

Hidden Markov Models



Next Session...HMM



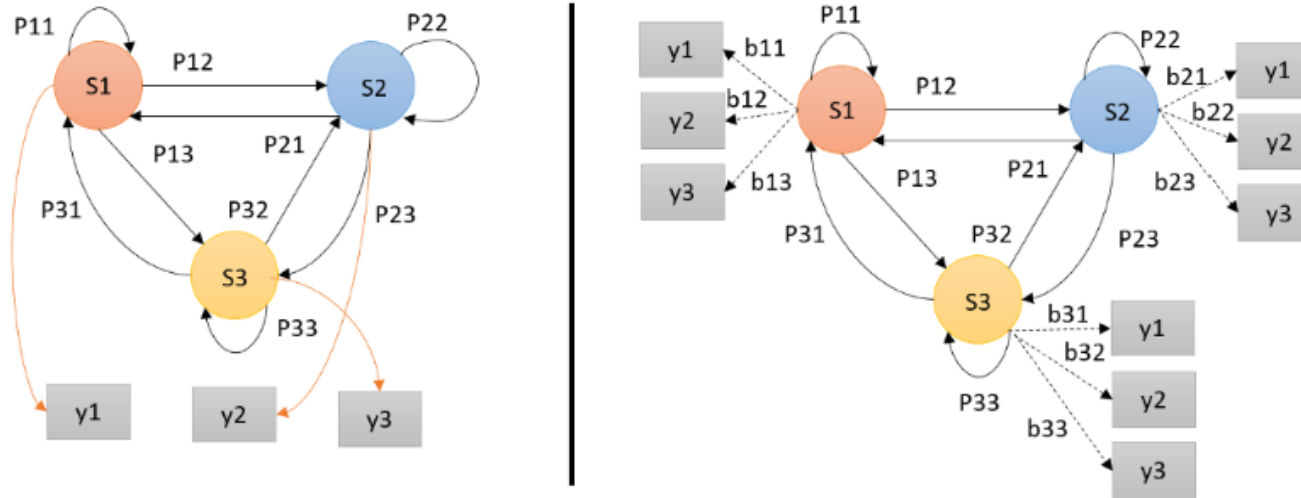
WHAT IS A HIDDEN MARKOV MODEL (HMM)?

- A Hidden Markov Model, is a stochastic model where the states of the model are hidden. Each state can emit an output which is observed.
- **Imagine:** You were locked in a room for several days and you were asked about the weather outside. The only piece of evidence you have is whether the person who comes into the room bringing your daily meal is carrying an umbrella or not.
 - **What is hidden?** Sunny, Rainy, Cloudy
 - **What can you observe?** Umbrella or Not



Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are probabilistic models, it implies that the Markov Model underlying the data is hidden or unknown. More specifically, we only know observational data and not information about the states.



From Markov Chain (left) to Hidden Markov Model (right); where S=states, y=possible observations, P=state transition probabilities and b=observation probabilities

HMM is determined by three model parameters;

- The *start probability*, a vector containing the probability for the state of being the **first** state of the sequence.
- The *state transition probabilities*; a matrix consisting of the probabilities of transitioning from state S_i to state S_j .
- The *observation probability*, the likelihood of a certain observation, y , if the model is in state S_i .



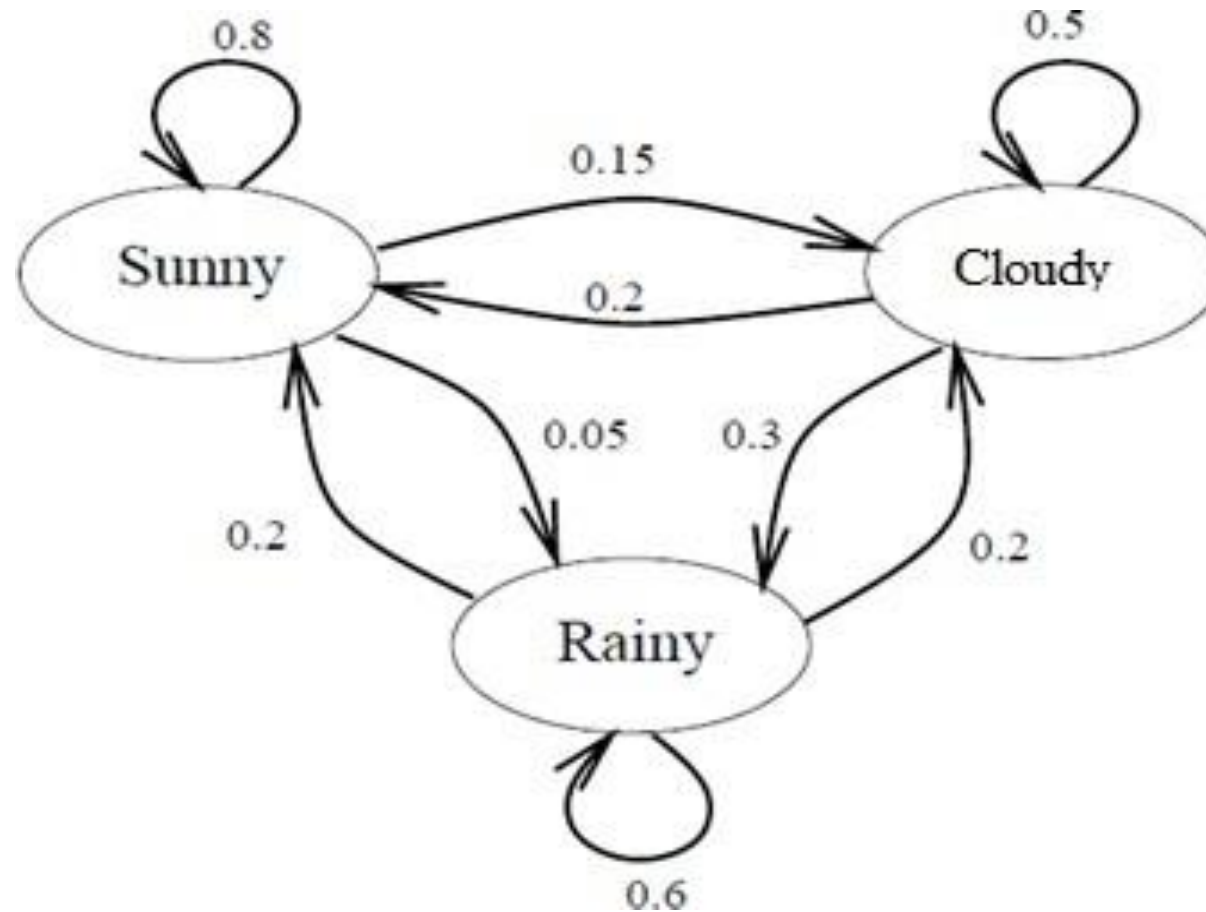
HMMs can be used to solve four fundamental problems;

1. Given the model parameters and the observation sequence, estimate the most likely (hidden) state sequence, this is called a **decoding problem**.
2. Given the model parameters and observation sequence, find the probability of the observation sequence under the given model. This process involves a maximum likelihood estimate of the attributes, sometimes called an **evaluation problem**.
3. Given observation sequences, estimate the model parameters, this is called a **training problem**.
4. Estimate the observation sequences, y_1, y_2, \dots , and model parameters, which maximizes the probability of y . This is called a **Learning or optimization problem**.

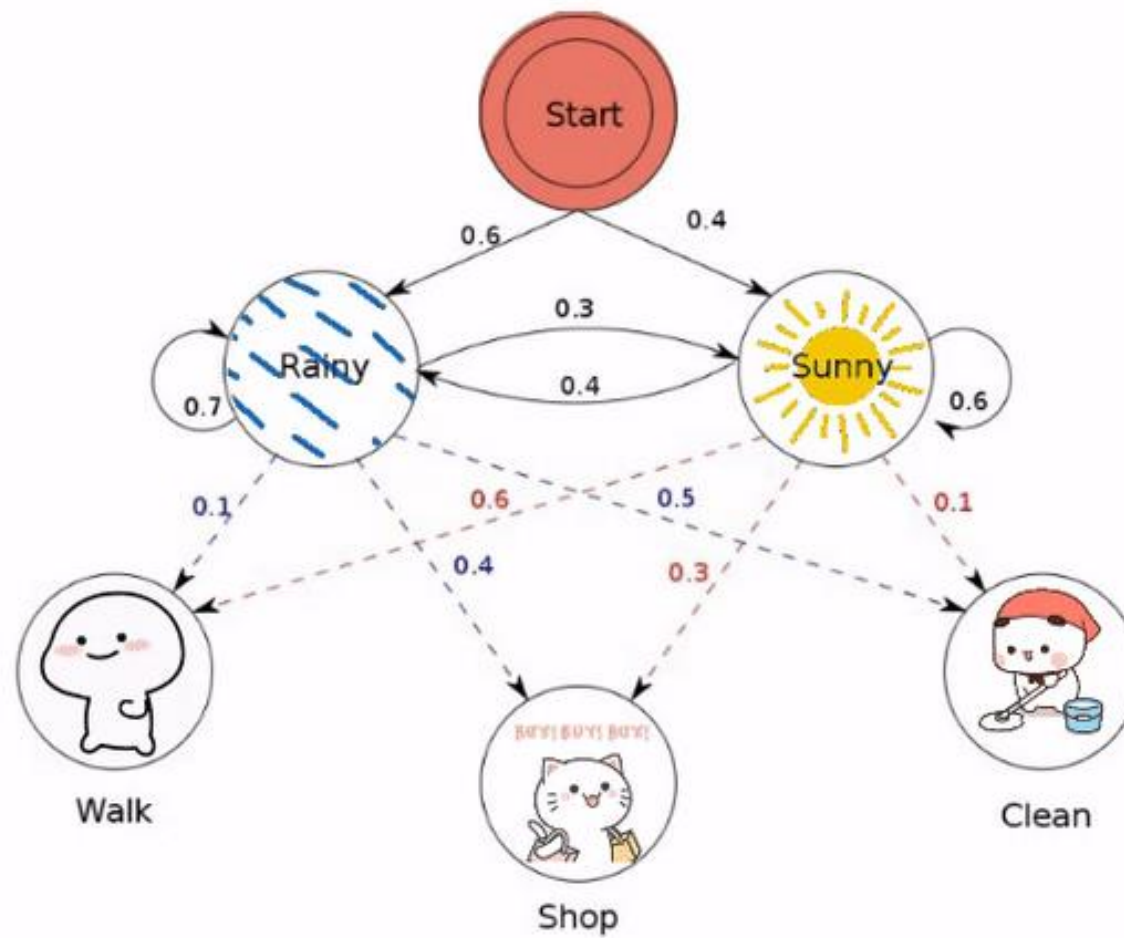


MARKOV CHAIN VS. HMM

- Markov Chain:



○ HMM:



- Let's assume that t days had passed. Therefore, we will have an observation sequence $O = \{o_1, \dots, o_t\}$, where $o_i \in \{\text{Walking, Cleaning, Shopping}\}$
- Each observation comes from an unknown state. Therefore, we will also have an unknown sequence $Q = \{q_1, \dots, q_t\}$, where $q_i \in \{\text{Sunny, Rainy}\}$
We would like to know: $P(q_1, \dots, q_t | o_1, \dots, o_t)$.



HMM MATHEMATICAL MODEL

- From Bayes' Theorem, we can obtain the probability for a particular day as:

$$P(q_i|o) = \frac{P(o_i|q_i)P(q_i)}{P(o_i)}$$

For a sequence of length t :

$$P(q_1, \dots, q_t|o_1, \dots, o_t) = \frac{P(o_1, \dots, o_t|q_1, \dots, q_t)P(q_1, \dots, q_t)}{P(o_1, \dots, o_t)}$$



- From the Markov property:

$$P(q_1, \dots, q_t) = \prod_{i=1}^t P(q_i | q_{i-1})$$

- Independent observations assumption:

$$P(o_1, \dots, o_t | q_1, \dots, q_t) = \prod_{i=1}^t P(o_i | q_i)$$



○ Thus:

$$P(q_1, \dots, q_t | o_1, \dots, o_t) \propto \underbrace{\prod_{i=1}^t P(o_i | q_i) \prod_{i=1}^t P(q_i | q_{i-1})}$$

HMM Parameters:

- Transition probabilities $P(q_i | q_{i-1})$
- Emission probabilities $P(o_i | q_i)$
- Initial state probabilities $P(q_i)$



HMM PARAMETERS

- A HMM is governed by the following parameters:

$$\lambda = \{A, B, \pi\}$$

- State-transition probability matrix A
 - Emission/Observation/State Conditional Output probabilities B
 - Initial (prior) state probabilities π
- Determine the fixed number of states (N):

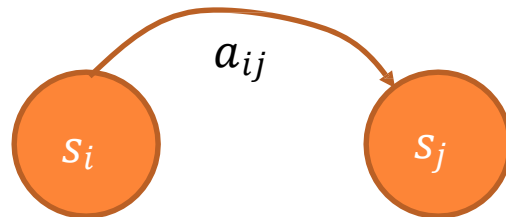
$$S = \{s_1, \dots, s_N\}$$



- State-transition probability matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1N} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2N} \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ a_{N1} & a_{N2} & \cdot & \cdot & \cdot & a_{NN} \end{bmatrix} \quad \begin{array}{l} \sum_{j=1}^N a_{ij} = 1 \text{ (Each row/Outgoing arrows)} \\ a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \leq i, j \leq N \\ a_{ij} \geq 0 \end{array}$$

$a_{ij} \rightarrow$ Transition probability from state s_i to s_j



- Emission probabilities: A state will generate an observation (output), but a decision must be taken according on how to model the output, i.e., as discrete or continuous.
 - **Discrete outputs are modeled using pmfs.**
 - Continuous outputs are modeled using pdfs.

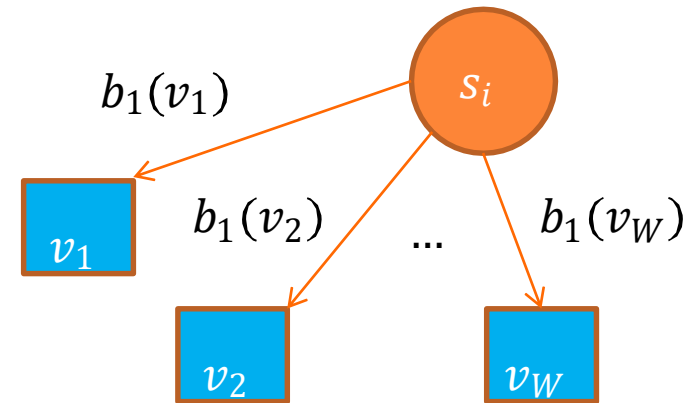


- Discrete Emission Probabilities:

Observation Set: $V = \{v_1, \dots, v_W\}$

$$b_i(v_k) = P(o_t = v_k | q_t = s_i), \quad 1 \leq k \leq W$$

$$B = \begin{bmatrix} b_1(v_1) & b_1(v_2) & \cdot & \cdot & \cdot & b_1(v_W) \\ b_2(v_1) & b_2(v_2) & \cdot & \cdot & \cdot & b_2(v_W) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_N(v_1) & b_N(v_2) & \cdot & \cdot & \cdot & b_N(v_W) \end{bmatrix}$$



- Initial (prior) probabilities: these are the probabilities of starting the observation sequence in state q_i .

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \cdot \\ \cdot \\ \cdot \\ \pi_N \end{bmatrix}$$

$$\pi_i = P(q_1 = s_i), \quad 1 \leq i \leq N$$

$$\sum_{i=1}^N \pi_i = 1$$



HMM components: Here ($X_i = Q_i$) and ($E_i = O_i$)







$Q = q_1 q_2 \dots q_N$	a set of N states
$A = a_{11} \dots a_{ij} \dots a_{NN}$	a transition probability matrix A , each a_{ij} representing the probability of moving from state i to state j , s.t. $\sum_{j=1}^N a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	a sequence of T observations , each one drawn from a vocabulary $V = v_1, v_2, \dots, v_V$
$B = b_i(o_t)$	a sequence of observation likelihoods , also called emission probabilities , each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \dots, \pi_N$	an initial probability distribution over states. π_i is the probability that the Markov chain will start in state i . Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^n \pi_i = 1$

Three Problems of HMM

An influential tutorial by [Rabiner \(1989\)](#), based on tutorials by Jack Ferguson in the 1960s, introduced the idea that hidden Markov models should be characterized by **three fundamental problems**:

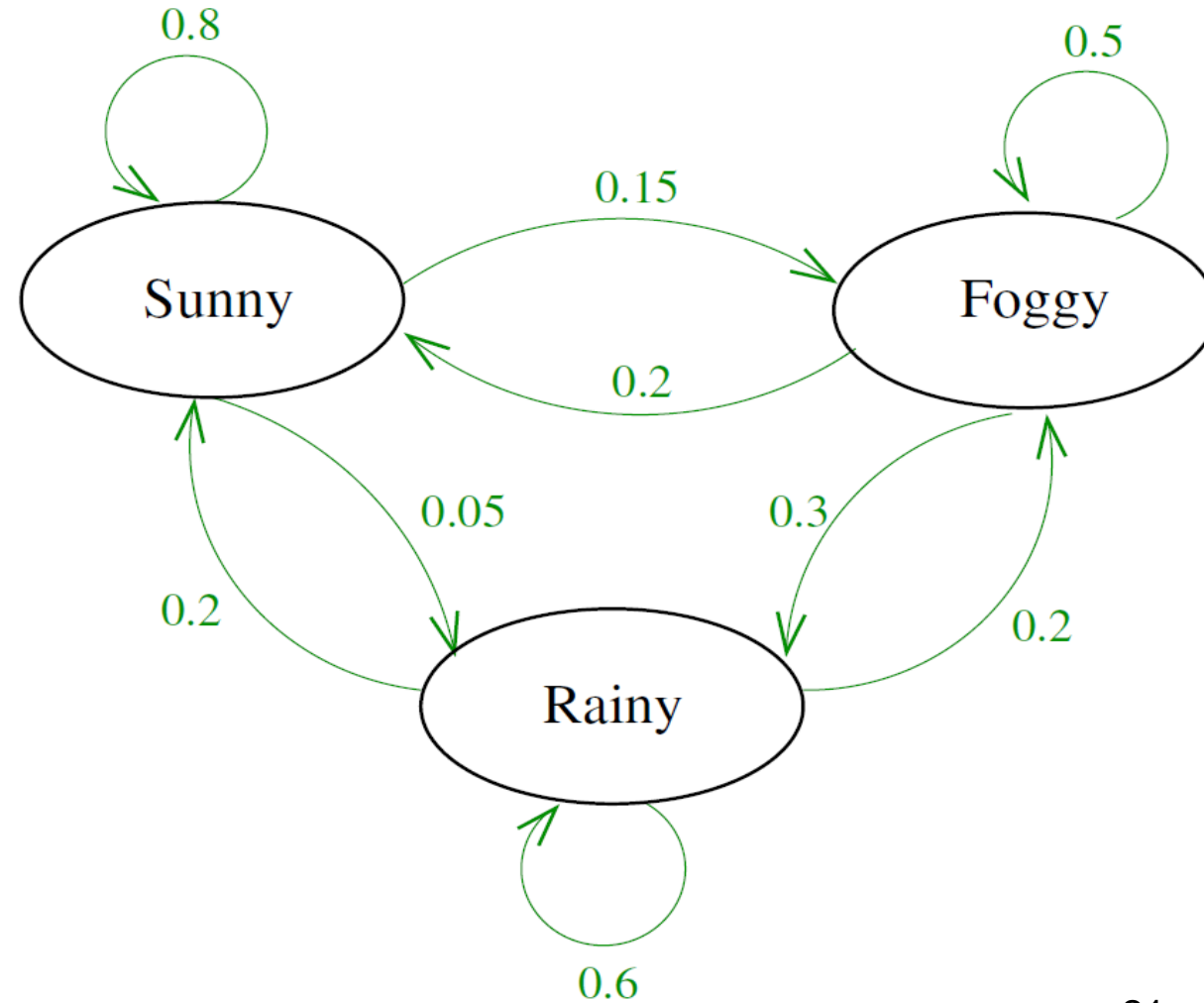
- Problem 1 (Likelihood):** Given an HMM $\lambda = (A, B)$ and an observation sequence O , determine the likelihood $P(O|\lambda)$.
- Problem 2 (Decoding):** Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q .
- Problem 3 (Learning):** Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B .

HMM Example

Today's weather	Tomorrow's weather		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

Probabilities $p(q_{n+1}|q_n)$ of tomorrow's weather based on today's weather

Markov model with Graz weather with state transition probabilities



-
- Given that today is foggy what is the probability that it will be rainy two days from now?
 - There are three ways to get from foggy today to rainy two days from now foggy foggy rainy, foggy rainy rainy and foggy sunny rainy. Therefore we have to sum over these paths.

$$\begin{aligned}
P(w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) &= P(w_2 = \text{Foggy}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_2 = \text{Rainy}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_2 = \text{Sunny}, w_3 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&= P(w_3 = \text{Rainy} \mid w_2 = \text{Foggy})P(w_2 = \text{Foggy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_3 = \text{Rainy} \mid w_2 = \text{Rainy})P(w_2 = \text{Rainy} \mid w_1 = \text{Foggy}) + \\
&\quad P(w_3 = \text{Rainy} \mid w_2 = \text{Sunny})P(w_2 = \text{Sunny} \mid w_1 = \text{Foggy}) \\
\\
&= (0.3)(0.5) + (0.6)(0.3) + (0.05)(0.2) \\
&= 0.34
\end{aligned}$$

Probability of Umbrella

Weather	Probability of umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3







Example 2

2. Suppose you do not know how the weather was when you were locked in. The following three days the caretaker always comes without an umbrella. Calculate the likelihood for the weather on these three days to have been $\{q_1 = \text{☀}, q_2 = \text{☁}, q_3 = \text{☀}\}$. As you do not know how the weather is on the first day, you assume the 3 weather situations are equi-probable on this day (cf. footnote on page 2), and the *prior probability* for sun on day one is therefore $P(q_1 = \text{☀}|q_0) = P(q_1 = \text{☀}) = 1/3$.

$$\begin{aligned} L(q_1 = \text{☀}, q_2 = \text{☁}, q_3 = \text{☀} | x_1 = \text{☂}, x_2 = \text{☂}, x_3 = \text{☂}) &= \\ P(x_1 = \text{☂} | q_1 = \text{☀}) \cdot P(x_2 = \text{☂} | q_2 = \text{☁}) \cdot P(x_3 = \text{☂} | q_3 = \text{☀}) \cdot \\ P(q_1 = \text{☀}) \cdot P(q_2 = \text{☁} | q_1 = \text{☀}) \cdot P(q_3 = \text{☀} | q_2 = \text{☁}) &= \\ 0.9 \cdot 0.7 \cdot 0.9 \cdot 1/3 \cdot 0.15 \cdot 0.2 &= 0.0057 \end{aligned}$$

Example1

$$P(q|x,y)=P(y|q)*P(q|x)$$

Today's weather	Tomorrow's weather		
			
	0.8	0.05	0.15
	0.2	0.6	0.2
	0.2	0.3	0.5

Weather	Probability of umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

$$P(q_1, \dots, q_t | o_1, \dots, o_t) \propto \underbrace{\prod_{i=1}^t P(o_i | q_i) \prod_{i=1}^t P(q_i | q_{i-1})}_{\text{HMM Parameters}}$$

HMM Parameters:

- Transition probabilities $P(q_i | q_{i-1})$
- Emission probabilities $P(o_i | q_i)$
- Initial state probabilities $P(q_i)$

1. Suppose the day you were locked in it was sunny. The next day, the caretaker carried an umbrella into the room. You would like to know, what the weather was like on this second day.

First we calculate the likelihood for the second day to be sunny:

$$\begin{aligned} L(q_2 = \text{Sun} | q_1 = \text{Sun}, x_2 = \text{Umbrella}) &= P(x_2 = \text{Umbrella} | q_2 = \text{Sun}) \cdot P(q_2 = \text{Sun} | q_1 = \text{Sun}) \\ &= 0.1 \cdot 0.8 = 0.08, \end{aligned}$$

then for the second day to be rainy:

$$\begin{aligned} L(q_2 = \text{Rain} | q_1 = \text{Sun}, x_2 = \text{Umbrella}) &= P(x_2 = \text{Umbrella} | q_2 = \text{Rain}) \cdot P(q_2 = \text{Rain} | q_1 = \text{Sun}) \\ &= 0.8 \cdot 0.05 = 0.04, \end{aligned}$$

and finally for the second day to be foggy:

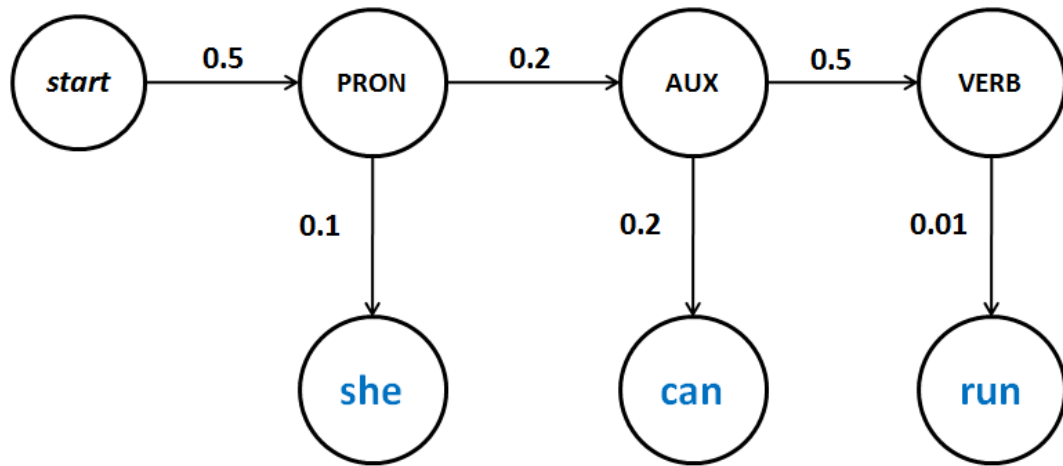
$$\begin{aligned} L(q_2 = \text{Foggy} | q_1 = \text{Sun}, x_2 = \text{Umbrella}) &= P(x_2 = \text{Umbrella} | q_2 = \text{Foggy}) \cdot P(q_2 = \text{Foggy} | q_1 = \text{Sun}) \\ &= 0.3 \cdot 0.15 = 0.045. \end{aligned}$$

Thus, although the caretaker did carry an umbrella, it is most likely that on the second day the weather was sunny.

- A Hidden Markov Model (HMM) is given in the table below;

Transition probabilities	Emission probabilities
$P(\text{NOUN} \text{PRON})=0.001$	$P(\text{she} \text{PRON})=0.1$
$P(\text{PRON} \text{START})=0.5$	$P(\text{run} \text{VERB})=0.01$
$P(\text{VERB} \text{AUX})=0.5$	$P(\text{can} \text{AUX})=0.2$
$P(\text{AUX} \text{PRON})=0.2$	$P(\text{can} \text{NOUN})=0.001$
$P(\text{NOUN} \text{AUX})=0.001$	$P(\text{run} \text{NOUN})=0.001$
$P(\text{VERB} \text{NOUN})=0.2$	
$P(\text{NOUN} \text{NOUN})=0.1$	

Calculate the probability $P(\text{she} | \text{PRON can} | \text{AUX run} | \text{VERB})$. [Or, calculate the probability $P(\text{she can run, PRON AUX VERB})$]



PRON, AUX, and VERB are hidden states

she, can, and run are observations

- The probability of the given sentence can be calculated using the given bi-gram probabilities as follow;
- $P(\text{she} | \text{PRON can} | \text{AUX run} | \text{VERB})$
- $= P(\text{PRON} | \text{START}) * P(\text{she} | \text{PRON}) * P(\text{AUX} | \text{PRON}) * P(\text{can} | \text{AUX}) * P(\text{VERB} | \text{AUX}) * P(\text{run} | \text{VERB})$
- $= 0.5 * 0.1 * 0.2 * 0.2 * 0.5 * 0.01$
- $= 0.00001$
- $= 10^{-5}$

ICE Cream Problem

- Imagine that you are a climatologist in the year 2799 studying the history of global warming. You cannot find any records of the weather in Amaravati, for the summer of 2021, but you do find Jason Eisner's diary, which lists how many ice creams Jason ate every day that summer.
- Our goal is to use these observations to estimate the temperature every day.
- We'll simplify this weather task by assuming there are only two kinds of days: cold (C) and hot (H).
- So the Eisner task is as follows: Given a sequence of observations O (each an integer representing the number of ice creams eaten on a given day) find the 'hidden' sequence Q of weather states (H or C) which caused Jason to eat the ice cream.
- Figure A.2 shows a sample HMM for the ice cream task.
- The two hidden states (H and C) correspond to hot and cold weather, and the observations (drawn from the alphabet $O = \{1,2,3\}$) correspond to the number of ice creams eaten by Jason on a given day

ICE Cream Problem

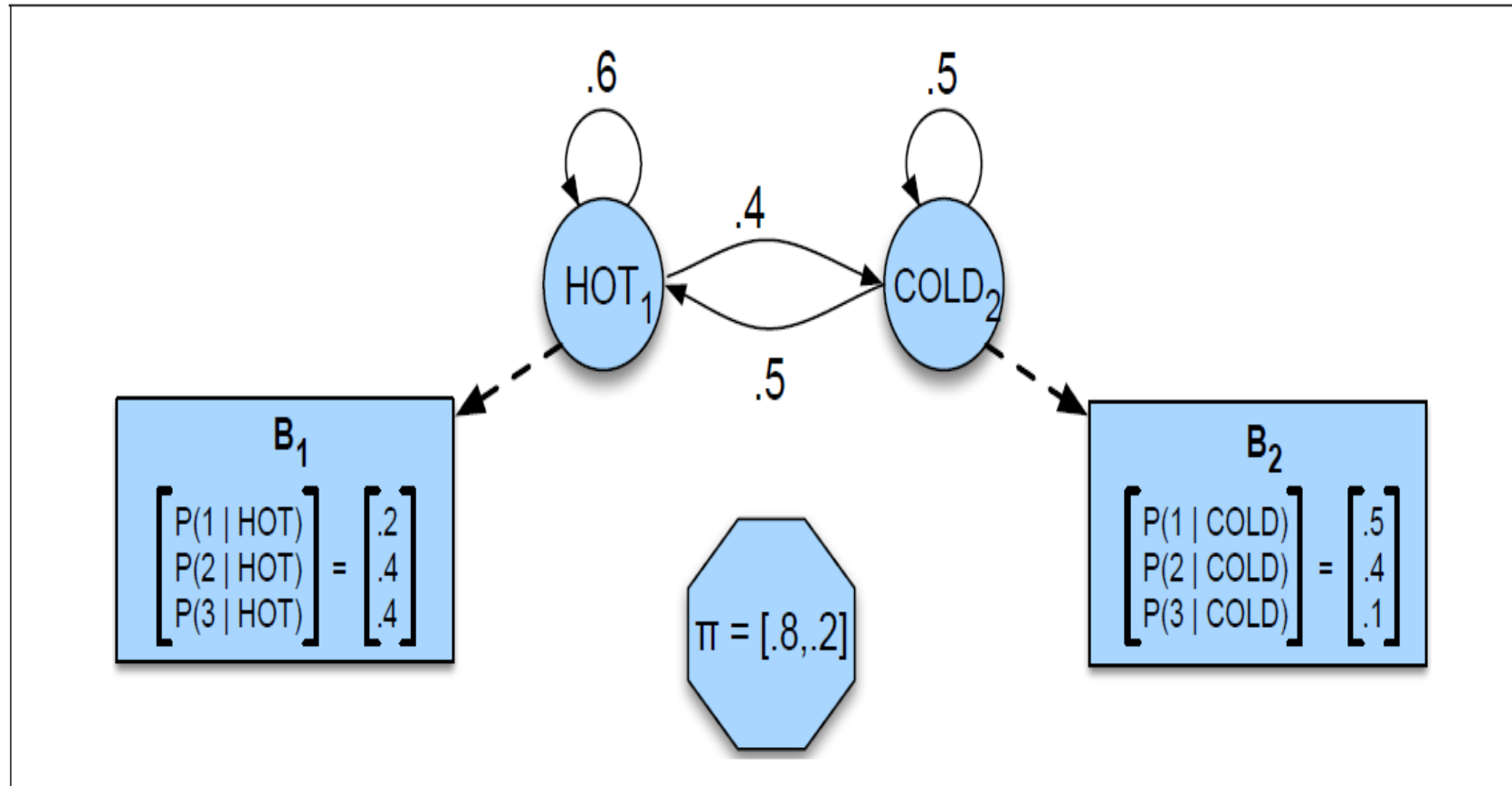


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Question ???

- Compute the probability of ice-cream events $3 \ 1$ instead by summing over all possible weather sequences, weighted by their probability.
- First, let's compute the joint probability of being in a particular weather sequence Q and generating a particular sequence O of ice-cream events.

The computation of the joint probability of our ice-cream observation 3 1 3 and one possible hidden state sequence **hot hot cold** is shown below

$$P(O, Q) = P(O|Q) \times P(Q) = \prod_{i=1}^T P(o_i|q_i) \times \prod_{i=1}^T P(q_i|q_{i-1})$$

$$\begin{aligned} P(3 \ 1 \ 3, \text{hot hot cold}) &= P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \\ &\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \end{aligned}$$

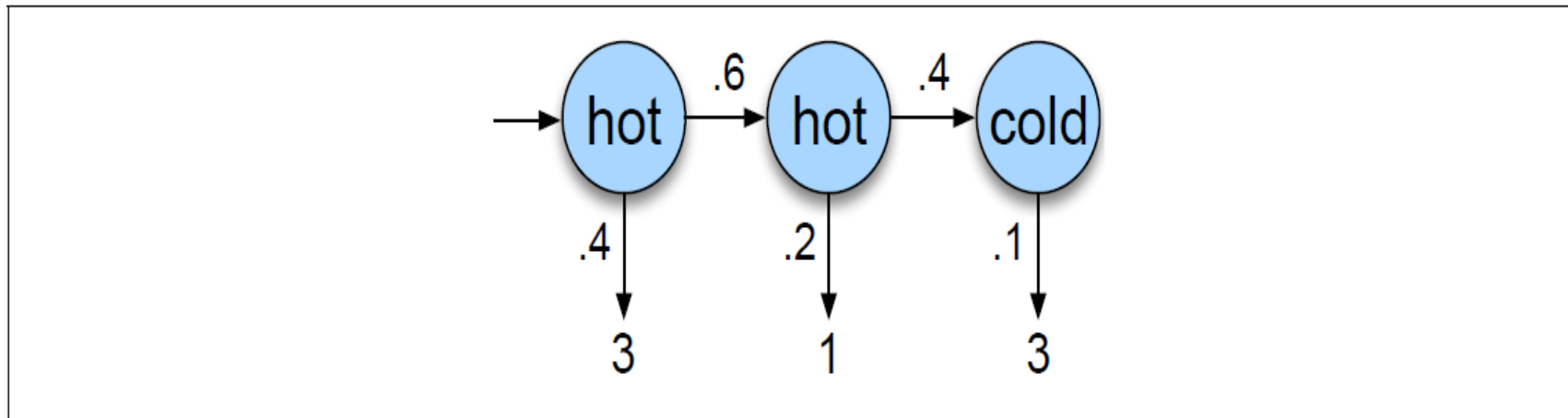


Figure A.4 The computation of the joint probability of the ice-cream events 3 1 3 and the hidden state sequence *hot hot cold*.

-
- $N = \text{No. of hidden states} = 2$
 - $T = \text{Given no. of observations} = 3$
 - $M = N^T = 2^3 = 8$ possibilities (HHC, CCC, HCH...)

- $N = 7$
- $T = 10$
- $N^T = 7^{10}$

How to solve this ?

Each cell of the forward algorithm trellis $\alpha_t(j)$ represents the probability of being in state j after seeing the first t observations, given the automaton λ . The value of each cell $\alpha_t(j)$ is computed by summing over the probabilities of every path that could lead us to this cell. Formally, each cell expresses the following probability:

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = j | \lambda) \quad (\text{A.11})$$

Here, $q_t = j$ means “the t^{th} state in the sequence of states is state j ”. We compute this probability $\alpha_t(j)$ by summing over the extensions of all the paths that lead to the current cell. For a given state q_j at time t , the value $\alpha_t(j)$ is computed as

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t) \quad (\text{A.12})$$

The three factors that are multiplied in Eq. A.12 in extending the previous paths to compute the forward probability at time t are

$\alpha_{t-1}(i)$	the previous forward path probability from the previous time step
a_{ij}	the transition probability from previous state q_i to current state q_j
$b_j(o_t)$	the state observation likelihood of the observation symbol o_t given the current state j

Decoding: The Viterbi Algorithm

For any model, such as an HMM, that contains hidden variables, the task of determining which sequence of variables is the underlying source of some sequence of observations is called the **decoding** task. In the ice-cream domain, given a sequence of ice-cream observations $3\ 1\ 3$ and an HMM, the task of the **decoder** is to find the best hidden weather sequence ($H\ H\ H$). More formally,

Decoding: Given as input an HMM $\lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, \dots, o_T$, find the most probable sequence of states $Q = q_1 q_2 q_3 \dots q_T$.

We might propose to find the best sequence as follows: For each possible hidden state sequence (HHH , HHC , HCH , etc.), we could run the forward algorithm and compute the likelihood of the observation sequence given that hidden state sequence. Then we could choose the hidden state sequence with the maximum observation likelihood. It should be clear from the previous section that we cannot do this because there are an exponentially large number of state sequences.³⁵

Viterbi Algorithm

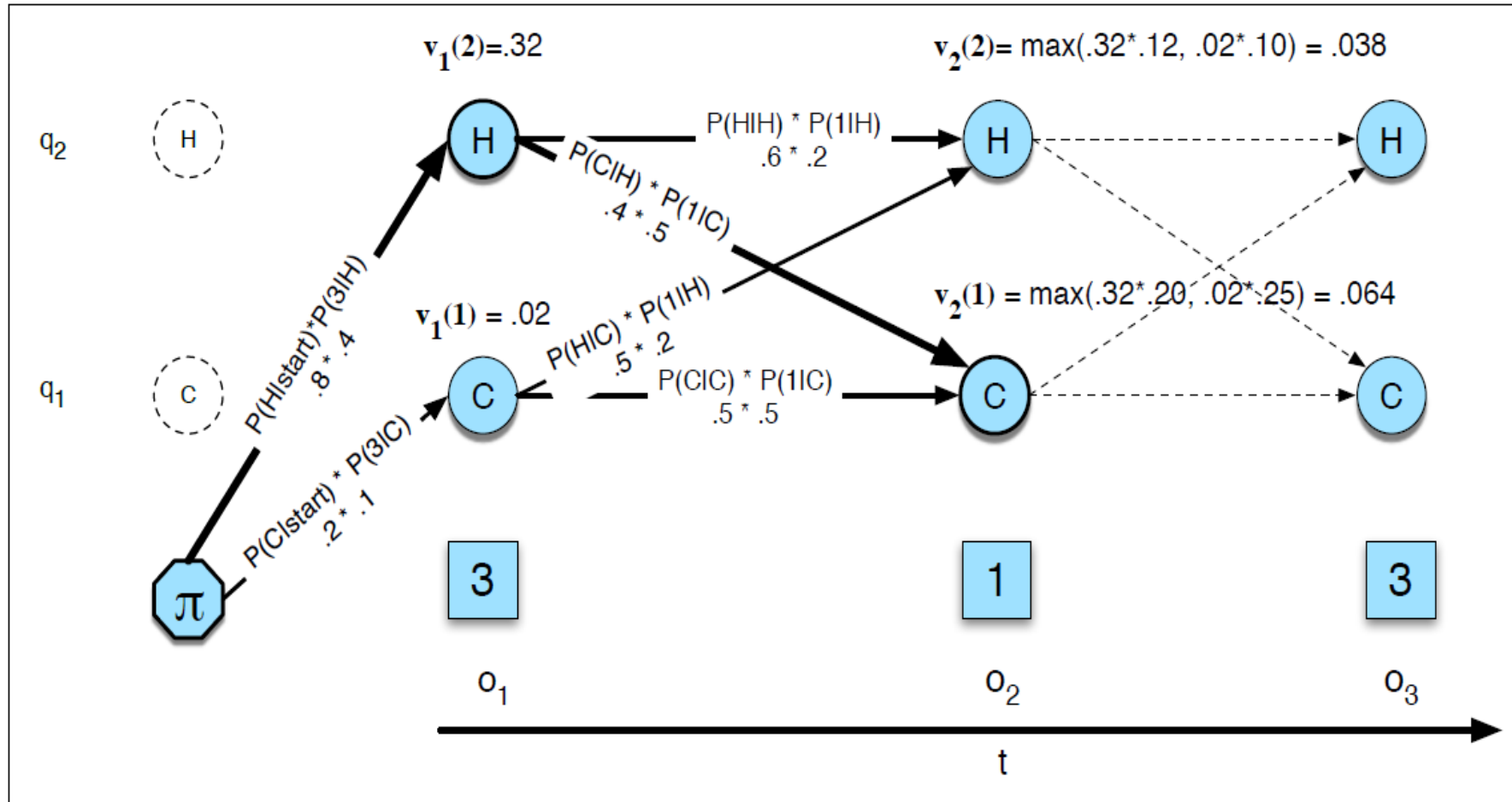


Figure A.8 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events 3 1 3. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.14: $v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. A.13: $v_t(j) = P(q_0, q_1, \dots, q_{t-1}, o_1, o_2, \dots, o_t, q_t = j | \lambda)$.

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j | \lambda) \quad (\text{A.13})$$

Note that we represent the most probable path by taking the maximum over all possible previous state sequences $\max_{q_1, \dots, q_{t-1}}$. Like other dynamic programming algorithms, Viterbi fills each cell recursively. Given that we had already computed the probability of being in every state at time $t - 1$, we compute the Viterbi probability by taking the most probable of the extensions of the paths that lead to the current cell. For a given state q_j at time t , the value $v_t(j)$ is computed as

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t) \quad (\text{A.14})$$

The three factors that are multiplied in Eq. A.14 for extending the previous paths to compute the Viterbi probability at time t are

$v_{t-1}(i)$	the previous Viterbi path probability from the previous time step
a_{ij}	the transition probability from previous state q_i to current state q_j
$b_j(o_t)$	the state observation likelihood of the observation symbol o_t given the current state j

function VITERBI(*observations* of len T , *state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix $viterbi[N, T]$

for each state s **from** 1 **to** N **do** ; initialization step

$viterbi[s, 1] \leftarrow \pi_s * b_s(o_1)$

$backpointer[s, 1] \leftarrow 0$

for each time step t **from** 2 **to** T **do** ; recursion step

for each state s **from** 1 **to** N **do**

$viterbi[s, t] \leftarrow \max_{s'=1}^N viterbi[s', t-1] * a_{s', s} * b_s(o_t)$

$backpointer[s, t] \leftarrow \operatorname{argmax}_{s'=1}^N viterbi[s', t-1] * a_{s', s} * b_s(o_t)$

$bestpathprob \leftarrow \max_{s=1}^N viterbi[s, T]$; termination step

$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^N viterbi[s, T]$; termination step

$bestpath \leftarrow$ the path starting at state $bestpathpointer$, that follows $backpointer[]$ to states back in time

return $bestpath$, $bestpathprob$

Figure A.9 Viterbi algorithm for finding optimal sequence of hidden states. Given an observation sequence and an HMM $\lambda = (A, B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence.

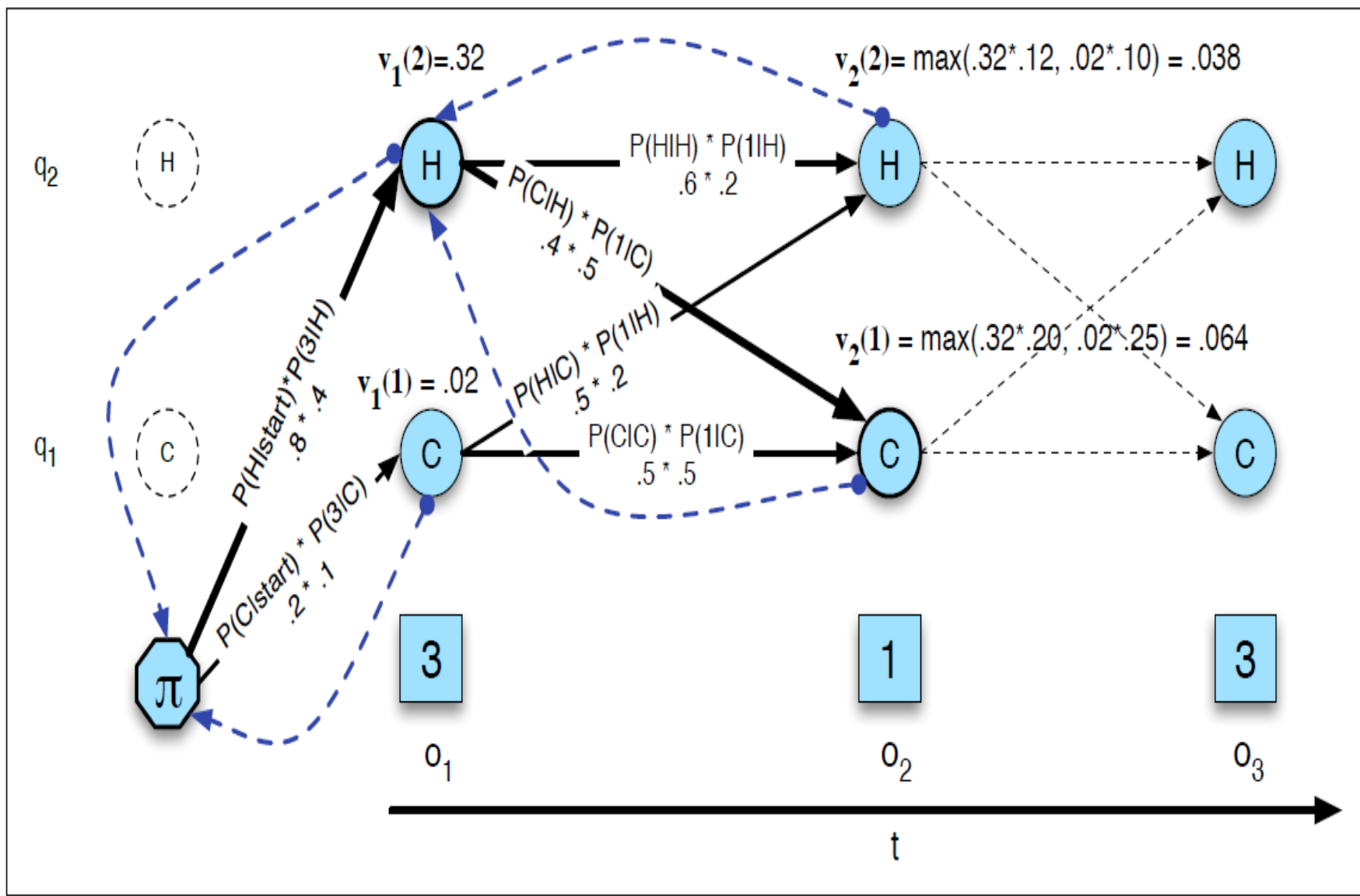


Figure A.10 The Viterbi backtrace. As we extend each path to a new state account for the next observation, we keep a backpointer (shown with broken lines) to the best path that led us to this state.

Finally, we can give a formal definition of the Viterbi recursion as follows:

1. Initialization:

$$\begin{aligned}v_1(j) &= \pi_j b_j(o_1) & 1 \leq j \leq N \\bt_1(j) &= 0 & 1 \leq j \leq N\end{aligned}$$

2. Recursion

$$\begin{aligned}v_t(j) &= \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); & 1 \leq j \leq N, 1 < t \leq T \\bt_t(j) &= \operatorname{argmax}_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); & 1 \leq j \leq N, 1 < t \leq T\end{aligned}$$

3. Termination:

$$\text{The best score: } P^* = \max_{i=1}^N v_T(i)$$

$$\text{The start of backtrace: } q_T^* = \operatorname{argmax}_{i=1}^N v_T(i)$$

Real HMM Examples

- **Speech recognition HMMs:**
 - Observations are acoustic signals (continuous valued)
 - States are specific positions in specific words (so, tens of thousands)
- **Machine translation HMMs:**
 - Observations are words (tens of thousands)
 - States are translation options
- **Robot tracking:**
 - Observations are range readings (continuous)
 - States are positions on a map (continuous)

- <https://jonathan-hui.medium.com/machine-learning-hidden-markov-model-hmm-31660d217a61>