Ch 17 Markov and Hidden Markov Model

17.3 Hidden Markov Model

- discrete-time, discrete-state Markov chain
- with hidden state and observation model

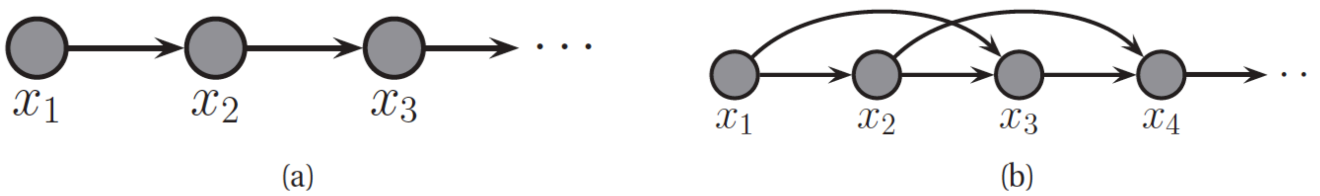


Figure 10.3 A first and second order Markov chain.

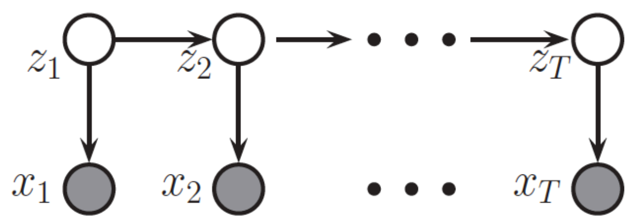


Figure 10.4 A first-order HMM.

결합 확률

- 은닉 상태 천이확률과 조건부 관측확률로 표현 가능

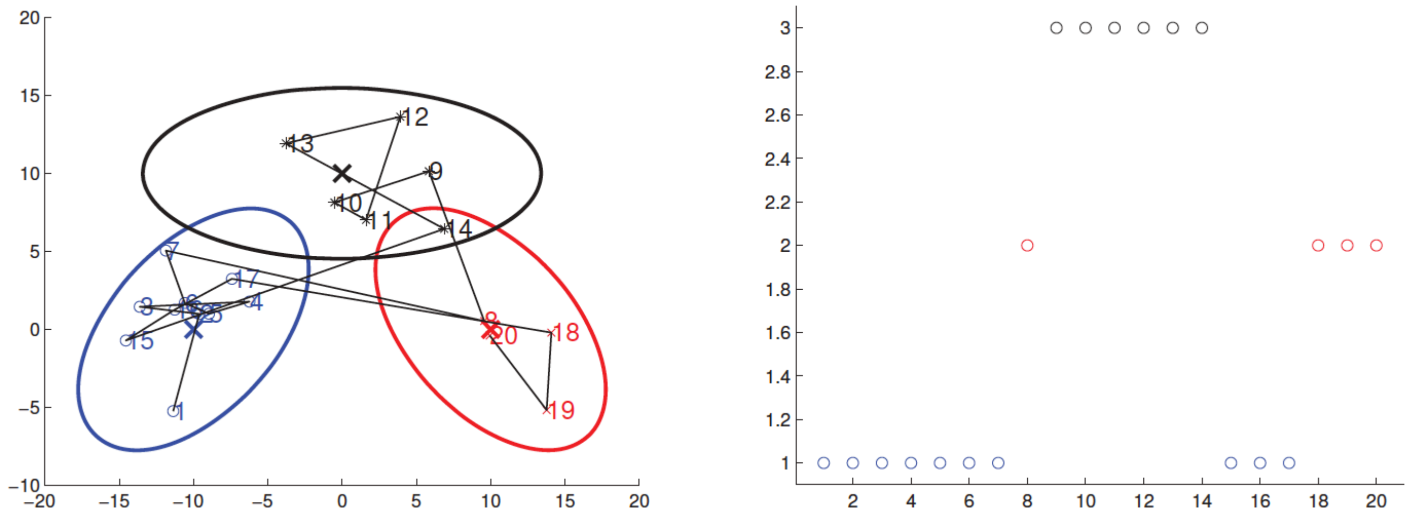
$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_{1:T}) p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = \left[p(z_1) \prod_{t=2}^{T} p(z_t|z_{t-1}) \right] \left[\prod_{t=1}^{T} p(\mathbf{x}_t|z_t) \right] \tag{17.39}$$

관측(observations)

- discrete 인 경우 observation matrix로 표현 가능
- continuous 인 경우 conditional gaussian으로 표현 가능

$$p(\mathbf{x}_t = l | z_t = k, \boldsymbol{\theta}) = B(k, l) \tag{17.40}$$

$$p(\mathbf{x}_t | z_t = k, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{17.41}$$



(a) (b)

Figure 17.7 (a) Some 2d data sampled from a 3 state HMM. Each state emits from a 2d Gaussian. (b) The hidden state sequence. Based on Figure 13.8 of (Bishop 2006b). Figure generated by `hmmLillypadDemo`.

17.3.1 HMM의 응용

- black-box density models on sequences
- long-range dependencies between observations mediated via the latent variables

음성 인식

- 관측 : 음성 신호
- 은닉 상태 : 단어
- 상태 천이 모델 : p(z_i | z_j), language model
- 관측 모델 : p(x|z), acoustic model

Speech tagging

- 관측 : 단어
- 은닉 상태 : POS(part of speech, ex. noun, verb, adjective)
- 상태 천이 모델 : grammer model

비디오로부터 활동 인식

- 관측 : 영상 feature
- 은닉 상태 : class of activity, ex) 뛰기, 걷기, 앉기..

17.4 Inference in HMMs

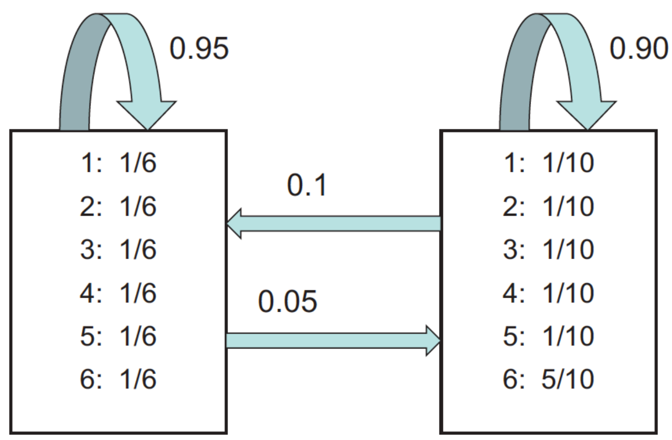
- inference란?
  - 관측열을 보고, 내제된 은닉 상태열을 추정하는 것
  - 편의상, 모든 모델 파라미터는 안다고 가정

부정직한 카지노 예제

- 공평(fair) 주사위 v.s 불공정(loaded) 주사위
- 관측열(rolls)를 보고, 어느시점에 어느 주사위가 던져졌는지 추론

Listing 17.1 Example output of `casinoDemo`

```
Rolls:    664153216162115234653214356634261655234232315142464156663246
Die:      LLLLLLLLLLLLLLLFFFFFLLLLLLLLLLLLLLFFFFFFFFFFFFFFFFFFLLLLLLLL
```



17.4.1 Inference의 종류

Filtering

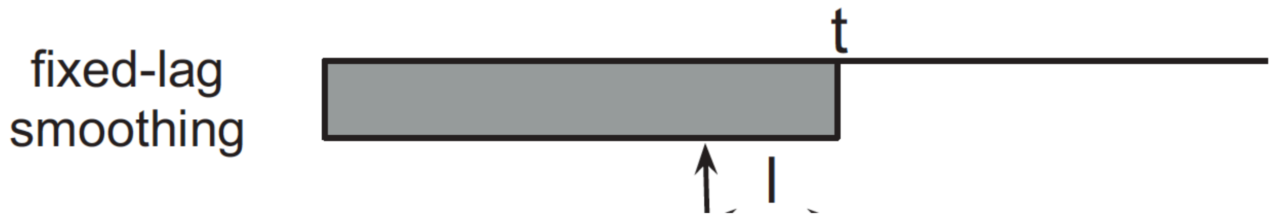
- 과거부터 현재까지의 관측열(x_1 ~ x_t)를 보고, 현재의 은닉 상태(z_t)를 추정하는 것
- on-line as data streams in

Smoothing

- 모든 관측(과거,현재,미래)을 보고, 특정시점의 은닉 상태(z_t) 추정
- 추가적 정보를 바탕으로 필터링 결과를 smooth
- hindsight, 지나고 나서야 나중에 깨닫는 것
- offline

Fixed-lag smoothing

- smoothing이기는 하지만 lag 정도의 시점의 은닉 상태 추정
- offline과 online의 특성 조합
- lag가 작아지면 판단 delay는 작아서 좋지만 정확도가 떨어진다.



Prediction

- 과거(그리고 현재) 관측을 바탕으로 미래 상태나 미래 관측 예측

$$p(z_{t+2}|\mathbf{x}_{1:t}) = \sum_{z_{t+1}} \sum_{z_t} p(z_{t+2}|z_{t+1}) p(z_{t+1}|z_t) p(z_t|\mathbf{x}_{1:t}) \quad (17.42)$$

$$p(\mathbf{x}_{t+h}|\mathbf{x}_{1:t}) = \sum_{z_{t+h}} p(\mathbf{x}_{t+h}|z_{t+h}) p(z_{t+h}|\mathbf{x}_{1:t}) \quad (17.43)$$

MAP estimation

- 모든 관측열을 바탕으로 가장 적합한 은닉 상태열을 추정하는 것
- ex) 음성으로 한 문장 듣기 => 단어열로 표현
- Viterbi-decoding

$$\arg\max_{\mathbf{z}_{1:T}} p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$$

Posterior samples

- sample from posterior

$$\mathbf{z}_{1:T} \sim p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}).$$

Probability of evidence

- 관측열 자체의 발생 확률
- 모든 가능한 은닉 상태열 경로를 모두 summing up 한 것

$$p(\mathbf{x}_{1:T}) = \sum_{\mathbf{z}_{1:T}} p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T})$$

17.4.2 Forward Algorithm

- 필터링을 얻고 싶다. p(z_t | x_1:t)

첫 번째 step : one-step ahead state prediction

- local evidence from anywhere * transition prob
- act as the new prior(belief) for time t

$$p(z_t = j|\mathbf{x}_{1:t-1}) \quad = \quad \sum_i p(z_t = j|z_{t-1} = i)p(z_{t-1} = i|\mathbf{x}_{1:t-1}) \tag{17.44}$$

두번째 step : update

- update belief given observation
- one-step ahead prediction 과 local evidence 를 곱한 후 normalize
- 취소선 : 조건부 독립

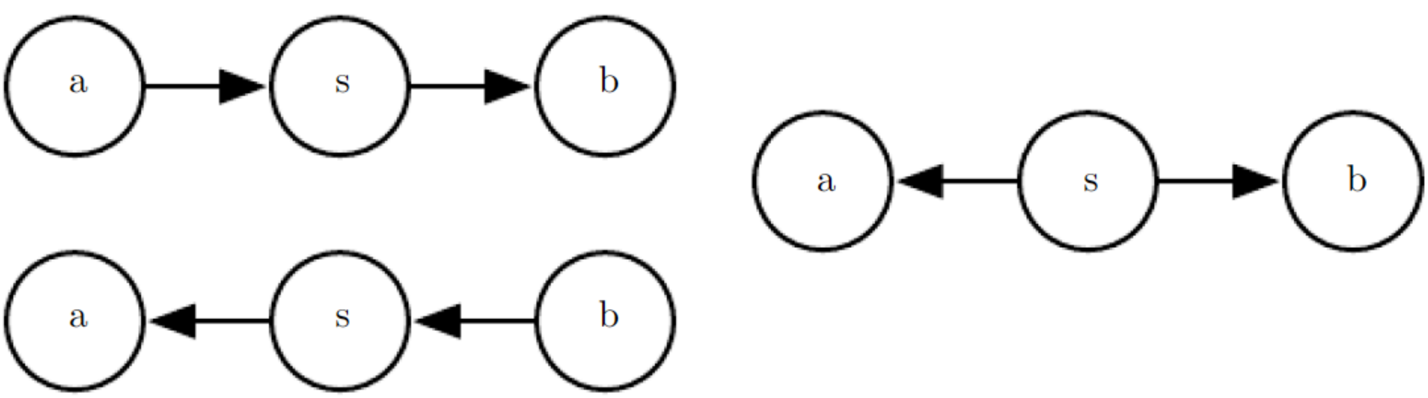
$$\alpha_t(j) \triangleq p(z_t = j|\mathbf{x}_{1:t}) = p(z_t = j|\mathbf{x}_t, \mathbf{x}_{1:t-1}) \tag{17.45}$$

$$= \frac{1}{Z_t} p(\mathbf{x}_t|z_t = j, \cancel{\mathbf{x}_{1:t-1}})p(z_t = j|\mathbf{x}_{1:t-1}) \tag{17.46}$$

$$\boldsymbol{\alpha}_t \propto \boldsymbol{\psi}_t \odot (\boldsymbol{\Psi}^T \boldsymbol{\alpha}_{t-1}) \tag{17.48}$$

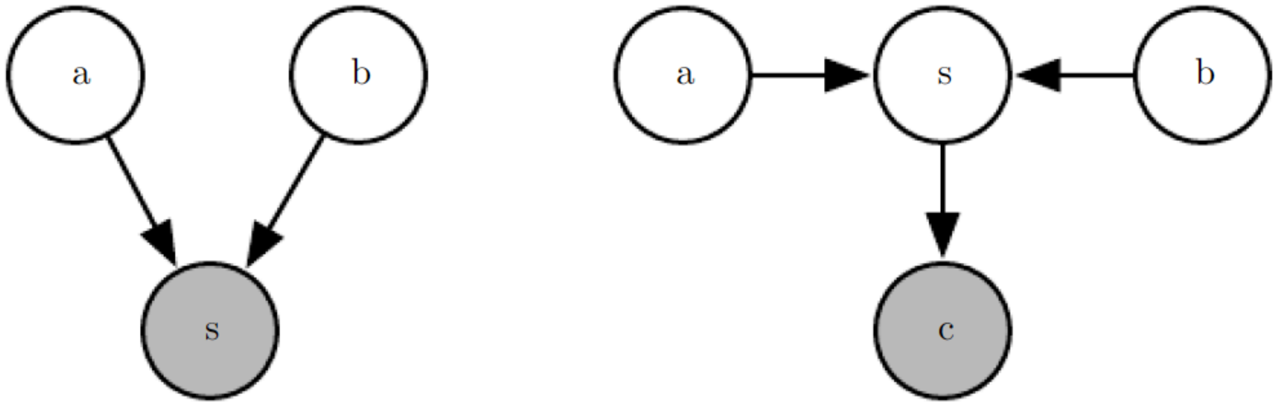
where $\psi_t(j) = p(\mathbf{x}_t|z_t = j)$ is the local evidence at time t, $\Psi(i,j) = p(z_t = j|z_{t-1} = i)$ is the transition matrix, and $\mathbf{u} \odot \mathbf{v}$ is the **Hadamard product**, representing elementwise vector

참고 : 조건부 독립



(a) (b)

- a) s가 관측되면 a,b는 조건부 독립
  - P(a|s,a) = P(a|s), P(b|s,a) = P(b|s)
- b) common-cause가 관측되면, effect들은 독립



(c) (d)

- c) result가 관측되면 원인들끼리 종속
  - 결근 by 휴가 or 질병
  - V-structured, explaining away

17.4.3 Forward-Backward Algorithm

- 최종 목적은 smoothed marginal을 구하는 것
  - p(z_t | x_1:T)
- 아래처럼 과거로부터 오는 chain과 미래로부터 온 chain으로 분리해서 생각
  - alpha : filtered belief state
  - beta : conditional likelihood of future evidence

The key decomposition relies on the fact that we can break the chain into two parts, the past and the future, by conditioning on z_t:

$$p(z_t = j|\mathbf{x}_{1:T}) \propto p(z_t = j, \mathbf{x}_{t+1:T}|\mathbf{x}_{1:t}) \propto \overbrace{\underbrace{p(z_t = j|\mathbf{x}_{1:t})}_{\text{past}}}^{\alpha}\overbrace{\underbrace{p(\mathbf{x}_{t+1:T}|z_t = j, \cancel{\mathbf{x}_{1:t}})}_{\text{future}}}^{\beta} \tag{17.50}$$

Let $\alpha_t(j) \triangleq p(z_t = j|\mathbf{x}_{1:t})$ be the filtered belief state as before. Also, define

$$\beta_t(j) \triangleq p(\mathbf{x}_{t+1:T}|z_t = j) \tag{17.51}$$

- beta도 DP 방식으로 계산 가능
  - one-step after beta, local evidence, 상태 천이 확률
  - 취소선 : 조건부 독립

$$\beta_{t-1}(i) = p(\mathbf{x}_{t:T}|z_{t-1} = i) \tag{17.54}$$

$$= \sum_j p(z_t = j, \mathbf{x}_t, \mathbf{x}_{t+1:T}|z_{t-1} = i) \tag{17.55}$$

$$= \sum_j p(\mathbf{x}_{t+1:T}|z_t = j, \cancel{z_{t-1} = i}, \cancel{\mathbf{x}_t})p(z_t = j, \mathbf{x}_t|z_{t-1} = i) \tag{17.56}$$

$$= \sum_j p(\mathbf{x}_{t+1:T}|z_t = j)p(\mathbf{x}_t|z_t = j, \cancel{z_{t-1} = i})p(z_t = j|z_{t-1} = i) \tag{17.57}$$

$$= \sum_j \beta_t(j)\psi_t(j)\psi(i, j) \tag{17.58}$$

17.4.3.3 시공간 복잡도

- state의 cardinality : K
- end of sequence : T

naive 한 구현인 경우

- O(K^2 * T)의 시간 복잡도
  - 각 time-step마다 K by K matrix 를 곱해야 하므로
  - K가 매우 크면(ex. 언어 모델) 계산 어려움

sparse한 transition matrix 라면

- O(TK)의 시간 복잡도

공간 복잡도가 더 문제

- O(KT) for storing alpha, betea
- divide-conquer 방법을 쓰면 O(K * logT) 로 줄어듬, 반대 급부로 시간 복잡도 증가

17.4.4 Viterbi 알고리즘

- 모든 관측열을 바탕으로 가장 적합한 은닉 상태열을 추정하는 것
- trellis diagram에서 최단 경로에 해당

$$\mathbf{z}^* = \arg\max_{\mathbf{z}_{1:T}} p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \tag{17.68}$$

MAP와 MPM의 차이

- MAP
  - Maximum a posterior
  - most probable sequence of states
- MPM
  - maximization of the posterior marginals
  - sequence of (marginally) most probable states

$$\hat{\mathbf{z}} = (\arg\max_{z_1} p(z_1|\mathbf{x}_{1:T}), \ldots, \arg\max_{z_T} p(z_T|\mathbf{x}_{1:T})) \tag{17.70}$$

MPM은 MAP보다 robust 하다.

why, note that in Viterbi, when we estimate z_t, we "max out" the other variables:

$$z_t^* = \arg\max_{z_t} \max_{\mathbf{z}_{1:t-1}, \mathbf{z}_{t+1:T}} p(\mathbf{z}_{1:t-1}, z_t, \mathbf{z}_{t+1:T}|\mathbf{x}_{1:T}) \tag{17.71}$$

whereas we when we use forwards-backwards, we sum out the other variables:

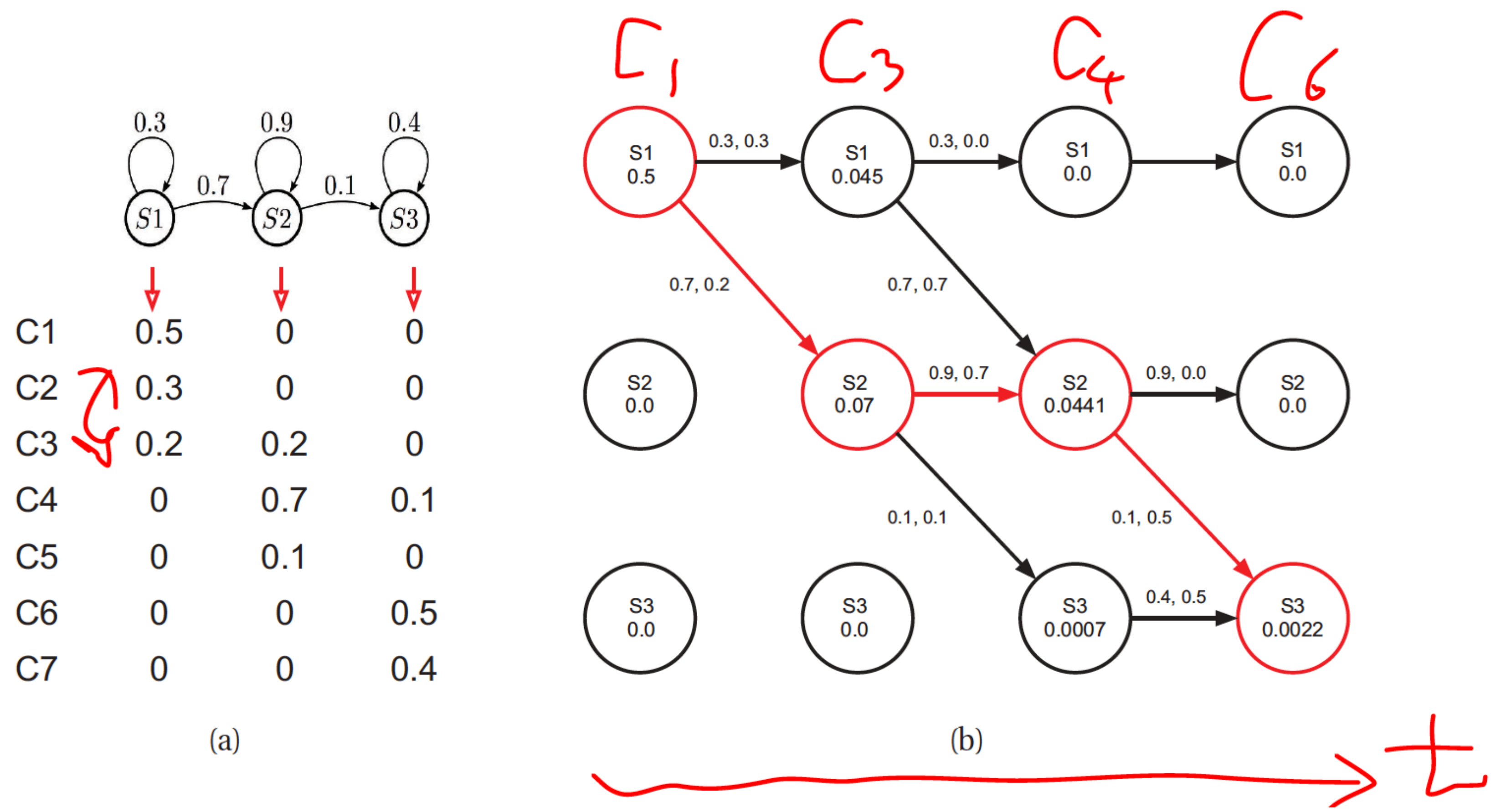
$$p(z_t|\mathbf{x}_{1:T}) = \sum_{\mathbf{z}_{1:t-1}, \mathbf{z}_{t+1:T}} p(\mathbf{z}_{1:t-1}, z_t, \mathbf{z}_{t+1:T}|\mathbf{x}_{1:T}) \tag{17.72}$$

17.4.4.2 Viterbi 알고리즘 상세

- delta : 현재까지의 최적의 path를 거칠 확률
  - 그 이전까지의 path도 최적이어야 한다.

$$\delta_t(j) \triangleq \max_{z_1,\ldots,z_{t-1}} p(\mathbf{z}_{1:t-1}, z_t = j|\mathbf{x}_{1:t}) \tag{17.73}$$

$$\delta_t(j) = \max_i \delta_{t-1}(i)\psi(i,j)\phi_t(j) \tag{17.74}$$



	S1	S2	S3
C1	0.5	0	0
C2	0.3	0	0
C3	0.2	0.2	0
C4	0	0.7	0.1
C5	0	0.1	0
C6	0	0	0.5
C7	0	0	0.4

(a) (b)

17.4.5 Forward filtering, bacward sampling

- sample paths from posterior

$$\mathbf{z}^s_{1:T} \sim p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \tag{17.83}$$

- do the forward pass, and then perform sampling in the backward pass

$$p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = p(z_T|\mathbf{x}_{1:T}) \prod_{t=T-1}^{1} p(z_t|z_{t+1}, \mathbf{x}_{1:T}) \tag{17.84}$$

We can then sample z_t given future sampled states using

$$z_t^s \quad \sim \quad p(z_t|z_{t+1:T}, \mathbf{x}_{1:T}) = p(z_t|z_{t+1}, \cancel{\mathbf{z}_{t+2:T}}, \mathbf{x}_{1:t}, \cancel{\mathbf{x}_{t+1:T}}) = p(z_t|z_{t+1}^s, \mathbf{x}_{1:t}) \tag{17.85}$$

The sampling distribution is given by

$$p(z_t = i|z_{t+1} = j, \mathbf{x}_{1:t}) \quad = \quad p(z_t|z_{t+1}, \mathbf{x}_{1:t}, \cancel{\mathbf{x}_{t+1}}) \tag{17.86}$$

$$= \frac{p(z_{t+1}, z_t|\mathbf{x}_{1:t+1})}{p(z_{t+1}|\mathbf{x}_{1:t+1})} \tag{17.87}$$

$$\propto \frac{p(\mathbf{x}_{t+1}|z_{t+1}, \cancel{z_t}, \cancel{\mathbf{x}_{1:t}})p(z_{t+1}, z_t|\mathbf{x}_{1:t})}{p(z_{t+1}|\mathbf{x}_{1:t+1})} \tag{17.88}$$

$$= \frac{p(\mathbf{x}_{t+1}|z_{t+1})p(z_{t+1}|z_t, \cancel{\mathbf{x}_{1:t}})p(z_t|\mathbf{x}_{1:t})}{p(z_{t+1}|\mathbf{x}_{1:t+1})} \tag{17.89}$$

$$= \frac{\phi_{t+1}(j)\psi(i, j)\alpha_t(i)}{\alpha_{t+1}(j)} \tag{17.90}$$