# Hidden Markov Models

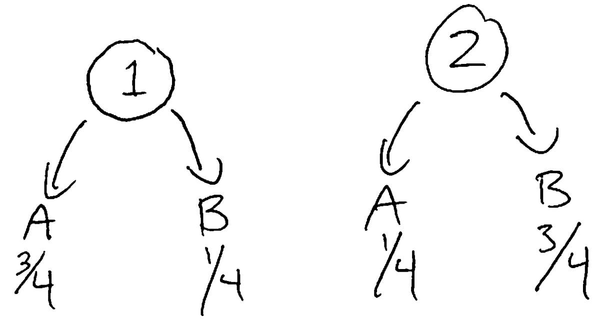


Transition Matrix ← special E boundary state

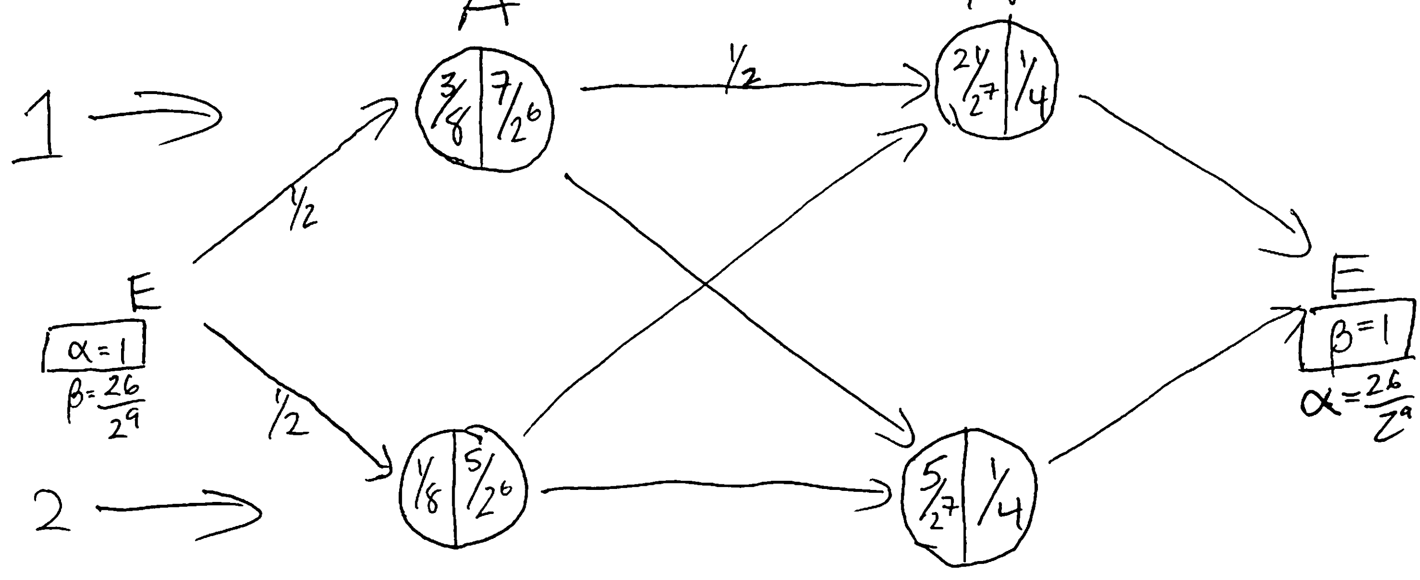
$$\begin{array}{c|ccc} & 1 & 2 & E \\ \hline 1 & & & \\ 2 & & \left[\begin{array}{ccc} 1/2 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 \\ 1/2 & 1/2 & 0 \end{array}\right] & \\ E & & & \end{array}$$

$$\begin{array}{cccc} & 1 & 2 & E \\ 1 & 1/2 & 1/4 & 1/4 \\ 2 & 1/4 & 1/2 & 1/4 \\ E & 1/2 & 1/2 & 0 \end{array}$$

Emmission Probabilities




observe AA




$\alpha(i,t)$ = ~~total~~ mass of all paths that get to state $i$ @ time $t$ and emit symbol $w_t$, up to and including the emission

$\beta(i,t)$ = ~~total~~ mass of all paths that leave state $i$ @ time $t$, not including the emmission @ time $t$

$$\alpha(1,2) = \left(\tfrac{3}{8}\tfrac{1}{2} + \tfrac{1}{8}\tfrac{1}{4}\right)\tfrac{3}{4} = \tfrac{21}{2^7}$$

$$\beta(1,1) = \tfrac{1}{2}\tfrac{3}{4}\tfrac{1}{4} + \tfrac{1}{4}\tfrac{1}{4}\tfrac{1}{4} = \tfrac{7}{2^6}$$

$$\alpha(2,2) = \left(\tfrac{3}{8}\tfrac{1}{4} + \tfrac{1}{8}\tfrac{1}{2}\right)\tfrac{1}{4} = \tfrac{5}{2^7}$$

$$\beta(2,1) = \tfrac{1}{4}\tfrac{3}{4}\tfrac{1}{4} + \tfrac{1}{2}\tfrac{1}{4}\tfrac{1}{4} = \tfrac{5}{2^6}$$

$\alpha(E,0)=1$

$\alpha(i,t+1)=\left(\sum_j \alpha_{jt} P(j\to i)\right) P(i \Rightarrow w_{t+1})$

$\beta(E,len+1)=1$

$\beta(i,t-1)=\sum_j P(i\to j) P(j\Rightarrow w_t)\beta_{jt}$

NOTE

$\alpha(E,len+1)=\beta(E,0)=Z$ total sequence probability

also for any $t$, $\sum_{i\in S} \alpha_{it}\beta_{it}=Z$, b/c all paths must go through exactly one state at time $t$
(great for debugging EM!)

Updates

$\mathbb{E}$ count of $1\to 1$ @ time $1 = \dfrac{\alpha(1,1)P(1\to 1)P(1\to A)\beta(1,2)}{Z}$

$\mathbb{E}$ count of $2\to H$ @ time $1 = \dfrac{\alpha(2,1)\beta(2,1)}{Z}$

for EM just sum all these $\mathbb{E}$ counts over all times in all sequences, normalize, and that's it.

NOTE $\mathbb{E}(1\to 1) = \dfrac{9/2^8}{26/2^9} = 9/13 > 1/2$ so the transition $1\to 1$ is "encouraged" to increase

# Decoding

## Viterbi

need for Dyn prog table but not needed for argmax

$V(i,t)$ = best path through $i$ at time $t$

$= \text{best path from} \operatorname{argmax}_j \underbrace{V(j,t-1).\text{prob}}\, p(j \to i)\, P(i \Rightarrow w_t)$

## Max Prob Tag

for time $t$ pick $\operatorname{argmax}_i \alpha_{it}\beta_{it}$

can lead to different results from Viterbi, often better fit to real world needs (# of correct tags)

# A Rainbow of HMMs



Continuous Outputs

Speech Recognition → FFT snippets

OCR → Like MNIST, but w/ a Language Model
Homework

Stocks → use generative/predictive power

Hidden Semi Markov Models

- HMM assumes #self transition RV is geometrically distributed
- HSMM uses arbitrary different distribution, but can't use normal FB algorithm

Sticky HMM's encourage self transitions through bias in the prior - important for HDP-HMM's, which have potentially ∞ # of states

used in Unsupervised POS tagging

consider

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| TOK | " | That | 's | what | she | said | " | she | said | . |
| POS | " | DT | VBZ | WP | PRP | VBD | " | PRP | VBD | . |

the 2 "said"'s will be confused b/c of their different contexts (followed by " vs .)

# UPOS Tagging Cont

also, we get 2 classes of "the", because of different noun types

## Tricks to improve performance

Condition on more nodes (Tri/Bi/Uni smoothing)

Use Morphology to figure out rare words
i.e. if running, swimming, etc are all in a class, then spelunking should be too, even without much context info.

Posterior Regularization — how do you constrain one word to be generated by a small # of states?