CS371N Lecture 15
HMMs

Announcements

- A4 due in a week
- Midterm next Thurs (9 days)
- OPTIONAL: independent final project proposals due after midterm

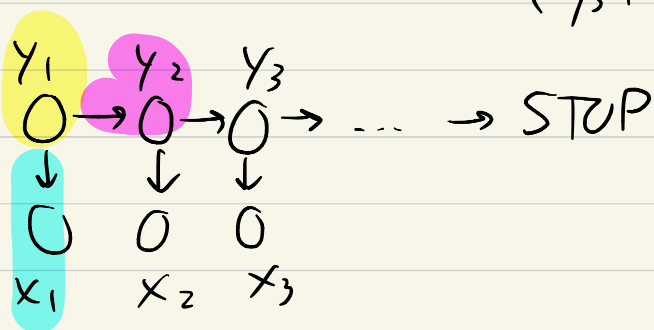
Recap Part-of-speech tagging

N	N	N	N
V	V	V	V
Fed	raises	interest	rates

Sequence models can help us predict a coherent tag sequence (discriminative: $P(\bar{y} | \bar{x})$)

HMMs Generative model of sequences

$$P(\bar{y}, \bar{x}) = P(y_1)\, P(x_1 | y_1)\, P(y_2 | y_1)\, P(x_2 | y_2)$$
$$P(y_3 | y_2) \ldots P(STOP | y_n)$$



Parameters: Initial : prob dist $P(y)$ over tags

Transitions : prob dist $P(y | y_{prev})$ over next tags

Emissions : prob dist $P(x | y)$ word | tag

N: dist over all words

V: dist over all words

<u>Ex</u> $\mathcal{T} = \{N, V, \text{STOP}\}$

$\mathcal{V} = \{\text{they}, \text{can}, \text{fish}\}$

Initial $P(y)$:

1.0	N
0	V
0	STOP

Transitions:

y_{prev} \ y	N	V	STOP
N	1/5	3/5	1/5
V	1/5	1/5	3/5

Emissions:

	they	fish	can
N	1	0	0
V	0	1/2	1/2

① Prob of $\begin{pmatrix} N & V & V \\ \text{they} & \text{can} & \text{fish} \end{pmatrix}$

$$P_{init}(N)\, P_e(\text{they} \mid N)\, P_t(V \mid N)\, P_e(\text{can} \mid V)$$
$$P_t(V \mid V)\, P_e(\text{fish} \mid V)\, P_t(\text{STOP} \mid V)$$

t

$1.0 \cdot 3/5 \cdot 1/5 \cdot 3/5$

$\cdot \quad \cdot \quad \cdot$

e $1.0 \quad 1/2 \quad 1/2$

multiply all these ⇒.

(2) Is there a higher-scoring tag sequence for "they can fish"?

No.

Goal of HMMs:

HMMs model $P(\bar{y}, \bar{x})$

They are not good generative models of text

What we use them for: $P(\bar{y}|\bar{x}) \propto P(\bar{y}, \bar{x})$

Compute $P(\bar{y}|\bar{x})$

$$P(\bar{y}|\bar{x}) = \frac{P(\bar{y}|\bar{x}) \cdot P(\bar{x})}{P(\bar{x})} = \frac{P(\bar{y}, \bar{x})}{P(\bar{x})}$$

"Given words $\bar{x}$, what is the conditional dist. over sequences $\bar{y}$?"

Inference in HMMs — Viterbi algorithm

Given $\bar{x}$, compute

$$\underset{\bar{y}}{\operatorname{argmax}}\ P(\bar{y} \mid \bar{x})$$

what is the most likely tag sequence?

$$= \underset{\bar{y}}{\operatorname{argmax}}\ P(\bar{y}, \bar{x})$$

$$= \underset{\bar{y}}{\operatorname{argmax}}\ \log P(\bar{y}, \bar{x})$$

Let $\tilde{y} = \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n$ be the pred $\bar{y}$

$$= \underset{\tilde{y}_1, \ldots, \tilde{y}_n}{\operatorname{argmax}}\ \log P(\tilde{y}_1) + \log P(x_1 \mid \tilde{y}_1) + \log P(\tilde{y}_2 \mid \tilde{y}_1) + \log P(x_2 \mid \tilde{y}_2) + \ldots$$

Viterbi Dynamic Program

Define $v_i(\tilde{y})$ chart

$n \times |\mathcal{T}|$ matrix

sent len   num tags

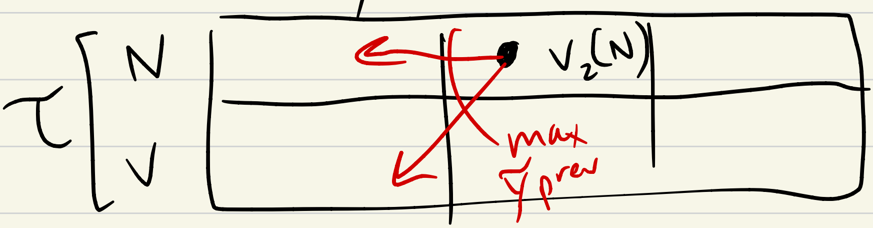
i is an index from $1 \dots n$

$\tilde{y} \in \mathcal{T}$

$v_i(\tilde{y}) =$ log prob of the best partial tag seq ending in $\tilde{y}$ at step i

Sent

V_i they can fish



which is higher?

backtrack to get sequence

compute v_i based v_{i-1}

Initial

$$v_1(\tilde{y}) = \overset{\text{emission}}{\log P(x_1 \mid \tilde{y})} + \overset{\text{initial}}{\log P(\tilde{y})}$$

Recurrent compute v_i using v_{i-1}

$$v_i(\tilde{y}) = \overset{\text{emission}}{\underline{\log P(x_i \mid \tilde{y})}}$$

$$+ \max_{\tilde{y}_{prev}} \left[\underline{\log P(\tilde{y} \mid \tilde{y}_{prev})} + \underline{v_{i-1}}(\tilde{y}_{prev}) \right]$$

$$v_2(\tilde{y}) = \log P(x_2 \mid \tilde{y}) + \max_{\tilde{y}_1} \Big[$$

$$\log P(\tilde{y} \mid \tilde{y}_1) + \underbrace{\log P(\tilde{y}_1) + \log(x_1 \mid \tilde{y}_1)}_{v_1(\tilde{y}_1)} \Big]$$

$S =$

N	-1
V	-1

log probs, don't necessarily normalize

$T =$

	N	V	STOP
N	-2	-1	-1
V	-1	-1	-2

$E =$

	they	fish	can
N	-1	-1	-3
V	-3	-1	-1

Viterbi:

for $i = 1 \dots n$

for $t \in \mathcal{T}$

compute $v_i(t)$

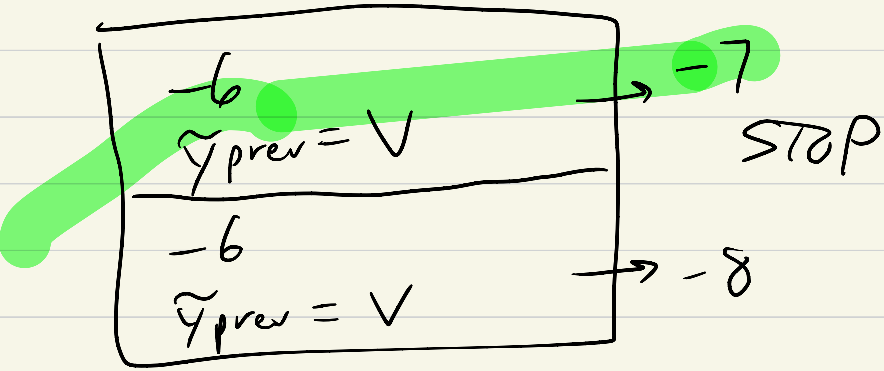
Ex: they can fish

	1	2		
	Initial	**Recurrent**		
N	-1 initi $-1 = -2$	em. can|N: $-3 + \max$	transition $-2 \; -2$ ✓ $-1 \; -4$	$\Rightarrow \boxed{-7}$ (yprev=N)
V	-1 initi -3 em. $= -4$	em: can|V: $-1 + \max$	$-1 \; -2$ $-1 \; -4$	$\rightarrow \boxed{-4}$

yprev = N

yprev = V

fish
3



Backtracking: find optimal sequence via "backpointers"

= N V N STOP

Exponential # of tag seqs

Markov property: y_i depends on y_{i-1} but not y_{i-2}

Details + Takeaways

① Training: learn an HMM from labeled (tagged) sentences by counting + normalizing

② Model vs. inference: maintain uncertainty + place dist over tag seqs.

contrast w/ BERT