The Viterbi Algorithm

Examples (cont.)

Suppose two events $A \subset \Omega, B \subset \Omega$ are not mutually exclusive:

$$A \cap B \neq f$$

Then

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

proof:

mutually exclusive

$$A \cup B = A \cup \overline{A} \cap B \qquad B = A \cap B \cup \overline{A} \cap B$$

$$\Pr(A \cup \overline{A}B) = \Pr(A) + \Pr(\overline{A} \cap B) \qquad \Pr(B) = \Pr(A \cap B) + \Pr(\overline{A} \cap B)$$

$$\Rightarrow \Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \quad \square$$

Examples (cont.)

if $A \subset \Omega$ then $\overline{A}$ is the event corresponding to "A did not occur", and

$$\Pr\left(\overline{A}\right) = 1 - \Pr\left(A\right)$$

ex) 1 roll of a fair die

if A = {roll is even} then $\overline{A}$ = {roll is odd}

Pr(A) = 1 - Pr($\overline{A}$) = 0.5

Examples (cont.)

ex) A fair coin is tossed 3 times in succession.

Events: *A*- get a total of 2 heads
B- get a head on second toss

Ω = {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}

	HHH	HHT	HTH	HTT	THH	THT	TTH	TTT
A:		x	x		x			
B:	x	x			x	x		

Pr(*A*) = 3/8 Pr(*B*) = 4/8 Pr($A \cap B$) = 2/8

$$\Pr\left(A \cup B\right) = 3/8 + 4/8 - 2/8 = 5/8$$

Conditional Probability

$$\Pr(A|B) \equiv \frac{\Pr(A \cap B)}{\Pr(B)}$$

ex) A fair coin is tossed 3 times in succession.

Events: *A*- get a total of 2 heads
B- get a head on second toss

Ω = {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}

A:		x	x		x			
B:	x	x			x	x		

Pr(*B*) = 4/8, Pr($A \cap B$) = 2/8, Pr(A | B) = (2/8) / (4/8) = 1/2

Examples (cont.)

ex) A fair die is thrown once:

Ω = {1, 2, 3, 4, 5, 6}

- *A*- roll a “2”
- *B*- roll is even
- Pr(*A*) = 1/6 Pr(*B*) = 3/6 $\Pr(A \cap B) = \Pr(A) = 1/6$

P(*A* | *B*) = (1/6)/(3/6) = 1/3

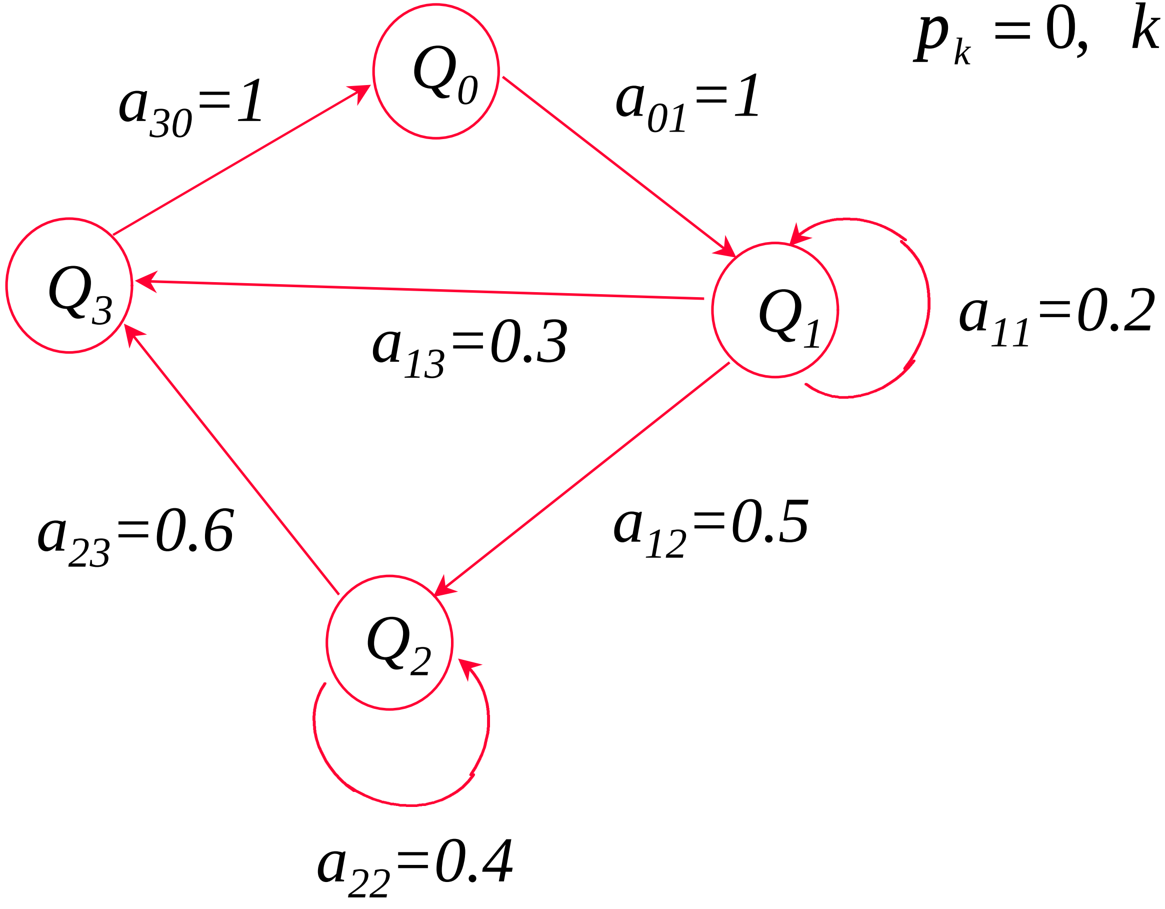
note Pr(*A* | *A*) = 1, and if *A* and *B* are independent events:

$$\Pr(A|B) = \Pr(A)$$

Hidden Markov Models (HMM's)

example 1)

$$p_0 = 1$$
$$p_k = 0, \quad k \neq 0$$



Example of an HMM

- The a_{ij} are *state transition probabilities*, give the probability of moving from state *i* to state *j*.
- Note that: $$\sum_j a_{ij} = 1$$
- At state Q_i, one of 3 output symbols, *R*, *B*, or *Y* is generated with probabilities $b_i(R), b_i(B),$ or $b_i(Y)$

State, Q_i	$b_i(R)$	$b_i(B)$	$b_i(Y)$
0	0.3	0.2	0.5
1	0.7	0.2	0.1
2	0.9	0	0.1
3	0.2	0.8	0

Example of an HMM (cont.)

- One output symbol is generated per state (like a Moore state machine).

possible output sequence: R, Y, B, B, R, Y, R, ...
state: Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2, ...

- Often the observed output symbols bear no obvious relationship to the state sequence (*i.e*. states are "hidden").
- Knowing the state sequence generally provides more useful information about the characteristics of the signal being analyzed than the observed output symbols (as was the case with syntactic recognition).

Definition of Hidden Markov Models

- there are *T* observation times: $t = 0, \ldots, T\text{-}1$
- there are *N* states: $Q_0, \ldots, Q_{N\text{-}1}$
- there are *M* observation symbols: $v_0, \ldots, v_{M\text{-}1}$
- state transition probabilities:

$$a_{ij} = \Pr\left(Q_j \text{ at time } t+1 \mid Q_i \text{ at time } t\right)$$

- symbol probabilities:

$$b_j(k) = \Pr\left(v_k \text{ at time } t \mid Q_j \text{ at time } t\right)$$

- initial state probabilities:

$$p_i = \Pr\left(Q_i \text{ at } t = 0\right)$$

Definition of Hidden Markov Models (cont.)

- Define the matrices *A*, *B*, and Π:

$$\{A\}_{ij} = a_{ij}, \quad i, j = 0, \ldots, N-1$$

$$\{B\}_{jk} = b_j(k), \quad j = 0, \ldots, N-1, \quad k = 0, \ldots, M-1$$

$$\{\Pi\}_i = p_i, \quad i = 0, \ldots, N-1$$

notation for HMM: $l = (A, B, \Pi)$

- Notation for observation sequence: $O = O_0, O_1, \ldots, O_{T-1}$
- Notation for state sequence: $I = i_0, i_1, \ldots, i_{T-1}$

Three Fundamental Problems

- Problem 1: Given the observation sequence $O = O_0, O_1, \ldots, O_{T-1}$ and the model $l = (A, B, \Pi)$, how do we compute the probability of the observation sequence, $Pr(O \mid \lambda)$?
- Problem 2: Given the observation sequence $O = O_0, O_1, \ldots, O_{T-1}$ and the model $l = (A, B, \Pi)$, how do we estimate the state sequence, $I = i_0, i_1, \ldots, i_{T-1}$ which produced the observations?
- Problem 3: How do we adjust the model parameters $l = (A, B, \Pi)$ to maximize $Pr(O \mid \lambda)$?

Relevance to Normal/Abnormal ECG Rhythm Detection

- Suppose we have one HMM that models normal rhythm, and a second HMM that models abnormal rhythm, and we have a measured observation sequence. Problem 1 can be used to determine which is the most likely model for the measured observations, hence, we can classify the rhythm as normal or abnormal.
- Suppose we have a single model which enables us to associate certain states with with the components of the ECG (P, QRS, and T waves). Problem 2 can be used to estimate the states from the observation sequence. The state sequence can then be used to detect P, QRS, and T waves.

Relevance to Normal/Abnormal ECG Rhythm Detection (cont.)

- Problem 3 is used to generate the model parameters that best fit a given training set of observations. In effect, the solution to Problem 3 allows us to build the model. This problem must be solved first before we can solve Problems 1 and 2. Problem 3 is more difficult to solve than Problems 1 and 2.

Markovian Property of State Sequences

- The sequence i_0, i_1, …, i_{T-1} has the Markov property:

$$\Pr\left(i_k | i_{k-1}, i_{k-2}, \ldots, i_0\right) = \Pr\left(i_k | i_{k-1}\right)$$

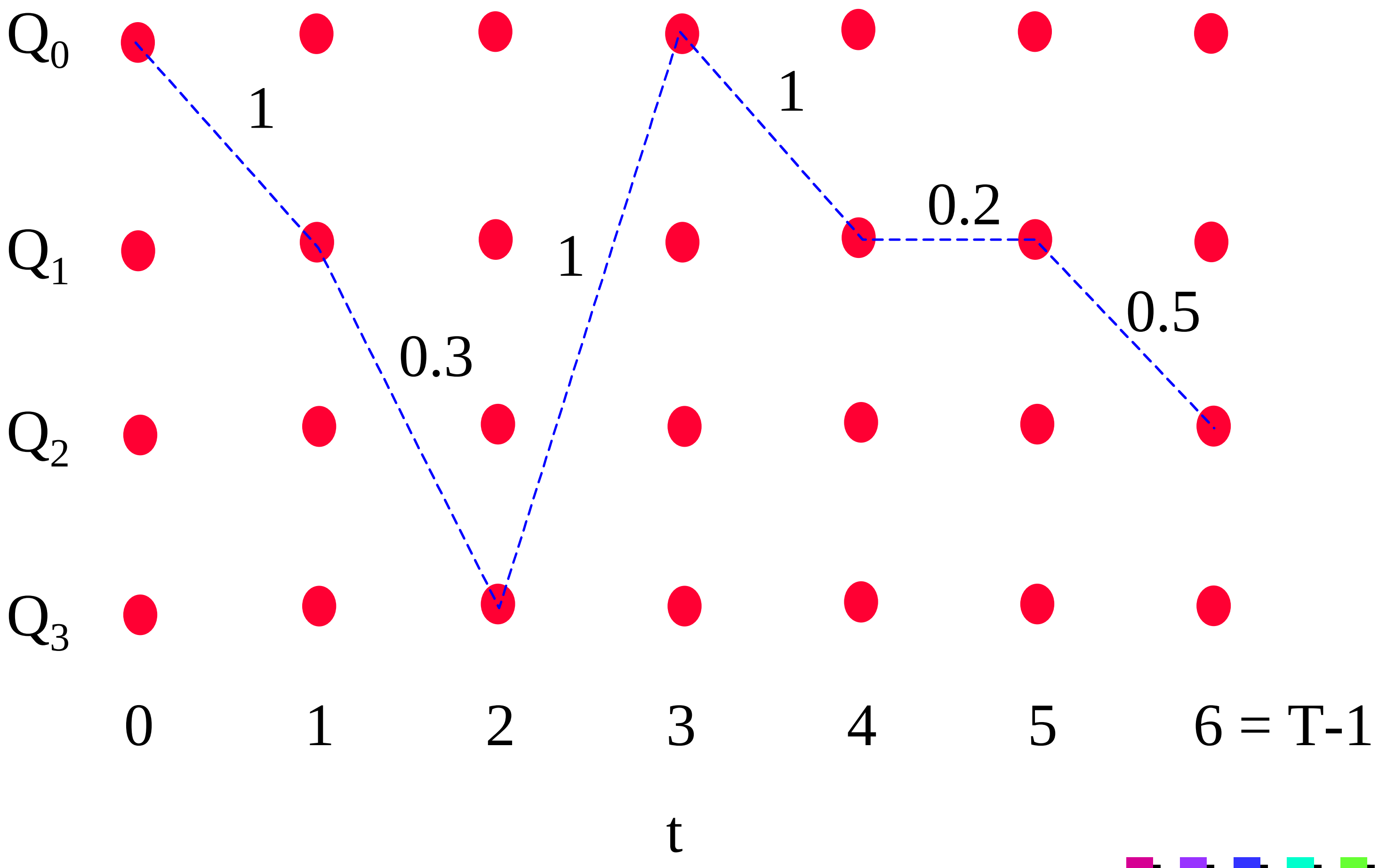
  that is, the state at time $t = k$, i_k, is independent of all previous states except i_{k-1}.

- A consequence of this property is (homework):

$$\Pr\left(i_k, i_{k-1}, i_{k-2}, \ldots, i_0\right) = \Pr\left(i_k | i_{k-1}\right) \Pr\left(i_{k-1} | i_{k-2}\right) \cdots \Pr\left(i_1 | i_0\right) \Pr\left(i_0\right)$$

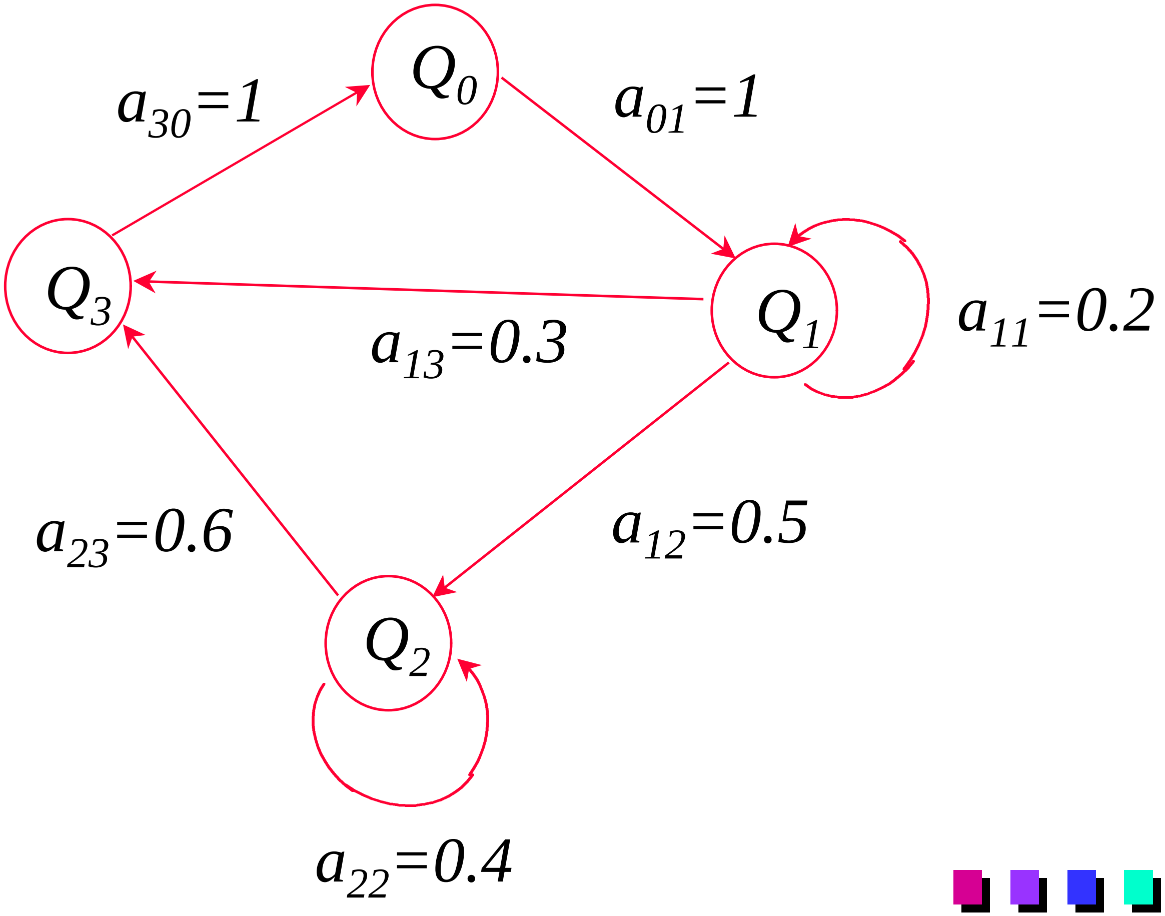
notation: $\Pr\left(i_k, i_{k-1}, i_{k-2}, \ldots, i_0\right) \equiv \Pr\left(i_k \cap i_{k-1} \cap i_{k-2} \cap \ldots \cap i_0\right)$

Trellis Representation of HMM in Example 1



Probability of state sequence: $I = Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2$

$\Pr(Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2) = 1*0.3*1*1*0.2*0.5 = 0.03$



Probability of a given *I* and *O*: $\Pr(I \cap O)$

observed output sequence: $R, \quad Y, \quad B, \quad B, \quad R, \quad Y, \quad R$

state: $Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2$

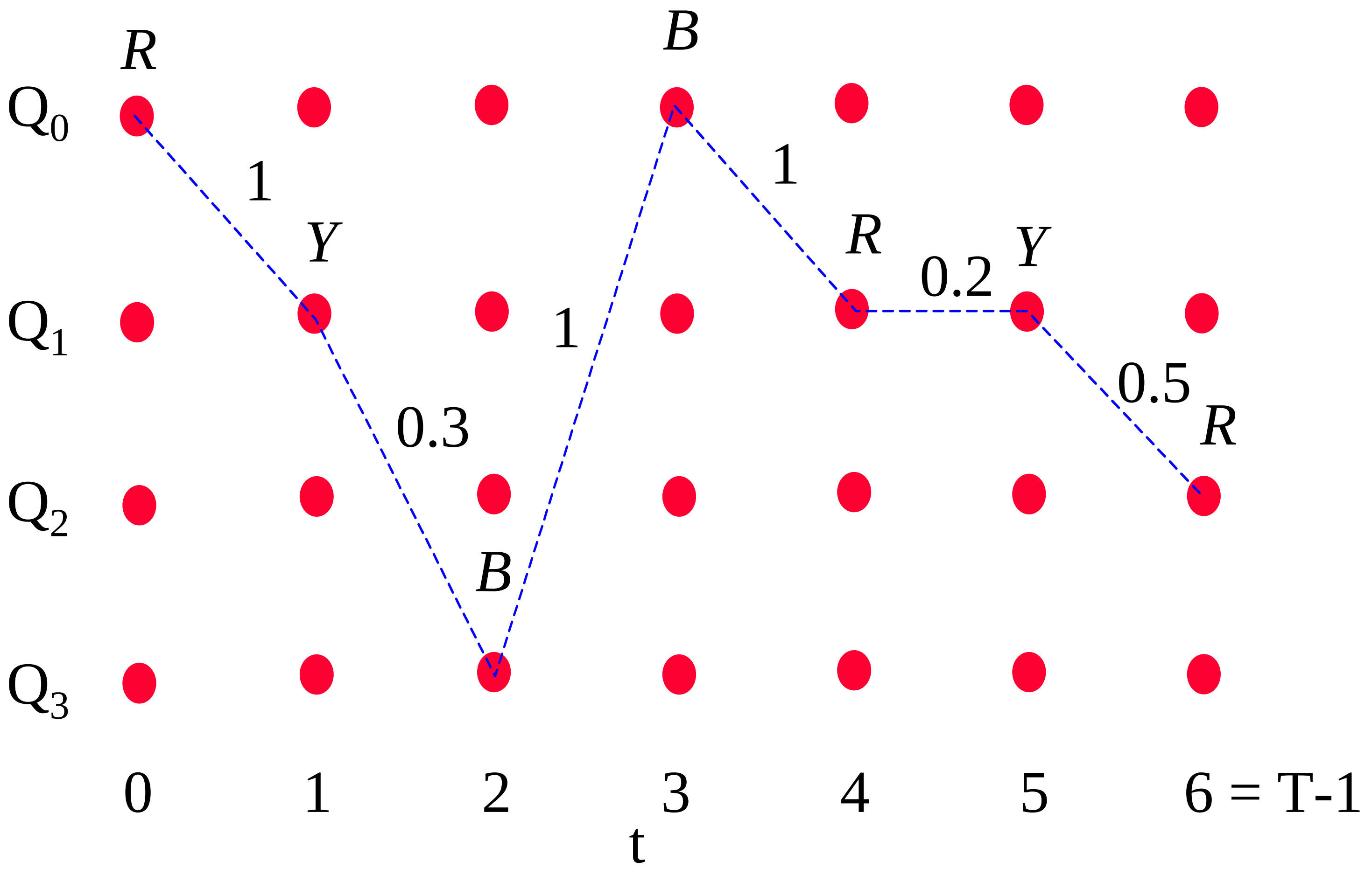
Note that:

$$\Pr(I \cap O) = \Pr(I)\Pr(O|I)$$

Back to Example 1

output sequence: *R*, *Y*, *B*, *B*, *R*, *Y*, *R*

state: Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2



Example (cont.)

output sequence: $R, \quad Y, \quad B, \quad B, \quad R, \quad Y, \quad R$

state: $Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2$

$$\Pr(I \cap O) = \Pr(I)\Pr(O|I)$$

$\Pr(I) = \Pr(Q_0, Q_1, Q_3, Q_0, Q_1, Q_1, Q_2)$

$= 1*0.3*1*1*0.2*0.5 = 0.03$

$\Pr(O \mid I) = \Pr(R, Y, B, B, R, Y, R)$

$= 0.3*0.1*0.8*0.2*0.7*0.1*0.9 = 0.0003024$

State, Q_i	$b_i(R)$	$b_i(B)$	$b_i(Y)$
0	0.3	0.2	0.5
1	0.7	0.2	0.1
2	0.9	0	0.1
3	0.2	0.8	0

Example (cont.)

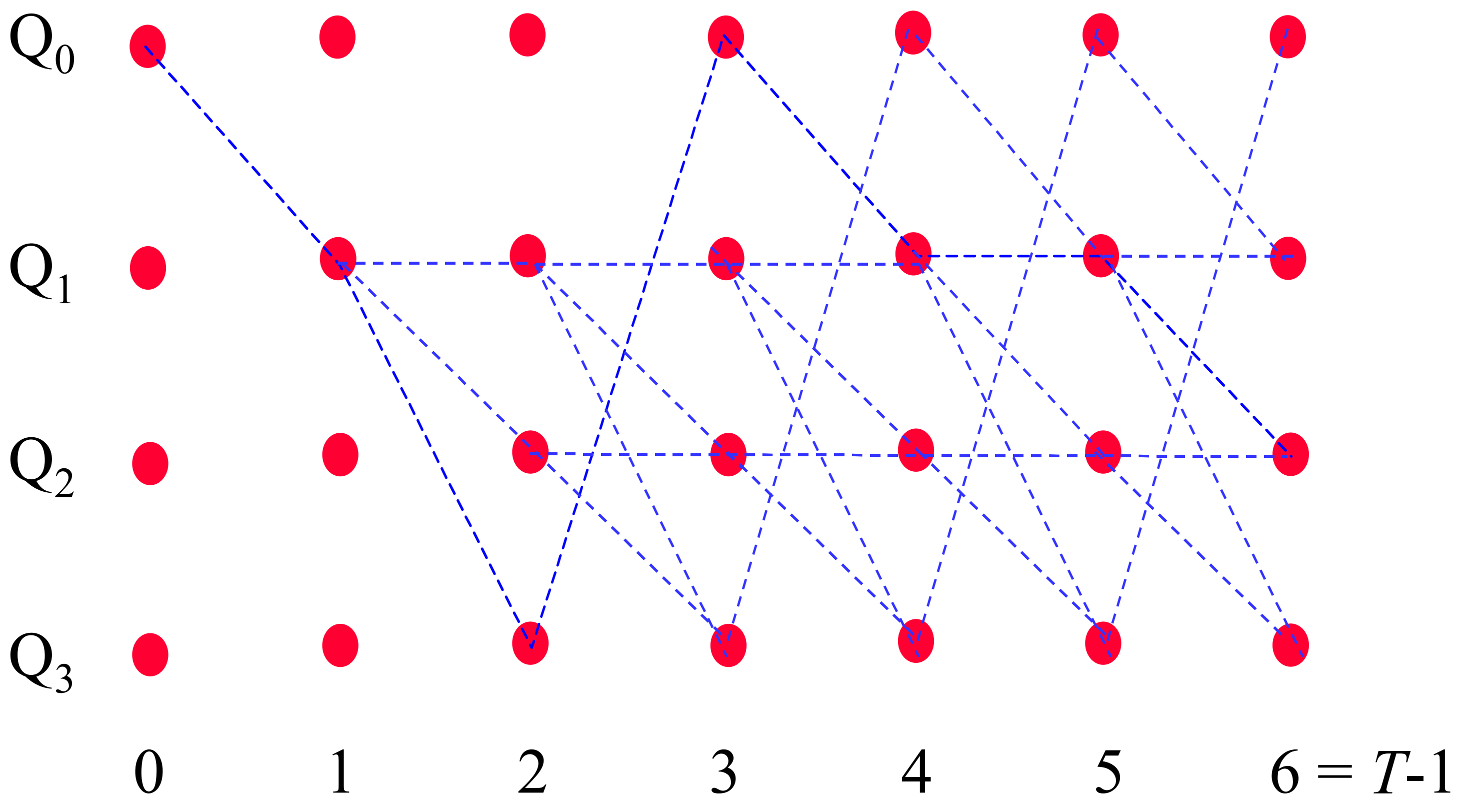
$\Pr(\ I\ ) = 0.03$
$\Pr(\ O|I\ ) = 0.0003024$

$$\Rightarrow \Pr(O \cap I) = 0.03 \times 0.0003024 = 9.072 \times 10^{-6}$$

ex) How many possible state sequences are there?

- in general, there are on the order of N^T possible state sequences, (for Example 1, that's $4^7 = 16{,}384$).
- Since some of the transition probabilities are zero, this number decreases to only 30.
- Let each state sequence be denoted by I_i, $i = 1, \ldots,$ $R=O(N^T)$.

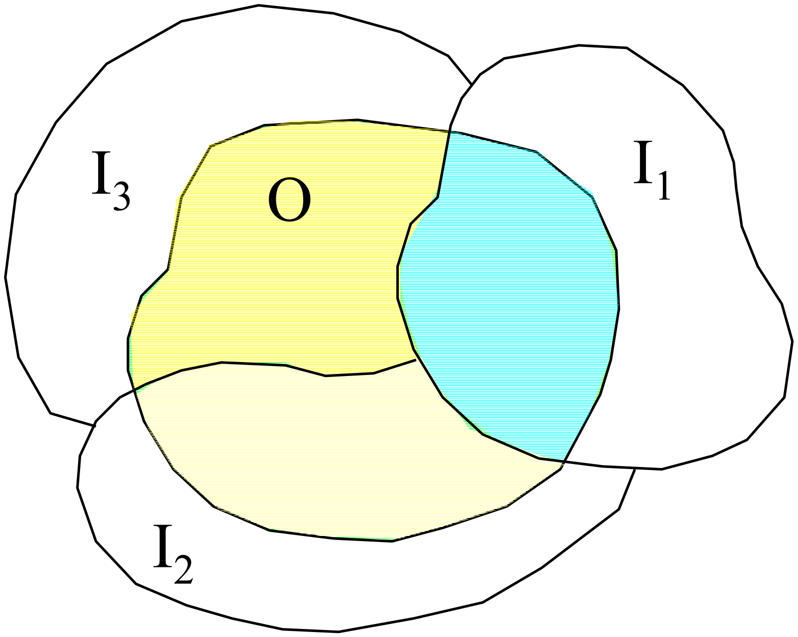
Total Number of Possible State Sequences: 30



Distributive-Type Property

since $I_i,\ i = 1,\ldots R \equiv O\left(N^T\right)$ are disjoint events:

R = 3



$$\Pr\left(\sum_{i=1}^{R} O \cap I_i\right) \underset{\text{(by axiom 2)}}{=} \sum_{i=1}^{R} \Pr\left(O \cap I_i\right) = \Pr(O)$$

since R is so large, this is not a practical solution to Problem 1

Solution to Problem 1: Forward-Backward Algorithm

We seek $\Pr(O|\lambda)$

Forward variable:

$$\alpha_t(i) = \Pr(O_0, O_1, \ldots, O_t, i_t = Q_i|\lambda)$$

- •this is the probability that we observe the partial observation sequence, $O_0, O_1, \ldots, O_t$ and arrive at state Q_i at time t (given the model λ).
- •In the forward-backward algorithm the forward variable is updated recursively.
- •Note that the events $O_0, O_1, \ldots, O_t, i_t = Q_i$ are disjoint for each Q_i.

Forward-Backward Algorithm (cont.)

$$\alpha_0(i) = \pi_i b_i(O_0), \quad 0 \le i \le N-1$$

- for $t = 0, 1, \ldots, T\text{-}2,\ 0 \le j \le N\text{-}1$

$$\alpha_{t+1}(j) = \left[\sum_{i=0}^{N-1} \alpha_t(i) a_{ij}\right] b_j(O_{t+1})$$

- then,

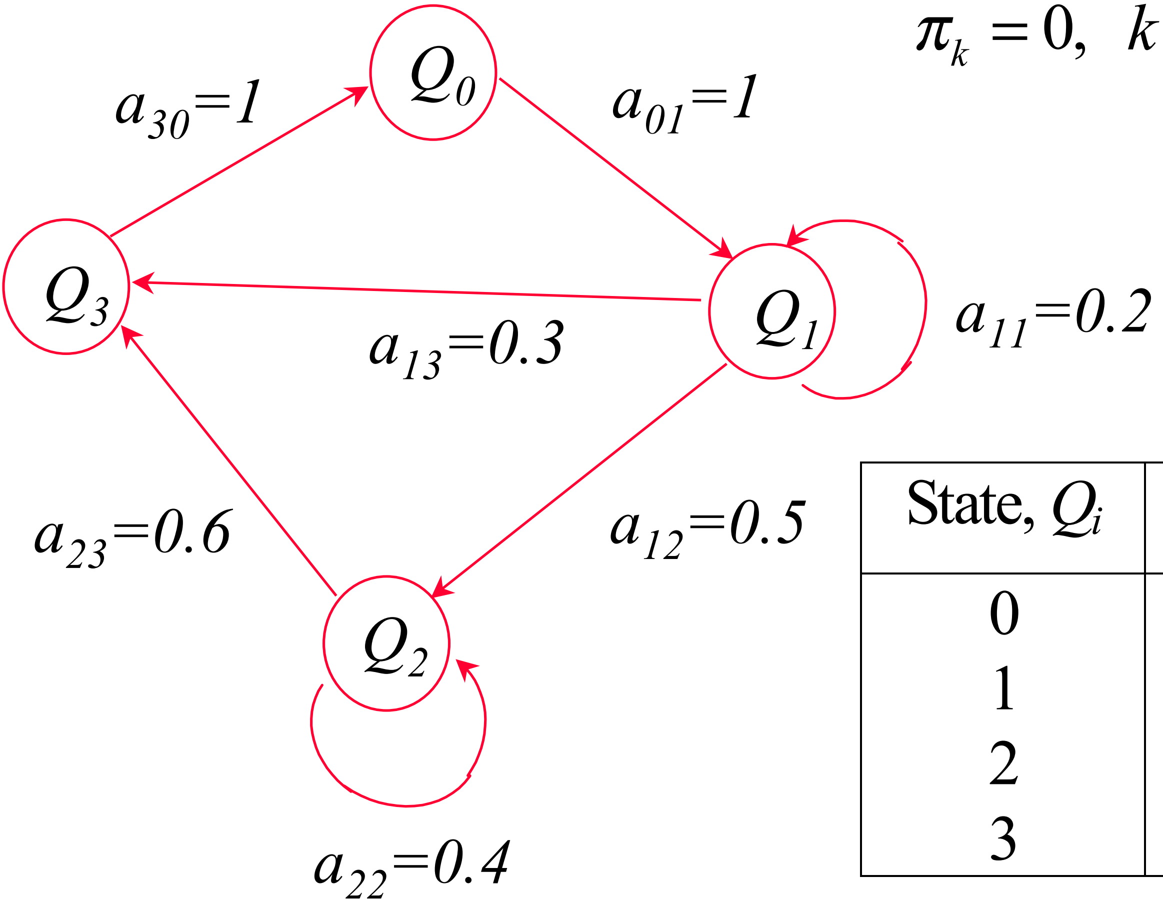
$$\Pr(O|\lambda) = \sum_{i=0}^{N-1} \alpha_{T-1}(i)$$

the algorithm can be easily implmented via arithmetic involving the matrices A, B, and Π.

Application of Forward-Backward Algorithm to Example 1

$$\pi_0 = 1$$

$$\pi_k = 0, \quad k \neq 0$$



State, Q_i	$b_i(R)$	$b_i(B)$	$b_i(Y)$
0	0.3	0.2	0.5
1	0.7	0.2	0.1
2	0.9	0	0.1
3	0.2	0.8	0

•observed output sequence: *R*, *Y*, *B*, *B*, *R*, *Y*, *R*

•we don’t know the state sequence

Application of Forward-Backward Algorithm to Example 1 (cont.)

$\alpha_t(j)$

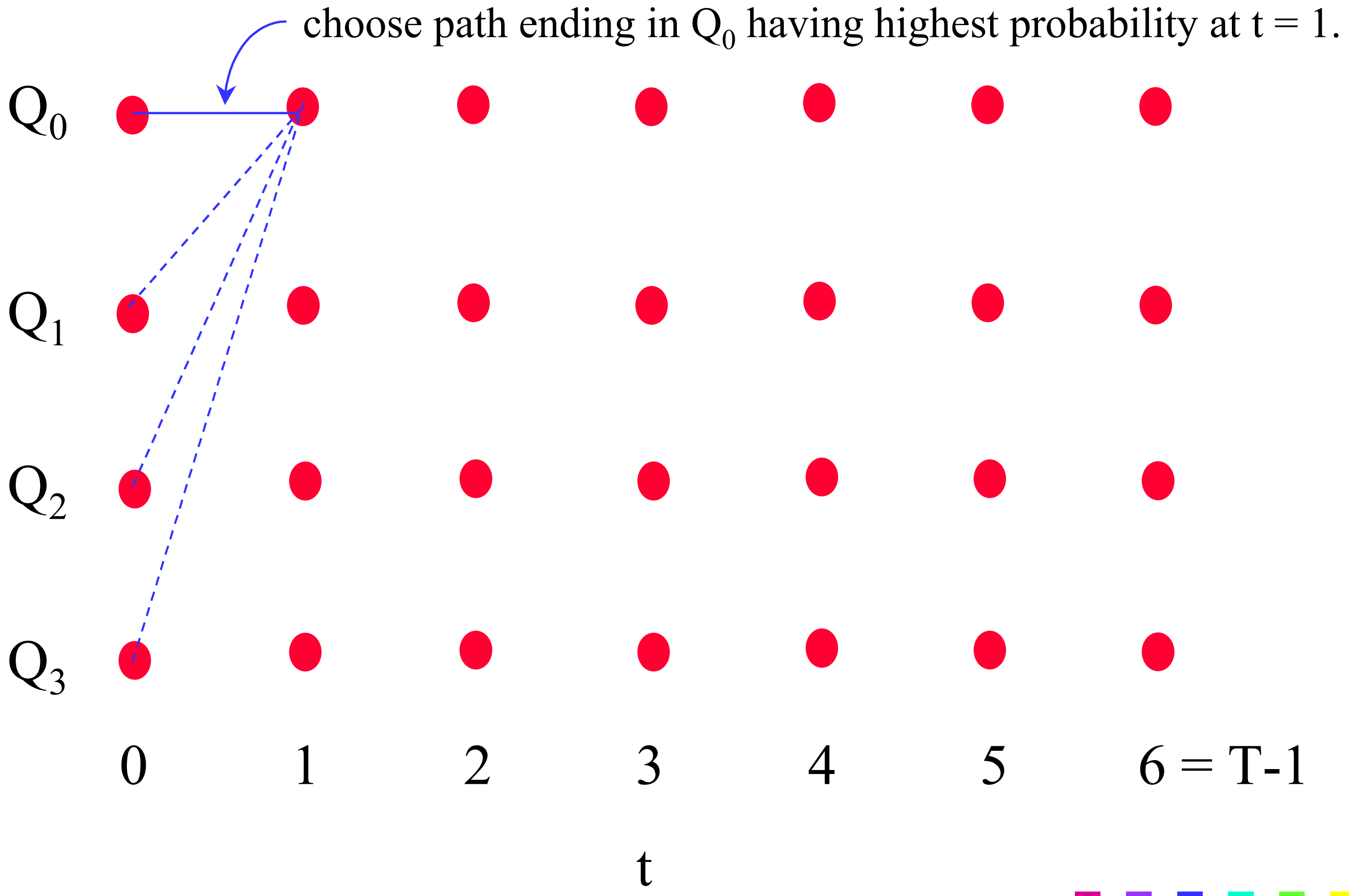
j t	0	1	2	3
0	0.3	0	0	0
1	0	0.03	0	0
2	0	1.2E-2	0	7.2E-3
3	1.44E-3	4.8E-5	0	2.88E-4
4	8.64E-5	1.0147E-3	2.16E-5	2.88E-6
5	1.44E-6	2.8934E-5	5.15E-5	0
6	0	5.0588E-6	3.1596E-5	7.9281E-6

$$\Pr(O|\lambda) = 4.4582\text{E} - 5$$

Solution to Problem 2: The Viterbi Algorithm

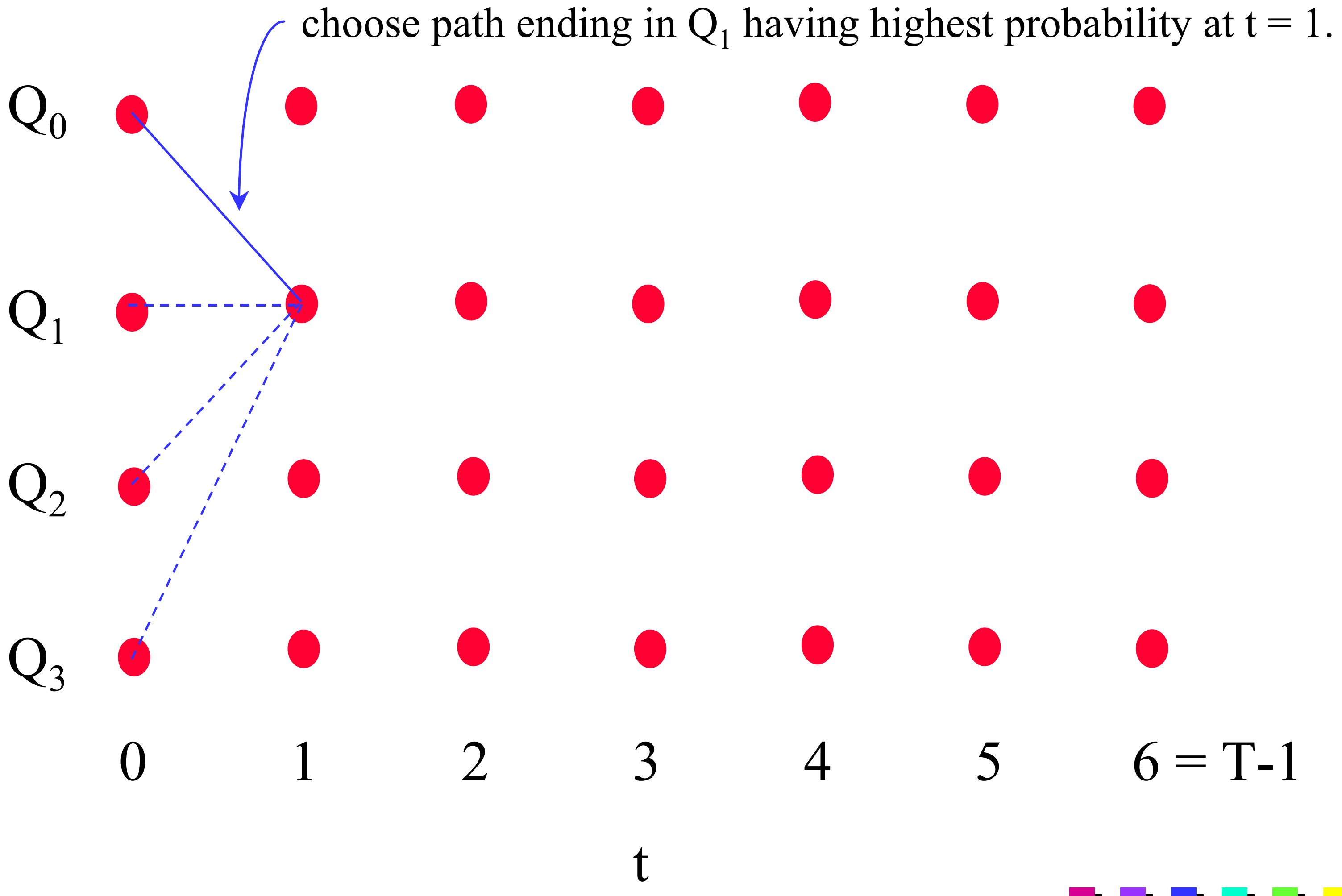
- We seek the state sequence that maximizes $\Pr(I|O,\lambda)$
- This is equivalent to maximizing $\Pr(I \cap O)$ (given λ)
- The trellis diagram representation of HHM's is useful in this regard. We seek the path through the trellis that has the maximum $\Pr(I \cap O)$
- At each column (time step) in the trellis, the Viterbi algorithm eliminates all but *N* possible state sequences.
- At each time step, the *N* retained sequences all end in different states.
- If more than one sequence ends in the same state, the sequence with the maximum probability is retained.

Viterbi Algorithm (cont.)

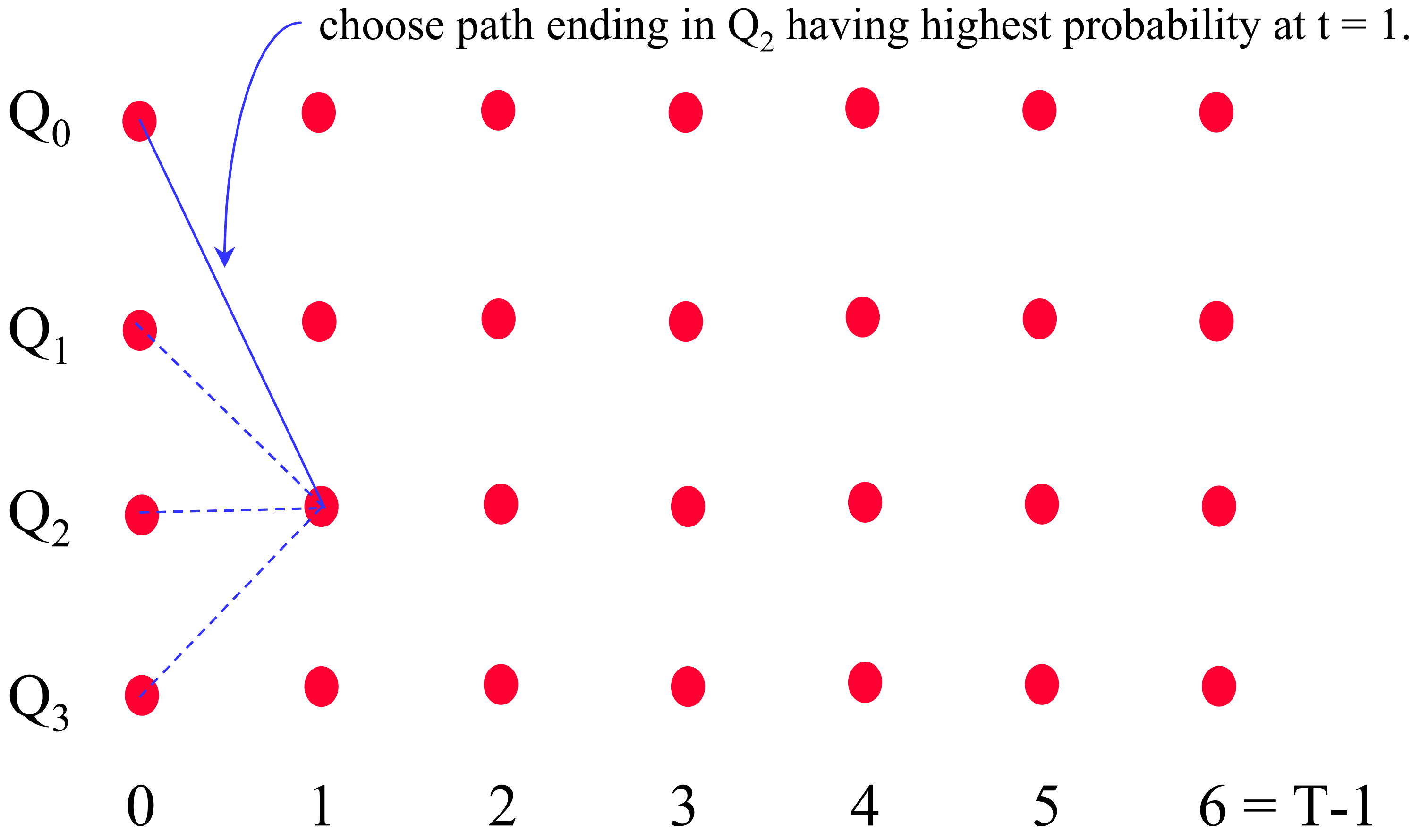


choose path ending in Q_0 having highest probability at $t = 1$.

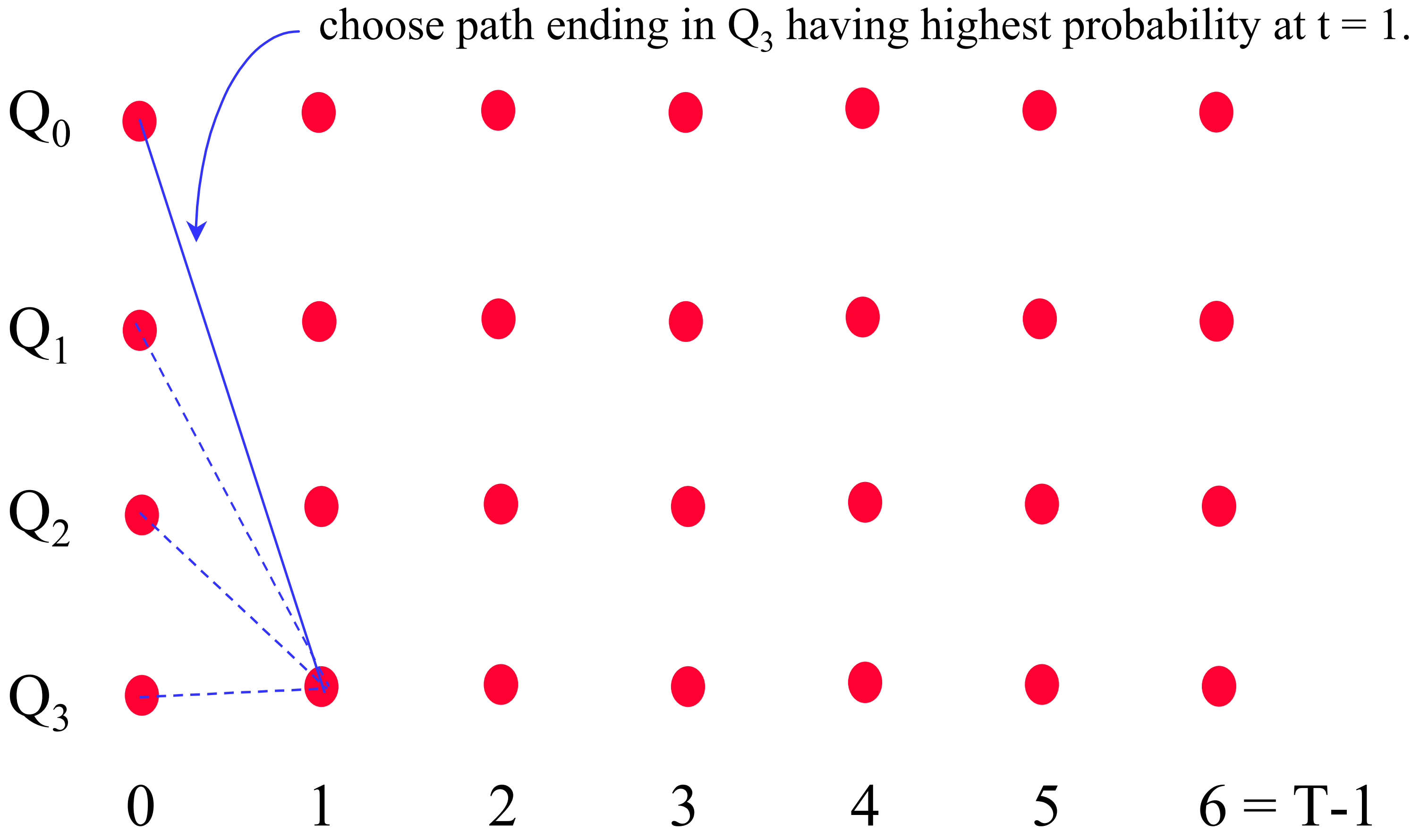
Viterbi Algorithm (cont.)


choose path ending in Q_1 having highest probability at $t = 1$.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1
t

Viterbi Algorithm (cont.)


choose path ending in Q_2 having highest probability at t = 1.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1

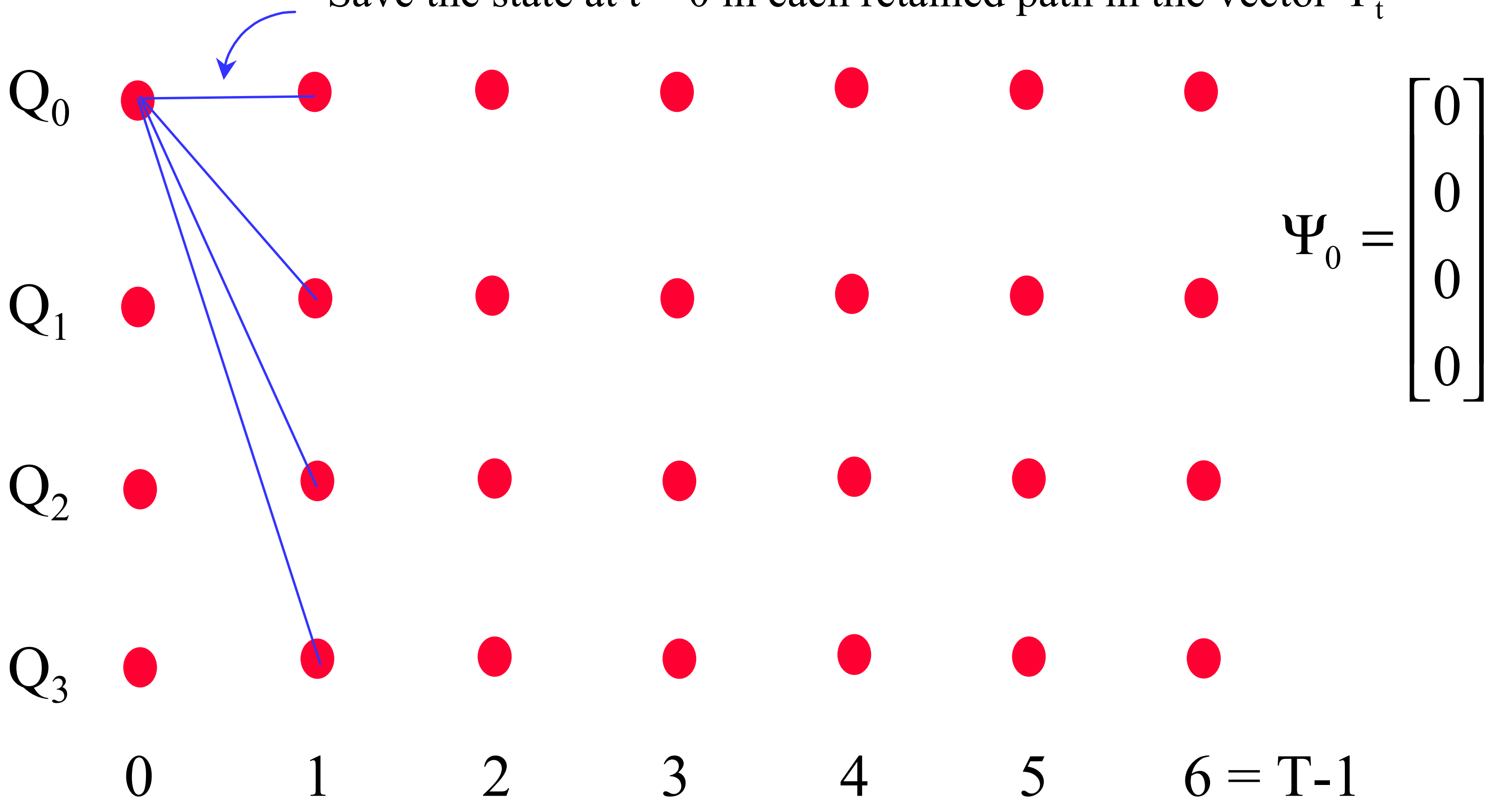
Viterbi Algorithm (cont.)



choose path ending in Q_3 having highest probability at t = 1.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1

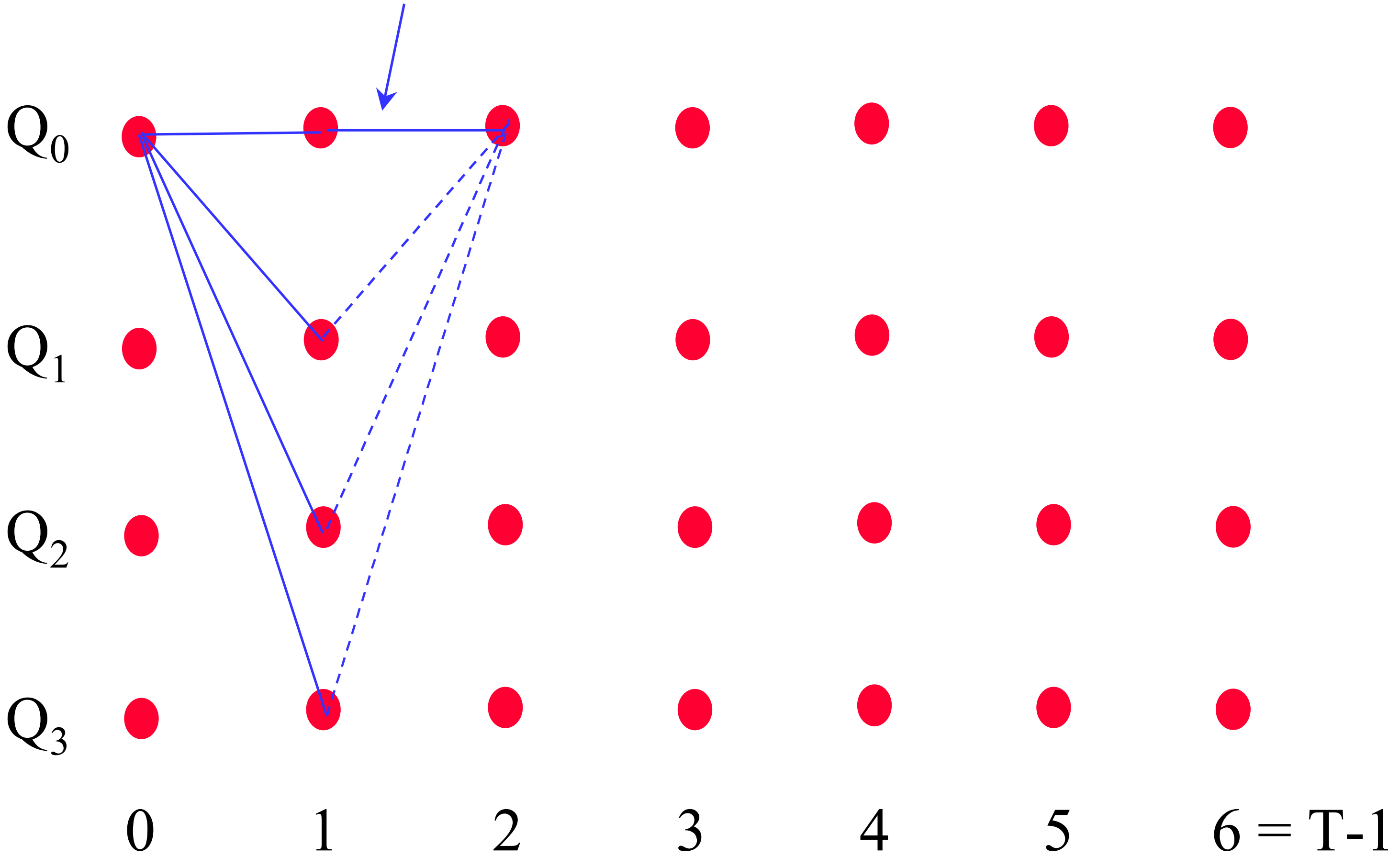
Viterbi Algorithm (cont.)

Save each of the N = 4 maximum probabilities in the vector δ_t
Save the state at t = 0 in each retained path in the vector Ψ_t



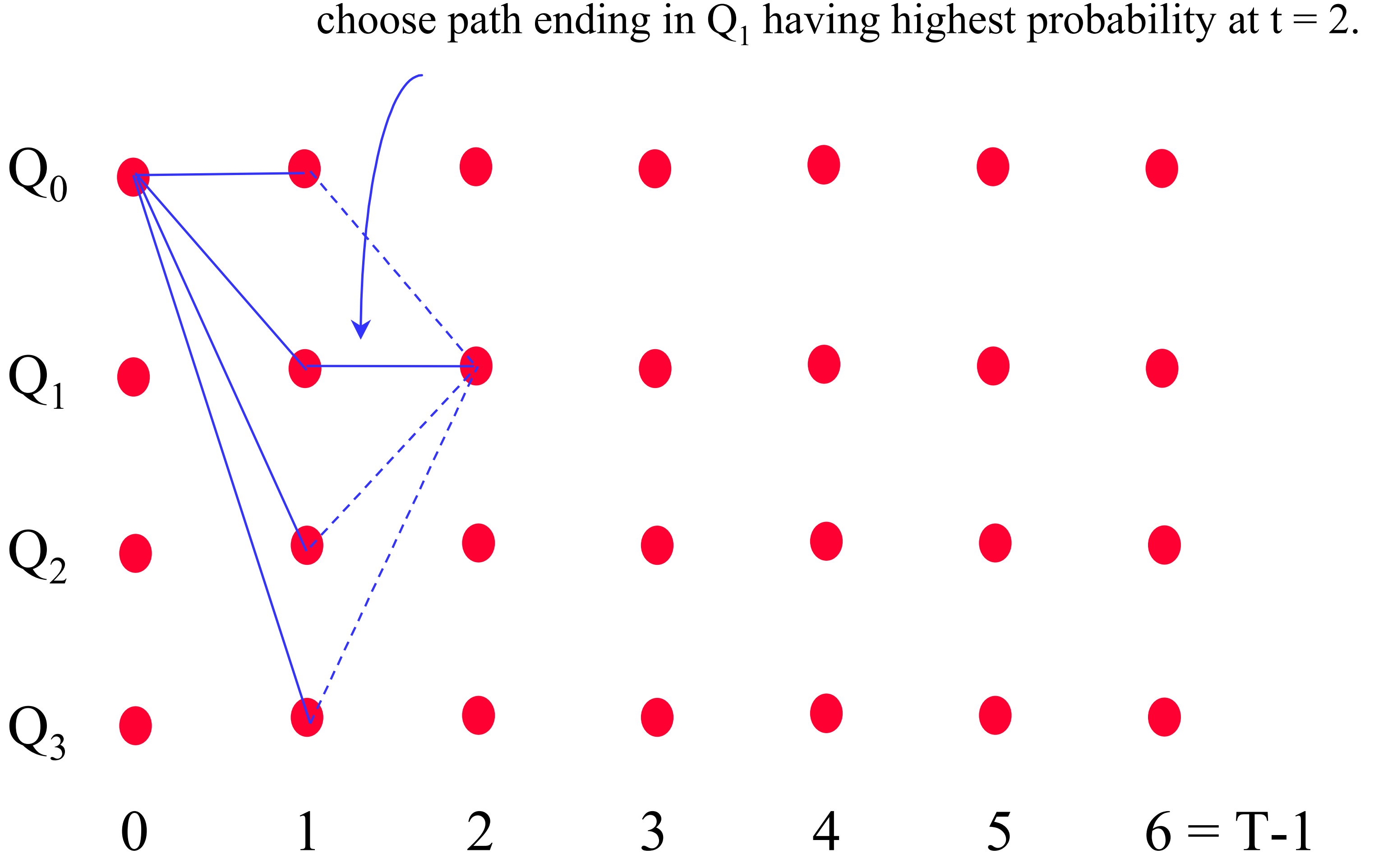
$$\Psi_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Viterbi Algorithm (cont.)

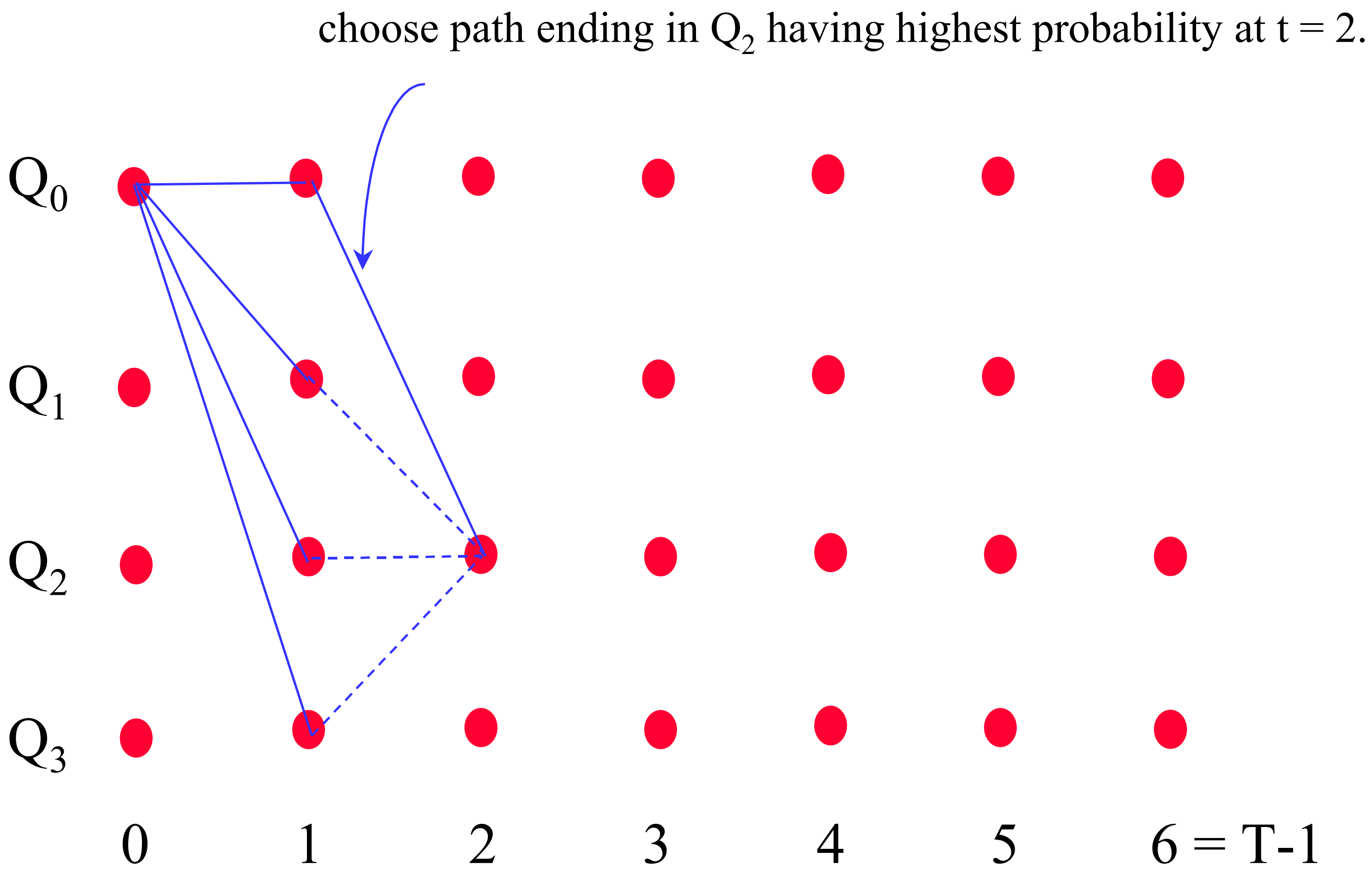


choose path ending in Q_0 having highest probability at t = 2.

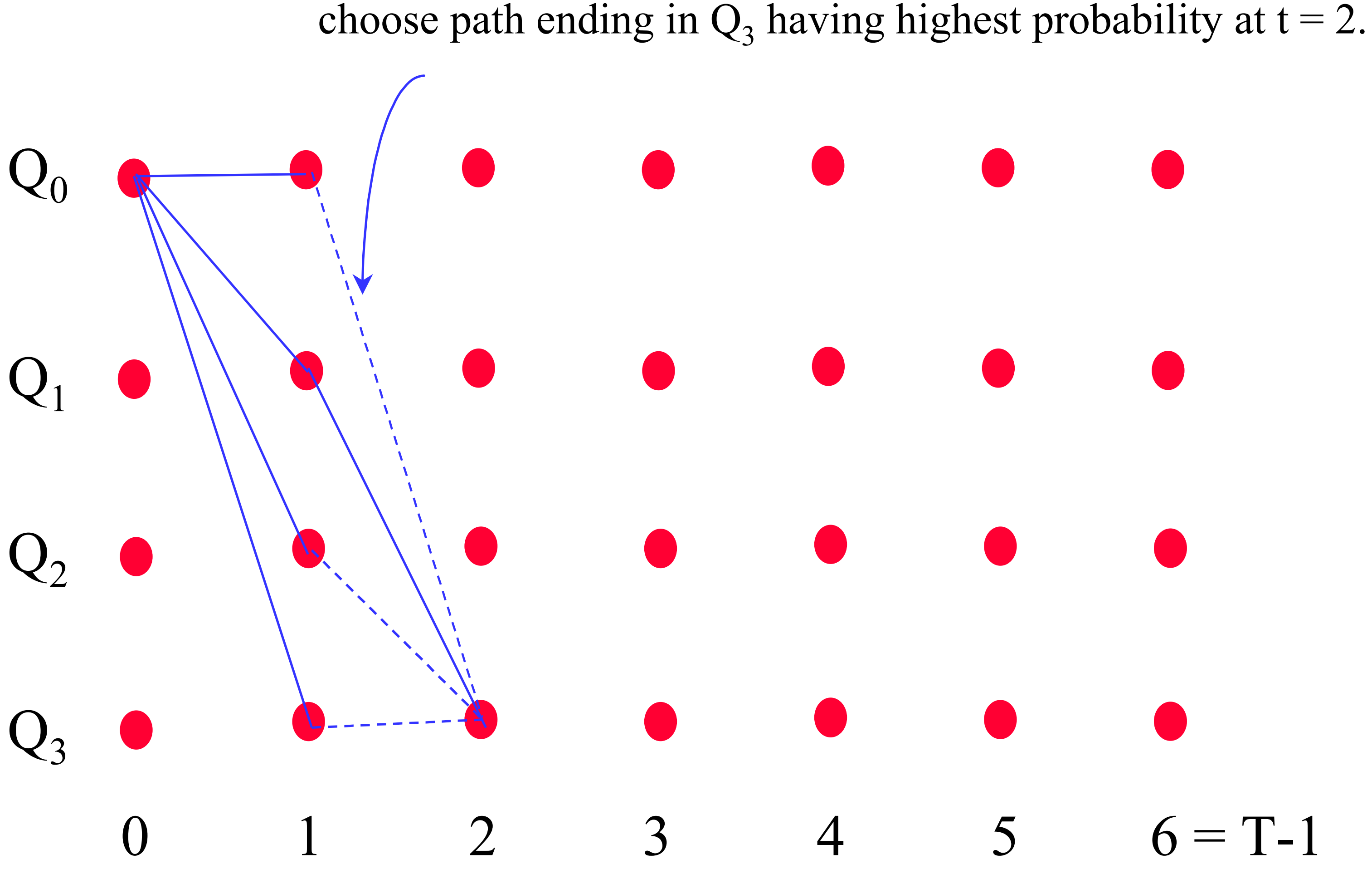
Viterbi Algorithm (cont.)


choose path ending in Q_1 having highest probability at t = 2.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1

Viterbi Algorithm (cont.)

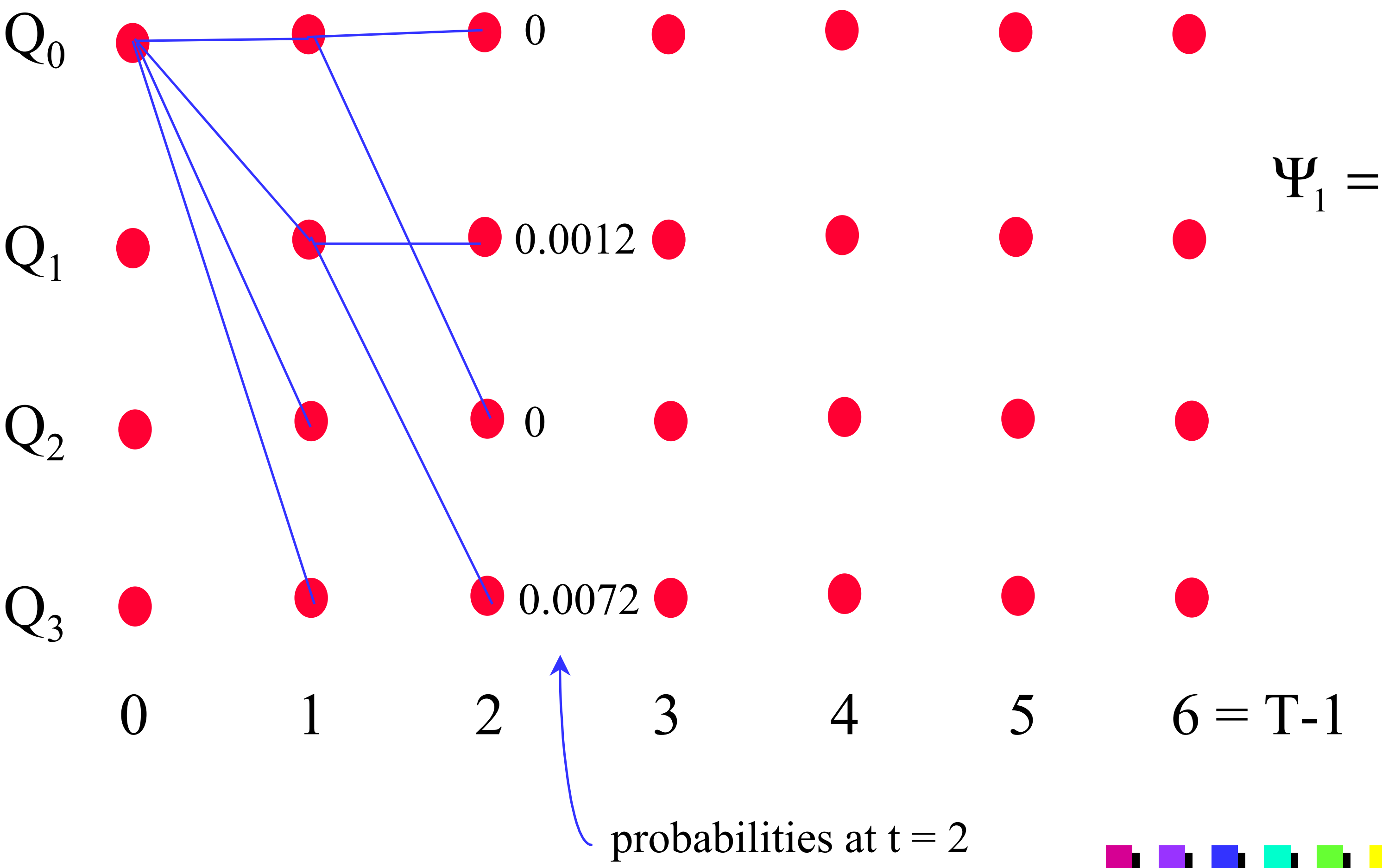

choose path ending in Q_2 having highest probability at t = 2.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1

Viterbi Algorithm (cont.)


choose path ending in Q_3 having highest probability at t = 2.
Q_0
Q_1
Q_2
Q_3
0
1
2
3
4
5
6 = T-1

Viterbi Algorithm (cont.)

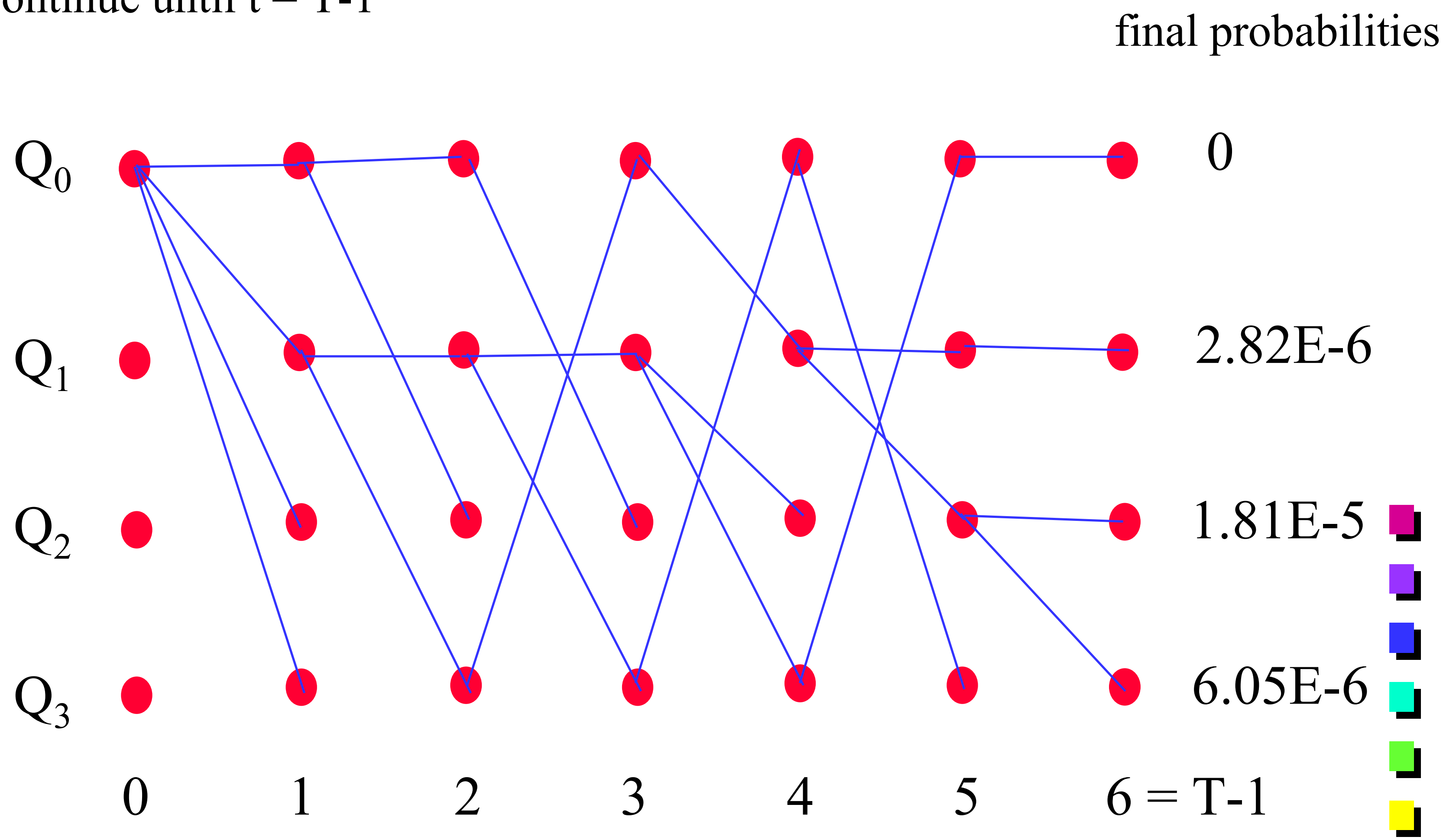
Save each of the N = 4 maximum probabilities in the vector δ_2
Save the state at t = 1 in each retained path in the vector Ψ_1



$$\Psi_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

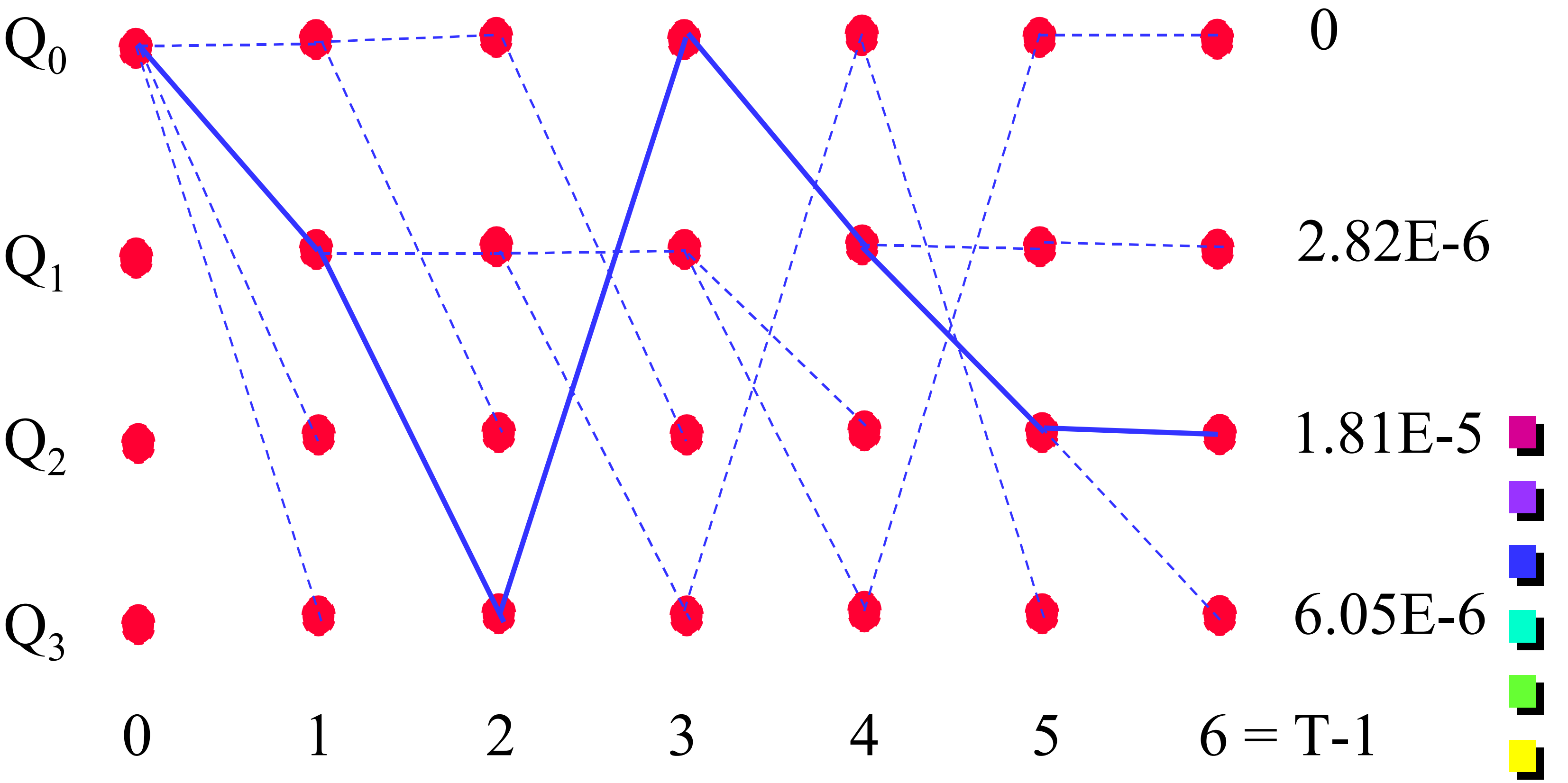
Viterbi Algorithm (cont.)

continue until t = T-1



Viterbi Algorithm (cont.)

- maximum final probability defines best path
- must backtrack through the Ψ_t to find it



The Viterbi Algorithm

- Initialization (*t* = 0):

$$d_0(i) = p_i b_i(O_0), \quad 0 \le i \le N-1$$

$$\Psi_1(i) = 0$$

- Time Recursion

$$\mathrm{For}\ 1 \le \mathrm{t} \le \mathrm{T}-1,\ \ 0 \le \mathrm{j} \le \mathrm{N}-1$$

$$d_{\mathrm{t}}(j) = \max_{0 \le i \le N\text{-}1}\left[d_{\mathrm{t}\text{-}1}(\mathrm{i}) a_{ij}\right] b_j(O_t)$$

$$\Psi_{\mathrm{t}}(j) = \arg\max_{0 \le i \le N\text{-}1}\left[d_{\mathrm{t}\text{-}1}(\mathrm{i}) a_{ij}\right]$$

The Viterbi Algorithm (cont.)

- Termination:

$$P_{\max} = \max_{0 \le i \le N-1} \left[d_{T-1}(i) \right]$$

$$i_{T-1} = \arg\max_{0 \le i \le N-1} \left[d_{T-1}(i) \right]$$

- State sequence backtracking:

For t=T-2, T-3, …, 0

$$i_t = \Psi_{t+1}\left(i_{t+1} \right)$$

Backward Variable

$$b_t(i) = \Pr(O_{t+1}, O_{t+2}, \ldots, O_{T-1} | i_t = Q_i, l)$$

To understand this variable, assume that the current time step is “*t* ”, the current state is “Q_i”, and we know the probabilities:

$$b_{t+1}(j), \quad j = 0, \ldots N-1$$

then it should be clear that:

$$b_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(O_{t+1}) b_{t+1}(j), \quad 0 \le i \le N-1, \quad 0 \le t \le T-2$$

since each of the *N* events:

$$O_{t+1}, O_{t+2}, \ldots, O_{T-1} | i_t = Q_j, \quad j = 0, \ldots, N-1$$

are disjoint.

Backward Variable (cont.)

The backward variable can be computed recursively, moving backward in time.

1. initialize at $t = T - 1$,

$$b_{T-1}(i) = 1, \quad i = 0, \ldots N - 1$$

2. for $t = T - 2 : -1 : 0$

$$b_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(O_{t+1}) b_{t+1}(j), \quad 0 \le i \le N - 1$$

More Definitions

The probability of landing in state Q_i at time t, given the observation sequence O is:

$$g_t(i) \equiv \Pr\left(i_t = Q_i | O, l\right)$$

consider the previous definitions:

$$a_t(i) = \Pr\left(O_0, O_1, \ldots, O_t, i_t = Q_i | l\right)$$

$$b_t(i) = \Pr\left(O_{t+1}, O_{t+2}, \ldots, O_{T-1} | i_t = Q_i, l\right)$$

hence, for a given model λ:

$$a_t(i) b_t(i) = \Pr\left(O_0, O_1, \ldots, O_{T-1} \cap i_t = Q_i\right)$$

More Definitions (cont.)

Hence:

$$g_{t}(i) = \frac{a_{t}(i) b_{t}(i)}{\Pr(O|\, l\,)}$$

now consider the probability that we go from state Q_i at time t to state Q_j at time $t+1$ given the observation O:

$$\xi_{\mathrm{t}}(\mathrm{i}, \mathrm{j}) \equiv \Pr\left(\mathrm{i}_{\mathrm{t}} = \mathrm{Q}_{\mathrm{i}}, \mathrm{i}_{\mathrm{t+1}} = \mathrm{Q}_{\mathrm{j}} | \mathrm{O}, \lambda\right)$$

it follows that

$$x_{\mathrm{t}}(i, j) = \frac{a_{t}(i) a_{ij} b_{j}(O_{t+1}) b_{t+1}(j)}{\Pr(O|\, l\,)}$$

More Definitions (cont.)

the average number of transitions made from Q_i:

$$\sum_{t=0}^{T-2} g_t(i)$$

the average number of transitions made from Q_i to Q_j:

$$\sum_{t=0}^{T-2} x_t(i, j)$$

Solution to Problem 3: Baum-Welch Algorithm

0. Initialize A, B, and Π

1. Compute $a_t(i)$, $\beta_t(\mathrm{i})$ and $\Pr(O|\,l\,)$

2. Compute $x_t(i, j)$ and $g_t(i)$

$$x_{\mathrm{t}}(i, j) = \frac{a_t(i) a_{ij} b_j(O_{t+1}) b_{t+1}(j)}{\Pr(O|\,l\,)} \qquad (\ ) \qquad \frac{{}_t(\ )\quad {}_t(\ )}{(\qquad)}$$

3. Compute $p_i = g\ \ (i), \qquad i \leq N - 1$

4. Compute $$\frac{\sum_{t}^{T-2}\ {}_t(\quad)}{{}^{T-2}\ g_t(\ )\ {}_{t=0}}$$

Baum-Welch Algorithm (cont.)

5. Compute

$$b_j(k) = \frac{\sum_{\substack{t=0 \\ O_t = v_k}}^{T-1} g_t(j)}{\sum_{t=0}^{T-1} g_t(j)}$$

7. go to step 2

$\Pr(O| l)$ should continue to increase until A, B, and Π converge to optimum values, at which point the algorithm is terminated.

Case Study: Coast et al.

- Used continuous density for observations:

$$b_i(v) = \frac{1}{\sqrt{2ps_i}} e^{-0.5((v-m_i)/s_i)^2}$$

  This alters most of the formulas we looked at but the basic ideas remain the same.

- Observations consisted of actual ECG samples.
- Used several rhythm HMM models in parallel
- Viterbi algorithm was used to select the most likely sequence (and hence rhythm type).