Markov Chains

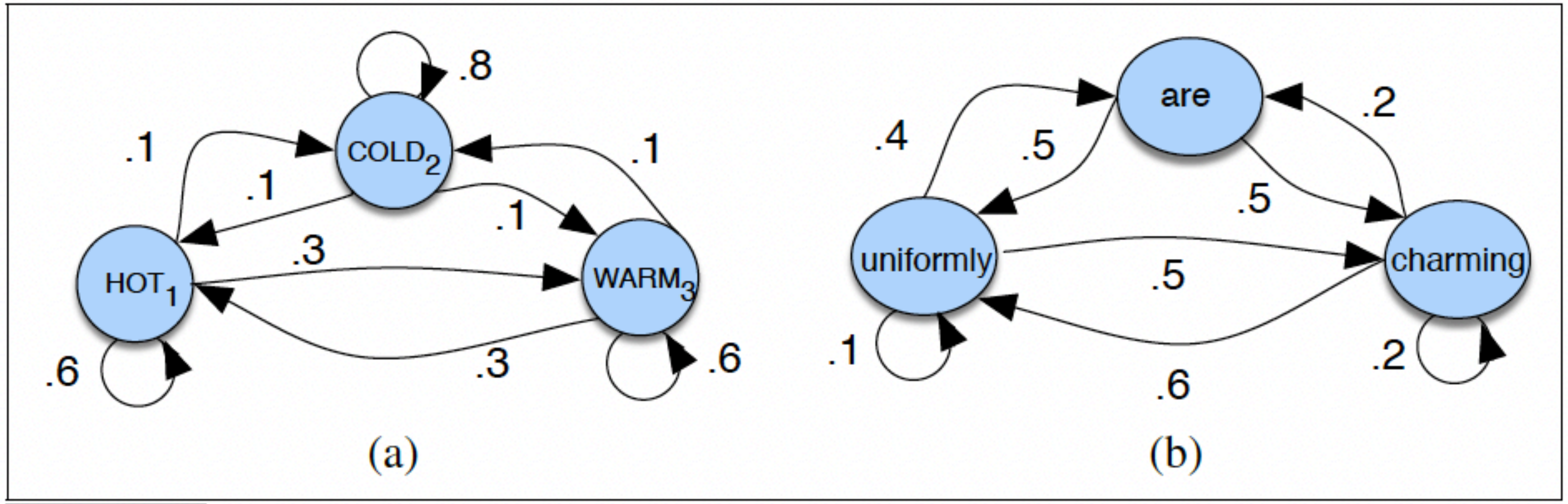


Figure A.1 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

Markov assumption

More formally, consider a sequence of state variables $q_1, q_2, ..., q_i$. A Markov model embodies the **Markov assumption** on the probabilities of this sequence: that when predicting the future, the past doesn't matter, only the present.

$$\textbf{Markov Assumption:} \quad P(q_i = a | q_1 ... q_{i-1}) = P(q_i = a | q_{i-1}) \qquad \text{(A.1)}$$

Markov Chains

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, ..., \pi_N$	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, $\sum_{i=1}^{n} \pi_i = 1$

The Weather-Ice Cream HMM

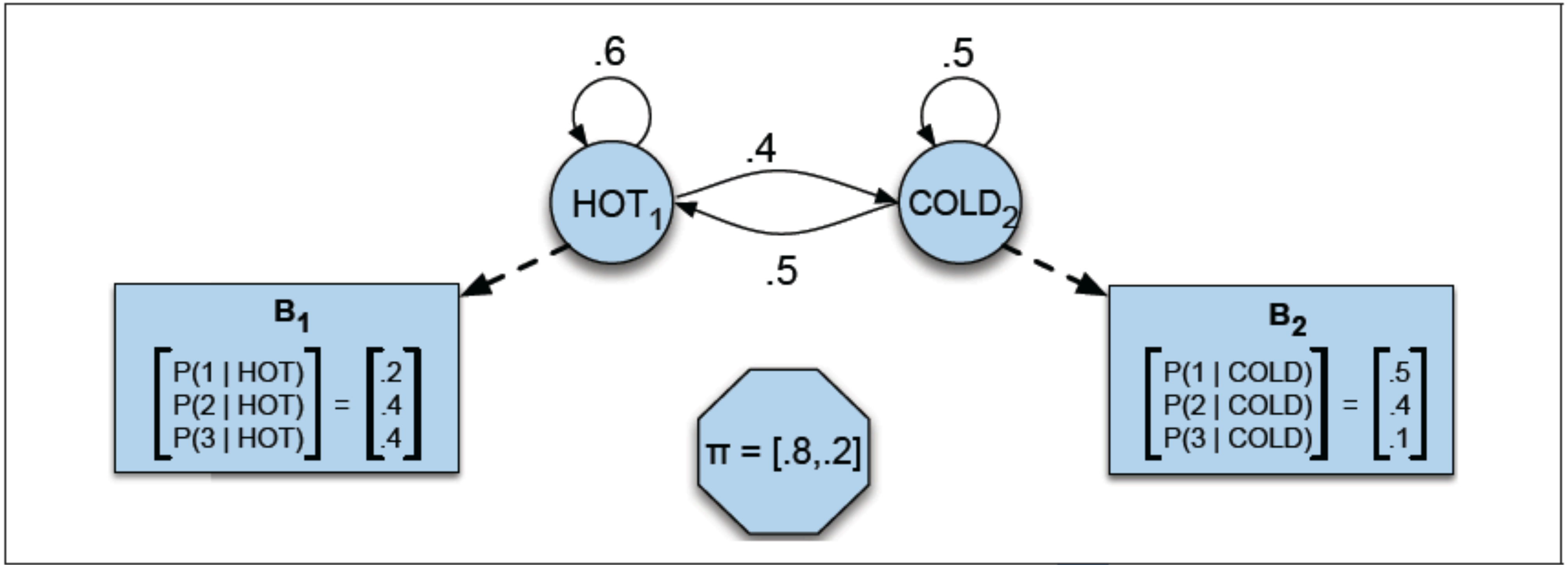


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Hidden Markov Models

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	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, $\sum_{i=1}^{n} \pi_i = 1$

The three problems for HMMs

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.

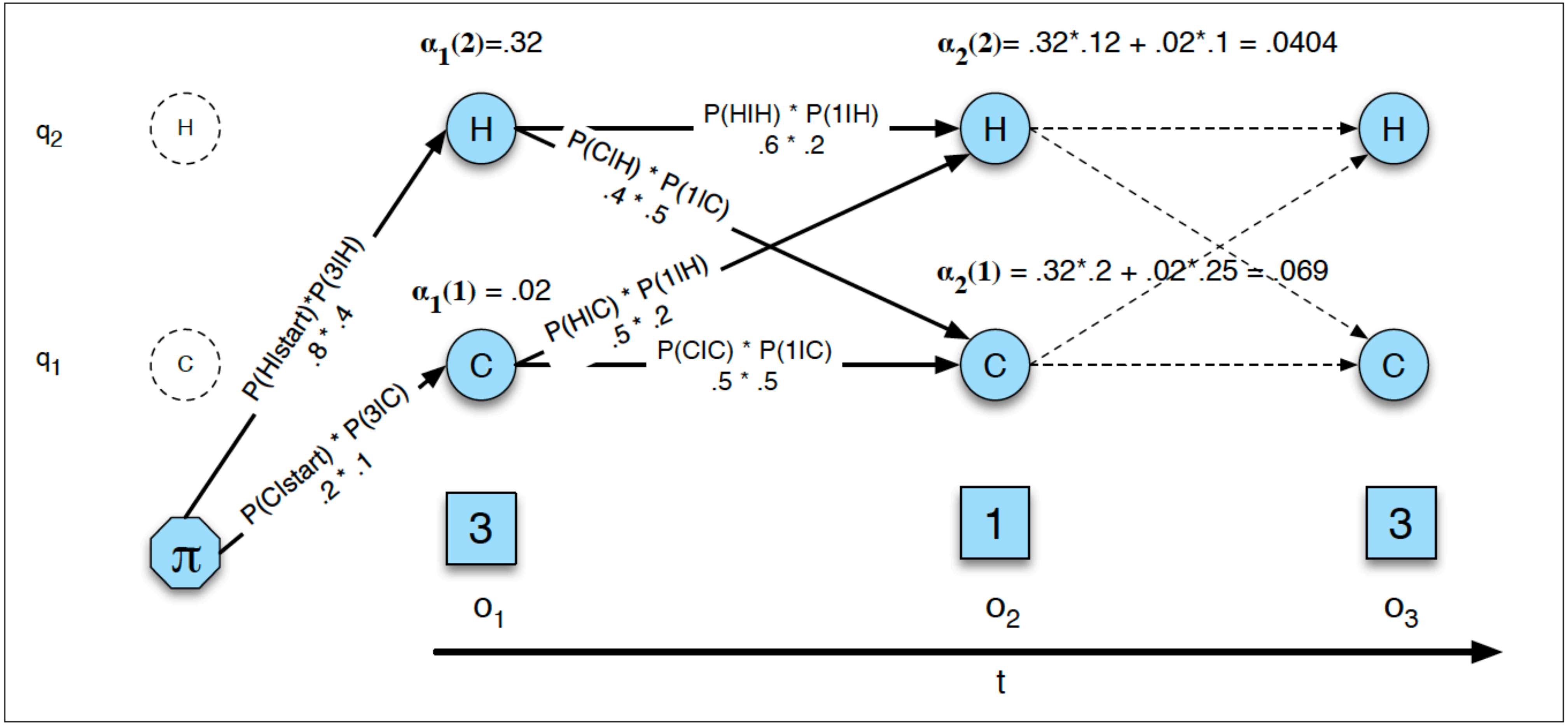


Figure A.5 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.12: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. A.11: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$.

Probabilistic Graphical Models

Also known as Bayes Nets or Belief Nets

Judea Pearl of UCLA got a Turing award for his work on these

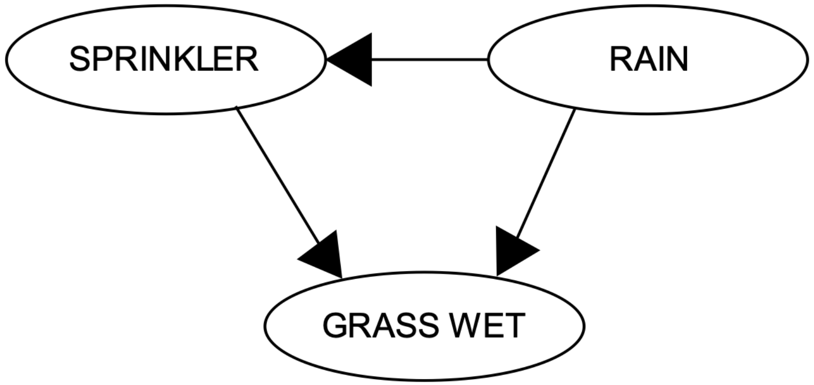
Special cases of these were known before e.g. Hidden Markov Models

Joint probability distributions

- Caanonical example is a multivariate Gaussian. The joint probability density is specified by the mean, a n-dimensional vector, and the covariance matrix, a $n \times n$ symmetric matrix.
- Suppose we have n binary random variables. Then the joint distribution can be specified by a table with 2^n entries. This quickly becomes intractable, both for specification and subsequently in estimation from data.
- The secret to tractability is "conditional independence" . This information can be captured by a directed acyclic graph (DAG).For such a graph, every node has well defined "parents" and the joint distribution is the product of "local conditional distributions"

$$P(R,S,G) = P(R)\,P(S|R)\,P(G|S,R)$$

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

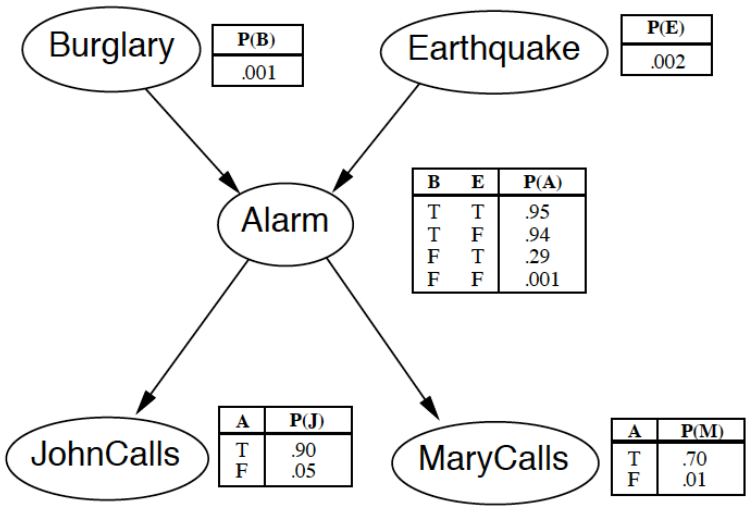


RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

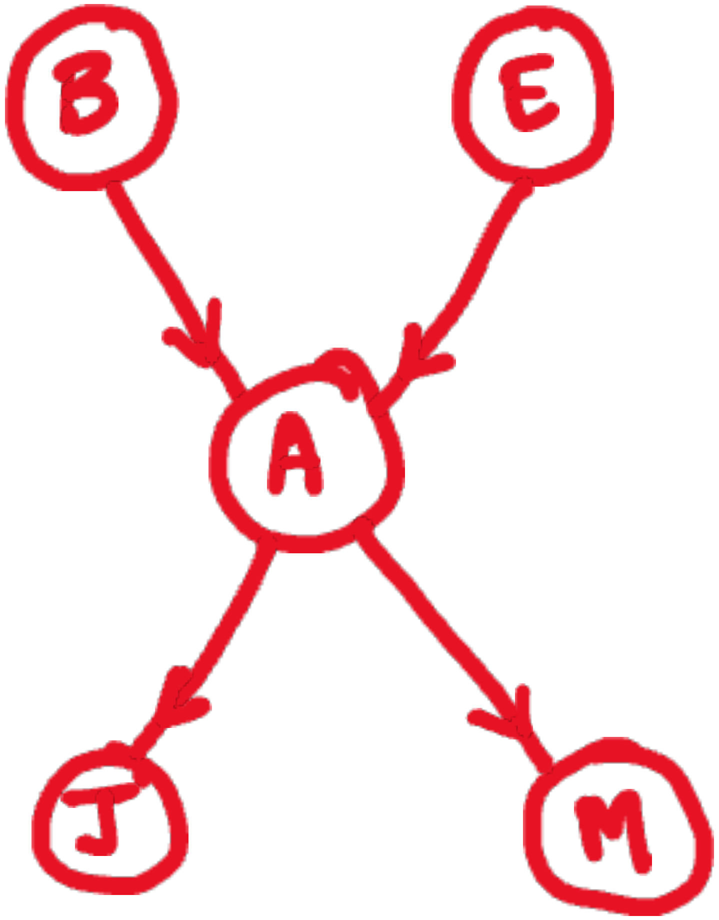
Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$
Network topology reflects "causal" knowledge:



Note: $\leq k$ parents $\Rightarrow$ $O(d^k n)$ numbers vs. $O(d^n)$

"Global" semantics defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \ldots, X_n) = \Pi_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i))$$



$$P(B, E, A, J, M) = P(B)\,P(E)\,P(A \mid B, E)\,P(J \mid A)\,P(M \mid A)$$

There are 2^5 entries in the joint probability distribution

This "factorized" representation makes it much more concise.

10 numbers instead of 31

Given the joint probability distribution we can answer various questions

- What is the probability that it is raining, given that the grass is wet?

$$\Pr(R = T \mid G = T) = \frac{\Pr(G = T, R = T)}{\Pr(G = T)} = \frac{\sum_{x \in \{T,F\}} \Pr(G = T, S = x, R = T)}{\sum_{x,y \in \{T,F\}} \Pr(G = T, S = x, R = y)}$$

$$\Pr(R=T \mid G=T) = \frac{\Pr(G=T, R=T)}{\Pr(G=T)} = \frac{\sum_{x\in\{T,F\}} \Pr(G=T, S=x, R=T)}{\sum_{x,y\in\{T,F\}} \Pr(G=T, S=x, R=y)}$$

We can calculate the probability of any case using the joint probability distribution e.g.

$$\begin{aligned}\Pr(G=T, S=T, R=T) &= \Pr(G=T \mid S=T, R=T)\Pr(S=T \mid R=T)\Pr(R=T)\\ &= 0.99 \times 0.01 \times 0.2\\ &= 0.00198.\end{aligned}$$

Then the numerical results (subscripted by the associated variable values) are

$$\Pr(R=T \mid G=T) = \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} = \frac{891}{2491} \approx 35.77\%.$$

The Weather-Ice Cream HMM
(Source: Jurafsky HMM handout)

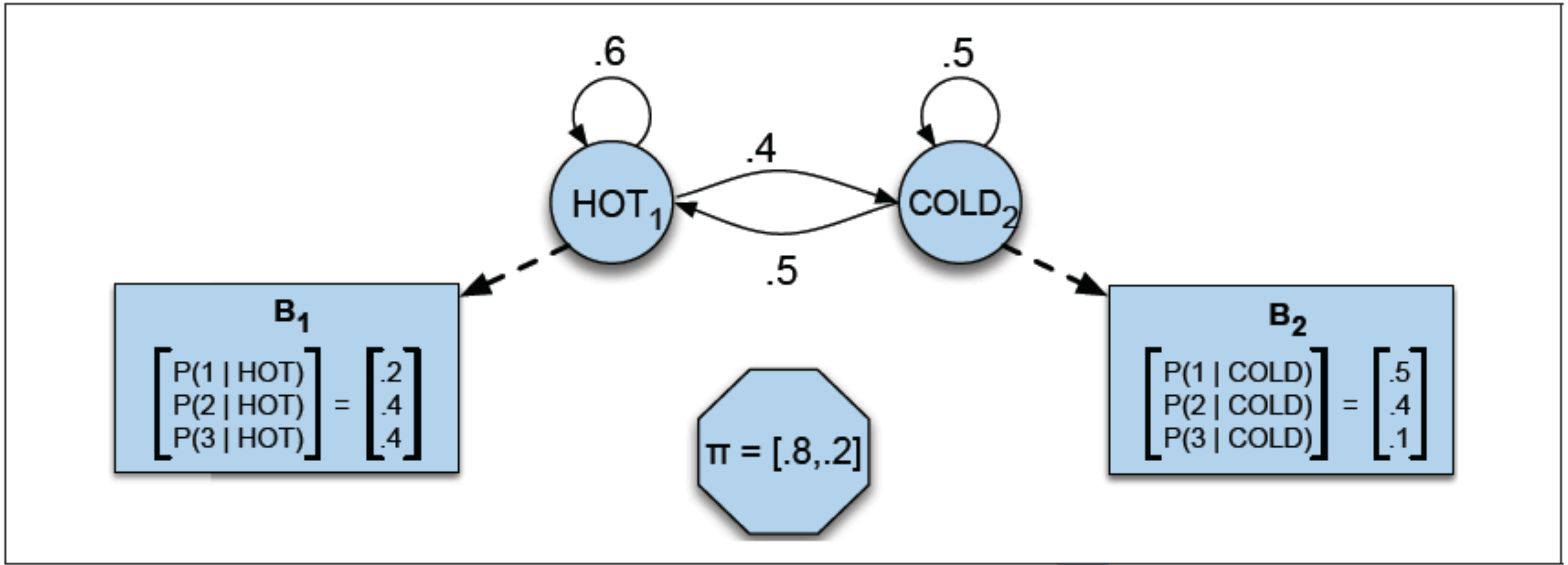


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

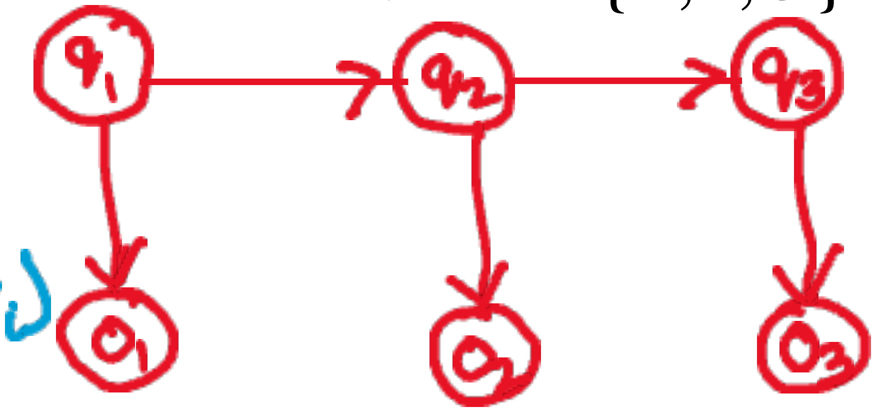
This is a stochastic automaton, not a DAG, but we can rewrite it as a DAG

DAG representation for the weather-ice cream model

- We use q_1, q_2, q_3 to denote the hidden states on day 1, 2, 3 etc.
- We use o_1, o_2, o_3 to denote the observations on day 1, 2, 3 etc.
- The q_i can take one of two values $\{\text{hot}, \text{cold}\}$
- The o_i can be one of three values $\{1,2,3\}$ (number of ice − creams)

Joint distribution

$$\prod_{i=1}^{n} P(x_i \mid \text{Parents}(x_i))$$



$$P(q_1, q_2, q_3, o_1, o_2, o_3) = P(q_1)\, P(q_2 \mid q_1)\, P(q_3 \mid q_2)\, P(o_1 \mid q_1)\, P(o_2 \mid q_2)\, P(o_3 \mid q_3)$$

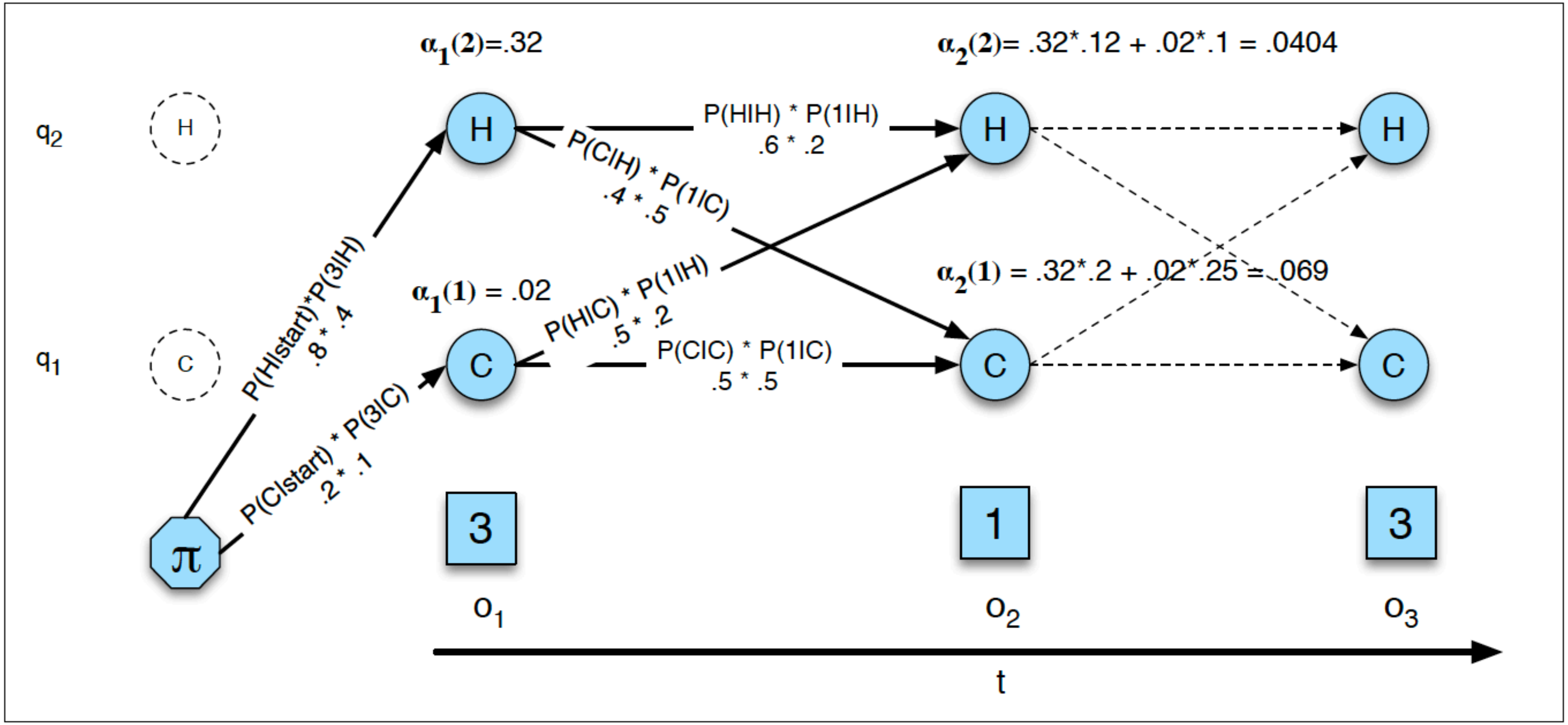


Figure A.5 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.12: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq. A.11: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$.

The α update algorithm

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

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

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

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

The Viterbi Algorithm: Sum replaced by Max

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

Note that we represent the most probable path by taking the maximum over all possible previous state sequences $\max_{q_1,\ldots,q_{t-1}}$. 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

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

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

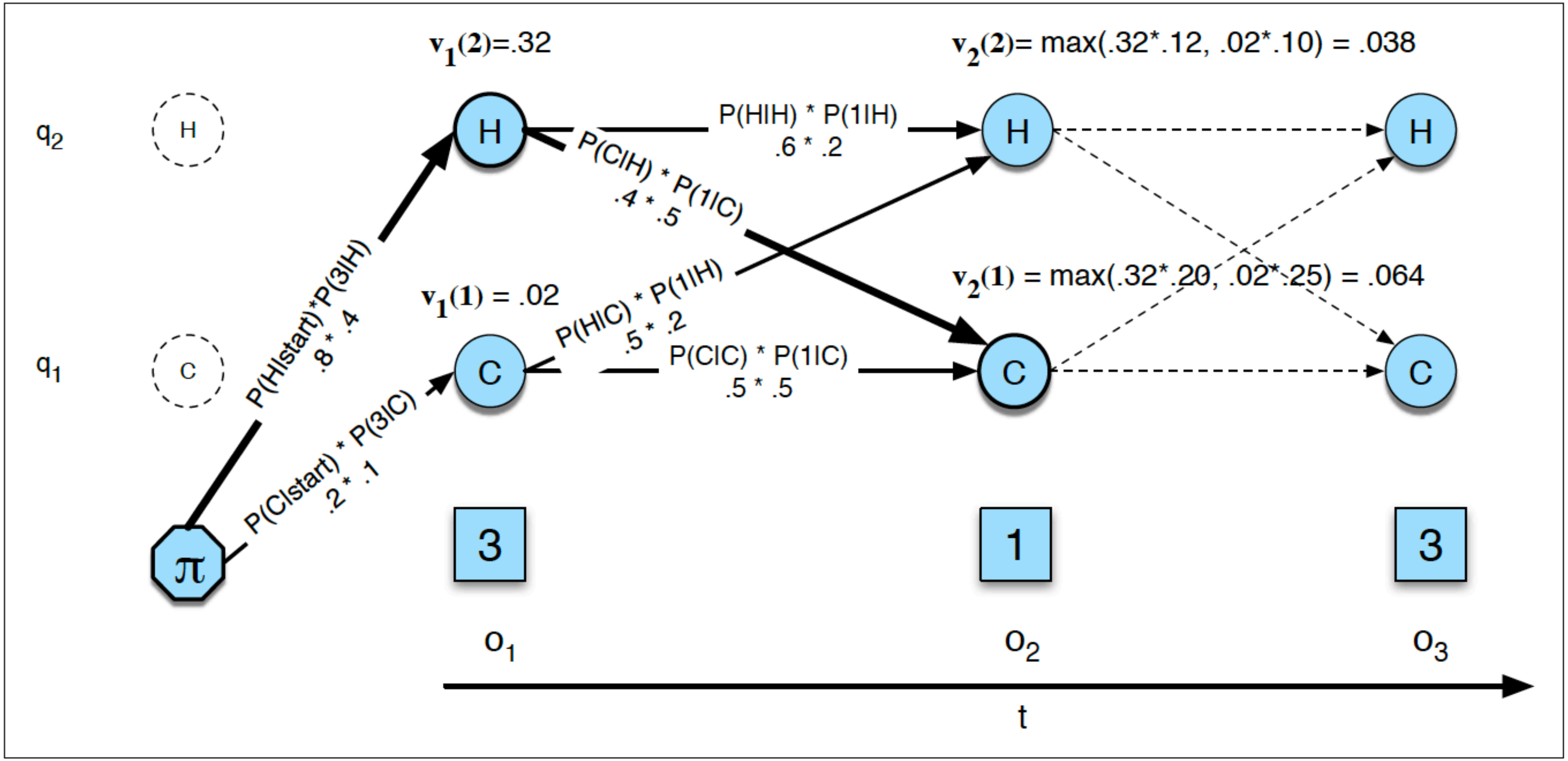


Figure A.8 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states 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: $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$.

The Viterbi backtrace

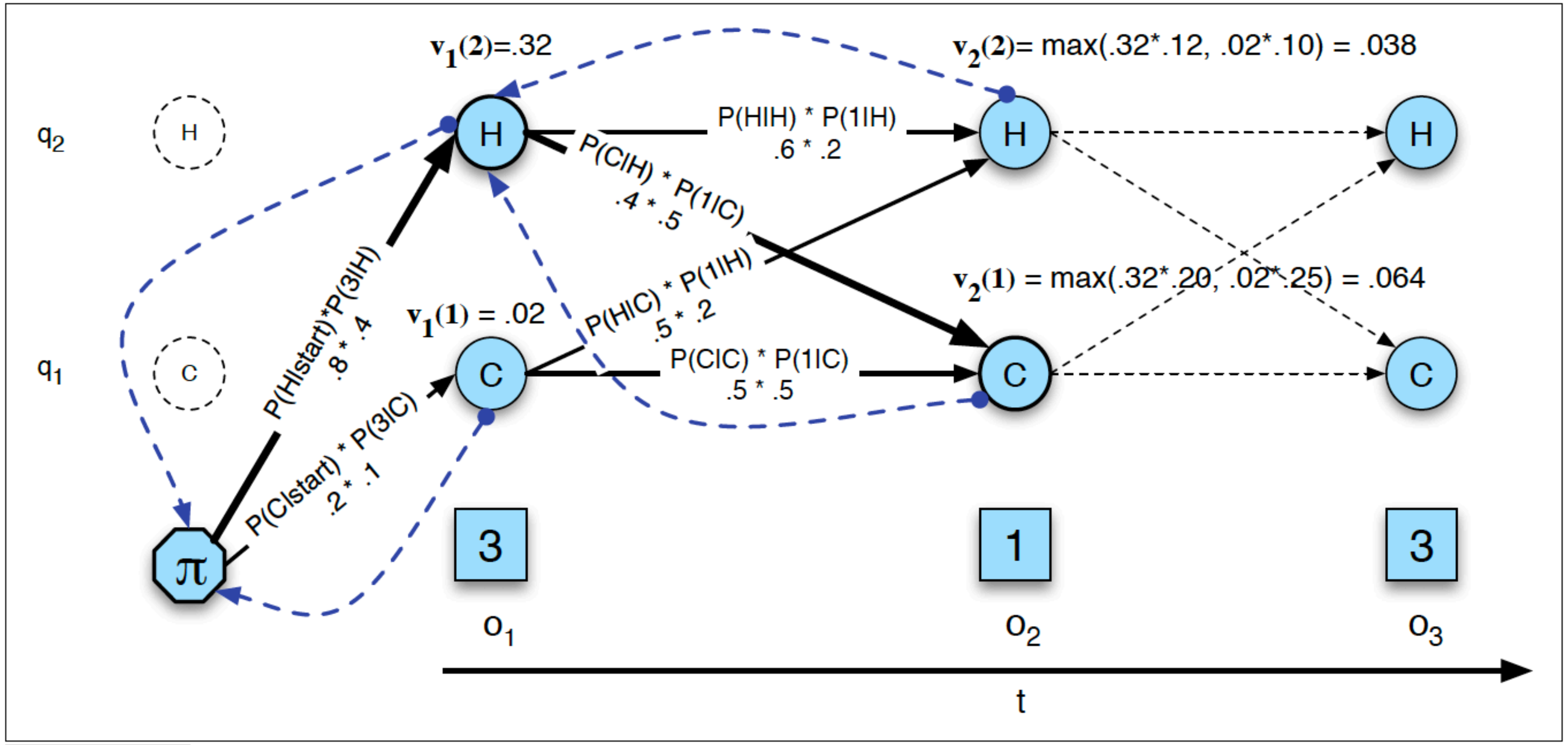


Figure A.10 The Viterbi backtrace. As we extend each path to a new state account for the next observation, we keep a backpointer (shown with broken lines) to the best path that led us to this state.