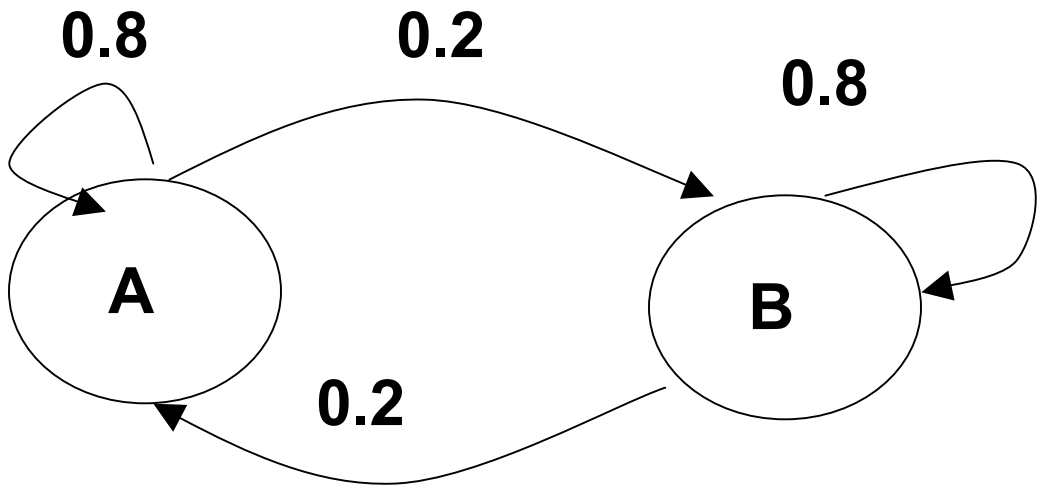
15-381: Artificial Intelligence

Hidden Markov Models (HMMs): Learning

HMMs

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3

Π_A=0.7
Π_b=0.3



A Hidden Markov model

- A set of states $\{s_1 \ldots s_n\}$

  - In each time point we are in exactly one of these states denoted by q_t
- Π_i, the probability that we start at state s_i
- A transition probability model, $P(q_t = s_i \mid q_{t-1} = s_j)$
- A set of possible outputs Σ

  - In time point t we emit a symbol $\sigma \in \Sigma$
- An emission probability model, $p(o_t = \sigma \mid s_i)$

Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ √
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$ √
- Computing $\text{argmax}_Q P(Q)$

Most probable path

- We are almost done …
- One final question remains

  How do we find the most probable path, that is Q* such that

$$P(Q^* \mid O) = \text{argmax}_Q P(Q|O)?$$

- This is an important path
  - The words in speech processing
  - The set of genes in the genome
  - etc.

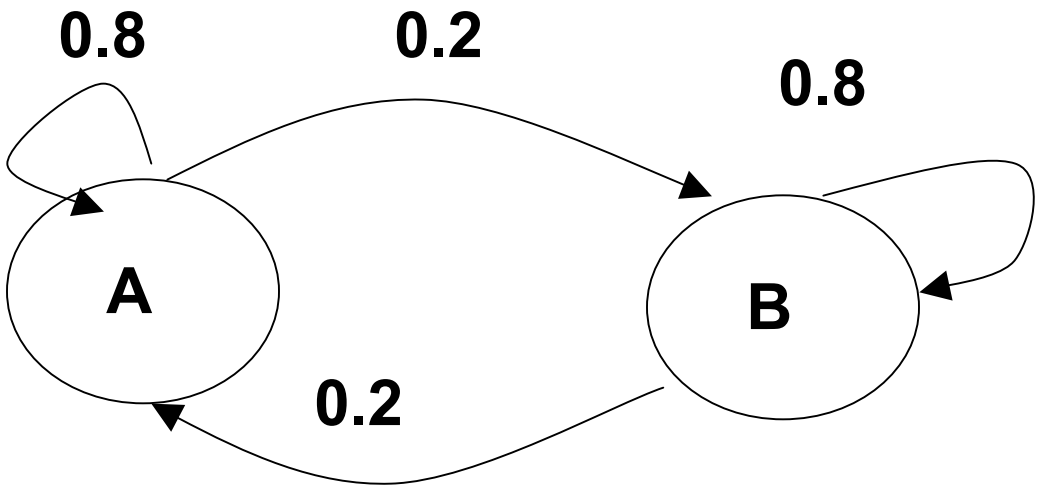
Example

- What is the most probable set of states leading to the sequence:

1,2,2,5,6,5,1,2,3 ?

Π_A=0.7
Π_b=0.3

v	P(v |A)	P(v |B)
1	.3	.1
2	.2	.1
3	.2	.1
4	.1	.2
5	.1	.2
6	.1	.3



Most probable path

$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

$$= \arg\max_Q P(O \mid Q)P(Q) = \arg\max_Q P(O,Q)$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

In other words we are interested in the most likely path from 1 to t that:

1. Ends in S_i
2. Produces outputs $O_1 \ldots O_t$

Most probable path

$$\arg\max_Q P(Q \mid O) = \arg\max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

$$= \arg\max_Q P(O \mid Q)P(Q) = \arg\max_Q P(O,Q)$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

Once we computed $\delta_t(i)$, we can solve for P(Q*|O):

$P(Q^* \mid O) = \text{argmax}_Q P(Q|O) = P(Q^* \mid O) =$

path defined by $\text{argmax}_j \delta_t(j)$,

Computing $\delta_t(i)$

$$\delta_1(i) = p(q_1 = s_i \wedge O_1)$$
$$= p(q_1 = s_i)p(O_1 \mid q_1 = s_i)$$
$$= \pi_i b_i(O_1)$$

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \wedge q_t = s_i \wedge O_1 \ldots O_t)$$

Q: Given $\delta_t(i)$, how can we compute $\delta_{t+1}(i)$?

A: To get from $\delta_t(i)$ to $\delta_{t+1}(i)$ we need to

1. Add an emission for time t+1 (O_{t+1})

2. Transition to state s_i

$$\delta_{t+1}(i) = \max_{q_1 \ldots q_t} p(q_1 \ldots q_t \wedge q_{t+1} = s_i \wedge O_1 \ldots O_{t+1})$$
$$= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i)$$
$$= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})$$

The Viterbi algorithm

$$\delta_{t+1}(i) = \max_{q_1 \ldots q_t} p(q_1 \ldots q_t \wedge q_{t+1} = s_i \wedge O_1 \ldots O_{t+1})$$

$$= \max_j \delta_t(j)\, p(q_{t+1} = s_i \mid q_t = s_j)\, p(O_{t+1} \mid q_{t+1} = s_i)$$

$$= \max_j \delta_t(j)\, a_{j,i} b_i(O_{t+1})$$

- Once again we use dynamic programming for solving $\delta_t(i)$
- Once we have $\delta_t(i)$, we can solve for our P(Q*|O)

By:

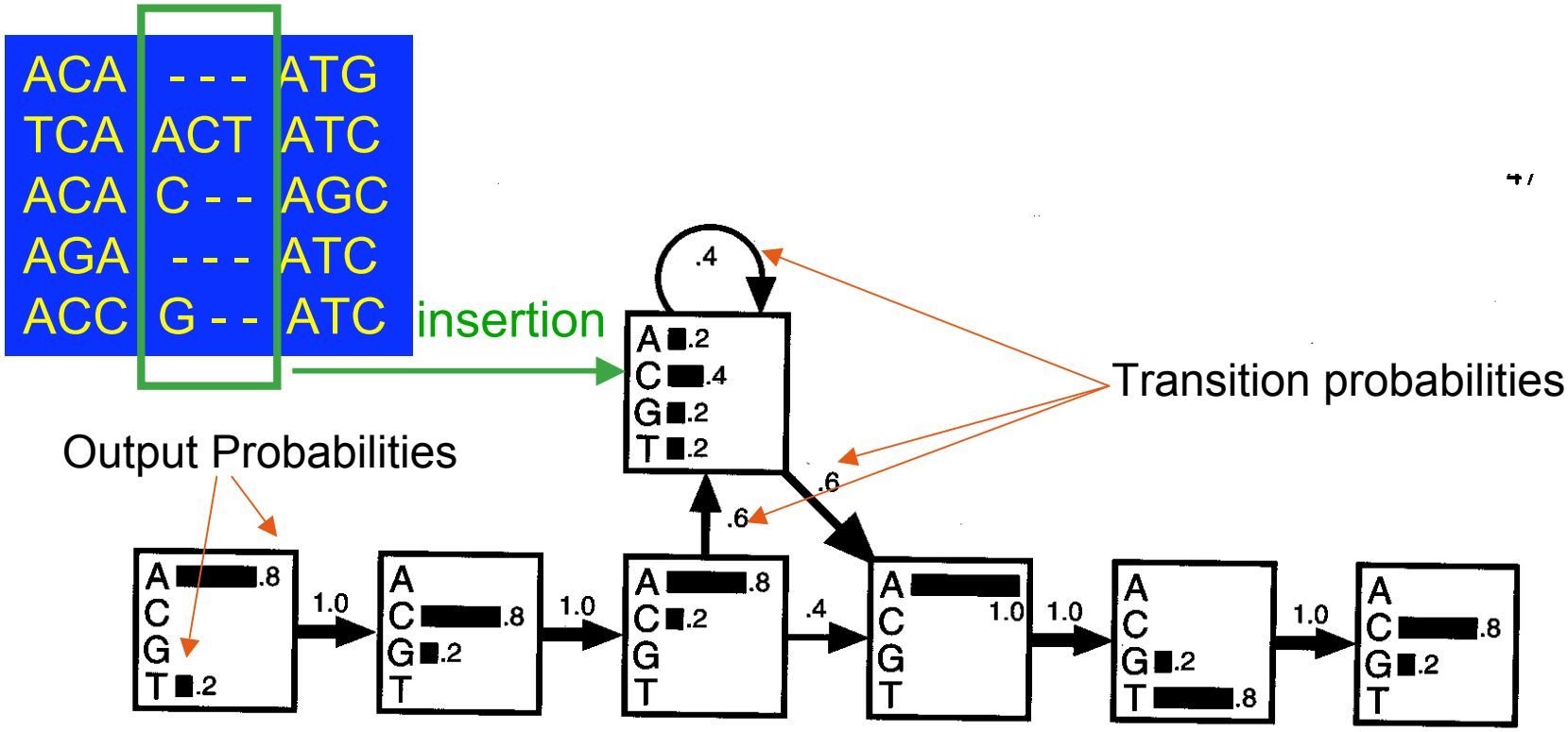
$P(Q^* \mid O) = \text{argmax}_Q P(Q|O) = P(Q^* \mid O) =$

path defined by $\text{argmax}_j\ \delta_t(j)$,

Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ √
- Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$ √
- Computing $\text{argmax}_Q P(Q)$ √

Building – *from an existing alignment*



A HMM model for a DNA motif alignments, The transitions are shown with arrows whose thickness indicate their probability. In each state, the histogram shows the probabilities of the four bases.

Learning in HMMs

- Until now we assumed that the emission and transition probabilities are known
- This is usually not the case
  - How is “AI” pronounced by different individuals?
  - What is the probability of hearing “class” after “AI”?

While we will discuss learning the transition and emission models, we will not discuss selecting the states.

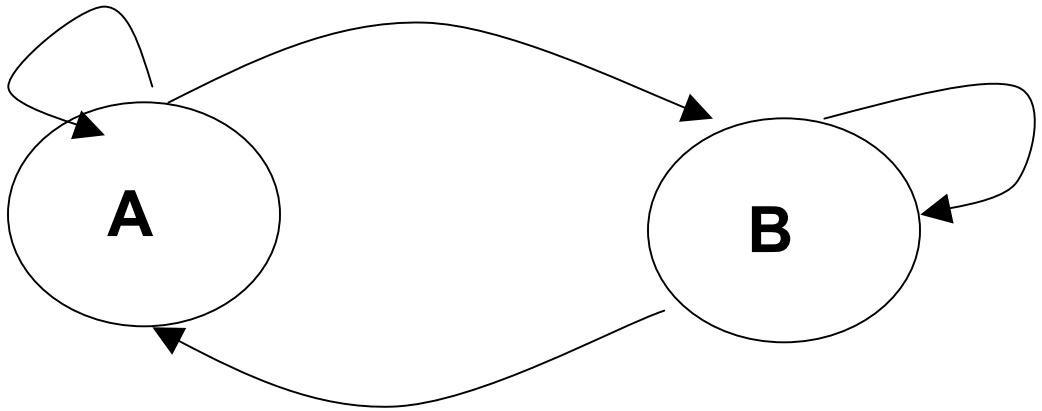
This is often the most important task and is heavily dependent on domain knowledge.

Example

- Assume the model below
- We also observe the following sequence:

  1,2,2,5,6,5,1,2,3,3,5,3,3,2 …..

- How can we determine the initial, transition and emission probabilities?



Initial probabilities

Q: assume we can observe the following sets of states:

AAABBAA
AABBBBB
BAABBAB

how can we learn the initial probabilities?

A: Maximum likelihood estimation

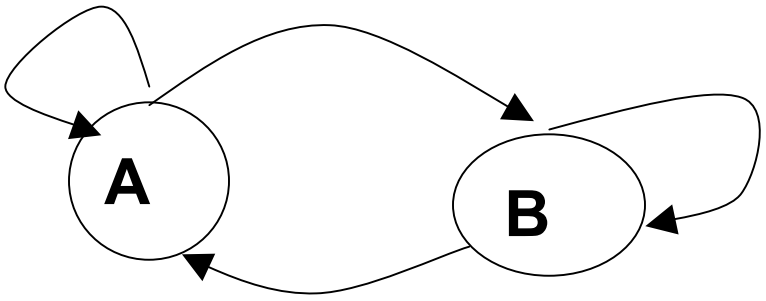
Find the initial probabilities π such that

$$\pi^* = \arg\max_\pi \prod_k \pi(q_1) \prod_{t=2}^{T} p(q_t \mid q_{t-1}) \Rightarrow$$

$$\pi^* = \arg\max_\pi \prod_k \pi(q_1)$$

k is the number of sequences avialable for training

π_A = #A/ (#A+#B)



Transition probabilities

Q: assume we can observe the set of states:

AAABBAAAABBBBBAAAABBBB

how can we learn the transition probabilities?

A: Maximum likelihood estimation

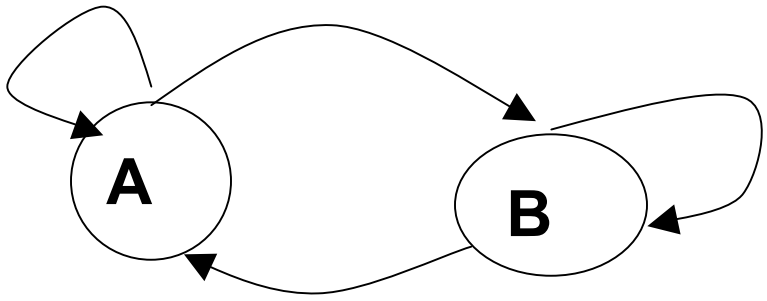
Find a transition matrix *a* such that

remember that we defined $a_{i,j}=p(q_t=s_j|q_{t-1}=s_i)$

$$a^* = \arg\max_a \prod_k \pi(q_1) \prod_{t=2}^{T} p(q_t \mid q_{t-1}) \Rightarrow$$

$$a^* = \arg\max_a \prod_{t=2}^{T} p(q_t \mid q_{t-1})$$

$$a_{A,B} = \#AB / (\#AB+\#AA)$$



Emission probabilities

Q: assume we can observe the set of states:

A A A B B A A A A B B B B B A A

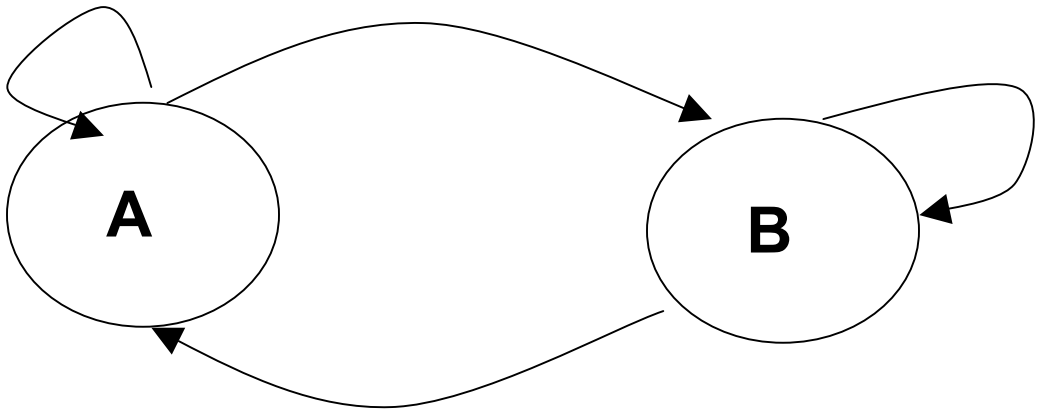
and the set of dice values

1 2 3 5 6 3 2 1 1 3 4 5 6 5 2 3

how can we learn the emission probabilities?

A: Maximum likelihood estimation

$b_A(5)$= #A5 / (#A1+#A2 + … +#A6)



Learning HMMs

- In most case we do not know the set of states (fully unsupervised)
- For these cases we can use an algorithm called expectation maximization (EM) to learn the HMM parameters

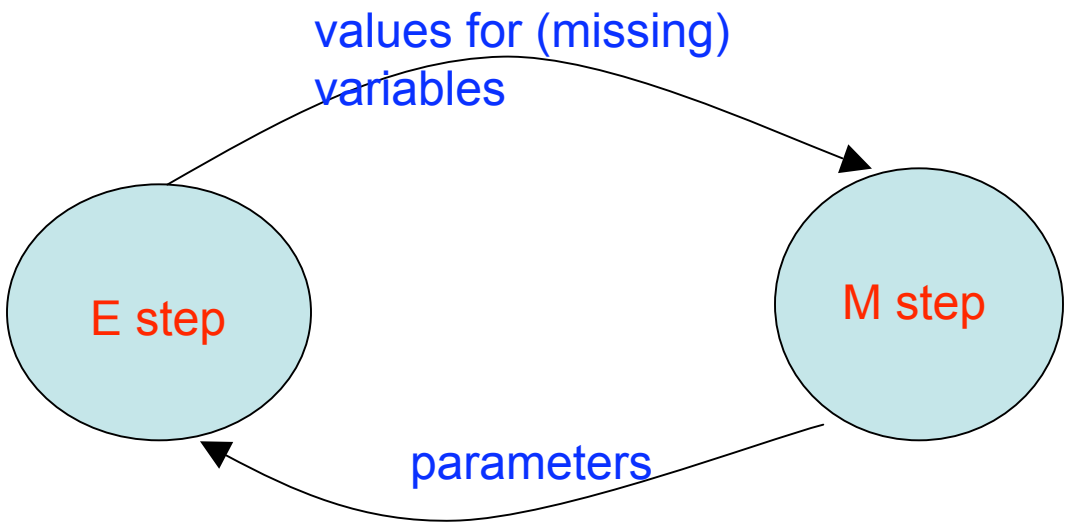
Expectation Maximization (EM)

- Appropriate for problems with ‘missing values’ for the variables.
- For example, in HMMs we usually do not observe the states

  - Other famous examples include clustering (variable: class membership) and Bayesian networks (for learning parameters and / or structure from incomplete data)

Expectation Maximization (EM)

- Two steps
- E step: Fill in the expected values for the missing variables
- M step: Regular maximum likelihood estimation (MLE) using the values computed in the E step and the values of the other variables
- Guaranteed to converge (though only to a local minima).



Expectation Maximization (EM)

- Two steps
- E step: Fill in the expected values for the missing variables
- M step: Regular maximum likelihood estimation (MLE) using the values computed in the E step and the values of the other variables
- Guaranteed to converge (though only to a local minima).

EM is another one of these vary general and highly popular algorithms. The key computational issue is how to derive the expectations in the E step.

Forward-Backward

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \ldots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable

$$\beta_t(i) = P(O_{t+1}, \cdots, O_n \mid s_t = i)$$

Forward-Backward

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \ldots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable

$$\beta_t(i) = P(O_{t+1}, \cdots, O_n \mid s_t = i) =$$

$$\sum_j a_{i,j} b_j(O_{t+1}) \beta_{t+1}(j)$$

Forward-Backward

- We already defined a *forward* looking variable

$$\alpha_t(i) = P(O_1 \ldots O_t \wedge q_t = s_i)$$

- We also need to define a *backward* looking variable

$$\beta_t(i) = P(O_{t+1}, \cdots, O_n \mid s_t = i)$$

- Using these two definitions we can show

P(A|B)=P(A,B)/P(B)

$$P(q_t = s_i \mid O_1, \cdots, O_n) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} S_t(i)$$

State and transition probabilities

- Probability of a state

$$P(q_t = s_i \mid O_1, \cdots, O_n) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} S_t(i)$$

- We can also derive a transition probability

$$P(q_t = s_i, q_{t+1} = s_j \mid o_1, \cdots, o_n) = S_t(i, j)$$

$$P(q_t = s_i, q_{t+1} = s_j \mid o_1, \cdots, o_n) =$$

$$= \frac{\alpha_t(i)P(q_{t+1} = s_j \mid q_t = s_i)P(o_{t+1} \mid q_{t+1} = s_j)\beta_{t+1}(j)}{\sum_j \alpha_t(j)\beta_t(j)} \stackrel{def}{=} S_t(i, j)$$

E step

- Compute $S_t(i)$ and $S_t(i,j)$ for all *t, i,* and *j* (*1≤t≤n*, *1≤i≤k*, *2≤j≤k*)

$$P(q_t = s_i \mid O_1, \cdots, O_n) = S_t(i)$$

$$P(q_t = s_i, q_{t+1} = s_j \mid o_1, \cdots, o_n) = S_t(i,j)$$

M step (1)

Compute transition probabilities:

$$a_{i,j} = \frac{\hat{n}(i,j)}{\sum_k \hat{n}(i,k)}$$

where

$$\hat{n}(i,j) = \sum_t S_t(i,j)$$

M step (2)

Compute emission probabilities (here we assume a multinomial distribution):

define:

$$B_k(j) = \sum_{t|o_t=j} S_t(k)$$

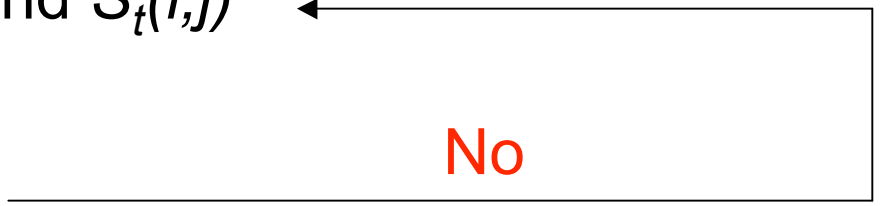
then

$$b_k(j) = \frac{B_k(j)}{\sum_i B_k(i)}$$

Complete EM algorithm for learning the parameters of HMMs (Baum-Welch)

- Inputs: 1 .Observations O_1 … O_n
  2. Number of states, model

1. Guess initial transition and emission parameters
2. Compute E step: $S_t(i)$ and $S_t(i,j)$
3. Compute M step
4. Convergence?
5. Output complete model



We did not discuss initial probability estimation. These can be deduced from multiple sets of observation (for example, several recorded customers for speech processing)

What you should know

- Why HMMs? Which applications are suitable?
- Infrence with and without observations
- Learning HMMs: EM algorithm (Baum-Welch)