Unit 5 Hidden Markov model for Time-Series

Lesson 1 Markov Models and Hidden Markov Models

1. The Markov assumption
2. Markov models for discrete sequences
3. Hidden Markov models

The Markov assumption

We want more flexibility than assuming each timestep is iid. but more simpler than letting number of parameters grow with seq. length T.

First order Markov assumption:

z_{t+1} is conditionally independent of $z_1, z_2, \cdots, z_{t-1}$ given z_t

ex:

$p(z_1, z_2, z_3) = p(z_1)p(z_2|z_1)p(z_3|z_1, z_2)$ under product rule

$p(z_3|z_2, z_1) = p(z_3|z_2)$ under Markov assumption

$p(z_1, \cdots, z_T) = p(z_1) \cdot \prod_{t=2}^{N} p(z_t|z_{t-1})$

Make identical distribution assumption across time

$p(z_2 = k | z_1 = j) = A_{jk}$

$p(z_3 = k | z_2 = j) = A_{jk}$

$\vdots$

$A(z_T = k | z_{T-1} = j) = A_{jk}$

} Share same parameters A for all steps

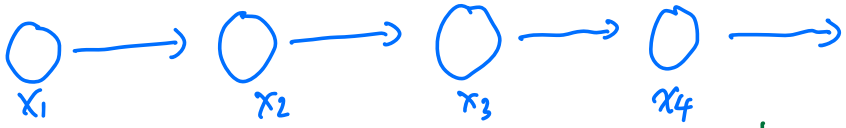
Summary:

(1) 1st order Markov: $p(z_{t+1} | z_t, z_{t-1}, \cdots) = p(z_{t+1} | z_t)$

(2) parameter sharing: $p(z_{t+1} = k | z_t = j) = p(z_2 = k | z_1 = j)$

achieve simple yet tractable model

1st order Markov model



Random Variable: Sequence $z_1, z_2, \cdots, z_T$ with each $z_t \in \{1, 2, \cdots, K\}$

Sample space: All possible sequence of length T using the K symbols.

Parameter:

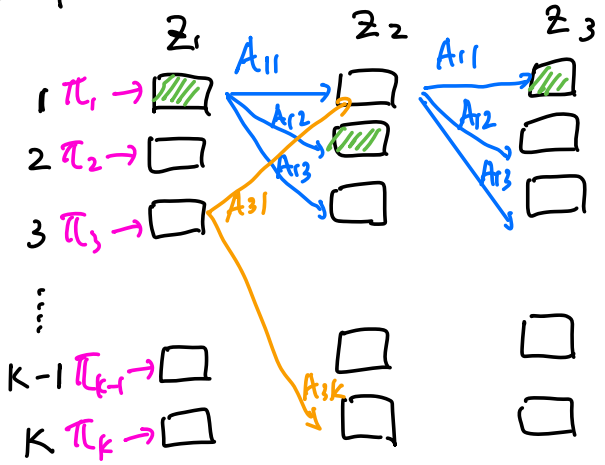
1. Initial State Prob:

$$p(z_1 = k) = \pi_k \qquad z_1 \sim Cat(\pi_1 \cdots \pi_K)$$

2. transition prob:

$$p(z_t = k \mid z_{t-1} = j) = A_{jk}$$

Example:



$p(1, 2, 1) = \pi_1 A_{12} A_{21}$

Cat: 单次实验, 多个结果 $\{0, 1, 2, \cdots, n\}$

Bern: 单次实验, 2个结果 $\{0, 1\}$

multi: 多次实验, 多个结果

Exercise:

① What is the marginal $p(z_1)$?

② What is the marginal $p(z_T)$?

Hidden Markov Model

Goal: Tractable model for observed data sequence $X_1, X_2, \dots, X_T$ with $X_t \in \mathbb{R}$ that has dependence between X_t and $X_1, X_2, \dots, X_{t-1}$, but affordable to do 2 tasks:

(1) Compute data likelihoods:

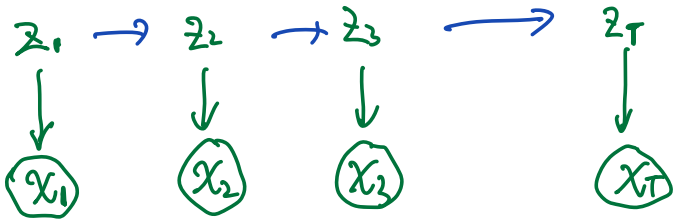
- joint $p(X_1, X_2, \dots, X_T \mid \theta)$ Evaluation Forward-Backward
- conditional $p(X_T \mid X_1, X_2, \dots, X_{T-1}, \theta)$ Decoding Viterbi

(2) Estimate parameters θ via penalized Maximum Likelihood. Learning EM

Idea:

hidden discrete state:

observed data:



$$p(X_{1:T}, Z_{1:T}) = \underbrace{p(Z_{1:T})}_{\text{First order Markov model}} \cdot \underbrace{p(X_{1:T} \mid Z_{1:T})}_{\text{each timestep iid given } Z_{1:T}}$$

$$= \left[p(Z_1) \prod_{t=2}^{N} p(Z_t \mid Z_{t-1}) \right] \cdot \prod_{t=1}^{T} p(X_t \mid Z_t)$$

$$= \text{Cat}(Z_1 \mid \pi) \prod_{t=2}^{T} \text{Cat}(Z_t \mid A_{Z_{t-1}}) \prod_{t=1}^{T} \text{NormPDF}(X_t \mid \mu_{Z_t}, \sigma^2_{Z_t})$$

with parameter $\theta = \{\underbrace{\pi, A}_{\text{used for } p(Z) \text{ Markov}}, \underbrace{\mu, \sigma}_{\text{used for } p(X \mid Z)}\}$

Lesson 2 EM for HMMs

1. EM for HMM parameter estimation
2. Expected log-likelihood for HMMs
3. M-step for HMMs
4. E-step for HMMs: overview
5. E-step for HMMs: forward-backward algorithm

Overview of EM for HMMs

Goal: Estimate parameters of an HMM using ML estimation

$$\max_{\pi, A, \mu, \sigma} \log p(x_1, x_2, \dots, x_T \mid \pi, A, \mu, \sigma)$$

Given: observed sequence $x_1, x_2, \dots, x_T$

Output: π : initial probability
A : transition probabilities — $K \times K$ matrix, row sum to 1
μ : means
σ : std deviations

Notation: $\theta = \{\pi, A, \mu, \sigma\}$

Challenges:

how to compute this likelihood?

- complete likelihood → easy: $p(x_{1:T}, z_{1:T} \mid \theta)$
- incomplete likelihood → hard: $p(x_{1:T} \mid \theta) = \sum_{z_{1:T} \in \mathcal{Z}} p(x_{1:T}, z_{1:T} \mid \theta)$

Complete likelihood:

$$p(x_{1:T}, z_{1:T}) = p(z_{1:T})\, p(x_{1:T} \mid z_{1:T}) \quad *$$

Assume $T=3$, $z_1 = a$, $z_2 = b$, $z_3 = c$

$$* = p(z_{1:3}) \prod_{t=1}^{3} p(x_t \mid z_t)$$
$$= p(a) \cdot p(b \mid a) \cdot p(c \mid b) \cdot \prod_{t=1}^{3} p(x_t \mid z_t)$$

incomplete like lihood:

$$p(x_{1:3}) = \sum_a \sum_b \sum_c p(x_{1:3}, z_1 = a, z_2 = b, z_3 = c)$$

Big idea:

(1) let $q(z|s)$ be an "approx. posterior". Define a valid distribution over the sequence $z = z_1, z_2, \cdots, z_T$

(2) Use the lower bound objective L:

$$\log p(x|\theta) = E_{q(z|s)}[\log p(x, z|\theta) - \log q(z|s)] \geq L(x, s, \theta)$$

(3) Iteratively optimize lower bound using coordinate ascent.

Init: θ^0

for iteration $i = 1, 2, \cdots$

E-step: $s_{1:T}^{(i)} \leftarrow \underset{s_{1:T}}{\arg\max}\, L(x_{1:T}, s_{1:T}, \theta^{(i-1)})$

M-step: $\theta^{(i)} \leftarrow \underset{\theta}{\arg\max}\, L(x_{1:T}, s_{1:T}^{(i)}, \theta)$

Punchline: Can do all key steps in affordable runtime $O(TK^2)$ or better. E step, M step, L calculation are all tractable

Define $q(z)$ and compute expected log likelihood (E-Step)

Q: What is $q(z)$? What is parameter s?

We can specify proba of adjacent timestamps:

$$q(z_t = k, z_{t+1} = l) = S_{tkl}$$

$k, l \in \{1, 2, \dots, K\}$

Parameter: S_t : $K \times K$ matrix

must satisfy $S_{tkl} \geq 0$, $\sum_k \sum_l S_{tkl} = 1$

Besides, for a valid q, neighbouring pairs should be consistent!

for a single timestamp:

$$p(z_t = k) = \sum_{l=1}^{K} (z_t = k, z_{t+1} = l) = \sum_l S_{tkl}$$

$$= \sum_{j=1}^{K} [z_{t-1} = j, z_t = k) = \sum_j S_{tjk}$$

That is ① valid PMF over $K \times K$

② consistent with neighbours

Recall:

$$p(X_{1:T}, z_{1:T}) = p(z_{1:T}) \cdot p(X_{1:T} \mid z_{1:T})$$

$$= p(z_1) \cdot \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \cdot \prod_{t=1}^{T} p(x_t \mid z_t)$$

$$= p(z_1 \mid \pi) \prod_{t=2}^{T} p(z_t \mid A_{t-1}) \cdot \prod_{t=1}^{T} N(x_t \mid \mu, \sigma)$$

Compute the log likelihood: (know z)

$$\log p(X_{1:T}, z_{1:T}) = \log p(z_{1:T}) + \log p(x \mid z_{1:T})$$

① ②

Let z in one-hot form: (written in z')

$$\log p(x \mid z) = \log \prod_{t=1}^{T} \prod_{k=1}^{K} N(x_t \mid \mu_k, \sigma_k^2)^{z'_{tk}}$$

$$= \sum_{t=1}^{T} \sum_{k=1}^{K} z'_{tk} N(x_t \mid \mu_k, \sigma_k^2)$$

② linear in T, linear in k

$$\log p(z_{1:T}) = \log\left[p(z_1|\pi)\prod_{t=2}^{T} p(z_t|A)\right] = \log(z_1|\pi) + \sum_{t=2}^{T}\log p(z_t|A)$$

① $p(z_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1k}}$

$p(z_t|z_{t-1}, A) = \prod_{k=1}^{K}\prod_{j=1}^{K} A_{jk}^{z'_{tk} z'_{t-1,j}}$ $= 1$ when $\begin{cases} z_t = k \\ z_{t-1} = j \end{cases}$

$$\Rightarrow \log p(z_{1:T}|\theta) = \sum_{k=1}^{K} z_{1k}\log\pi_k + \sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K} z'_{tk} z'_{t-1,j}\log A_{jk}$$

① linear in T; quadratic in K; $O(TK^2)$

Compute the expectation of log likelihood: (only know $q(z)$)

$$E_{q(z|\delta)} \log p(x_{1:T}, z_{1:T})$$

2 facts:

$$E_{q(z|\delta)}[z'_{tk}] = \sum_{\ell}\delta_{tk\ell} \overset{\text{Notation}}{=} r_{tk} \quad ?$$

$$E_{q(z|\delta)}[z'_{tk} z'_{t+1,\ell}] = \delta_{tk\ell}$$

Given:

$$\log p(x_{1:T}, z_{1:T}) = \log(x|z) + \log(z|\theta)$$

$$= \sum_{k=1}^{K} z_{1k}\log\pi_k + \sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K} z'_{tk} z'_{t-1,j}\log A_{jk}$$

$$+ \sum_{t=1}^{T}\sum_{k=1}^{K} z'_{tk}\, N(x_t|\mu_k, \sigma_k^2) \quad *$$

$$E(*) = \sum_{k=1}^{K} r_{1k}\log\pi_k + \sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K}\delta_{tjk}\log A_{jk} +$$

$$\sum_{t=1}^{T}\sum_{k=1}^{K} r_{tk}\, N(x_t|\mu_k, \sigma_k^2)$$

Summary:

1. Can compute $E[\log p(x, z)]$ easily using s of $q(z|s)$
2. we have define useful notation for marginals at each time t.

$$q(z_t = k) = r_{tk} = r_{tk}(s) = \begin{cases} \sum_{i=1}^{k} s_{tki} & \text{for } t = 1, 2, \cdots, T-1 \\ \sum_{j=1}^{k} s_{t-1,jk} & \text{for } t = T \end{cases}$$

M step for HMM

Using simplified expression for expected complete likelihood, can see M-step takes as input:

- r_{tk} probability of assigning timestep t to cluster k

  $r_{tk} = \sum_i s_{tki}$ (deterministic given s)

- s_{tjk} probability of assigning z_t to j and z_{t+1} to k

r, s: t-step specific parameters

π, A, μ, σ: global parameters for HMM

Given r, s, we can see M-step is simplified:

$$E(*) = \sum_{k=1}^{k} r_{1k} \log \pi_k + \sum_{t=2}^{T} \sum_{j=1}^{k} \sum_{k=1}^{k} s_{tjk} \log A_{jk} +$$

$$\sum_{t=1}^{T} \sum_{k=1}^{k} r_{tk} \mathcal{N}(x_t | \mu_k, \sigma_k^2)$$

M step: $\arg\max_{\theta} \mathcal{L}(x, s, \theta) = \arg\max_{\theta} E(*)$

$$\pi^* = \arg\max \sum_{k=1}^{k} r_{1k} \log \pi_k \Rightarrow \pi_k = \frac{r_{1k}}{1}$$

r_{1k}: # times cluster k to 1st step

$\Rightarrow r_1 \in \Delta^K$

$$A_j^* \leftarrow \underset{A_j \in \Delta^K}{\operatorname{argmax}} \sum_{t=1}^{T-1} \sum_k S_{tjk} \log A_{jk} \qquad A_{jk}^* = \frac{\sum_t S_{tjk}}{\sum_t \sum_k S_{tjk}}$$

$\sum_t S_{tjk}$: # time $j \to k$

$\sum_t \sum_k S_{tjk}$: # time $j \to$ anything

A_j sums to 1

$$\mu_k^*, \sigma_k^* \leftarrow \underset{\mu_k, \sigma_k}{\operatorname{argmax}} \sum_t r_{tk} \log \mathcal{N}(x_t; \mu_k, \sigma_k)$$

$\sum_t r_{tk}$: expected # time state k

Summary:

1. $S_t \to$ adjacent time step joint
   $r_t \to$ single time step margin

2. E step $\to S_t, r_t$

How to do the E-step?

Recall: $\log p(x|\theta) \geq L(x, s, \theta) + KL(q(z|s) \,|\, p(z|x,\theta))$ (posterior)

Best possible E step update could make $KL = 0$

and thus $\log p(x|\theta) = L(x, s, \theta)$

This achieve by $q(z|s) = p(z|x,\theta)$

In other word, we match our learned distribution q to the hidden given data posterior $p(z|x,\theta)$

while we could derive the optimal update by solving

$$s^* = \underset{s}{\operatorname{argmax}}\, L(x, s, \theta)$$

s that meet sum to one and neighbor consistency constraints

we could find the same optimal s^*, "matching the posterior" will be simpler.

- procedure: Analyse the posterior $p(z|x,\theta)$. specifically its moments for

  marginals t : $p(z_t | x_{1:T})$

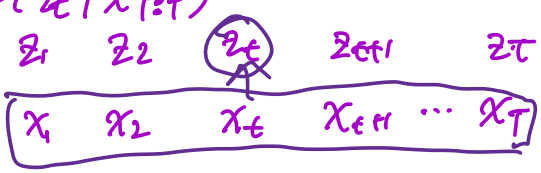
  pairwise joints $(t, t+1)$: $p(z_t, z_{t+1} | x_{1:T})$

We'll see both can be computed exactly via **dynamic programming**

- single Timestep Marginal Posterior

For each timestep t, we have:

$$p(z_t | x_{1:T}) = \frac{p(z_t, x_{1:T})}{p(x_{1:T})}$$



$$= \frac{p(x_{1:t}, z_t)\, p(x_{t+1:T} | x_{1:t}, z_t)}{p(x_{1:T})}$$ product rule

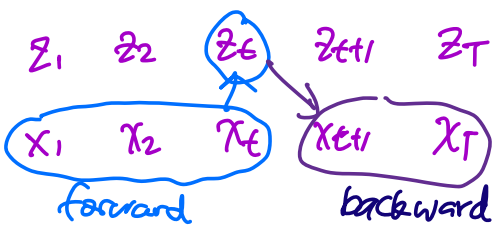
$$= p(x_{1:t}, z_t)\, p(x_{t+1:T} | z_t)$$ HMM conditional independence assumption

$$= \frac{p(x_{1:t}) \cdot p(z_t | x_{1:t}) \cdot p(x_{t+1:T} | z_t)}{p(x_{1:T})}$$

$p(x_{1:t})$ constant

$$\alpha_{tk} = p(z_t = k | x_{1:t}) \rightarrow \text{forward}$$

$$\beta_{tk} = p(x_{t+1:T} | z_t = k) \rightarrow \text{backward}$$



Thus:

$$p(z_t | x_{1:T}) = \frac{\alpha_{tk} \cdot \beta_{tk}}{\sum_\ell \alpha_{t\ell} \beta_{t\ell}}$$ ← normalization

• Adjacent Timestep Joint Posterior

For each timestep t in $1, 2, \dots T-1$, we have

$$p(z_t, z_{t+1} \mid x_{1:T}) = \frac{p(z_t, z_{t+1}, x_{1:T})}{p(x_{1:T})}$$

$$= \frac{1}{p(x_{1:T})} \cdot p(x_{1:t+2} \mid x_{1:t+1}, z_t, z_{t+1}) \cdot p(x_{1:t+1}, z_t, z_{t+1}) \quad *$$

$$p(x_{1:t}, z_t, z_{t+1}) = p(x_{t+1} \mid z_{t+1}, z_t, x_{1:t}) \cdot p(z_{t+1}, z_t, x_{1:t})$$

$$= p(x_{t+1} \mid z_{t+1}) \cdot p(z_{t+1} \mid z_t, x_{1:t}) \cdot p(z_t, x_{1:t})$$

$$= p(x_{t+1} \mid z_{t+1}) \cdot p(z_{t+1} \mid z_t) \cdot p(z_t \mid x_{1:t}) \cdot p(x_{1:t})$$

$$* \Rightarrow \frac{1}{p(x_{1:T})} \cdot \underbrace{p(x_{t+2:T} \mid z_{t+1})}_{\beta_{t+1}} \cdot \underbrace{p(x_{t+1} \mid z_{t+1})}_{\mathcal{N}(x_{t+1} \mid \mu_{z_{t+1}}, \sigma^2_{z_{t+1}})} \cdot \underbrace{p(z_{t+1} \mid z_t)}_{A_{z_t, z_{t+1}}} \cdot \underbrace{p(z_t \mid x_{1:t})}_{\alpha_t} \cdot p(x_{1:t})$$

$$p(z_t = j, z_{t+1} = k \mid x_{1:T}) = \frac{\beta_{t+1,k} \cdot L_{t+1,k} \cdot A_{j,k} \cdot \alpha_{tj}}{\sum_\ell \sum_m \beta_{t+1,m} \cdot L_{t+1,m} \cdot A_{\ell,m} \cdot \alpha_{t\ell}}$$

where $L_{tk} = p(x_t \mid z_t = k, \theta) = \mathcal{N}(x_t \mid \mu_k, \sigma_k^2)$

Punchline:

Given α, β, can set $S_{tkl} = p(z_t = k, z_{t+1} = l)$

$$= \frac{\alpha A L \beta}{\text{Normalization}}$$

forward and backward algorithm

Computing forward messages & via Dynamic Programming (Forward Algo)