Log-linear language models & final projects

CS 585, Fall 2019
Introduction to Natural Language Processing
http://people.cs.umass.edu/~miyyer/cs585/

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

[slides adapted from Mike Collins & Jordan Boyd-Graber]

questions from last class….

- HW0?
- final projects?
- latex?

let’s talk about projects!

Timeline

- Group information: due Sep 19
- Project proposal: 2-4 pages, due Oct 3
- Project milestone 1: 2 pages, due Oct 24
- Project milestone 2: 2 pages, due Nov 21
- Poster presentations: end of Dec
- Final project report: due Dec 19

Group information

- either 4 or 5 people per group, no exceptions
- if you want to form your own group, you must do so before Sep 19 and submit the names/IDs of group members to this google form: https://forms.gle/NeGzmzL15M2jAMBeA
- if you do not submit this form, we will randomly assign you to a group on Sep 19

Other deliverables

- We will provide Overleaf templates for the proposal, milestones, and final report.
  - You MUST use these templates, no exceptions
- We'll provide feedback after every deliverable
- Please feel free to see any of us in office hours to discuss your projects; all of the TAs are very familiar with NLP research!

Project

- Either *build* natural language processing systems, or *apply* them for some task.
- Use or develop a dataset. Report empirical results or analyses with it.
- Different possible areas of focus
  - Implementation & development of algorithms
  - Defining a new task or applying a linguistic formalism
  - Exploring a dataset or task

Formulating a proposal

- What is the ***research question***?
- What's been done before?
- What experiments will you do?
- How will you know whether it worked?
  - If data: held-out accuracy
  - If no data: manual evaluation of system output.
    Or, annotate new data

The Heilmeier Catechism

- What are you trying to do? Articulate your objectives using absolutely no jargon.
- How is it done today, and what are the limits of current practice?
- What is new in your approach and why do you think it will be successful?
- Who cares? If you are successful, what difference will it make?
- What are the risks?
- How much will it cost?
- How long will it take?
- What are the mid-term and final “exams” to check for success?

https://en.wikipedia.org/wiki/George_H._Heilmeier#Heilmeier.27s_Catechism

An example proposal

- Introduction / problem statement
- Motivation (why should we care? why is this problem interesting?)
- Literature review (what has prev. been done?)
- Possible datasets
- Evaluation
- Tools and resources
- Project milestones / tentative schedule

NLP Research

- All the best publications in NLP are open access!
  - Conference proceedings: ACL, EMNLP, NAACL (EACL, LREC...)
  - Journals: TACL, CL
  - "aclweb": ACL Anthology-hosted papers http://aclweb.org/anthology/
  - NLP-related work appears in other journals/conferences too: data mining (KDD), machine learning (ICML, NIPS), AI (AAAI), information retrieval (SIGIR, CIKM), social sciences (Text as Data), etc.
- Reading tips
  - Google Scholar
    - Find papers
    - See paper's number of citations (imperfect but useful correlate of paper quality) and what later papers cite it
    - [... or SemanticScholar...]
  - For topic X: search e.g. [[nlp X]], [[aclweb X]], [[acl X]], [[X research]]...
  - Authors' webpages
    find researchers who are good at writing and whose work you like
  - Misc. NLP research reading tips:
    http://idibon.com/top-nlp-conferences-journals/

Some common tasks….

- Detection tasks
  - Sentiment detection
  - Sarcasm and humor detection
  - Emoticon detection / learning
- Structured linguistic prediction
  - Targeted sentiment analysis (i liked __ but hated __)
  - Relation, event extraction (who did what to whom)
  - Narrative chain extraction
  - Parsing (syntax, semantics, discourse...)
- Text generation tasks
  - Machine translation
  - Document summarization
  - Poetry / lyrics generation (e.g. recent work on hip-hop lyrics)
  - Text normalization (e.g. translate online/Twitter text to standardized English)
- End to end systems
  - Question answering
  - Conversational dialogue systems (hard to eval?)
- Predict external things from text
  - Movie revenues based on movie reviews ... or online buzz? http://www.cs.cmu.edu/~ark/movie$-data/
- Visualization and exploration (harder to evaluate)
  - Temporal analysis of events, show on timeline
  - Topic models: cluster and explore documents
- Figure out a task with a cool dataset
  - e.g. Urban Dictionary

More specific examples:

- SemEval 2020 shared tasks
  http://alt.qcri.org/semeval2020/index.php?id=tasks

- Analyzing and Interpreting Neural Networks for NLP
  BlackboxNLP: https://www.aclweb.org/anthology/W19-48

- Stanford CS224N projects
  https://web.stanford.edu/class/cs224n/project.html

- NYU DS-GA 1012 example projects
  https://docs.google.com/document/d/1ET7o4P31XggTnZTIWnx4eqsjers9FD8o73LcJskn03M/edit

Some ideas from the TAs…

- Analyzing generations produced by large-scale pre-trained language models
  - grammaticality, meaning, coherence, cohesion, relevance, etc.
  - differences between human-written and machine-generated text
  - can they write a novel, book, short story, poem, or essay?

- Analyzing the knowledge encoded in large-scale pre-trained language models
  - linguistic / relational / factual knowledge
  - can they answer math questions?
  - gender and other demographic biases

- Dissecting neural architectures, e.g., Transformers
  - where linguistic information is captured?
  - can attention mechanisms provide meaningful “explanations” for the model outputs?

- Improving neural network training and making them more scalable:
  - linguistically-informed Transformers for faster NLP
  - data selection, data augmentation
  - model compression

- Improving language modeling pretraining
  - augmented with additional tasks (multi-task learning)
  - additional supervision, e.g. syntax-sensitive dependencies
  - more general or better suited for certain downstream tasks
  - can we control for style, content, task-specific behavior, etc?

Sources of data

- All projects must use (or make, and use) a textual dataset. Many possibilities.
  - For some projects, creating the dataset may be a large portion of the work; for others, just download and more work on the system/modeling side

- SemEval and CoNLL Shared Tasks:
  dozens of datasets/tasks with labeled NLP annotations
  - Sentiment, NER, Coreference, Textual Similarity, Syntactic Parsing, Discourse Parsing, and many other things...
  - e.g. SemEval 2015 ... CoNLL Shared Task 2015 ...
  - https://en.wikipedia.org/wiki/SemEval (many per year)
  - http://ifarm.nl/signll/conll/ (one per year)

- General text data (not necessarily task specific)
  - Books (e.g. Project Gutenberg)
  - Reviews (e.g. Yelp Academic Dataset https://www.yelp.com/academic_dataset)
  - Web
  - Tweets

Tools

- Tagging, parsing, NER, coref, ...
  - Stanford CoreNLP http://nlp.stanford.edu/software/corenlp.shtml
  - spaCy (English-only, no coref) http://spacy.io/
  - Twitter-specific tools (ARK, GATE)
- Many other tools and resources
  tools ... word segmentation ... morph analyzers ...
  resources ... pronunciation dictionaries ... wordnet, word embeddings, word clusters ...
- Long list of NLP resources
  https://medium.com/@joshdotai/a-curated-list-of-speech-and-natural-language-processing-resources-4d89f94c032a
- Deep learning? Try out AllenNLP, PyTorch, Tensorflow (https://allennlp.org, https://pytorch.org/, https://www.tensorflow.org/)

Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical ____

Trigram Models

- Estimate a distribution $P(w_i|w_1, w_2, \ldots w_{i-1})$ given previous "history" $w_1, \ldots, w_{i-1} =$

Third, the notion "grammatical in English" cannot be identified in any way with the notion "high order of statistical approximation to English". It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

- **Trigram estimates:**

$$\begin{aligned} P(\text{model}|w_1, \ldots w_{i-1}) \quad = \quad & \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\ & \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\ & \lambda_3 P_{ML}(\text{model}) \end{aligned}$$

backoff!

where $\lambda_i \geq 0, \sum_i \lambda_i = 1, P_{ML}(y|x) = \frac{Count(x,y)}{Count(x)}$

Trigram Models

- Estimate a distribution $P(w_i|w_1, w_2, \ldots w_{i-1})$ given previous "history" $w_1, \ldots, w_{i-1} =$

Third, the notion "grammatical in English" cannot be identified in any way with the notion "high order of statistical approximation to English". It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

- **Trigram estimates:**

$$P(\text{model}|w_1, \ldots w_{i-1}) = \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \lambda_3 P_{ML}(\text{model})$$

what else can we condition this prediction on?

backoff!

where $\lambda_i \geq 0, \sum_i \lambda_i = 1, P_{ML}(y|x) = \frac{Count(x,y)}{Count(x)}$

Trigram Models

$$
\begin{aligned}
P(\text{model}|w_1, \ldots w_{i-1}) \quad = \quad & \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\
& \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\
& \lambda_3 P_{ML}(\text{model})
\end{aligned}
$$

- Makes use of only bigram, trigram, unigram estimates
- Many other "features" of $w_1, \ldots, w_{i-1}$ may be useful, e.g.,:

$P_{ML}(\text{model} \mid w_{i-2} = \text{any})$

$P_{ML}(\text{model} \mid w_{i-1}$ is an adjective$)$

$P_{ML}(\text{model} \mid w_{i-1}$ ends in "ical"$)$

$P_{ML}(\text{model} \mid author = \text{Chomsky})$

$P_{ML}(\text{model} \mid$ "model" does not occur somewhere in $w_1, \ldots w_{i-1})$

$P_{ML}(\text{model} \mid$ "grammatical" occurs somewhere in $w_1, \ldots w_{i-1})$

Trigram Models

$$P(\text{model}|w_1, \ldots w_{i-1}) = \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \lambda_3 P_{ML}(\text{model})$$

- Makes use of only bigram, trigram, unigram estimates
- Many other "features" of $w_1, \ldots, w_{i-1}$ may be useful, e.g.,:

$P_{ML}(\text{model} \mid w_{i-2} = \text{any})$

$P_{ML}(\text{model} \mid w_{i-1}$ is an adjective$)$

$P_{ML}(\text{model} \mid w_{i-1}$ ends in "ical"$)$

$P_{ML}(\text{model} \mid author = \text{Chomsky})$

$P_{ML}(\text{model} \mid$ "model" does not occur somewhere in $w_1, \ldots w_{i-1})$

$P_{ML}(\text{model} \mid$ "grammatical" occurs somewhere in $w_1, \ldots w_{i-1})$

how can we estimate the parameters of these different language models?

Trigram Models

$$
\begin{aligned}
P(\text{model}|w_1, \ldots w_{i-1}) \quad = \quad & \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\
& \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\
& \lambda_3 P_{ML}(\text{model})
\end{aligned}
$$

- Makes use of only bigram, trigram, unigram estimates
- Many other "features" of $w_1, \ldots, w_{i-1}$ may be useful, e.g.,:

$P_{ML}(\text{model} \mid w_{i-2} = \text{any})$

$P_{ML}(\text{model} \mid w_{i-1}$ is an adjective$)$

$P_{ML}(\text{model} \mid w_{i-1}$ ends in "ical"$)$

$P_{ML}(\text{model} \mid author = \text{Chomsky})$

$P_{ML}(\text{model} \mid$ "model" does not occur somewhere in $w_1, \ldots w_{i-1})$

$P_{ML}(\text{model} \mid$ "grammatical" occurs somewhere in $w_1, \ldots w_{i-1})$

how can we use all of this information in a single language model?

A Naive Approach

$$
\begin{aligned}
&P(\text{model}|w_1, \ldots w_{i-1}) = \\
&\lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\
&\lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\
&\lambda_3 P_{ML}(\text{model}) + \\
&\lambda_4 P_{ML}(\text{model}|w_{i-2} = \text{any}) + \\
&\lambda_5 P_{ML}(\text{model}|w_{i-1} \text{ is an adjective}) + \\
&\lambda_6 P_{ML}(\text{model}|w_{i-1} \text{ ends in “ical”}) + \\
&\lambda_7 P_{ML}(\text{model}|author = \text{Chomsky}) + \\
&\lambda_8 P_{ML}(\text{model}|\text{“model” does not occur somewhere in } w_1, \ldots w_{i-1}) + \\
&\lambda_9 P_{ML}(\text{model}|\text{“grammatical” occurs somewhere in } w_1, \ldots w_{i-1})
\end{aligned}
$$

This quickly becomes very unwieldy...

The General Problem

- We have some **input domain** $\mathcal{X}$

- Have a finite **label set** $\mathcal{Y}$

- Aim is to provide a **conditional probability** $P(y \mid x)$ for any x, y where $x \in \mathcal{X}, y \in \mathcal{Y}$

Language Modeling

- x is a "history" $w_1, w_2, \ldots w_{i-1}$, e.g.,

  Third, the notion "grammatical in English" cannot be identified in any way with the notion "high order of statistical approximation to English". It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

- y is an "outcome" w_i

Feature Vector Representations

- Aim is to provide a conditional probability $P(y \mid x)$ for "decision" y given "history" x

A feature is some function $\phi(x)$; in LMs $\phi(context)$.

Features are often binary *indicators*; i.e. $\phi(x) \in \{0,1\}$

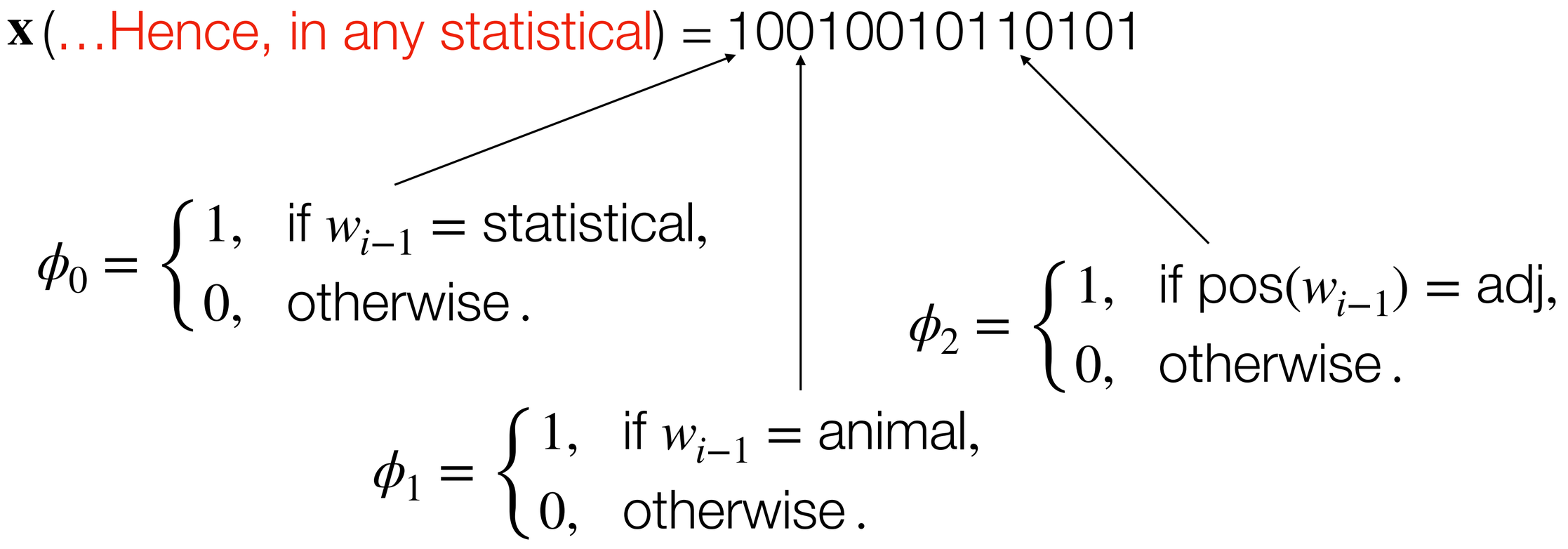
If you have *m* features, you can form a **feature vector** $\mathbf{x} \in \mathbb{R}^{\mathbf{m}}$

what could be some useful indicator features for language modeling?

We can come up with practically any features for *context*, *word* pairs

In the general case, we compute features for input *x*, output *y* pairs.

We can form *feature vectors* out of these features that look like:



We can come up with practically any features for *context*, *word* pairs

In the general case, we compute features for input *x*, output *y* pairs.

We can form *feature vectors* out of these features that look like:

$\mathbf{x}$(…Hence, in any statistical) = 100100101101101

$$\phi_0 = \begin{cases} 1, & \text{if } w_{i-1} = \text{statistical}, \\ 0, & \text{otherwise}. \end{cases}$$

$$\phi_1 = \begin{cases} 1, & \text{if } w_{i-1} = \text{animal}, \\ 0, & \text{otherwise}. \end{cases}$$

$$\phi_2 = \begin{cases} 1, & \text{if pos}(w_{i-1}) = \text{adj}, \\ 0, & \text{otherwise}. \end{cases}$$

do we have to manually specify all of these features?

Defining Features in Practice

- We had the following "trigram" feature:

$$\phi_3(x) = \begin{cases} 1, & \text{if } w_{i-2} = \text{any}, w_{i-1} = \text{statistical} \\ 0, & \text{otherwise}. \end{cases}$$

We can generalize this to any combination of two context words

$$\phi_{N(u,v)}(x) = \begin{cases} 1, & \text{if } w_{i-2} = u, w_{i-1} = v \\ 0, & \text{otherwise}. \end{cases}$$

where $N(u, v)$ is a function that maps each (u, v) to a different integer

given features **x**, how do we predict the next word *y*?

$$s = Wx + b$$

what is dimensionality of *s*?

features $x \in \mathbb{R}^m$

weight matrix $W \in \mathbb{R}^{|V| \times m}$

given features **x**, how do we predict the next word *y*?

$$s = Wx + b$$

score vector $s \in \mathbb{R}^{|V|}$

weight matrix $W \in \mathbb{R}^{|V| \times m}$

features $x \in \mathbb{R}^{m}$

each row of W contains weights for a (word *y*, **x**) pair

example on paper

how do we obtain probabilities?

$$s = Wx + b$$

score vector $s \in \mathbb{R}^{|V|}$

features $x \in \mathbb{R}^m$

weight matrix $W \in \mathbb{R}^{|V| \times m}$

$$p_i = \frac{e^{s_i}}{\sum_j e^{s_j}}; p = \text{softmax}(s)$$

Softmax function

Define: "goodness score" for potential output y.

$$s = Wx + b$$

Log-linear distribution then is

$$p(y \mid x, W, b) = \frac{e^{s_y}}{\sum_{y' \in V} e^{s_{y'}}}$$

Softmax function: turns goodness scores into probabilities that sum to 1. Exponentiate then normalize.

$$\text{softmax}(\{s_1 \ldots s_{|V|}\}) \rightarrow \left(\frac{\exp(s_1)}{\sum_{y' \in V} \exp(s_{y'})}, \frac{\exp(s_2)}{\sum_{y' \in V} \exp(s_{y'})}, \ldots, \frac{\exp(s_{|V|})}{\sum_{y' \in V} \exp(s_{y'})} \right)$$

```
def softmax(scores):
    exponentiated = [exp(s) for s in scores]
    Z = sum(exponentiated)
    return [escore/Z for escore in exponentiated]
```

“Log-linear” ?

$$p(y \mid x, W) = \frac{e^{W_y x}}{\sum_{y' \in V} e^{W_{y'} x}}$$

$$\log p(y \mid x, W) = W_y x - \log \sum_{y' \in V} e^{W_{y'} x}$$

“Log-linear” ?

$$p(y \mid x, W) = \frac{e^{W_y x}}{\sum_{y' \in V} e^{W_{y'} x}}$$

$$\log p(y \mid x, W) = W_y x - \log \sum_{y' \in V} e^{W_{y'} x}$$

linear in weights and features…

… except for this!
known as log-sum-exp,
very important for these models

“Log-linear” ?

$$\log p(y \mid x, W) \propto W_y x$$

why is this true?

$$p(y \mid x, W) = \frac{e^{W_y x}}{\sum_{y' \in V} e^{W_{y'} x}}$$

$$\log p(y \mid x, W) = W_y x - \log \sum_{y' \in V} e^{W_{y'} x}$$

linear in weights and features…

… except for this!
known as log-sum-exp,
very important for these models

python demo!

next time: how do we learn the weights W?

next next time: how do we learn the features x?