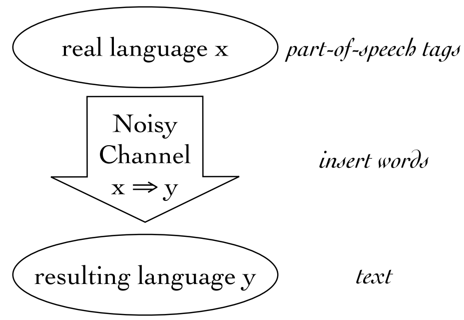
CS 181: Natural Language Processing

Lecture 8: Hidden Markov Models

Kim Bruce
Pomona College
Spring 2008

Disclaimer: Slide contents borrowed from many sources on web!

HMM Approach



Try to recover x from y

Predicting weather

- Jason Eisner of Johns Hopkins kept a careful diary of how many ice cream cones he ate every day.
- Based on the diary, and his long term records of ice cream eating, we would like to determine the weather, based on the number of cones he ate.

Predicting Weather from Ice Cream

	p(…|C)	p(…|H)	p(…|START)
p(1|…)	0.7	0.1	
p(2|…)	0.2	0.2	
p(3|…)	0.1	0.7	
p(C|…)	0.8	0.1	0.5
p(H|…)	0.1	0.8	0.5
p(STOP|…)	0.1	0.1	0

Predictions

ice creams

weather		2	3	3	1	1
	H	0.1	0.056	0.03136	0.0025088	0.000200704
	C	0.1	0.008	0.00064	0.0021952	0.001229312

```
v[X,t+1] = MAX(v[H,t]*P(X|H),V[C,t]*P(X|C))*P(n|X)
for X = H or C
```

Spread sheet: icecreamPredWeather.xls

Obtained by unrolling FST

Drawbacks

- Bigrams not as accurate, go with trigrams
- Sparse data!
- Back up to bigram or unigram if fails
- Can also train to find best linear combination.
- *Same ideas work with speech recognition*
  - *speech ⇒ text*

Transformation-Based Tagging

PoS Taggers

- Rule-Based Tagger - English Two Level Analysis ✔ *Done last time*
- Stochastic Tagger: Hidden Markov Model
  - ✔ *Done*
- Transformation-based Tagger

Also Called Brill

- Like rule-based to specify tags,
- ... but learn rules from tagged training corpus
- Assign tags by successive approximation
- Input:
  - Tagged corpus
  - Lexicon w/associated tags (also from corpus)

Brill Tagging

- Set most probable tag for each word as starting value (use morphology to help)
- Change tags according to rules of type
  - If word-1 is a X and word is a Y then change tag to Z.
  - Keep to fixed neighborhood (3) of word whose tag is being changed.
- Sample rules from templates:
  - Change NN to VB if prev tag is TO
  - Change VBP to VB if one of prev 3 tags is MD

Brill Tagging

- Train on tagged corpus
  - Write rule templates
  - Examine all possible rules following templates.
  - Select one w/ best improvement over entire corpus
  - Re-tag according to rule
  - Continue until insufficient improvement
- Save ordered set of rules.
- Early rules make errors -- corrected by later rules.

Problems

- Brill slower than HMM
  - Solution: compile to FST
- How to deal with new words?
  - Assume are nouns
  - Assume distributed like words occurring once in training set
  - Use morphological information (e.g. end w/ “ed”) to tag.

Evaluation

- Start w/ hand-coded "Gold Standard".
  - 97% agreement by humans, but 100% if allowed to discuss.
  - Baseline tagger (unigram most-likely tag) 90%
  - Most algorithms ~ 97%

Evaluating Systems

- Recall: # of answers got right divided by number of possible right answers
  - *Measures completeness in extraction of info*
- Precision: # of answers got right divided by number of answers attempted
  - *Measures accuracy of answers*

Factors Affecting Performance

- Amount of training data available
- Tag set
- Difference between training and test corpus
- Dictionary
- Unknown words

Hidden Markov & Maximum Entropy Models

HMM

- Compute likelihood:
  - Given HMM, $\lambda = (A,B)$, and observation sequence, O, determine $P(O \mid \lambda)$
- Decoding: ✔
  - Given HMM, $\lambda = (A,B)$, and observation sequence, O, determine best hidden state sequence.
- Learning:
  - Given an observation sequence, O, and set of states of HMM, learn (A,B)

Review Assumptions

- Limited horizon:
  - $P(x_{t+1} \mid x_1,\ldots,x_t) = P(x_{t+1} \mid x_t)$
- Time invariant:
  - $P(x_{t+1} \mid x_t) = P(x_2 \mid x_1)$
- State (part of speech) generates a word:
  - $(o_t \mid x_1,\ldots,x_t, o_1,\ldots,o_{t-1}) = P(o_t \mid x_t)$
- All only approximately true!

Compute Likelihood

- What is likelihood of observation sequence $o_1, \ldots, o_n$ given model λ? $P(O|\lambda)$
- If knew the hidden states, Q, easy:
  - $P(O|Q) = \prod_i P(o_i \mid q_i)$
- What is prob. of outcome & states?
  - $\prod_i P(o_i \mid q_i) * p(q_i \mid q_{i-1})$
    $= \prod_i P(o_i \mid q_i) * \prod_i p(q_i \mid q_{i-1})$
- $P(O) = \sum_Q P(O,Q) = \sum_Q P(O|Q)P(Q)$

Computationally Hard

- If N states and input of length T, then N^T possible state sequences!
- Use dynamic programming-like approach
- Table of size N by T. Like before, but add up probabilities rather than taking max!

HMM

- Compute likelihood: ✔
  - Given HMM, $\lambda = (A,B)$, and observation sequence, O, determine $P(O \mid \lambda)$
- Decoding: ✔
  - Given HMM, $\lambda = (A,B)$, and observation sequence, O, determine best hidden state sequence.
- Learning: *Skip, at least for now!*
  - Given an observation sequence, O, and set of states of HMM, learn (A,B)

Context-Free Grammars

Motivation

- Chunks of sentences behave as units
- Want to recover from input.
- Reason: Chunks are basis of meaning

Word Categories

noun	names of things	boy, dog, truth
verb	action or state	become, hit
pronoun	used for noun	I, you, we, she
adverb	modify V, Adj, Adv	sadly, very
adjective	modify N	happy, clever
conjunction	joins things	and, but, while
preposition	relation of N	to, from, into
Interjection	an outcry	ouch, oh, alas

Part of Speech

- Substitution test
  - All items of class should be freely substitutable for each other (at least in terms of grammar)
  - The {red, soft, prickly, small} pillow ...

Constituency

- Constituent: A group of words that behaves as a single unit or phrase.
- Sample noun phrases:
  - the big dog
  - the election that took place Tuesday
  - a fifth of Scotch
  - Mary
  - you

Constituency

- Can help determine meaning.
- *I hit the man with the cleaver*
  - I hit [the man with the cleaver]
  - I hit [the man] with a cleaver
- *You could not go to class tomorrow*
  - You [could not] go to class tomorrow
  - You could [not go] to class tomorrow

Constituent Phrases

- Name phrases based on word that *heads* the constituent.
  - the girl from Ipanema: NP: head is "girl"
  - very red: AP: head is "red"
  - by the dock: PP: head is "by"
  - scored a basket: VP: head is "scored"
- Words are smallest constituents, then phrases (N vs NP)

Evidence for Constituency

- Appear in similar environments (*substitutable*)
- Can move constituent as a whole, but not its components.
  - Joe threw snowballs in the winter
  - In the winter, Joe threw snowballs
  - *but not:* The winter, Joe threw snowballs in.

Grammars

- Context-free grammars model constituency
  - Also called phrase-structure, BNF
- Formally goes back to Chomsky (and Backus and Naur, independently), but something like it first suggested by Wundt in 1890.

Formal Def of CFG

- G = <T, N, S, R>, where
  - T is a set of *terminals* (lexicon)
  - N is a set of *non-terminals*. In linguistics, often also identify $P \subseteq N$, *preterminals*, which always rewrite as terminals.
  - $S \in N$ is *start state*.
  - R is set of rules of form $X \rightarrow \gamma$, where X is non-terminal and γ is sequence composed of terminals and non-terminals.
- $L(G) = \{w \in T^* \mid S \rightarrow^* w\}$

Example CFG

- T = {this, that, a, the, man, book, flight, meal, include, read, does}
- N = {S, NP, NOM, VP, Det, Noun, Verb, Aux}
- S - start
- R =

S → NP VP	VP → Verb
S → Aux NP VP	VP → Verb NP
S → VP	Det → that | this | a | the
NP → Det NOM	Noun → book | flight | meal | man
NOM → Noun	Verb → book | include | read
NOM → Noun NOM	Aux → does

Any Questions?