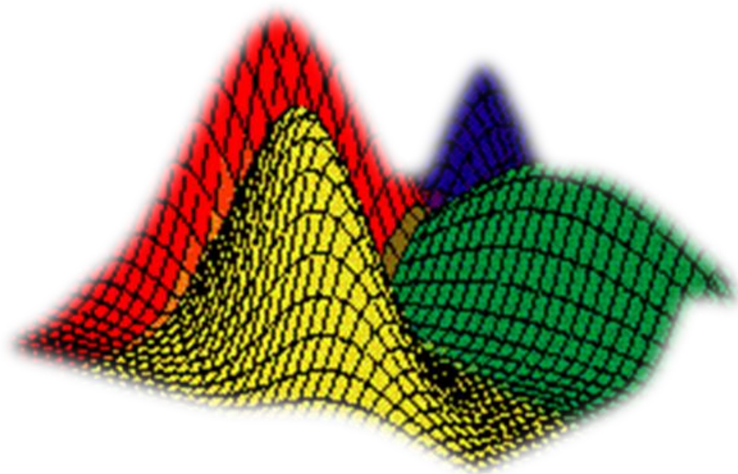
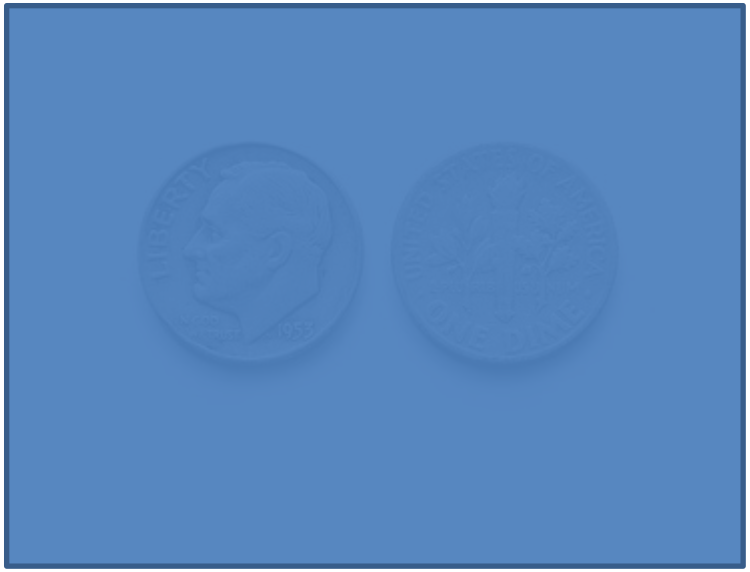


CSE 473: Pattern Recognition

Hidden Markov Model



Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$

Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

- Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

Markov Models

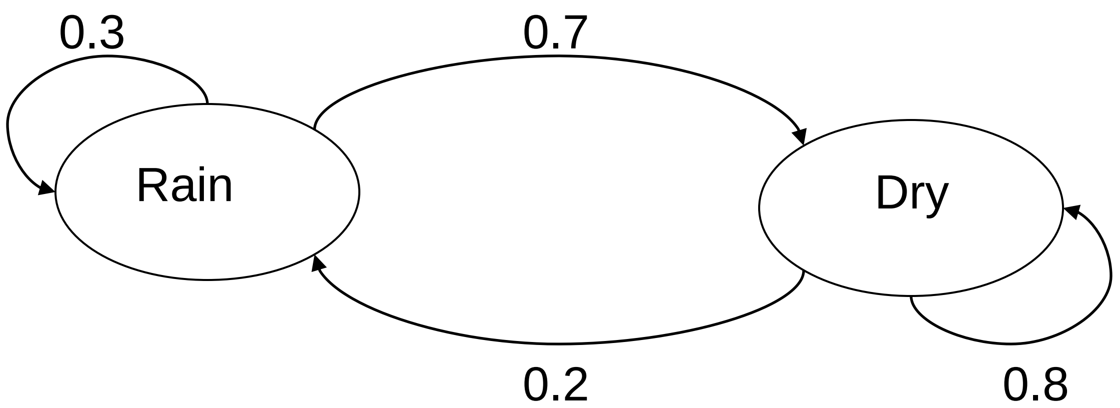
- Set of states: $\{s_1, s_2, \dots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \dots, s_{ik}, \dots$

- Markov chain property: probability of each subsequent state depends only on what was the previous state:

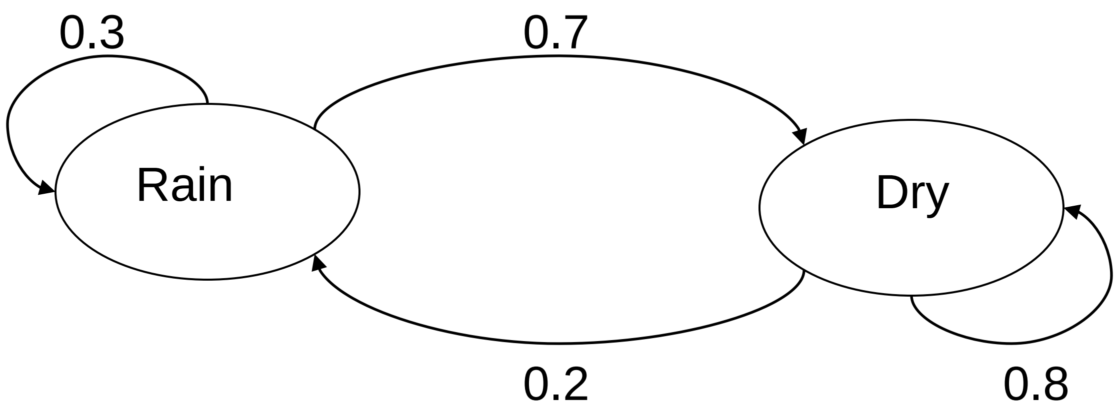
$$P(s_{ik} \mid s_{i1}, s_{i2}, \dots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

- To define Markov model, the following probabilities have to be specified: transition probabilities $a_{ij} = P(s_j \mid s_i)$ and initial probabilities $\pi_i = P(s_i)$

Example of Markov Model

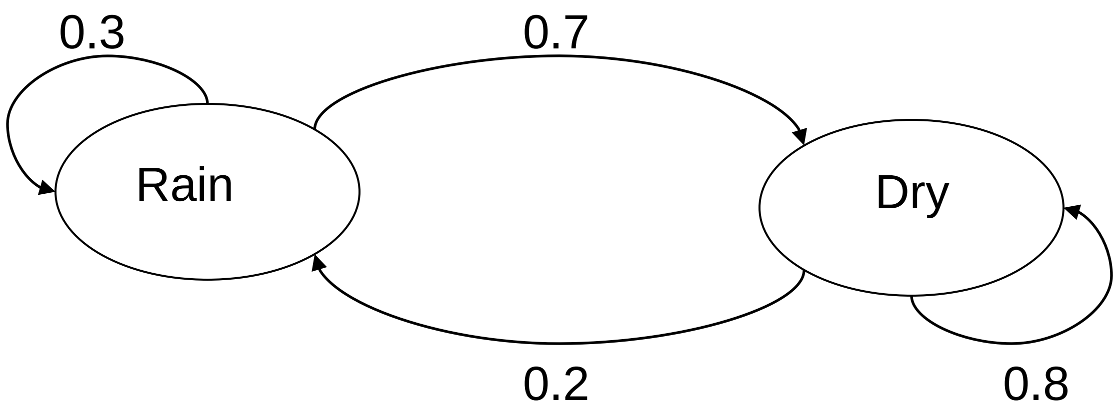


Example of Markov Model



- Two states : ‘Rain’ and ‘Dry’.

Example of Markov Model

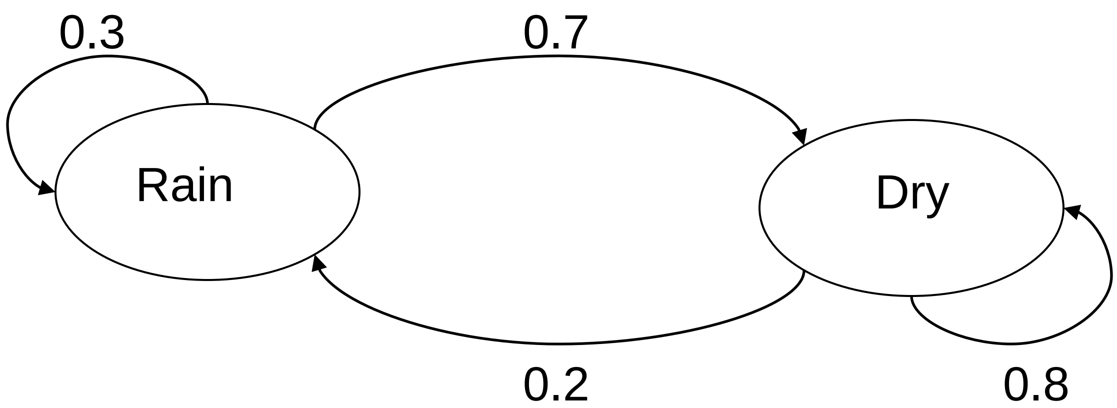


- Two states : ‘Rain’ and ‘Dry’.

- Transition probabilities:

  - P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 ,
  - P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8

Example of Markov Model



• Two states : ‘Rain’ and ‘Dry’.

•Transition probabilities:

- •P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 ,
- • P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8

• Initial probabilities: say P(‘Rain’)=0.4 , P(‘Dry’)=0.6 .

Calculation of sequence probability

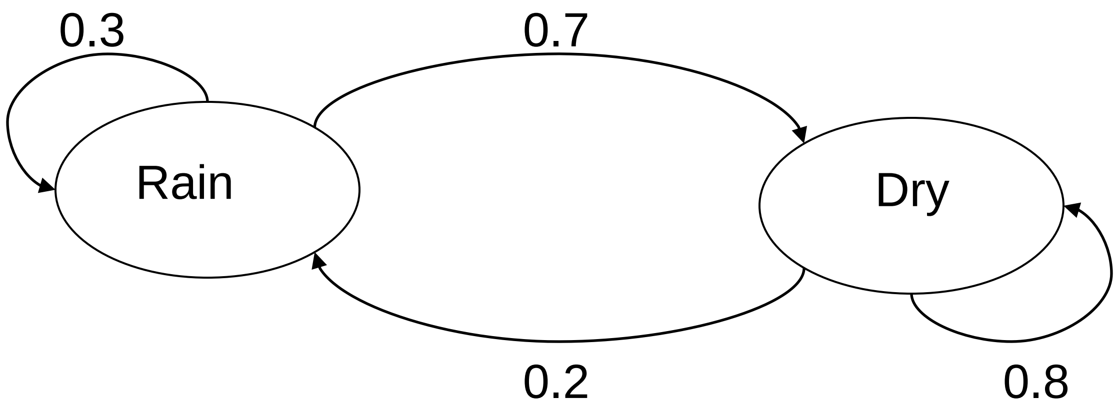
- By Markov chain property, probability of state sequence can be found by the formula:

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

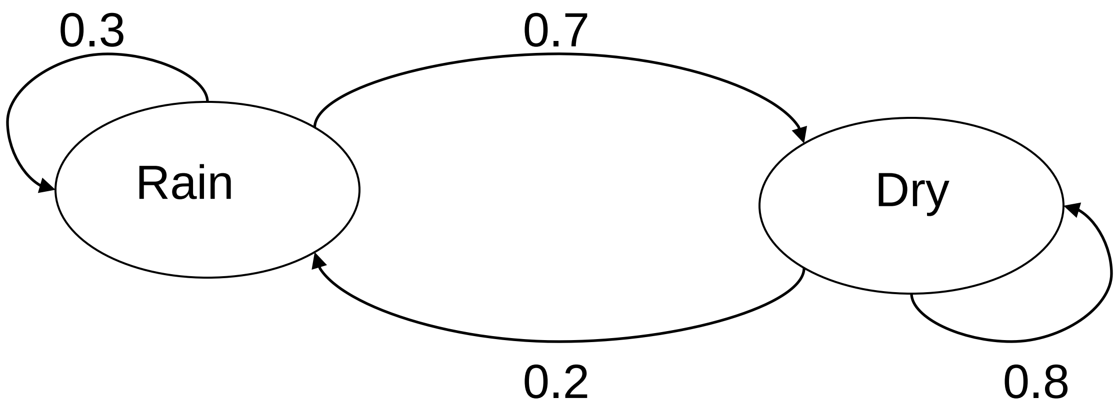
Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {'Dry','Dry','Rain',Rain'}.

P({'Dry','Dry','Rain',Rain'})

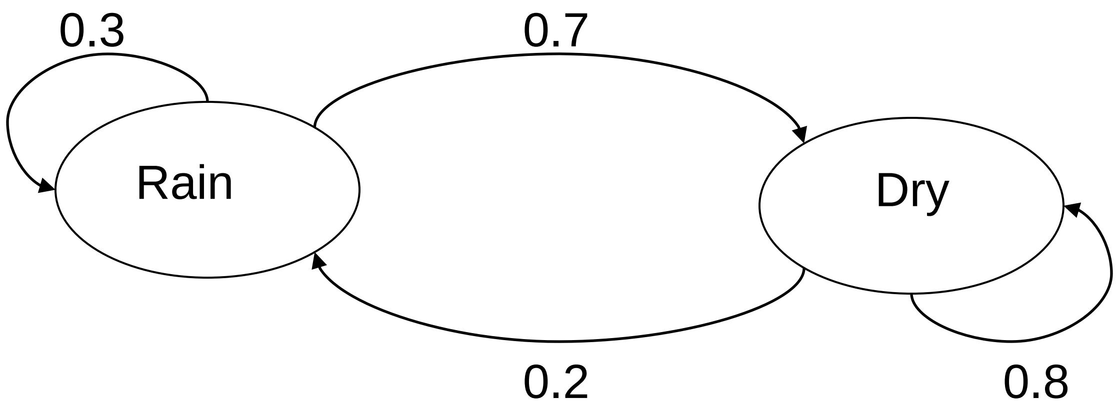
Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {‘Dry’,’Dry’,’Rain’,Rain’}.

P({‘Dry’,’Dry’,’Rain’,Rain’})
= P(‘Rain’|’Rain’) P(‘Rain’|’Dry’) P(‘Dry’|’Dry’) P(‘Dry’)

Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {‘Dry’,’Dry’,’Rain’,Rain’}.

$$P(\{\text{'Dry','Dry','Rain',Rain'}\})$$
$$= P(\text{'Rain'}|\text{'Rain'})\ P(\text{'Rain'}|\text{'Dry'})\ P(\text{'Dry'}|\text{'Dry'})\ P(\text{'Dry'})$$
$$= 0.3*0.2*0.8*0.6$$

Hidden Markov Model

Hidden Markov Model

Hidden Markov Model

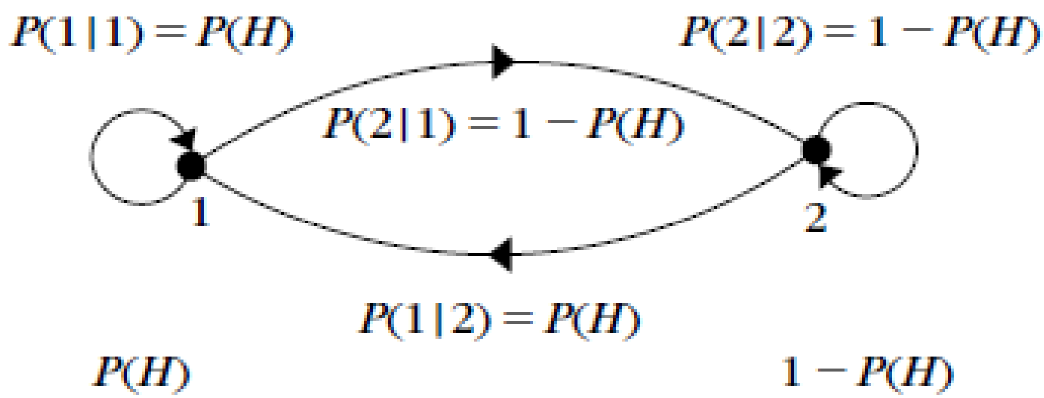
H T H H T T T H H H . . . . .

Hidden Markov Model

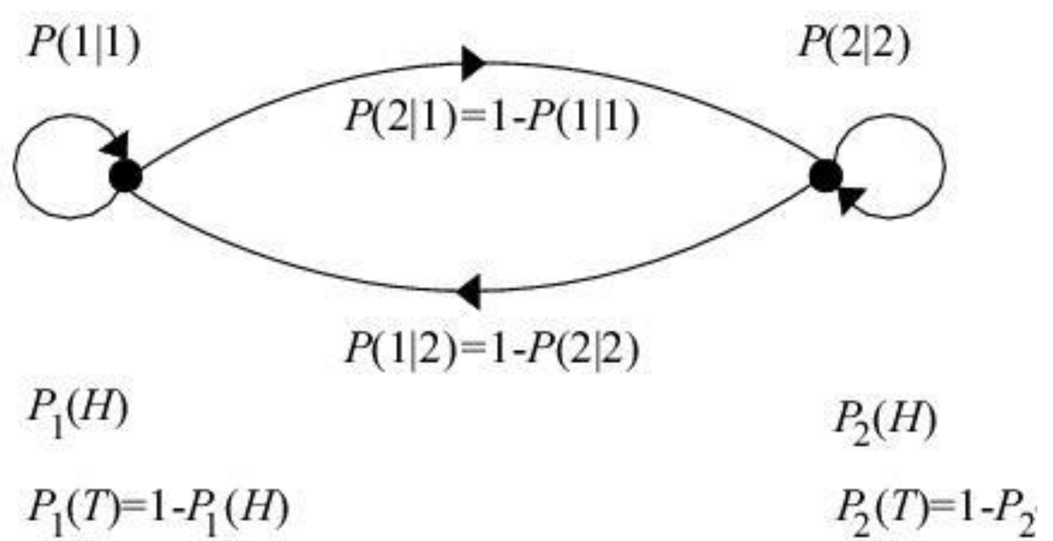
H T H H T T T H H H . . . . .

Can we guess which coin is tossed at different times?

Not A Hidden Markov Model



Hidden Markov Model



Hidden Markov models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$

- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

- Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

Hidden Markov models

- States are not visible, but each state randomly generates one of M observations (or visible states)

$$\{v_1, v_2, \ldots, v_M\}$$

Hidden Markov models

• States are not visible, but each state randomly generates one of M observations (or visible states)

$$\{v_1, v_2, \ldots, v_M\}$$

•To define hidden Markov model, the following probabilities have to be specified:

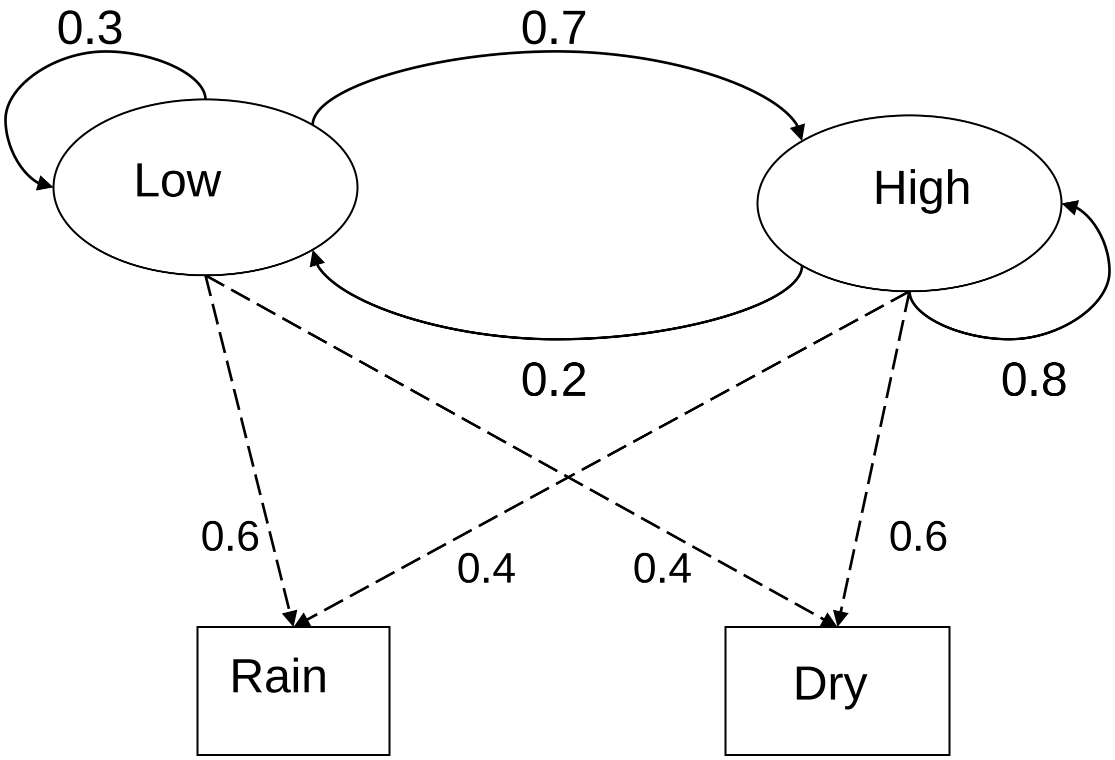
•matrix of transition probabilities $A=(a_{ij})$, $a_{ij}= P(s_j \mid s_i)$

•matrix of observation probabilities $B=(b_i(v_m))$,
where, $b_i(v_m) = P(v_m|s_i)$

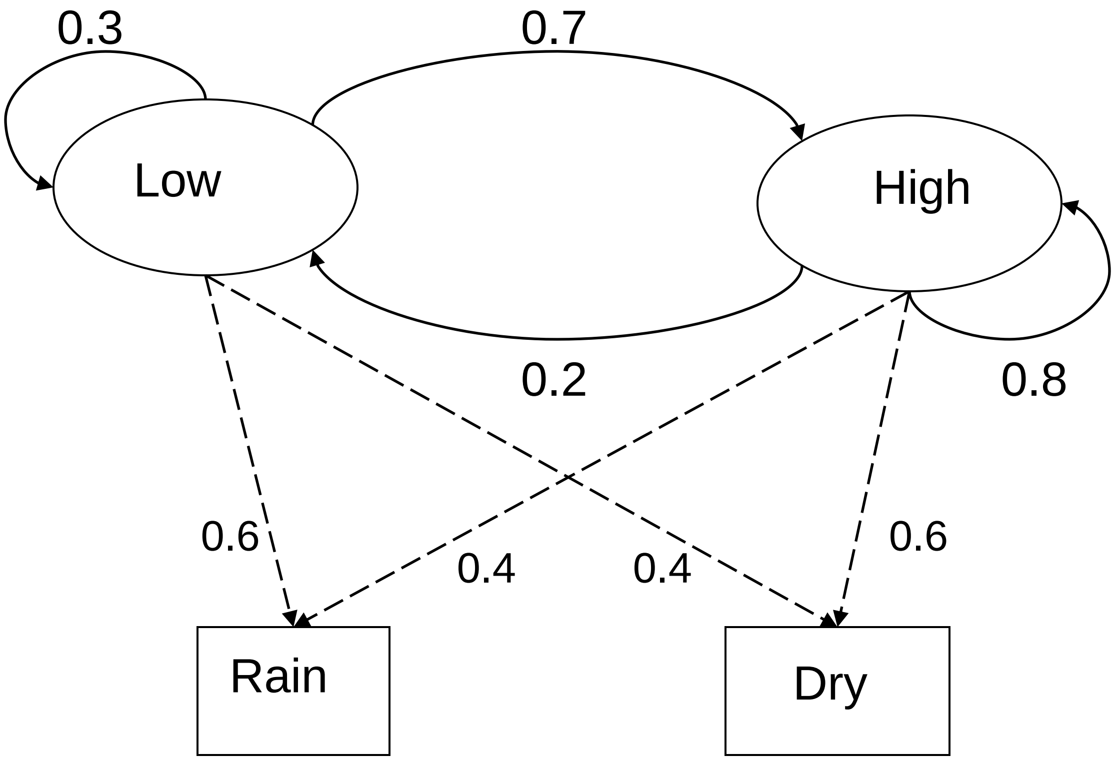
•vector of initial probabilities $\pi=(\pi_i)$, $\pi_i = P(s_i)$

•Model is represented by $M=(A, B, \pi)$.

Example of Hidden Markov Model



Example of Hidden Markov Model



- Initial probabilities: say P('Low')=0.4 , P('High')=0.6 .

Example of Hidden Markov Model

- Two states : ‘Low’ and ‘High’ atmospheric pressure.
- Two observations : ‘Rain’ and ‘Dry’.
- Transition probabilities: $P(\text{‘Low’}|\text{‘Low’})=0.3$, $P(\text{‘High’}|\text{‘Low’})=0.7$, $P(\text{‘Low’}|\text{‘High’})=0.2$, $P(\text{‘High’}|\text{‘High’})=0.8$
- Observation probabilities : $P(\text{‘Rain’}|\text{‘Low’})=0.6$, $P(\text{‘Dry’}|\text{‘Low’})=0.4$, $P(\text{‘Rain’}|\text{‘High’})=0.4$, $P(\text{‘Dry’}|\text{‘High’})=0.3$.
- Initial probabilities: say $P(\text{‘Low’})=0.4$, $P(\text{‘High’})=0.6$.

Calculation of observation sequence probability

•Suppose we want to calculate a probability of a sequence of observations in our example, {'Dry','Rain'}.

•Consider all possible hidden state sequences:

P({'Dry','Rain'}) = P({'Dry','Rain'} , {'Low','Low'})
+P({'Dry','Rain'} , {'Low','High'})
+ P({'Dry','Rain'} , {'High','Low'})
+ P({'Dry','Rain'} , {'High','High'})

Calculation of observation sequence probability

- P({‘Dry’,’Rain’}) = P({‘Dry’,’Rain’} , {‘Low’,’Low’})
  + P({‘Dry’,’Rain’} , {‘Low’,’High’})
  + P({‘Dry’,’Rain’} , {‘High’,’Low’})
  + P({‘Dry’,’Rain’} , {‘High’,’High’})

where first term is :
P({‘Dry’,’Rain’} , {‘Low’,’Low’})=
P({‘Dry’,’Rain’} | {‘Low’,’Low’}) P({‘Low’,’Low’}) =
P(‘Dry’|’Low’)P(‘Rain’|’Low’) P(‘Low’)P(‘Low’|’Low)
= 0.4*0.4*0.6*0.4*0.3

Main issues using HMMs

- **Evaluation problem.**

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \ldots o_K$, calculate the probability that model M has generated sequence O.

$O=o_1 \ldots o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

Main issues using HMMs

•**Decoding problem.**

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\ o_2\ ...\ o_K$, calculate the most likely sequence of hidden states s_i that produced this observation sequence O.

$O=o_1...o_K$ denotes a sequence of observations $o_k \in \{v_1,\ldots,v_M\}$.

Main issues using HMMs

• **Learning problem.**

Given some training observation sequences $O=o_1\, o_2 \ldots o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

$O=o_1 \ldots o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

The evaluation Problem

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\ o_2\ ...\ o_K$, calculate the probability that model M has generated sequence O.

where, $O=o_1...o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

The evaluation Problem

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \,...\, o_K$, calculate the probability that model M has generated sequence O.

where, $O=o_1...o_K$ *denotes a sequence of observations* $o_k \in \{v_1, \ldots, v_M\}$.

Alternately, find $P(O|M)$ or $P(o_1\, o_2 \,...\, o_K|M)$

The evaluation Problem

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \ldots o_K$, calculate the probability that model M has generated sequence O.

where, $O=o_1 \ldots o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

Alternately, find $P(O|M)$ or $P(o_1\, o_2 \ldots o_K|M)$

For simplicity we write it as $P(O)$ or $P(o_1\, o_2 \ldots o_K)$

The evaluation Problem

Objective:

- find P(O) or P(o_1 o_2 ... o_K)

$$P(O) = \sum_i p(O, \Omega_i)$$

where, Ω_i is a possible state sequence

$$S_{i_1}, S_{i_2}, \ldots, S_{i_m}, \ldots, S_{i_K}$$

The evaluation Problem

Objective:

- find $P(O)$ or $P(o_1\ o_2 \ldots o_K)$

$$P(O) = \sum_i p(O, \Omega_i)$$

where, Ω_i is a possible state sequence

$$S_{i_1}, S_{i_2}, \ldots, S_{i_m}, \ldots, S_{i_K}$$

There are N^K possible state sequences!!

The evaluation Problem

Objective:

- find P(O) or P(o_1 o_2 ... o_K)

$$P(O) = \sum_i p(O, \Omega_i)$$

where, Ω_i is a possible state sequence

$$S_{i_1}, S_{i_2}, \ldots, S_{i_m}, \ldots, S_{i_K}$$

Complexity is O(N^K)

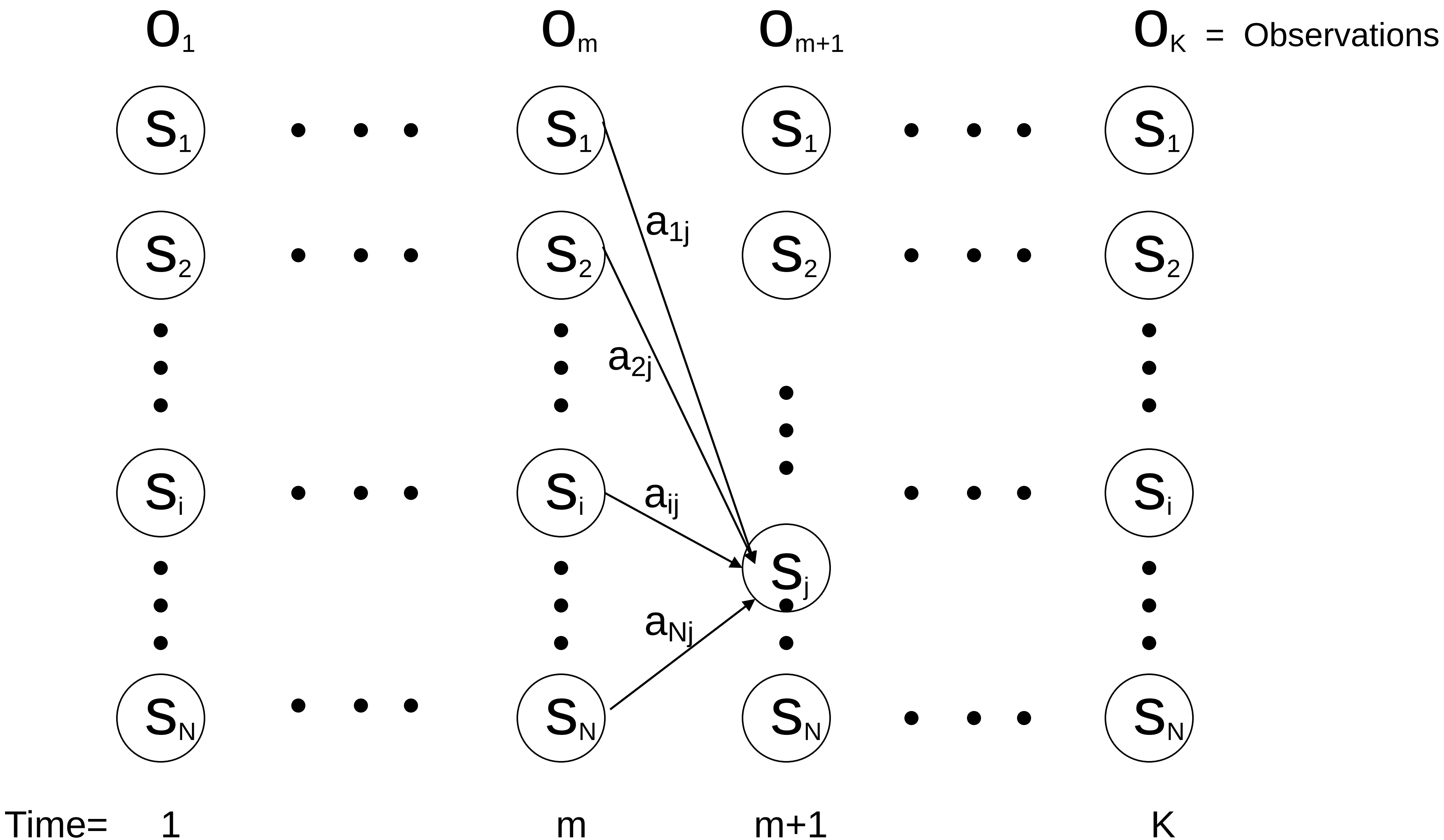
Alternate Solution to The evaluation Problem

- Use **Forward-Backward HMM algorithms** for efficient calculations.

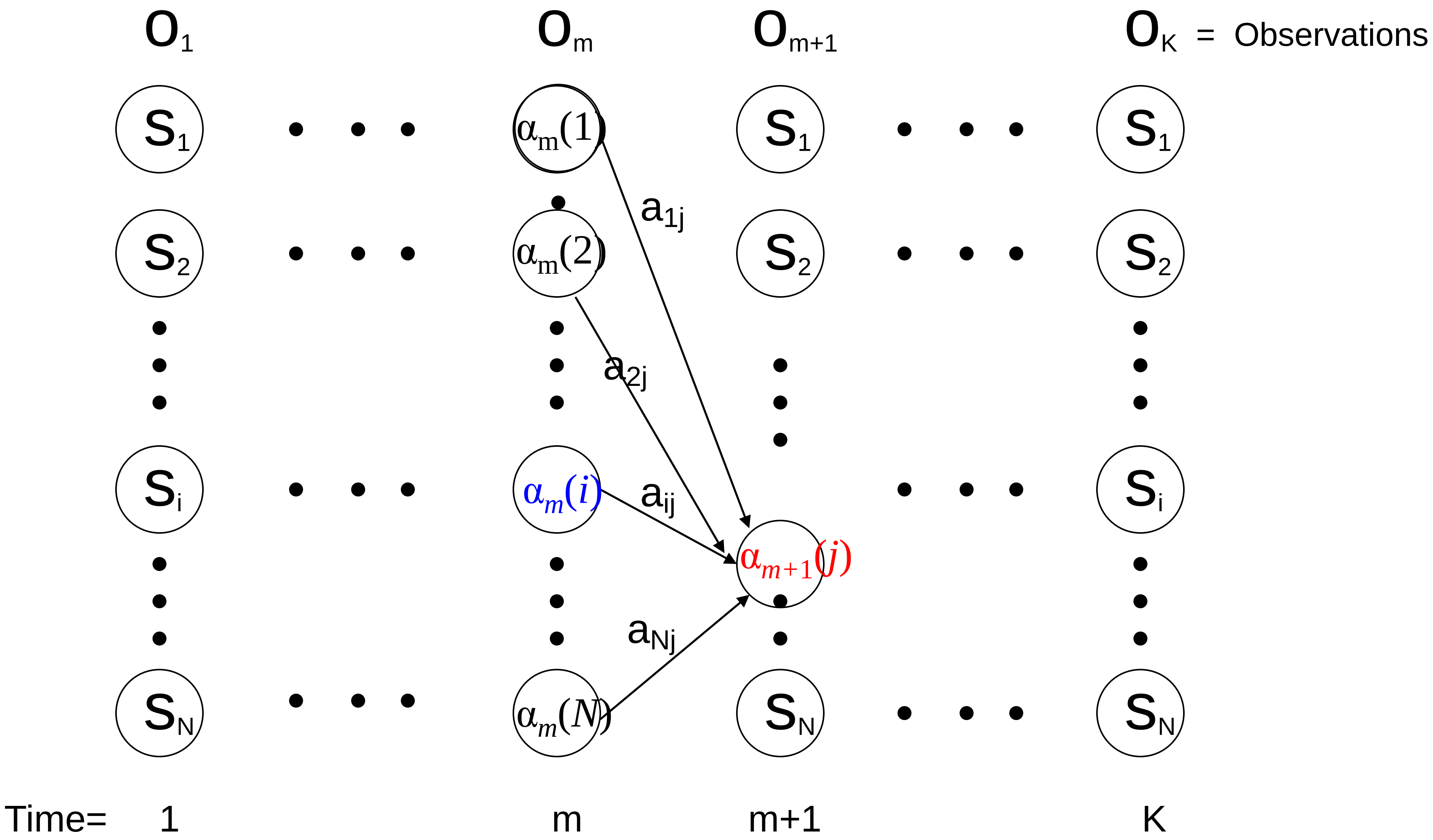
- Define the forward variable $\alpha_m(i)$ as the joint probability of
  - the partial observation sequence $o_1\, o_2 \ldots o_m$ and
  - the hidden state at time m is s_i :

$$\alpha_m(i) = P(o_1, o_2 \ldots o_m, q_m = s_i)$$

Trellis representation of an HMM



Trellis representation of an HMM



Trellis representation of an HMM

Therefore, we can write,

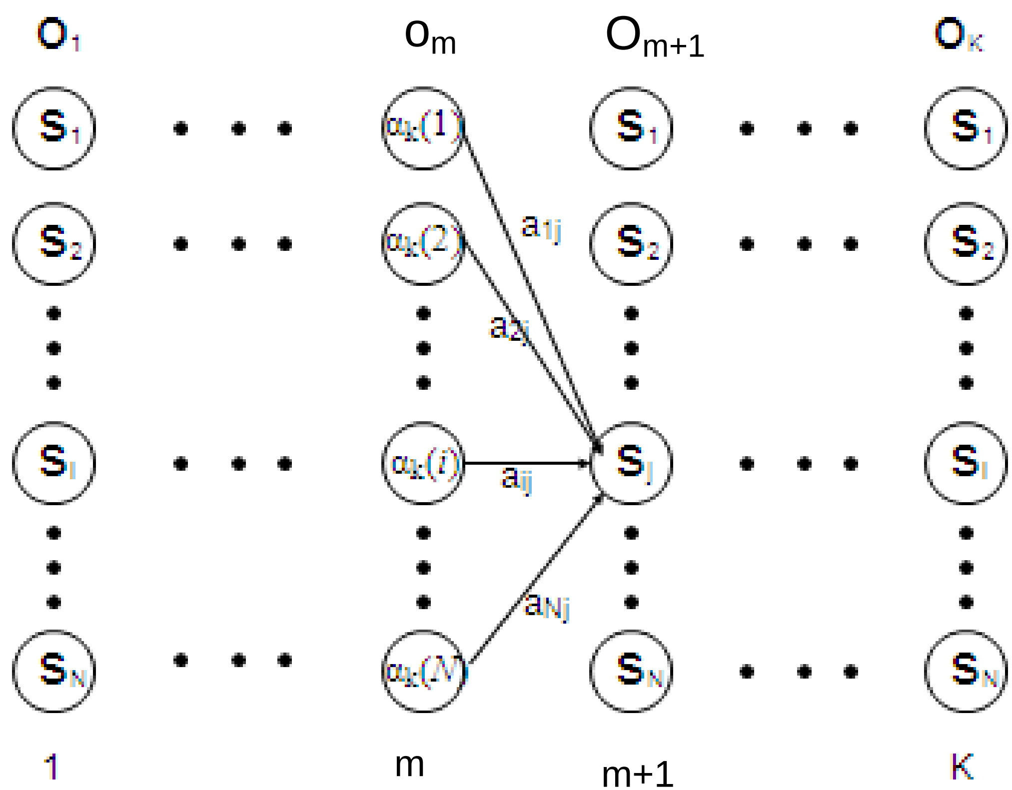
$\alpha_{m+1}(j) = P(o_1\, o_2 \ldots\, o_{m+1},\, q_{m+1} = s_j)$

$= \Sigma_i\, P(o_1\, o_2 \ldots\, o_{m+1},\, q_m = s_i,\, q_{m+1} = s_j)$

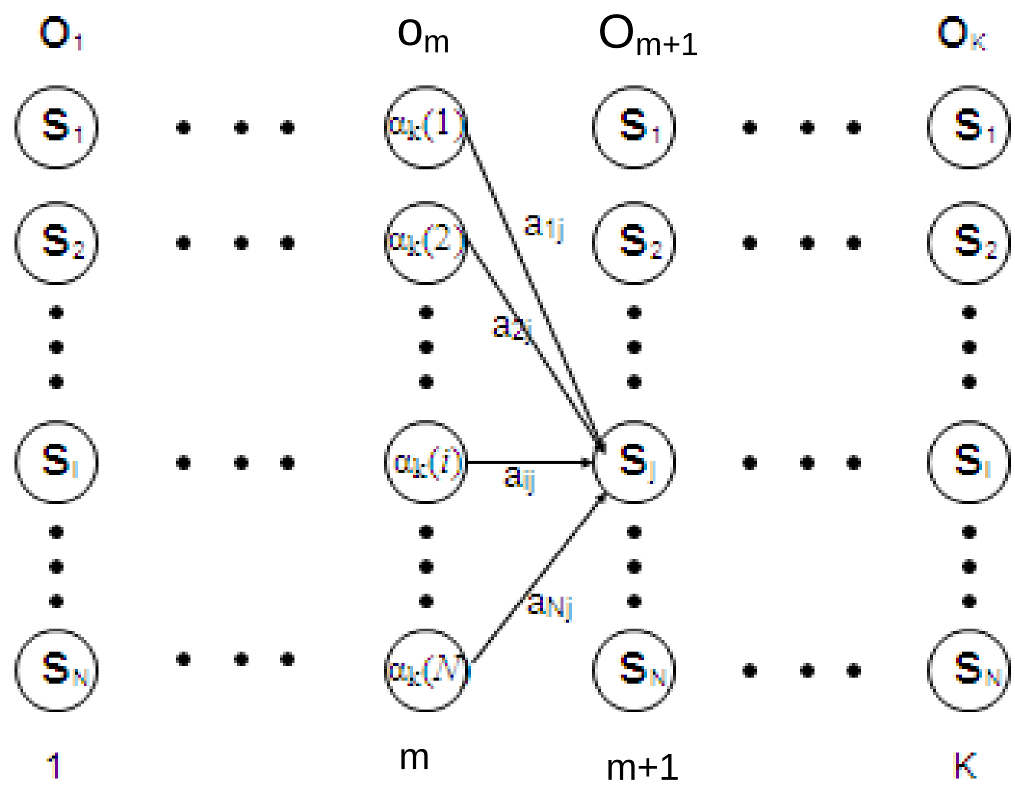
$= \Sigma_i\, P(o_1\, o_2 \ldots o_m,\, q_m = s_i)\, a_{ij}\, b_j(o_{m+1})$

$= [\Sigma_i\, \alpha_m(i)\, a_{ij}\,]\, b_j(o_{m+1})$,

for $1 <= j <= N$, $1 <= m <= K-1$.



Trellis representation of an HMM



Now $P(o_1\, o_2 \ldots o_K)$

can be written as $\Sigma_i\, P(o_1\, o_2 \ldots o_K\,,\, q_K = s_i) = \Sigma_i\, \alpha_K(i)$

Forward recursion for HMM

- Initialization:

$$\alpha_1(i) = P(o_1, q_1 = s_i) = \pi_i \, b_i(o_1), \quad 1 <= i <= N.$$

- Forward recursion:

$$\alpha_{m+1}(j) = [\Sigma_i \, \alpha_m(i) \, a_{ij}] \, b_j(o_{m+1}), \qquad 1 <= j <= N, \ 1 <= m <= K-1.$$

- Termination:

$$P(o_1 \, o_2 \ldots o_K) = \Sigma_i \, \alpha_K(i)$$

- Complexity :

  N^2K operations.

Backward recursion for HMM

- Define the backward variable $\beta_m(j)$ as the conditional probability of
  - the partial observation sequence $o_{m+1}\ o_{m+2} \dots o_K$
  - given that the hidden state at time m is s_j :

$$\beta_m(j) = P(o_{m+1}\ o_{m+2} \dots o_K \mid q_m = s_j)$$

Backward recursion for HMM

- Define $\beta_m(j)$ in terms of $\beta_{m+1}(i)$'s:

- $\beta_{m+1}(i)$ is the conditional probability of

  - the partial observation sequence $o_{m+2}\, o_{m+3} \ldots o_K$
  - given that the hidden state at time m+1 is s_i :

$$\beta_{m+1}(i) = P(o_{m+2}\, o_{m+3} \ldots o_K \,|\, q_{m+1} = s_i)$$

Backward recursion for HMM

- Define $\beta_m(j)$ in terms of $\beta_{m+1}(i)$'s: the probability of

where $\beta_{m+1}(i) = P(o_{m+2}\, o_{m+3} \dots o_K \mid q_{m+1} = s_i)$

Now,

$\beta_m(j) = P(o_{m+1}\, o_{m+2} \dots o_K \mid q_m = s_j)$

$= \Sigma_i\, P(o_{m+1}\, o_{m+2} \dots o_K,\, q_{m+1} = s_i \mid q_m = s_j)$

$= \Sigma_i\, P(o_{m+2}\, o_{m+3} \dots o_K \mid q_{m+1} = s_i)\, a_{ji}\, b_i(o_{m+1})$

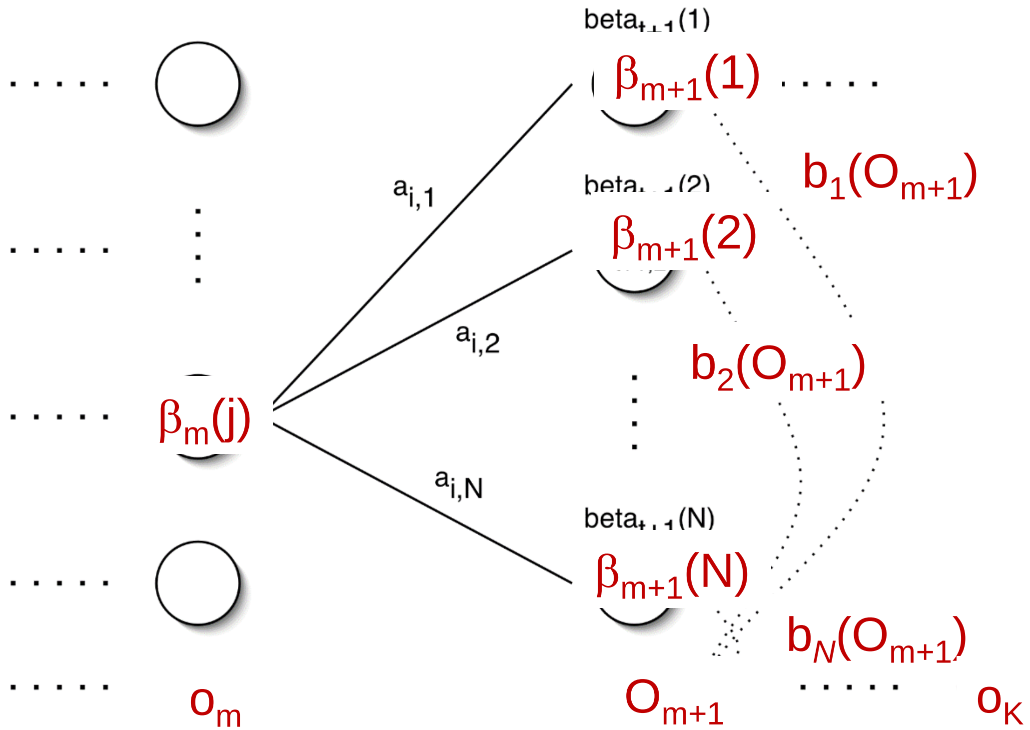
$= \Sigma_i\, \beta_{m+1}(i)\, a_{ji}\, b_i(o_{m+1})$, $\quad 1 \le j \le N,\ 1 \le m \le K-1.$

Backward recursion for HMM

$= \Sigma_i \, P(o_{m+1}\, o_{m+2} \ldots o_K \,, q_{m+1}= s_i \mid q_m= s_j)$

$= \Sigma_i \, P(o_{m+2}\, o_{m+3} \ldots o_K \mid q_{m+1}= s_i)\, a_{ji}\, b_i (o_{m+1})$

$= \Sigma_i \, \beta_{m+1}(i)\, a_{ji}\, b_i (o_{m+1})$, $1<=j<=N, 1<=m<=K-1.$



Backward recursion for HMM

- Initialization:

  $\beta_K(i)= 1$, $1<=i<=N$.

- Backward recursion:

  $\beta_m(j)= \Sigma_i \beta_{m+1}(i)\, a_{ji}\, b_i(o_{m+1})$, $1<=j<=N, 1<=m<=K-1$.

- Termination:

  $P(o_1\, o_2 \ldots o_K) = \Sigma_i P(o_1\, o_2 \ldots o_K , q_1= s_i) =$

  $\Sigma_i P(o_1\, o_2 \ldots o_K \,|q_1= s_i)\, P(q_1= s_i) = \Sigma_i \beta_1(i)\, b_i(o_1)\, \pi_I$

Main issues using HMMs (2)

•**Decoding problem.**

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \dots o_K$, calculate the most likely sequence of hidden states s_i that produces this observation sequence O.

$O=o_1 \dots o_K$ denotes a sequence of observations $o_k \in \{v_1, \dots, v_M\}$.

Decoding problem

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\,o_2\,...\,o_K$, calculate the most likely sequence of hidden states s_i that produces this observation sequence O.

We want to find:
the state sequence $Q= q_1\ldots q_K$ maximizing

$$P(Q \mid o_1\,o_2\,...\,o_K)$$

Decoding problem

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \ldots o_K$, calculate the most likely sequence of hidden states s_i that produces this observation sequence O.

We want to find:

the state sequence $Q= q_1 \ldots q_K$ maximizing

$$P(Q \mid o_1\, o_2 \ldots o_K)$$

or, equivalently

$$P(Q , o_1\, o_2 \ldots o_K)$$

Decoding problem

- Find max value of $P(Q, o_1 o_2 \ldots o_K)$

Brute Force Method:

Try for all possible sequences of states

N^K possible sequences for Q

Decoding problem

- Use efficient **Viterbi algorithm** instead

- Define variable $\delta_m(i)$ as the maximum probability of
  - producing observation sequence $o_1\ o_2 \ldots o_m$
  - when moving along any hidden state sequence $q_1 \ldots q_{m-1}$ and
  - getting into $q_m = s_i$

Decoding problem

- Use efficient **Viterbi algorithm** instead

- Define variable $\delta_m(i)$ as the maximum probability of
  - producing observation sequence $o_1\, o_2 \ldots o_m$
  - when moving along any hidden state sequence $q_1 \ldots q_{m-1}$ and
  - getting into $q_m = s_i$

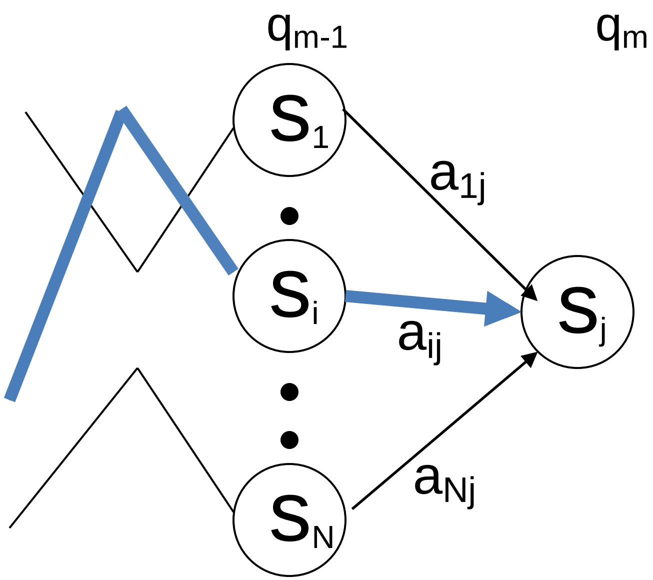
Therefore, $\delta_m(i) = \max P(q_1 \ldots q_{m-1}, q_m = s_i, o_1\, o_2 \ldots o_m)$

where max is taken over all possible paths $q_1 \ldots q_{m-1}$.

Decoding problem

- General idea:

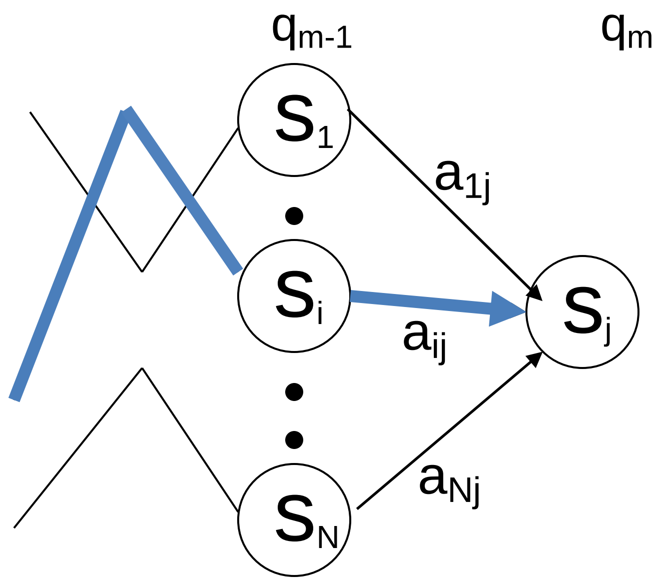
if best path ending in $q_m = s_j$ goes through $q_{m-1} = s_i$ then it should coincide with best path ending in $q_{m-1} = s_i$.



Decoding problem

- General idea:

  if best path ending in $q_m = s_j$ goes through $q_{m-1} = s_i$ then it should coincide with best path ending in $q_{m-1} = s_i$.

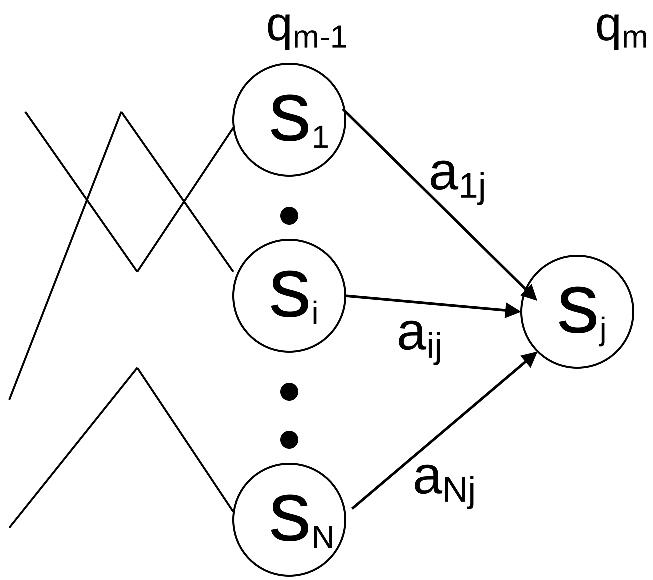


- $\delta_m(j) = \max P(q_1 \dots q_{m-1}, q_m = s_j, o_1 o_2 \dots o_m)$

Decoding problem

- General idea:

  if best path ending in $q_m = s_j$ goes through $q_{m-1} = s_i$ then it should coincide with best path ending in $q_{m-1} = s_i$.



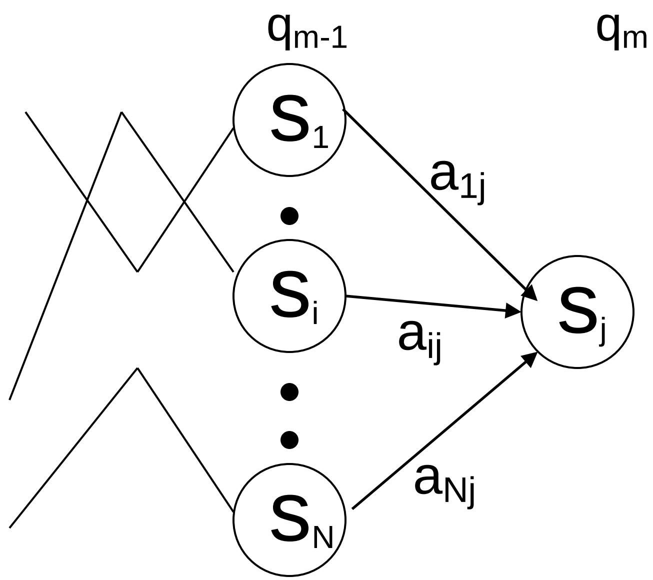
- $\delta_m(j) = \max P(q_1 \ldots q_{m-1}, q_m = s_j, o_1 o_2 \ldots o_m)$

  $= \max_i [\, a_{ij}\, b_j(o_m) \;\; \max P(q_1 \ldots q_{m-1} = s_i, o_1 o_2 \ldots o_{m-1}) \,]$

Decoding problem

- General idea:

  if best path ending in $q_m = s_j$ goes through $q_{m-1} = s_i$ then it should coincide with best path ending in $q_{m-1} = s_i$.



- $\delta_m(j) = \max P(q_1 \ldots q_{m-1}, q_m = s_j, o_1 o_2 \ldots o_m)$

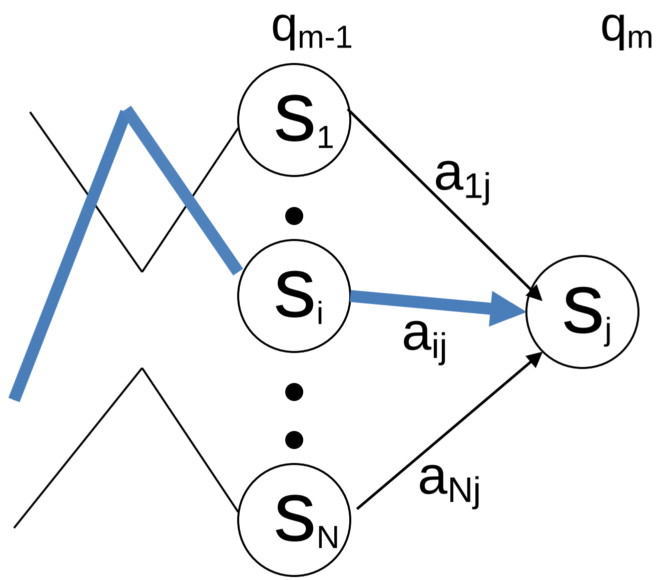
  $= \max_i [\, a_{ij}\, b_j(o_m)\ \max P(q_1 \ldots q_{m-1} = s_i, o_1 o_2 \ldots o_{m-1})\,]$

  $= \max_i [\, a_{ij}\, b_j(o_m)\, \delta_{m-1}(i)]$

Decoding problem

- General idea:

if best path ending in $q_m = s_j$ goes through $q_{m-1} = s_i$ then it should coincide with best path ending in $q_{m-1} = s_i$.



- To backtrack best path, keep info that predecessor of s_j was s_i.

Decoding problem

- Initialization:

$$\delta_1(i) = \max P(q_1 = s_i\ ,\ o_1) = \pi_i\, b_i\,(o_1)\ ,\ 1 <= i <= N$$

- Forward recursion:

$$\delta_m(j) = \max_i \left[\, a_{ij}\, b_j\,(o_m)\, \delta_{m-1}(i)\, \right],\qquad 1 <= j <= N,\ 2 <= m <= K.$$

- Termination: choose best path ending at time K

$$\max_i \left[\, \delta_K(i)\, \right]$$

- Backtrack best path.

Issues in HMMs (3)

•**Learning/Training problem.**

Given some training observation sequences $O=o_1 o_2 \ldots o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

$O=o_1,\ldots, o_m,\ldots,o_K$ denotes a sequence of observations where, $o_m \in \{v_1, \ldots, v_M\}$.

Learning/Training Problem

Given some training observation sequences $O=o_1\, o_2 \ldots o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

- There is no algorithm producing optimal parameter values.

Learning/Training Problem

Given some training observation sequences $O=o_1 o_2 \dots o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

- There is no algorithm producing optimal parameter values.

- Use iterative Expectation-Maximization (EM) algorithm to find local maximum of $P(O \mid M)$ - **Baum-Welch algorithm.**

Learning/Training Problem

Idea of EM :

Initialization: Assume initial value of A, B, π

and calculate $P(O \mid M)$

E-step:

Estimate new values of the model parameters: A, B, π

from the previous values of A, B, π

M-step:

Find $P(O \mid M)$ with the new values of A, B, π

Repeat these steps until $P(O \mid M)$ declines

Learning/Training Problem

- we need to calculate the following parameters:

$$a_{ij} = P(s_j \mid s_i) = \frac{\text{No. of transitions from state } s_i \text{ to state } s_j}{\text{No. of transitions out of state } s_i}$$

$$b_i(v_n) = P(v_n \mid s_i) = \frac{\text{No. of times observation } v_n \text{ occurs in state } s_i}{\text{No. of times in state } s_i}$$

$$\pi(i) = P(s_i) = \frac{\text{No. of times in state } s_i \text{ at time } m = 1}{\text{No. of times in any state at time } m = 1}$$

Baum-Welch algorithm

- the algorithm estimates the expected value::

$$a_{ij} = P(s_j \mid s_i) = \frac{\text{Expected No. of transitions from state } s_i \text{ to state } s_j}{\text{Expected No. of transitions out of state } s_i}$$

$$b_i(v_n) = P(v_n \mid s_i) = \frac{\text{Expected No. of times observation } v_n \text{ occurs in state } s_i}{\text{Expected No. of times in state } s_i}$$

$$\pi(i) = P(s_i) = \frac{\text{Expected No. of times in state } s_i \text{ at time } m = 1}{\text{Expected No. of times in any state at time } m = 1}$$

Baum-Welch algorithm

- Define variable $\xi_m(i,j)$ as the probability of being in state s_i at time m and in state s_j at time m+1, given the observation sequence $o_1\, o_2 \ldots o_K$.

$$\xi_m(i,j) = P(q_m = s_i\,,\, q_{m+1} = s_j \mid o_1\, o_2 \ldots o_K)$$

Baum-Welch algorithm

- Define variable $\xi_m(i,j)$ as the probability of being in state s_i at time m and in state s_j at time m+1, given the observation sequence $o_1\, o_2 \dots o_K$.

$$\xi_m(i,j) = P(q_m = s_i\,,\, q_{m+1} = s_j \mid o_1\, o_2 \dots o_K)$$

$$\xi_m(i,j) = \frac{P(q_m = s_i\,,\, q_{m+1} = s_j\,,\, o_1\, o_2 \dots o_K)}{P(o_1\, o_2 \dots o_K)}$$

Baum-Welch algorithm

- Define variable $\xi_m(i,j)$ as the probability of being in state s_i at time m and in state s_j at time m+1, given the observation sequence $o_1\, o_2 \ldots o_K$.

$$\xi_m(i,j) = P(q_m = s_i\,,\, q_{m+1} = s_j \mid o_1\, o_2 \ldots o_K)$$

$$\xi_m(i,j) = \frac{P(q_m = s_i\,,\, q_{m+1} = s_j\,,\, o_1\, o_2 \ldots o_K)}{P(o_1\, o_2 \ldots o_K)}$$

$$= \frac{P(q_m = s_i\,,\, o_1\, o_2 \ldots o_m)\; a_{ij}\; b_j(o_{m+1})\; P(o_{m+2} \ldots o_K \mid q_{m+1} = s_j)}{P(o_1\, o_2 \ldots o_K)}$$

Baum-Welch algorithm

- Define variable $\xi_m(i,j)$ as the probability of being in state s_i at time m and in state s_j at time $m+1$, given the observation sequence $o_1\, o_2 \ldots o_K$.

$$\xi_m(i,j) = P(q_m = s_i\ , q_{m+1} = s_j \mid o_1\, o_2 \ldots o_K)$$

$$= \frac{P(q_m = s_i\ , o_1\, o_2 \ldots o_m)\ a_{ij}\ b_j(o_{m+1})\ P(o_{m+2} \ldots o_K \mid q_{m+1} = s_j)}{P(o_1\, o_2 \ldots o_K)}$$

$$= \frac{\alpha_m(i)\ a_{ij}\ b_j(o_{m+1})\ \beta_{m+1}(j)}{P(o_1\, o_2 \ldots o_K)}$$

Baum-Welch algorithm

- Define variable $\gamma_m(i)$ as the probability of being in state s_i at time m, given the observation sequence $o_1\, o_2 \ldots o_K$.

$$\gamma_m(i) = P(q_m = s_i \mid o_1\, o_2 \ldots o_K)$$

$$\gamma_m(i) = \frac{P(q_m = s_i,\, o_1\, o_2 \ldots o_k)}{P(o_1\, o_2 \ldots o_k)} = \frac{\alpha_m(i)\, \beta_m(i)}{P(o_1\, o_2 \ldots o_k)}$$

Baum-Welch algorithm

- We calculated $\xi_m(i,j) = P(q_m = s_i\ , q_{m+1} = s_j \mid o_1\, o_2 \ldots o_K)$
  and $\gamma_m(i) = P(q_m = s_i \mid o_1\, o_2 \ldots o_K)$

- Expected number of transitions from state s_i to state s_j
  $= \Sigma_m\, \xi_m(i,j)$

- Expected number of transitions out of state s_i $= \Sigma_m\, \gamma_m(i)$

- Expected number of times observation v_n occurs in state s_i =
  $= \Sigma_m\, \gamma_m(i)$, m is such that $o_m = v_n$

Baum-Welch algorithm: E-Step

Estimate the expected values as

$$a_{ij} = \frac{\text{Expected No. of transitions from state } s_i \text{ to state } s_j}{\text{Expected No. of transitions out of state } s_i} = \frac{\sum_m \xi_m(i,j)}{\sum_m \gamma_m(i)}$$

$$b_i(v_m) = \frac{\text{Expected No. of times observation } v_n \text{ occurs in state } s_i}{\text{Expected No. of times in state } s_i} = \frac{\sum_{m, O_m = v_n} \gamma_m(i)}{\sum_m \gamma_m(i)}$$

$$\pi(i) = P(s_i) = \frac{\text{Expected No. of times in state } s_i \text{ at time } m=1}{\text{Expected No. of times in any state at time } m=1} = \frac{\gamma_1(i)}{\sum_i \gamma_1(i)}$$

Learning/Training Algorithm

Initialization: Assume initial value of A, B, π

and calculate $P(O \mid M)$

E-step:

Estimate new values of the model parameters: A, B, π

from the previous values of A, B, π

M-step:

Find $P(O \mid M)$ with the new values of A, B, π

Repeat these steps until $P(O \mid M)$ declines