sequence modeling:
Viterbi algorithm

CS 585, Fall 2018

Introduction to Natural Language Processing

http://people.cs.umass.edu/~miyyer/cs585/

Mohit Iyyer

College of Information and Computer Sciences

University of Massachusetts Amherst

some slides from Jordan Boyd-Graber

questions from last time…

- audio on recorded lectures????????? **idk, communicating w/ the echo360 ppl now**
- could you go slower / repeat important concepts that aren't on the slides? **yes, sorry!**
- is it possible to do the project alone? in a team of 3, one person ends up doing nothing eventually! **unfortunately not, but we will evaluate team members separately**

POS Tagging

- Input: Plays well with others
- Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS
- Output: Plays/VBZ well/RB with/IN others/NNS

Penn Treebank POS tags

Hidden Markov Models

- We have an input sentence $x = x_1, x_2, \ldots, x_n$
  (x_i is the i'th word in the sentence)

- We have a tag sequence $y = y_1, y_2, \ldots, y_n$
  (y_i is the i'th tag in the sentence)

- We'll use an HMM to define

$$p(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$$

  for any sentence $x_1 \ldots x_n$ and tag sequence $y_1 \ldots y_n$ of the same length.

- Then the most likely tag sequence for x is

$$\arg\max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1, y_2, \ldots, y_n)$$

HMM Definition

Assume K parts of speech, a lexicon size of V, a series of observations $\{x_1, \ldots, x_N\}$, and a series of unobserved states $\{z_1, \ldots, z_N\}$.

- π A distribution over start states (vector of length K):
  $\pi_i = p(z_1 = i)$
- θ Transition matrix (matrix of size K by K):
  $\theta_{i,j} = p(z_n = j | z_{n-1} = i)$
- β An emission matrix (matrix of size K by V):
  $\beta_{j,w} = p(x_n = w | z_n = j)$

Two problems: How do we move from data to a model? (Estimation) How do we move from a model and unlabled data to labeled data? (Inference)

today: inference!

probability of a tag sequence

$$P(\mathsf{x} = x_1, x_2, \cdots, x_n, \mathsf{z} = z_1, z_2, \cdots, z_n) =$$

$$P(z_1) * P(x_1 \mid z_1) * p(z_2 \mid z_1) * p(x_2 \mid z_2) * \cdots$$

initial prob π emission prob β transition prob θ

let's quickly review *estimation* before continuing….

Reminder: How do we estimate a probability?

- For a multinomial distribution (i.e. a discrete distribution, like over words):

$$\theta_i = \frac{n_i + \alpha_i}{\sum_k n_k + \alpha_k} \tag{1}$$

- α_i is called a smoothing factor, a pseudocount, etc.

just like in naive Bayes, we'll be counting to estimate these probabilities!

Training Sentences

x = tokens
z = POS tags

x	here	come	old	flattop
z	MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	V

and	I	love	her
CONJ	PRO	V	PRO

Initial Probability π

POS	Frequency	Probability
MOD	1.1	0.234
DET	1.1	0.234
CONJ	1.1	0.234
N	0.1	0.021
PREP	0.1	0.021
PRO	0.1	0.021
V	1.1	0.234

let's use add-alpha smoothing with alpha = 0.1

Training Sentences

here	come	old	flattop
MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	N

and	I	love	her
CONJ	PRO	V	PRO

Training Sentences

here	come	old	flattop
MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	N

and	I	love	her
CONJ	PRO	V	PRO

Training Sentences

here	come	old	flattop
MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	N

and	I	love	her
CONJ	PRO	V	PRO

Transition Probability θ

- We can ignore the words; just look at the parts of speech. Let's compute one row, the row for verbs.
- We see the following transitions: V $\rightarrow$ MOD, V $\rightarrow$ CONJ, V $\rightarrow$ V, V $\rightarrow$ PRO, and V $\rightarrow$ PRO

POS	Frequency	Probability
MOD	1.1	0.193
DET	0.1	0.018
CONJ	1.1	0.193
N	0.1	0.018
PREP	0.1	0.018
PRO	2.1	0.368
V	1.1	0.193

Training Sentences

here	come	old	flattop
MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	N

and	I	love	her
CONJ	PRO	V	PRO

Training Sentences

here	come	old	flattop
MOD	V	MOD	N

a	crowd	of	people	stopped	and	stared
DET	N	PREP	N	V	CONJ	V

gotta	get	you	into	my	life
V	V	PRO	PREP	PRO	N

and	I	love	her
CONJ	PRO	V	PRO

Emission Probability β

Let's look at verbs . . .

Word	a	and	come	crowd	flattop
Frequency	0.1	0.1	1.1	0.1	0.1
Probability	0.0125	0.0125	0.1375	0.0125	0.0125

Word	get	gotta	her	here	i
Frequency	1.1	1.1	0.1	0.1	0.1
Probability	0.1375	0.1375	0.0125	0.0125	0.0125

Word	into	it	life	love	my
Frequency	0.1	0.1	0.1	1.1	0.1
Probability	0.0125	0.0125	0.0125	0.1375	0.0125

Word	of	old	people	stared	stopped
Frequency	0.1	0.1	0.1	1.1	1.1
Probability	0.0125	0.0125	0.0125	0.1375	0.1375

now… given that we’ve estimated an HMM, how do we use it to get POS tags for unlabeled data?

Viterbi Algorithm

- Given an unobserved sequence of length L, $\{x_1, \ldots, x_L\}$, we want to find a sequence $\{z_1 \ldots z_L\}$ with the highest probability.

how many different possible tag sequences exist?

Viterbi Algorithm

- Given an unobserved sequence of length L, $\{x_1, \ldots, x_L\}$, we want to find a sequence $\{z_1 \ldots z_L\}$ with the highest probability.
- It's impossible to compute K^L possibilities.
- So, we use dynamic programming to compute most likely tags for each token subsequence from 0 to t that ends in state k.
- Memoization: fill a table of solutions of sub-problems
- Solve larger problems by composing sub-solutions
- Base case:

$$\delta_1(k) = \pi_k \beta_{k,x_i} \tag{1}$$

- Recursion:

$$\delta_n(k) = \max_j \left(\delta_{n-1}(j)\theta_{j,k}\right)\beta_{k,x_n} \tag{2}$$

Viterbi Algorithm

- Given an unobserved sequence of length L, $\{x_1, \dots, x_L\}$, we want to find a sequence $\{z_1 \dots z_L\}$ with the highest probability.
- It's impossible to compute K^L possibilities.
- So, we use dynamic programming to compute most likely tags for each token subsequence from 0 to t that ends in state k.
- Memoization: fill a table of solutions of sub-problems
- Solve larger problems by composing sub-solutions
- Base case:

$$\delta_1(k) = \pi_k \beta_{k,x_i} \tag{1}$$

- Recursion:

$$\delta_n(k) = \max_{j} \left(\delta_{n-1}(j)\theta_{j,k}\right)\beta_{k,x_n} \tag{2}$$

Viterbi Algorithm

- Given an unobserved sequence of length L, $\{x_1, \ldots, x_L\}$, we want to find a sequence $\{z_1 \ldots z_L\}$ with the highest probability.
- It's impossible to compute K^L possibilities.
- So, we use dynamic programming to compute most likely tags for each token subsequence from 0 to t that ends in state k.
- Memoization: fill a table of solutions of sub-problems
- Solve larger problems by composing sub-solutions
- Base case:

$$\delta_1(k) = \pi_k \beta_{k,x_i} \tag{1}$$

- Recursion:

$$\delta_n(k) = \max_j \left(\delta_{n-1}(j)\theta_{j,k}\right)\beta_{k,x_n} \tag{2}$$

Viterbi Algorithm

- Given an unobserved sequence of length L, $\{x_1, \ldots, x_L\}$, we want to find a sequence $\{z_1 \ldots z_L\}$ with the highest probability.
- It's impossible to compute K^L possibilities.
- So, we use dynamic programming to compute most likely tags for each token subsequence from 0 to t that ends in state k.
- Memoization: fill a table of solutions of sub-problems
- Solve larger problems by composing sub-solutions
- Base case:

$$\delta_1(k) = \pi_k \beta_{k,x_i} \tag{1}$$

- Recursion:

$$\delta_n(k) = \max_j \left(\delta_{n-1}(j)\theta_{j,k}\right) \beta_{k,x_n} \tag{2}$$

what is the complexity of this algorithm?

$$K^2L$$

need to keep backpointers!

- But just computing the max isn't enough. We also have to remember where we came from. (Breadcrumbs from best previous state.)

$$\Psi_n = \text{argmax}_j \delta_{n-1}(j)\theta_{j,k} \tag{3}$$

let's do an example for the sentence **come and get it**

POS	π_k	β_{k,x_1}	$\log \delta_1(k) = \log(\pi_k \beta_{k,x_1})$
MOD	0.234	0.024	-5.18
DET	0.234	0.032	-4.89
CONJ	0.234	0.024	-5.18
N	0.021	0.016	-7.99
PREP	0.021	0.024	-7.59
PRO	0.021	0.016	-7.99
V	0.234	0.121	-3.56

come and get it

Why logarithms?

1. More interpretable than a float with lots of zeros.
2. Underflow is less of an issue
3. Addition is cheaper than multiplication

$$log(ab) = log(a) + log(b) \tag{4}$$

POS	$\log \delta_1(j)$		$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		
N	-7.99		
PREP	-7.59		
PRO	-7.99		
V	-3.56		

come **and** get it

POS	$\log \delta_1(j)$		$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		???
N	-7.99		
PREP	-7.59		
PRO	-7.99		
V	-3.56		

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		???
N	-7.99		
PREP	-7.59		
PRO	-7.99		
V	-3.56		

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		???
N	-7.99		
PREP	-7.59		
PRO	-7.99		
V	-3.56		

come **and** get it

$$\log\left(\delta_0(\text{V})\theta_{\text{V, CONJ}}\right) = \log \delta_0(k) + \log \theta_{\text{V, CONJ}} = -3.56 + -1.65$$

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		???
N	-7.99		
PREP	-7.59		
PRO	-7.99		
V	-3.56	-5.21	

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18		
DET	-4.89		
CONJ	-5.18		???
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18	-8.48	
DET	-4.89	-7.72	
CONJ	-5.18	-8.47	???
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\mathrm{CONJ}}$	$\log \delta_2(\mathrm{CONJ})$
MOD	-5.18	-8.48	
DET	-4.89	-7.72	
CONJ	-5.18	-8.47	???
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\mathrm{CONJ}}$	$\log \delta_2(\mathrm{CONJ})$
MOD	-5.18	-8.48	
DET	-4.89	-7.72	
CONJ	-5.18	-8.47	
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

$\log \delta_1(k) = -5.21 - \log \beta_{\mathrm{CONJ,\ and}} =$

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18	-8.48	
DET	-4.89	-7.72	
CONJ	-5.18	-8.47	
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

$$\log \delta_1(k) = -5.21 - \log \beta_{\text{CONJ, and}} = -5.21 - 0.64$$

POS	$\log \delta_1(j)$	$\log \delta_1(j)\theta_{j,\text{CONJ}}$	$\log \delta_2(\text{CONJ})$
MOD	-5.18	-8.48	
DET	-4.89	-7.72	
CONJ	-5.18	-8.47	-6.02
N	-7.99	≤ -7.99	
PREP	-7.59	≤ -7.59	
PRO	-7.99	≤ -7.99	
V	-3.56	-5.21	

come **and** get it

Viterbi Algorithm

backpointer!

POS	$\delta_1(k)$	$\delta_2(k)$	b_2	$\delta_3(k)$	b_3	$\delta_4(k)$	b_4
MOD	-5.18						
DET	-4.89						
CONJ	-5.18	-6.02	V				
N	-7.99						
PREP	-7.59						
PRO	-7.99						
V	-3.56						
WORD	come	and		get		it	

Viterbi Algorithm

POS	$\delta_1(k)$	$\delta_2(k)$	b_2	$\delta_3(k)$	b_3	$\delta_4(k)$	b_4
MOD	-5.18	-0.00	X				
DET	-4.89	-0.00	X				
CONJ	-5.18	-6.02	V				
N	-7.99	-0.00	X				
PREP	-7.59	-0.00	X				
PRO	-7.99	-0.00	X				
V	-3.56	-0.00	X				
WORD	come	and		get		it	

POS	$\delta_1(k)$	$\delta_2(k)$	b_2	$\delta_3(k)$	b_3	$\delta_4(k)$	b_4
MOD	-5.18	-0.00	X	-0.00	X		
DET	-4.89	-0.00	X	-0.00	X		
CONJ	-5.18	-6.02	V	-0.00	X		
N	-7.99	-0.00	X	-0.00	X		
PREP	-7.59	-0.00	X	-0.00	X		
PRO	-7.99	-0.00	X	-0.00	X		
V	-3.56	-0.00	X	-9.03	CONJ		
WORD	come	and		get		it	

POS	$\delta_1(k)$	$\delta_2(k)$	b_2	$\delta_3(k)$	b_3	$\delta_4(k)$	b_4
MOD	-5.18	-0.00	X	-0.00	X	-0.00	X
DET	-4.89	-0.00	X	-0.00	X	-0.00	X
CONJ	-5.18	-6.02	V	-0.00	X	-0.00	X
N	-7.99	-0.00	X	-0.00	X	-0.00	X
PREP	-7.59	-0.00	X	-0.00	X	-0.00	X
PRO	-7.99	-0.00	X	-0.00	X	-14.6	V
V	-3.56	-0.00	X	-9.03	CONJ	-0.00	X
WORD	come	and		get		it	

POS	$\delta_1(k)$	$\delta_2(k)$	b_2	$\delta_3(k)$	b_3	$\delta_4(k)$	b_4
MOD	-5.18	-0.00	X	-0.00	X	-0.00	X
DET	-4.89	-0.00	X	-0.00	X	-0.00	X
CONJ	-5.18	-6.02	V	-0.00	X	-0.00	X
N	-7.99	-0.00	X	-0.00	X	-0.00	X
PREP	-7.59	-0.00	X	-0.00	X	-0.00	X
PRO	-7.99	-0.00	X	-0.00	X	-14.6	V
V	-3.56	-0.00	X	-9.03	CONJ	-0.00	X
WORD	come	and		get		it	

most probable POS seq: V CONJ V PRO

let's talk about projects!

Timeline

- Project proposal: 2-4 pages, due Oct 19
- Progress report: 4-6 pages, due Nov 16
- Poster presentations: near end of classes
- Final report: 12+ pages, due Dec 20

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/

A few examples

- 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

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/)