Hidden Markov Models

Slides by Margrit Betke, Yiwen Gu

Reading: Rabiner, Proceedings of the IEEE, 77(2), 1989, up to page 275

Agenda (10/29, 10/31)

- From State Machine To Markov Model
- Working with Hidden Markov Models

Finite State Machine (FSM)

= Automata

Markov Network
= FSM with Transition Probabilities

Finite State Machine with
Deterministic Outputs

Hidden Markov Model
= Markov Network with Output Probabilities

Finite State Machine (FSM)

= Automata

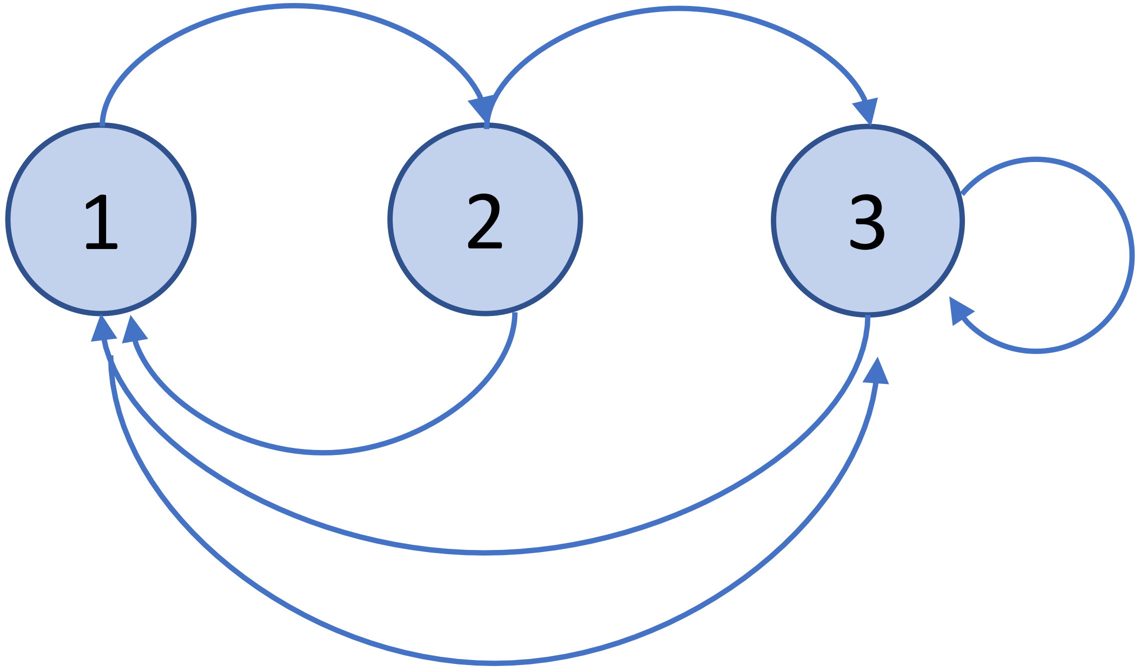
Markov Network
= FSM with Transition Probabilities

Finite State Machine with
Deterministic Outputs

Hidden Markov Model
= Markov Network with Output Probabilities

Finite State Machine (FSM) = Automata

- A state machine is a machine's AI logic in graph form.
- Key Concepts:
  - States/nodes/vertices
  - Transitions/edges/arcs



States = N = 3,

Set of states Q = $\{q_1,\ldots, q_N\}$

Finite State Machine (FSM)

= Automata

Markov Network

= FSM with Transition Probabilities

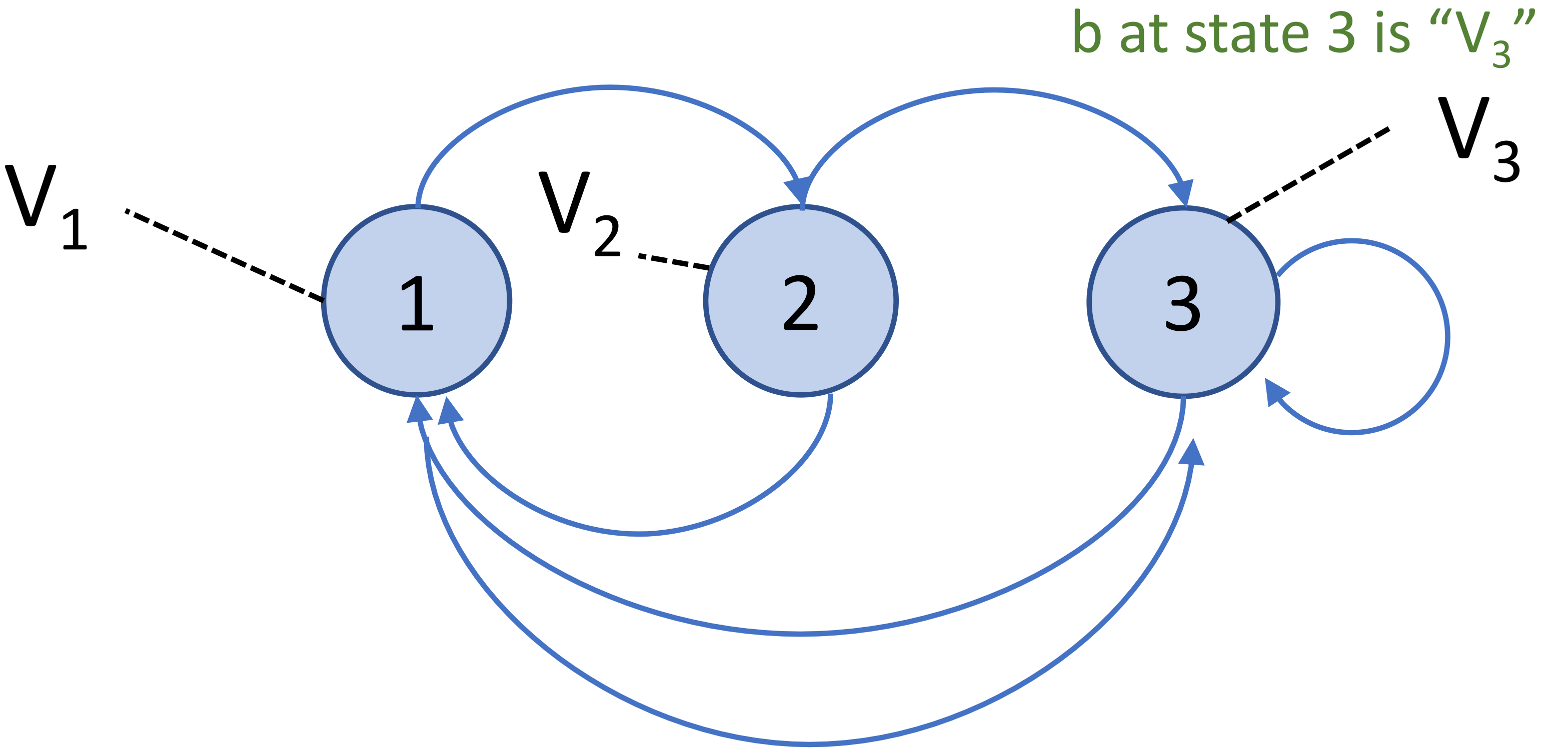
Finite State Machine with Deterministic Outputs

Hidden Markov Model

= Markov Network with Output Probabilities

FSM with Deterministic Outputs (b's)

- Each state outputs a **symbol** deterministically
  i.e., for every state and input combination, there is a specific, predefined output).
- The symbols in an FSM form a **vocabulary**.
- The set of all possible vocabularies (sequences of the symbols) is called **language**.
- Generator and/or Recognizer



States = N = 3, Set of states Q = $\{q_1, \ldots, q_N\}$

Outputs = M = 3 , $\{V_1, V_2, V_3\}$

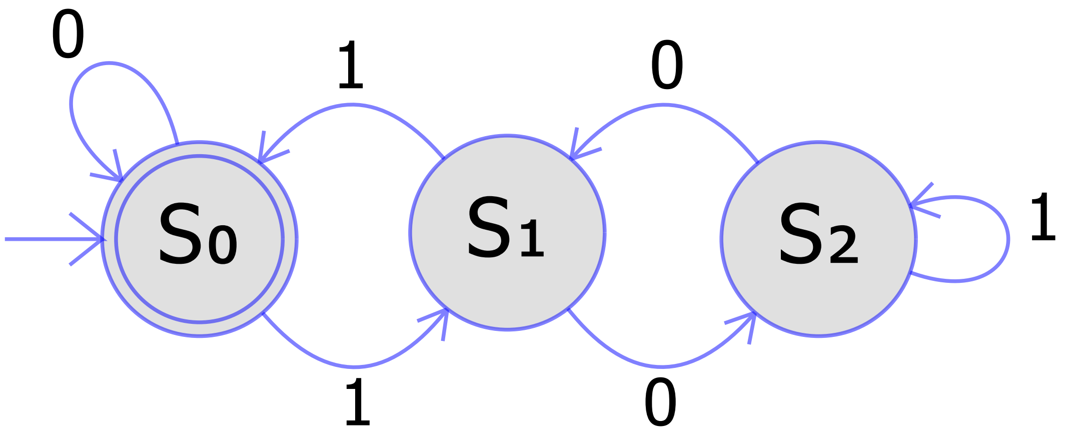
Example: Deterministic FSM as a Recognizer

Formal Definition:

A *Deterministic Finite Automaton(Acceptor)* (DF**A**) is a tuple M=⟨Q,Σ,δ,q0,F⟩

- Q - a finite set of **states**
- Σ - a finite set of input symbol **alphabet**
- δ - a transition **function**
- q0∈Q - the **start state**
- F⊆Q - the **final (or "accepting") states**

Q: What is this doing?



Source : Wiki

Finite State Machine (FSM)

= Automata

Markov Network

= FSM with Transition Probabilities

Finite State Machine with Deterministic Outputs

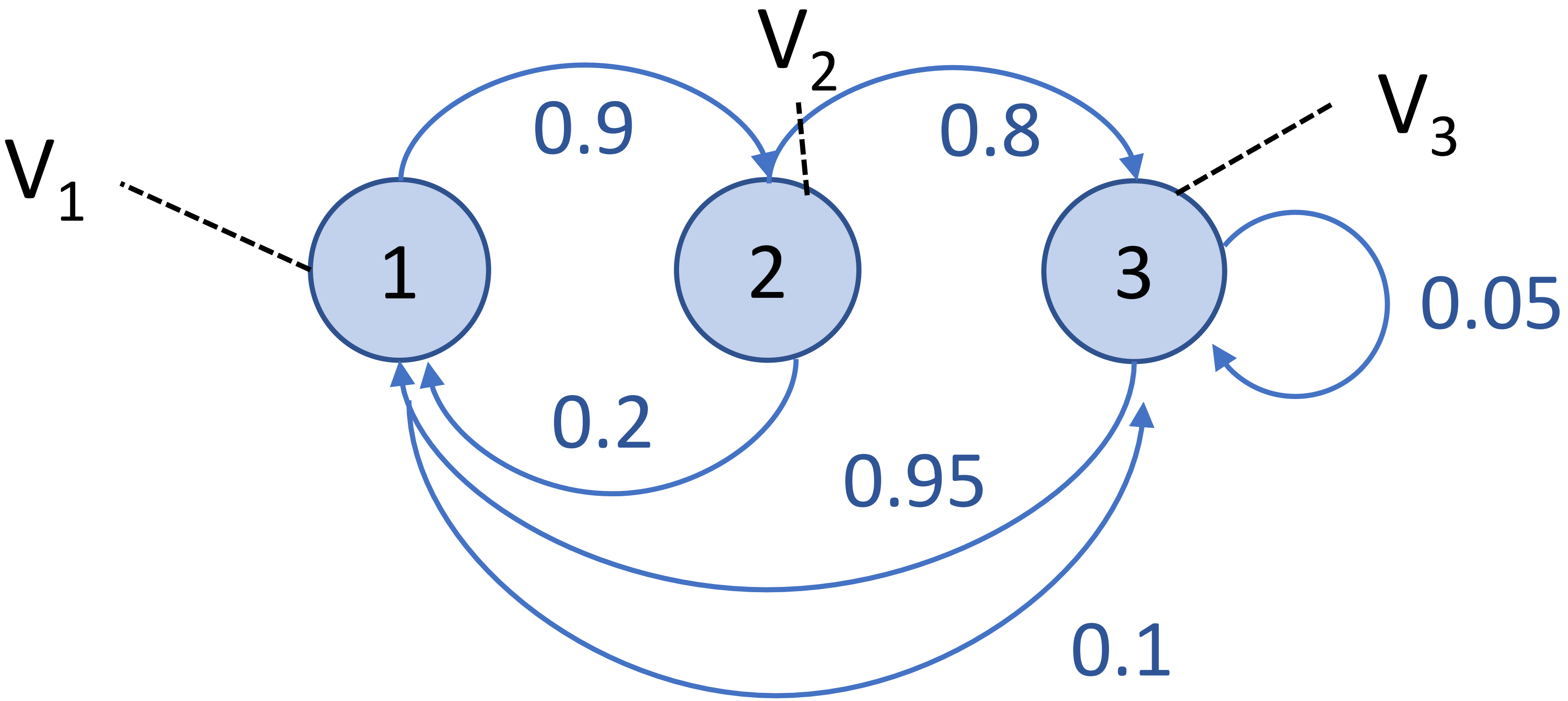
Hidden Markov Model

= Markov Network with Output Probabilities

Markov Network (Chains)

= FSM with Transition Probabilities (a's)

States correspond to physical events, observations, measurements



Markov property: the future state of a system depends only on its current state, not on the sequence of events that preceded it

Probability Axioms (Kolmogorov Axioms)

1. Probabilities are non-negative reals:

   $P(A) >= 0$ for all events A

2. The entire event space has probability 1:

   $\Sigma \ P(A) = 1$ (sum over all events A)

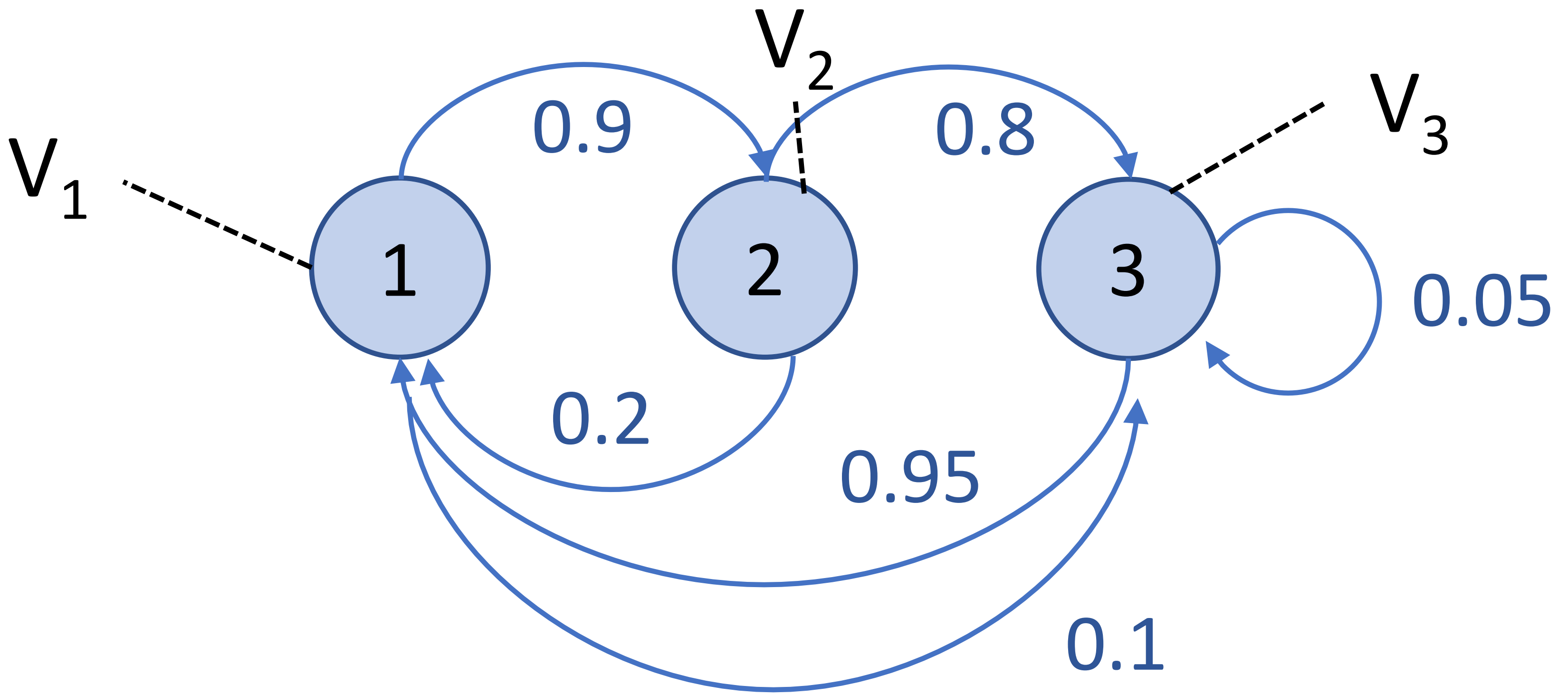
3. Probabilities add for pairwise disjoint events:

   A & B disjoint events: $P(A \cup B) = P(A) + P(B)$

Markov Chains

= FSM with Transition Probabilities (a's)

States correspond to physical events, observations, measurements



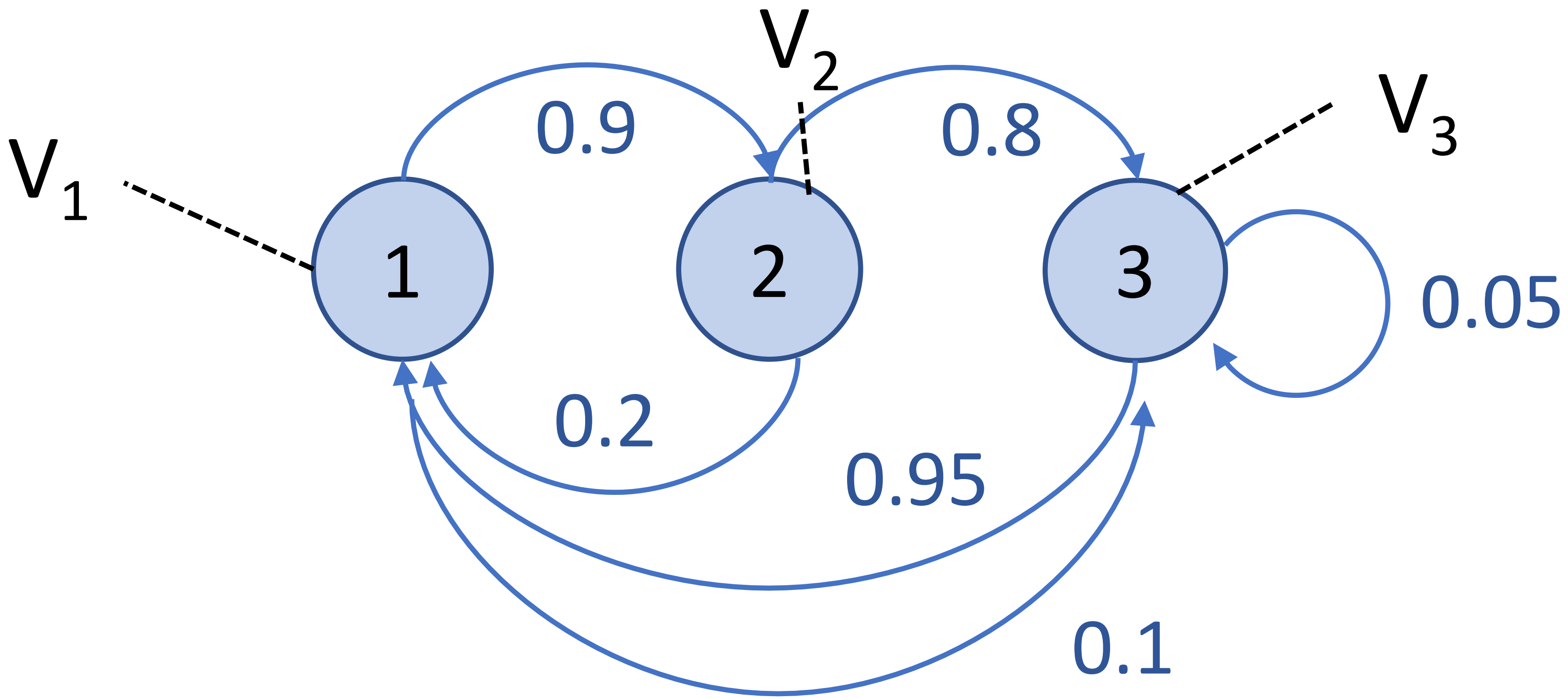
$a_{12} = 0.9$

$$\sum_{j=1}^{\#states} \text{Prob(state i -> state j)} = \sum_{j=1}^{\#states} a_{ij} = 1 \qquad \text{for all states i}$$

Markov Chains

= FSM with Transition Probabilities (a's)

States correspond to physical events, observations, measurements

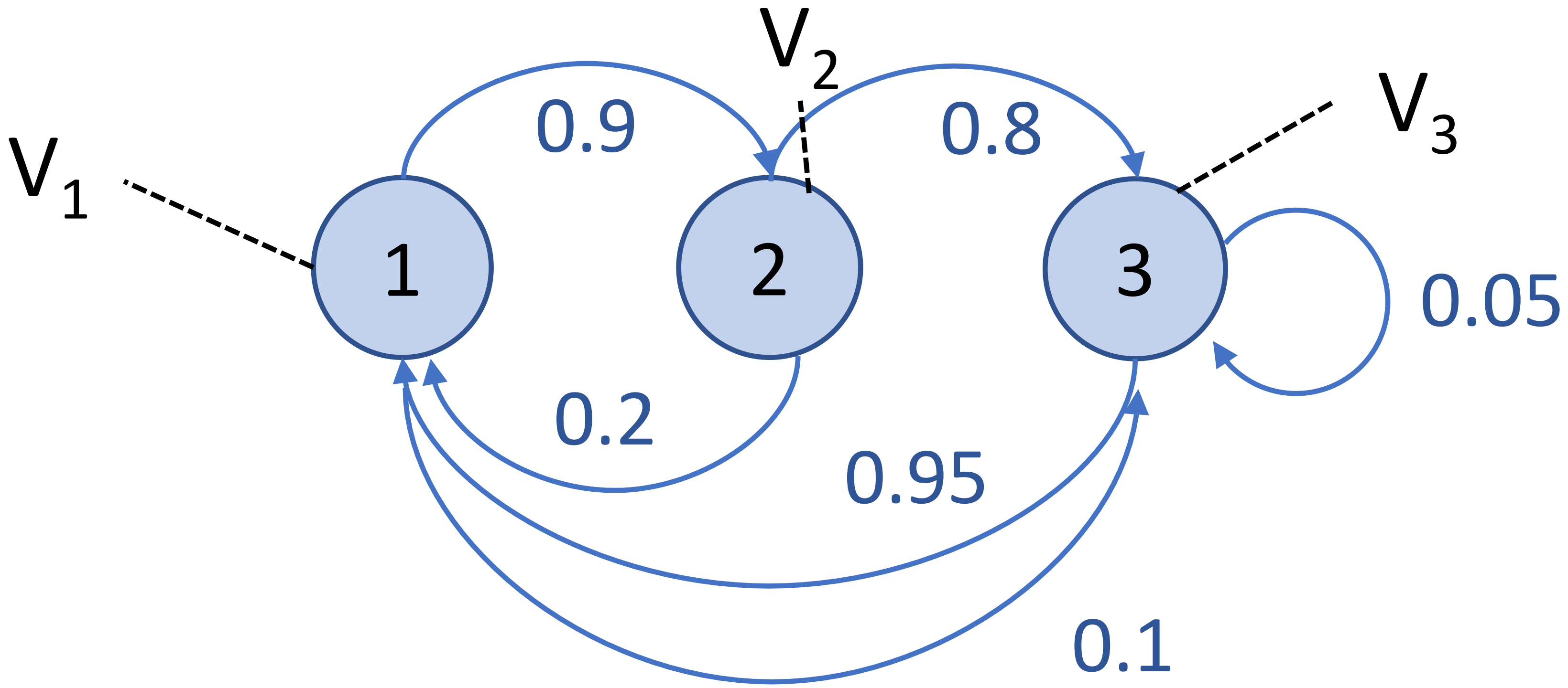


$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

$\pi_i =$ Probability that state i is the initial state

$\pi_1 + \pi_2 + \pi_3 = 1$

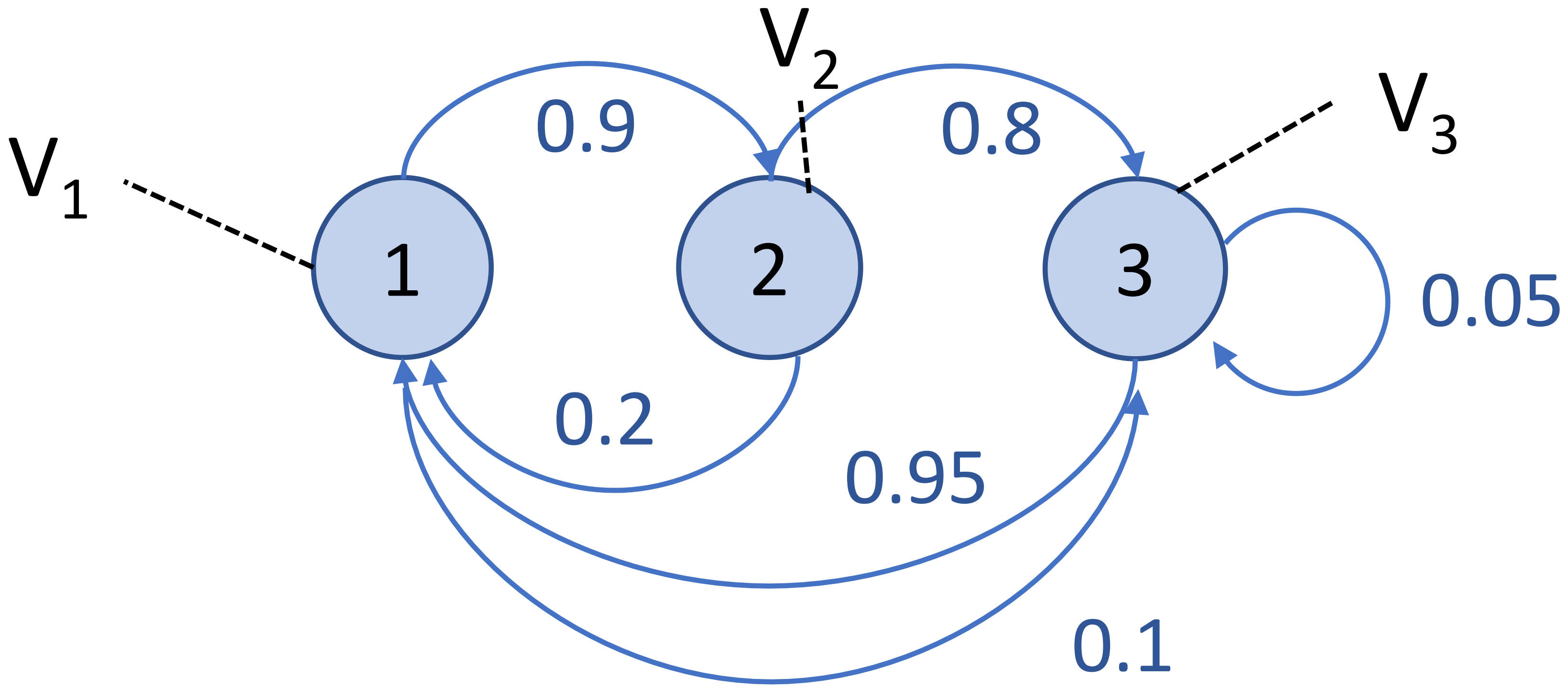
Markov Chains



Given a sequence of outputs, we can calculate the probability of observing it.

E.g, outputs = $V_2 V_1 V_3$

Markov Chains



Probability of observing a sequence of outputs $V_2V_1V_3$ =

Prob(O) = Prob($V_2V_1V_3$) = $\pi_2 \cdot$ Prob(2 -> 1) $\cdot$ Prob(1->3)

0.2 0.2 0.1 = 0.004

Finite State Machine (FSM)

= Automata

Markov Network
= FSM with Transition Probabilities

Finite State Machine with
Deterministic Outputs

Hidden Markov Model
= Markov Network with Output Probabilities

states not directly observable (= hidden)

Hidden Markov Model

Markov model: each state outputs and must output one symbol, making the state outputs deterministic (**observable**).

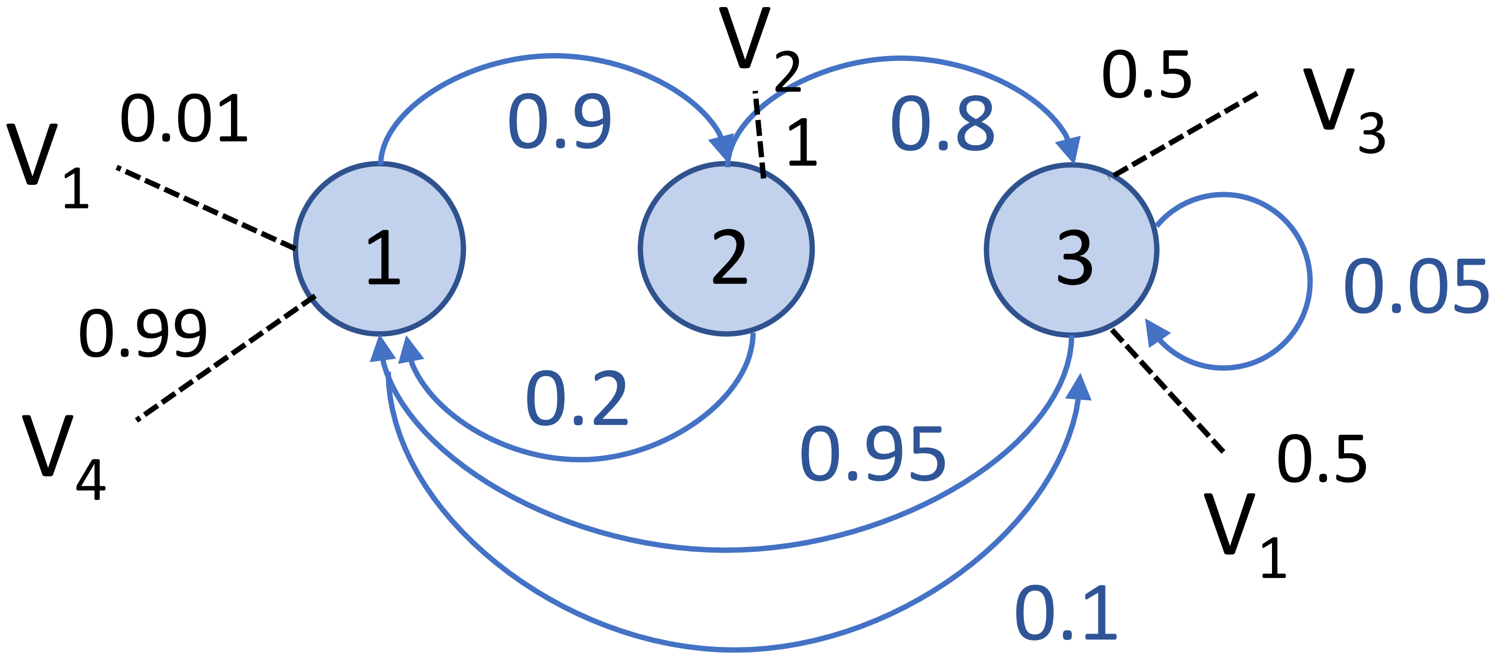
However, if instead, each state can output different symbols where each symbol is associated with a certain probability, then the outputs are non-deterministic (**hidden**) and the resulting model is called **Hidden Markov Model** (**HMM**).

At a high level, a HMM is a Markov model with Markov transition process and non-observable (hidden) states.

HMM Notations

- A set of N **states**: $\mathbf{S} = \{S_1, S_2, S_3, \ldots, S_N\}$
  - denote q_t : the actual state at time t
- A set of M distinct symbols as **vocabulary** : $\mathbf{V} = \{V_1, V_2, V_3, \ldots, V_M\}$
  - sometimes symbols are in lower case
- Transition probabilities as a **matrix**: $A = \{a_{ij}\}$
  - $a_{ij} = \text{Prob}(q_t = S_j \mid q_{t-1} = S_i), \ 1 \le i, j \le N$
- Initial probabilities: $\boldsymbol{\pi} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3, \ldots, \boldsymbol{\pi}_N\}$
  - $\boldsymbol{\pi}_i = \text{Prob}(q_1 = S_i), \ 1 \le i \le N$
- A **matrix** for observation likelihoods (aka. Emission probabilities): $B = \{b_j(k)\}$
  - $b_j(k) = \text{Prob}(V_k \text{ at } t \mid q_t = S_j), \ 1 \le j \le N, 1 \le k \le M$
  - The probability of V_k being generated from a state q_j
- A sequence of T observations (observed symbols at time T): $\mathbf{O} = O_1O_2O_3\ldots O_T$
  - Each O_i is drawn from the vocabulary **V**
- HMM: $\boldsymbol{\lambda} = (S, V, A, B, \boldsymbol{\pi})$ or more commonly just $(A, B, \boldsymbol{\pi})$

HMM Example



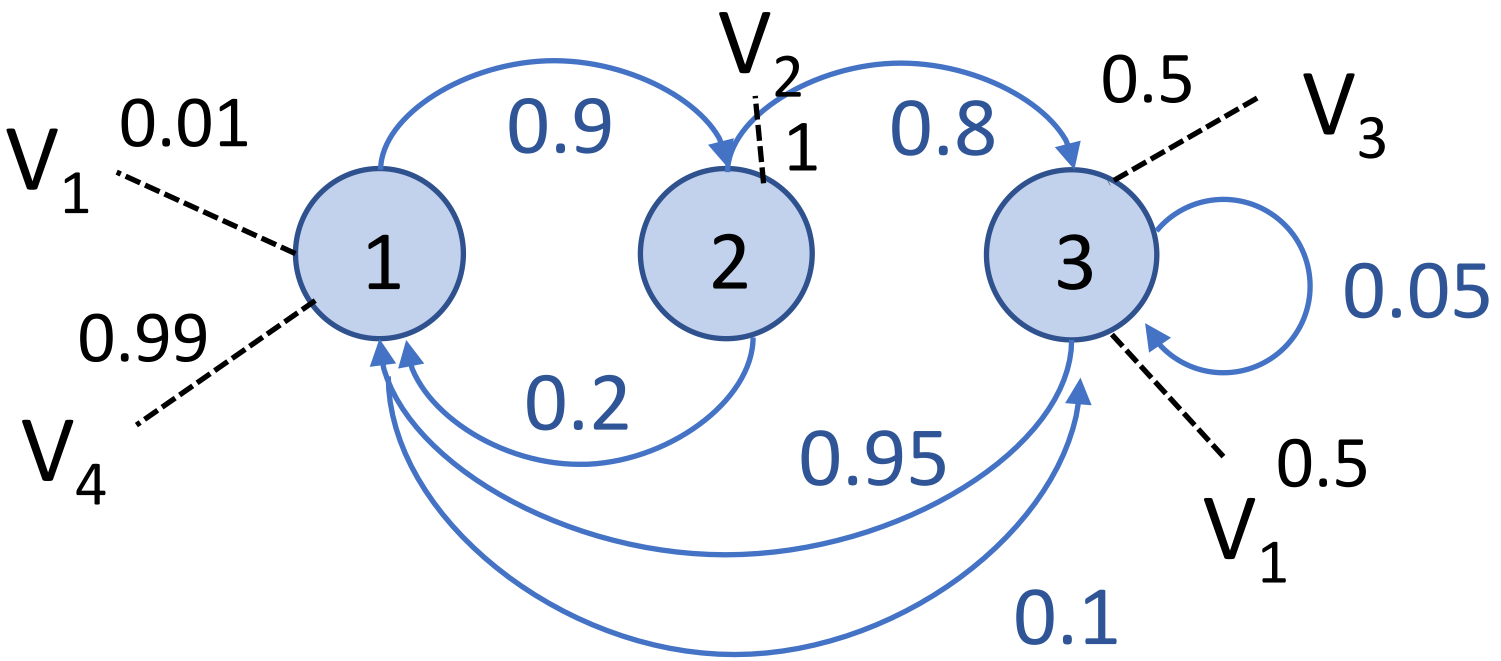
S = ?

V = ?

A = ?

B = ?

HMM Example – States and Symbols



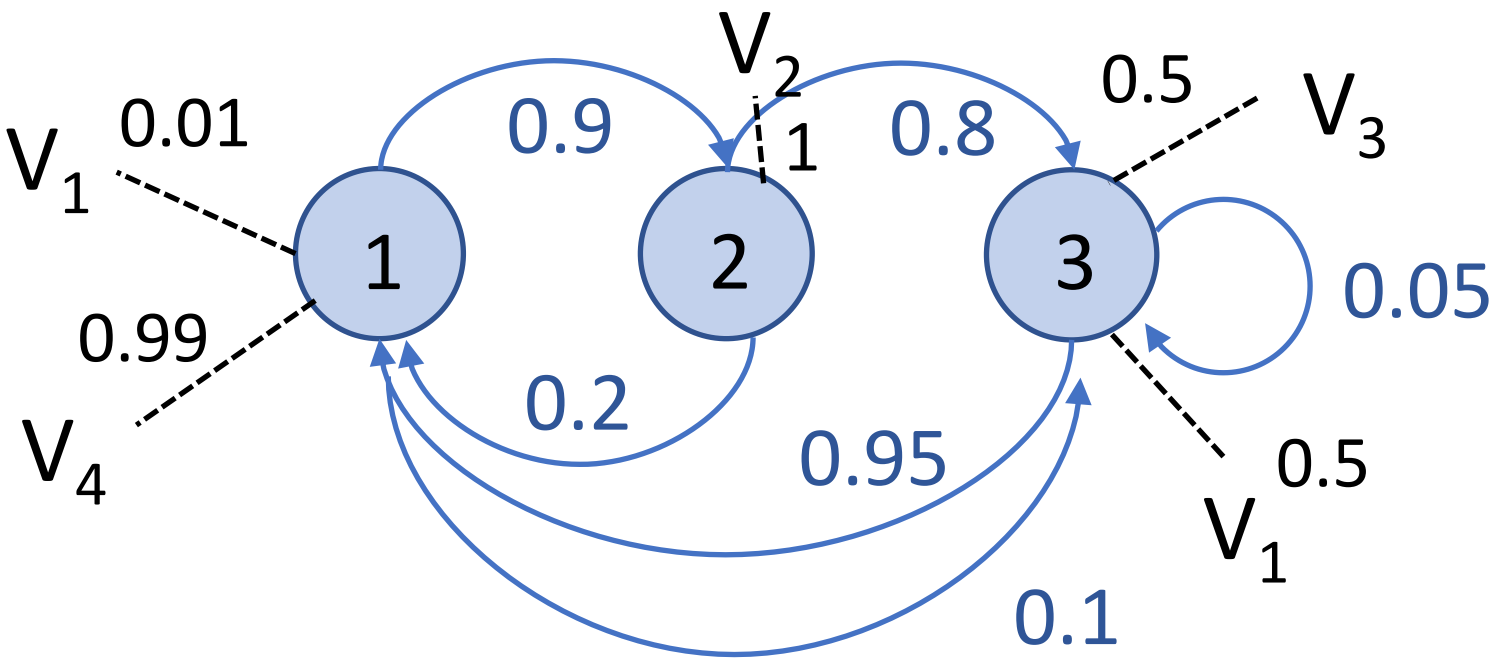
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

A = ?

B = ?

HMM Example – Transition Probability



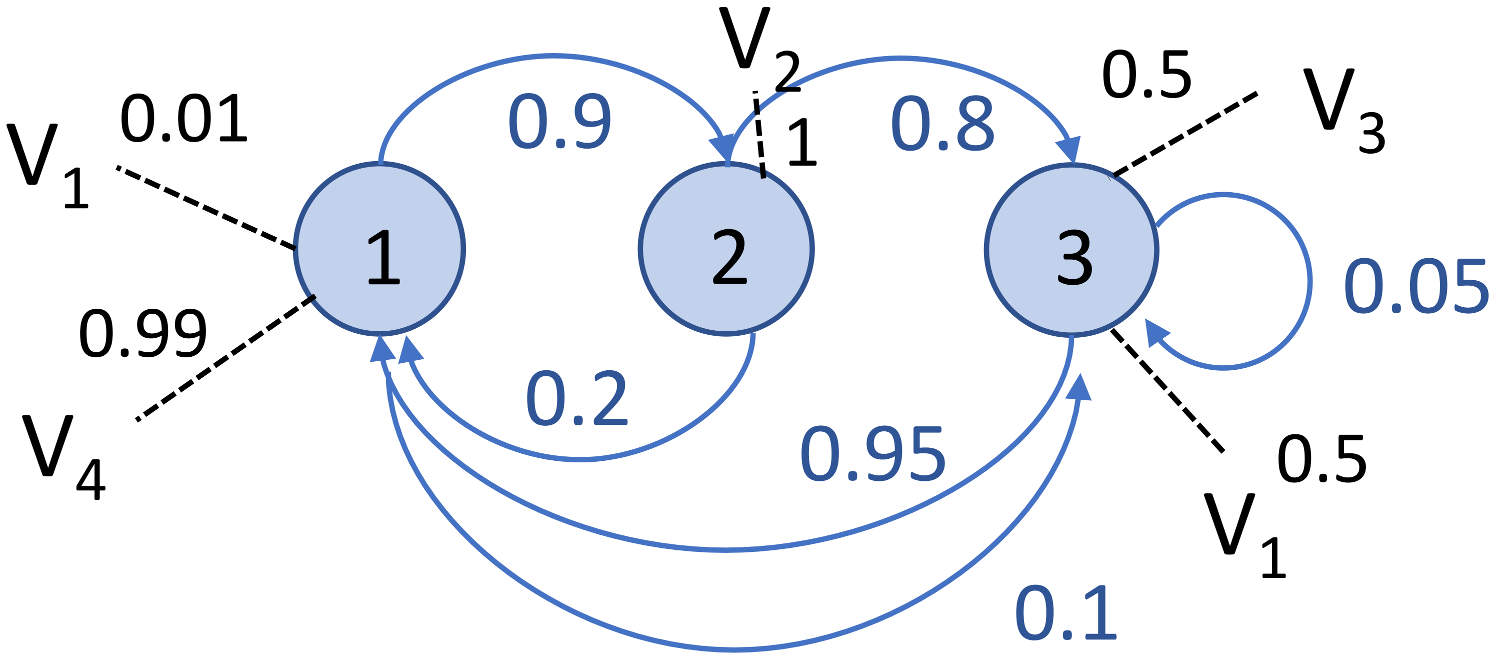
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\} = ?$

What is the size of A?

HMM Example – Transition Probability

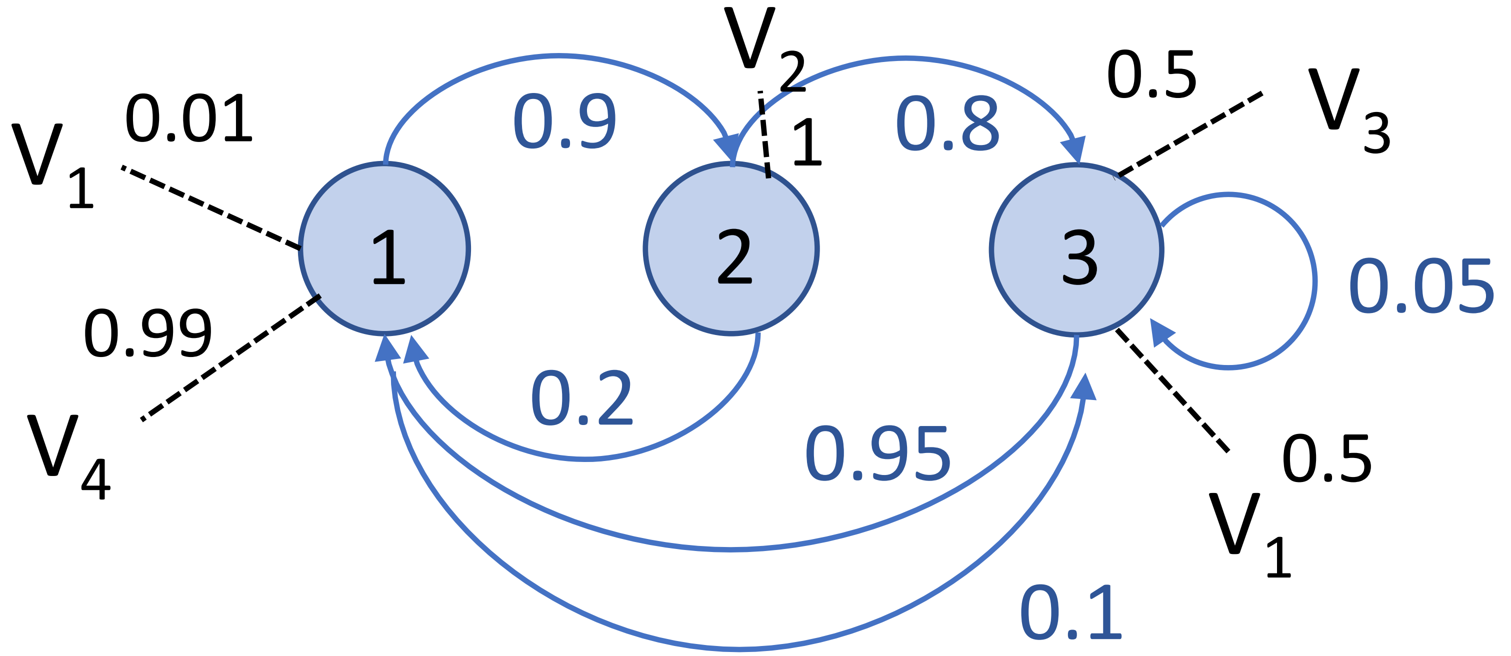


$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0, \ 0.9, \ 0.1], [0.2, \ 0, \ 0.8], [0.95, 0, \ 0.05]]$

HMM Example – Emission Probability



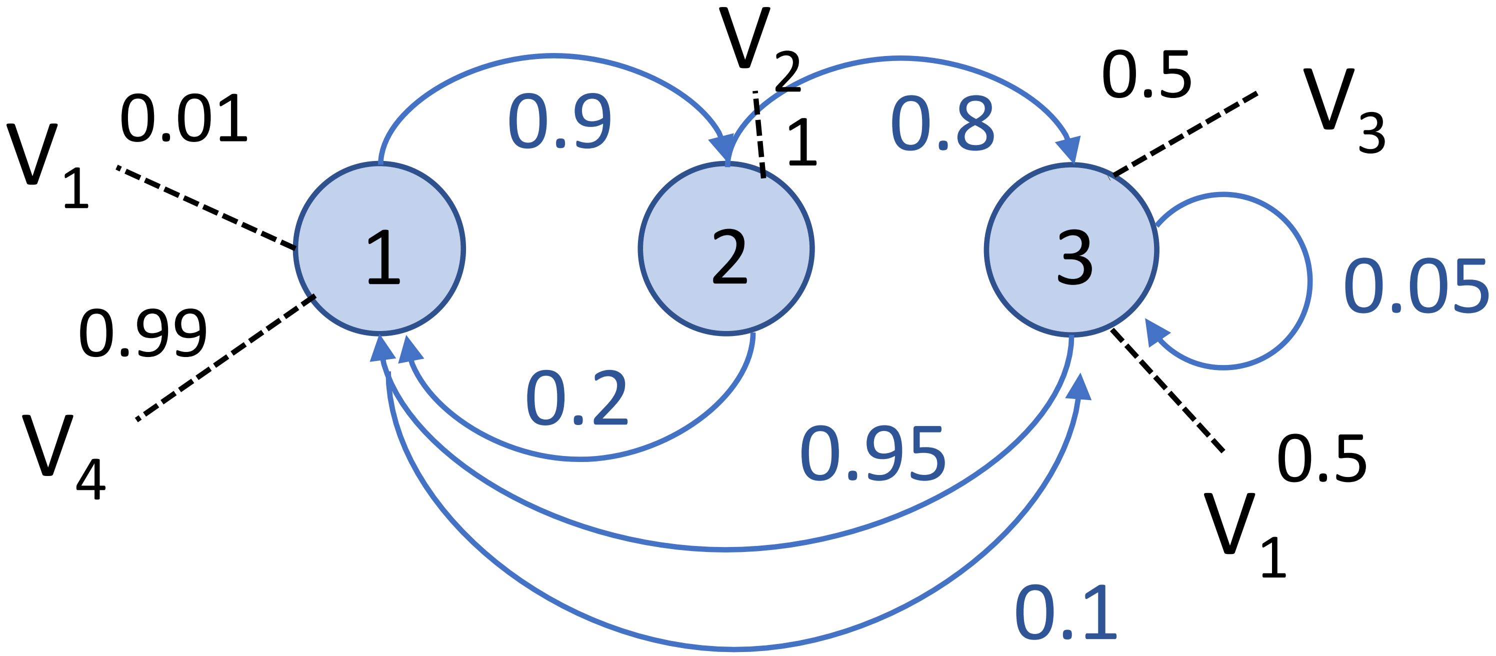
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0,\ 0.9,\ 0.1],$
$[0.2,\ 0,\ 0.8],$
$[0.95,\ 0,\ 0.05]]$

B = ?

HMM Example – Emission Probability



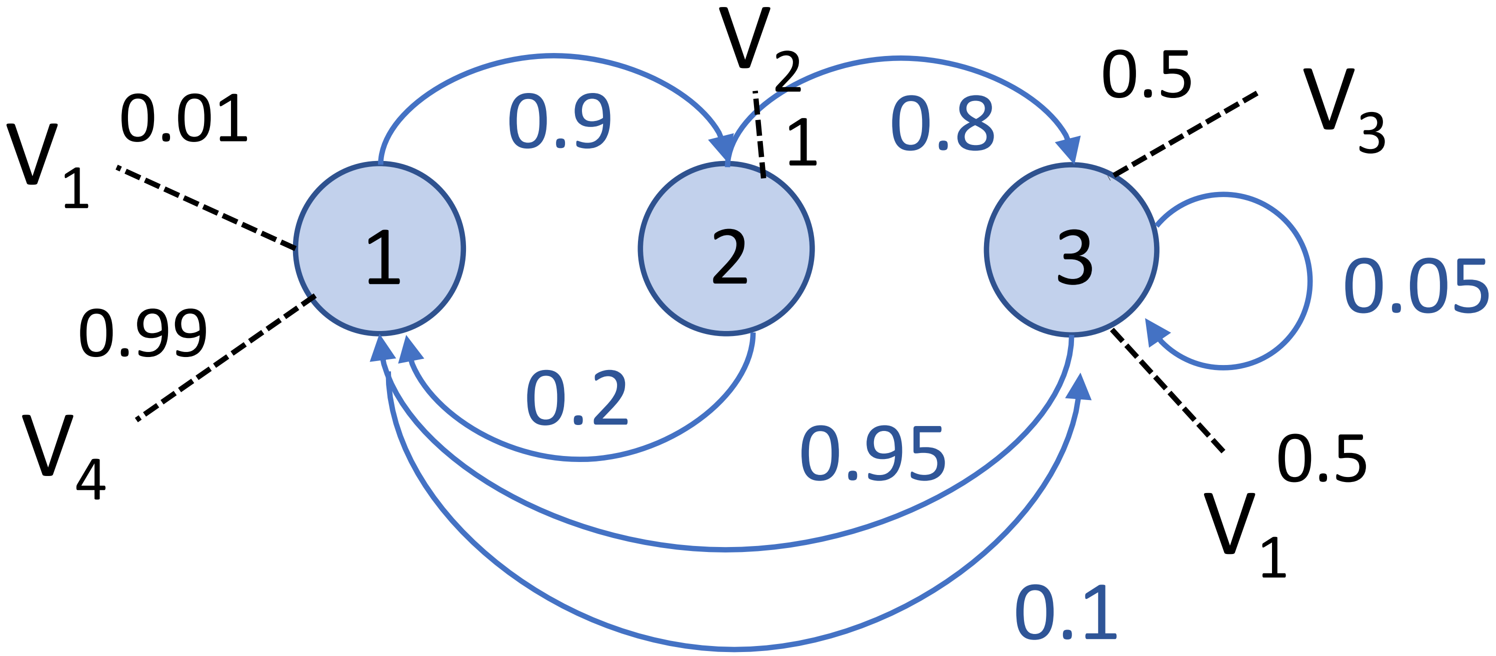
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0,\ 0.9,\ 0.1],$
$[0.2,\ 0,\ 0.8],$
$[0.95,\ 0,\ 0.05]]$

$B = [b_1 , b_2 , b_3] = ?$

HMM Example – Emission Probability



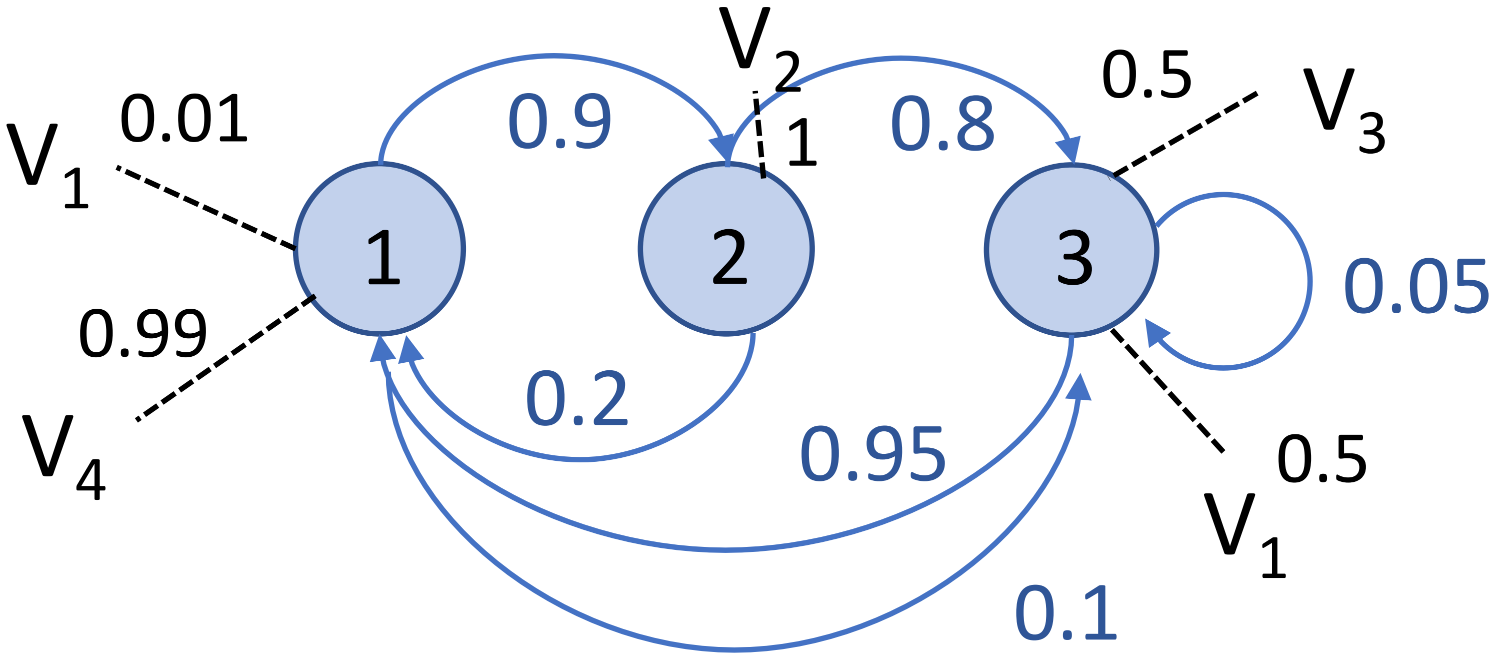
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0, \; 0.9, \; 0.1],$
$[0.2, \; 0, \; 0.8],$
$[0.95, 0, \; 0.05]]$

$B = [b_1 , b_2 , b_3]$
where
$b_i = [b_i(V_1) , b_i(V_2) , b_i(V_3), b_i(V_4)]$

HMM Example – Emission Probability



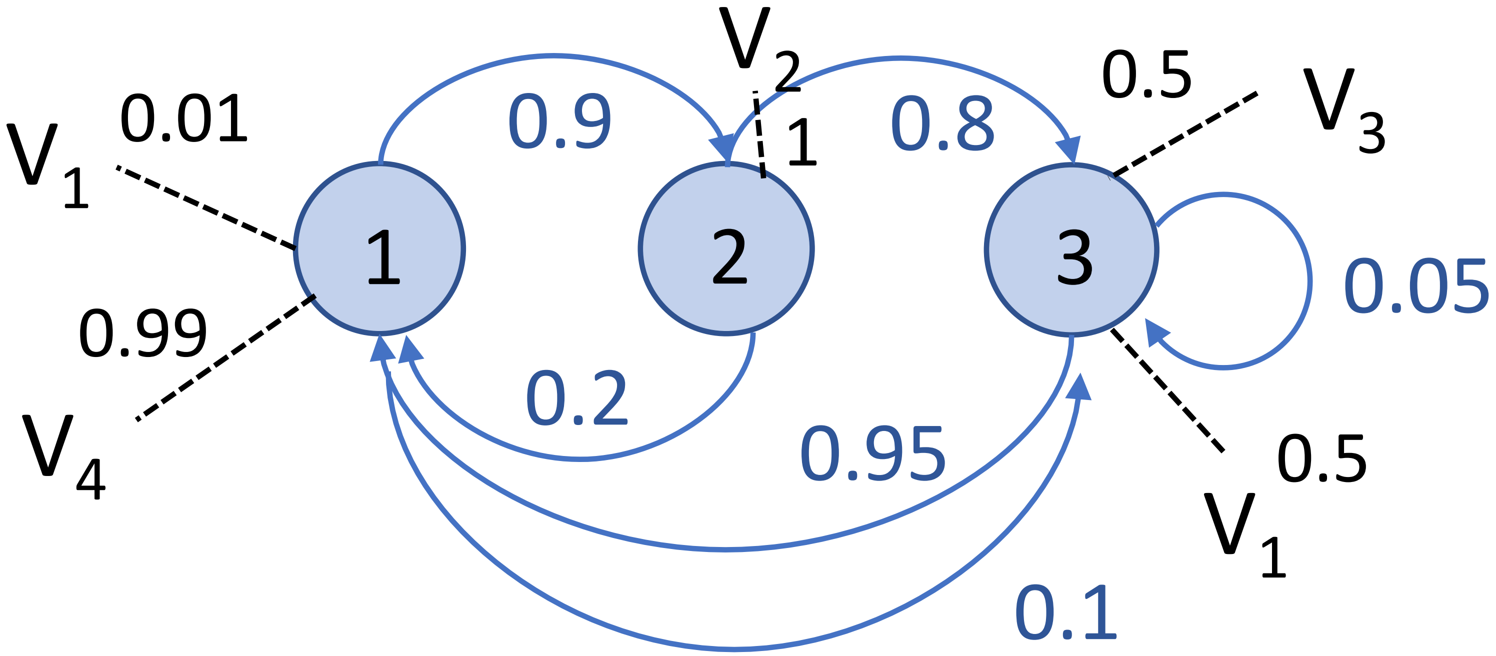
$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0,\ 0.9,\ 0.1],$
$[0.2,\ 0,\ 0.8],$
$[0.95,\ 0,\ 0.05]]$

$B = [b_1 , b_2 , b_3]$
where
$b_i = [b_i(V_1) , b_i(V_2) , b_i(V_3), b_i(V_4)]$
where
$b_i(V_1)$ = the prob of observing V_1 at state q_i

HMM Example – Emission Probability



$Q = [q_1, q_2, q_3]$

$V = [V_1, V_2, V_3, V_4]$

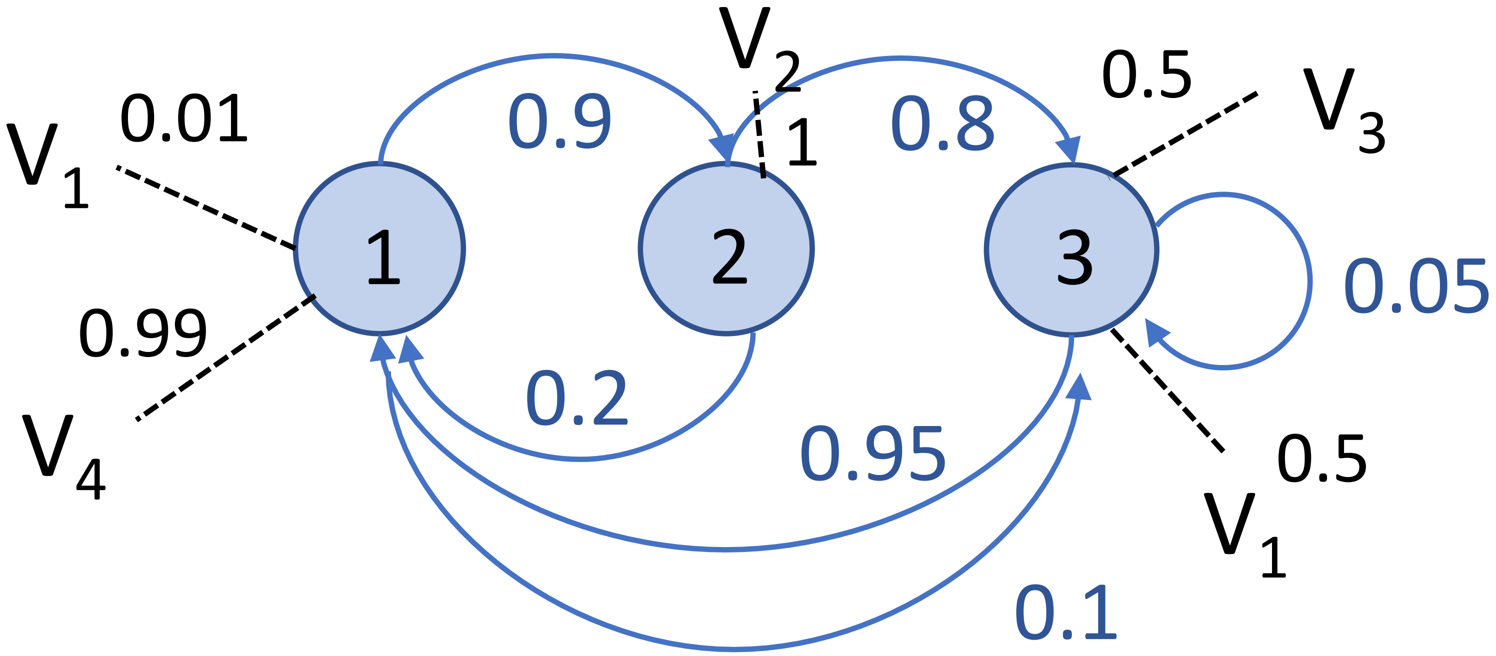
$A = \{a_{ij}\}$
$= [[0, \; 0.9, \; 0.1],$
$[0.2, \; 0, \; 0.8],$
$[0.95, 0, \; 0.05]]$

$B = [b_1 , b_2 , b_3]$
where
$b_i = [b_i(V_1) , b_i(V_2) , b_i(V_3), b_i(V_4)]$
where
$b_i(V_1)$ = the prob of observing V_1 at state q_i

$$\sum_{k=1}^{M} \text{Prob (Output } V_k \text{ in state i)} = \sum_{k=1}^{M} b_i (V_k) = 1$$

$b_1(V_1) = 0.01$, $b_1(V_2) = 0$, $b_1(V_3) = 0$, $b_1(V_4) = 0.99$
$b_2(V_1) = 0$, $b_2(V_2) = 1$, $b_2(V_3) = 0$, $b_2(V_4) = 0$
$b_3(V_1) = 0.5$, $b_3(V_2) = 0$, $b_3(V_3) = 0.5$, $b_3(V_4) = 0$

HMM Example – λ



$S = [S_1, S_2, S_3]$

$V = [V_1, V_2, V_3, V_4]$

$A = \{a_{ij}\}$
$= [[0, \ 0.9, \ 0.1], [0.2, \ 0, \ 0.8], [0.95, 0, \ 0.05]]$

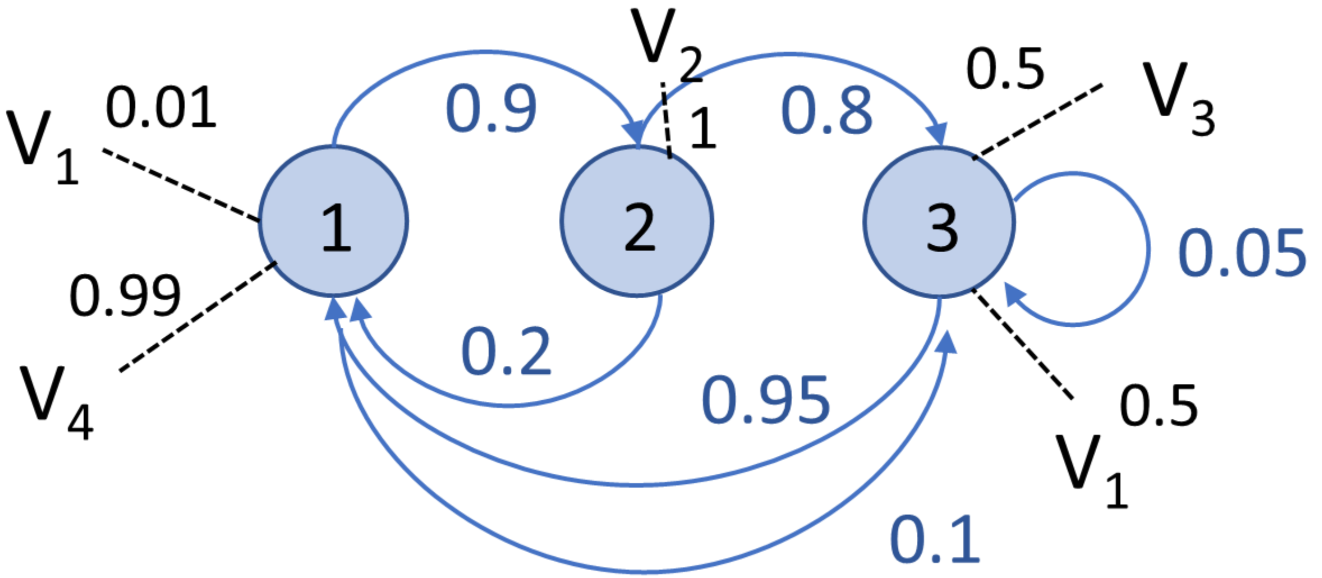
$B = [b_1, b_2, b_3]$
where
$b_i = [b_i(V_1), b_i(V_2), b_i(V_3), b_i(V_4)]$

$\pi = [0.5, 0.2, 0.3]$

$\boldsymbol{\lambda} := (A, B, \boldsymbol{\pi})$

Define specific HMM λ:
Transition matrix *A*, Emission probability *B*, Initial state vector π

HMM Example – Observation Probabilities

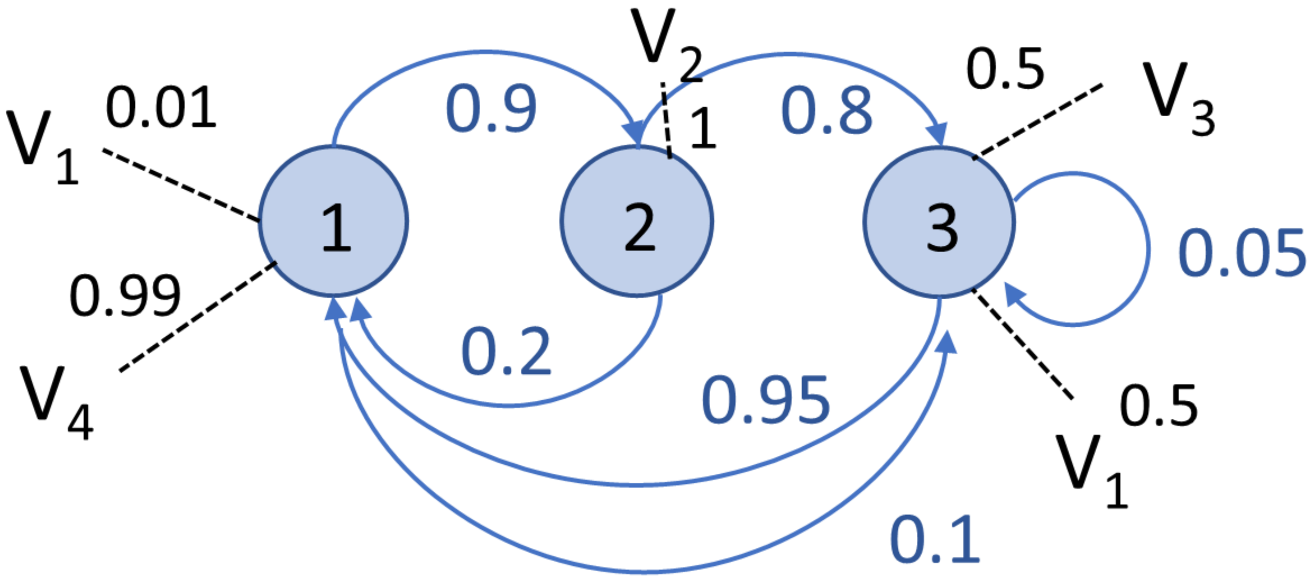


$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

What is the probabilities of observing V_2?

$$\begin{aligned} P(V_2) &= \pi_1 \cdot b_1(V_2) + \pi_2 \cdot b_2(V_2) + \pi_3 \cdot b_3(V_2) \\ &= 0 + 0.2 \cdot 1 + 0 \\ &= 0.2 \end{aligned}$$

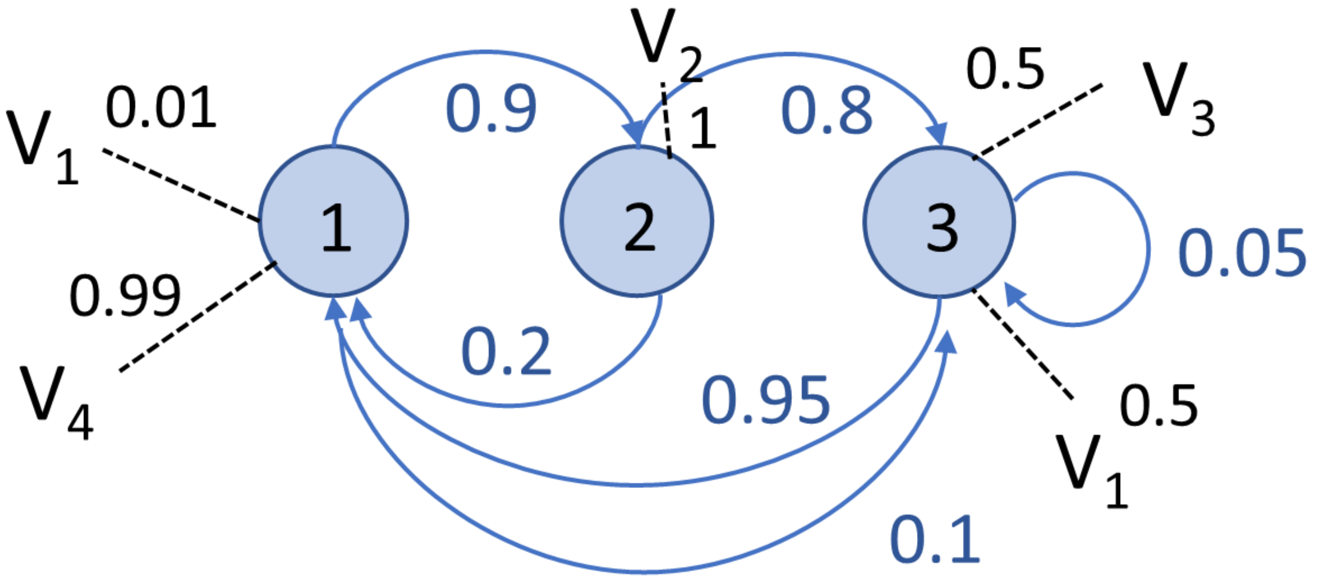
HMM Example – Observation Probabilities



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

What is the probabilities of observing $V_2\,V_2$?

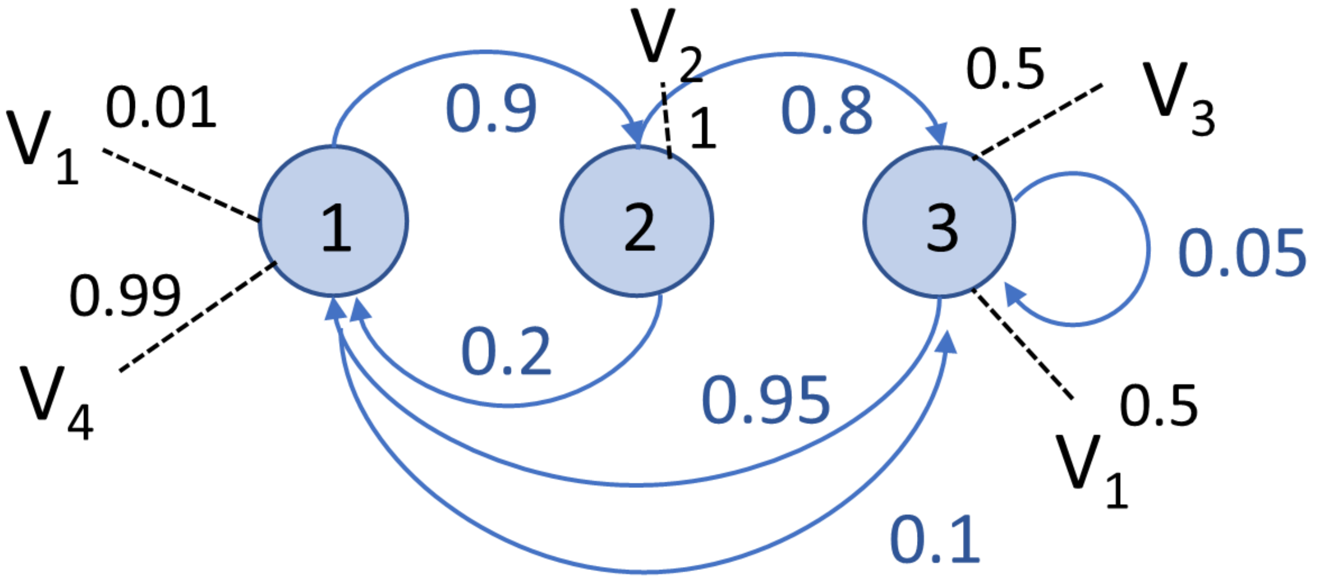
HMM Example – Observation Probabilities



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

$$\begin{aligned}
P(V_2V_2) = {} & \pi_1 \cdot b_1(V_2) \cdot a_{11} \cdot b_1(V_2) + \pi_1 \cdot b_1(V_2) \cdot a_{12} \cdot b_2(V_2) + \pi_1 \cdot b_1(V_2) \cdot a_{13} \cdot b_3(V_2) + \\
& \pi_2 \cdot b_2(V_2) \cdot a_{21} \cdot b_1(V_2) + \pi_2 \cdot b_2(V_2) \cdot a_{22} \cdot b_2(V_2) + \pi_2 \cdot b_2(V_2) \cdot a_{23} \cdot b_3(V_2) + \\
& \pi_3 \cdot b_3(V_2) \cdot a_{31} \cdot b_1(V_2) + \pi_3 \cdot b_3(V_2) \cdot a_{32} \cdot b_2(V_2) + \pi_3 \cdot b_3(V_2) \cdot a_{33} \cdot b_3(V_2) \\
= {} & 0 + 0 + 0 \\
= {} & 0
\end{aligned}$$

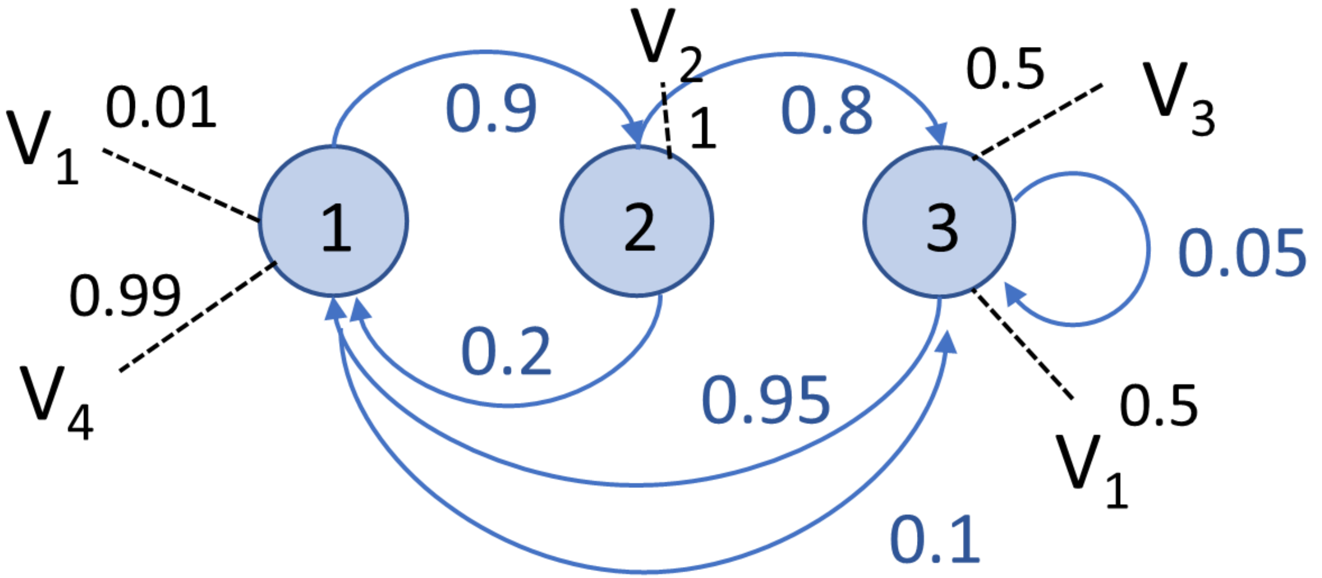
HMM Example – Observation Probabilities



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

What is the probabilities of observing $V_2\,V_1$?

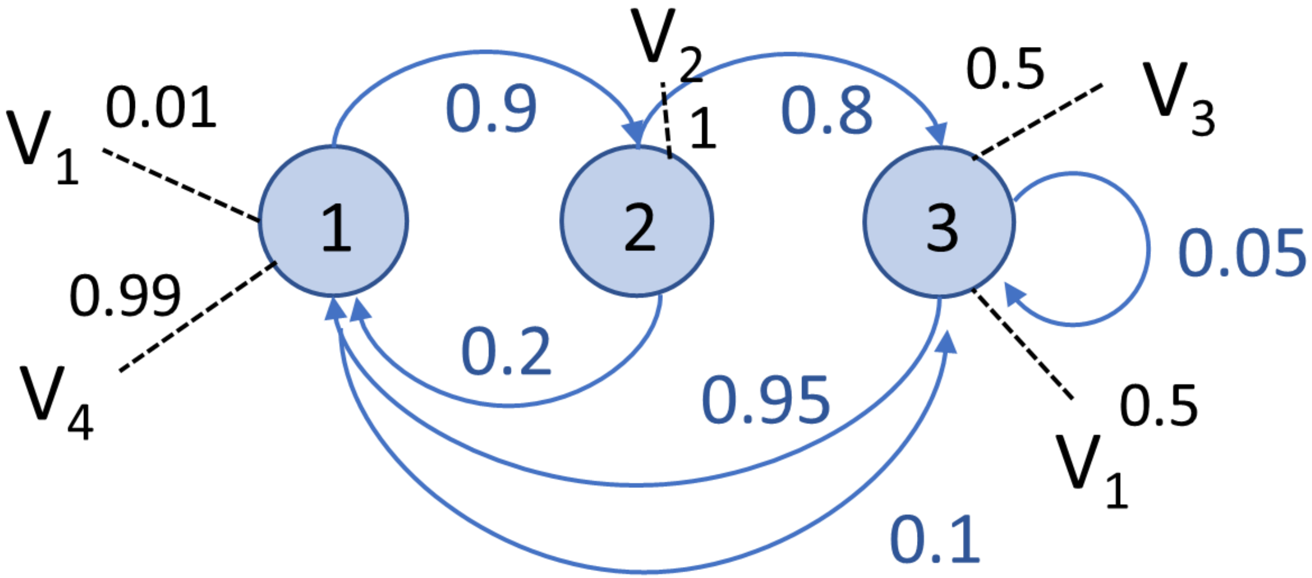
HMM Example – Observation Probabilities



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

$$
\begin{aligned}
P(V_2V_1) = \; & \pi_1 \cdot b_1(V_2) \cdot a_{11} \cdot b_1(V_1) + \pi_1 \cdot b_1(V_2) \cdot a_{12} \cdot b_2(V_1) + \pi_1 \cdot b_1(V_2) \cdot a_{13} \cdot b_3(V_1) + \\
& \pi_2 \cdot b_2(V_2) \cdot a_{21} \cdot b_1(V_1) + \pi_2 \cdot b_2(V_2) \cdot a_{22} \cdot b_2(V_1) + \pi_2 \cdot b_2(V_2) \cdot a_{23} \cdot b_3(V_1) + \\
& \pi_3 \cdot b_3(V_2) \cdot a_{31} \cdot b_1(V_1) + \pi_3 \cdot b_3(V_2) \cdot a_{32} \cdot b_2(V_1) + \pi_3 \cdot b_3(V_2) \cdot a_{33} \cdot b_3(V_1) \\
= \; & 0 + (0.2 * 1 * 0.2 * 0.01 + 0 + 0.2 * 1 * 0.8 * 0.5) + 0 \\
= \; & 0.0804
\end{aligned}
$$

HMM Example – Observation Probabilities



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

What is the probabilities of observing $V_2\, V_1\, V_3$?

Working with Hidden Markov Models

There are three fundamental problems in HMM:

- Evaluation Problem:

  How likely is it that HMM λ computed O?

  Forward or backward procedure

- Recognition Problem:

  Does HMM λ recognize O?

  Viterbi Algorithm

- Learning (= Training) Problem:

  Adjust λ so that Prob $(O|\lambda)$ is locally maximized.

Problem 1: Evaluation Problem

Problem Definition: Given an observation sequence $O=O_1O_2O_3\ldots O_T$ and the model $\boldsymbol{\lambda}$, what is $P(O|\boldsymbol{\lambda})$?

- The most straightforward way: enumerate every possible state sequence of length *T*
  - This is what we have just practiced and we have seen how fast the math gets ugly.

Problem 1: Evaluation Problem

Problem Definition: Given an observation sequence $O=O_1O_2 \ldots O_T$ and the model $\boldsymbol{\lambda}$, what is $P(O|\boldsymbol{\lambda})$?

- Consider **one** such fixed state sequence $Q=q_1q_2 \ldots q_T$, we have
  - $P(O|Q,\lambda) = \Pi_{t=1}^{T} P(O_t|q_t,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdot \ \ldots \cdot b_{q_T}(O_T)$
- The prob. of such a state seq Q can be written as
  - $P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$
- The joint prob. of O and Q:
  - $P(O, Q\,|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$
- With **all** possible state sequences, summing up
  - $P(O\,|\lambda) = \sum_{all\ Q} P(O, Q\,|\lambda) = \sum_{all\ Q} P(O|Q,\lambda)P(Q|\lambda)$

Problem 1: Evaluation Problem

Problem Definition: Given an observation sequence $O = O_1 O_2 O_3 \ldots O_T$ and the model $\boldsymbol{\lambda}$, what is $P(O|\boldsymbol{\lambda})$?

- The most straightforward way: enumerate every possible state sequence of length *T*
  - This is what we have just practiced and we have seen how fast the math gets ugly.
  - The time complexity is exponential. $O(2T\ N^T)$
- More efficient procedure?

Forward Procedure

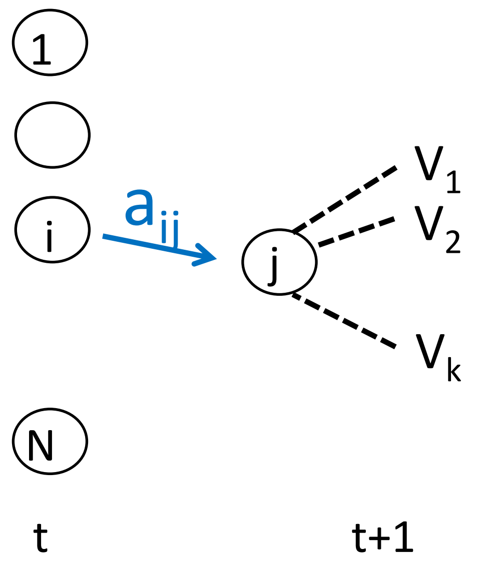
- Define forward variable $\alpha_t(i)$:

$$\alpha_t(i) = P(O_1 O_2 ... O_t, q_t = S_i | \lambda)$$

$O = O_1 O_2 \dots O_{t-1} O_t O_{t+1} \dots O_T$

  - Given the model, $\alpha_t(i)$ is the probability of the **partial** observation sequence $O_1 O_2 \dots O_t$ when it reaches state S_i at time t.
- How about $\alpha_{t+1}(j)$ for some state S_j?
  - assuming $\alpha_t(i)$ is known

$$\alpha_{t+1}(j) = \left(\sum_i \alpha_t(i) \cdot a_{ij}\right) \cdot b_j(O_{t+1})$$



- We can solve $\alpha_t(i)$ inductively.
- Time complexity. $O(T N^2)$

Forward Procedure

1. **Initialization**

   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$

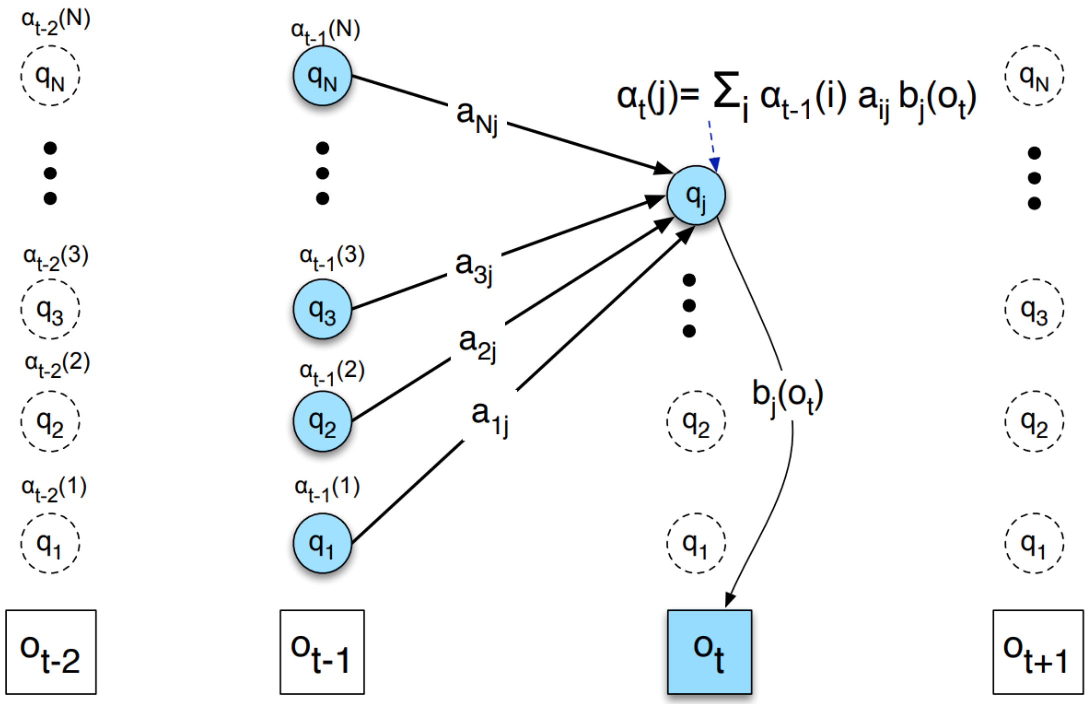
2. **Induction**

   For t = 1, 2, …, T-1

   $\alpha_{t+1}(j) = [\, \sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\,]\, b_j\,(o_{t+1}),\quad 1 <= j <= N$

3. **Termination**

   $\text{Prob}(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



$\alpha_t(i) = P(o_1 o_2 \ldots o_t \,\&\, i \mid \lambda)$

Forward Procedure

1. **Initialization**

   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$

2. **Induction**

   For t = 1, 2, …, T-1

   $\alpha_{t+1}(j) = [\ \sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\ ]\ b_j\,(o_{t+1})$, $1 <= j <= N$

3. **Termination**

   $\text{Prob}(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$

α	1	2	3						T
1									
2									
3		**Fill table forward**							
4									
N									

$\alpha_t(i) = P(o_1 o_2 \ldots o_t \,\&\, i \mid \lambda)$

Example: Forward Procedure

1. Initialization

   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$

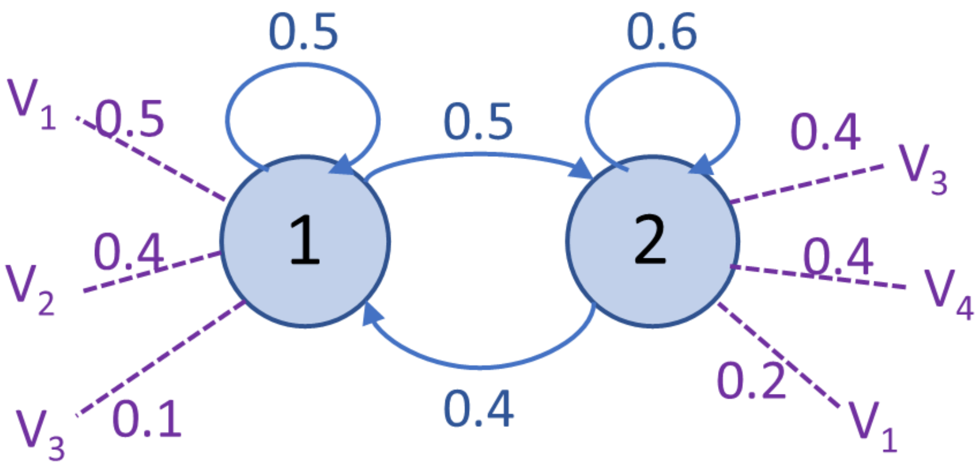
2. Induction

   For t = 1, 2, …, T-1

   $\alpha_{t+1}(j) = [\ \sum_{i=1}^{N} \alpha_t(i)\ a_{ij}\ ]\ b_j(o_{t+1})$, $1 <= j <= N$

3. Termination

   $Prob(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



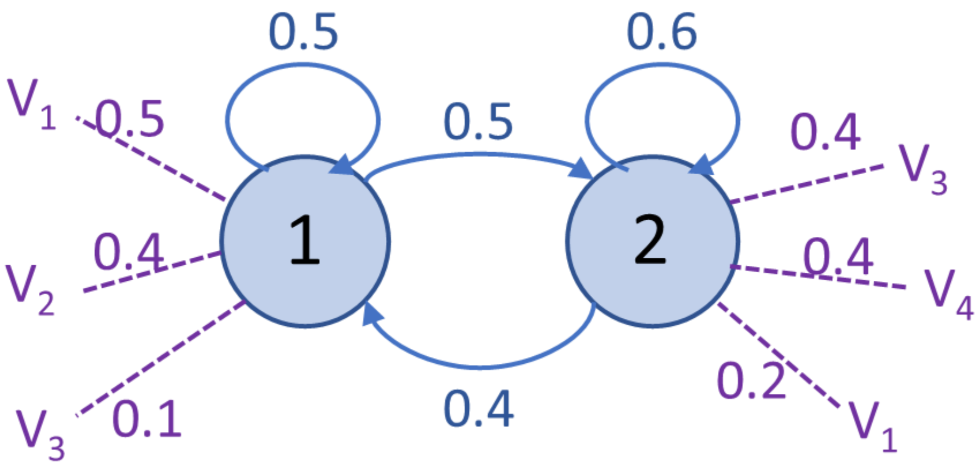
$\pi_1 = 0.2, \pi_2 = 0.8$

α	1	2	3
1			
2			

Observation $O = V_3V_1V_4$ – What is the probability?

Example: Forward Procedure

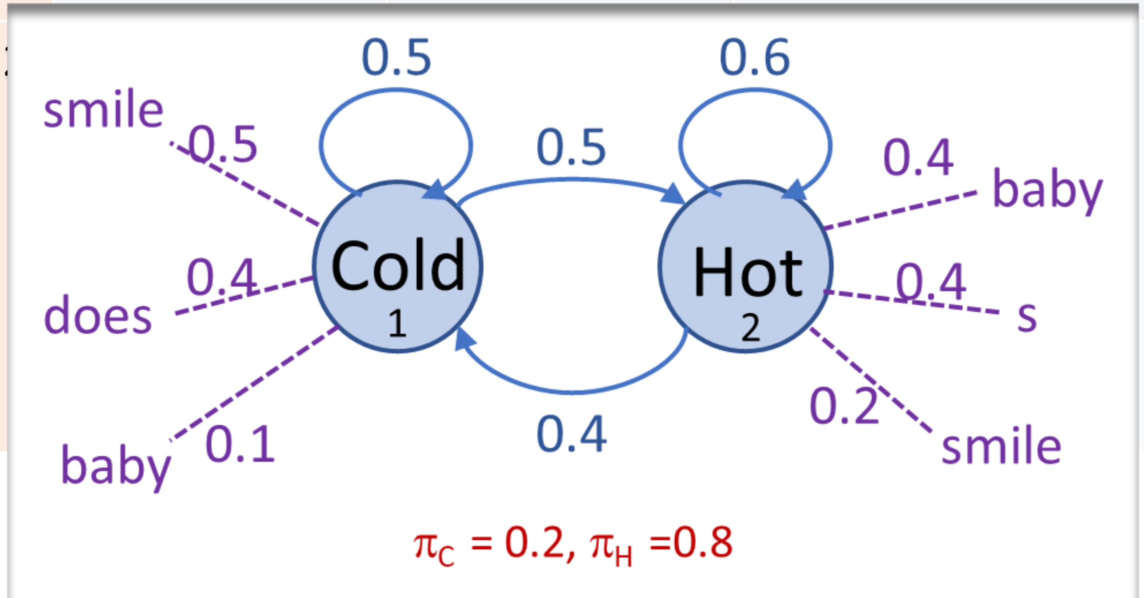
1. Initialization
   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$
2. Induction
   For t = 1, 2, …, T-1
   $\alpha_{t+1}(j) = [\, \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \,]\, b_j(o_{t+1})$, $1 <= j <= N$
3. Termination
   $Prob(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



$\pi_1 = 0.2, \pi_2 = 0.8$

Observation $O = V_3V_1V_4$ – What is the probability?

α	1	2	3
1			



Observation $O = V_3V_1V_4$ → O = "baby smile s"

Example: Forward Procedure

1. Initialization

   $\alpha_1(i) = \pi_i b_i(o_1)$, $\quad 1 <= i <= N$

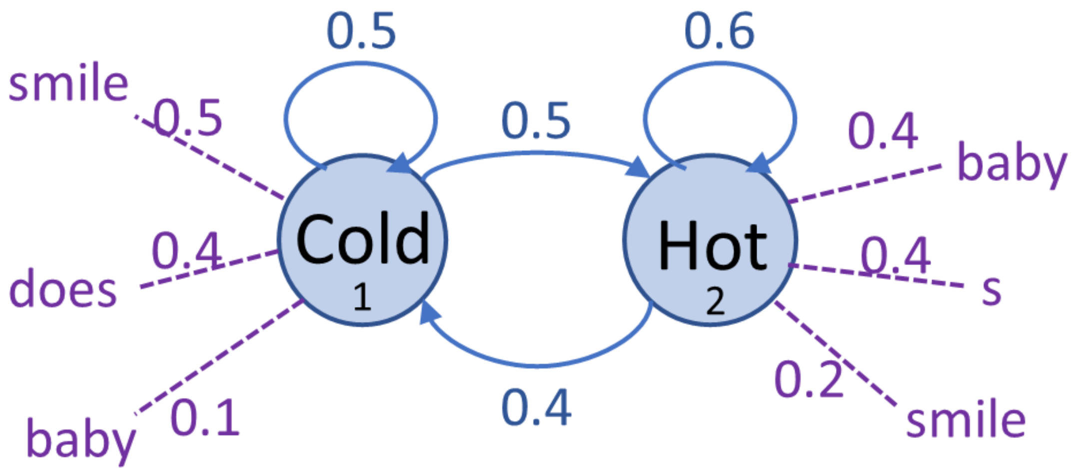
2. Induction

   For t = 1, 2, …, T-1

   $\alpha_{t+1}(j) = [\, \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \,]\, b_j(o_{t+1}), \quad 1 <= j <= N$

3. Termination

   $\text{Prob}(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



$\pi_C = 0.2,\ \pi_H = 0.8$

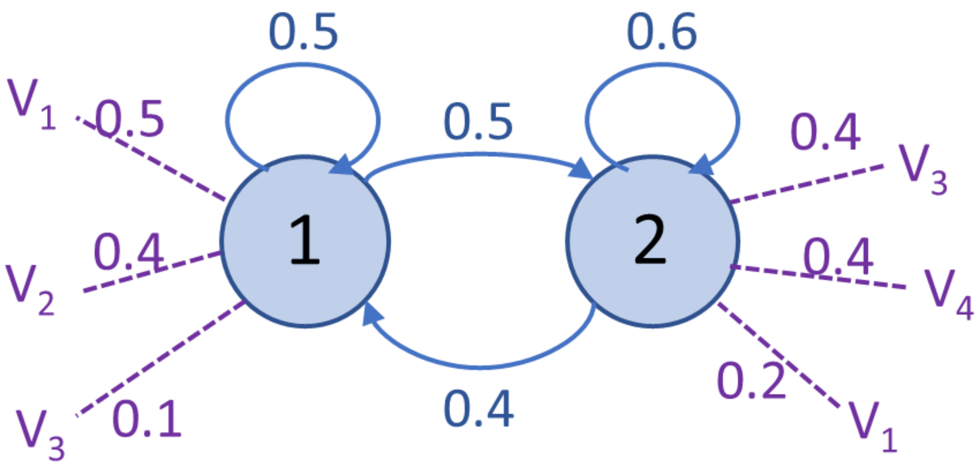
Observation O = “baby smile s” – What is the probability?

α	1	2	3
1 Cold	$\alpha_1(C) = \pi_C b_C(\text{baby}) = 0.2 * 0.1 = 0.02$	$\alpha_2(C) = (\alpha_1(C) \cdot a_{CC} + \alpha_1(H) \cdot a_{HC}) \cdot b_c(\text{smile}) = (0.02 * 0.5 + 0.32 * 0.4) * 0.5 = 0.069$	$\alpha_3(C) = (\alpha_2(C) \cdot a_{CC} + \alpha_2(H) \cdot a_{HC}) \cdot b_c(\text{s}) = (0.069 * 0.5 + 0.0404 * 0.4) * 0 = 0$
2 Hot	$\alpha_1(H) = \pi_H b_H(\text{baby}) = 0.8 * 0.4 = 0.32$	$\alpha_2(H) = (\alpha_1(C) \cdot a_{CH} + \alpha_1(H) \cdot a_{HH}) \cdot b_H(\text{smile}) = (0.02 * 0.5 + 0.32 * 0.6) * 0.2 = 0.0404$	$\alpha_3(H) = (\alpha_2(C) \cdot a_{CH} + \alpha_2(H) \cdot a_{HH}) \cdot b_H(\text{s}) = (0.069 * 0.5 + 0.0404 * 0.6) * 0.4 = 0.023496$

$$P(O|\lambda) = \alpha_3(C) + \alpha_3(H) = 0.023496$$

Example: Forward Procedure

1. Initialization
   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$
2. Induction
   For t = 1, 2, …, T-1
   $\alpha_{t+1}(j) = [\, \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \,]\, b_j(o_{t+1})$, $1 <= j <= N$
3. Termination
   $Prob(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



$\pi_1 = 0.2, \pi_2 = 0.8$

Observation $O = V_3V_1V_4$ – What is the probability?

α	1	2	3
1	$\alpha_1(\mathbf{1}) = \pi_1 b_1(V_3) = 0.2 * 0.1 = 0.02$	$\alpha_2(\mathbf{1}) = (\alpha_1(\mathbf{1}) \cdot a_{11} + \alpha_1(\mathbf{2}) \cdot a_{21}) \cdot b_1(V_1)$	$\alpha_3(\mathbf{1}) = (\alpha_2(\mathbf{1}) \cdot a_{11} + \alpha_2(\mathbf{2}) \cdot a_{21}) \cdot b_1(V_4)$
2	$\alpha_1(\mathbf{2}) = \pi_2 b_2(V_3) = 0.8 * 0.4 = 0.32$	$\alpha_2(\mathbf{2}) = (\alpha_1(\mathbf{1}) \cdot a_{12} + \alpha_1(\mathbf{2}) \cdot a_{22}) \cdot b_2(V_1)$	$\alpha_3(\mathbf{2}) = (\alpha_2(\mathbf{1}) \cdot a_{12} + \alpha_2(\mathbf{2}) \cdot a_{22}) \cdot b_2(V_4)$

$$P(O|\lambda) = \alpha_3(\mathbf{1}) + \alpha_3(\mathbf{2})$$

Example: Forward Procedure

1. Initialization

   $\alpha_1(i) = \pi_i b_i(o_1)$, $1 <= i <= N$

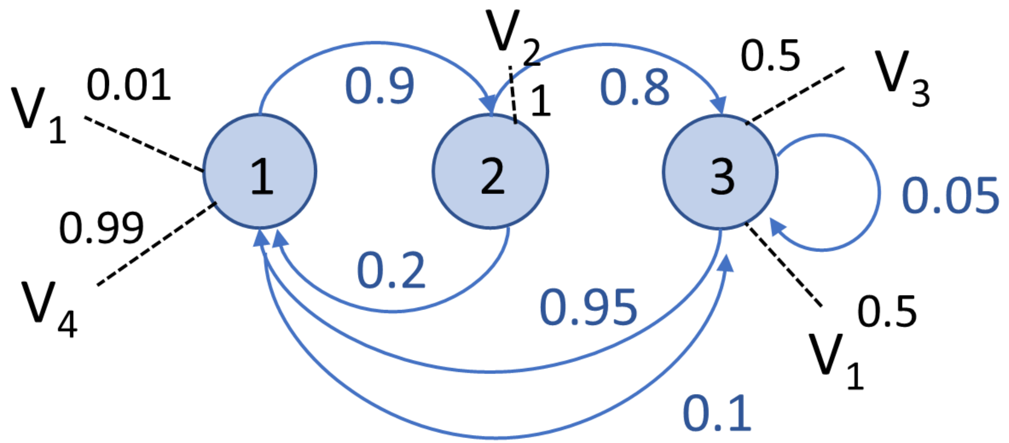
2. Induction

   For t = 1, 2, …, T-1

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

3. Termination

   $Prob(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$



$\pi_1 = 0.5$
$\pi_2 = 0.2$
$\pi_3 = 0.3$

$P(O=V_2V_1V_3) = ?$

α	1	2	3
1	$\alpha_1(\mathbf{1})$	$\alpha_2(\mathbf{1})$	$\alpha_3(\mathbf{1})$
2	$\alpha_1(\mathbf{2})$	$\alpha_2(\mathbf{2})$	$\alpha_3(\mathbf{2})$
3	$\alpha_1(\mathbf{3})$	$\alpha_2(\mathbf{3})$	$\alpha_3(\mathbf{3})$

Backward Procedure

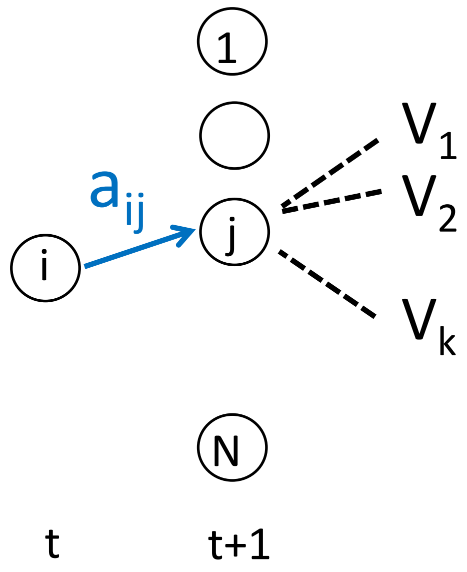
- Define backward variable $\beta_t(i)$:

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T | q_t = S_i, \lambda)$$

  - This is the probability of the observing the **future** sequence $O_{t+1}O_{t+2}...O_T$ given that the current (i.e. at time t) state is S_i.

- It can be computed using future β_{t+1} as follows
  - assuming we know those future values

$$\beta_t(i) = \sum_{j=1} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$



BOSTON UNIVERSITY

Backward Procedure

$$\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T \mid \text{state at } t = i, \lambda)$$

1. Initialization

   $\beta_T(i) = 1$, for all i, $1 <= i <= N$

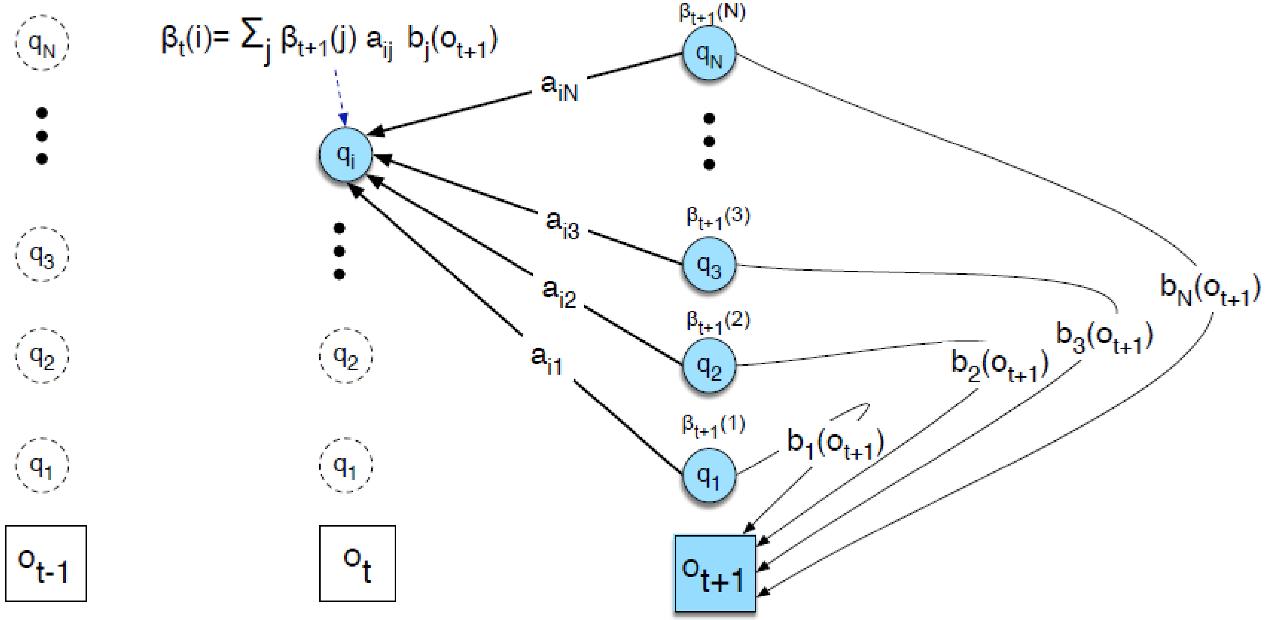
2. Induction

   For t = T-1, …,1, and for all i, $1 <= i <= N$

   $$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_{t+1}(j)\ b_j(o_{t+1})$$

3. Termination

   $$\text{Prob}(O \mid \lambda) = \sum_{i=1}^{N} \beta_1(i)\ \pi_i\ b_i(o_1)$$



Results of Backward & Forward Procedure **must be the same**!

Backward Procedure

$\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T \mid \text{state at } t = i, \lambda)$

1. Initialization

   $\beta_T(i) = 1$, for all i, $1 <= i <= N$

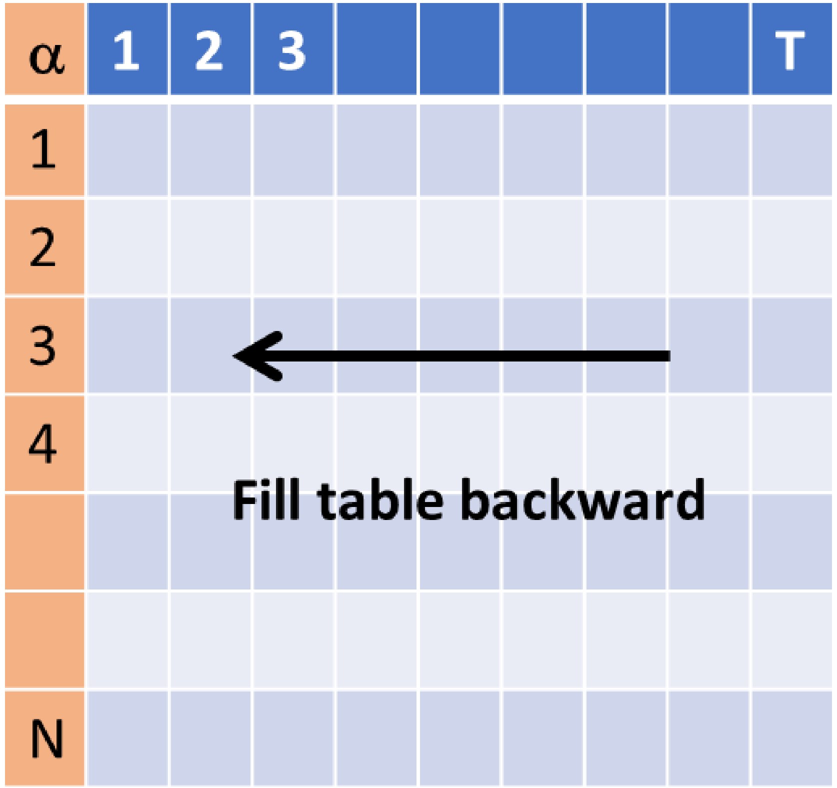
2. Induction

   For t = T-1, …,1, and for all i, $1 <= i <= N$

   $\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_{t+1}(j)\ b_j(o_{t+1})$

3. Termination

   $\text{Prob}(O \mid \lambda) = \sum_{i=1}^{N} \beta_1(i)\, \pi_i\, b_i(o_1)$



Results of Backward & Forward Procedure **must be the same**!

Working with Hidden Markov Models

There are three fundamental problems in HMM:

- Evaluation Problem:

  How likely is it that HMM λ computed O?

  Forward or backward procedure

- Recognition Problem:

  Does HMM λ recognize O?

  Viterbi Algorithm

- Learning (= Training) Problem:

  Adjust λ so that Prob $(O|\lambda)$ is locally maximized.

Problem 2: Recognition Problem

Problem Statement: Given an observation sequence $O=O_1O_2O_3 \ldots O_T$ and the model $\boldsymbol{\lambda}$, what is the **optimal** state sequence $Q = q_1q_2q_3 \ldots q_T$?

- "Optimal?
  - Choose each state q_t that is **individually** most likely
  - Choose the single best **path** $q_1q_2q_3 \ldots q_T$

i.e. to maximize $P(Q|O,\boldsymbol{\lambda})$,
equivalent to maximizing $P(Q,O|\boldsymbol{\lambda})$

Sometimes not possible
($a_{ij} = 0$ for some i and j)

New Problem Statement: Given an observation sequence O and the model $\boldsymbol{\lambda}$, what is the state sequence Q that maximizes the probability $P(Q,O|\boldsymbol{\lambda})$?

Viterbi Algorithm

- Define $\delta_t(i)$:

$Q_{t-1} = q_1 q_2 \dots q_{t-1}$

$$\delta_t(i) = \max_{\boxed{Q_{t-1}}} P(\boxed{Q_{t-1}}, q_t = S_i, O_t|\lambda)$$

  - Given the model, $\delta_t(i)$ is the best score (highest probability) along a single path, at time t, which accounts for the first t observations and ends in state S_i.

- How about $\delta_{t+1}(j)$ for some state S_j?

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1})$$

- To actually retrieve the state sequence, we need to keep track of the argument which maximized $\delta_{t+1}(j)$ for each t and j. We do this via the array $\psi_t(j)$

Viterbi Algorithm

1. Initialization: $\delta_1(i) = \pi_i b_i(o_1)$ for all i, $1 <= i <= N$

   $\psi_1(i) = 0$

2. Recursion: For $t = 2, \ldots, T\text{-}1$, and for all j, $1 <= j <= N$

   $\delta_t(j) = \max_{i=1..N} [\delta_{t-1}(i)\ a_{ij}\ ]\ b_j\ (o_{t+1})$

   $\psi_t(j) = \operatorname{argmax}_{i=1..N} [\delta_{t-1}(i)\ a_{ij}\ ]$

3. Termination: $P(Q^*, O \mid \lambda) = \max_{i=1..N}\ \delta_T(i)$

   $q_T^* = \operatorname{argmax}_{i=1..N}\ \delta_T(i)$

4. Path backtracking:

   $q_t^* = \psi_{t+1}\, q_{t+1}^*$ for $t = T\text{-}1, T\text{-}2, \ldots, 1$

Example: Viterbi Algorithm

1. **Initialization:** $\delta_1(i) = \pi_i b_i(o_1)$ for all i, $1 <= i <= N$

   $\psi_1(i) = 0$

2. **Recursion:** For $t = 2, \ldots, T\text{-}1$, and for all j, $1 <= j <= N$

   $\delta_t(j) = \max_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]\, b_j(o_{t+1})$

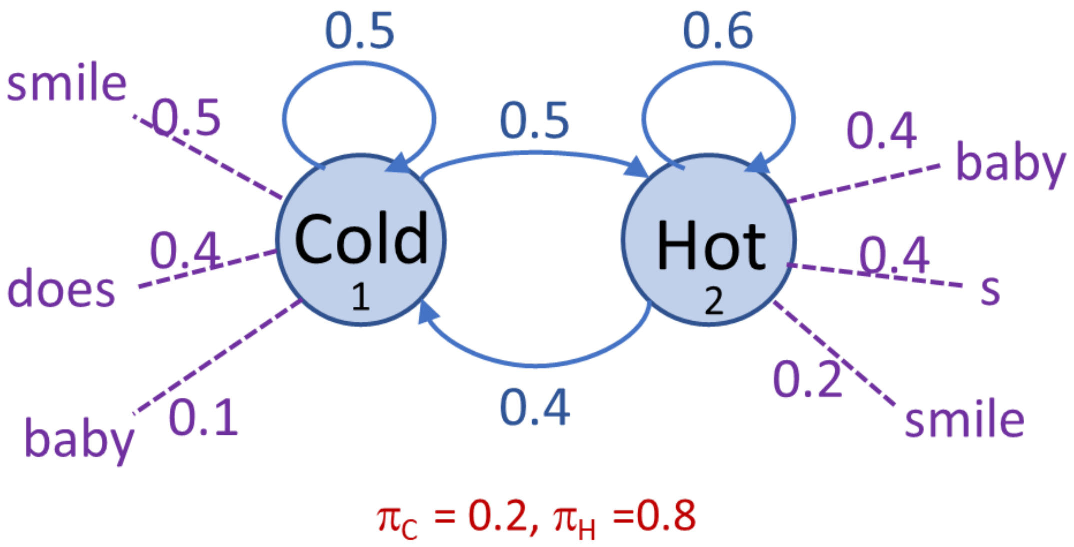
   $\psi_t(j) = \text{argmax}_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]$

3. **Termination:** $P(Q^*, O \mid \lambda) = \max_{i=1..N} \delta_T(i)$

   $q_T^* = \text{argmax}_{i=1..N} \delta_T(i)$

4. **Path backtracking:**

   $q_t^* = \psi_{t+1} q_{t+1}^*$ for $t = T\text{-}1, T\text{-}2, \ldots, 1$



Observation O = "baby smile s" – What is the best path?

δ	1	2	3
Cold			
Hot			

Example: Viterbi Algorithm

1. **Initialization:** $\delta_1(i) = \pi_i b_i(o_1)$ for all i, $1 <= i <= N$

   $\psi_1(i) = 0$

2. **Recursion:** For $t = 2, \ldots, T-1$, and for all j, $1 <= j <= N$

   $\delta_t(j) = \max_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]\, b_j(o_{t+1})$

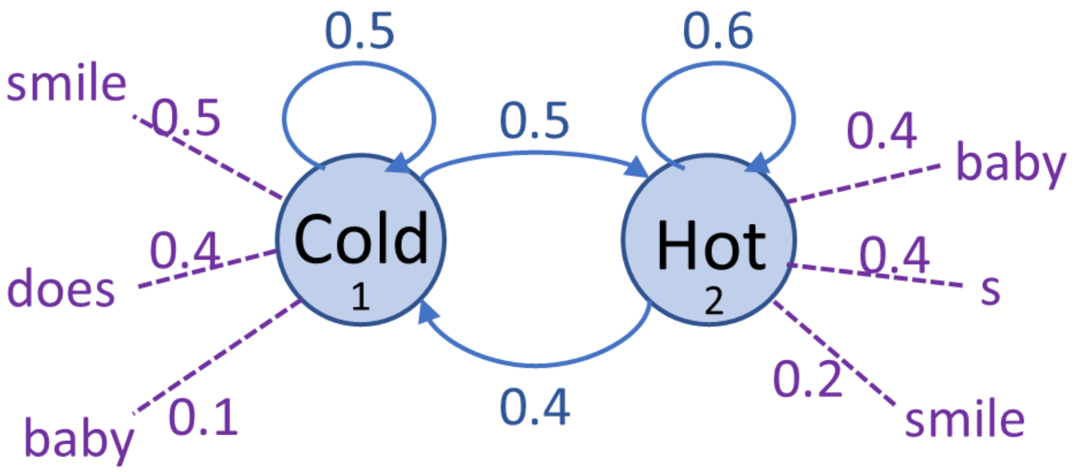
   $\psi_t(j) = \operatorname{argmax}_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]$

3. **Termination:** $P(Q^*, O \mid \lambda) = \max_{i=1..N} \delta_T(i)$

   $q_T^* = \operatorname{argmax}_{i=1..N} \delta_T(i)$

4. **Path backtracking:**

   $q_t^* = \psi_{t+1}\, q_{t+1}^*$ for $t = T-1, T-2, \ldots, 1$



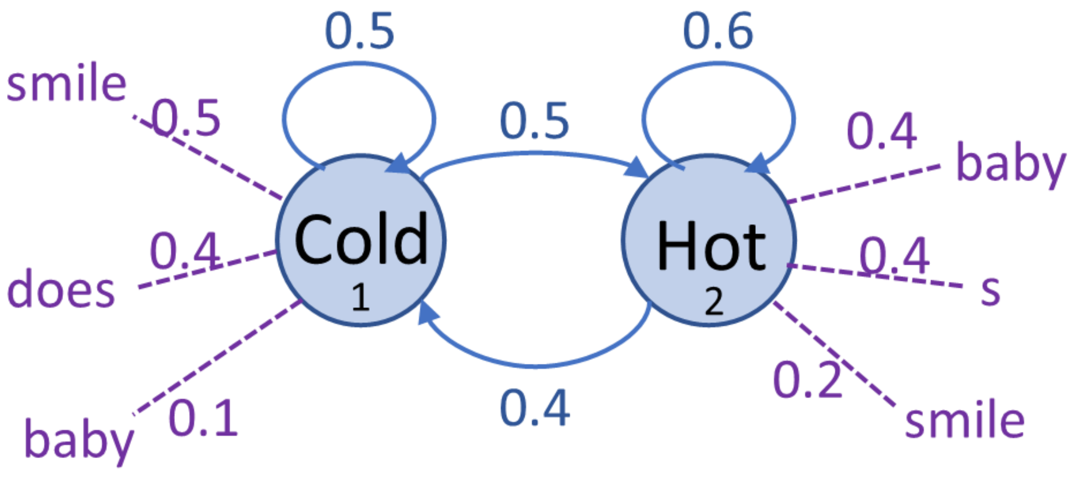
$\pi_C = 0.2,\ \pi_H = 0.8$

Observation O = "baby smile s" – What is the best path?

δ	1	2	3
Cold	$\delta_1(C) = \pi_C b_C(\text{baby}) = 0.2 * 0.1 = 0.02$	$\delta_2(C) = max(\delta_1(C) \cdot a_{CC}, \delta_1(H) \cdot a_{HC}) \cdot b_c(\text{smile}) = max(0.02 * 0.5, 0.32 * 0.4) * 0.5 = 0.064$	$\delta_3(C) = max(\delta_2(C) \cdot a_{CC}, \delta_2(H) \cdot a_{HC}) \cdot b_c(\text{s})$
Hot	$\delta_1(H) = \pi_H b_H(\text{baby}) = 0.8 * 0.4 = 0.32$	$\delta_2(H) = max(\delta_1(C) \cdot a_{CH}, \delta_1(H) \cdot a_{HH}) \cdot b_H(\text{smile}) = \max(0.02 * 0.5, 0.32 * 0.6) * 0.2 = 0.038$	$\delta_3(H) = max(\delta(C) \cdot a_{CH}, \delta_2(H) \cdot a_{HH}) \cdot b_H(\text{s})$

Example: Viterbi Algorithm

1. **Initialization:** $\delta_1(i) = \pi_i b_i(o_1)$ for all *i, 1 <= i <= N*
   $\psi_1(i) = 0$
2. **Recursion:** For *t = 2, …, T-1*, and for all *j, 1 <= j <= N*
   $\delta_t(j) = \max_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]\, b_j(o_{t+1})$
   $\psi_t(j) = \text{argmax}_{i=1..N} [\delta_{t-1}(i)\, a_{ij}]$
3. **Termination:** $P(Q^*, O \mid \lambda) = \max_{i=1..N} \delta_T(i)$
   $q_T^* = \text{argmax}_{i=1..N} \delta_T(i)$
4. **Path backtracking:**
   $q_t^* = \psi_{t+1}\, q_{t+1}^*$ for *t = T-1, T-2,…, 1*



$\pi_C = 0.2,\ \pi_H = 0.8$

Observation O = "baby smile s" – What is the best path?

δ	1	2	3
Cold			
Hot			

Code snippet

Working with Hidden Markov Models

There are three fundamental problems in HMM:

- Evaluation Problem:

  How likely is it that HMM λ computed O?

  Forward or backward procedure

- Recognition Problem:

  Does HMM λ recognize O?

  Viterbi Algorithm

- Learning (= Training) Problem:

  Adjust λ so that Prob $(O|\lambda)$ is locally maximized.

Problem 3: Training HMM

Problem Statement: Given an observation sequence $O=O_1O_2O_3 \ldots O_T$ and the model $\boldsymbol{\lambda}$, how do we adjust $\boldsymbol{\lambda} = (A, B, \boldsymbol{\pi})$ to maximize $P(O|\boldsymbol{\lambda})$?

- If we can solve this problem, then we can train a model starting from some random parameters.
- But there is no optimal way to estimate the parameters.
- One can at best use some iterative procedure to **locally** maximize the probabilities
  - Baum-Welch

Problem 3: Training HMM

- Define

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

  - This is the probability of being in state S_i at time t and state S_j at time t + 1, given the observations and the model.

- Define

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$

  - This is the probability of being in state S_i at time t, given the observations and the model.

Problem 3: Training HMM

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

What are the sums of these two quantities over time steps t from 1 to T - 1?

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from } S_i$$
$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from } S_i \text{ to } S_j$$

This is because they follow Poisson binomial distribution.

Problem 3: Training HMM

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$

we can then update $\boldsymbol{\lambda}$ = (A, B, $\boldsymbol{\pi}$) as

Wait, how do we compute this?

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\overline{b_j(v_k)} = \frac{\sum_{t=1}^{T-1} \mathbf{1}(v_k = O_t)\gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\overline{\pi_i} = \gamma_1(i)$$

Problem 3: Training HMM

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$
$$= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{P(O|\lambda)}$$
$$= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{\sum_i \sum_j P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}$$

where

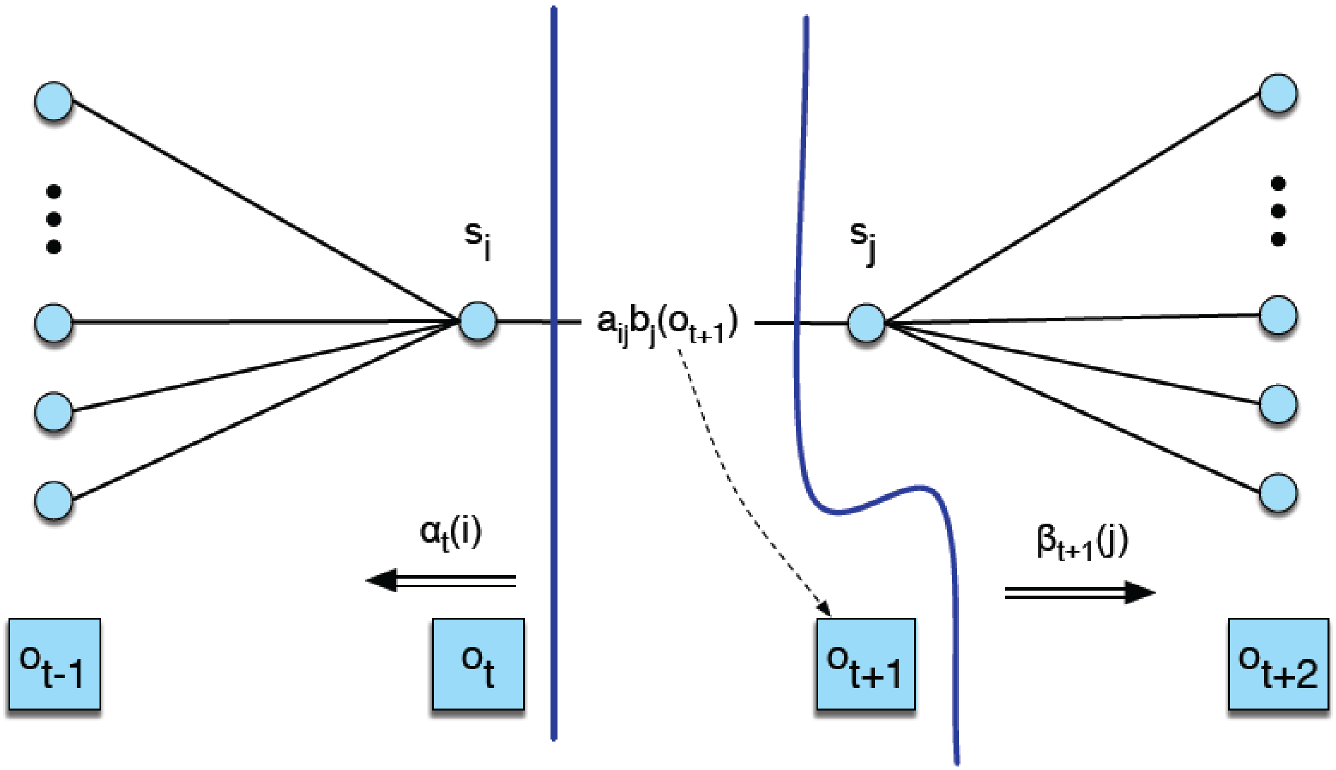
$$P(q_t = S_i, q_{t+1} = S_j, O|\lambda) = \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Recall Notations

$\alpha_t(i) =$ Prob $(O_1 \ldots O_t$ and $q_t = i \mid \lambda)$

$\beta_t(i) =$ Prob $(O_{t+1} \ldots O_T \mid q_t = i$ and $\lambda)$

$$P(q_t = S_i, q_{t+1} = S_j, O|\lambda) = \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)$$

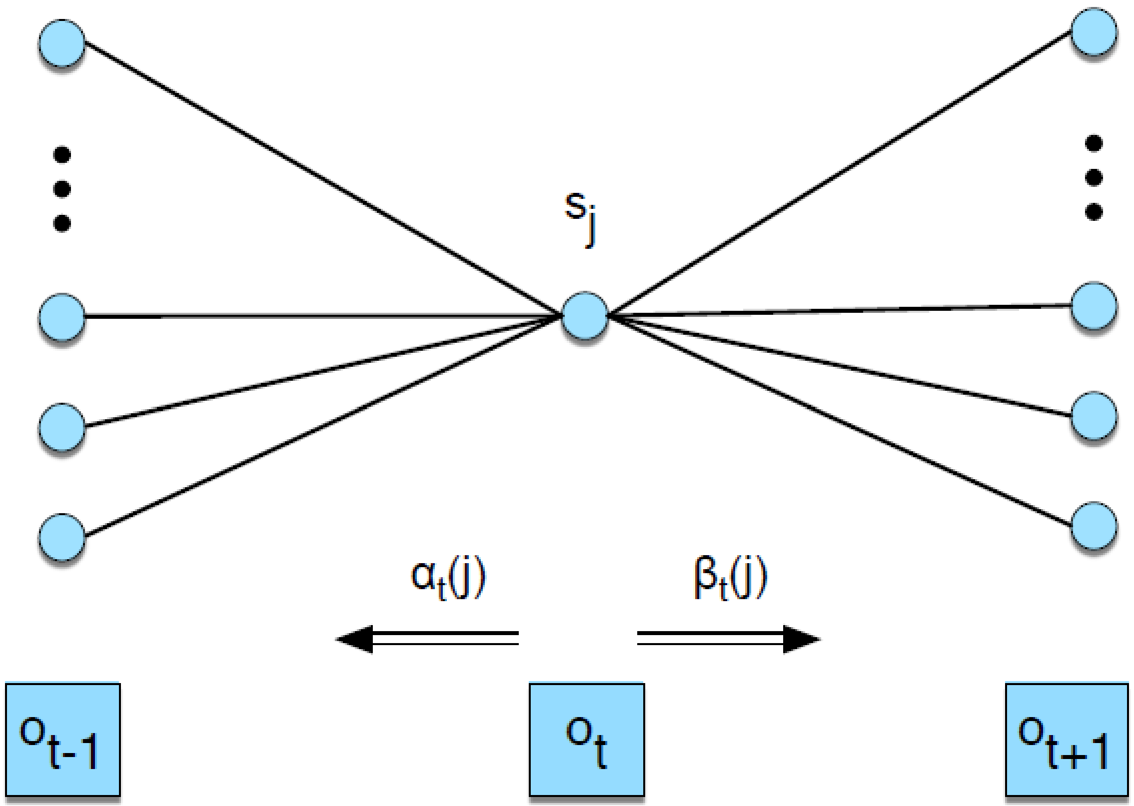


$\alpha_t(i)$ = Prob ($O_1 \ldots O_t$ and $q_t = i \mid \lambda$)

$\beta_t(i)$ = Prob ($O_{t+1} \ldots O_T \mid q_t = i$ and λ)

Recall Notations

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$



Problem 3: Training HMM

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = P(q_t = S_i | O, \lambda)$$

$\alpha_t(i) =$ Prob $(O_1 \ldots O_t$ and $q_t = i \mid \lambda)$

$\beta_t(i) =$ Prob $(O_{t+1} \ldots O_T \mid q_t = i$ and $\lambda)$

$$\xi_t(i, j) = \alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j) \,/\, \text{Prob}(O|\lambda)$$

$$= \alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j) \,/\, \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)$$

$$\gamma_t(i) = \alpha_t(i)\, \beta_t(i) \,/\, \text{Prob}(O|\lambda)$$

$$= \alpha_t(i)\, \beta_t(i) \,/\, \sum_{i=1}^{n} \alpha_t(i)\, \beta_t(i)$$

Baum-Welch Re-estimation

1. $\bar{\pi}_i = \gamma_1(i)$
2. $\bar{a}_{ij} = \sum_{t=1}^{T} \xi_t(i,j) / \sum_{t=1}^{T} \gamma_t(i)$
3. $\bar{b}_j(k) = \sum_{t=1,\, O_t = V_k}^{T} \gamma_t(j) / \sum_{t=1}^{T} \gamma_t(j)$

If Prob(O | $\bar{\lambda}$) > Prob (O | λ)

Re-estimate $\bar{\lambda}$ until (local) maximum is reached

Problem3: Training HMMs

- No analytic solution (= no "simply plug in" solution)
- Use iterative algorithm that learns to represent training data better and better

"Baum-Welch Reestimation Algorithm"

<u>Input:</u> HMM λ, training sequence O

1) Initialization: Guess all probabilities to be uniform
2) Repeat until no better HMM λ can be found:

    Update all probabilities according to equations on next slide

<u>Output:</u> HMM $\lambda*$

Learning Outcomes

- Understand FSM, Deterministic FSM, Markov Network(Chains), HMM
  - The “Markov” Property
  - The “Hidden”
- Know the three fundamental problems in HMM, and how to solve them
  - Forward procedure
  - Viterbi Algorithm
  - Baum-Welch Reestimation Algorithm – using the forward-backward quantities