Hidden Markov Models (HMM)

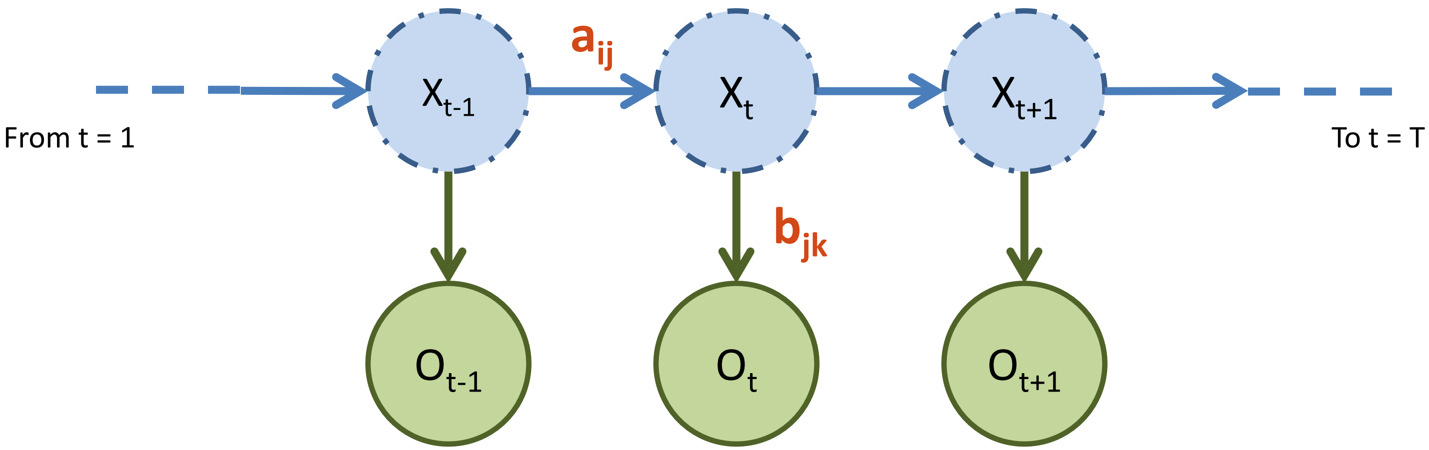
Tutorial Part-1

FOCUS: • Concepts • Implementations

Akshaya Thippur
Judith Butepage

TERMINOLOGY:

Time instants	t *in* {1,2 . . . T}
Hidden States / States / Emitters	X_t
Outputs / Emissions / Observations / Visible States	O_t
All possible states / states set	X_t *in* {1,2 . . . N}
All possible emissions / emissions set	O_t *in* {1,2 . . . K}
Initial state distribution / Initial state probabilities	p_i in q *or* π_i in π
Transition probabilities / State transition probabilities	a_{ij} *in* row-stochastic matrix A
Emission probabilities / Observation probabilities	b_{jk} *in* row-stochastic matrix B



HMM PROBLEMS:

0. Predicting most likely current emission

1. Evaluation Problem = ?

2. Decoding Problem = ?

3. Learning Problem =?

☺ Puppy Platone Example ☺

Platone is a 2 month old adorable Collie. Platone fills his days learning with his best friend and master Alex and from his simple yet interesting world. Most of the day he is playing or exploring. This suggests that he is active and entertained or bored and looking for something new to do. Platone also barks at times for attention from Alex when he is hungry or needs something. Then he gets tired and sleepy quite often and just lies down on the floor. Alex is learning to understand his puppy and being a computer science geek^^ he decides to model Platone using a HMM …

What are the states?

What are the emissions?

What are the model parameters?

☺ Puppy Platone Example ☺ - BREAKING IT DOWN

Platone is a 2 month old adorable Collie. Platone fills his days learning with his best friend and master Alex and from his simple yet interesting world. Most of the day he is playing or exploring. This suggests that he is active and entertained or bored and looking for something new to do. Platone also barks at times for attention from Alex when he is hungry or needs something. Then he gets tired and sleepy quite often and just lies down on the floor. Alex is learning to understand his puppy and being a computer science geek^^ he decides to model Platone using a HMM …

What are the states?

[Active, Bored, Hungry, Sleepy]

What are the emissions?

[Play, Explore, Bark, Lie Down]

What are the model parameters?

A = matrix 4states x 4states

B = matrix 4states x 4emissions

q = matrix 4states x 1 initial probabilities

PROBLEM 0: Most likely current emission

Given:

- A, B, q
- Distribution of hidden states at time (t)

Unknown:

- The exact time instance
- Emission sequence until time (t)
- Hidden state sequence until time (t)

Find:

- Most likely next emission at time (t)

PROBLEM 0: Most likely current emission (Puppy Platone Example)

Given:

A =

$X_t \mid X_{t+1}$	A	B	H	S
A	0.6	0.1	0.1	0.2
B	0.0	0.3	0.2	0.5
H	0.8	0.1	0.0	0.1
S	0.2	0.0	0.1	0.7

B =

$X_t \mid O_t$	p	e	b	l
A	0.6	0.2	0.1	0.1
B	0.1	0.4	0.1	0.4
H	0.0	0.0	0.7	0.3
S	0.0	0.0	0.1	0.9

$P(X_t)$ =

A	B	H	S
0.4	0.2	0.1	0.3

Find:

- $P(O_t \mid A, B, P(X_t))$

Solution:

- On the board …

PROBLEM 1: Evaluation

Given:

- A, B, q
- Emission sequence **O** = $\{O_1, O_2 \ldots O_T\}$

Unknown:

- Hidden state sequence **X** = $\{X_1, X_2 \ldots X_T\}$ that actually produced **O**.

To Find:

- Probability that the given sequence **O** occurred regardless of which **X** produced the sequence.

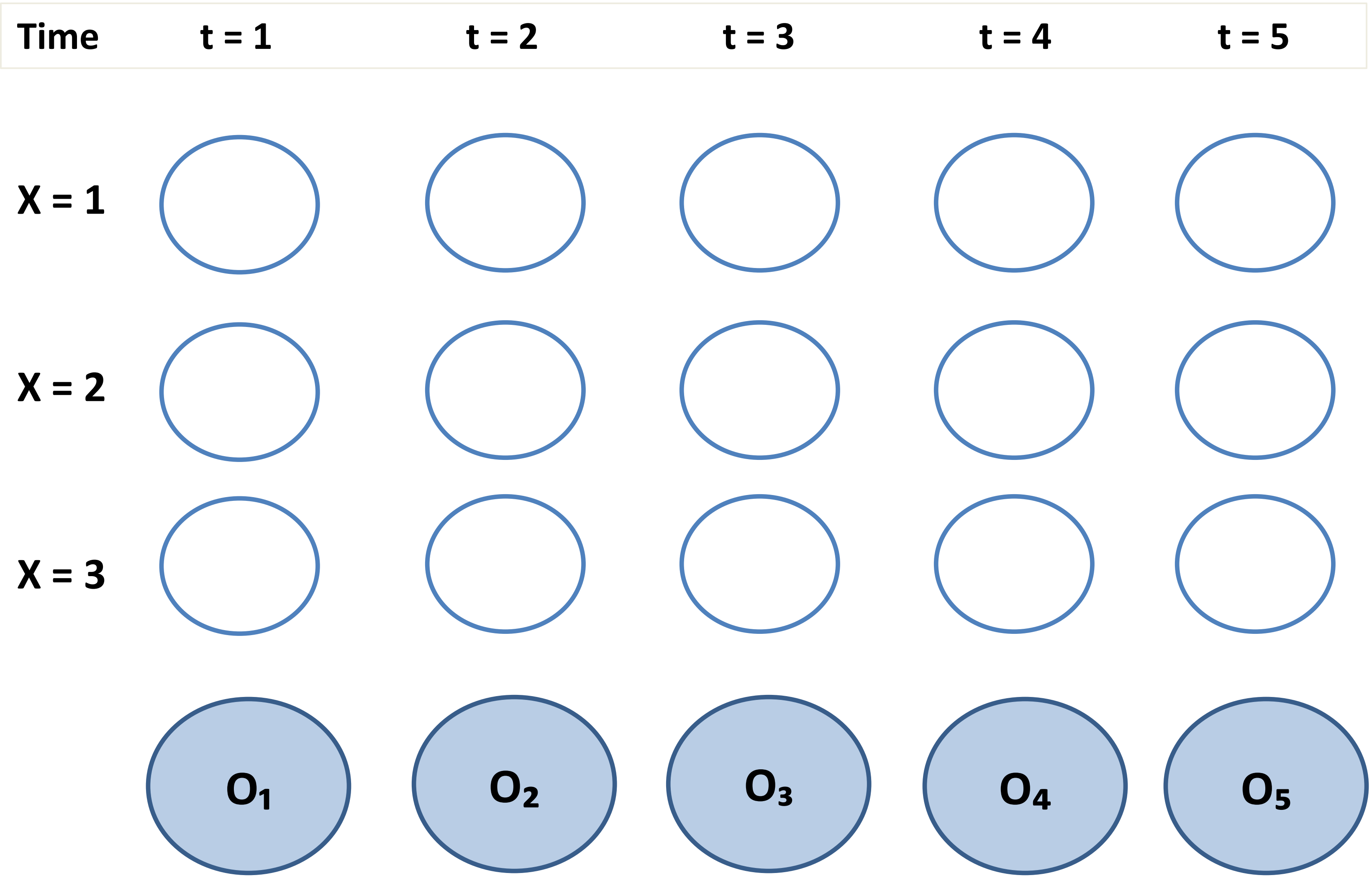
FORWARD ALGORITHM: (α - PASS)

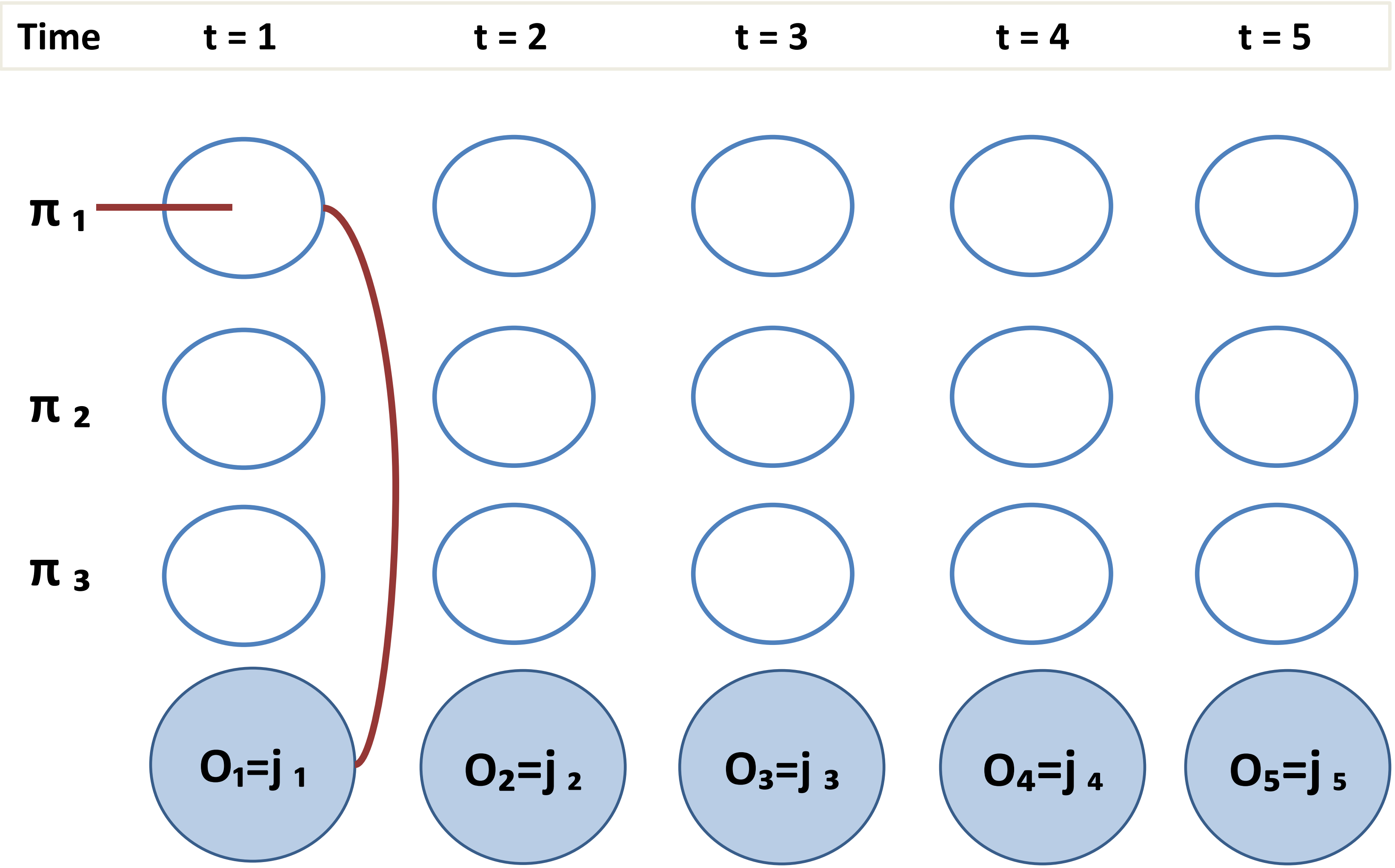
$\alpha_t(i)$ = Probability that the model is in the hidden state $X_t(i)$ (i *in* [1,2,…,N])

&&

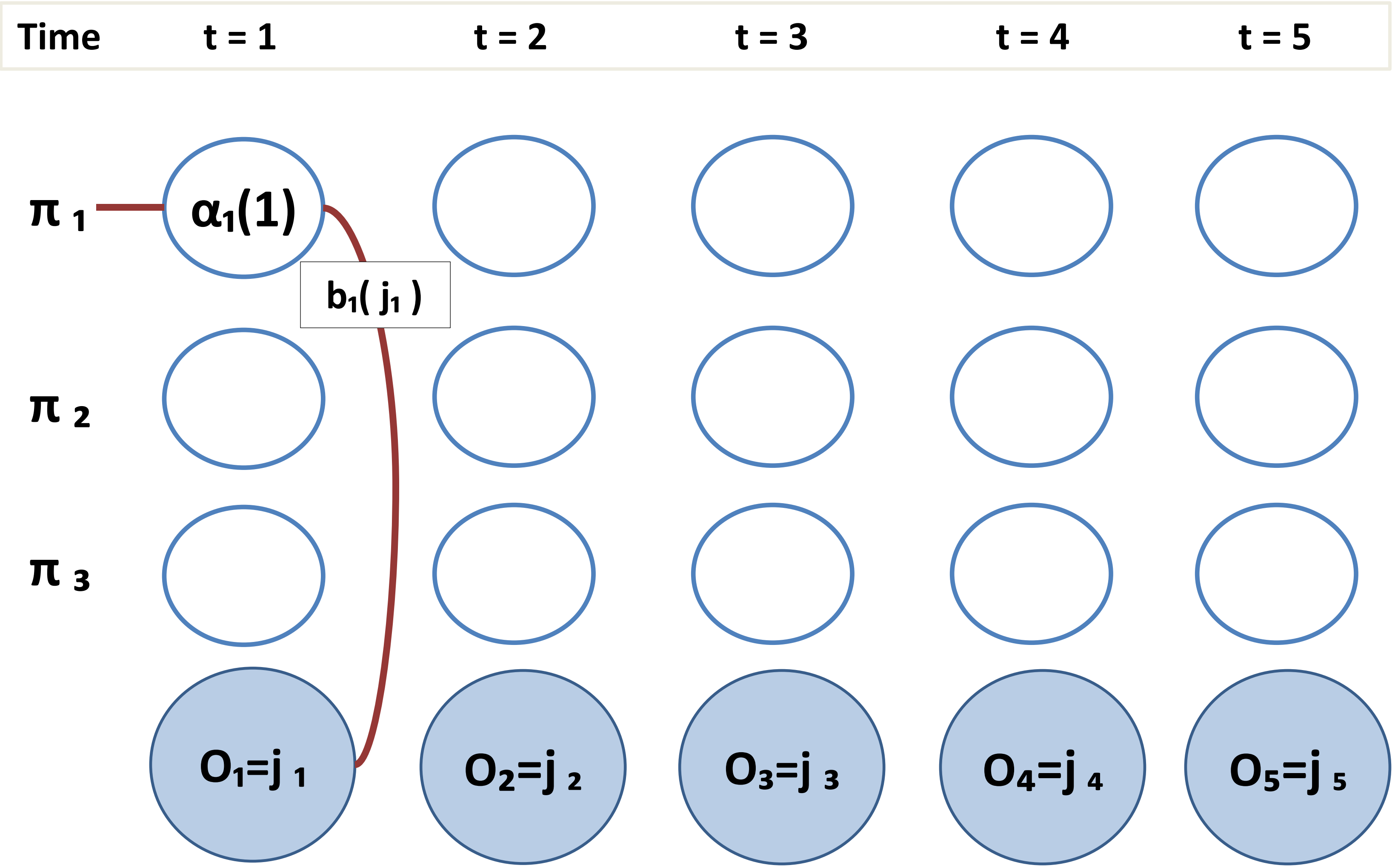
has generated the emission sequence up to O_t, where O_t has taken a value $O_t(k)$ (k *in* [1,2,…,K]) according to the emission sequence already observed.

- Introduce: $\alpha_t(i) = p(O_{1:t}, X_t = i|\lambda) \quad \forall \quad t = 1, \ldots, T$
- Initialize as: $\alpha_1(i) = \pi_i b_i(O_1)$
- For 2≤t≤T: $\alpha_t(i) = [\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}] b_i(O_t)$
- Which gives us: $p(O_{1:T}|\lambda) = \sum_{i=1}^{N} p(O_{1:T}, X_T = i|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

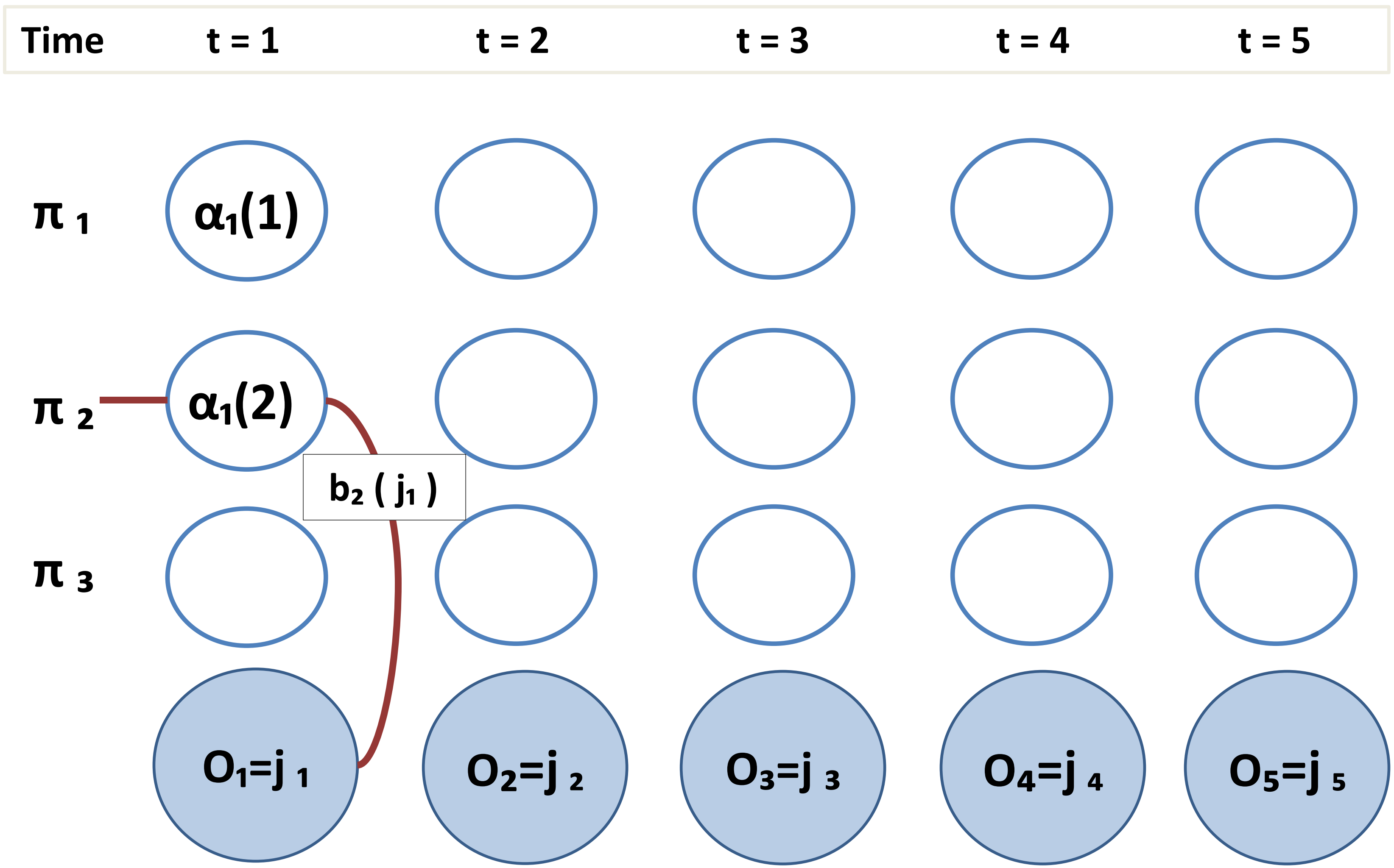

Time
t = 1
t = 2
t = 3
t = 4
t = 5
X = 1
X = 2
X = 3
O_1
O_2
O_3
O_4
O_5



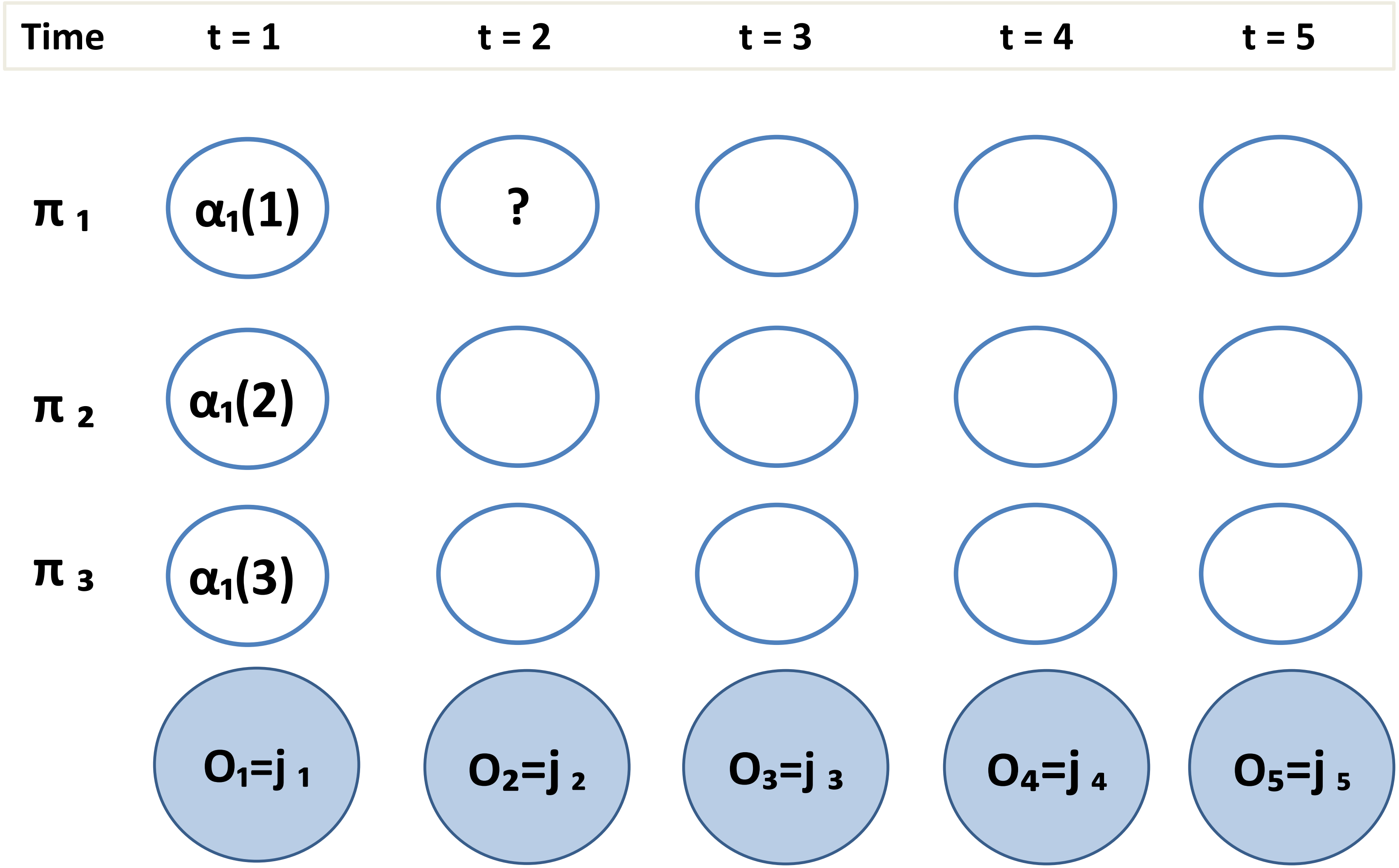
$$\alpha_1(1) = P(\, O_1{=}j_1 \mid X_1 = 1)\, P(X_1 = 1)$$

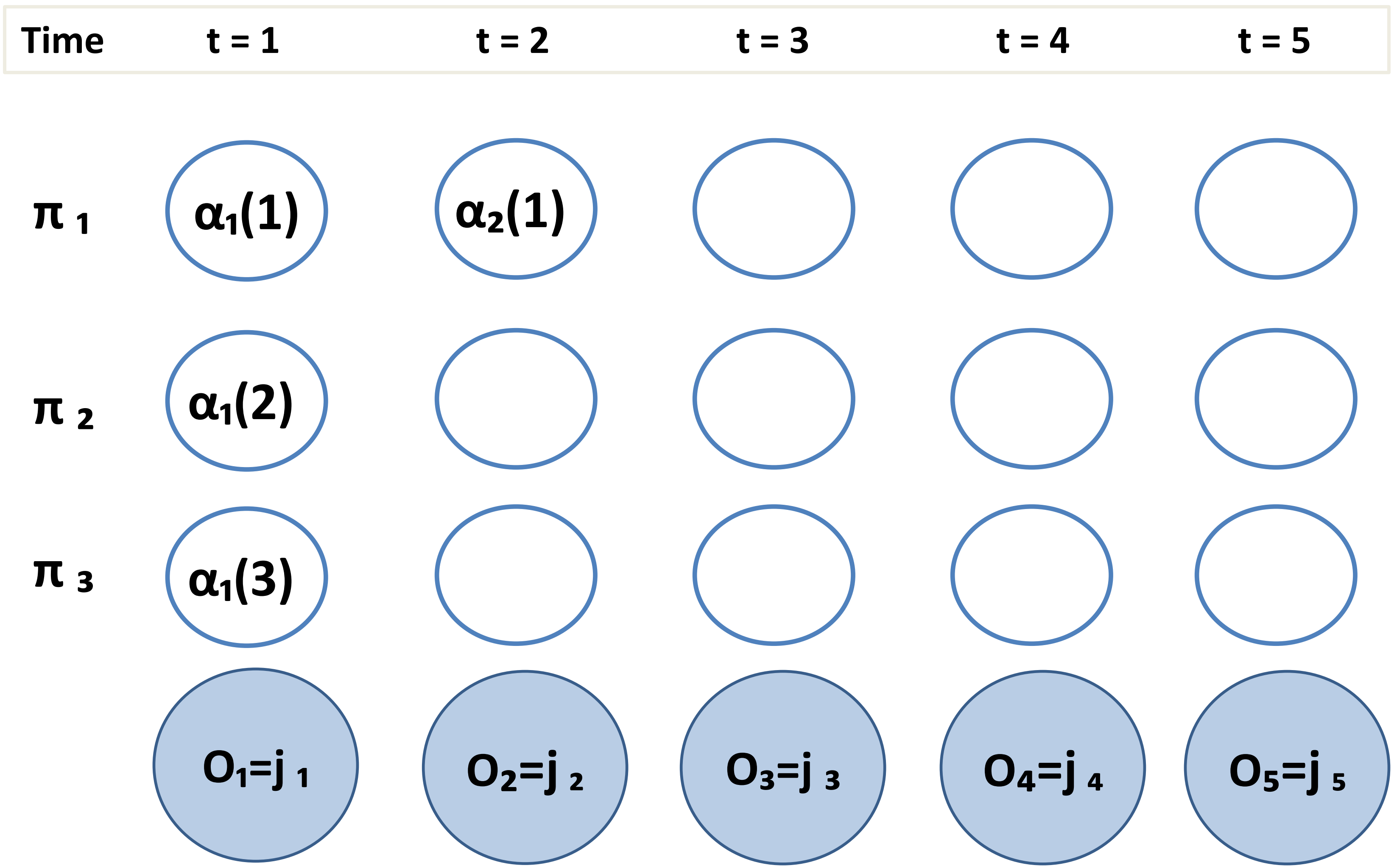


$$\alpha_1(1) = P(O_1=j_1 \mid X_1 = 1)\, P(X_1 = 1) = b_1(j_1)\, \pi_1$$

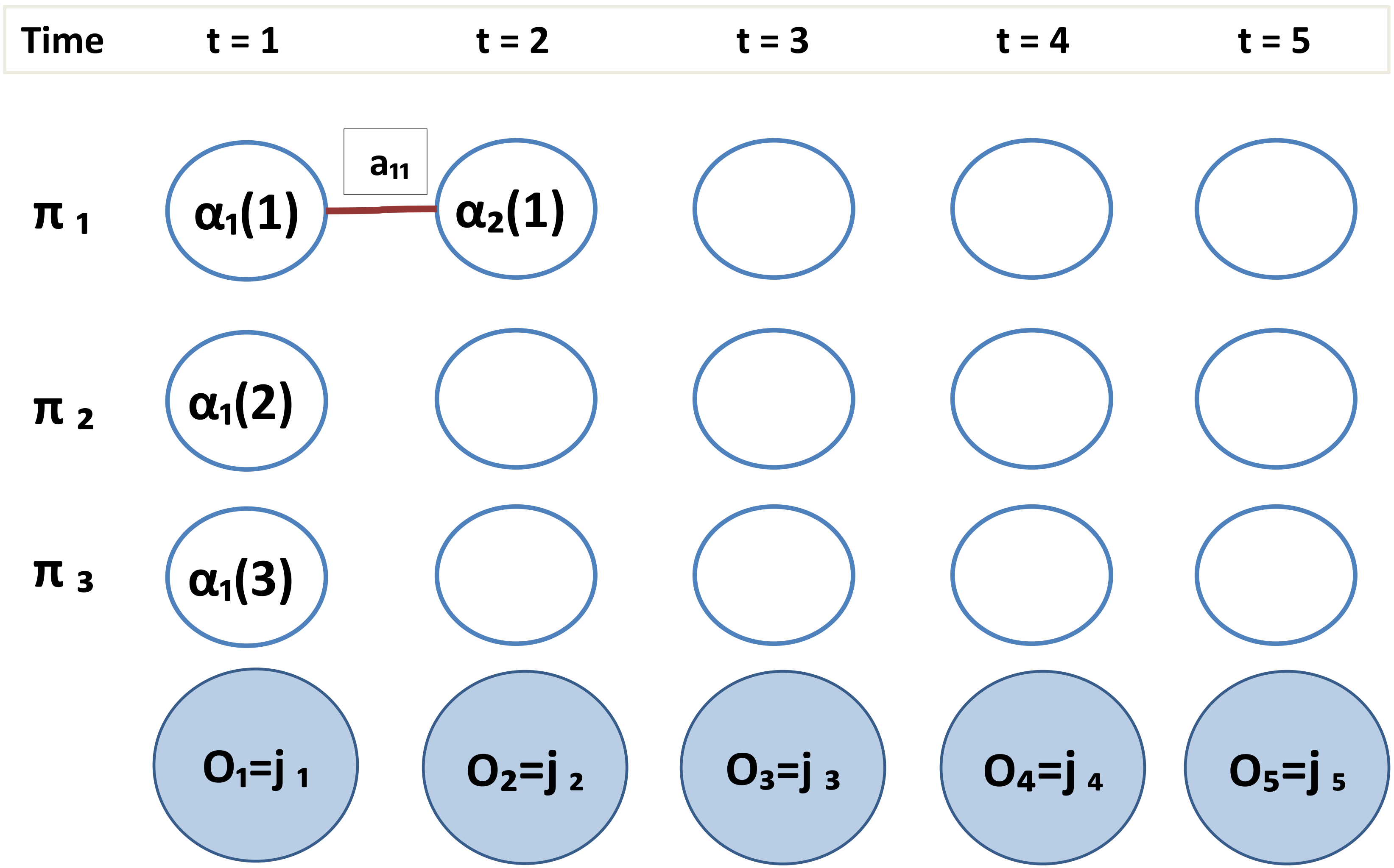


$$\alpha_1(2) = P(O_1 = j_1 \mid X_1 = 2)\, P(X_1 = 2) = b_2(j_1)\, \pi_2$$


Time
t = 1
t = 2
t = 3
t = 4
t = 5
π 1
π 2
π 3
α₁(1)
?
α₁(2)
α₁(3)
O₁=j 1
O₂=j 2
O₃=j 3
O₄=j 4
O₅=j 5

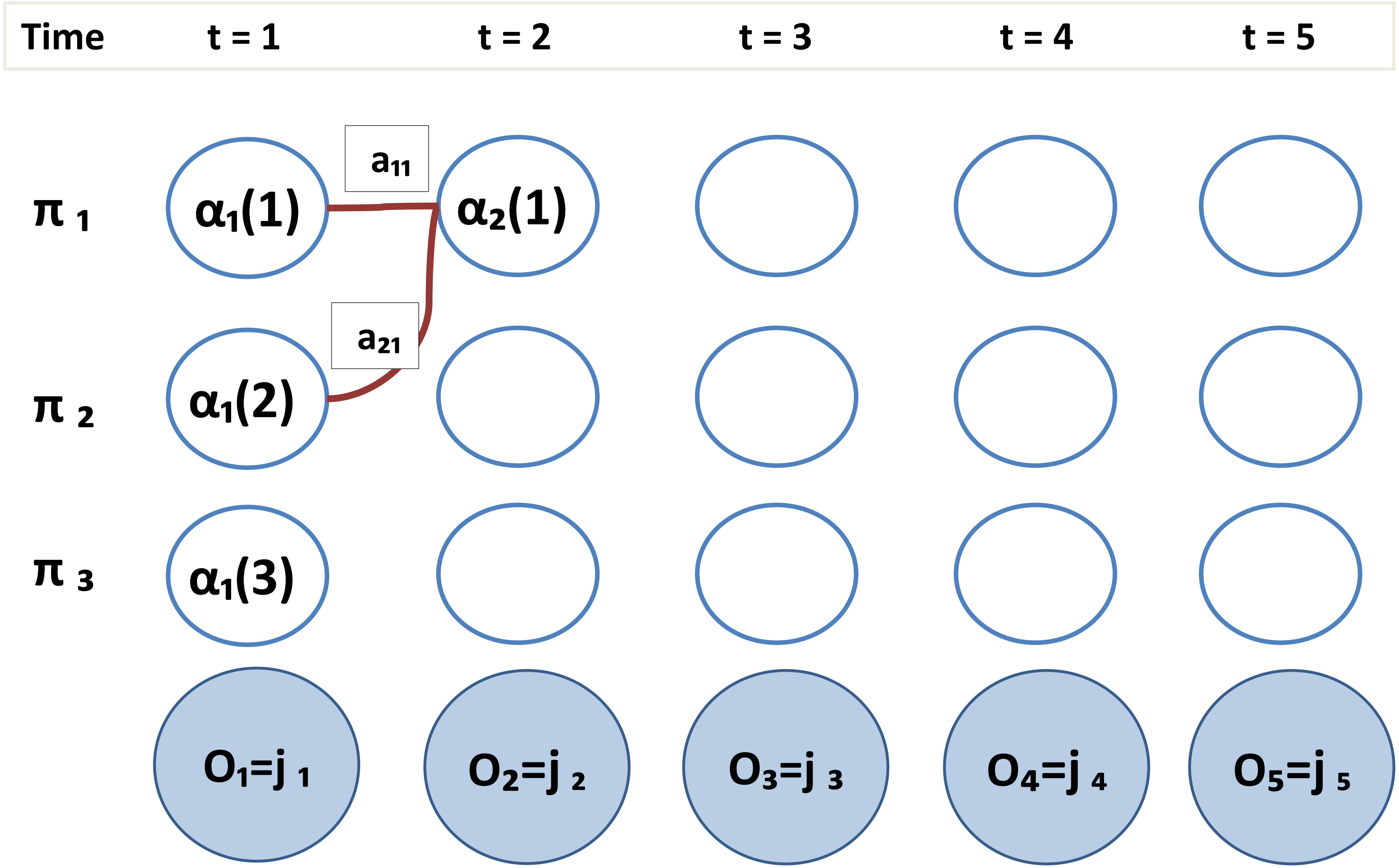


$$\alpha_2(1) = \left(\sum_i P(X_2 = 1 \mid X_1 = i)\, P(O_1 = j_1, X_1 = i)\right) P(O_2 = j_2 \mid X_2 = 1)$$



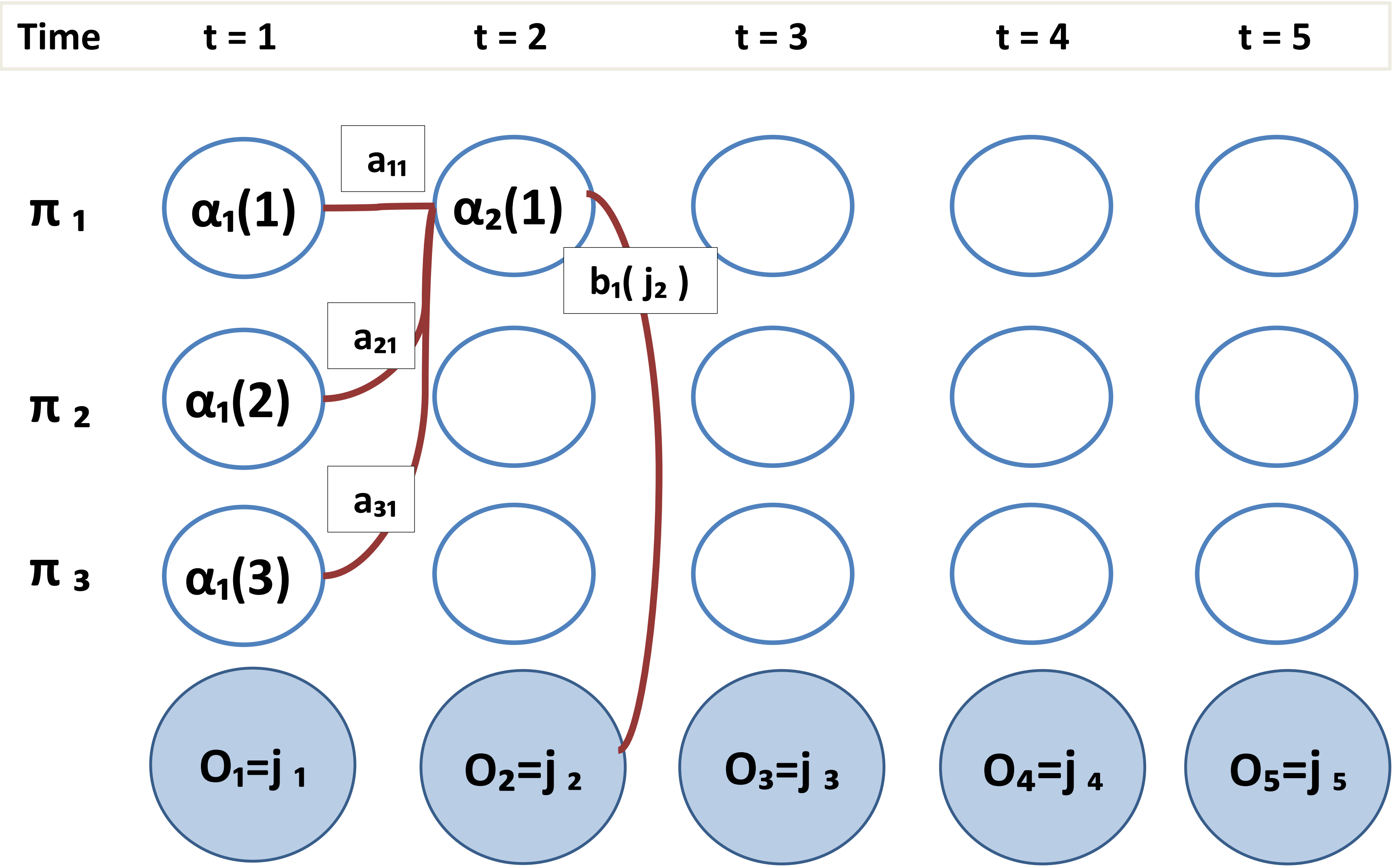
$$\alpha_2(1) = \left(\textstyle\sum_i P(X_2 = 1 | X_1 = i)\, P(O_1 = j_1, X_1 = i)\right) P(O_2 = j_2 \mid X_2 = 1)$$

$$= \left(\textstyle\sum_i a_{i1}\, \alpha_1(i)\right) b_1(j_2)$$



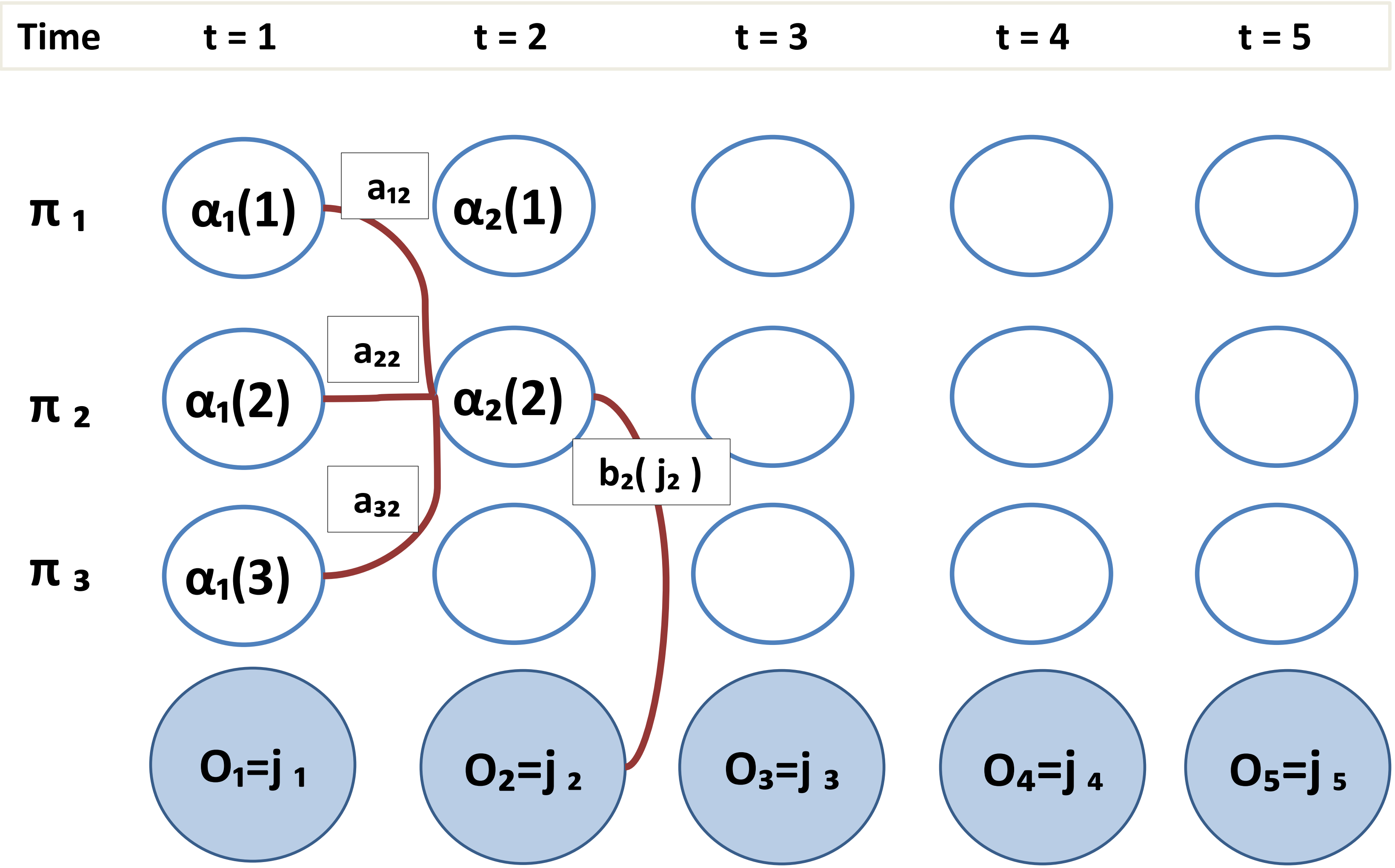
$$\alpha_2(1) = \left(\sum_i P(X_2 = 1 | X_1 = i)\, P(O_1=j_1, X_1 = i)\right) P(O_2=j_2 \mid X_2 = 1)$$

$$= \left(\sum_i a_{i1}\, \alpha_1(i)\right)\, b_1(j_2)$$



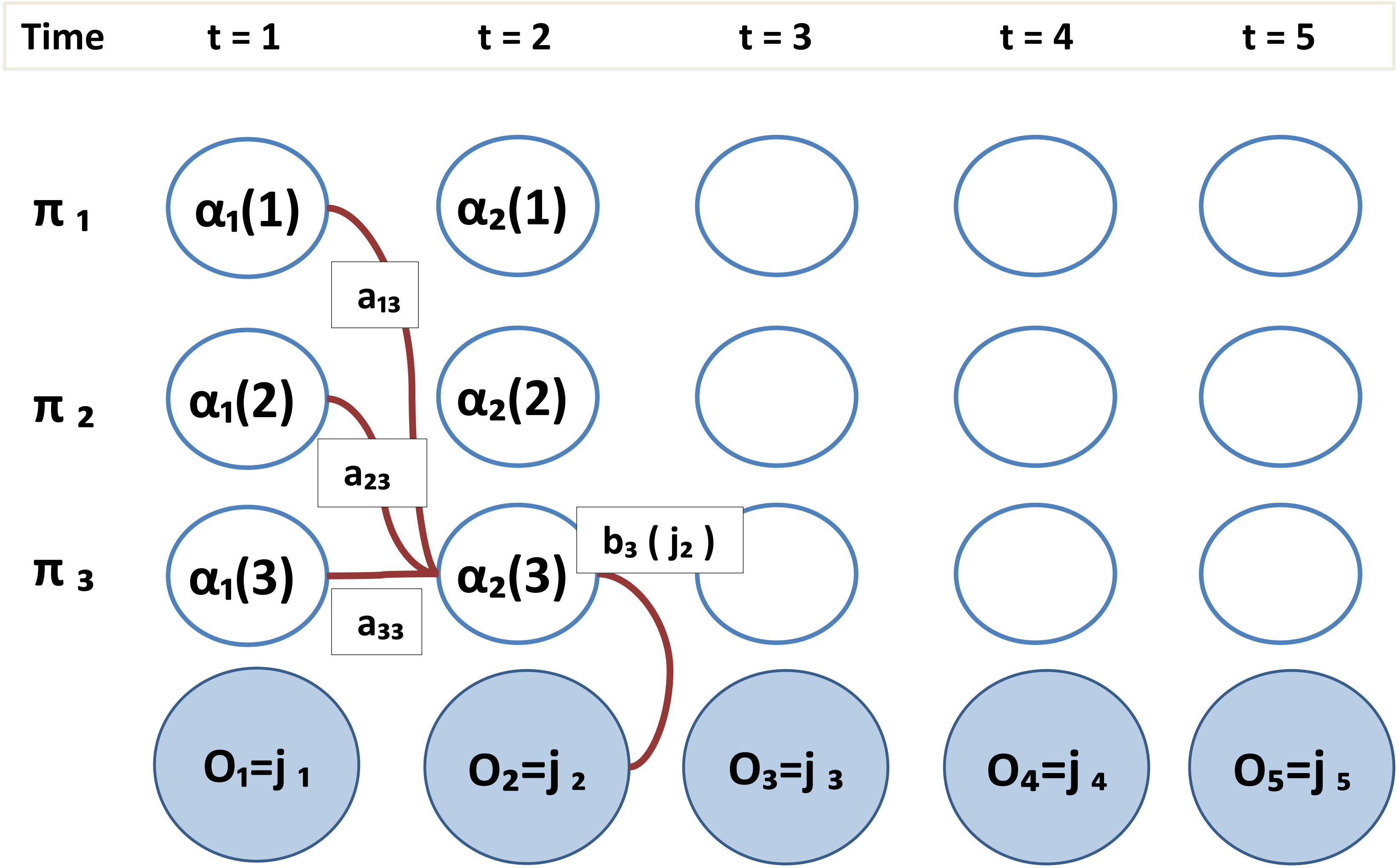
$$\alpha_2(1) = \left(\sum_i P(X_2 = 1 | X_1 = i)\, P(O_1=j_1, X_1 = i)\right) P(O_2=j_2 \mid X_2 = 1)$$

$$= \left(\sum_i a_{i1}\, \alpha_1(i)\right) b_1(j_2)$$



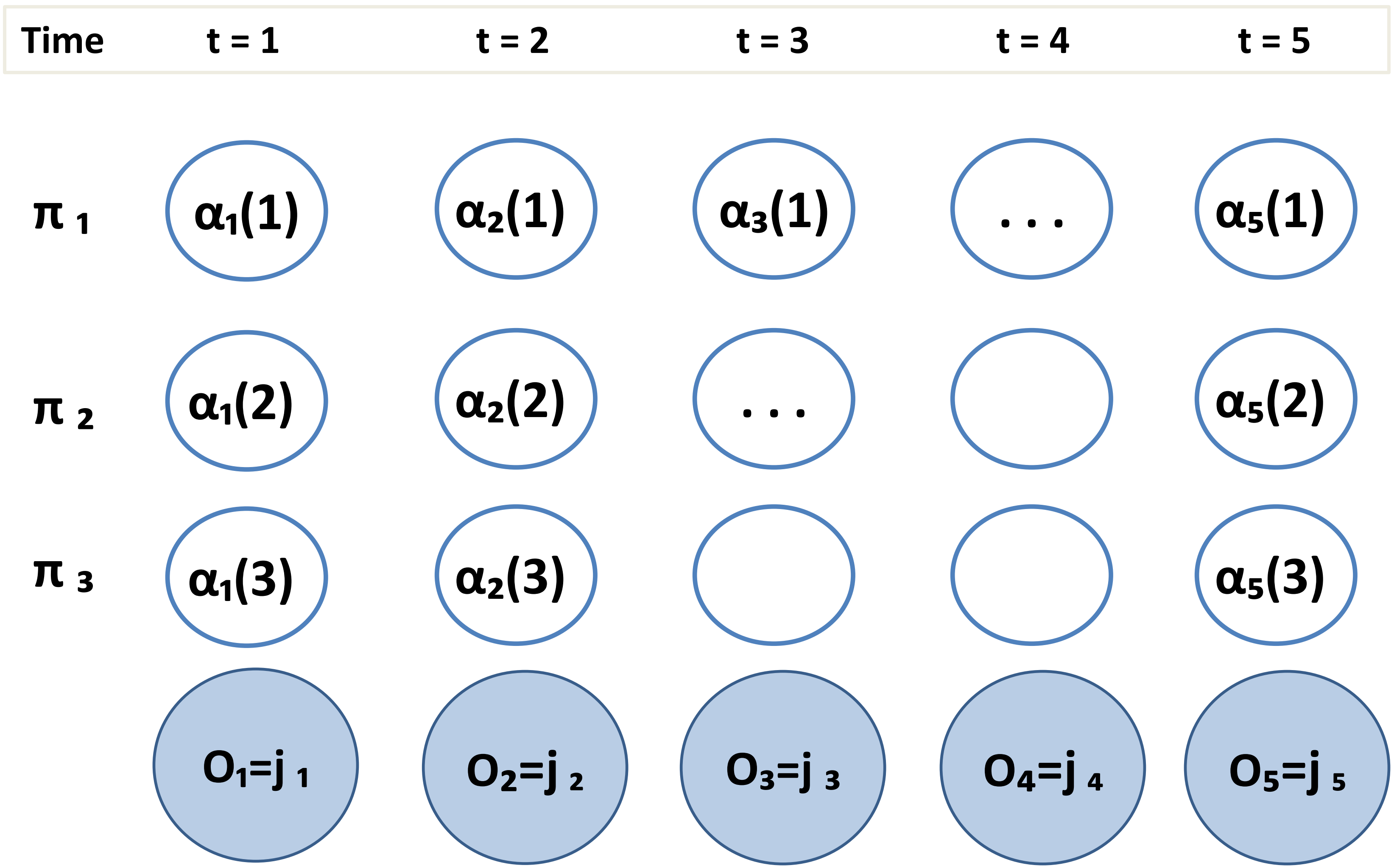
$$\alpha_2(2) = \left(\sum_i P(X_2 = 2 \,|\, X_1 = i)\, P(O_1=j_1, X_1 = i)\right) P(O_2=j_2 \,|\, X_2 = 2)$$

$$= \left(\sum_i a_{i2}\, \alpha_1(i)\right) b_2(j_2)$$



$$\alpha_2\,(3) \;=\; \left(\textstyle\sum_i P(\,X_2 = 3\,|\,X_1 = i)\; P(O_1{=}j_1\,,\,X_1 = i)\,\right) P(O_2{=}j_2 \mid X_2 = 3)$$

$$=\; \left(\textstyle\sum_i a_{i3}\,\alpha_1(i)\right)\; b_3\,(\,j_2\,)$$



$$P(O_{1..5}) \quad = \quad \sum_i \alpha_5\,(i)$$

FORWARD ALGORITHM: (α - PASS)

$\alpha_t(i)$ = Probability that the model is in the hidden state $X_t(i)$ (i *in* [1,2,…,N])

&&

has generated the emission sequence up to $O_{t,}$ where O_t has taken a value $O_t(k)$ (k *in* [1,2,…,K]) according to the emission sequence already observed.

- Introduce: $\alpha_t(i) = p(O_{1:t}, X_t = i|\lambda) \quad \forall \quad t = 1, \ldots, T$
- Initialize as: $\alpha_1(i) = \pi_i b_i(O_1)$
- For 2≤t≤T: $\alpha_t(i) = [\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}] b_i(O_t)$
- Which gives us: $p(O_{1:T}|\lambda) = \sum_{i=1}^{N} p(O_{1:T}, X_T = i|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

PROBLEM 1: Evaluation (Puppy Platone Example)

Given:

A =

X_t | X_{t+1}	**A**	**B**	**H**	**S**
A	0.6	0.1	0.1	0.2
B	0.0	0.3	0.2	0.5
H	0.8	0.1	0.0	0.1
S	0.2	0.0	0.1	0.7

B =

X_t | O_t	**p**	**e**	**b**	**l**
A	0.6	0.2	0.1	0.1
B	0.1	0.4	0.1	0.4
H	0.0	0.0	0.7	0.3
S	0.0	0.0	0.1	0.9

q =

A	**B**	**H**	**S**
0.5	0.0	0.0	0.5

O = { **l, p, p, b** }

Find:

- P(**O** | A, B, q)

Solution:

- On the board …

PROBLEM 2: Decoding

… See Tutorial: HMM2

PROBLEM 3: Learning

Given:

- Emission sequence **O** = $\{O_1, O_2 \ldots O_T\}$ (T is a few orders larger than usually seen in text book problems)
- Initial guesses of A, B (maybe)

Unknown:

- Hidden state sequence **X** = $\{X_1, X_2 \ldots X_T\}$ that actually produced **O**.

To Find:

- A, B

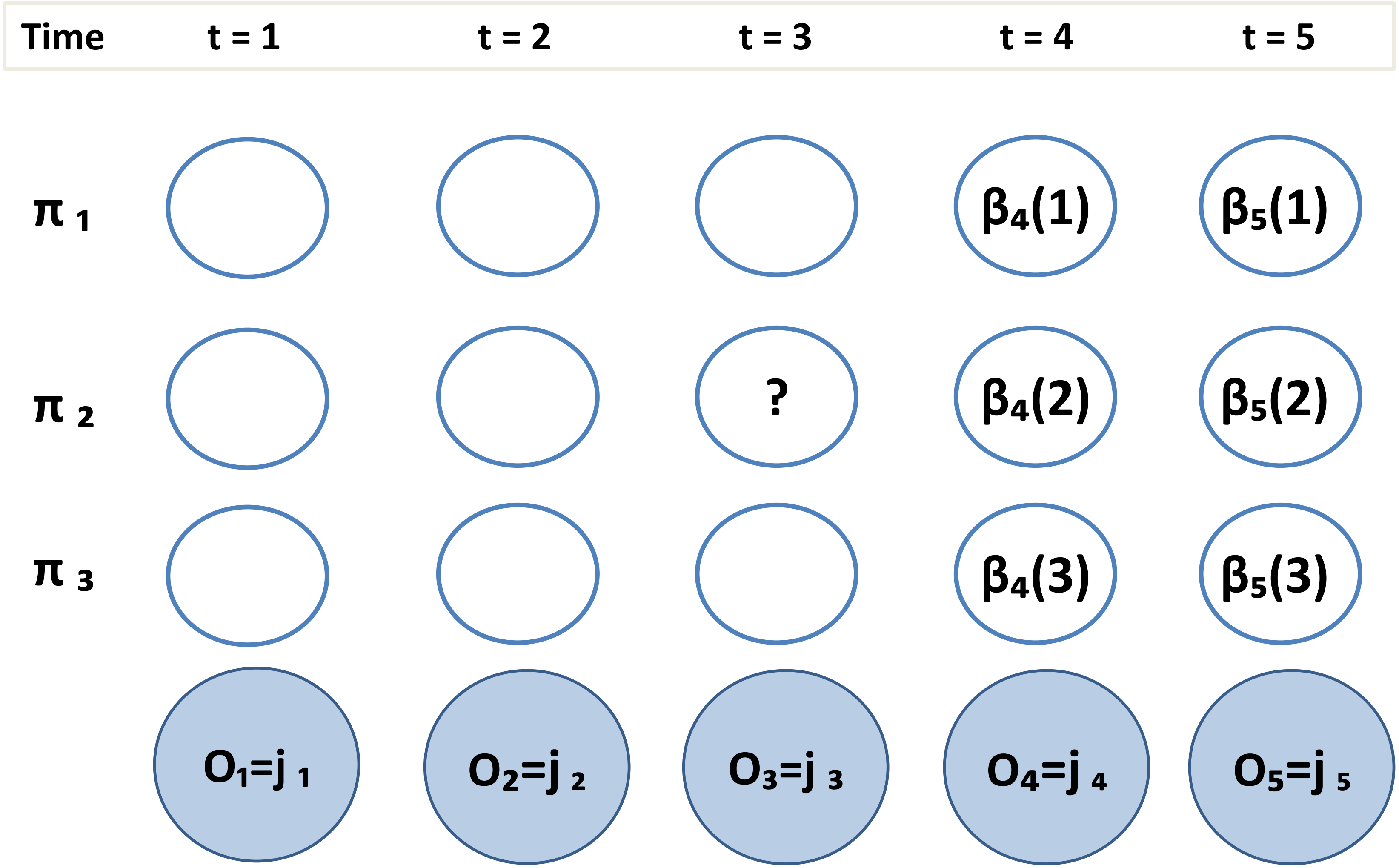
BACKWARD ALGORITHM: (β - PASS)

$\beta_t(i)$ = Probability that the model is in the hidden state $X_t(i)$ (i *in* [1,2,…,N])

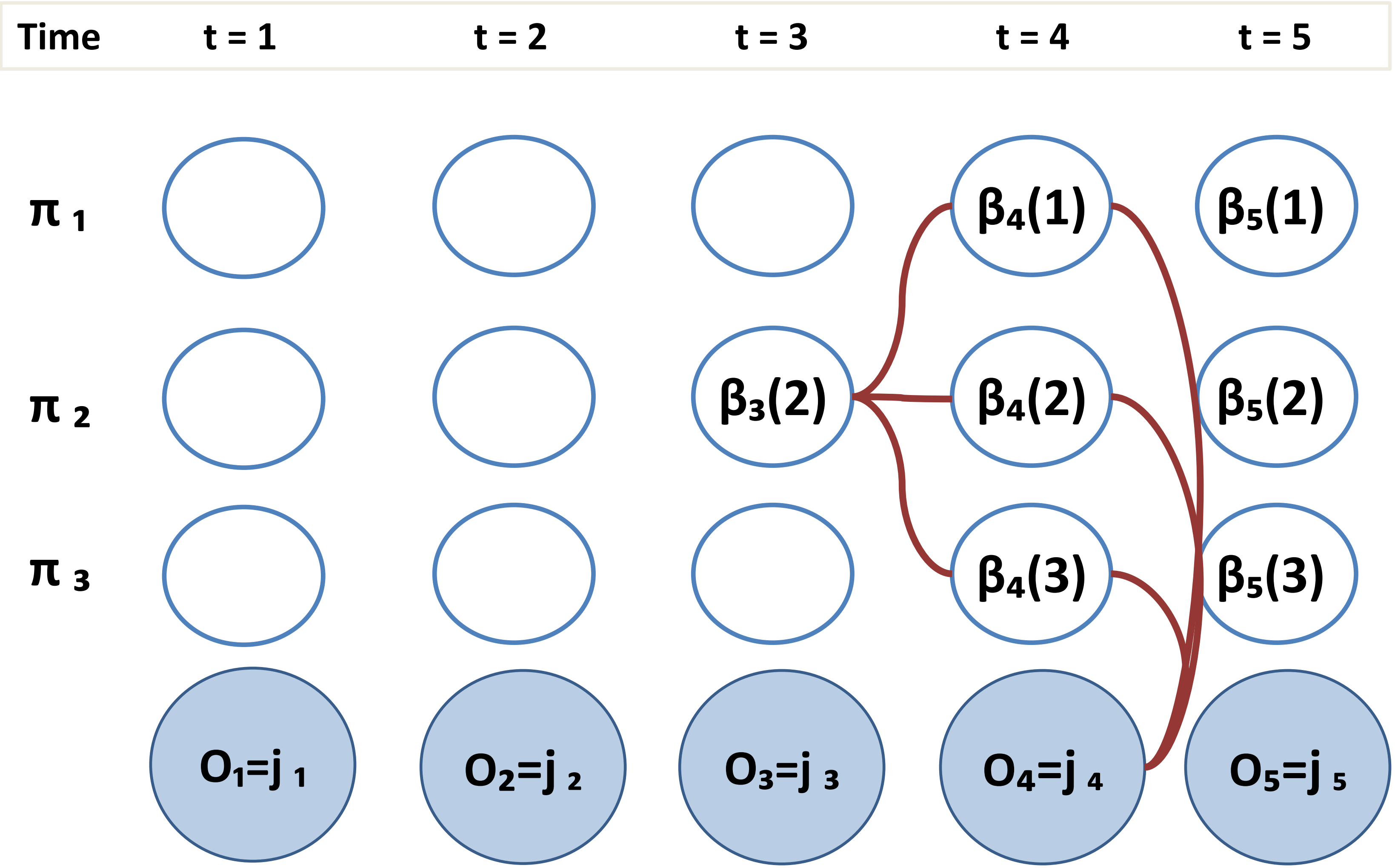
&&

will generate the remainder of the emission sequence, from O_{t+1} to O_T, as specified by the emission sequence **O**.

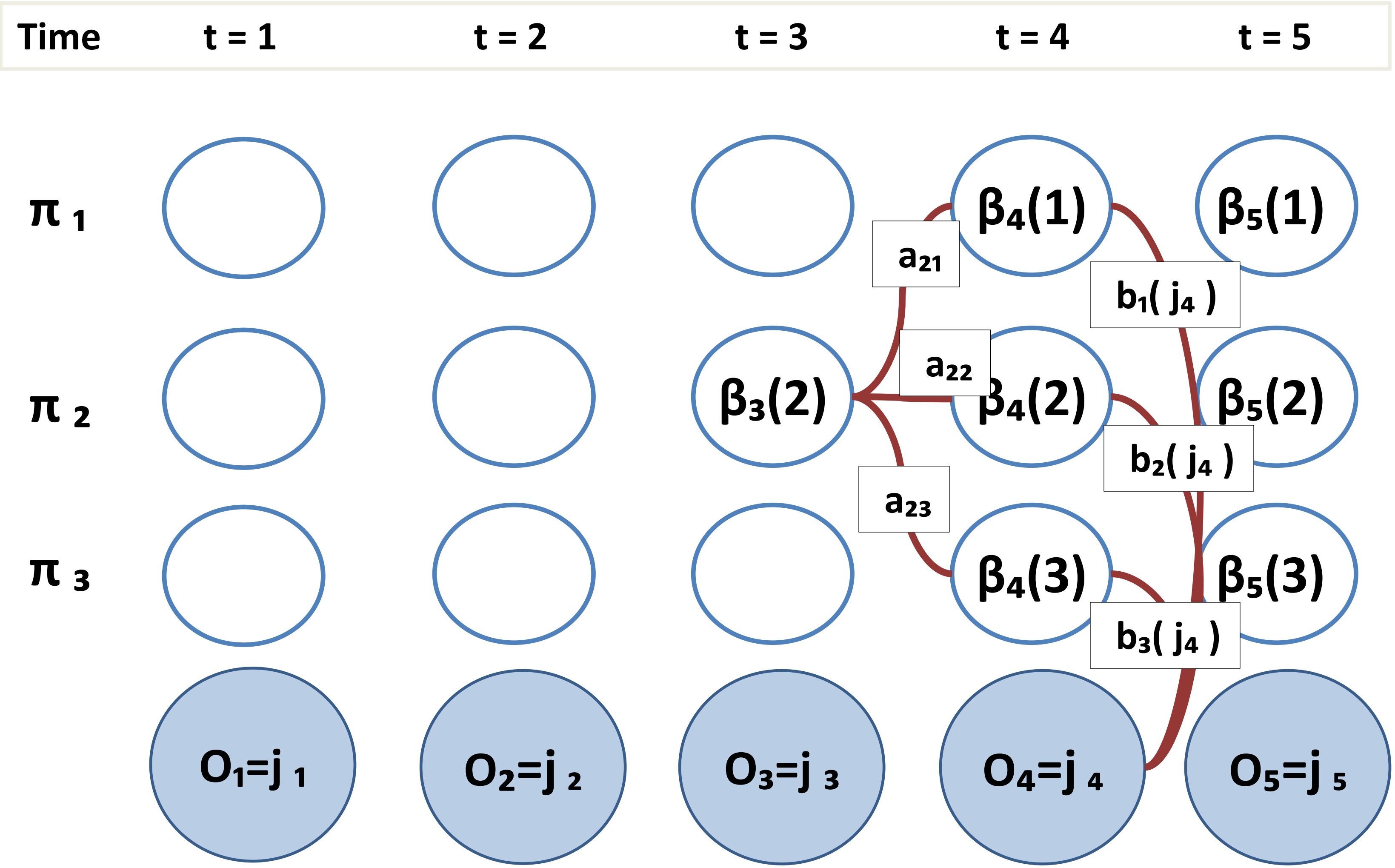
- Assume measurement sequence $O_{1:T}$
- Introduce: $\beta_t(i) = p(O_{t+1:T} | X_t = i, \lambda)$
- Initialize: $\beta_T(i) = 1, \forall i = 1, \ldots, N$
- For t<T: $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$



$$\beta_3(2) = P(O_{4..5} \mid X_3 = 2)$$



$$\beta_3(2) = \sum_i P(X_4 = i \mid X_3 = 2) P(O_4=j_4 \mid X_4 = i) P(O_5 \mid X_4 = i)$$



$$\beta_3(2) = \sum_i a_{2i}\, b_i(j_4)\, \beta_4(i)$$

BACKWARD ALGORITHM: (β - PASS)

$\beta_t(i)$ = Probability that the model is in the hidden state $X_t(i)$ (i *in* [1,2,...,N])

&&

will generate the remainder of the emission sequence, from O_{t+1} to O_T, as specified by the emission sequence **O**.

- Assume measurement sequence $O_{1:T}$
- Introduce: $\beta_t(i) = p(O_{t+1:T} | X_t = i, \lambda)$
- Initialize: $\beta_T(i) = 1, \forall i = 1, \ldots, N$
- For t<T: $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

From Prof. Patric Jensfelt – Lecture Slides DD2380

GAMMA CALCULATIONS: (1)

1) **Di – Gamma Function**

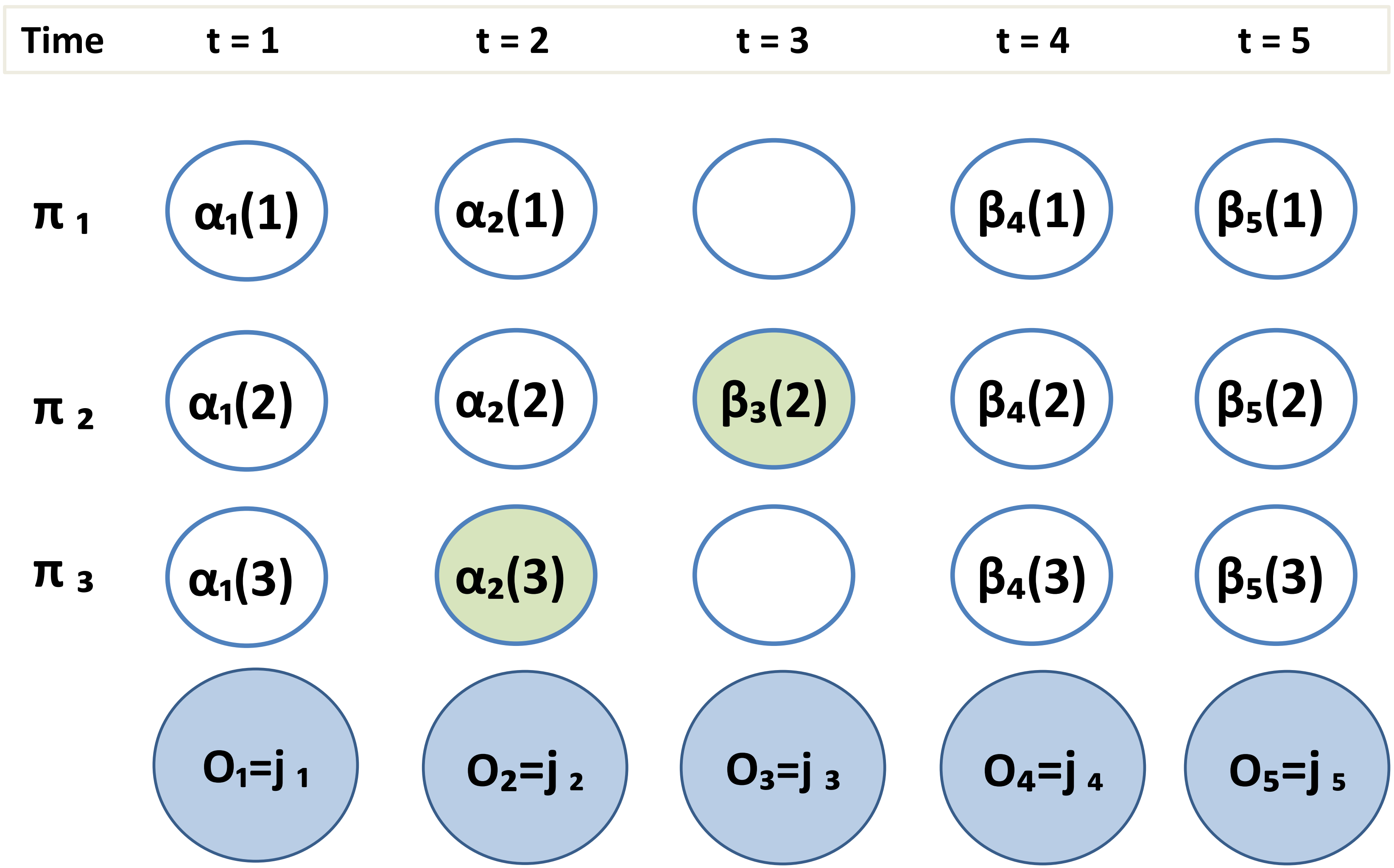
$$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} \quad = p(X_t = i, X_{t+1} = j | O_{1:T}, \lambda)$$

<u>Interpretation</u>: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i) && at time (t+1) the hidden state is (X_{t+1}=j)?

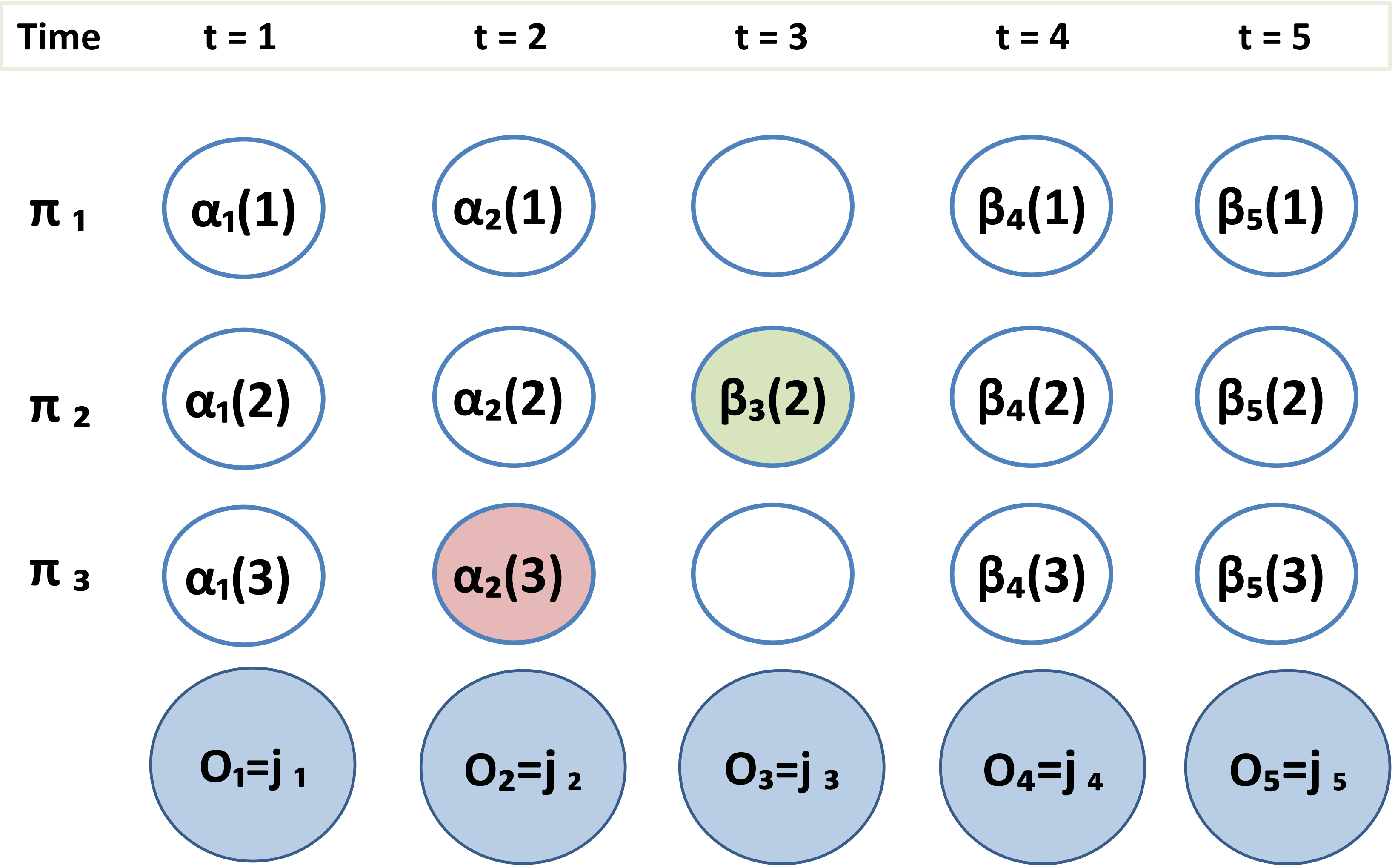
2) **Gamma Function** (Marginalizing out X_{t+1})

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i,j) \quad = p(X_t = i | O_{1:T}, \lambda)$$

<u>Interpretation</u>: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i)?

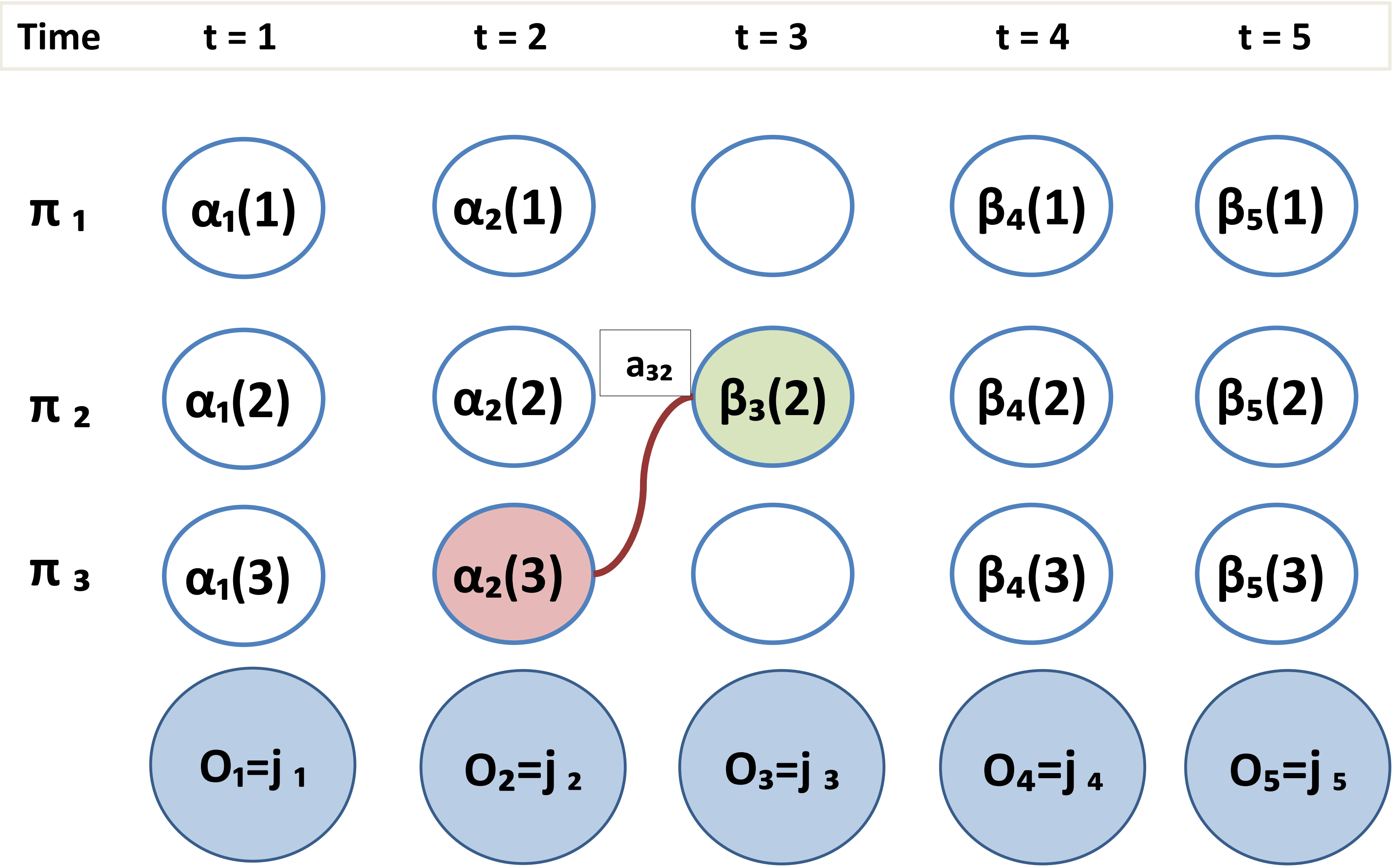


$$P(X_2 = 3 , X_3 = 2 \mid O_{1..5}) = ???$$



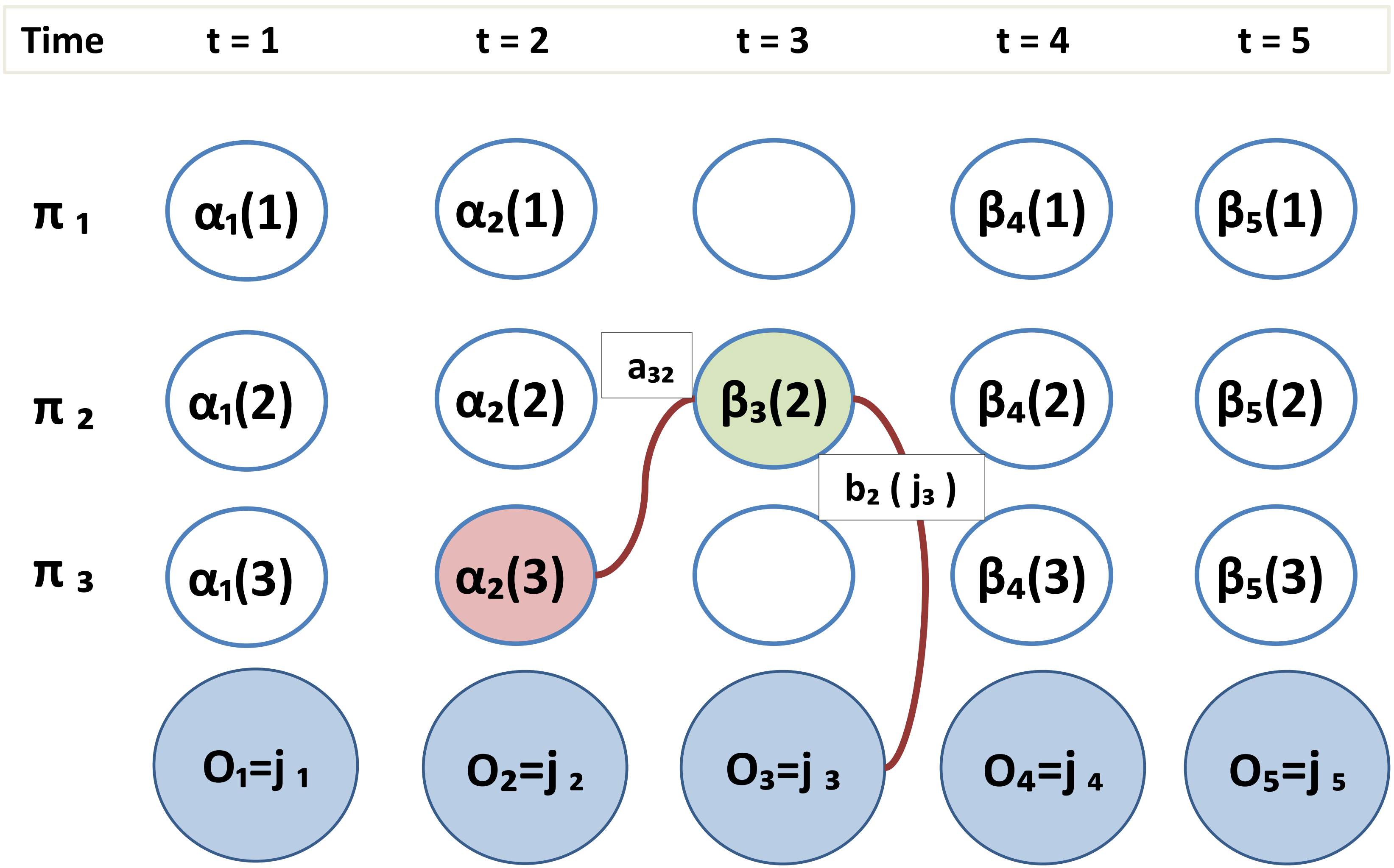
$P(X_2 = 3 , X_3 = 2 \mid O_{1..5}) =$

$P(O_{1..2}, X_2 = 3)\ldots$



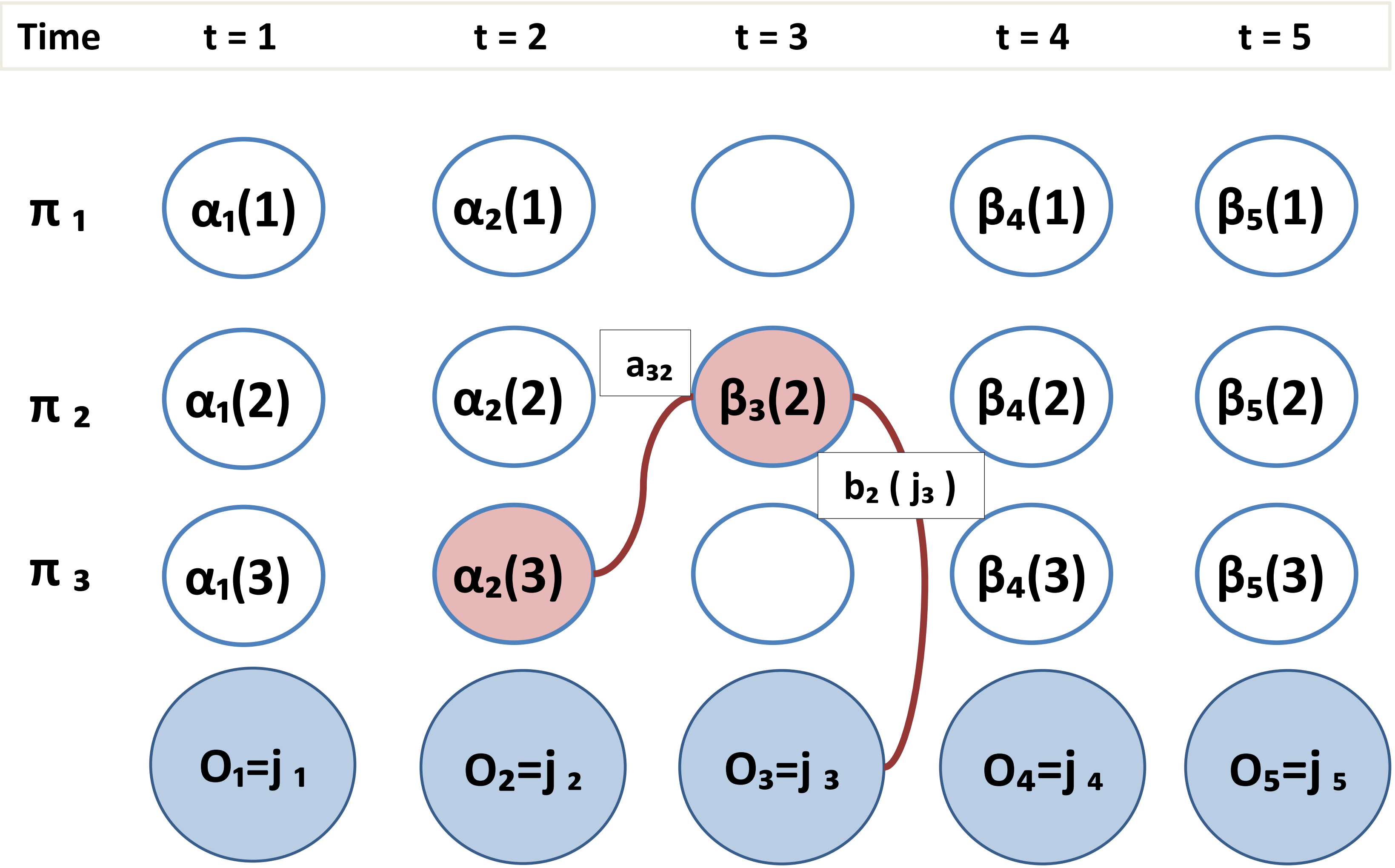
$P(X_2 = 3 , X_3 = 2 \mid O_{1..5}) =$

$P(O_{1..2}, X_2 = 3)\, P(X_3 = 2 \mid X_2 = 3)...$



$$P(X_2 = 3 , X_3 = 2 \mid O_{1..5}) =$$

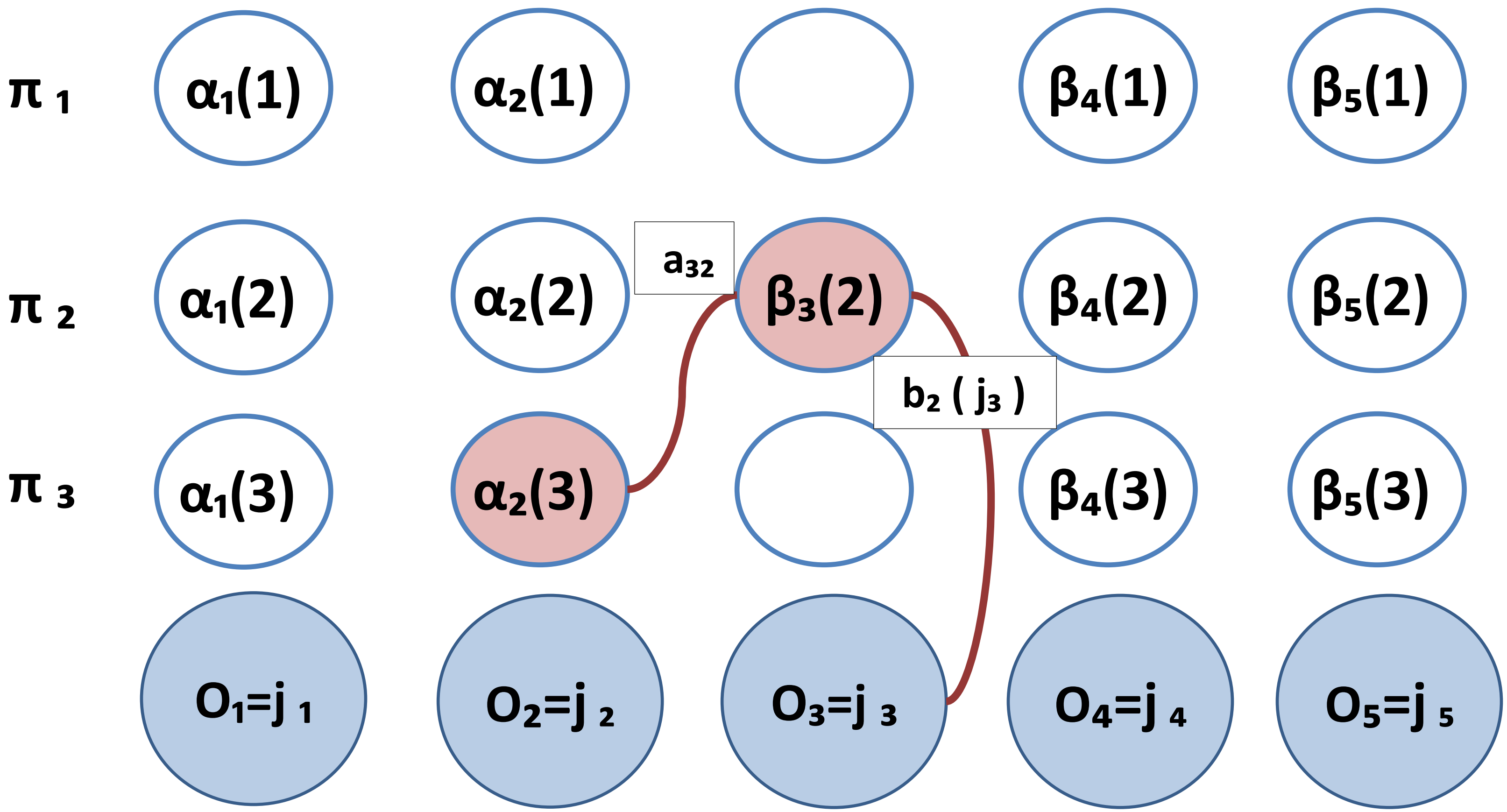
$$P(O_{1..2}, X_2 = 3)\, P(X_3 = 2 \mid X_2 = 3)\, P(O_2=j_3 \mid X_3 = 2)\ldots$$



$P(X_2 = 3 , X_3 = 2 \mid O_{1..5}) =$

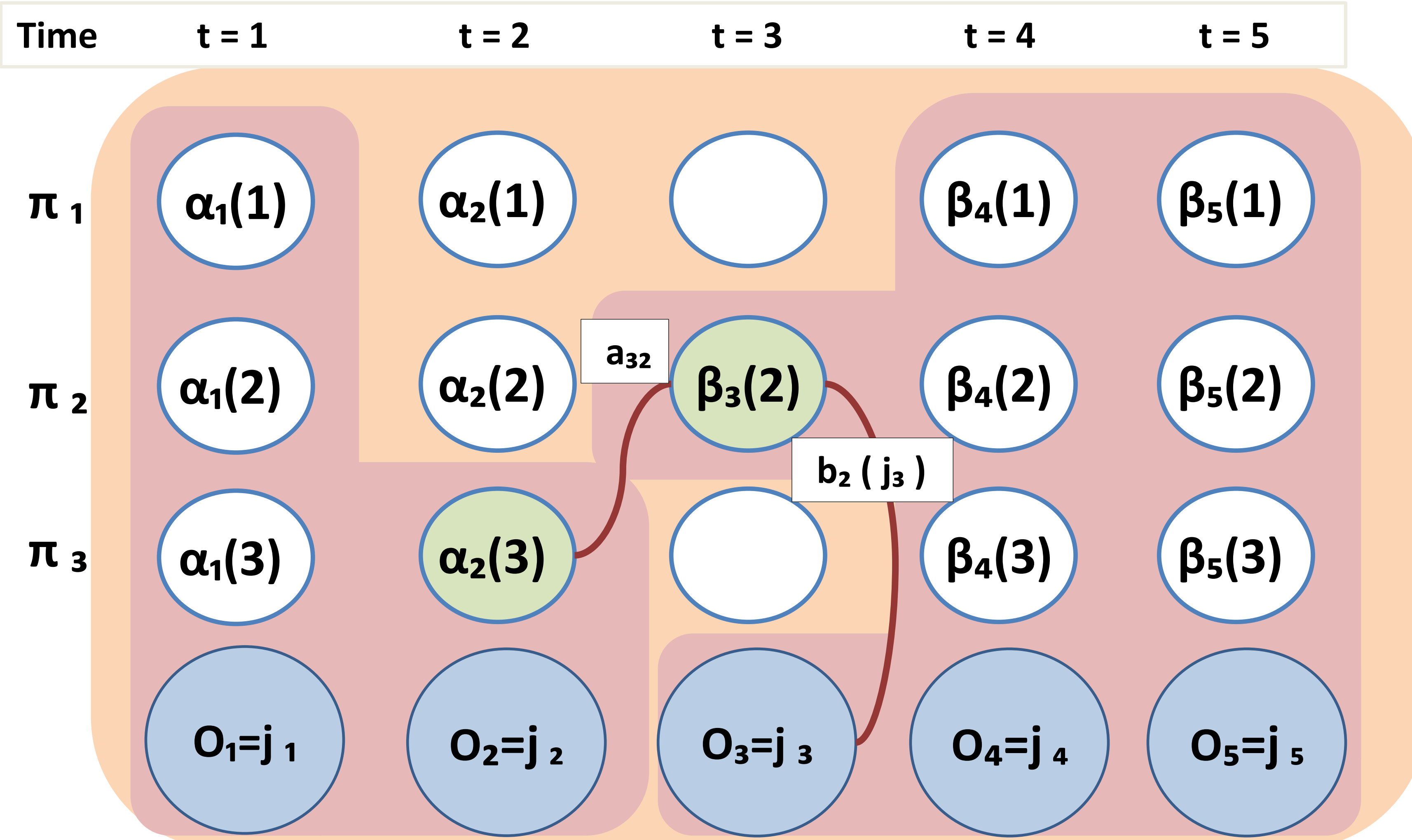
$P(O_{1..2}, X_2 = 3)\ P(X_3 = 2 \mid X_2 = 3)\ P(O_2=j_3 \mid X_3 = 2)\ P(O_{4..5} \mid X_3 = 2)\ldots$

Time	t = 1	t = 2	t = 3	t = 4	t = 5



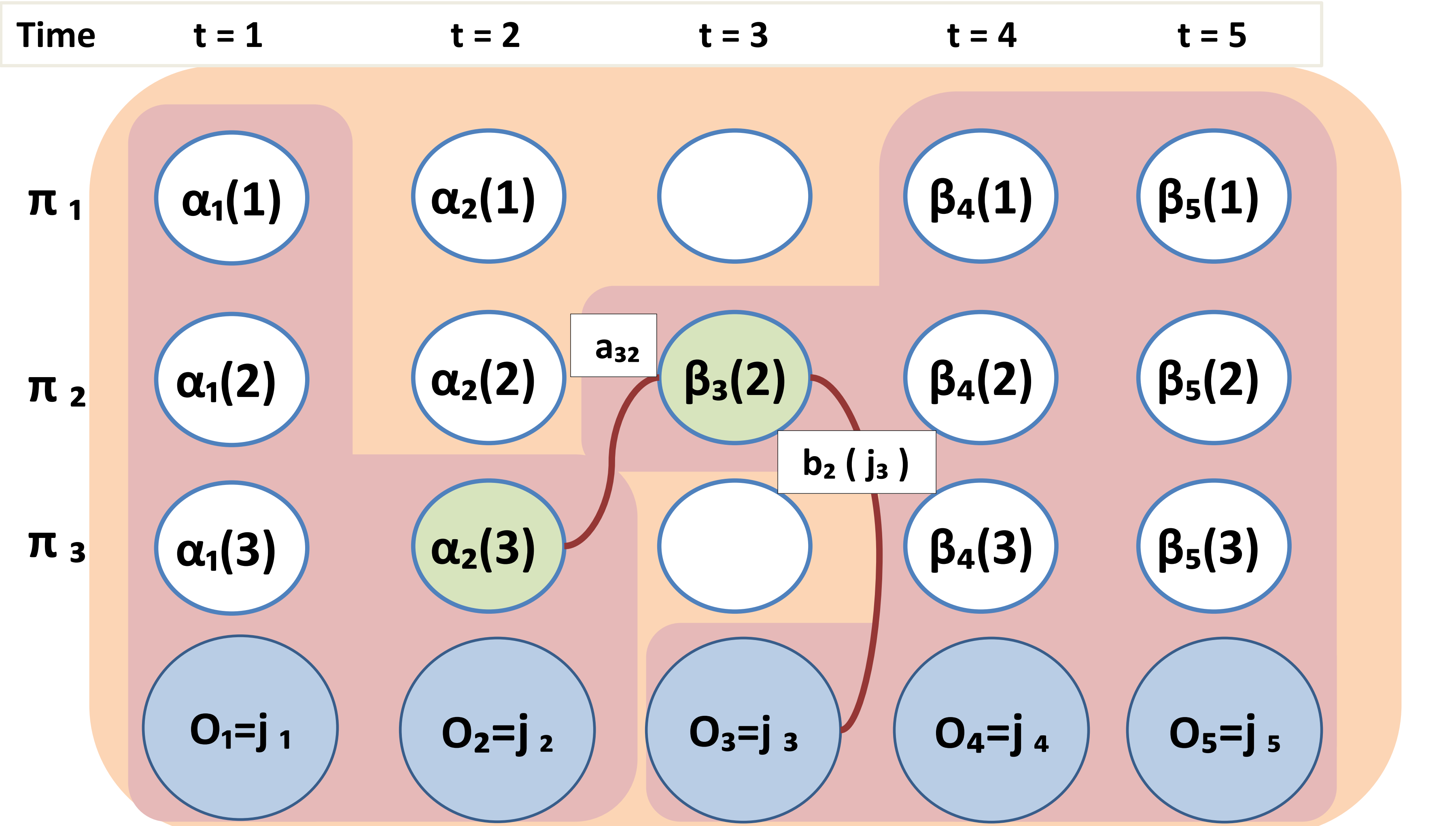
$P(X_2 = 3, X_3 = 2 \mid O_{1..5}) =$

$$\alpha_2(3)\, a_{32}\, b_2(j_3)\, \beta_3(2) \ldots$$



$$P(X_2 = 3, X_3 = 2 \mid O_{1..5}) = \frac{\alpha_2(3)\, a_{32}\, b_2(j_3)\, \beta_3(2)}{\sum_i \alpha_5(i)}$$

Normalization



$$\Upsilon_2(3,2) = \frac{\alpha_2(3)\, a_{32}\, b_2(j_3)\, \beta_3(2)}{\sum_i \alpha_5(i)}$$

GAMMA CALCULATIONS: (1)

1) **Di – Gamma Function**

$$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)} \quad = p(X_t = i, X_{t+1} = j | O_{1:T}, \lambda)$$

Interpretation: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i) && at time (t+1) the hidden state is (X_{t+1}=j)?

2) **Gamma Function** (Marginalizing out X_{t+1})

$$\gamma_t(i) = \sum_{j=1}^{N} \gamma_t(i,j) \quad = p(X_t = i | O_{1:T}, \lambda)$$

Interpretation: Given the entire observation sequence and current estimate of the HMM, what is the probability that at time (t) the hidden state is (X_t=i)?

GAMMA CALCULATIONS: (2)

A) Transition estimates

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \ i,j = 1,\ldots,N \quad = \frac{\text{E \{\# of transitions from state (i) to state (j)\}}}{\text{E \{\# of transitions from state (i) to state (don't care)\}}}$$

B) Emission estimates

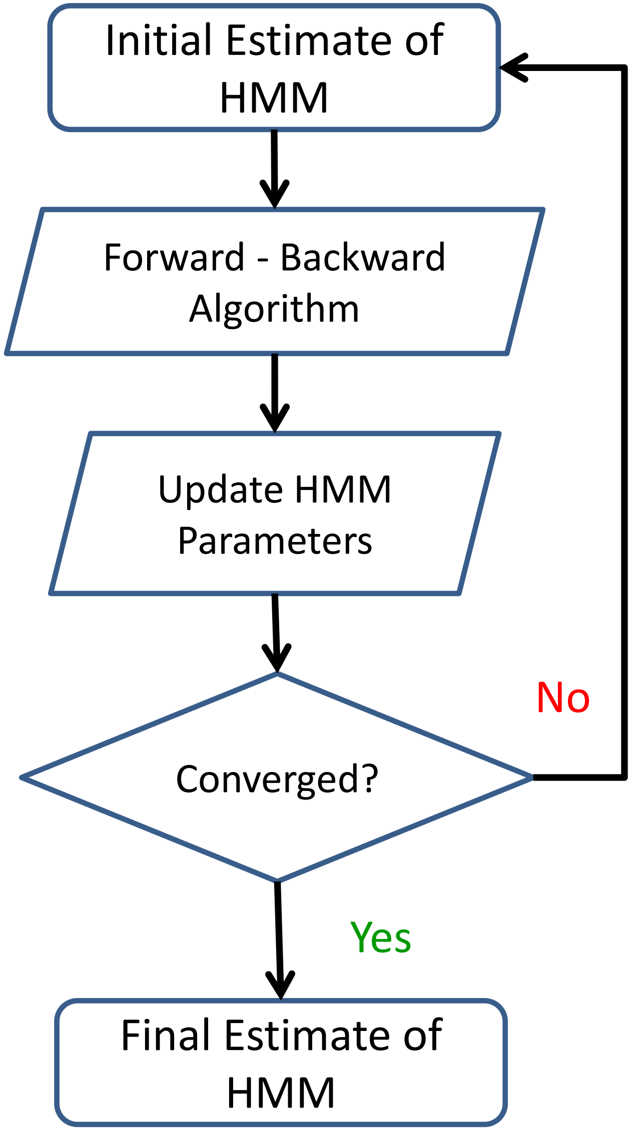
$$b_j(k) = \frac{\sum_{\substack{t=1,2,\ldots,T-1 \\ O_t = k}} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \forall \quad j = 1,\ldots,N, \quad k = 1,\ldots,K$$

$$= \frac{\text{E \{\# of emission (k) from state (i)\}}}{\text{E \{\# of transitions from state (i) to state (don't care)\}}}$$

q) Initial state probabilities

$$\pi_i = \gamma_1(i) \quad \forall \quad i = 1,\ldots,N$$

BAUM-WELCH ALGORITHM:



- Given an observation sequence $O_{1:T}$, the number of states, N, and the number of observation outcomes, M.

1. Initialize $\lambda=(A,B,\pi)$
2. Compute $\alpha_t(i)$, $\beta_t(k)$, $\gamma_t(i,j)$ and $\gamma_t(i)$
3. Re-estimate the model $\lambda=(A,B,\pi)$
4. Repeat from 2 until $p(O|\lambda)$ levels out

PROBLEM 3: Learning (Puppy Platone Example)

Given:

- **O** = {**p, e, e, l**}
- Guesses for model :

q_e =

A	B	H	S
0.25	0.25	0.25	0.25

A_e =

X_t | X_{t+1}	A	B	H	S
A	0.3	0.3	0.2	0.2
B	0.1	0.4	0.1	0.4
H	0.5	0.2	0.1	0.2
S	0.2	0.1	0.1	0.6

B_e =

X_t | O_t	p	e	b	l
A	0.5	0.1	0.2	0.2
B	0.2	0.5	0.1	0.2
H	0.1	0.1	0.6	0.2
S	0.1	0.1	0.3	0.5

Unknown:

- Hidden state sequence until time (t)

Find:

- Model parameters A_f ,B_f ,q_f

Solution:

- On the board … (Only Backward Algorithm)

PROBLEM 3: Learning (Puppy Platone Example)

Next Steps:

- Calculate Di – Gamma and Gammas at every time step = (T-1) x N x N calculations.
- This means that there is a "version of A and B" at every time step.
- We calculate an "average" over these versions to obtain an estimate of A and B for every iteration of Baum-Welch Algorithm.

GENERAL TIPS AND POINTERS:

1) Using natural logarithms and summations instead of direct products. (**Stamp-implementation tutorial)**

2) Forward algorithm: The $\alpha_t(i)$ s can be normalized across hidden states at every time instant for numerical stability. Algorithm is modified slightly!

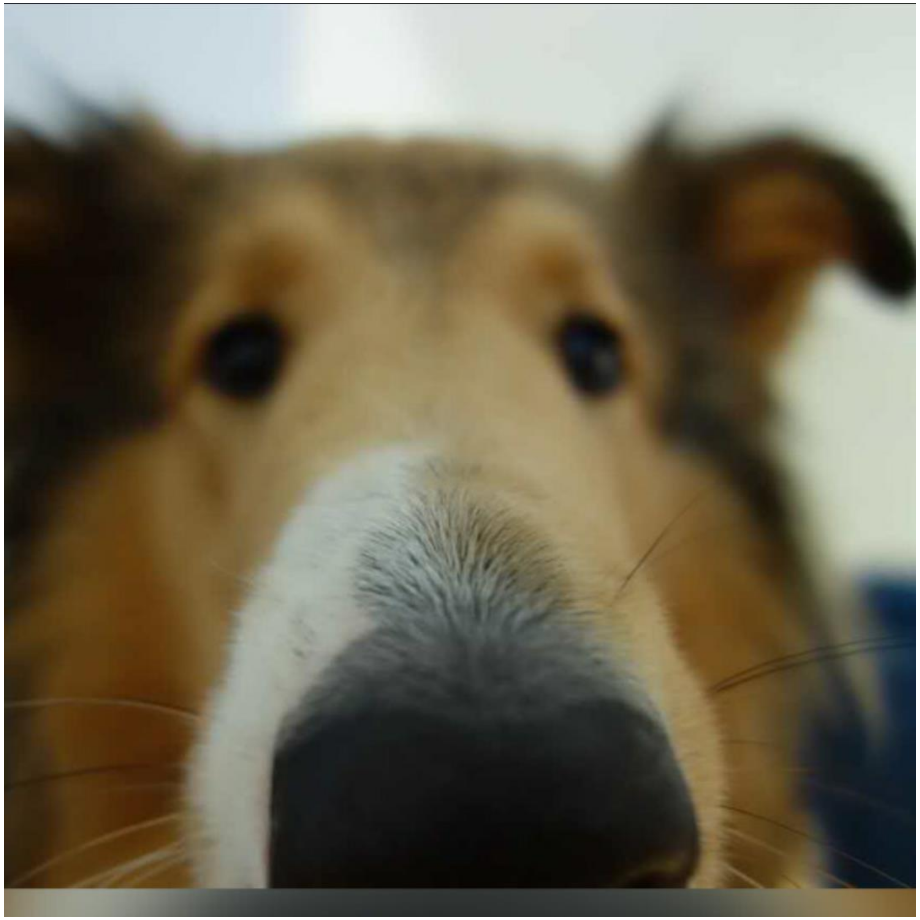
3) Type of matrix initializations are important for convergence using Baum-Welch Algorithm. Types = Flat / Random Row-Stochastic / Suggestive

4) ASSIGNMENT THINK → (Optional – NOT FOR PASSING GRADE!)

   - Why do you need to compare matrices in HW2?
   - Why is this not a trivial problem?
   - How will you solve it?

PROGRAMMING HINTS & SUGGESTIONS:

1. How will you define a 2D – Array?
2. KISS → Implement matrix multiplication
3. Use *doubles* instead of *floats*
4. Do not use any fancy libraries as external files – Kattis will not accept it !
5. For implementations **USE** the Stamp Tutorial (Check webpage). There are variations with our slides.
6. **Optional implementation exercises on Kattis – USE THEM!!**



QUESTIONS?