

CSE 842 Natural Language Processing

Lecture 7: Hidden Markov Model (II)

2/4/2009

CSE842, Spring 2009, MSU

1

The Three Basic Problems for HMMs

- **Problem 1 (Evaluation):** Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi = (A, B, \pi)$, **how do we efficiently compute $P(O|\Phi)$** , the probability of the observation sequence, given the model
- **Problem 2 (Decoding):** Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi = (A, B, \pi)$, **how do we choose a corresponding state sequence $Q=(q_1, q_2, \dots, q_T)$** that is optimal in some sense (i.e., best explains the observations)
- **Problem 3 (Learning):** **How do we adjust the model parameters $\Phi = (A, B, \pi)$** to maximize $P(O|\Phi)$?

2/4/2009

CSE842, Spring 2009, MSU

2

From Rabiner

The Evaluation Problem

With an observation sequence $O=(o_1, o_2, \dots, o_T)$, state sequence $Q=(q_1, q_2, \dots, q_T)$, and model Φ :

Probability of O , given state sequence Q and model Φ , is:

$$P(O|Q, \Phi) = \prod_{t=1}^T P(o_t | q_t, \Phi)$$

assuming conditional independence between observations. This expands:

$$P(O|Q, \Phi) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

The probability of the state sequence Q can be written:

$$P(Q|\Phi) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

2/4/2009

CSE842, Spring 2009, MSU

3

The Evaluation Problem

- Given an observation sequence O and HMM Φ , compute $P(O|\Phi)$
- Why is this hard? Sum over all possible sequences of states!

$$P(O|\Phi) = \sum_{all\ Q} P(O, Q|\Phi) = \sum_{all\ Q} P(Q|\Phi) P(O|Q, \Phi)$$

$$= \sum_{q_1 q_2 \dots q_T} \pi_{q_1} \cdot b_{q_1}(o_1) \cdot a_{q_1 q_2} \cdot b_{q_2}(o_2) \cdot a_{q_2 q_3} \dots a_{q_{T-1} q_T} \cdot b_{q_T}(o_T)$$

- We cannot do an explicit sum over all paths because it's intractable: $O(N^T)$
- What we do instead: **the Forward Algorithm**. $O(N^2T)$

2/4/2009

CSE842, Spring 2009, MSU

4

The Forward Algorithm

- Define variable α which has meaning of "the probability of observations o_1 through o_t and being in state i at time t , given our HMM Φ "

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i | \Phi)$$

Compute α and $P(O|\Phi)$ with the following procedure:

$$\alpha_1(i) = \pi_i \cdot b_i(o_1) \quad 1 \leq i \leq N$$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}) \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$

Termination:

$$P(O|\Phi) = \sum_{i=1}^N \alpha_T(i)$$

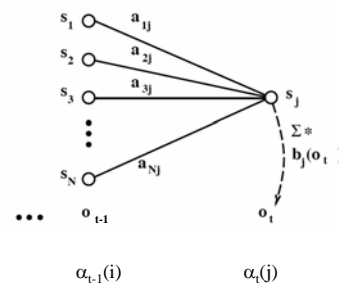
2/4/2009

CSE842, Spring 2009, MSU

5

The Inductive Step

- Computation of $\alpha_t(j)$ by summing all previous values $\alpha_{t-1}(i)$ for all i



2/4/2009

CSE842, Spring 2009, MSU

6

Forward Algorithm

$\alpha_1(1) = \pi_1 b_1(o_1)$

$\alpha_1(2) = \pi_2 b_2(o_1)$

$\alpha_1(3) = \pi_3 b_3(o_1)$

$\alpha_2(j) = \sum_{i=1}^N \alpha_{j-1}(i) a_{ij} b_j(o_j)$

s_1

s_2

s_3

o_1 o_2 o_3 o_4

2/4/2009 CSE842, Spring 2009, MSU 7

Forward Algorithm

$P(o_1 o_2 o_3 o_4 | \Phi) = \alpha_4(1) + \alpha_4(2) + \alpha_4(3)$

s_1

s_2

s_3

o_1 o_2 o_3 o_4

2/4/2009 CSE842, Spring 2009, MSU 8

Forward Algorithm: An Example

- Given the following model of the weather:
(states indicate pressure)

	state M	state H	state L	
$P(\text{sun})$	0.50	0.75	0.25	$\pi_M = 0.50$
$P(\text{rain})$	0.50	0.25	0.75	$\pi_H = 0.20$
				$\pi_L = 0.30$

2/4/2009 CSE842, Spring 2009, MSU 9

Forward Algorithm: An Example

- What is the probability of the observation sequence:
s r r s r (s=sun,r=rain)?

2/4/2009 CSE842, Spring 2009, MSU 10

Forward Algorithm: An Example

- What is the probability of the observation sequence:
s r r s r (s=sun,r=rain)?
- Compute value of $\Sigma \alpha$ at time 5...

$\alpha_1(M)=0.5 \cdot 0.5$

$\alpha_1(M)=0.25$

$\alpha_1(H)=0.2 \cdot 0.75$

$\alpha_1(H)=0.15$

$\alpha_1(L)=0.3 \cdot 0.25$

$\alpha_1(L)=0.075$

$\alpha_2(M) = [0.25 \cdot 0.3 + 0.15 \cdot 0.4 + 0.075 \cdot 0.4] \cdot 0.5 = 0.0825$

$\alpha_2(H) = [0.25 \cdot 0.4 + 0.15 \cdot 0.5 + 0.075 \cdot 0.1] \cdot 0.25 = 0.0456$

$\alpha_2(L) = [0.25 \cdot 0.3 + 0.15 \cdot 0.1 + 0.075 \cdot 0.5] \cdot 0.75 = 0.0956$

$\alpha_3(M) = [0.0825 \cdot 0.3 + 0.0456 \cdot 0.4 + 0.0956 \cdot 0.4] \cdot 0.5 = 0.0406$

$\alpha_3(H) = [0.0825 \cdot 0.4 + 0.0456 \cdot 0.5 + 0.0956 \cdot 0.1] \cdot 0.25 = 0.0163$

$\alpha_3(L) = [0.0825 \cdot 0.3 + 0.0456 \cdot 0.1 + 0.0956 \cdot 0.5] \cdot 0.75 = 0.0578$

2/4/2009 CSE842, Spring 2009, MSU 11

Forward Algorithm: An Example

$\alpha_4(M) = [0.0406 \cdot 0.3 + 0.0163 \cdot 0.4 + 0.0578 \cdot 0.4] \cdot 0.5 = 0.0209$

$\alpha_4(H) = [0.0406 \cdot 0.4 + 0.0163 \cdot 0.5 + 0.0578 \cdot 0.1] \cdot 0.75 = 0.0226$

$\alpha_4(L) = [0.0406 \cdot 0.3 + 0.0163 \cdot 0.1 + 0.0578 \cdot 0.5] \cdot 0.25 = 0.0107$

$\alpha_5(M) = [0.0209 \cdot 0.3 + 0.0226 \cdot 0.4 + 0.0107 \cdot 0.4] \cdot 0.5 = 0.0098$

$\alpha_5(H) = [0.0209 \cdot 0.4 + 0.0226 \cdot 0.5 + 0.0107 \cdot 0.1] \cdot 0.25 = 0.0052$

$\alpha_5(L) = [0.0209 \cdot 0.3 + 0.0226 \cdot 0.1 + 0.0107 \cdot 0.5] \cdot 0.75 = 0.0104$

0.0254

2/4/2009 CSE842, Spring 2009, MSU 12

The Three Basic Problems for HMMs

- Problem 1 (Evaluation): Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi = (A, B, \pi)$, how do we efficiently compute $P(O | \Phi)$, the probability of the observation sequence, given the model
- Problem 2 (Decoding): Given the observation sequence $O=(o_1, o_2, \dots, o_T)$, and an HMM model $\Phi = (A, B, \pi)$, how do we choose a corresponding state sequence $Q=(q_1, q_2, \dots, q_T)$ that is optimal in some sense (i.e., best explains the observations)
- Problem 3 (Learning): How do we adjust the model parameters $\Phi = (A, B, \pi)$ to maximize $P(O | \Phi)$?

2/4/2009

CSE842, Spring 2009, MSU

13

The Decoding Problem

- Given observations $O=(o_1, o_2, \dots, o_T)$, and HMM $\Phi=(A, B, \pi)$, how do we choose best state sequence $Q=(q_1, q_2, \dots, q_T)$?
- The forward algorithm computes $P(O | \Phi)$
- Now we want to find Q that maximizes $P(Q, O | \Phi)$
- Decoding:
 - Viterbi Decoding: dynamic programming, slight modification of the forward algorithm

2/4/2009

CSE842, Spring 2009, MSU

14

Viterbi Algorithm

- Question: *What is best score along a single path, up to time t , ending in state i ?*
- Use inductive procedure
- Best sequence defined as:

$$v_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \Phi)$$

- First iteration ($t=1$):

$$v_1(i) = P(q_1 = i, o_1 | \Phi)$$

$$v_1(i) = P(q_1 = i | \Phi) \cdot P(o_1 | q_1 = i, \Phi)$$

$$v_1(i) = \pi_i \cdot b_i(o_1)$$

2/4/2009

CSE842, Spring 2009, MSU

15

Viterbi Algorithm

- Second iteration ($t=2$)

$$v_2(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \Phi)$$

$$v_2(i) = \max_{q_1} P(q_1 = k, q_2 = i, o_1, o_2 | \Phi)$$

$$v_2(i) = \max_{q_1} (P(q_1 = k, o_1 | \Phi) \cdot P(q_2 = i, o_2 | q_1 = k, o_1, \Phi))$$

$$v_2(i) = \max_{q_1} (v_1(k) \cdot P(q_2 = i | q_1 = k, o_1, \Phi) \cdot P(o_2 | q_2 = i, q_1 = k, o_1, \Phi))$$

2/4/2009

CSE842, Spring 2009, MSU

16

Viterbi Algorithm

- Second iteration ($t=2$) (continued...)

$$v_2(i) = \max_{q_1} (v_1(k) \cdot P(q_2 = i | q_1 = k, o_1, \Phi) \cdot P(o_2 | q_2 = i, q_1 = k, o_1, \Phi))$$

$$v_2(i) = \max_{q_1} (v_1(k) \cdot P(q_2 = i | q_1 = k, \Phi) \cdot P(o_2 | q_2 = i, \Phi))$$

$$v_2(i) = \max_k v_1(k) \cdot a_{ki} \cdot b_i(o_2)$$

conditional independence assumption

$$v_2(i) = \left(\max_k v_1(k) \cdot a_{ki} \right) \cdot b_i(o_2)$$

2/4/2009

CSE842, Spring 2009, MSU

17

Viterbi Algorithm

- In general, for any value of t :

$$v_t(j) = \left(\max_i v_{t-1}(i) \cdot a_{ij} \right) \cdot b_j(o_t)$$

- Best path from $\{1, 2, \dots, t\}$ is *not* dependent on future times $\{t+1, t+2, \dots, T\}$ (from definition)

- Best path from $\{1, 2, \dots, t\}$ is not necessarily the same as the best path from $\{1, 2, \dots, (t-1)\}$ concatenated with the best path $\{(t-1), t\}$

2/4/2009

CSE842, Spring 2009, MSU

18

Viterbi Algorithm

- Keep in memory only $v_{t-1}(i)$ for all i .
- For each time t and state j , need $(N \text{ multiply and compare}) + (1 \text{ multiply})$
- For each time t , need $N * ((N \text{ multiply and compare}) + (1 \text{ multiply}))$
- To find best path, need $O(N^2T)$ operations.
- This is *much* better than NT possible paths, especially for large T !

2/4/2009 CSE842, Spring 2009, MSU 19

Viterbi Algorithm

$v_t(j) = \max_{1 \leq i \leq N} [v_{t-1}(i) \alpha_{ij}] b_j(o_t)$

2/4/2009 CSE842, Spring 2009, MSU 20

Viterbi Algorithm

$v_t(j) = \max_{1 \leq i \leq N} [v_{t-1}(i) \alpha_{ij}] b_j(o_t)$

2/4/2009 CSE842, Spring 2009, MSU 21

Viterbi Algorithm

$v_t(j) = \max_{1 \leq i \leq N} [v_{t-1}(i) \alpha_{ij}] b_j(o_t)$

2/4/2009 CSE842, Spring 2009, MSU 22

Viterbi Algorithm

$v_t(j) = \max_{1 \leq i \leq N} [v_{t-1}(i) \alpha_{ij}] b_j(o_t)$

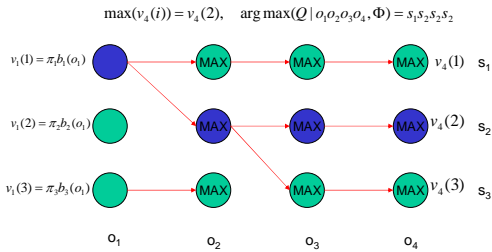
2/4/2009 CSE842, Spring 2009, MSU 23

Viterbi Algorithm

$v_t(j) = \max_{1 \leq i \leq N} [v_{t-1}(i) \alpha_{ij}] b_j(o_t)$

2/4/2009 CSE842, Spring 2009, MSU 24

Viterbi Algorithm



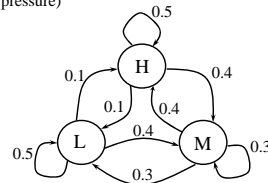
2/4/2009

CSE842, Spring 2009, MSU

25

The same example for Viterbi Algorithm

- Given the following model of the weather:
(states indicate pressure)



	state M	state H	state L	
$P(\text{sun})$	0.50	0.75	0.25	$\pi_M = 0.50$
$P(\text{rain})$	0.50	0.25	0.75	$\pi_H = 0.20$
				$\pi_L = 0.30$

2/4/2009

CSE842, Spring 2009, MSU

26

The same example for Viterbi Algorithm

- Same observations: `s r r s r`
- Compute Viterbi path up to time 5...

$$\begin{aligned} v_1(M) &= 0.5 \cdot 0.5 & v_1(H) &= 0.2 \cdot 0.75 & v_1(L) &= 0.3 \cdot 0.25 \\ \boxed{v_1(M) &= 0.25} & v_1(H) &= 0.15 & v_1(L) &= 0.075 \end{aligned}$$

$$\begin{aligned} v_2(M) &= \max[0.25 \cdot 0.3, 0.15 \cdot 0.4, 0.075 \cdot 0.4] \cdot 0.5 = 0.0375 \\ v_2(H) &= \max[0.25 \cdot 0.4, 0.15 \cdot 0.5, 0.075 \cdot 0.1] \cdot 0.25 = 0.0250 \\ v_2(L) &= \max[0.25 \cdot 0.3, 0.15 \cdot 0.1, 0.075 \cdot 0.5] \cdot 0.75 = 0.0563 \end{aligned}$$

$$\begin{aligned} v_3(M) &= \max[0.0375 \cdot 0.3, 0.0250 \cdot 0.4, \boxed{0.0563 \cdot 0.4}] \cdot 0.5 = 0.0113 \\ v_3(H) &= \max[0.0375 \cdot 0.4, 0.0250 \cdot 0.5, 0.0563 \cdot 0.1] \cdot 0.25 = 0.0038 \\ v_3(L) &= \max[0.0375 \cdot 0.3, 0.0250 \cdot 0.1, \boxed{0.0563 \cdot 0.5}] \cdot 0.75 = 0.0211 \end{aligned}$$

2/4/2009

CSE842, Spring 2009, MSU

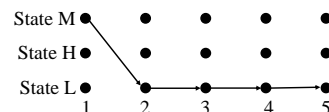
27

The same example for Viterbi Algorithm

$$\begin{aligned} v_4(M) &= \max[0.0113 \cdot 0.3, 0.0038 \cdot 0.4, \boxed{0.0211 \cdot 0.4}] \cdot 0.5 = 0.0042 \\ v_4(H) &= \max[0.0113 \cdot 0.4, 0.0038 \cdot 0.5, 0.0211 \cdot 0.1] \cdot 0.75 = 0.0034 \\ v_4(L) &= \max[0.0113 \cdot 0.3, 0.0038 \cdot 0.1, \boxed{0.0211 \cdot 0.5}] \cdot 0.25 = 0.0026 \end{aligned}$$

$$\begin{aligned} v_5(M) &= \max[0.0042 \cdot 0.3, \boxed{0.0034 \cdot 0.4}, 0.0026 \cdot 0.4] \cdot 0.5 = 0.0007 \\ v_5(H) &= \max[0.0042 \cdot 0.4, \boxed{0.0034 \cdot 0.5}, 0.0026 \cdot 0.1] \cdot 0.25 = 0.0004 \\ v_5(L) &= \max[0.0042 \cdot 0.3, 0.0034 \cdot 0.1, \boxed{0.0026 \cdot 0.5}] \cdot 0.75 = \boxed{0.0010} \end{aligned}$$

Maximum state score at time 5 = 0.0010 in state L



2/4/2009

CSE842, Spring 2009, MSU

28

Compare Forward Algorithm and Viterbi Algorithm

- both Viterbi and Forward produce answer in N^2T calculations
- Forward and Viterbi yield different results because Forward is computing total probability of being in a state, regardless of any state sequence used to get to that state. Viterbi gives best state sequence until time t , but not overall probability of being in a given state.
- for Viterbi search, we want most likely state sequence, and we can only be in one state at any time t . So we take the "max" to determine which state.
- for the Forward procedure, we want the total probability, so we sum the probabilities over all possible paths to any given state.

2/4/2009

CSE842, Spring 2009, MSU

29

The Three Basic Problems for HMMs

- Problem 1 (Evaluation): Given the observation sequence $O=(o_1 o_2 \dots o_T)$, and an HMM model $\Phi=(A, B, \pi)$, how do we efficiently compute $P(O|\Phi)$, the probability of the observation sequence, given the model
- Problem 2 (Decoding): Given the observation sequence $O=(o_1 o_2 \dots o_T)$, and an HMM model $\Phi=(A, B, \pi)$, how do we choose a corresponding state sequence $Q=(q_1 q_2 \dots q_T)$ that is optimal in some sense (i.e., best explains the observations)
- Problem 3 (Learning): How do we adjust the model parameters $\Phi=(A, B, \pi)$ to maximize $P(O|\Phi)$?

2/4/2009

CSE842, Spring 2009, MSU

30

The Learning Problem in HMM: Baum-Welch

- Baum-Welch = Forward-Backward Algorithm (Baum 1972)
- Is a special case of the EM or Expectation Maximization algorithm (Dempster, Laird, Rubin)
- The algorithm will let us train the transition probabilities $A = \{a_{ij}\}$ and the emission probabilities $B = \{b_j(o_t)\}$ of the HMM

2/4/2009

CSE842, Spring 2009, MSU

31

The Learning Problem: Caveats

- Network structure of HMM is always created by hand
 - no algorithm for double induction of optimal structure and probabilities has been able to beat simple hand built structures.
- Baum-Welch only guaranteed to return local max, rather than global optimum

2/4/2009

CSE842, Spring 2009, MSU

32

Starting out with Observable Markov Models

- How to train?
- Run the model on the observation sequence O .
- Since it's not hidden, we know which states we went through, hence which transitions and observations were used.
- Given that information, training:
 - $B = \{b_j(o_t)\}$: Since every state can only generate one observation symbol, observation likelihoods B are all 1.0
 - $A = \{a_{ij}\}$: $a_{ij} = \frac{C(i \rightarrow j)}{\sum_{q \in Q} C(i \rightarrow q)}$

2/4/2009

CSE842, Spring 2009, MSU

33

Hidden Markov Model: - Expectation Maximization

- For HMM, cannot compute these counts directly from observed sequences (unless the hidden states are annotated as in our homework 1)
- Use Forward-backward Algorithm (Baum-Welch algorithm)
 - A special case of the EM algorithm.

2/4/2009

CSE842, Spring 2009, MSU

34

Review: The Forward Algorithm

- Define variable α which has meaning of "the probability of observations o_1 through o_t and being in state i at time t , given our HMM Φ "

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i | \Phi)$$

Compute α and $P(O | \Phi)$ with the following procedure:

$$\alpha_1(i) = \pi_i \cdot b_j(o_1) \quad 1 \leq i \leq N$$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}) \quad \begin{array}{l} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{array}$$

Termination:

$$P(O | \Phi) = \sum_{i=1}^N \alpha_T(i)$$

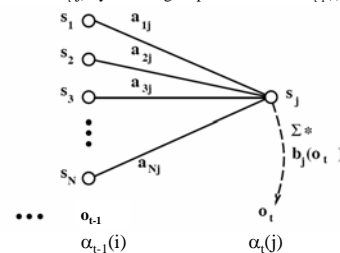
2/4/2009

CSE842, Spring 2009, MSU

35

The inductive step in Forward Algorithm

- Computation of $\alpha_t(j)$ by summing all previous values $\alpha_{t-1}(i)$ for all i



2/4/2009

CSE842, Spring 2009, MSU

36

The Backward algorithm

- In the same way that we defined α , we can define β
- Define variable β **backward probability**, which has meaning of "the probability of observations o_{t+1} through o_T , given that we're in state i at time t , and given our HMM"

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, | q_t = i, \Phi)$$

- This is the probability of generating partial observations O_{t+1}^T from time $t+1$ to the end, given that the HMM is in state i at time t and given Φ .
- We compute it by induction:
 - Initialization: $\beta_T(i) = 1, 1 \leq i \leq N$
 - Induction: $\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), t = T-1, 1 \leq i \leq N$

Termination: $\beta_0(\cdot) = \sum_{j=1}^N \pi_j \cdot b_j(o_1) \cdot \beta_1(j)$

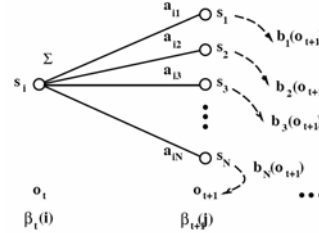
2/4/2009

CSE842, Spring 2009, MSU

37

Inductive step in the backward algorithm

Computation of $\beta_t(i)$ by weighted sum of all successive values β_{t+1}



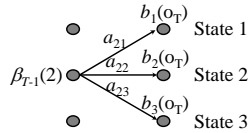
2/4/2009

CSE842, Spring 2009, MSU

38

Backward Procedure: Illustration

- For times other than T , illustrated as follows:



$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

$$\beta_{T-1}(2) = \sum_{j=1}^N a_{2j} b_j(o_T) \beta_T(j)$$

$$\beta_{T-1}(2) = a_{21} b_1(o_T) \cdot 1 + a_{22} b_2(o_T) \cdot 1 + a_{23} b_3(o_T) \cdot 1$$

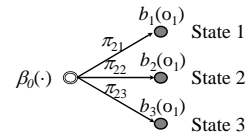
2/4/2009

CSE842, Spring 2009, MSU

39

Backward Procedure: Illustration

- Finally, we can compute $\beta_0(\cdot)$, going to a special "beginning of utterance" state that emits the π values in the transition to the first "real" states, from time 0 to time 1.



$$\beta_0(\cdot) = \sum_{j=1}^N \pi_j b_j(o_1) \beta_1(j)$$

$$\beta_0(\cdot) = \pi_1 b_1(o_1) \beta_1(1) + \pi_2 b_2(o_1) \beta_1(2) + \pi_3 b_3(o_1) \beta_1(3)$$

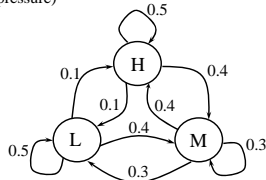
2/4/2009

CSE842, Spring 2009, MSU

40

Backward Algorithm: the same example

- Given the following model of the weather: (states indicate pressure)



	state M	state H	state L	
$P(\text{sun})$	0.50	0.75	0.25	$\pi_M = 0.50$
$P(\text{rain})$	0.50	0.25	0.75	$\pi_H = 0.20$
				$\pi_L = 0.30$

2/4/2009

CSE842, Spring 2009, MSU

41

Backward Procedure: An Example

- What is the probability of the observation sequence: s r r s r (s=sun,r=rain)? given the model from Example #1

- Compute value of β at from time 5 to time 0:

$$\beta_5(M)=1.0 \quad \beta_5(H)=1.0 \quad \beta_5(L)=1.0$$

$$\beta_4(M) = [0.3 \cdot 0.50 \cdot 1.0 + 0.4 \cdot 0.25 \cdot 1.0 + 0.3 \cdot 0.75 \cdot 1.0] = 0.4750$$

$$\beta_4(H) = [0.4 \cdot 0.50 \cdot 1.0 + 0.5 \cdot 0.25 \cdot 1.0 + 0.1 \cdot 0.75 \cdot 1.0] = 0.4000$$

$$\beta_4(L) = [0.4 \cdot 0.50 \cdot 1.0 + 0.1 \cdot 0.25 \cdot 1.0 + 0.5 \cdot 0.75 \cdot 1.0] = 0.6000$$

$$\beta_3(M) = [0.3 \cdot 0.50 \cdot 0.475 + 0.4 \cdot 0.75 \cdot 0.400 + 0.3 \cdot 0.25 \cdot 0.600] = 0.2363$$

$$\beta_3(H) = [0.4 \cdot 0.50 \cdot 0.475 + 0.5 \cdot 0.75 \cdot 0.400 + 0.1 \cdot 0.25 \cdot 0.600] = 0.2600$$

$$\beta_3(L) = [0.4 \cdot 0.50 \cdot 0.475 + 0.1 \cdot 0.75 \cdot 0.400 + 0.5 \cdot 0.25 \cdot 0.600] = 0.2000$$

2/4/2009

CSE842, Spring 2009, MSU

42

Backward Procedure: An Example

$$\beta_2(M) = [0.3 \cdot 0.50 \cdot 0.236 + 0.4 \cdot 0.25 \cdot 0.260 + 0.3 \cdot 0.75 \cdot 0.200] = 0.1064$$

$$\beta_2(H) = [0.4 \cdot 0.50 \cdot 0.236 + 0.5 \cdot 0.25 \cdot 0.260 + 0.1 \cdot 0.75 \cdot 0.200] = 0.0947$$

$$\beta_2(L) = [0.4 \cdot 0.50 \cdot 0.236 + 0.1 \cdot 0.25 \cdot 0.260 + 0.5 \cdot 0.75 \cdot 0.200] = 0.1287$$

$$\beta_1(M) = [0.3 \cdot 0.50 \cdot 0.106 + 0.4 \cdot 0.25 \cdot 0.095 + 0.3 \cdot 0.75 \cdot 0.129] = 0.0544$$

$$\beta_1(H) = [0.4 \cdot 0.50 \cdot 0.106 + 0.5 \cdot 0.25 \cdot 0.095 + 0.1 \cdot 0.75 \cdot 0.129] = 0.0428$$

$$\beta_1(L) = [0.4 \cdot 0.50 \cdot 0.106 + 0.1 \cdot 0.25 \cdot 0.095 + 0.5 \cdot 0.75 \cdot 0.129] = 0.0720$$

$$\beta_0(\cdot) = [0.5 \cdot 0.50 \cdot 0.054 + 0.2 \cdot 0.75 \cdot 0.043 + 0.3 \cdot 0.25 \cdot 0.072] = \mathbf{0.0254}$$

Note that $\beta_0(\cdot) = \sum \alpha_T(i)$; this value is the *total probability* of the observation sequence.

The result is the same as the result from the forward algorithm (see earlier notes).

2/4/2009

CSE842, Spring 2009, MSU

43

Back to the forward-backward algorithm

- Start with some guess/estimate of A, B, and π and iteratively update these probabilities based on the expected counts

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

- How to calculate these expected counts?
 - If we know the probability of being in state i at a particular point in time t , and if we know this probability for every time t , we can sum over all times t to estimate the total counts of being in state i .
 - Similarly for the expected number of transitions from state i to state j .

2/4/2009

CSE842, Spring 2009, MSU

44

From Rabiner

Computing ξ

- Now we can define ξ , the probability of being in state i at time t given an observation sequence and HMM.

$$\xi_t(i) = P(q_t = i | O, \Phi)$$

$$= \frac{P(O, q_t = i | \Phi)}{P(O | \Phi)} = \frac{P(O, q_t = i | \Phi)}{\sum_{j=1}^N P(O, q_t = j | \Phi)}$$

What is $P(O, q_t = i | \Phi)$?

2/4/2009

CSE842, Spring 2009, MSU

45

Forward and Backward

$$P(O, q_t = i | \Phi) = \alpha_t(i) \beta_t(i)$$

Where $\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i | \Phi)$

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = i, \Phi)$$

How to derive that?

2/4/2009

CSE842, Spring 2009, MSU

46

Forward and Backward

$$P(O, q_t = i | \Phi) = \alpha_t(i) \beta_t(i)$$

Where $\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = i | \Phi)$

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = i, \Phi)$$

How to derive that?

$$P(O, q_t = i | \Phi)$$

$$= P(O_{1..t}, O_{t+1..T}, q_t = i | \Phi)$$

$$= P(O_{t+1..T} | O_{1..t}, q_t = i, \Phi) \cdot P(O_{1..t}, q_t = i | \Phi)$$

$$= \alpha_t(i) \cdot \beta_t(i)$$

2/4/2009

CSE842, Spring 2009, MSU

47

Computing ξ

- Now we can define ξ , the probability of being in state i at time t given an observation sequence and HMM.

$$\xi_t(i) = P(q_t = i | O, \Phi)$$

$$= \frac{P(O, q_t = i | \Phi)}{P(O | \Phi)} = \frac{P(O, q_t = i | \Phi)}{\sum_{j=1}^N P(O, q_t = j | \Phi)}$$

Therefore:

$$\xi_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)}$$

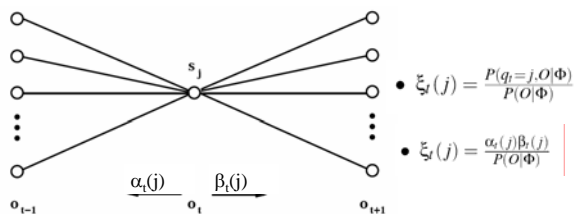
2/4/2009

CSE842, Spring 2009, MSU

48

Computing ξ

Computation of $\xi_j(t)$, the probability of being in state j at time t .



2/4/2009

CSE842, Spring 2009, MSU

49

Re-estimating the observation likelihood (emission prob.)

$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

- For numerator, sum $\xi_t(j)$ for all t in which o_t is symbol v_k .

$$\hat{b}_j(v_k) = \frac{\sum_{t=1, s.t. o_t=v_k}^T \xi_t(j)}{\sum_{t=1}^T \xi_t(j)}$$

2/4/2009

CSE842, Spring 2009, MSU

50

Re-estimation of transition probabilities

- We will estimate \hat{a}_{ij} via this intuition:

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

- Numerator intuition:
 - Assume we had some estimate of probability that a given transition $i \rightarrow j$ was taken at time t in observation sequence.
 - If we knew this probability for each time t , we could sum over all t to get expected value (count) for $i \rightarrow j$.

2/4/2009

CSE842, Spring 2009, MSU

51

Re-estimation of a_{ij}

- Let γ_t be the probability of being in state i at time t and state j at time $t+1$, given $O_{1..T}$ and model Φ :

$$\gamma_t(i, j) = P(q_t = i, q_{t+1} = j | O, \Phi)$$

$$= \frac{P(q_t = i, q_{t+1} = j, O | \Phi)}{P(O | \Phi)}$$

$$\text{Let } \gamma_t'(i, j) = P(q_t = i, q_{t+1} = j, O | \Phi)$$

2/4/2009

CSE842, Spring 2009, MSU

52

From γ' to γ

$$\begin{aligned} \gamma_t'(i, j) &= P(q_t = i, q_{t+1} = j, O | \Phi) \\ &= \alpha_i(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \end{aligned} \quad \text{How to derive this?}$$

$$P(O | \Phi) = \sum_{k=1}^N \alpha_T(k)$$

$$\gamma_t(i, j) = \frac{\gamma_t'(i, j)}{P(O | \Phi)} = \frac{\alpha_i(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{k=1}^N \alpha_T(k)}$$

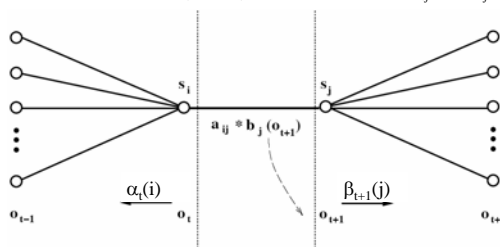
2/4/2009

CSE842, Spring 2009, MSU

53

Computing γ'

The component of $P(q_t = i, q_{t+1} = j, O | \Phi)$: α, β, a_{ij} and $b_j(o_{t+1})$



2/4/2009

CSE842, Spring 2009, MSU

54

From γ to a_{ij}

$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{j=1}^N \sum_{t=1}^{T-1} \gamma_t(i, j)}$$

2/4/2009

CSE842, Spring 2009, MSU

55

HMM Parameter Estimation Summary

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{j=1}^N \sum_{t=1}^{T-1} \gamma_t(i, j)}$$

The ratio between the expected number of transitions from state i to j and the expected number of all transitions from state i

$$\hat{b}_j(v_k) = \frac{\sum_{t=1, s.t. O_t=v_k}^T \xi_t(j)}{\sum_{t=1}^T \xi_t(j)}$$

The ratio between the expected number of times the observation data emitted from state j is v_k , and the expected number of times any observation is emitted from state j

2/4/2009

CSE842, Spring 2009, MSU

56

Forward-Backward Algorithm

- 1) Initialize $\Phi=(A,B,\pi)$
- 2) Expectation step/ E-step
 - Compute γ, ξ (through α, β)
- 3) Maximization step / M-step
 - Estimate new $\Phi'=(A,B,\pi)$
- 4) Replace Φ with Φ'
- 5) If not converged go to 2

2/4/2009

CSE842, Spring 2009, MSU

57

Summary

- We learned the Baum Welch algorithm for learning the A and B matrices of an individual HMM
- It doesn't require training data to be labeled at the state level; all you have to know is that an HMM covers a given sequence of observations, and you can learn the optimal A and B parameters (local optimal) for this data by an iterative process.

2/4/2009

CSE842, Spring 2009, MSU

58