# Hidden Markov Model

# Introduction

- HMM is developed and published in 1960s and 70s by L.E. Baum and coworkers.
- It is a probabilistic models(Dynamic Bayesian Network).
- HMM is a tool for representing probability distributions over sequence of observations.
- It is a combination of Markov process and Bayes theory.
- It is represented as H(A,B,$\pi$). Where A is the state transition probability, B is the observation probability and Pi is the prior probability.

# Applications of HMM

- Uses
  - Speech recognition
    - Recognizing spoken words and phrases
  - Gesture recognition
    - Recognizing hand gestures
  - Text processing
    - Parsing raw records into structured records
  - Bioinformatics
    - Protein sequence prediction
  - Financial
    - Stock market forecasts (price pattern prediction)
    - Comparison shopping services

# Main Problems of 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 P(O|M) which shows the probability of observation sequences given to HMM .
- **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 produced the most probable state path to get the given observation sequences.
- **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 $o_k \in \{v_1,\ldots,v_M\}$.*

# Problem 1

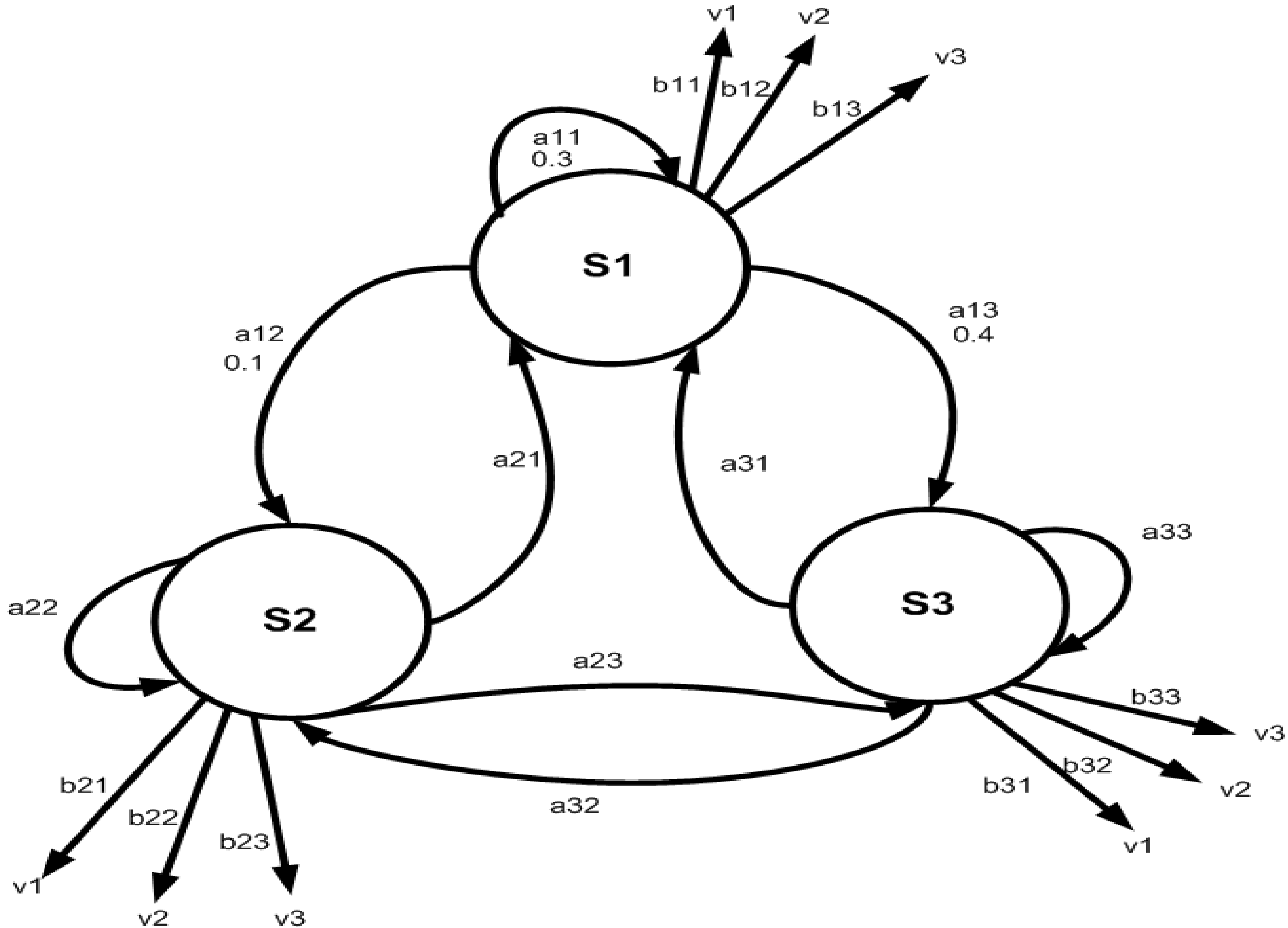
- Calculate the probability of state sequences where the observation sequences are v1 v3 v2.
- For solving this problem forward algorithm is used i.e. Calculate the probability

$$\alpha_j(t) = b_{jk} * \sum_{i=1}^{3} \alpha_i(t-1) * a_{ij}$$

- Suppose there are three states s1, s2, s3. i.e. hidden unit in HMM. Calculate the probability that it generates the sequence v1 v3 v2(evaluation problem). Where A is the transition probability from one state to another state $a_{ij}$ and B is the observation probability of visible state.

  B= $b_{jk}$ = P($v_k$(t)|$s_j$(t))

- s0 is the initial state at t=0 and v0 is the initial observation sequence at time t=0.



S1
S2
S3
a11
0.3
a12
0.1
a13
0.4
a21
a31
a22
a33
a23
a32
b11
b12
b13
b21
b22
b23
b31
b32
b33
v1
v2
v3

Transition probability matrix

$A = a_{ij}$

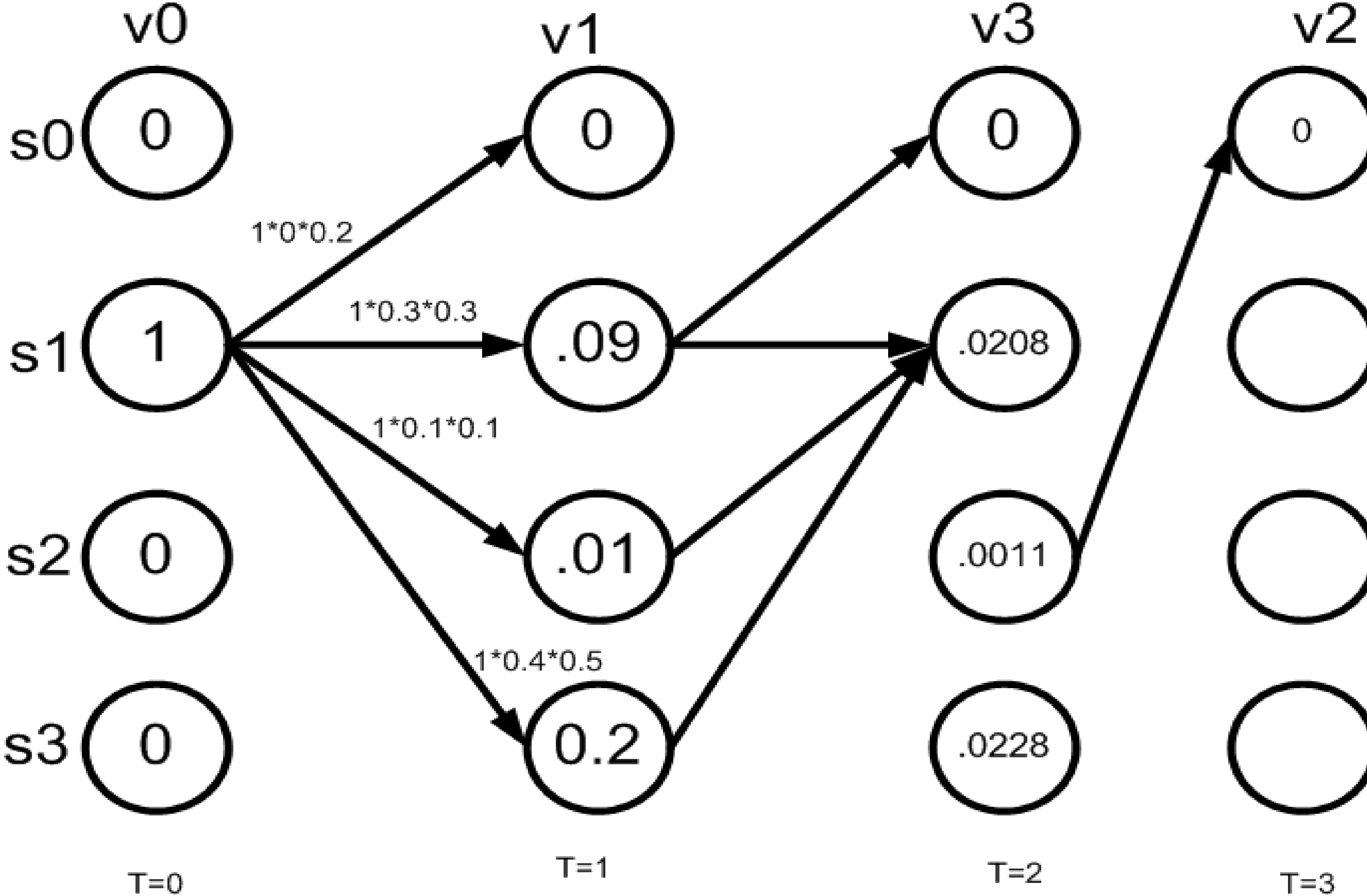
| | s0 | s1 | s2 | s3 |
|---|---|---|---|---|
| s0 | 1 | 0 | 0 | 0 |
| s1 | 0.2 | 0.3 | 0.1 | 0.4 |
| s2 | 0.2 | 0.5 | 0.2 | 0.1 |
| s3 | 0.8 | 0.1 | 0.0 | 0.1 |

Observation probability matrix

$B = b_{jk}$

| | v0 | v1 | v3 | v2 |
|---|---|---|---|---|
| s0 | 1 | 0 | 0 | 0 |
| s1 | 0 | 0.3 | 0.4 | 0.3 |
| s2 | 0 | 0.2 | 0.1 | 0.7 |
| s3 | 0 | 0.5 | 0.4 | 0.1 |

- Initial probabilities: say P(s1)=0.3 , P(s2)=0.2, P(s3)=0.1.
- Prior probability: [1 0 0]



v0
v1
v3
v2
s0
s1
s2
s3
0
1
0
0
1*0*0.2
1*0.3*0.3
1*0.1*0.1
1*0.4*0.5
0
.09
.01
0.2
0
.0208
.0011
.0228
0
T=0
T=1
T=2
T=3

- Where s0 is the initial state(start state at t=0).
- here 4 hidden states and 4 visible states. The number shown in circle is $\alpha_j$(t).
- From figure we see that the system was in hidden state s1 at t=0 i.e.($\alpha_1$(0)=1) and ($\alpha_j$(0)=0, j‡ 1).
- After that the visible state v1 is emitted at t=1, then calculate

$$(\alpha_0(1))= (\alpha_1(0))*a_{10}*b_{01}=1*0.2*0=0$$

$$(\alpha_1(1))= (\alpha_1(0))*a_{11}*b_{11}=1*0.3*0.3=.09$$

After that the visible state v3 is emitted at t=2, then $\alpha$ is calculated as:

$$\alpha_j(t) = b_{jk} * \sum_{i=1}^{3} \alpha_i(t-1) * a_{ij}$$

$$(\alpha_0(2)) = b_{03} \sum_{i=1}^{3} \alpha_i(1) * a_{i0} = 0$$

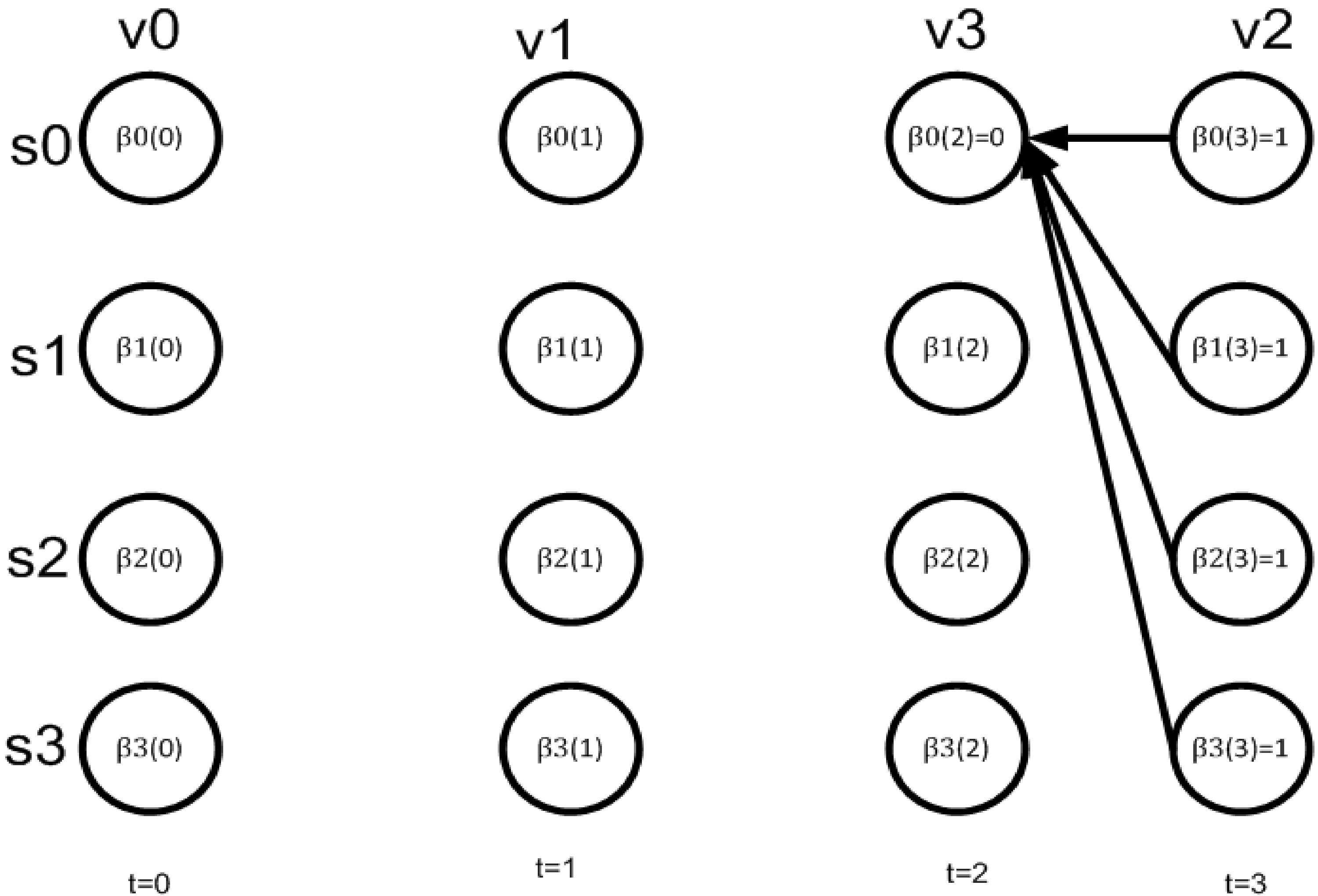
$$(\alpha_1(2)) = b_{13} \sum_{i=1}^{3} \alpha_i(1) * a_{i1} =$$

$$0.4*(.09*0.3+0.01*0.5+0.2*0.1)=0.0208$$

# Forward Calculation

| | t=0<br>v0 | t=1<br>v1 | t=2<br>v3 | t=3<br>v2 |
|---|---|---|---|---|
| $\alpha_0(t)$ | 0 | 0 | 0 | 0 |
| $\alpha_1(t)$ | 1 | 0.09 | 0.0208 | 0.00079 |
| $\alpha_2(t)$ | 0 | 0.01 | 0.0011 | 0.001596 |
| $\alpha_3(t)$ | 0 | 0.2 | 0.0228 | 0.000237 |

# Backward Algorithm



- Where s3 is the final state(Final state at t=3).
- Here 4 hidden states and 4 visible states. The number shown in circle is β $_j$(t).
- From figure we see that the system was in hidden state s at t=3 and the observation sequence is v2.
- Here we assume that ($\beta_0$(3)=1) and ($\beta_j$(3)=1, j>= 1).

- $$\boldsymbol{\beta_j(t) = \sum_{i=0}^{3} \beta_i(t+1)^* \, a_{ji}^* \, b_{ik}v(t+1)}$$

- After that the visible state v3 is emitted at t=2, then calculate
  - $\beta_0(2)= (\beta_0(1))^*a_{00}{}^*b_{0k}v(1)+ (\beta_1(1))^*a_{01}{}^*b_{1k}v(1)(\beta_2(1))^*a_{02}{}^*b_{2k}v(1)+ (\beta_3(1))^*a_{03}{}^*b_{3k}v(1)=0$, Here k=3

In similar way $\beta_1(2)$, $\beta_2(2)$, $\beta_3(2)$ is calculated.

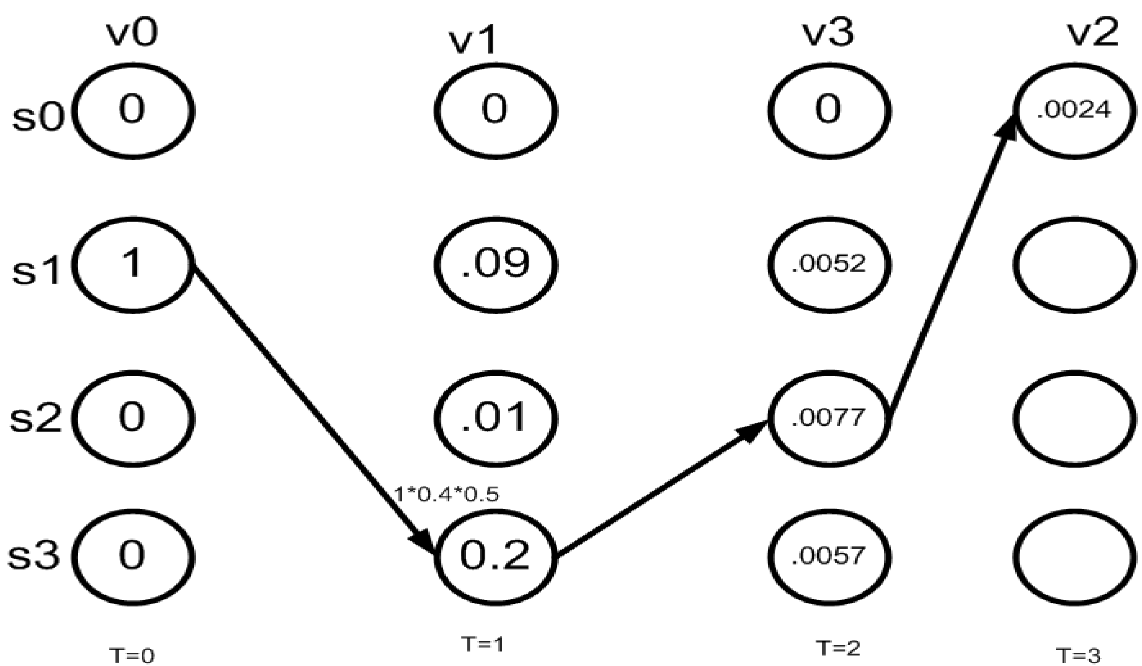
- After that the visible state v1 is emitted at t=1, then β is calculated as using above formula.
- In the similar way β is calculated for v0 at t=0.

# Backward Calculation

| | t=0<br>v0 | t=1<br>v1 | t=2<br>v3 | t=3<br>v2 |
|---|---|---|---|---|
| $\beta_0(t)$ | .0092 | 0 | 0 | 1 |
| $\beta_1(t)$ | 0 | 0.0092 | 0.21 | 1 |
| $\beta_2(t)$ | 0 | 0.00265 | 0.06. | 1 |
| $\beta_3(t)$ | 0 | 0.0086 | 0.17 | 1 |

# Decoding problem

- Find the optimal path for the sequence v0 v1 v3 v2?
- The problem is solved using Viterbi algorithm.



- It is similar to dynamic programming approach.
- Here we calculate the maximum probability among all the hidden states.
- After that highest probability hidden state will be explored.
- This process is continued upto final sequence.

- At t=0 i.e.($\alpha_1(0)=1$) and ($\alpha_j(0)=0$, j‡ 1).
- At t=1 i.e. $\alpha_1(1)=\max(\alpha_0(1), \alpha_1(1), \alpha_2(1), \alpha_3(1))$
- At t=2 i.e. $\alpha_j(t) = b_{j\mathrm{k}} * max(\alpha_i(t-1) * a_{ij})$
- After that

  $\psi(t) = \text{Arg max } (\alpha_j(t))$ where j=0,1,2,3

# Viterbi Algorithm

| | t=0<br>v0 | t=1<br>v1 | t=2<br>v3 | t=3<br>v2 |
|---|---|---|---|---|
| $\alpha_0(t)$ | 0 | 0 | 0 | 0 |
| $\alpha_1(t)$ | **1** | 0.09 | 0.0208 | 0.00079 |
| $\alpha_2(t)$ | 0 | 0.01 | 0.0011 | **.001596** |
| $\alpha_3(t)$ | 0 | **0.2** | **0.0228** | 0.000237 |

# Learning problem (1)

- **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, that is maximizes $P(O \mid M)$ .

- There is no algorithm producing optimal parameter values.

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

# Learning problem (2)

- If training data has information about sequence of hidden states (as in word recognition example), then use maximum likelihood estimation of parameters:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Number of transitions from state } S_j \text{ to state } S_i}{\text{Number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Number of times observation } V_m \text{ occurs in state } S_i}{\text{Number of times in state } S_i}$$

# Baum-Welch algorithm

General idea:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Expected number of transitions from state } S_j \text{ to state } S_i}{\text{Expected number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Expected number of times observation } V_m \text{ occurs in state } S_i}{\text{Expected number of times in state } S_i}$$

$$\pi_i = P(s_i) = \text{Expected frequency in state } S_i \text{ at time } k=1.$$

# Baum-Welch algorithm: expectation step(1)

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

$$\xi_k(i,j) = P(q_k = s_i\, , q_{k+1} = s_j \mid o_1\, o_2 \dots\, o_K)$$

$$\xi_k(i,j) = \frac{P(q_k = s_i\ , q_{k+1} = s_j\ , o_1\, o_2 \dots\, o_k)}{P(o_1\, o_2 \dots\, o_k)} =$$

$$\frac{P(q_k = s_i\ , o_1\, o_2 \dots\, o_k)\ a_{ij}\ b_j(o_{k+1})\ P(o_{k+2} \dots\, o_K \mid q_{k+1} = s_j)}{P(o_1\, o_2 \dots\, o_k)} =$$

$$\frac{\alpha_k(i)\ a_{ij}\ b_j(o_{k+1})\ \beta_{k+1}(j)}{\Sigma_i\, \Sigma_j\ \alpha_k(i)\ a_{ij}\ b_j(o_{k+1})\ \beta_{k+1}(j)}$$

# Learning problem (2)

- If training data has information about sequence of hidden states (as in word recognition example), then use maximum likelihood estimation of parameters:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Number of transitions from state } S_j \text{ to state } S_i}{\text{Number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Number of times observation } V_m \text{ occurs in state } S_i}{\text{Number of times in state } S_i}$$

# Baum-Welch algorithm

General idea:

$$a_{ij} = P(s_i \mid s_j) = \frac{\text{Expected number of transitions from state } S_j \text{ to state } S_i}{\text{Expected number of transitions out of state } S_j}$$

$$b_i(v_m) = P(v_m \mid s_i) = \frac{\text{Expected number of times observation } V_m \text{ occurs in state } S_i}{\text{Expected number of times in state } S_i}$$

$$\pi_i = P(s_i) = \text{Expected frequency in state } S_i \text{ at time } k=1.$$

# Baum-Welch algorithm: expectation step(1)

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

$$\xi_k(i,j) = P(q_k = s_i\,, q_{k+1} = s_j \mid o_1\, o_2 \dots\, o_K)$$

$$\xi_k(i,j) = \frac{P(q_k = s_i\,, q_{k+1} = s_j\,, o_1\, o_2 \dots\, o_k)}{P(o_1\, o_2 \dots\, o_k)} =$$

$$\frac{P(q_k = s_i\,, o_1\, o_2 \dots\, o_k)\, a_{ij}\, b_j(o_{k+1})\, P(o_{k+2} \dots\, o_K \mid q_{k+1} = s_j)}{P(o_1\, o_2 \dots\, o_k)} =$$

$$\frac{\alpha_k(i)\, a_{ij}\, b_j(o_{k+1})\, \beta_{k+1}(j)}{\Sigma_i \Sigma_j\, \alpha_k(i)\, a_{ij}\, b_j(o_{k+1})\, \beta_{k+1}(j)}$$