n) = w_n$.
- Trigrams: $g(w_1, \ldots, w_n) = (w_{n-1}, w_n)$.
- Number of parameters grows as $O(V)$, $O(V^2)$, and $O(V^3)$, respectively.

The notation $O(\cdot)$ is called "Big Oh" and means "no greater than a constant times", so that $O(f(n))$ means a sequence that is bounded by $C \cdot f(n)$ for large enough n.

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- What are some problems with this model?

Sparse data problem

The next group of slides presents one way of quantifying the problem of sparse data in language modeling

Half Earth

Aug 06, 2018 · by Foundation Staff · 0 Comment · Google Earth, Half-Earth Project, Map of Life

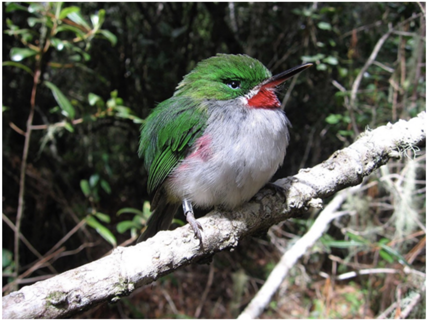
By Jeremy Malczyk, Michelle Duong, Ajay Ranipeta, Chris Heltne, Walter Jetz of Map of Life, Yale University, and the E.O. Wilson Biodiversity Foundation Half-Earth Project

This article originally in *Medium*, July 30, 2018

half-earth project

Learn how you can be part of bringing Half-Earth to life.

The Significance of Biodiversity



Narrow-billed Tody, Todus angustirostris. Photo by Julie Hart.

https://eowilsonfoundation.org/mapping-species-for-half-earth/

https://www.half-earthproject.org/

Specious species

- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).

Specious species

- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.

Specious species

- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.
- What is the chance I'll find a new species?

Specious species

- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.
- What is the chance I'll find a new species?
- What if Wilson observes 100 unique species?

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).

https://en.wikipedia.org/wiki/Good-Turing_frequency_estimation

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).
- Good-Turing: Estimate of the probability that the next animal is a new species? $\widehat{p}_{GT} = 100/100,000 = 10^{-3}$.

https://en.wikipedia.org/wiki/Good-Turing_frequency_estimation

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).
- Good-Turing: Estimate of the probability that the next animal is a new species? $\widehat{p}_{GT} = 100/100,000 = 10^{-3}$.
- This is an estimate of the missing probability mass.

https://en.wikipedia.org/wiki/Good-Turing_frequency_estimation

Sparse data: Species $\approx$ words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.

Sparse data: Species $\approx$ words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)

Sparse data: Species $\approx$ words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)
- This means that the maximum likelihood estimate (MLE) is zero, and the probability that my model predicts the next word will be zero.

Sparse data: Species $\approx$ words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)
- This means that the maximum likelihood estimate (MLE) is zero, and the probability that my model predicts the next word will be zero.
- The MLE is supported on the observed data. We need to spread out the probability over unseen events.

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\widehat{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- How can the model be strengthened?

Smoothing and Interpolation

Smoothing:

$$\widetilde{p}(w_3 \mid w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3) + \varepsilon}{\text{count}(w_1, w_2) + V\varepsilon}$$

Smoothing and Interpolation

Linear interpolation:

$$p(w_3 \mid w_1, w_2) = \lambda_3 \widehat{p}(w_3 \mid w_1, w_2) + \lambda_2 \widehat{p}(w_3 \mid w_2) + \lambda_1 \widehat{p}(w_3)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

This is a type of “mixture model”

Modern language models

The algorithm assigns a "score" to possible next words v:

$$s(v; \overbrace{w_1, \ldots, w_n}^{\text{previous words}})$$
$$\uparrow$$
$$\text{possible next word}$$

The score is given by

$$s(v; w_1, \ldots, w_n) = \beta_v^T \varphi(w_1, \ldots, w_n)$$

where $\varphi(w_1, \ldots, w_n)$ is an encoding of the context into a big vector, and β_v is an "embedding" of word v.

Converted into a probability with softmax

Distributed representations

So, the groups are types of embeddings.

How good is a language model?

The next group of slides presents a useful way of quantifying how good a language model is

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3}\left(\frac{1}{4}+4+8\right)=4.08\overline{3}$$

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3}\left(\frac{1}{4}+4+8\right)=4.08\overline{3}$$

The *geometric mean* of 1/4, 4, 8 is

$$\sqrt[3]{\frac{1}{4}\cdot 4\cdot 8}=2$$

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3}\left(\frac{1}{4}+4+8\right)=4.08\overline{3}$$

The *geometric mean* of 1/4, 4, 8 is

$$\sqrt[3]{\frac{1}{4}\cdot 4\cdot 8}=2$$

The geometric mean is no greater than the arithmetic mean

Recall: Geometric mean

The *geometric mean* of $x_1, \ldots, x_n$ is

$$\sqrt[n]{x_1 x_2 \cdots x_n} = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}}$$

Recall: Geometric mean

The *geometric mean* of $x_1, \ldots, x_n$ is

$$\sqrt[n]{x_1 x_2 \cdots x_n} = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}}$$

How good is a language model? Perplexity

Perplexity is defined as

$$\text{Perplexity}(\theta) = \left(\prod_{i=1}^{N} p_\theta(w_i \mid w_{1:i-1}) \right)^{-\frac{1}{N}}$$

where $w_1, w_2, \ldots, w_N$ is a large chunk of text that wasn't used to train the language model.

How good is a language model? Perplexity

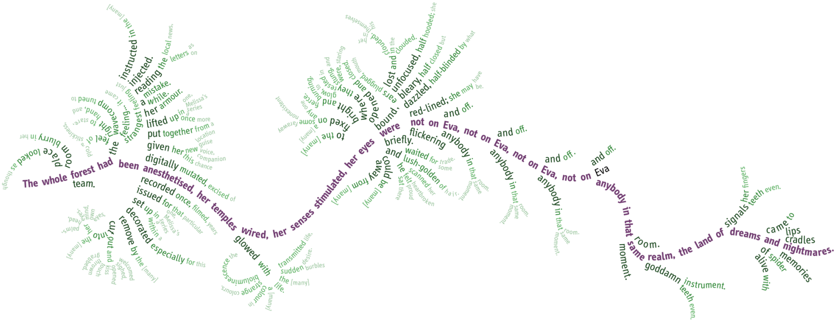
- Perplexity is the inverse of the geometric mean of the word probabilities
- Intuitively, if the perplexity is 100, the actual next word is in the top 100 words of the model (on average).
- This is the (geometric) average “branching factor” for the model on real text

Remixing Noon

Text generated from Channel Skin by Jeff Noon

"The whole forest had been anesthetised, her temples wired, her senses stimulated, her eyes were not on Eva, not on Eva, not on Eva, not on anybody in that same realm, the land of dreams and nightmares."

Viability: 0.000000326%



https://revdancatt.com/2017/03/01/markov_noon

Modern language models

Suppose a computer program assigns a "score" to possible next words:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

Suppose a computer program assigns a “score” to possible next words:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

Can convert this to a language model by the “softmax” operation:

$$p(w \mid w_1, \ldots, w_n) = \frac{\exp(s(w; w_1, \ldots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \ldots, w_n))}$$

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

Suppose a computer program assigns a "score" to possible next words:

$$s(v;\ \underbrace{w_1, \ldots, w_n}_{\text{word history}}\ )$$

Can convert this to a language model by the "softmax" operation:

$$p(w \mid w_1, \ldots, w_n) = \frac{\exp(s(w; w_1, \ldots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \ldots, w_n))}$$

In ChatGPT, the function $s(v; w_{1:n})$ is learned on large amounts of text (unsupervised) using a type of deep neural network called a *transformer*.

---

https://en.wikipedia.org/wiki/Softmax_function

Modern language models

In fact, the score is computed as

$$s(v; \underbrace{w_1, \ldots, w_n}_{\text{word history}}) = \beta_v^T g(w_1, \ldots, w_n)$$

- the "grouping function" $g(w_1, \ldots, w_n)$ is like an embedding, that maps the word history to a high dimensional vector
- the weights β_v are essentially parameters in a big logistic regression

We'll see how this is done when we discuss neural networks

Summary of today: Language models

- A language model is used to predict or generate the next word
- Used many different applications
- Probabilities need to be “smoothed” to avoid zeros
- Perplexity is a measure of a language model’s predictive power
- ChatGPT and descendents represent frontier of AI