Ch3-p6: Hidden Markov Models

- Markov Chains

  - Goal: make a sequence of decisions

    - Processes that unfold in time, states at time *t* are influenced by a state at time *t*-1

    - Applications: speech recognition, gesture recognition, parts of speech tagging and DNA sequencing,

    - Any temporal process without memory
      $\boldsymbol{\omega}^T = \{\omega(1), \omega(2), \omega(3), \ldots, \omega(T)\}$ sequence of states
      We might have $\boldsymbol{\omega}^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$

    - The system can revisit a state at different steps and not every state need to be visited

- First-order Markov models
  - Our productions of any sequence is described by the transition probabilities; the time-independent probability of having state ω_j at step $t + 1$ given that the state at time t was ω_i

$$P(\omega_j(t + 1) \mid \omega_i(t)) = a_{ij} \quad (a_{ij} \neq a_{ji}, \text{ in general})$$

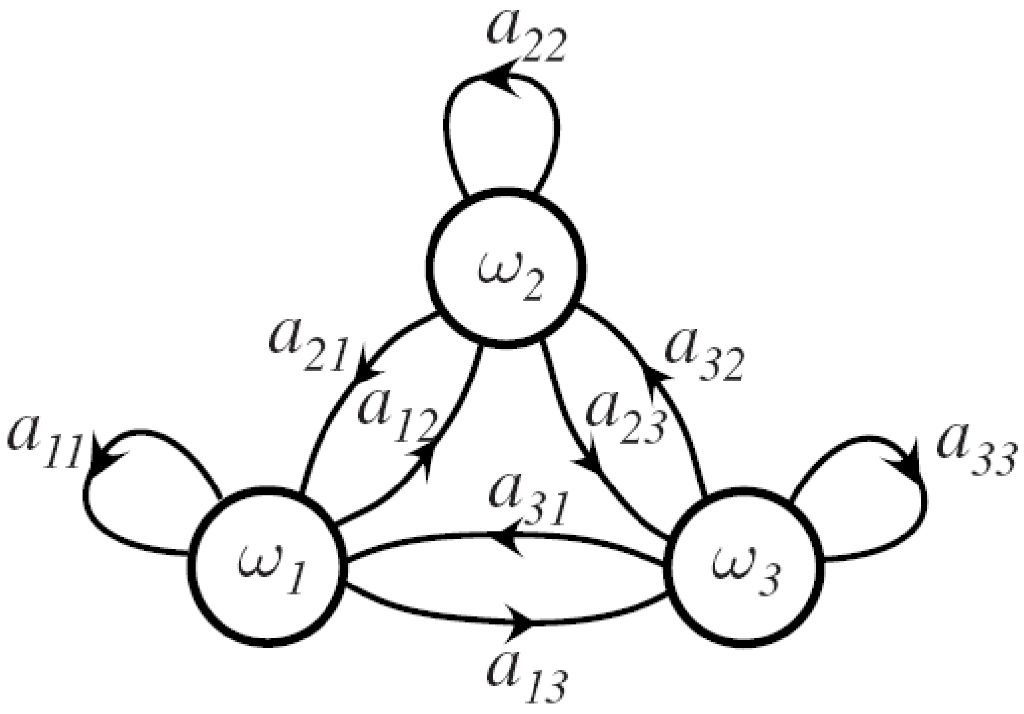


Figure 3.9: The discrete states, ω_i, in a basic Markov model are represented by nodes, and the transition probabilities, a_{ij} , by links. In a first-order discrete time Markov model, at any step t the full system is in a particular state $\omega(t)$. The state at step $t+1$ is a random function that depends solely on the state at step t and the transition probabilities.

We are given a particular model $\boldsymbol{\theta} = (a_{ij}, \boldsymbol{\omega}^T) \rightarrow$ $P(\boldsymbol{\omega}^6 \mid \boldsymbol{\theta}) = a_{14} \,.\, a_{42} \,.\, a_{22} \,.\, a_{21} \,.\, a_{14} \,.\, P(\omega(1) = \omega_1)$

$\boldsymbol{\omega}^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$

Example: speech recognition

"production of spoken words"

Production of the word: "pattern" represented by phonemes

/p/ /a/ /tt/ /er/ /n/ // (// = silent state)

Transitions from /p/ to /a/, /a/ to /tt/, /tt/ to er/, /er/ to /n/ and /n/ to a silent state

– *visible states: states* which are directly accessible to external measurement

Hidden Markov Model: Extension of Markov Chains

- We define particular sequence of such visible states as $\mathbf{V}^T = \{v(1), v(2), ..., v(T)\}$ and thus we might have $\mathbf{V}^6 = \{v_5, v_1, v_1, v_5, v_2, v_3\}$.

  Because we have access only to the visible states, while the ω_i are unobservable, such a full model is called a *Hidden Markov Model.*

  Thus, a hidden Markov model is a standard Markov process augmented by a set of measurable states and several probabilistic relations between those states and the hidden states.

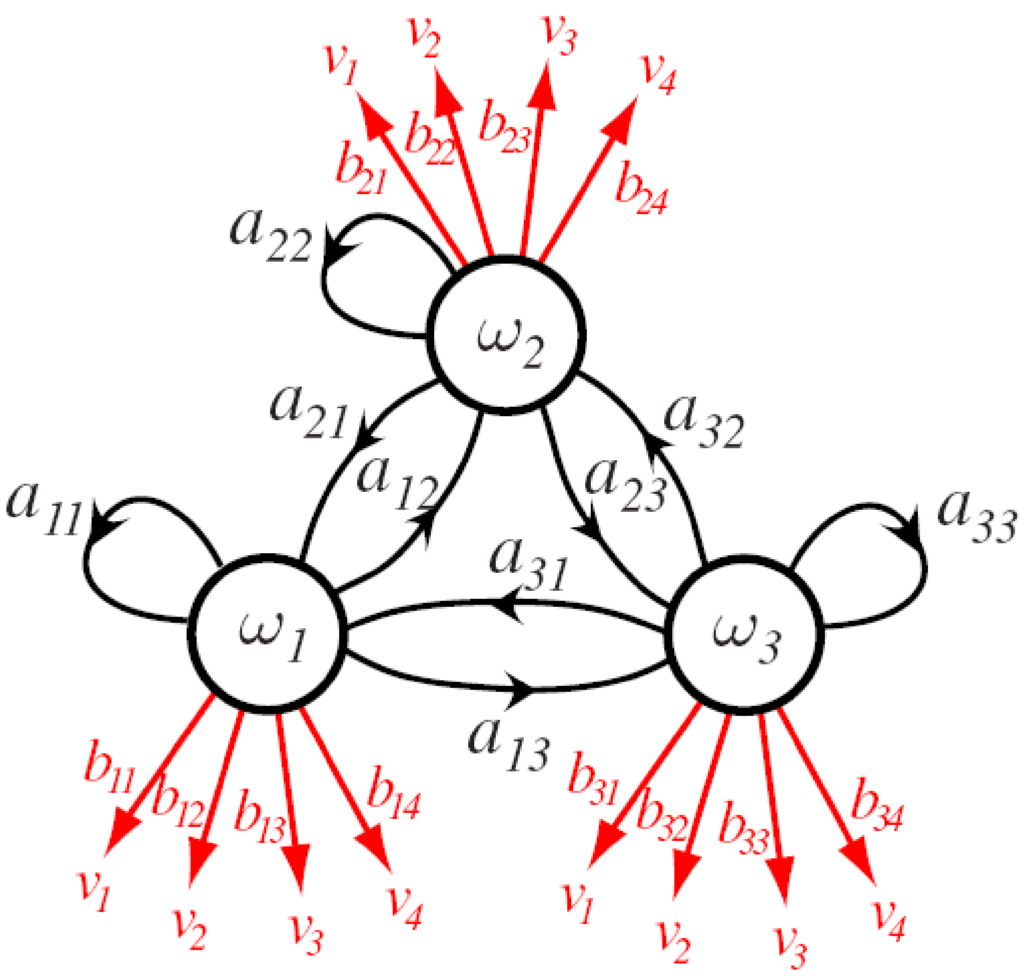


Figure 3.10: Three hidden units in an HMM and the transitions between them are shown in black while the visible states and the emission probabilities of visible states are shown in red. This model shows all transitions as being possible; in other HMMs, some such candidate transitions are not allowed.

Hidden Markov Model Computation

- Markov networks are strictly *causal* — the probabilities depend only upon *previous* states. A Markov model is called *ergodic* if every one of the states has a non-zero probability of occurring given some starting state.
- A final or absorbing state ω_0 is one which, if entered, is never left (i.e., $a_{00} = 1$).

$$\begin{aligned} a_{ij} &= P(\omega_j(t+1)|\omega_i(t)) \\ b_{jk} &= P(v_k(t)|\omega_j(t)). \end{aligned}$$

$$\begin{aligned}\sum_j a_{ij} &= 1 \text{ for all } i \text{ and}\\ \sum_k b_{jk} &= 1 \text{ for all } j,\end{aligned}$$

where the limits on the summations are over all hidden states and all visible symbols, respectively.

- 3 problems are associated with this model
  - The evaluation problem
  - The decoding problem
  - The learning problem

- **The Evaluation problem.** Suppose we have an HMM, complete with transition probabilities a_{ij} and b_{jk}. Determine the probability that a particular sequence of visible states $\mathbf{V}^T$ was generated by that model.
- **The Decoding problem.** Suppose we have an HMM as well as a set of observations $\mathbf{V}^T$. Determine the most likely sequence of *hidden* states $\boldsymbol{\omega}^T$ that led to those observations.
- **The Learning problem.** Suppose we are given the coarse structure of a model (the number of states and the number of visible states) but *not* the probabilities a_{ij} and b_{jk}. Given a set of training observations of visible symbols, determine these parameters.

• The evaluation problem

The probability that the model produces a sequence V^T of visible states is:

$$P(\mathbf{V}^T \mid \underset{\uparrow}{\boldsymbol{\Theta}}) = \sum_{r=1}^{r_{\max}} P(\mathbf{V}^T \mid \boldsymbol{\omega}_r^T) P(\boldsymbol{\omega}_r^T) \; ; \quad \mathrm{r}_{\max} = c^T$$

parameters

c # of hidden states

where each *r* indexes a particular sequence

$\boldsymbol{\omega}_r^T = \{\omega(1),\ \omega(2),\ \ldots, \omega(T)\}$ of *T* hidden states

(1) $P(\mathbf{V}^T \mid \boldsymbol{\omega}_r^T) = \prod_{t=1}^{t=T} P(v(t) \mid \omega(t))$ Conditional Independence

(2) $P(\boldsymbol{\omega}_{\mathrm{r}}^{\mathrm{T}}) = \prod_{t=1}^{t=T} P(\omega(t) \mid \omega(t-1))$ Makov chain of order 1

Using equations (1) and (2), we can write:

$$P(\mathbf{V}^T \mid \mathbf{\Theta}) = \sum_{r=1}^{r_{\max}} \prod_{t=1}^{t=T} P(v(t) \mid \omega(t))\, P(\omega(t) \mid \omega(t-1))$$

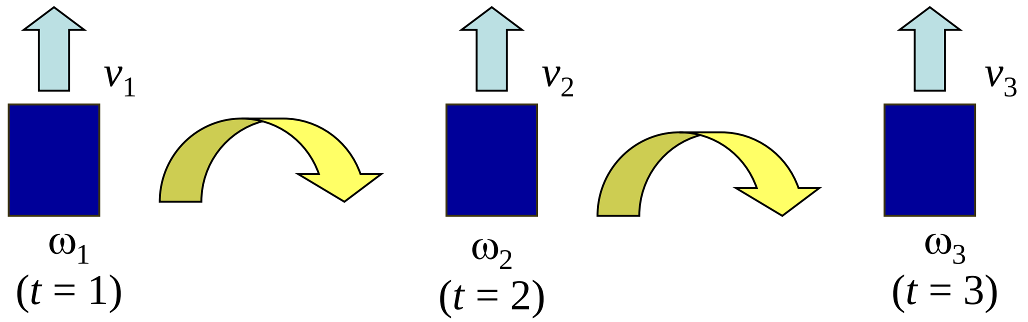
Interpretation: The probability that we observe the particular sequence of T visible states $\mathbf{V}^T$ is equal to the sum over all $r_{\max}$ possible sequences of hidden states of the conditional probability that the system has made a particular transition multiplied by the probability that it then emitted the visible symbol in our target sequence.

This is an $O(c^T T)$ calculation, which is quite prohibitive in practice. For instance, if $c = 10$ and $T = 20$, we must perform on the order of 10^{21} calculations!!!

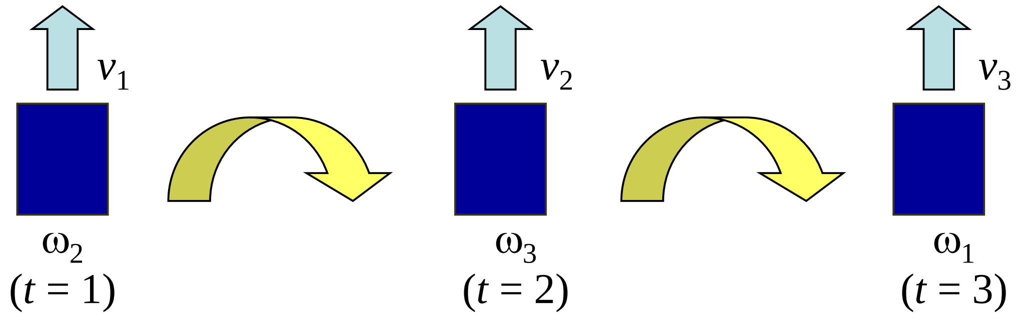
Example: Let ω_1, ω_2, ω_3 be the hidden states; v_1, v_2, v_3 be the visible states and $\mathbf{V}^3 = \{v_1, v_2, v_3\}$ is the sequence of visible states

$P(\{v_1, v_2, v_3\}|\Theta) = P(\omega_1).P(v_1 \mid \omega_1).P(\omega_2 \mid \omega_1).P(v_2 \mid \omega_2).P(\omega_3 \mid \omega_2).P(v_3 \mid \omega_3)$
$+P(\omega_2).P(v_1 \mid \omega_2).P(\omega_3 \mid \omega_2).P(v_2 \mid \omega_3).P(\omega_1 \mid \omega_3).P(v_3 \mid \omega_1)+\ldots+$ (possible terms in the sum = all possible ($3^3= 27$) cases !)

First possibility:



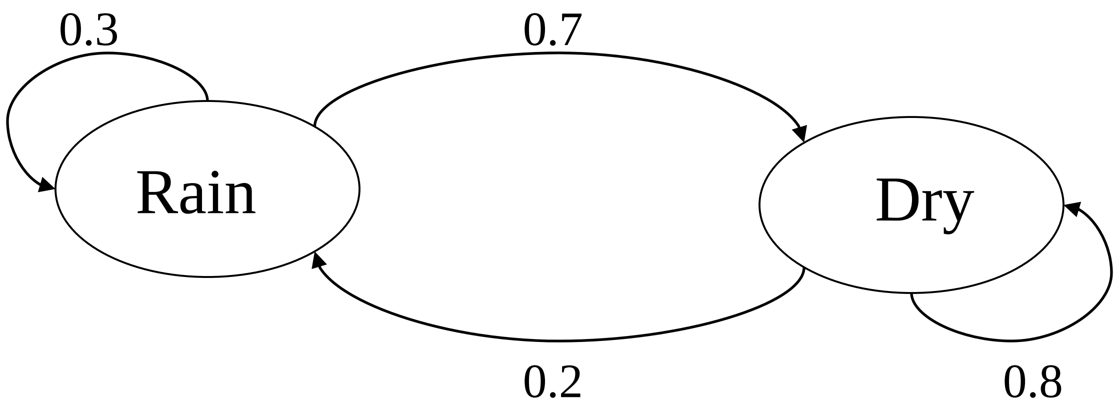
Second Possibility:



Therefore:

$$P(\{v_1, v_2, v_3\} \mid \Theta) = \sum_{\substack{\text{possible sequence} \\ \text{of hidden states}}} \prod_{t=1}^{t=3} P(v(t) \mid \omega(t)).P(\omega(t) \mid \omega(t-1))$$

Example of Markov Model



- Two states : ‘Rain’ and ‘Dry’.
- Transition probabilities: P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 , P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8
- Initial probabilities: say P(‘Rain’)=0.4 , P(‘Dry’)=0.6 .

Calculation of sequence probability

• By Markov chain property, probability of state sequence can be found by the formula:

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

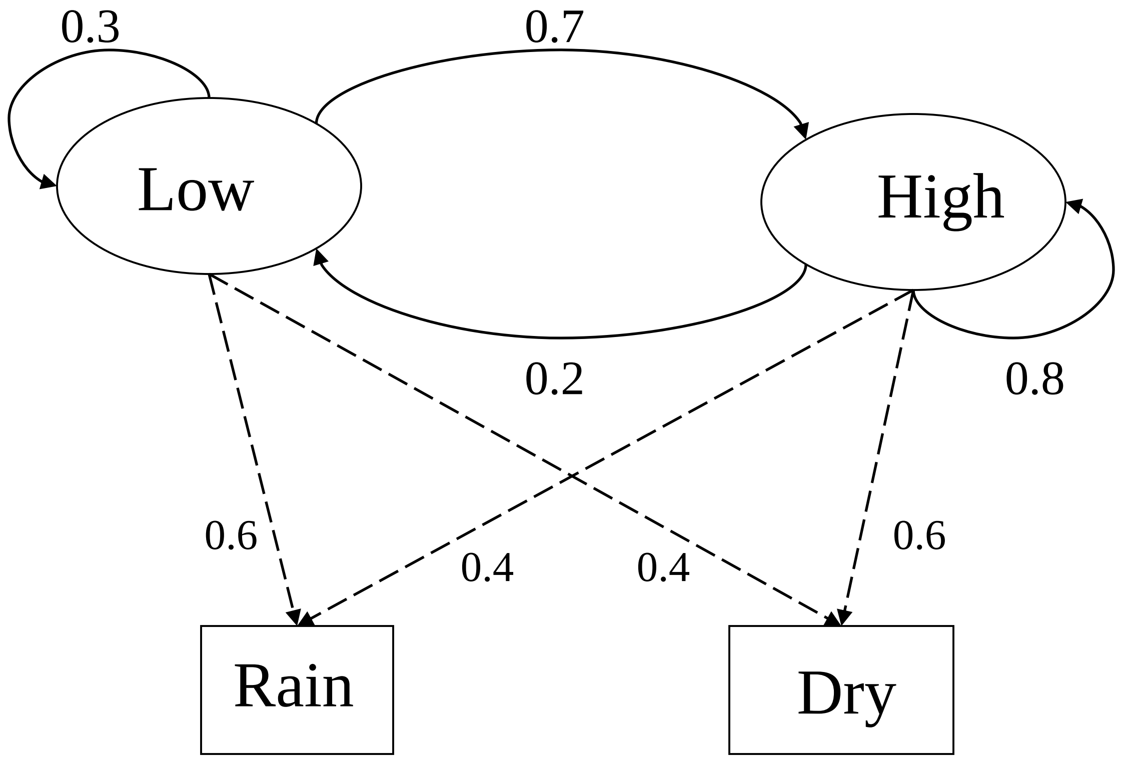
• Suppose we want to calculate a probability of a sequence of states in our example, {'Dry', 'Dry', 'Rain', 'Rain'}.

P({'Dry', 'Dry', 'Rain', 'Rain'}) =

P('Rain'|'Rain') P('Rain'|'Dry') P('Dry'|'Dry') P('Dry')=

$= 0.3 \times 0.2 \times 0.8 \times 0.6 = 0.0288$

Example of Hidden Markov Model



Example of Hidden Markov Model

- Two states : 'Low' and 'High' atmospheric pressure.
- Two observations : 'Rain' and 'Dry'.
- Transition probabilities: P('Low'|'Low')=0.3 , P('High'|'Low')=0.7 , P('Low'|'High')=0.2, P('High'|'High')=0.8
- Observation probabilities : P('Rain'|'Low')=0.6 , P('Dry'|'Low')=0.4 , P('Rain'|'High')=0.4 , P('Dry'|'High')=0.3 .
- Initial probabilities: say P('Low')=0.4 , P('High')=0.6 .

Calculation of observation sequence probability

• Suppose we want to calculate a probability of a sequence of observations in our example, {‘Dry’,’Rain’}.

• Consider all possible hidden state sequences:

P({‘Dry’,’Rain’}) = P({‘Dry’,’Rain’} , {‘Low’,’Low’}) + P({‘Dry’,’Rain’} , {‘Low’,’High’}) + P({‘Dry’,’Rain’} , {‘High’,’Low’}) + P({‘Dry’,’Rain’} , {‘High’,’High’})

where first term is :

P({‘Dry’,’Rain’} , {‘Low’,’Low’})=

P({‘Dry’,’Rain’} | {‘Low’,’Low’}) P({‘Low’,’Low’}) =

P(‘Dry’|’Low’)P(‘Rain’|’Low’) P(‘Low’)P(‘Low’|’Low’)

$= 0.4\times0.4\times0.6\times0.4\times0.3=0.01152$

Efficient Calculations: Forward-Backward algorithms

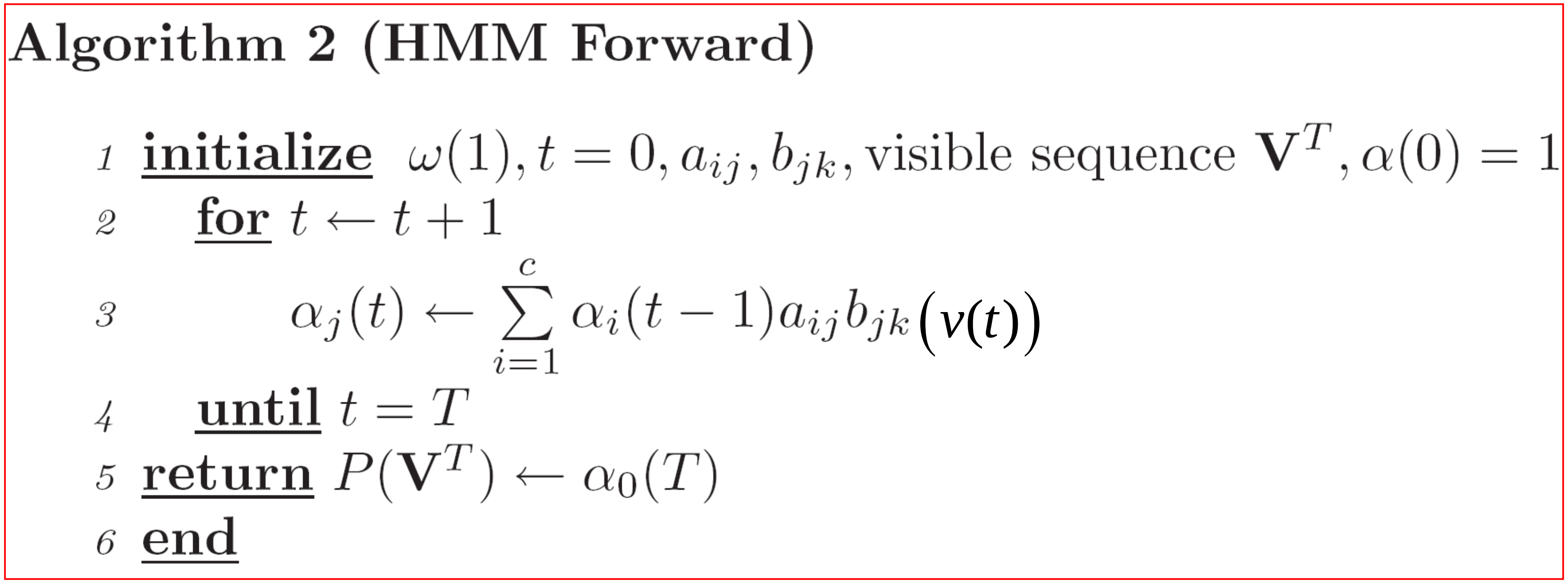
A computationally simpler algorithm for the same goal is as follows.

We can calculate P($\mathbf{V}^T$) recursively, since each term P(v(t)|ω(t))P(ω(t)|ω(t − 1)) involves only $v(t)$, ω(t) and ω(t − 1). We do this by defining forward variable $\alpha_j(t)$

$$\alpha_j(t) = \begin{cases} 0 & t = 0 \text{ and } j \neq \text{initial state} \\ 1 & t = 0 \text{ and } j = \text{initial state} \\ \left[\sum_i \alpha_i(t-1)a_{ij}\right] b_{jk}\left(v(t)\right) & \text{otherwise} \end{cases}$$

$b_{jk}(v(t))$ means the transition probability b_{jk} selected by the visible state emitted at time t. Thus the only non-zero contribution to the sum is for the index k which matches the visible state v(t).

Thus $\alpha_j(t)$ represents the probability that our HMM is in hidden state ω_j at step t having generated the first t elements of $\mathbf{V}^T$ i.e. $\alpha_j(\mathrm{t}) = P(v(1), v(2), \ldots, v(\mathrm{t}), \omega_j(\mathrm{t}) \mid \boldsymbol{\Theta})$



Algorithm 2 (HMM Forward)

```
initialize $\omega(1), t = 0, a_{ij}, b_{jk}$, visible sequence $\mathbf{V}^T, \alpha(0) = 1$
    for $t \leftarrow t + 1$
        $\alpha_j(t) \leftarrow \sum_{i=1}^{c} \alpha_i(t-1) a_{ij} b_{jk}(v(t))$
    until $t = T$
return $P(\mathbf{V}^T) \leftarrow \alpha_0(T)$
end
```

α_0 denotes the probability of the associated sequence ending to the known final state.

It is $O(c^2T)$. For c = 10, T = 20 $\longrightarrow$ 2000 calculations

Algorithm 3 (HMM Backward)

```
1 initialize $\omega(T), t = T, a_{ij}, b_{jk}$, visible sequence $V^T$
2     for $t \leftarrow t - 1$;
4         $\beta_i(t) \leftarrow \sum_{j=1}^{C} \beta_j(t+1) a_{ij} b_{jk}(v(t+1))$
5     until $t = 1$
7 return $P(V^T) \leftarrow \beta_i(0)$ for the known initial state
8 end
```

where

$$\beta_i(t) = \begin{cases} 0 & \omega_i(t) \neq \text{sequence's final state and } t = T \\ 1 & \omega_i(t) = \text{sequence's final state and } t = T \\ \sum_j a_{ij} b_{jk} v(t+1) \beta_j(t+1) & \text{otherwise,} \end{cases}$$

$\beta_i(t) = P$(model will gen. the seq. from $t+1$ to T given $\omega_i(t)$)
$= P(v(t+1), v(t+2), \ldots, v(T) \mid \omega_i(t), \boldsymbol{\Theta})$

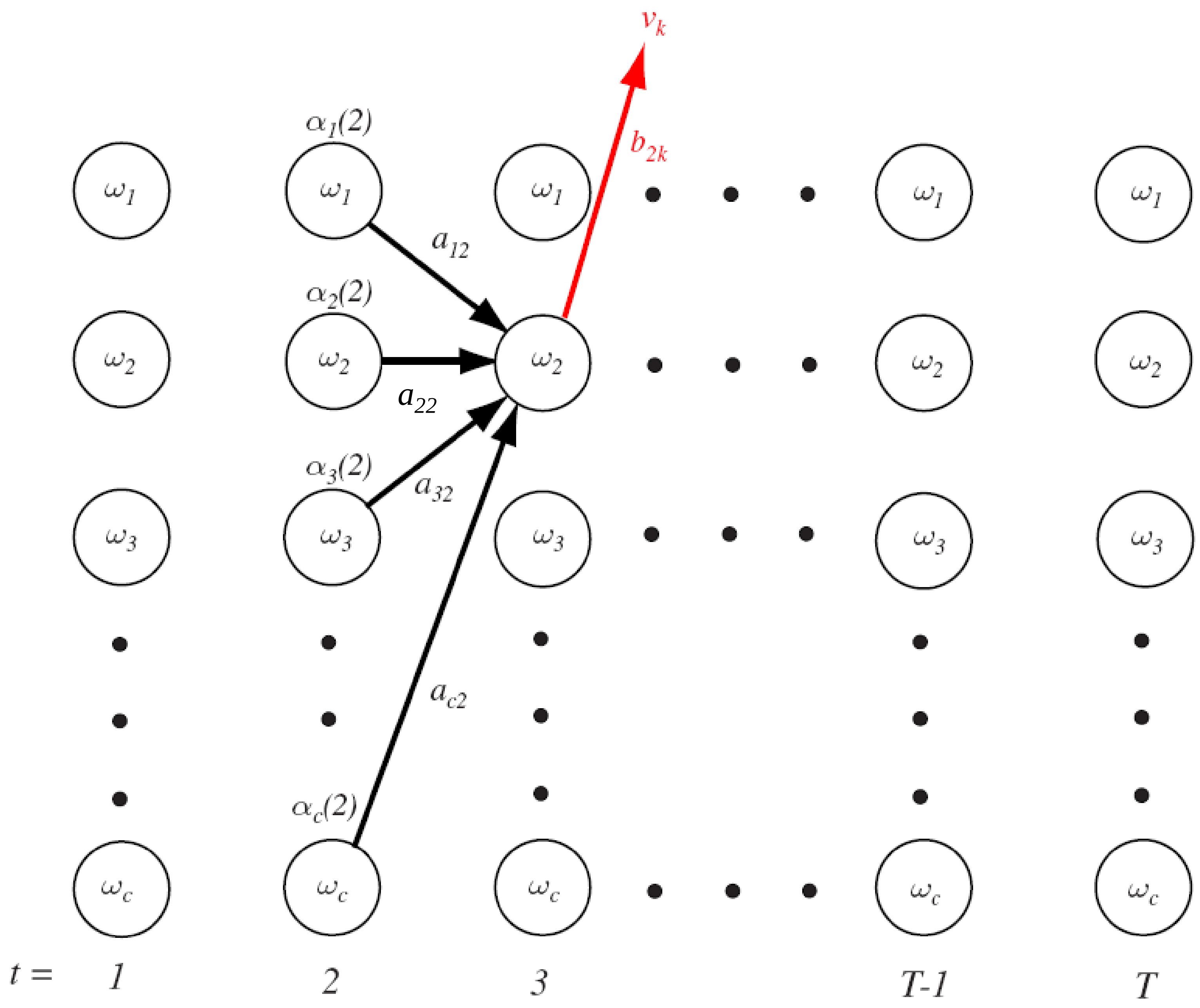

v_k
b_{2k}
$\alpha_1(2)$
$\alpha_2(2)$
$\alpha_3(2)$
$\alpha_c(2)$
a_{12}
a_{22}
a_{32}
a_{c2}
ω_1
ω_2
ω_3
ω_c
$t =$ 1 2 3 T-1 T

Figure 3.11: The computation of probabilities by the Forward algorithm can be visualized by means of a trellis — a sort of "unfolding" of the HMM through time. Suppose we seek the probability that the HMM was in state ω_2 at $t = 3$ and generated the observed visible up through that step (including the observed visible symbol v_k). The probability the HMM was in state ω_j ($t = 2$) and generated the observed sequence through $t = 2$ is $\alpha_j(2)$ for $j = 1, 2, ..., c$. To find $\alpha_2(3)$ we must sum these and multiply the probability that state ω_2 emitted the observed symbol v_k. Formally, for this particular illustration we have

$$\alpha_2(3) = b_{2k} \sum_{j=1}^{c} \alpha_j(2) a_{j2}.$$

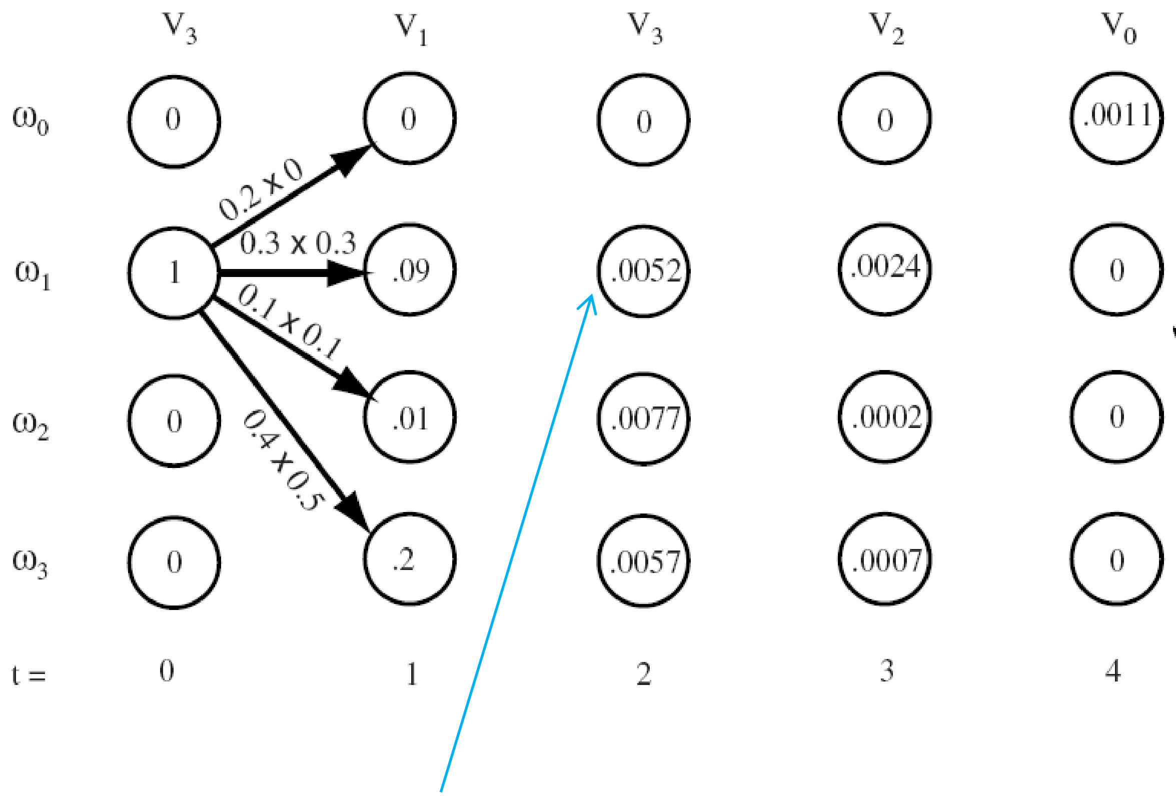
Example 4:Hidden Markov Model

Consider an HMM such as shown in Fig. 3.10, but with an explicit absorber state and unique null visible symbol v_0 with the following transition probabilities (where the matrix indexes begin at 0):

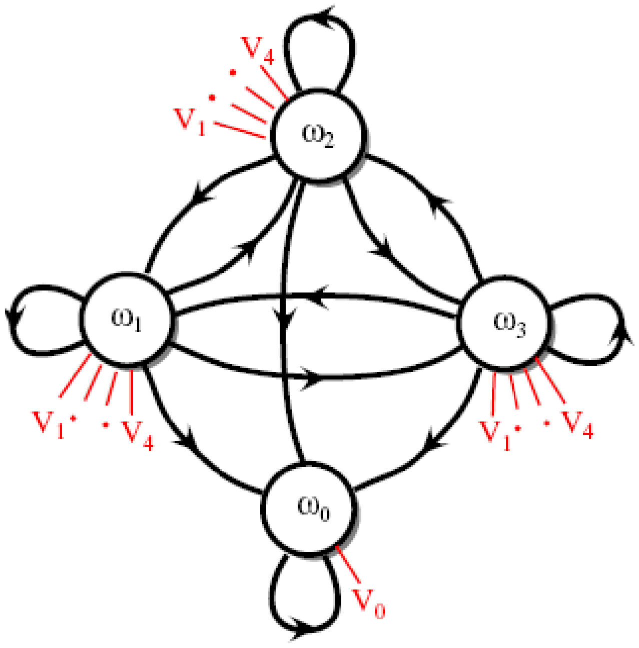
a_{ij}	$\square_0$	$\square_1$	$\square_2$	$\square_3$
$\square_0$	1	0	0	0
$\square_1$	0.2	0.3	0.1	0.4
$\square_2$	0.2	0.5	0.2	0.1
$\square_3$	0.8	0.1	0	0.1

b_{jk}	v_0	v_1	v_2	v_3	v_4
$\square_0$	1	0	0	0	0
$\square_1$	0	0.3	0.4	0.1	0.2
$\square_2$	0	0.1	0.1	0.7	0.1
$\square_3$	0	0.5	0.2	0.1	0.2

What is the probability it generates the particular sequence $\mathbf{V}^5 = \{v_3, v_1, v_3, v_2, v_0\}$?



v_3 v_1 v_3 v_2 v_0
ω_0: 0 0 0 0 .0011
ω_1: 1 .09 .0052 .0024 0
ω_2: 0 .01 .0077 .0002 0
ω_3: 0 .2 .0057 .0007 0
0.2 x 0
0.3 x 0.3
0.1 x 0.1
0.4 x 0.5
t = 0 1 2 3 4



$$\alpha_1(2) = b_{13}\sum_{i=0}^{3}\alpha_i(1)a_{i1} = 0.1(0\times 0 + 0.09\times 0.3 + 0.01\times 0.5 + 0.2\times 0.1) = 0.0052$$

Figure: The HMM (above) consists of four hidden states (one of which is an absorber state, ω_0), each emitting one of five visible states; only the allowable transitions to visible states are shown. The trellis for this HMM is shown below. In each node is α_i(t)—the probability the model generated the observed visible sequence up to *t*. For instance, we know that the system was in hidden state ω_1 at $t = 1$, and thus $\alpha_1(0) = 1$ and $\alpha_i(0) = 0$ for $i \neq 1$. The arrows show the calculation of $\alpha_i(1)$. For instance, since visible state v_1 was emitted at $t = 1$, we have $\alpha_0(1) = \alpha_1(0)a_{10}b_{10} = 1[0.2 \times 0] = 0$ as shown by the top arrow. Likewise the next highest arrow corresponds to the calculation $\square_1(1)=\alpha_1(0)a_{11}b_{11}=1[0.3\times0.3] = 0.09$. In this example, the calculation of $\alpha_i(1)$ is particularly simple, since only transitions from the known initial hidden state need be considered; all other transitions have zero contribution to $\alpha_i(1)$. For subsequent times, however, the calculation requires a sum over all hidden states at the previous time, as given by line 3 in the Forward algorithm. The probability shown in the final (absorbing) state gives the probability of the full sequence observed, $P(\mathbf{V}^T|\boldsymbol{\theta}) = 0.0011$.

If we denote our model - the *a*'s and *b*'s - by $\boldsymbol{\theta}$, we have by Bayes' formula

$$P(\boldsymbol{\theta}|\mathbf{V}^T) = \frac{P(\mathbf{V}^T|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{V}^T)}$$

In HMM pattern recognition we would have a number of HMMs, one for each category and classify a test sequence according to the model with the highest probability.

The Forward algorithm gives us $P(\mathbf{V}^T|\boldsymbol{\theta})$.

The prior probability of the model, $P(\boldsymbol{\theta})$, is given by some external source, such as a *language model* in the case of speech.

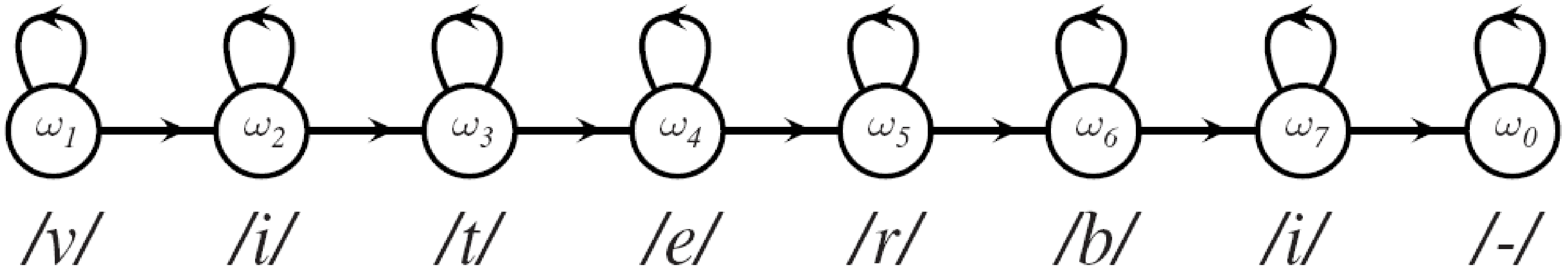


Figure 3.12: A left-to-right HMM commonly used in speech recognition. For instance, such a model could describe the utterance "viterbi," where ω_1 represents the phoneme /v/, □$_2$ represents /i/, ..., and □$_0$ a final silent state. Such a left-to-right model is more restrictive than the general HMM in Fig. 3.11, and preclude transitions "back" in time.

Normalization is Important

- **Normalization is required to avoid such recursive algorithms from accumulating large amounts of computational noise.**

- **We can apply a normalization factor at each step of the calculation:**

$$\alpha'_j(t) = \frac{\alpha_j(t)}{\prod_{i=0}^{t} Q_i}$$

where the scale factor, Q, is given by:

$$Q_i = \sum_{i=1}^{c} \alpha'_i(t) \quad Q_0 = 1$$

- **This is applied once per state per unit time, and simply involves scaling the current α's by their sum at each epoch (e.g., a frame).**

- **Also, likelihoods tend to zero as time increases and can cause underflow. Therefore, it is more common to operate on log probabilities to maintain numerical precision . This converts products to sums but still involves essentially the same algorithm (though an approximation for the log of a sum is used to compute probabilities involving the summations).**

Classification Using HMMs

If we concatenate our HMM parameters into a single vector, $\boldsymbol{\theta}$, we can write Bayes formula as:

$$P(\boldsymbol{\theta}|\mathbf{v}^T) = \frac{P(\mathbf{v}^T|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{v}^T)}$$

The forward algorithm gives us $P(\mathbf{v}^T|\boldsymbol{\theta})$

•We ignore the denominator term (evid.) during the maximization.

•In some applications, we use domain knowledge to compute P($\boldsymbol{\theta}$). For example, in speech recognition, this most often represents the probability of a word or sound, which comes from a "language model." It is also possible to use HMMs to model P($\boldsymbol{\theta}$) (e.g., statistical language modeling in speech recognition).

•In a typical classification application, there are a set of HMMs, one for each category, and the above calculation is performed for each model (θ_i).

• The decoding problem (optimal state sequence)

Given a sequence of visible states V^T, the decoding problem is to find the most probable sequence of hidden states, which is called maximum a posteriori (MAP) estimate of the state sequence.
This problem can be expressed mathematically as:

find the single “best” state sequence (hidden states)

$$\hat{\omega}(1), \hat{\omega}(2), \ldots, \hat{\omega}(T) \;\; such \;\; that:$$

$$\hat{\omega}(1), \hat{\omega}(2), \ldots, \hat{\omega}(T) =$$

$$\arg\max_{\omega(1),\omega(2),\ldots,\omega(T)} P\left[\omega(1), \omega(2), \ldots, \omega(T), v(1), v(2), \ldots, V(T) \,|\, \Theta\right]$$

$$\mathbf{v}^{T^*} = \arg\max_r P\left(\mathbf{v}^T \middle| \boldsymbol{\omega}_r^T\right) P\left(\boldsymbol{\omega}_r^T\right)$$

Note that the summation disappeared, since we want to find only one unique best case !

Where: $\lambda = \Theta = [\pi, \mathbf{A}, \mathbf{B}]$ is an HMM

$\pi = \mathrm{P}(\omega(1))$ (initial state probability)

$\mathbf{A} = a_{ij} = \mathrm{P}(\omega(\mathrm{t}+1) = j \mid \omega(\mathrm{t}) = i)$

$\mathbf{B} = b_{jk} = \mathrm{P}(v(\mathrm{t}) = k \mid \omega(\mathrm{t}) = j)$

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1c} \\ \vdots & \ddots & \vdots \\ a_{c1} & \dots & a_{cc} \end{bmatrix} \qquad a_{ij} = P\left(\omega_j(t+1) \middle| \omega_i(t)\right)$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & \dots & b_{1M} \\ \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots \\ b_{T1} & & a_{TM} \end{bmatrix} \qquad b_{ij} = P\left(v_j(t) \middle| \omega_i(t)\right)$$

Initial state distribution: $\pi(0) = \{\pi_1, \pi_2, \dots, \pi_c\}$ $\quad \pi_i = P(\omega_i \mid t = 0)$

π_i is the probability that the model is in state ω_i at the time t=0.

Algorithm 4: (HMM Decoding)

Viterbi Algorithm

```
begin initialize Path $\leftarrow \{\}$, $t \leftarrow 0$
        for $t \leftarrow t + 1$
            $j \leftarrow j + 1$
            for $j \leftarrow j + 1$
                $\alpha_j(t) \leftarrow b_{jk}v(t) \sum_{i=1}^{c} \alpha_i(t-1)a_{ij}$
            until $j = c$
            $j' \leftarrow \arg\max_j \alpha_j(t)$
            Append $\omega_{j'}$ to Path
        until $t = T$
    return Path
end
```

$O(c^T T)$ calculation $\rightarrow O(c^2 T)$

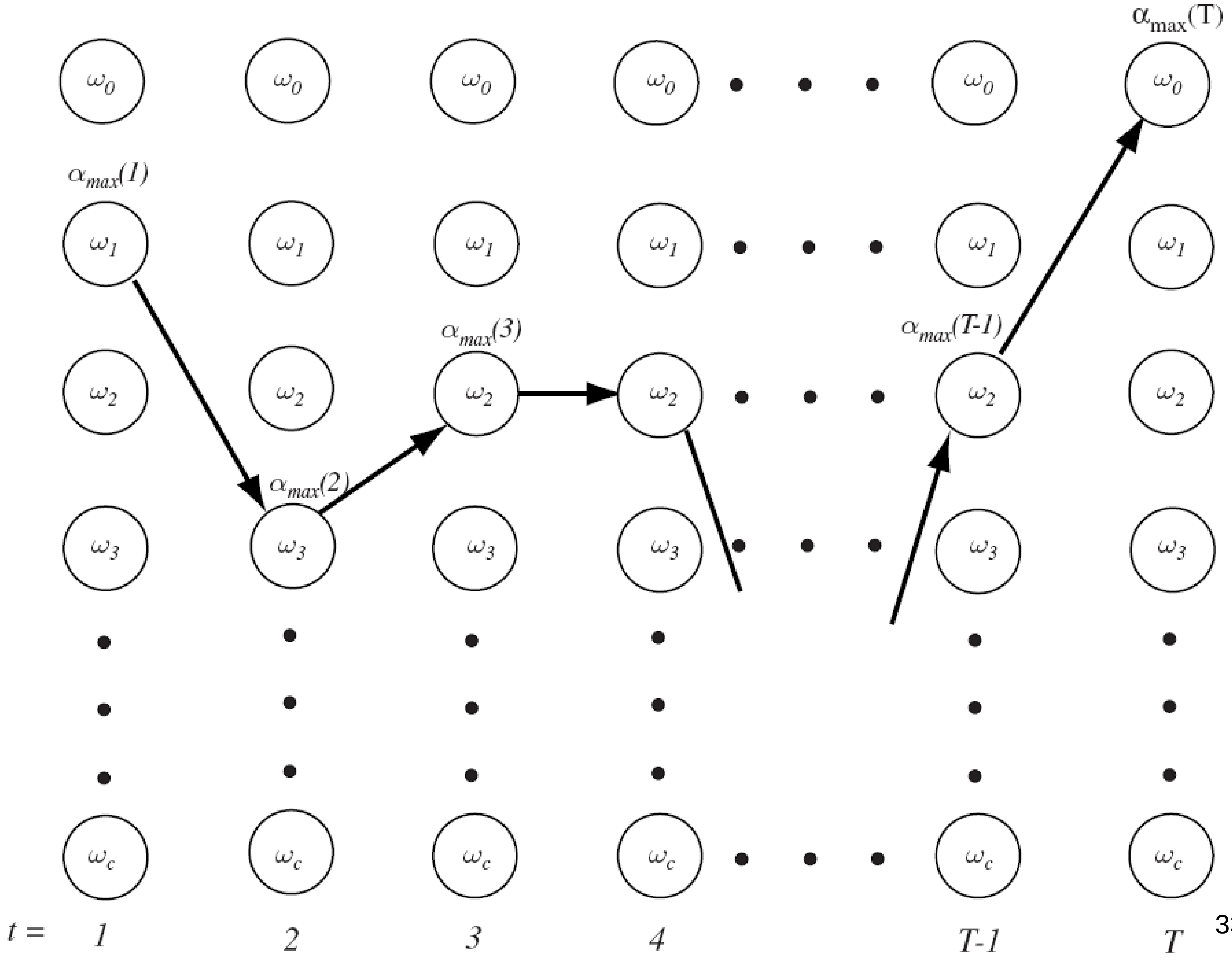
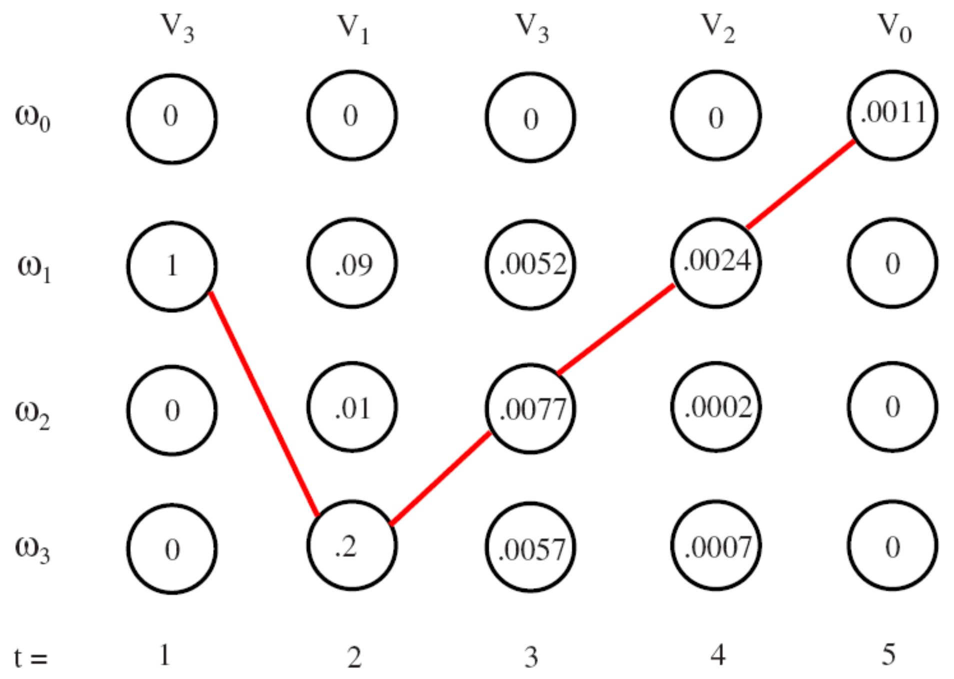

$\alpha_{max}(T)$
$\alpha_{max}(1)$
$\alpha_{max}(2)$
$\alpha_{max}(3)$
$\alpha_{max}(T-1)$
ω_0
ω_1
ω_2
ω_3
ω_c
$t =$ 1 2 3 4 T-1 T

FIGURE 3.13. Viterbi Decoding Trellis: The decoding algorithm finds at each time step *t* the state that has the highest probability of having come from the previous step and generated the observed visible state v_k. The full path is the sequence of such states. Because this is a local optimization (dependent only upon the single previous time step, not the full sequence), the algorithm does not guarantee that the path is indeed allowable. For instance, it might be possible that the maximum at $t = 5$ is ω_1 and at t=6 is ω_2, and thus these would appear in the path. This can even occur if $a_{12} = P(\omega_2 (t + 1)|\omega_1 (t)) = 0$, precluding that transition.

Example 5:HMM decoding

We find the path for the data of Example 4 for the sequence $\{\omega_1, \omega_3, \omega_2, \omega_1, \omega_0\}$. Note especially that the transition from ω_3 to ω_2 is not allowed according to the transition probabilities a_{ij} given in Example 4. The path *locally* optimizes the probability through the trellis.



- **The learning problem (parameter estimation)**

This third problem consists of determining a method to adjust the model parameters $\mathbf{\Theta} = [\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}]$ to satisfy a certain optimization criterion. We need to find the best model

$$\hat{\mathbf{\Theta}} = [\hat{\boldsymbol{\pi}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}]$$

Such that to maximize the probability of the observation sequence:

$$\underset{\mathbf{\Theta}}{Max}\, P(\mathbf{V}^T \mid \mathbf{\Theta})$$

We use an iterative procedure such as Baum-Welch (Forward-Backward) or Gradient to find this local optimum

- $\alpha_i(t)$= P(model generates visible sequence up to step t given hidden state $\omega_i(t)$)
- $\beta_i(t)$= P(model will generate the sequence from t+1 to T given $\omega_i(t)$)

$$\beta_i(t) = \begin{cases} 0 & \omega_i(t) \neq \text{sequence's final state and } t = T \\ 1 & \omega_i(t) = \text{sequence's final state and } t = T \\ \sum_j a_{ij} b_{jk} v(t+1) \beta_j(t+1) & \text{otherwise,} \end{cases}$$

By the definition of $\beta_i(T)$ in the above Eq., this will be either 0 (if $\omega_i(T)$ is not the final hidden state) or 1 (if it is). Thus it is clear that $\beta_i(T-1) = \sum_j a_{ij} b_{jk} v(T) \beta_j(T)$.

But the $\alpha_i(t)$ and $\beta_i(t)$ we determined are merely estimates of their true values, since we don't know the actual value of the transition probabilities a_{ij} and b_{jk} in the above Eq.

We now define a posteriori probability $\gamma_i(t) = p\left(\omega_i(t) \mid \mathbf{V}, \boldsymbol{\theta}\right)$

which is the probability of being in state ω_i at time t for the state sequence **V**. Note that:

$$\gamma_i(t) = p\left(\omega_i(t) \mid \mathbf{V}, \boldsymbol{\theta}\right) = \frac{p\left(\mathbf{V}, \omega_i(t) \mid \boldsymbol{\theta}\right)}{p\left(\mathbf{V} \mid \boldsymbol{\theta}\right)}$$

$$= \frac{p\left(\mathbf{V}, \omega_i(t) \mid \boldsymbol{\theta}\right)}{\sum_{j=1}^{c} p\left(\mathbf{V}, \omega_j(t) \mid \boldsymbol{\theta}\right)}$$

Also note that because of Markovian conditional independence:

$$\alpha_i(t)\beta_i(t) = p\left(v_1, v_2, \ldots, v_t, \omega_i(t) \,|\, \boldsymbol{\theta}\right) \times$$
$$p\left(v_{t+1}, v_{t+2}, \ldots, v_T \,|\, \omega_i(t), \boldsymbol{\theta}\right) = p\left(\mathbf{V}, \omega_i(t) \,|\, \boldsymbol{\theta}\right)$$

so we can define $\gamma_i(t)$ in term of $\alpha_i(t)$ and $\beta_i(t)$ as:

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{c} \alpha_j(t)\beta_j(t)}$$

We also define $\xi_{ij}(t) = p\left(\omega_i(t), \omega_j(t+1) \,|\, \mathbf{V}, \boldsymbol{\theta}\right)$

which is the probability of being in state ω_i at time t and being in state ω_j at time $t+1$ given the model generated the entire training sequence $\mathbf{V}^T$ by any path.

This can also be expanded as:

$$\xi_{ij}(t) = \frac{p\left(\omega_i(t), \omega_j(t+1), \mathbf{V} \mid \boldsymbol{\theta}\right)}{p\left(\mathbf{V} \mid \boldsymbol{\theta}\right)}$$

$$= \frac{\alpha_i(t) a_{ij} b_{jk}(v_k(t+1)) \beta_j(t+1)}{\sum_{j=1}^{c} \sum_{i=1}^{c} \alpha_i(t) a_{ij} b_{jk}(v_k(t+1)) \beta_j(t+1)}$$

or as:

$$\xi_{ij}(t) = \frac{p\left(\omega_i(t) \mid \mathbf{V}\right) p\left(v_{t+1}, v_{t+2}, \ldots, v_T, \omega_j(t+1) \mid \omega_i(t), \boldsymbol{\theta}\right)}{p\left(v_{t+1}, v_{t+2}, \ldots, v_T \mid \omega_i(t), \boldsymbol{\theta}\right)}$$

$$= \frac{\gamma_i(t) a_{ij} b_{jk}(v_k(t+1)) \beta_j(t+1)}{\beta_i(t)}$$

$$\gamma_i(t) = \sum_{j=1}^{c} \xi_{ij}(t)$$

If we sum these quantities across time, we can get some useful values i.e., the expression $\sum_{t=1}^{T} \gamma_i(t)$

is the expected number of times in state ω_i and therefore is the expected number of transitions away from state ω_i. Similarly,

$$\sum_{t=1}^{T-1} \xi_{ij}(t)$$

is the expected number of transitions from state ω_i to state ω_j for **V**. These follow from the fact that

$$\sum_{t=1} \gamma_i(t) = \sum_t E\left[I_t(i)\right] = E\left[\sum_t I_t(i)\right]$$

$$\sum_{t=1} \xi_{ij}(t) = \sum_t E\left[I_t(i,j)\right] = E\left[\sum_t I_t(i,j)\right]$$

Where $I_t(i)$ is an indicator random variable that is 1 when we are in state ω_i at time t and $I_t(i,j)$ is a random variable that is 1 when we move from state ω_i to state ω_j after time t.

Thus $\hat{a}_{ij}$ (the estimate of the probability of a transition from $\omega_i(t-1)$ to $\omega_j(t)$) can be found by taking the ratio between the expected number of transitions from ω_i to ω_j and the total expected number of any transitions from ω_i.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

We can obtain an improved estimate $\hat{b}_{ij}$ by calculating the ratio between the frequency that any particular symbol v_k is emitted and that for any symbol.

$$\hat{b}_{ik} = \frac{\text{expected number of times in state } \omega_i \text{ and observe symbol } v_\text{k}}{\text{expected number of times in state } \omega_i}$$

$$\hat{b}_{ik} = \frac{\sum_{\substack{t=1 \\ v(t)=v_k}}^{T} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Then, we start with rough or arbitrary estimates of a_{ij} *and* b_{ik}, *calculate* improved estimates by above Eqs., and repeat until some convergence criterion is met.

- **Parameter Updates:**

  Baum-Welch Forward-Backward Algorithm

- Parameters Learning Algorithm

```
Begin initialize

 $a_{ij}$, $b_{ik}$, training sequence $V^T$, convergence criterion $\theta$, $z=0$
  Do $z \leftarrow z+1$
    compute $\hat{a}(z)$ from $a(z-1)$ and $b(z-1)$
        compute $\hat{b}(z)$ from $a(z-1)$ and $b(z-1)$
        $a_{ij}(z) \leftarrow \hat{a}_{ij}(z)$
        $b_{ik}(z) \leftarrow \hat{b}_{ik}(z)$
   Until $max_{i,j,k} \{a_{ij}(z)-a_{ij}(z-1), b_{jk}(z)-b_{ik}(z-1)\} < \theta$
   Return $a_{ij} \leftarrow a_{ij}(z)$; $b_{ik} \leftarrow b_{ik}(z)$
End
```

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

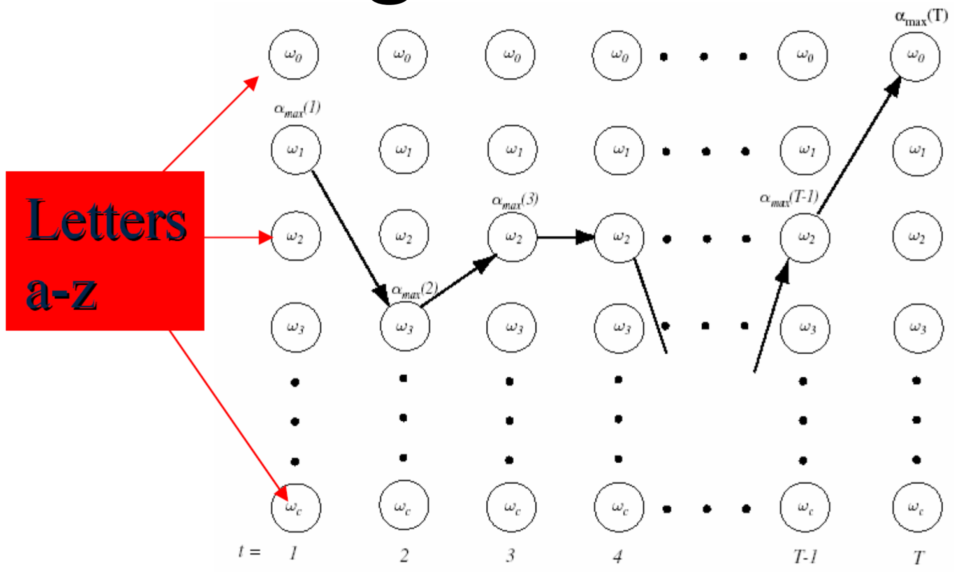
$$\hat{b}_{ik} = \frac{\sum_{\substack{t=1 \\ v(t)=v_k}}^{T} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Convergence

- Requires several presentations of each training sequence (fewer than 5 common in speech)
- Another stopping criterion:
  - Overall probability that learning model could have generated the training data
- Seeding the parameters with good initial guesses is very important.

HMM Word Recognition

- Two approaches:
  - HMM can model all possible words
  - Each state corresponds to each letter of alphabet
  - Letter transition probabilities are calculated for each pair of letters
  - Letter confusion probabilities are symbol probabilities
  - Decoding problem gives most likely word
- Separate HMMs are used to model each word
  - Evaluation problem gives probability of observation which is used as a class-conditional probability



- Each word, e.g., cat, dog, etc, has an associated HMM
- For a test utterance determine which model has highest probability
- HMMs for speech are left-to-right models
- HMM produces a class conditional class-probability