Day22 in 2021s

SPR Day 18

Hidden Markov Models
Parameter Estimation and Likelihood Computation

Reading: Bishop PRML 13.2.1
13.2.2

Outline: Overview of EM for HMMs
Defining $q(z)$ and computing expected log likelihood
M step overview
E step intro
- FORWARD algorithm
- BACKWARD algorithm

Recap of E-step

Goal: Estimate parameters of an HMM using maximum likelihood method

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

each $x_t \in \mathbb{R}$

Given: observed sequence $x_1, x_2, x_3, \cdots x_T$

Output: π initial probabilities
A transition probabilities
μ means
σ stddeviations

Notation

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

All HMM parameters in one symbol θ

Challenges:
- how to compute this likelihood?

Complete likelihood is Easy: $p(x_{1:T}, z_{1:T} \mid \theta)$

Incomplete likelihood is Hard: $p(x_{1:T} \mid \theta) = \sum_{z_{1:T} \in \Omega} p(x_{1:T}, z_{1:T} \mid \theta)$ (sum rule)

Ω denotes all possible sequences of length T using symbols $\{1, 2, \cdots K\}$

Number of terms in sum is $|\Omega| = K^T$
grows exponentially with T (# timesteps)
Not easy! What else can we do?

Big Idea:

(1) Let $q(z|s)$ be an "approximate" posterior,
defining a valid distribution over
the sequence $z = z_1, z_2, \cdots z_T$

(2) Use the lower bound objective $\mathcal{L}$

$$\log p(x|\theta) \geq \mathbb{E}_{q(z|s)}\left[\log p(x,z|\theta) - \log q(z|s)\right]$$
$$= \mathcal{L}(x, s, \theta)$$

(3) Iteratively optimize lower bound
using coordinate ascent

Init: θ^0
for iteration $i = 1, 2, \ldots$

E-step: $s^i_{1:T} \leftarrow \arg\max_{s_{1:T}} \mathcal{L}(x_{1:T}, s_{1:T}, \theta^{i-1})$

M-step: $\theta^i \leftarrow \arg\max_{\theta} \mathcal{L}(x_{1:T}, s^i_{1:T}, \theta)$

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

Idea: Define a tractable distribution

$$q(z_{1:T} | s) \quad \text{over sequences } z_1, z_2, \cdots z_T$$

with each $z_t \in \{1, 2, \cdots K\}$

If we want onehot notation, write vector $\text{ONEHOT}(z_t)$

How? Define joint probability at each adjacent pair of timesteps: $(z_1, z_2), (z_2, z_3), \cdots (z_{T-1}, z_T)$

for $t = 1, 2, \cdots T-1$: $\quad q(z_t = j, z_{t+1} = k) = s_{tjk}$

Requirements:

1. each pairwise joint is valid PMF over $K \times K$ outcomes

s_t: $K \times K$ matrix, non-negative, sums to 1

$$\sum_j \sum_k s_{tjk} = 1 \quad \text{for } t = 1, 2, \cdots T-1$$

$$s_{tjk} \geq 0 \quad \text{for } t = 1, 2, \cdots T-1, \; j \in \{1, \cdots K\}, \; k \in \{1 \cdots K\}$$

2. neighboring pairs have consistent marginals

for all t,

$$q(z_t = k) = \sum_{j=1}^{K} q(z_{t-1} = j, z_t = k) = \sum_j s_{t-1,j,k}$$
$$= \sum_{l=1}^{K} q(z_t = k, z_{t+1} = l) = \sum_l s_{t,k,l}$$

Using this distribution, we can compute:

$$\mathbb{E}_{q(z|s)}\left[\text{ONEHOT}(z_t)_k\right] = \sum_{l=1}^{K} s_{tkl} \quad \text{for } t = 1, 2, \cdots T$$

$$\mathbb{E}_{q(z|s)}\left[\text{ONEHOT}(z_{t-1})_j \cdot \text{ONEHOT}(z_t)_k\right] = s_{tjk} \quad \text{for } t = 1, 2, \cdots T-1$$

scalar · scalar

If we know z: complete log likelihood for HMM

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

$$= \log \text{CatPMF}(z_1 | \pi) + \sum_{t=1}^{T-1} \log \text{CatPMF}(z_{t+1} | \text{ONEHOT}(z_t)^T A) + \sum_{t=1}^{T} \sum_{k=1}^{K} \text{ONEHOT}(z_t)_k \log \text{Norm}(x_n | \mu_k, \sigma_k^2)$$

$$= \sum_{k=1}^{K} \text{ONEHOT}(z_t)_k \log \pi_k + \sum_{t=1}^{T-1} \sum_{j=1}^{K} \sum_{k=1}^{K} \text{ONEHOT}(z_t)_j \, \text{ONEHOT}(z_{t+1})_k \log A_{jk} + \sum_{t=1}^{T} \sum_{k=1}^{K} \text{ONEHOT}(z_t)_k \log \text{Norm}(x_n | \mu_k, \sigma_k^2)$$

If we only know $q(z)$, replace above with expectation

$$\mathbb{E}_{q(z|s)}[\log p(x, z)] = \sum_{k=1}^{K} r_{1k}(s) \log \pi_k + \sum_{t=1}^{T-1} \sum_{j=1}^{K} \sum_{k=1}^{K} s_{tjk} \log A_{jk} + \sum_{t=1}^{T} \sum_{k} r_{tk}(s) \log \text{Norm}(x_t | \mu_k, \sigma_k^2)$$

We have defined useful notation for marginals at each time t

$$q(z_t = k) = r_{tk} = r_{tk}(s) = \begin{cases} \sum_{l=1}^{K} s_{tkl} & \text{for } t \in 1, 2, \ldots T-1 \\ \sum_{j=1}^{K} s_{t-1,jk} & \text{for } t = T \end{cases}$$

M-Step for HMMs

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

- r_{tk} probability of assigning timestep t to cluster k (deterministic given s: $r_{tk} = \sum_\ell s_{tk\ell}$)
- s_{tjk} probability of assigning z_t to j and z_{t+1} to k

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

$$\pi^* \leftarrow \underset{\pi \in \Delta^K}{\operatorname{argmax}} \sum_k r_{1k} \log \pi_k \qquad \boxed{\pi^*_k = \frac{r_{1k}}{1}}$$

$$A^*_{j\cdot} \leftarrow \underset{A_j \in \Delta^K}{\operatorname{argmax}} \sum_t \sum_k s_{tjk} \log A_{jk} \qquad \boxed{A^*_{jk} = \frac{\sum_t s_{tjk}}{\sum_t \sum_k s_{tjk}}}$$

$$\mu^*_k, \Sigma^*_k \leftarrow \underset{\mu_k, \Sigma_k}{\operatorname{argmax}} \sum_t r_{tk} \log \mathrm{Norm}(x_t \mid \mu_k, \Sigma_k)$$

μ^*_k, Σ^*_k: same as GMM M-step using $r = r(s)$

Note: likely want to use Penalty or MAP for Σ and π, maybe also A

How to do the E-step?

Recall: $\log p(x|\theta) \geq \mathcal{L}(x, s, \theta) + KL\left(q(z|s) \,\middle|\, p(z|x,\theta)\right)$

lower bound objective

KL term ≥ 0

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

and thus $\log p(x|\theta) = \mathcal{L}(x, s, \theta)$ [bound is tight]

This is achieved by finding $\underline{s}$ such that

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

In words, we match our learned distribution q to the hiddens-given-data posterior $p(z|x,\theta)$

While we could instead derive the optimal update by solving

$$s^* = \underset{\substack{s \text{ that meet} \\ \text{sum to one} \\ \text{and} \\ \text{neighbor consistency} \\ \text{constraints}}}{\operatorname{argmax}} \mathcal{L}(x, s, \theta)$$

we would 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): $p(z_t | x_{1:T})$

and 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:

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

by Bayes rule

which of K states at time t

given all data from start to end of sequence

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

product rule

$$\begin{array}{ccccccccc}
\underline{1} & & \underline{2} & \cdots & \underline{t} & & \underline{t+1} & \cdots & \underline{T} \\
z_1 & \rightarrow & z_2 \rightarrow & & \rightarrow z_t & \rightarrow & z_{t+1} \rightarrow & & \rightarrow z_T \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
x_1 & & x_2 & & x_t & & x_{t+1} & & x_T
\end{array}$$

$\boxed{x_{t+1} \text{ is conditionally indep of } x_t \text{ given } z_t}$

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

via HMM conditional independence assumption

$$= \frac{\overset{\text{const}}{p(x_{1:t})}\, p(z_t | x_{1:t})\, p(x_{t+1:T} | z_t)}{\underset{\text{const}}{p(x_{1:T})}}$$

Path forward: Suppose we can compute

$$\alpha_{tk} = p(z_t = k | x_{1:t}) \quad \text{for} \quad \begin{array}{l} t = 1, 2, \cdots T \\ k = 1, 2, \cdots K \end{array}$$

$$\beta_{tk} = p(x_{t+1:T} | z_t = k) \quad \text{for} \quad \begin{array}{l} t = 1, 2, \cdots T \\ k = 1, 2, \cdots K \end{array}$$

Then, we can compute single timestep posterior marginal as:

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

Adjacent Timestep Joint Posterior

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

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

$$= \frac{1}{p(x_{1:T})} p(z_t, z_{t+1}, x_{1:t+1}) \, p(x_{t+2:T} | z_{t+1})$$ product rule & x_{t+2} indep of $x_{1:t+1}$ given z_{t+1}

$$= \frac{1}{p(x_{1:T})} p(z_t, x_{1:t}) \, p(z_{t+1} | z_t, \cancel{x_{1:t}}) \, p(x_{t+1} | z_{t+1}, \cancel{z_t}, \cancel{x_{1:t}}) \, p(x_{t+2:T} | z_{t+1})$$ product rule

- const wrt z
- easy given θ
- using α, β defined on prev page

$$= \boxed{\frac{1}{p(x_{1:T})}} \; \overbrace{\frac{p(z_t | x_{1:t})}{\boxed{\frac{1}{p(x_{1:t})}}}}^{\alpha_t} \; \underbrace{p(z_{t+1} | z_t)}_{= A_{z_t, z_{t+1}}} \; \underbrace{p(x_{t+1} | z_{t+1})}_{\mathrm{NormPDF}(x_{t+1} | \mu_{z_{t+1}}, \sigma_{z_{t+1}})} \; \overbrace{p(x_{t+2:T} | z_{t+1})}^{\beta_{t+1}}$$ conditional independence assumptions

---

Thus, we can compute the adjacent timestep posterior if we have precomputed $\{\alpha_t, \beta_t\}_{t=1}^{T}$ as:

$$p(z_t = j, z_{t+1} = k | x_{1:T}) = \frac{\alpha_{tj} A_{jk} L_{t+1,k} \beta_{t+1,k}}{\sum_{l=1}^{K} \sum_{m=1}^{K} \alpha_{tl} A_{lm} L_{t+1,m} \beta_{t+1,m}}$$ for $t = 1, 2, \dots T-1$

where $L_{tk} = p(x_t | z_t = k, \theta) = \mathrm{NormPDF}(x_t | \mu_k, \sigma_k^2)$ when emission model is Gaussian

Computing Forward Messages α via Dynamic Programming (FORWARD alg.)

Definition: $\alpha_{tk} = p(z_t = k \mid x_{1:t})$ for $t = 1, 2, \dots T$

Procedure: Dynamic Program, with base case $t=1$ and recurrence relation that computes α_{t+1} from α_t

$$\alpha_{t+1,:} = f(\alpha_{t:})$$

$$\alpha_{t+1,k} = p(z_{t+1} = k \mid x_{1:t+1}) = \sum_j p(z_t = j, z_{t+1} = k \mid x_{1:t+1})$$

multiply by $\frac{p(x_{t+1} \mid x_{1:t})}{p(x_{t+1} \mid x_{1:t})}$

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

product rule

$$= \frac{1}{p(x_{t+1} \mid x_{1:t})} \sum_j p(z_t = j \mid x_{1:t})\, p(z_{t+1} = k \mid z_t = j, \cancel{x_{1:t}})\, p(x_{t+1} \mid z_{t+1} = k, \cancel{z_t}, \cancel{x_{1:t}})$$

cond. indep.   cond. indep.

Recursive update:

$$\boxed{\alpha_{t+1,k} = \frac{\sum_{j=1}^{K} \alpha_{tj} A_{jk} L_{t+1,k}}{\sum_\ell \sum_j \alpha_{tj} A_{j\ell} L_{t+1,\ell}}}$$

alpha_update(α_t, A, L_{t+1})

"Forward" update from α_t to α_{t+1}

Input: α_t : length K vector, sums to 1

A : K×K transition proba matrix

L_{t+1} : length K vector "likelihood" $L_{t+1,k} = p(x_{t+1} \mid z_t = k)$

FORWARD algorithm

Base Case $t=1$:
$$\alpha_{1k} = \frac{\pi_k L_{1k}}{\sum_\ell \pi_\ell L_{1\ell}}$$

for $t = 2, 3, 4, \dots T$:

α_{tk} = alpha_update(α_{t-1}, A, L_t)

return $\{\alpha_t\}_{t=1}^{T}$

Runtime

linear in T

quadratic in K

Using Forward Messages to compute incomplete log likelihood. 11

Recall $\log p(x_{1:T}|\theta)$ is useful to know
difficult to compute
by naively summing out $z_{1:T}$

Studying the recurrence relation in alpha_update, we focus on the denominator and see

$$p(x_{t+1}|x_{1:t}) = \sum_{j=1}^{K}\sum_{k=1}^{K} \alpha_{tj} A_{jk} L_{t+1,k}$$

$$= \text{mat\_mult}(\alpha_t, A) \cdot L_{t+1}$$

dot product or inner product

Insight: Compute

$$\log p(x_{1:T}) = \underbrace{\log p(x_1)}_{\text{use denominator in base case } (t=1) \text{ of FORWARD alg.}} + \underbrace{\sum_{t=2}^{T} \log p(x_t|x_{1:t-1})}_{\text{use denominator in step } t \text{ of FORWARD alg.}}$$

Computing backward messages β_t via dynamic programming (BACKWARD alg.)

Definition: $\beta_{tk} = p(x_{t+1:T} \mid z_t = k)$, for $t = 1, 2, \ldots T$

special case: $\beta_{Tk} = 1 \quad \forall k$

because T is final timestep. x_{T+1} does not exist

Procedure: Dynamic Programming

Need to fill in table $\beta = K \begin{array}{c} T \\ \boxed{\quad\quad\quad\quad} \end{array}$

Start with $\beta_{T:} = 1$

Work back in time using recurrance relation

$\beta_{t-1} = f(\beta_t)$ for $t = T, T-1, \ldots 3, 2, 1$

BACKWARD

Init: $\beta_{T:} = 1$

for $T-1, T-2, \ldots 1$:

$\beta_{tk} = \text{beta\_update}(\beta_{t+1}, L_{t+1}, A)$

Relation

$$\beta_{t-1,k} = p(x_{t:T} \mid z_{t-1} = k) = \sum_{l=1}^{K} p(x_{t:T}, z_t = l \mid z_{t-1} = k) \quad \text{sum rule}$$

$$= \sum_{l=1}^{K} p(x_t, x_{t+1:T} \mid z_t = l, z_{t-1} = k)\, p(z_t = l \mid z_{t-1} = k)$$

$$= \sum_{l=1}^{K} \overset{\beta}{p(x_{t+1:T} \mid z_t = l)}\, \underbrace{p(x_t \mid z_t = l)}_{L_{tl}}\, \underbrace{p(z_t = l \mid z_{t-1} = k)}_{A_{kl}}$$

x_{t+1} indep of prev x or $z_{1:t-1}$ given z_t

x_t indep of all else given z_t

beta_update

Input: β_t K-len vector

L_t K-len vector of likelihood

A $K \times K$ trans. proba

$$\boxed{\beta_{t-1,k} = \sum_{l=1}^{K} \beta_{tl} L_{tl} A_{kl}}$$

Recap: E Step

Goal: Update s parameters of $q(z|s)$ to optimal values

Procedure:

Input: π, A, μ, σ (HMM params)

Step 1: Calc likelihood $L = \{L_{tk}\}$

$$L_{tk} = \text{NormPDF}(x_t | \mu_k, \sigma_k^2)$$

Step 2: Calc forward messages $\alpha = \{\alpha_{tk}\}$

$$\log p(x|\theta), \alpha = \text{FORWARD}(\pi, A, L) \qquad O(TK^2)$$

Can compute $\log p(x|\theta)$ in FORWARD easily

Calc backward messages $\beta = \{\beta_{tk}\}$

$$\beta = \text{BACKWARD}(A, L) \qquad O(TK^2)$$

Step 3: Update s_t to match adjacent timestep joint posterior

for $t = 1, 2, \dots T-1$:

$$s_{tjk} = p(z_t = j, z_{t+1} = k | x_{1:T}) \quad \text{for } j,k \text{ in } 1 \dots K$$

$$\propto \alpha_{tj} A_{jk} L_{t+1,k} \beta_{t+1,k}$$

Step 4: Update $r = \{r_{tk}\}$ to match single timestep posterior

for $t = 1, 2, \dots T$

$$r_{tk} \propto \alpha_{tk} \beta_{tk}$$

Remember: $r = r(s)$

We can deterministically calculate r from s.

Just useful to have direct expression.

return $\log p(x_{1:T}|\theta), s, r$