Chapter 4 Hidden Markov Model

First-Order Markov Model

Notations

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$: A set of c possible states
- $\boldsymbol{\omega}^T = \{\omega(1), \omega(2), \ldots, \omega(T)\}$: A state sequence of Length T where $w(t) \in \Omega$, $1 \leq t \leq T$

  e.g. $\omega^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$
- $\mathbf{A} = [a_{ij}]_{c \times c}$: The transition probability matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{c1} & \cdots & \cdots & a_{cc} \end{bmatrix}$$

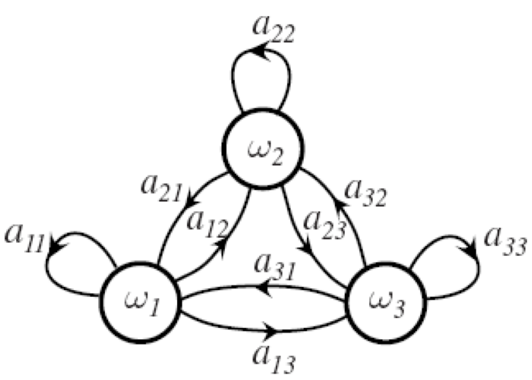
  where

$$\begin{aligned} a_{ij} &= P\left(\omega(t+1) = \omega_j \mid \omega(t) = \omega_i\right) \\ &= P\left(\omega_j \mid \omega_i\right) \end{aligned}$$

  probability of transferring from state ω_i to state ω_j is **time-independent**

$$\sum_{j=1}^{c} a_{ij} = 1$$

State transition diagram



$$\begin{aligned} P\left(\boldsymbol{\omega}^T\right) &= \prod_{t=1}^{T} P(\omega(t) \mid \omega(1), \ldots, \omega(t-1)) \\ &= \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1)) \end{aligned}$$

Hidden Markov Model (HMM)

Basic assumptions

- The state at each step is **invisible**
- The invisible state emits one **visible symbol** at each step

Notions

- $\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$: A set of K possible symbols
- $\mathbf{V}^T = \{v(1), v(2), \ldots, v(T)\}$: An observed symbol sequence of length T where $v(t) \in \mathcal{V}(1 \leq t \leq T)$
- $\mathbf{B} = [b_{jk}]_{c \times K}$: The observation symbol probability matrix

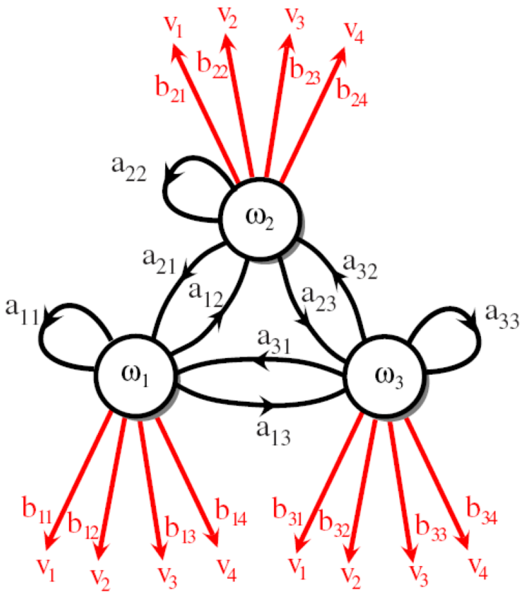
$$\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ \cdots & \cdots & \cdots & \cdots \\ b_{c1} & \cdots & \cdots & b_{cK} \end{bmatrix}$$

$$b_{jk} = P\left(v_k \mid \omega_j\right), \sum_{k=1}^{K} b_{jk} = 1$$

  which is the probability of emitting symbol v_k at state w_j

- $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_c)$: The initial state probability where $\pi_j = P\left(\omega(1) = \omega_j\right)$

State transition diagram



$$\begin{aligned} P\left(\mathbf{V}^T \mid \boldsymbol{\omega}^T\right) &= \prod_{t=1}^{T} P(v(t) \mid \omega(t)) \\ &= \prod_{t=1}^{T} b_{\omega(t) v(t)} \end{aligned}$$

Three central problems in HMM

$$\begin{aligned} &\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}: \text{ the complete set of H} \\ &\mathbf{V}^T : \text{ the observed symbol} \end{aligned}$$

Evaluation

Given $\boldsymbol{\theta}$, determine the probability of generating $\mathbf{V}^T$ **to evaluate** $P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)$

Decoding

Given $\boldsymbol{\theta}$ and V^T, determine the most likely hidden state sequence **to identify** $\boldsymbol{\omega}^T$ **which maximizes** $P\left(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta}\right)$

Learning

Given $\mathbf{V}^T$, determine model parameters $\boldsymbol{\theta}$ **to identify $\boldsymbol{\theta}$ which maximizes $P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)$**

Evaluation

HMM forward algorithm

Define $\alpha_j(t) = P\left(v(1), v(2), \ldots, v(t), \omega(t) = \omega_j \mid \boldsymbol{\theta}\right)$

which means the probability of being in hidden state ω_j at step t and having generated the first t symbols of $\mathbf{V}^T$

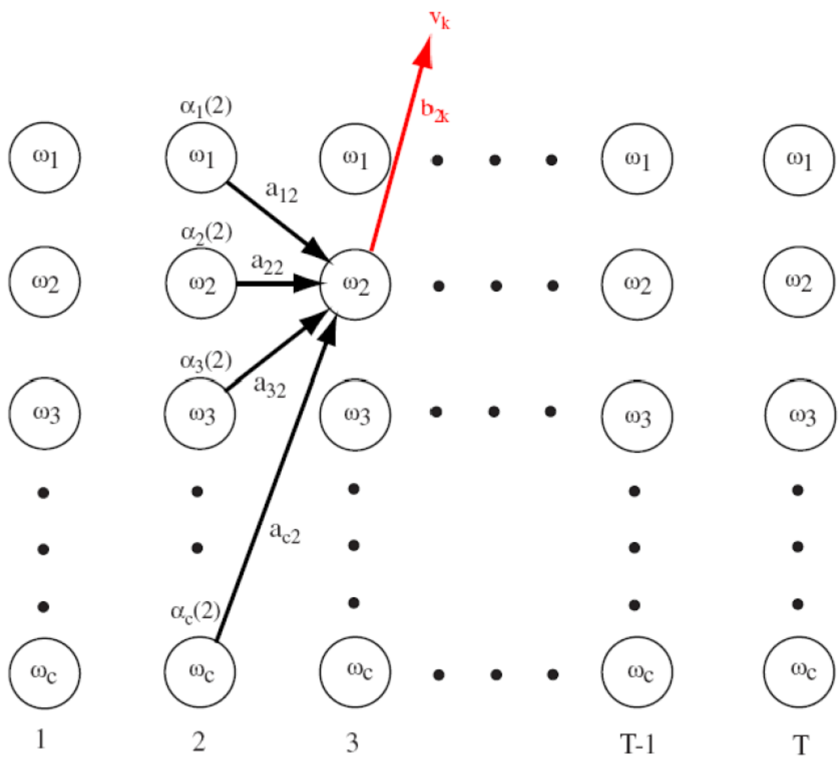
where the goal is $P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right) = \sum_{j=1}^{c} \alpha_j(T)$

then calculate **recursively**:

$$\begin{aligned}\alpha_j(1) = P\left(v(1), \omega(1) = \omega_j \mid \boldsymbol{\theta}\right) &= P\left(\omega(1) = \omega_j \mid \boldsymbol{\theta}\right) \cdot P\left(v(1) \mid \omega_j, \boldsymbol{\theta}\right) \\ &= \pi_j b_{jv(1)}\end{aligned}$$

$$\begin{aligned}\alpha_j(t) &= \sum_{i=1}^{c} P\left(v(1), \ldots, v(t-1), \omega(t-1) = \omega_i, v(t), \omega(t) = \omega_j \mid \boldsymbol{\theta}\right) \\ &= \sum_{i=1}^{c} P\left(v(1), \ldots, v(t-1), \omega(t-1) = \omega_i \mid \boldsymbol{\theta}\right) \cdot P\left(\omega_j \mid \omega_i, \boldsymbol{\theta}\right) \cdot P\left(v(t) \mid \omega_j, \boldsymbol{\theta}\right) \\ &= \left[\sum_{i=1}^{c} \alpha_i(t-1) a_{ij}\right] b_{jv(t)}\end{aligned}$$

A trellis diagram (网格图):



Pseudo-code for HMM forward algorithm

1. **Initialize** $t = 1$ and $\alpha_j(t) = \pi_j b_{jv(t)}$ $(1 \leq j \leq c)$
2. **For** $t = 2$ to T
3. **For** $j = 1$ to c
4. $$\alpha_j(t) = \left[\sum_{i=1}^{c} \alpha_i(t-1) a_{ij}\right] b_{jv(t)}$$
5. **End**
6. **End**
7. **Return** $P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \alpha_j(T)$

e.g.

An illustrative example

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_4\}$ $(c = 4)$ $\quad \mathcal{V} = \{v_1, v_2, \ldots, v_5\}$ $(K = 5)$

$$\mathbf{A} = [a_{ij}]_{c \times c} = \begin{pmatrix} \boxed{1 \quad 0 \quad 0 \quad 0} \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0.0 & 0.1 \end{pmatrix}$$

$$\mathbf{B} = [b_{jk}]_{c \times K} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{pmatrix}$$

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_c) = \boxed{(0, 1, 0, 0)}$$

Specific properties for $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$

- ω_1 can be viewed as an ***absorbing*** state, which won't transit to other states once entered
- at state ω_1, **only the symbol** v_1 **is emitted**
- at states other than ω_1, **the symbol** v_1 **won't be emitted**
- the **initial state** should be ω_2

The forward procedure for evaluating $P(\mathbf{V}^5 \mid \boldsymbol{\theta})$ with $\mathbf{V}^5 = \{v_4, v_2, v_4, v_3, v_1\}$

	v_4	v_2	v_4	v_3	v_1
ω_1	0	0	0	0	.00011
ω_2	0.1	.009	.00052	.00024	0
ω_3	0	.001	.00072	.00002	0
ω_4	0	.020	.00057	.00007	0
$t =$	1	2	3	4	5

The value of $\alpha_j(t)$ is shown in the circles of the trellis

$P(\mathbf{V}^5 \mid \boldsymbol{\theta})$

$= \sum_{j=1}^{4} \alpha_j(5)$

$= 0.00011$

HMM backward algorithm

Define $\beta_j(t) = P\left(v(t+1), v(t+2), \ldots, v(T) \mid \omega(t) = \omega_j, \boldsymbol{\theta}\right)$

which means the probability of observing the rest $T - t$ symbols in $\mathbf{V}^T$ given that the hidden state at step t is ω_j

where the goal is $P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right) = \sum_{j=1}^{c} \pi_j b_{jv(1)} \beta_j(1)$

then calculate **recursively**:

$$\beta_j(T) = 1$$

$$\begin{aligned}
\beta_j(t) &= \sum_{i=1}^{c} P\left(v(t+1), \omega(t+1) = \omega_i, v(t+2), \ldots, v(T) \mid \omega(t) = \omega_j, \boldsymbol{\theta}\right) \\
&= \sum_{i=1}^{c} P\left(v(t+2), \ldots, v(T) \mid \omega(t+1) = \omega_i, \boldsymbol{\theta}\right) \cdot P\left(\omega_i \mid \omega_j, \boldsymbol{\theta}\right) \cdot P\left(v(t+1) \mid \omega_i, \boldsymbol{\theta}\right) \\
&= \sum_{i=1}^{c} \beta_i(t+1) a_{ji} b_{iv(t+1)}
\end{aligned}$$

Pseudo-code for HMM backward algorithm

1. **Initialize** $t = T$ **and** $\beta_j(T) = 1 \ (1 \le j \le c)$
2. **For** $t = T - 1$ to 1
3.   **For** $j = 1$ to c
4. $\beta_j(t) = \sum_{i=1}^{c} \beta_i(t+1) a_{ji} b_{iv(t+1)}$
5.   **End**
6. **End**
7. **Return** $P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \pi_j b_{jv(1)} \beta_j(1)$

Decoding

the goal is to find $\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P\left(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta}\right) = \arg\max_{\boldsymbol{\omega}^T} P\left(\boldsymbol{\omega}^T, \mathbf{V}^T \mid \boldsymbol{\theta}\right) \cdot P(\mathbf{V} \mid \boldsymbol{\theta})$

because $P(\mathbf{V} \mid \boldsymbol{\theta})$ is a constant to ω, the goal is the same to find

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P\left(\boldsymbol{\omega}^T, \mathbf{V}^T \mid \boldsymbol{\theta}\right)$$

The Viterbi algorithm

Define $\delta_j(t) = \max_{\omega(1),...,\omega(t-1)} P(\omega(1), \ldots, \omega(t-1), \omega(t) = \omega_j, v(1), \ldots, v(t) \mid \boldsymbol{\theta})$

which means **the highest probability (best score)** of the state sequence and observed symbols **till step t**, where the state at step t is ω_j

(有关于$2t$个随机变量，将第t步的状态固定在ω_j，遍历前$t-1$个状态共c^{t-1}次，找到使上式最大的$\omega(1), \ldots, \omega(t-1)$组合)

First calculate the first step:

$$\begin{aligned}\delta_j(1) = P(\omega(1) = \omega_j, v(1) \mid \boldsymbol{\theta}) &= P(\omega(1) = \omega_j \mid \boldsymbol{\theta}) \cdot P(v(1) \mid \omega_j, \boldsymbol{\theta}) \\ &= \pi_j b_{jv(1)}\end{aligned}$$

then

$$\begin{aligned}\delta_j(t) &= \max_{\omega(1),...,\omega(t-1)} P(\omega(1), \ldots, \omega(t-1), \omega(t) = \omega_j, v(1), \ldots, v(t) \mid \boldsymbol{\theta}) \\ &= \max_{\omega(t-1)} \left[\max_{\omega(1),...,\omega(t-2)} P(\omega(1), \ldots, \omega(t-1), \omega(t) = \omega_j, v(1), \ldots, v(t) \mid \boldsymbol{\theta})\right] \\ &= \max_{1 \le i \le c} \left[\max_{\omega(1),...,\omega(t-2)} P(\omega(1), \ldots, \omega(t-2), \omega(t-1) = \omega_i, v(1), \ldots, v(t-1) \mid \boldsymbol{\theta}) \cdot P(\omega_j \mid \omega_i, \boldsymbol{\theta}) \cdot P(v(t) \mid \omega_j, \boldsymbol{\theta})\right] \\ &= \left[\max_{1 \le i \le c} \delta_i(t-1) a_{ij}\right] b_{jv(t)}\end{aligned}$$

the pseudo-code

Pseudo-code for the Viterbi algorithm

1. **Initialize** $\delta_j(1) = \pi_j b_{jv(1)}$ **and** $\psi_j(1) = 0 \ (1 \le j \le c)$
2. **For** $t = 2$ to T
3. **For** $j = 1$ to c
4. $\delta_j(t) = \left[\max_{1 \le i \le c} \delta_i(t-1) a_{ij}\right] b_{jv(t)}; \ \psi_j(t) = \arg\max_{1 \le i \le c} \delta_i(t-1) a_{ij}$
5. **End**
6. **End**
7. **Decode** $\omega^*(T) = \arg\max_{1 \le j \le c} \delta_j(T)$
8. **Decode** $\omega^*(t) = \psi_{\omega^*(t+1)}(t+1) \ (1 \le t \le T-1)$ **with path backtracking (路径回溯)**

where ψ_{t+1} saves the max index i^* of the step t

Learning

Generally, there is **no known algorithm** which can obtain the **optimal** solution to the above problem

Try to find a local optimum based on iterative updating: in each iteration, update $\boldsymbol{\theta}$ to $\hat{\boldsymbol{\theta}}$ such that $P\left(\mathbf{V}^T \mid \hat{\boldsymbol{\theta}}\right) \geq P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)$

The Baum-Welch algorithm

Define $\gamma_{ij}(t) = P\left(\omega(t)=\omega_i, \omega(t+1)=\omega_j \mid \mathbf{V}^T, \boldsymbol{\theta}\right)$

which means the probability of being in state ω_i at step t, and state ω_j at step $t+1$, given the observed symbol sequence

$$\begin{aligned}
\gamma_{ij}(t) &= P\left(\omega(t)=\omega_i, \omega(t+1)=\omega_j \mid \mathbf{V}^T, \boldsymbol{\theta}\right) \\
&= \frac{P\left(\omega(t)=\omega_i, \omega(t+1)=\omega_j, \mathbf{V}^T \mid \boldsymbol{\theta}\right)}{P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)} \\
&= \frac{P\left(v(1), \ldots, v(t), \omega(t)=\omega_i, \omega(t+1)=\omega_j, v(t+1), \ldots, v(T) \mid \boldsymbol{\theta}\right)}{P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)} \\
&= \frac{\alpha_i(t) a_{ij} b_{jv(t+1)} \beta_j(t+1)}{P\left(\mathbf{V}^T \mid \boldsymbol{\theta}\right)} \\
&= \frac{\alpha_i(t) a_{ij} b_{jv(t+1)} \beta_j(t+1)}{\sum_{i=1}^{c} \sum_{j=1}^{c} \alpha_i(t) a_{ij} b_{jv(t+1)} \beta_j(t+1)}
\end{aligned}$$

where

$$\begin{aligned}
\alpha_i(t) &= P\left(v(1), v(2), \ldots, v(t), \omega(t)=\omega_i \mid \boldsymbol{\theta}\right) \\
a_{ij} &= P\left(\omega_j \mid \omega_i\right) \\
b_{jv(t+1)} &= P\left(v(t+1) \mid \omega_j\right) \\
\beta_j(t+1) &= P\left(v(t+2), v(t+3), \ldots, v(T) \mid \omega(t+1)=\omega_j, \boldsymbol{\theta}\right)
\end{aligned}$$

Pseudo-code for the Baum-Welch algorithm

1. **Randomly initialize** $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$
2. **Repeat**
3. **Estimate** $\alpha_j(t)$ $(1 \leq j \leq c, 1 \leq t \leq T)$ **by invoking the forward algorithm**
4. **Estimate** $\beta_j(t)$ $(1 \leq j \leq c, 1 \leq t \leq T)$ **by invoking the backward algorithm**
5. **Set** $\gamma_{ij}(t) = \dfrac{\alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}{\sum_{i=1}^{c} \sum_{j=1}^{c} \alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}$ $(1 \leq i, j \leq c, 1 \leq t \leq T-1)$
6. **Set** $\hat{\boldsymbol{\theta}} = \{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\boldsymbol{\pi}}\}$ **such that** $\forall\, 1 \leq i, j \leq c, 1 \leq k \leq K:$

$$\hat{\pi}_i = \sum\nolimits_{j=1}^{c} \gamma_{ij}(1) \qquad \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_{ij}(t)}{\sum_{t=1}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)} \qquad \hat{b}_{ik} = \frac{\sum_{t=1, v(t)=v_k}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)}{\sum_{t=1}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)}$$

7. **Update** $\boldsymbol{\theta} \leftarrow \hat{\boldsymbol{\theta}}$
8. **Until convergence**

practical convergence condition: $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\| \leq \epsilon$