CS344: Introduction to Artificial Intelligence (associated lab: CS386)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay

Lecture 10 and 11: forward and backward probabilities; HMM Training; sequence labeling

27th and 31st Jan, 2011

Forward probability $F(k,i)$

- Define $F(k,i)$= Probability of being in state S_i having seen $o_0 o_1 o_2 \ldots o_k$
- $F(k,i) = P(o_0 o_1 o_2 \ldots o_k , S_i)$
- With m as the length of the observed sequence
- $P(\text{observed sequence}) = P(o_0 o_1 o_2 .. o_m)$

$$= \Sigma_{p=0,N} P(o_0 o_1 o_2 .. o_m , S_p)$$

$$= \Sigma_{p=0,N} F(m , p)$$

Forward probability (contd.)

$$F(k, q)$$

$$= P(o_0 o_1 o_2 .. o_k, S_q)$$

$$= P(o_0 o_1 o_2 .. o_k, S_q)$$

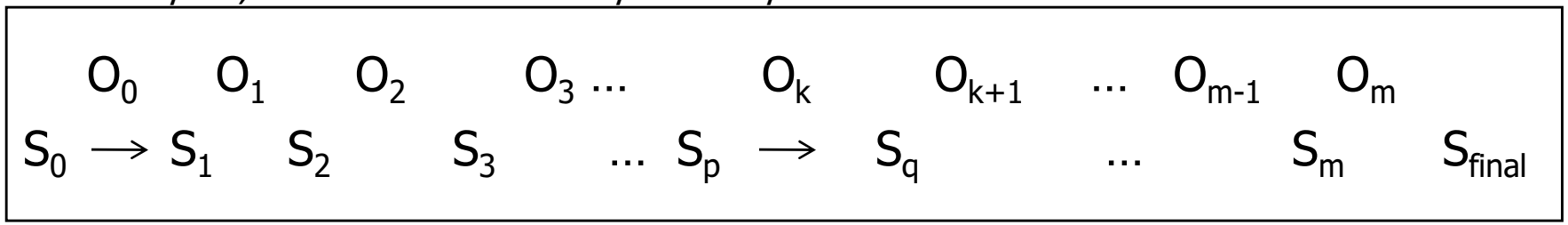
$$= P(o_0 o_1 o_2 .. o_{k-1}, o_k, S_q)$$

$$= \Sigma_{p=0,N} P(o_0 o_1 o_2 .. o_{k-1}, S_p, o_k, S_q)$$

$$= \Sigma_{p=0,N} P(o_0 o_1 o_2 .. o_{k-1}, S_p).$$
$$P(o_k, S_q / o_0 o_1 o_2 .. o_{k-1}, S_p)$$

$$= \Sigma_{p=0,N} F(k-1,p).\ P(o_k, S_q / S_p)$$

$$= \Sigma_{p=0,N} F(k-1,p).\ P(S_p \xrightarrow{o_k} S_q)$$



Backward probability *B(k,i)*

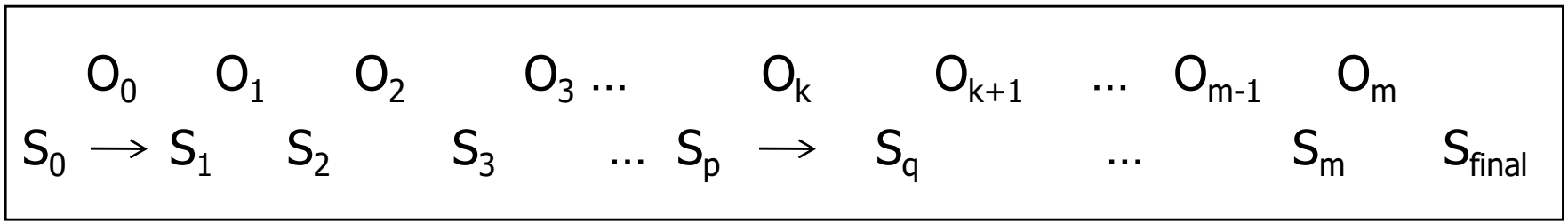
- Define *B(k,i)*= Probability of seeing $o_k o_{k+1} o_{k+2} \ldots o_m$ given that the state was S_i
- $B(k,i)=P(o_k o_{k+1} o_{k+2} \ldots o_m \mid S_i)$
- With *m* as the length of the observed sequence
- $P(\text{observed sequence})=P(o_0 o_1 o_2 .. o_m)$

  $= P(o_0 o_1 o_2 .. o_m \mid S_0)$

  $=B(0,0)$

Backward probability (contd.)

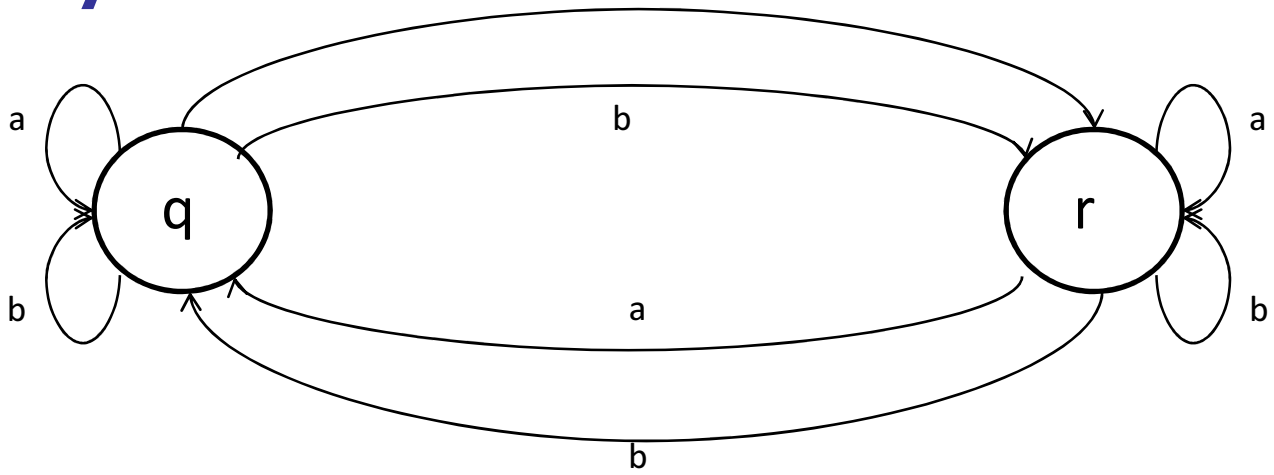
$$
\begin{aligned}
&B(k, p) \\
&= P(o_k o_{k+1} o_{k+2} \ldots o_m \mid S_p) \\
&= P(o_{k+1} o_{k+2} \ldots o_m , o_k \mid S_p) \\
&= \Sigma_{q=0,N} P(o_{k+1} o_{k+2} \ldots o_m , o_k , S_q \mid S_p) \\
&= \Sigma_{q=0,N} P(o_k , S_q \mid S_p) \\
&\quad P(o_{k+1} o_{k+2} \ldots o_m \mid o_k , S_q , S_p) \\
&= \Sigma_{q=0,N} P(o_{k+1} o_{k+2} \ldots o_m \mid S_q). P(o_k , \\
&\quad S_q \mid S_p) \\
&= \Sigma_{q=0,N} B(k+1,q). P(S_p \xrightarrow{o_k} S_q)
\end{aligned}
$$



HMM Training

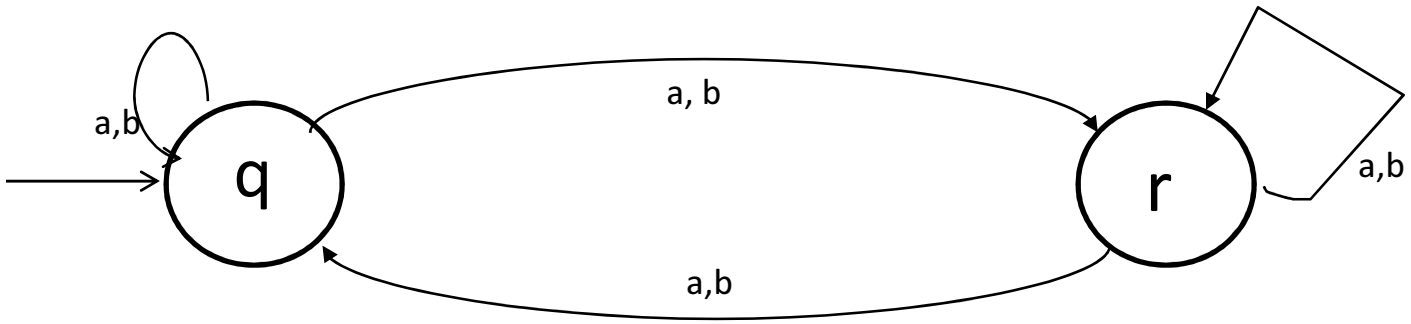
Baum Welch or Forward Backward Algorithm

Key Intuition



Given:	Training sequence
Initialization:	Probability values
Compute:	Pr (state seq | training seq)
	get expected count of transition
	compute rule probabilities
Approach:	Initialize the probabilities and recompute them…
	EM like approach

Baum-Welch algorithm: counts



String = abb aaa bbb aaa

Sequence of states with respect to input symbols

o/p seq

State seq

$$\rightarrow q \xrightarrow{a} r \xrightarrow{b} q \xrightarrow{b} q \xrightarrow{a} r \xrightarrow{a} q \xrightarrow{a} r \xrightarrow{b} q \xrightarrow{b} q \xrightarrow{b} q \xrightarrow{a} r \xrightarrow{a} q \xrightarrow{a} r$$

Calculating probabilities from table

$$P(q \xrightarrow{a} r) = 5/8$$

$$P(q \xrightarrow{b} r) = 3/8$$

$$P(s^i \xrightarrow{w_k} s^j) = \frac{c(s^i \xrightarrow{w_k} s^j)}{\sum_{l=1}^{T}\sum_{m=1}^{A} c(s^i \xrightarrow{w_m} s^l)}$$

Table of counts

Src	Dest	O/P	Count
q	r	a	5
q	q	b	3
r	q	a	3
r	q	b	2

T=#states

A=#alphabet symbols

Now if we have a non-deterministic transitions then multiple state seq possible for the given o/p seq (ref. to previous slide's feature). Our aim is to find expected count through this.

Interplay Between Two Equations

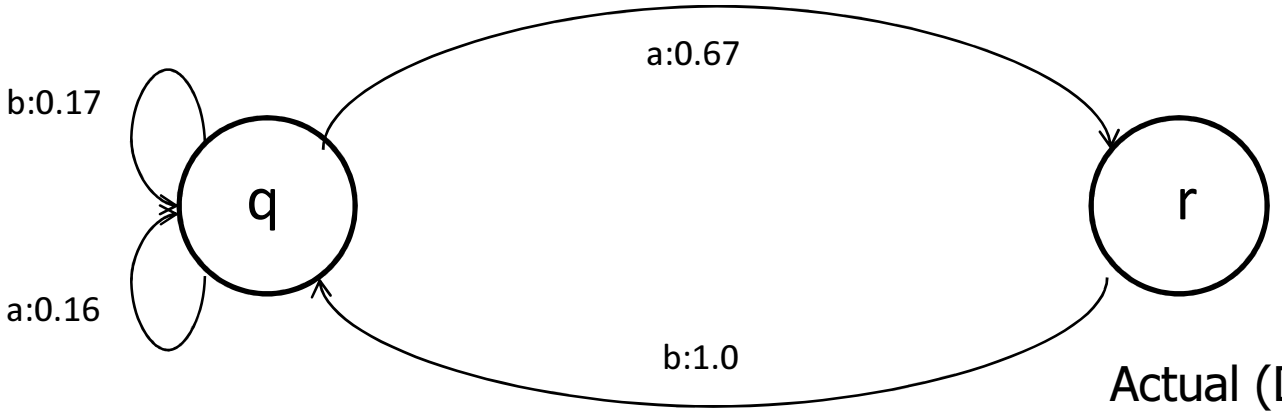
$$P(s^i \xrightarrow{W_k} s^j) = \frac{c(s^i \xrightarrow{W_k} s^j)}{\sum_{l=0}^{T} \sum_{m=0}^{A} c(s^i \xrightarrow{Wm} s^l)}$$

$$C(s^i \xrightarrow{W_k} s^j) =$$

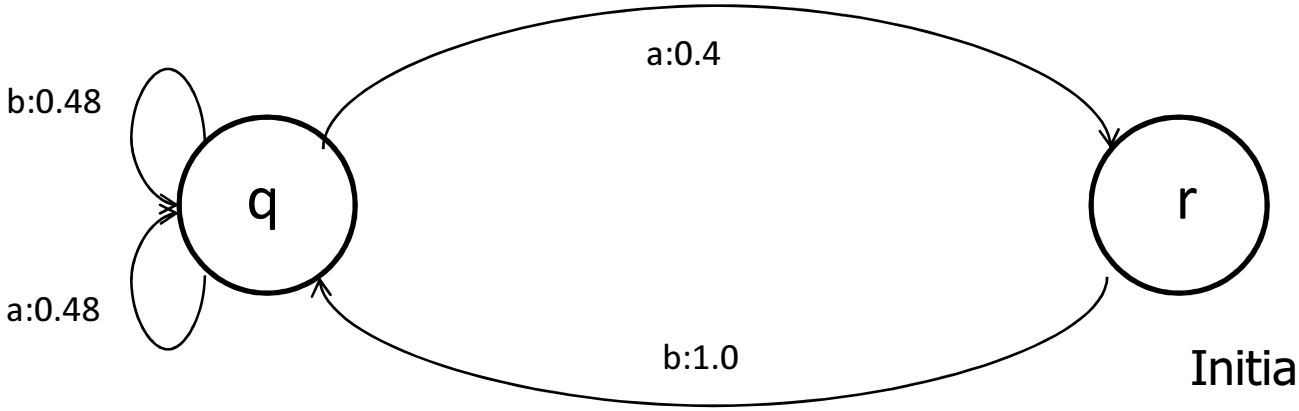
$$\sum_{S_{0,n+1}} P(S_{0,n+1} \mid W_{0,n}) \times \eta(s^i \xrightarrow{W_k} s^j, S_{0,n+1}, w_{0,n})$$

No. of times the transitions $s^i \xrightarrow{w_k} s^j$ occurs in the string

Illustration



Actual (Desired) HMM



Initial guess

One run of Baum-Welch algorithm: *string ababb*

$\in \to a$	$a \to b$	$b \to a$	$a \to b$	$b \to b$	$b \to \in$	P(path)	$q \xrightarrow{a} r$	$r \xrightarrow{b} q$	$q \xrightarrow{a} q$	$q \xrightarrow{b} q$
q	r	q	r	q	q	0.00077	0.00154	0.00154	0	0.00077
q	r	q	q	q	q	0.00442	0.00442	0.00442	0.00442	0.00884
q	q	q	r	q	q	0.00442	0.00442	0.00442	0.00442	0.00884
q	q	q	q	q	q	0.02548	0.0	0.000	0.05096	0.07644
Rounded Total →						0.035	0.01	0.01	0.06	0.095
New Probabilities (P) →							0.06 =(0.01/(0.01+0.06+0.095)	1.0	0.36	0.581

State sequences

* ε is considered as starting and ending symbol of the input sequence string. Through multiple iterations the probability values will converge.

Computational part (1/2)

$$C(s^i \xrightarrow{W_k} s^j) = \sum_{S_{0,n+1}} [P(S_{0,n+1} \mid W_{0,n}) \times n(s^i \xrightarrow{W_k} s^j, S_{0,n+1}, W_{0,n})]$$

$$= \frac{1}{P(W_{0,n})} \sum_{S_{0,n+1}} [P(S_{0,n+1}, W_{0,n}) \times n(s^i \xrightarrow{W_k} s^j, S_{0,n+1}, W_{0,n})]$$

$$= \frac{1}{P(W_{0,n})} \sum_{t=0,n} \sum_{S_{0,n+1}} [P(S_t = s^i, W_t = w_k, S_{t+1} = s^j, S_{0,n+1}, W_{0,n})]$$

$$= \frac{1}{P(W_{0,n})} \sum_{t=0,n} [P(S_t = s^i, W_t = w_k, S_{t+1} = s^j, W_{0,n})]$$

$$S0 \xrightarrow{w_0} S1 \xrightarrow{w_1} S1 \xrightarrow{w_2} \ldots Si \xrightarrow{w_k} Sj \ldots \rightarrow Sn\text{-}1 \xrightarrow{w_{n-1}} Sn \xrightarrow{w_n} Sn+1$$

Computational part (2/2)

$$\sum_{t=0}^{n} P(S_t = s^i, S_{t+1} = s^j, W_t = w_k, W_{0,n})$$

$$= \sum_{t=0}^{n} P(W_{0,t-1}, S_t = s^i, S_{t+1} = s^j, W_t = w_k, W_{t+1,n})$$

$$= \sum_{t=0}^{n} P(W_{0,t-1}, S_t = s^i) P(S_{t+1} = s^j, W_t = w_k \mid W_{0,t-1}, S_t = s^i) P(W_{t+1,n} \mid S_{t+1} = s^j)$$

$$= \sum_{t=0}^{n} F(t-1, i) P(S_{t+1} = s^j, W_t = w_k \mid S_t = s^i) B(t+1, j)$$

$$= \sum_{t=0}^{n} F(t-1, i) P(S_{t+1} = s^j, W_t = w_k \mid S_t = s^i) B(t+1, j)$$

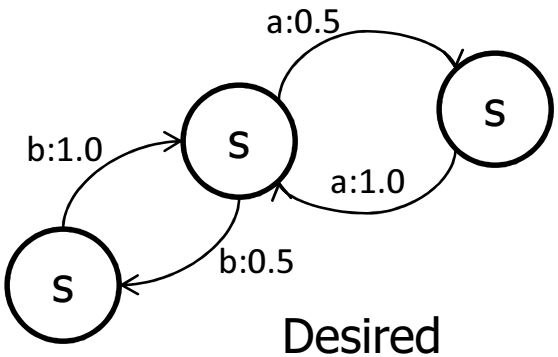
$$= \sum_{t=0}^{n} F(t-1, i) P(s^i \xrightarrow{w_k} s^j) B(t+1, j)$$

$$S0 \xrightarrow{w_0} S1 \xrightarrow{w_1} S1 \xrightarrow{w_2} \ldots Si \xrightarrow{w_k} Sj \ldots \rightarrow Sn\text{-}1 \xrightarrow{w_{n-1}} Sn \xrightarrow{w_n} Sn+1$$

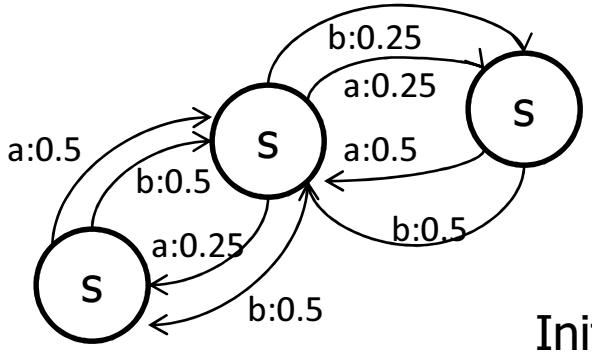
Discussions

1. Symmetry breaking:

Example: Symmetry breaking leads to no change in initial values



Desired



Initialized

2 Struck in Local maxima

3. Label bias problem

Probabilities have to sum to 1.

Values can rise at the cost of fall of values for others.