Hidden Markov Models

Introduction to
Artificial Intelligence
COS302
Michael L. Littman
Fall 2001

Administration

Exams need a swift kick.
Form project groups next week.
Project due on the last day.
***BUT*! There will be milestones.**
In 2 weeks, synonyms via web.
3 weeks, synonyms via wordnet.

Shannon Game Again

Recall
Sue swallowed the large green __.
Ok: pepper, frog, pea, pill
Not ok: beige, running, very
Parts of speech could help:
noun verbed det adj adj noun

POS Language Model

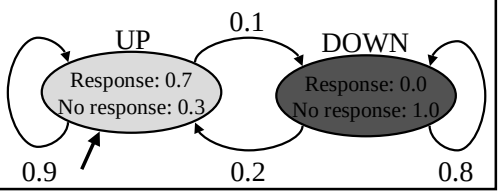
How could we define a probabilistic model over sequences of parts of speech?
How could we use this to define a model over word sequences?

Hidden Markov Model

Idea: We have states with a simple transition rule (Markov model). We observe a probabilistic function of the states. Therefore, the states are not what we see…

HMM Example

Browsing the web, connection is either up or down. Observe response or no.



HMM Definition

N states, M observations

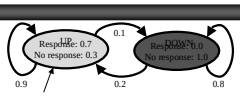
$\pi(s)$: prob. starting state is s

p(s,s'): prob. of s to s' transition

b(s, k): probability of obs k from s

$k_0\ k_1 \ldots k_l$: observation sequence

HMM Problems



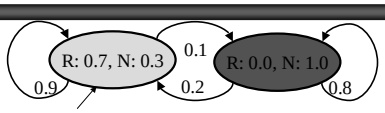
Probability of a state sequence

Probability of an obs. sequence

Most like state sequence for an observation sequence

Most likely model given obs. seq.

States Example



Probability of:

U U D D U U U

is

1.0 0.9 0.1 0.8 0.2 0.9 0.9 = .01166

Derivation

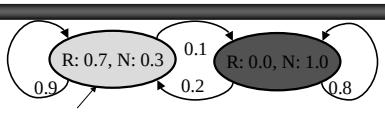
$Pr(s_0 \ldots s_l)$

$= Pr(s_0)\ Pr(s_1 \mid s_0)\ Pr(s_2 \mid s_0\ s_1) \ldots Pr(s_l \mid s_0\ s_1\ s_2 \ldots s_{l-1})$

$= Pr(s_0)\ Pr(s_1 \mid s_0)\ Pr(s_2 \mid s_1) \ldots Pr(s_l \mid s_{l-1})$

$= \pi(s_0)\ p(s_0, s_1)\ p(s_1, s_2) \ldots p(s_{l-1}, s_l)$

States/Obs. Example



Probability of:

R R N N N R N

U U D D U U U

given

is

0.7 0.7 1.0 1.0 0.3 0.7 0.3 = .0308700

Derivation

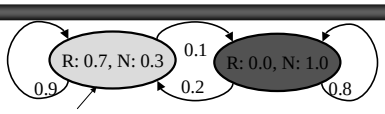
$Pr(k_0 \ldots k_l \mid s_0 \ldots s_l)$

$= Pr(k_0 \mid s_0 \ldots s_l)\ Pr(k_1 \mid s_0 \ldots s_l, k_0)\ Pr(k_2 \mid s_0 \ldots s_l, k_0, k_1) \ldots Pr(k_l \mid s_0 \ldots s_l, k_0 \ldots k_{l-1})$

$= Pr(k_0 \mid s_0)\ Pr(k_1 \mid s_1)\ Pr(k_2 \mid s_2) \ldots Pr(k_l \mid s_l)$

$= b(s_0, k_0)\ b(s_1, k_1)\ b(s_2, k_2) \ldots b(s_l, k_l)$

Observation Example



Probability of:

R R N N N R N

= sum{$s_0 \ldots s_6$} Pr($s_0 \ldots s_6$)
Pr(R R N N N R N | $s_0 \ldots s_6$)

Derivation

Pr($k_0 \ldots k_l$)
= sum{$s_0 \ldots s_l$} Pr($s_0 \ldots s_6$) Pr($k_0 \ldots k_l$ |$s_0 \ldots s_l$)
= sum{$s_0 \ldots s_l$} Pr(s_0) Pr(k_0|s_0) Pr(s_1 | s_0)
Pr(k_1 | s_1) Pr(s_2 | s_1) Pr(k_2 | s_2) …
Pr(s_l | s_{l-1}) Pr(k_l | s_l)
= sum{$s_0 \ldots s_l$} $\pi(s_0)$ b(s_0,k_0) p(s_0, s_1)
b(s_1,k_1) p(s_1, s_2) b(s_2,k_2) … p(s_{l-1}, s_l)
b(s_l,k_l)

Uh Oh, Better Get α

Grows exponentially with l: M^l

How do this more efficiently?

α(i,t): probability of seeing first t observations *and* ending up in state i: Pr($k_0 \ldots k_t$, s_t=i)

α(i,0) = $\pi(s_0)$ b(k_0, s_i)

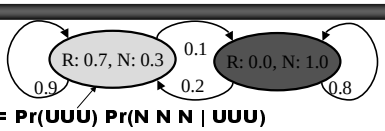
α(i,t) = sum$_j$ α(j,t-1) p(s_j,s_i) b(k_t, s_i)

Return Pr($k_0 \ldots k_l$) = sum$_j$ α(j,l)

Partial Derivation

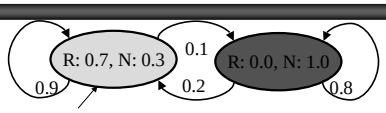
α(i,t)
= Pr($k_0 \ldots k_t$, s_t=i)
= sum$_j$ Pr($k_0 \ldots k_{t-1}$, s_{t-1}=j)
Pr($k_0 \ldots k_t$, s_t=i | $k_0 \ldots k_{t-1}$, s_{t-1}=j)
= sum$_j$ α(j,t-1) Pr($k_0 \ldots k_{t-1}$ | $k_0 \ldots k_{t-1}$, s_{t-1}=j)
Pr(s_t=i, k_t | $k_0 \ldots k_{t-1}$, s_{t-1}=j)
= sum$_j$ α(j,t-1) Pr(s_t=i | $k_0 \ldots k_{t-1}$, s_{t-1}=j)
Pr(k_t | s_t=i, $k_0 \ldots k_{t-1}$, s_{t-1}=j)
= sum$_j$ α(j,t-1) Pr(s_t=i | s_{t-1}=j) Pr(k_t | s_t=i)
= sum$_j$ α(j,t-1) p(s_j,s_i) b(k_t, s_i)

Pr(N N N)



= Pr(UUU) Pr(N N N | UUU)
+ Pr(UUD) Pr(N N N | UUD)
+ Pr(UDU) Pr(N N N | UDU)
+ Pr(UDD) Pr(N N N | UDD)
+ Pr(DUU) Pr(N N N | DUU)
+ Pr(DUD) Pr(N N N | DUD)
+ Pr(DDU) Pr(N N N | DDU)
+ Pr(DDD) Pr(N N N | DDD)

Pr(N N N)



α	N	N	N
UP			
DOWN			

POS Tagging

- Simple tagging model says part of speech depends only on previous part of speech, word depends only on part of speech.
- So, HMM state is previous part of speech, observation is word.
- What are the probabilities and where could they come from?

Shannon Game

- If we have a POS-based language model, how do we compute probabilities?
- Pr(Sue ate the small blue candy.)
- = sum_{tags} Pr(sent|tags)

Best Tag Sequence

- Useful problem to solve:
- Given sentence, find most likely sequence of POS tags

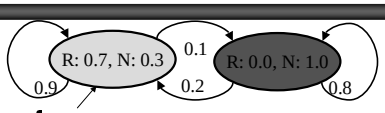
Sue	saw	the	fly	.
noun	verb	det	noun	.

- $\max_{\text{tags}}$ Pr(tags|sent)
- = $\max_{\text{tags}}$ Pr(sent|tags) Pr(tags) / Pr(sent)
- = c $\max_{\text{tags}}$ Pr(sent|tags) Pr(tags)

Viterbi

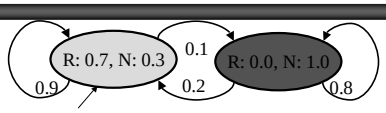
- $\delta(i,t)$: probability of *most likely* state sequence that ends in state i when seeing first t observations:
- $\max\{s_0 \ldots s_{t-1}\}$ Pr($k_0 \ldots k_t, s_0 \ldots s_{t-1}, s_t=i$)
- $\delta(i,0) = \pi(s_0)\ b(k_0, s_i)$
- $\delta(i,t) = \max_j \delta(j,t\text{-}1)\ p(s_j,s_i)\ b(k_t, s_i)$
- Return $\max_j \delta(j,l)$ (trace it back)

max Pr(s*) Pr(NNN|s*)



max { Pr(UUU) Pr(N N N | UUU),
Pr(UUD) Pr(N N N | UUD),
Pr(UDU) Pr(N N N | UDU),
Pr(UDD) Pr(N N N | UDD),
Pr(DUU) Pr(N N N | DUU),
Pr(DUD) Pr(N N N | DUD),
Pr(DDU) Pr(N N N | DDU),
Pr(DDD) Pr(N N N | DDD) }

max Pr(s*) Pr(NNN|s*)



δ	N	N	N
UP			
DOWN			

PCFG

- An analogy…
- Finite-state Automata (FSAs):
- Hidden Markov Models (HMMs) ::
- Context-free Grammars (CFGs):
- Probabilistic context-free grammars (PCFGs)

Grammars & Recursion

- S → 1.0 NP VP
- NP → 0.45 NP PP, 0.55 N
- PP → 1.0 P NP
- VP → 0.3 VP PP, 0.7 V NP
- N → 0.4 mice, 0.25 hats, 0.35 pimples
- P → 1.0 with
- V → 1.0 wore

Computing with PCFGs

- Can compute Pr(sent) and also $\max_{tree}$ Pr(tree|sent) using algorithms like the ones we discussed for HMMs. Still polynomial time.

What to Learn

- HMM Definition
- Computing probability of state and observation sequences
- Part of Speech Tagging
- Viterbi: most likely state sequence
- PCFG Definition

Homework 7 (due 11/21)

1. Give a maximization scheme for filling in the two blanks in a sentence like "I hate it when ___ goes ___ like that." Be somewhat rigorous to make the TA's job easier.
2. Derive that (a) $Pr(k_0 \ldots k_l) = \text{sum}_j\ \alpha(j,l)$, and (b) $\alpha(i,0) = \pi(s_0)\ b(k_0, s_i)$.
3. Email me your project group of three or four.

Homework 8 (due 11/28)

1. Write a program that decides if a pair of words are synonyms using the web. I'll send you the list, you send me the answers.
2. more soon

Homework 9 (due 12/5)

1. **Write a program that decides if a pair of words are synonyms using wordnet. I'll send you the list, you send me the answers.**
2. **more soon**