‘A Tutorial on Hidden Markov Models’ – IEEE 1989 paper

June 2025

Jungsik Noh

UT Southwestern Medical Center

AR-HMM

- AR(1)-Hidden Markov Model

$$S_t \text{ : a hidden state } \in \{1, \dots, M\}$$

$$X_t = \phi_{0,S_t} + \phi_{1,S_t} X_{t-1} + \epsilon_t,\ \epsilon_t \sim N(0, \sigma^2_{S_t}) \text{ for } t = 1, 2, \dots, T,\ X_0 = 0$$

1. Initial probability (say, $\boldsymbol{\pi}$, 1×M vector)
   - $\pi_i = P(S_1 = i),\ i = 1, 2, \dots, M$ (# of hidden states)
2. Transition probability (say, $\boldsymbol{A}$, M×M matric)
   - $a_{ij} = P(S_{t+1} = j \mid S_t = i)$
3. PDF of X_t for a given $S_t = i$
   - $b_i(x_t \mid x_{t-1}) = pdf_{X_t}(x_t \mid S_t = i,\ \ X_{t-1} = x_{t-1}) = pdf\ of\ N\left(\phi_{0,i} + \phi_{1,i} x_{t-1},\ \ \sigma_i^2\right)$

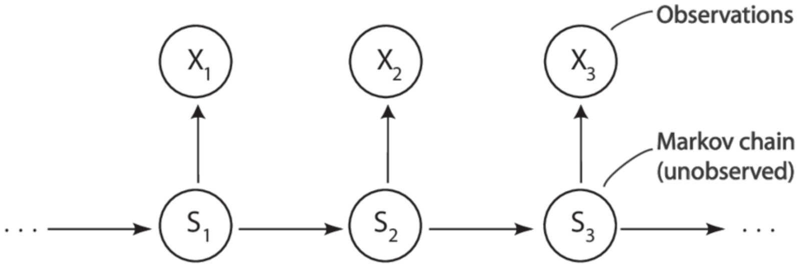
     (x_t's denote observations or data point.)
   - $\boldsymbol{B} = [b_i(x_t | x_{t-1})]$, T ×M matrix, a matrix of pdf values for given observations

- $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$: a vector with all parameters representing a model

EM for Hidden Markov Model (HMM)

A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

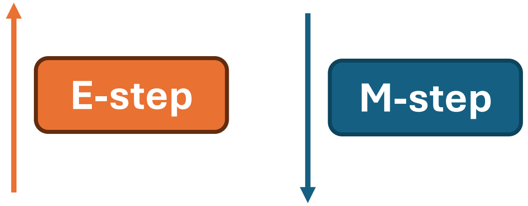
LAWRENCE R. RABINER, FELLOW, IEEE



- 3 key problems to fit a HMM

1. **Evaluation** : How to compute PDF (probability) for given observations and the model ($\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$)
   - $pdf_{X_1, X_2, \ldots, X_T}(x_1, x_2, \ldots, x_T \mid \boldsymbol{\lambda})$ using forward/backward equations
   - $P(X_1, X_2, \ldots, X_T | \boldsymbol{\lambda})$ (abuse the notation)
2. **Learning** : How to maximize the likelihood function $P(\boldsymbol{X} \mid \boldsymbol{\lambda})$
   - Estimate the transition probability given observations and the model
   - Update the emission parameters, $\boldsymbol{B}$ (EM algorithm)
3. **Decoding** : How to find the most likely hidden states for given observations and the model
   - Viterbi algorithm
   - $argmax_{s_1, s_2, \ldots, s_T}\, P(S_1 = s_1, \ldots, S_T = s_T \mid X_1 = x_1, \ldots, X_T = x_T, \boldsymbol{\lambda})$

- Transition probability
  - $a_{ij} = P(S_{t+1} = j \mid S_t = i)$
- Initial state probability
  - $\pi_i = P(S_1 = i)$



- Emission parameters for X_t

(eg) If $X_t \sim N(\mu_{S_t}, \sigma_{S_t}^2)$, then

$$\boldsymbol{\theta} = (\mu_1, \ldots, \mu_M, \sigma_1^2, \ldots, \sigma_M^2)$$

- Note differences in notation between the paper and the slides.
- $q_t \rightarrow S_t, S_i \rightarrow i,\ O_t \rightarrow X_t$ (paper -> slide), etc.

EM for Hidden Markov Model (HMM)

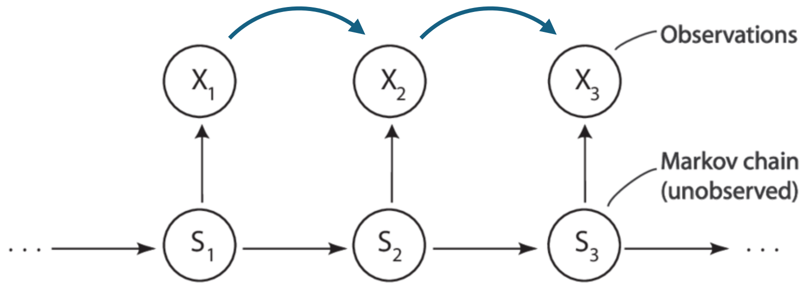
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

LAWRENCE R. RABINER, FELLOW, IEEE

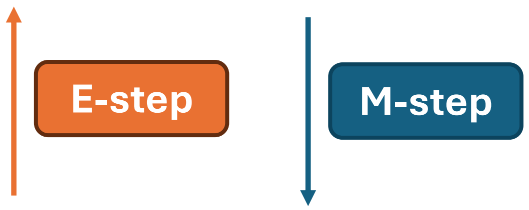
- 3 key problems to fit a HMM

1. **Evaluation** : How to compute PDF (probability) for given observations and the model ($\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$)
   – $pdf_{X_1, X_2, \ldots, X_T}(x_1, x_2, \ldots, x_T \mid \boldsymbol{\lambda})$ using forward/backward equations
   – $P(X_1, X_2, \ldots, X_T | \boldsymbol{\lambda})$ (abuse the notation)
2. **Learning** : How to maximize the likelihood function $P(\boldsymbol{X} \mid \boldsymbol{\lambda})$
   – Estimate the transition probability given observations and the model
   – Update the emission parameters, $\boldsymbol{B}$ (EM algorithm)
3. **Decoding** : How to find the most likely hidden states for given observations and the model
   – Viterbi algorithm
   – $argmax_{s_1, s_2, \ldots, s_T}\, P(S_1 = s_1, \ldots, S_T = s_T \mid X_1 = x_1, \ldots, X_T = x_T, \boldsymbol{\lambda})$

- Note differences in notation between the paper and the slides.
- $q_t \rightarrow S_t, S_i \rightarrow i,\ O_t \rightarrow X_t$ (paper -> slide), etc.



- Transition probability
  – $a_{ij} = P(S_{t+1} = j \mid S_t = i)$
- Initial state probability
  - $\pi_i = P(S_1 = i)$



- Emission parameters for X_t

(eg) If $X_t \sim N(\mu_{S_t}, \sigma_{S_t}^2)$, then

$$\boldsymbol{\theta} = (\mu_1, \ldots, \mu_M, \sigma_1^2, \ldots, \sigma_M^2)$$

Naïve joint PDF of $\boldsymbol{X}$

- Compute $pdf_{X_1,X_2,\dots,X_T}(x_1, x_2, \dots, x_T \mid \boldsymbol{\lambda})$ for given observations and parameters.
- Consider a fixed state sequence, $(S_1, S_2, \dots, S_T) = (s_1, s_2, \dots, s_T)$, and a corresponding conditional pdf.
  - $pdf_{X_1,X_2,\dots,X_T}(x_1, x_2, \dots, x_T \mid S_1 = s_1, S_2 = s_2, \dots, S_T = s_T, \boldsymbol{\lambda})$
  - (By law of total probability)

$$pdf_{X_1,X_2,\dots,X_T}(x_1, x_2, \dots, x_T \mid \boldsymbol{\lambda})$$

$$= \sum_{all\ possible\ s_1,\dots,s_T} \mathrm{P}(S_1 = s_1, S_2 = s_2, \dots, S_T = s_T|\boldsymbol{\lambda}) \cdot pdf_{X_1,X_2,\dots,X_T}(x_1, x_2, \dots, x_T \mid S_1 = s_1, S_2 = s_2, \dots, S_T = s_T, \boldsymbol{\lambda})$$

- $P(X_1, X_2, \dots, X_T \mid S_1, S_2, \dots, S_T, \boldsymbol{\lambda})$ (using $P(A \cap \mathrm{B} \mid C) = P(A|C)P(B|A \cap C)$)

$= P(X_1|S_1, S_2, \dots, S_T, \boldsymbol{\lambda})\, P(X_2|\, X_1,\ S_1, S_2, \dots, S_T, \boldsymbol{\lambda}) P(X_3|\, X_1,\ X_2,\ S_1, S_2, \dots, S_T, \boldsymbol{\lambda}) \cdots P(X_T|X_1, X_2, \dots, X_{T-1},\ \ S_1, S_2, \dots, S_T, \boldsymbol{\lambda})$

$= P(X_1 \mid S_1)\, P(X_2 \mid X_1, S_2)\, P(X_3|X_2, S_3) \cdots P(X_T|\, X_{T-1}, S_T)$

$= b_{S_1}(X_1) b_{S_2}(X_2|X_1) \cdots b_{S_T}(X_T|X_{T-1})$

- $\mathrm{P}(S_1 = s_1, S_2 = s_2, \dots, S_T = s_T) = \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \cdots a_{s_{T-1} s_T}$

Therefore,

$$pdf_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\lambda}) = \sum_{all\ possible\ s_1,\dots,s_T} \pi_{s_1} b_{s_1}(x_1)\, a_{s_1 s_2} b_{s_2}(x_2|x_1) a_{s_2 s_3} b_{s_3}(x_3|x_2) \cdots a_{s_{T-1} s_T} b_{s_T}(x_T|x_{T-1})$$

- Summation over M^T cases: computationally prohibitive

1. Evaluation: Forward equation

- Consider $\alpha_t(j) \;\coloneqq\; pdf_{X_1,X_2,\ldots,X_t,S_t}(x_1, x_2, \ldots, x_t, j \mid \boldsymbol{\lambda})$.

(1) Initialization

$$\alpha_1(i) = pdf_{X_1,S_1}(x_1, i \mid \lambda) = P(S_1 = i)P(X_1 = x_1 | S_1 = i) = \pi_i b_i(x_1 | x_0) \qquad (x_0 = 0), \quad \text{for } i = 1, 2, \ldots, M$$

(2) Induction ($t = 1, 2, \ldots, T-1$)

$$\begin{aligned}
\alpha_{t+1}(j) &= P(X_1, X_2, \ldots, X_{t+1}, S_{t+1} = j) && \text{(abuse of notation)} \\
&= \textstyle\sum_{i=1}^{M} P(X_1, X_2, \ldots, X_t, X_{t+1}, S_t = i, S_{t+1} = j) && (P(A) = \textstyle\sum_i P(A \cap B_i)\ \text{for } \{B_i\} \text{ is a partition}) \\
&= \textstyle\sum_{i=1}^{M} P(X_1, X_2, \ldots, X_t,\ S_t = i, S_{t+1} = j)\, P(X_{t+1} | X_1, X_2, \ldots, X_t,\ S_t = i, S_{t+1} = j) && \text{(def. of cond. Prob.)} \\
&= \textstyle\sum_{i=1}^{M} P(X_1, X_2, \ldots, X_t,\ S_t = i)\, P(S_{t+1} = j \mid X_1, X_2, \ldots, X_t,\ S_t = i)\, P(X_{t+1} | X_t, S_{t+1} = j) \\
&= \left\{\textstyle\sum_{i=1}^{M} \alpha_t(i)\, a_{ij}\right\} b_j(X_{t+1} | X_t)
\end{aligned}$$

(3) Termination

$$\sum_{i=1}^{M} \alpha_T(i) = \sum_{i=1}^{M} P(X_1, X_2, \ldots, X_T, S_T = i) = P(\boldsymbol{X} | \boldsymbol{\lambda}) = pdf_{X_1,X_2,\ldots,X_T}(x_1, x_2, \ldots, x_T \mid \boldsymbol{\lambda}) \qquad \text{(joint PDF value)}$$

- Consider a $(T \times M)$ matrix of $[\alpha_t(j)]$.

Backward equation

- Consider $\beta_t(i) \coloneqq pdf_{X_{t+1},X_{t+2},\dots,X_T \mid S_t,X_t}(x_{t+1},x_{t+2},\dots,x_T \mid i,x,\boldsymbol{\lambda})$.

(1) Initialization

$\beta_T(j) = 1$, for $j = 1,2,\dots,M$

(2) Induction ($t = 1,2,\dots,T-1$)

$$\beta_t(i) = P(X_{t+1},X_{t+2},\dots,X_T \mid S_t = i, X_t)$$

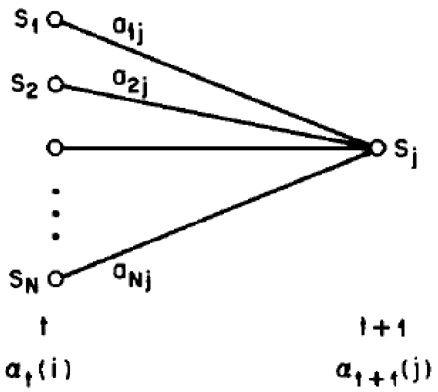
$$= \sum_{j=1}^{M} P(X_{t+1},X_{t+2},\dots,X_T,\ S_{t+1} = j \mid S_t = i, X_t) \qquad \text{($(S_{t+1} = j)$'s form a partition)}$$

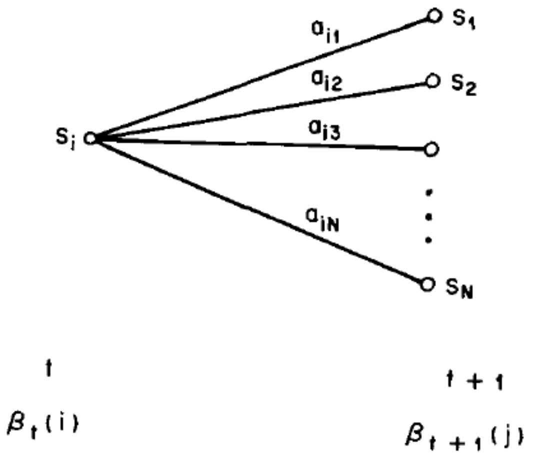
$$= \sum_{j=1}^{M} P(S_{t+1} = j \mid S_t = i, X_t)\, P(X_{t+1},X_{t+2},\dots,X_T \mid S_{t+1} = j, S_t = i, X_t)$$

$$= \sum_{j=1}^{M} a_{ij}\, P(X_{t+1} \mid S_{t+1} = j, S_t = i, X_t)\ P(X_{t+2},\dots,X_T \mid X_{t+1}, S_{t+1} = j, S_t = i, X_t)$$

$$= \sum_{j=1}^{M}\{a_{ij}\, b_j(X_{t+1}|X_t)\, \beta_{t+1}(j)\}$$

- Consider a $(T \times M)$ matrix of $[\beta_t(i)]$





Conditional probability of one state variable

- Consider $\gamma_t(i) \coloneqq pdf_{S_t \mid X_1,\dots,X_T}(i \mid x_1, \dots, x_2, \boldsymbol{\lambda}) = P(S_t = i \mid X_1 = x_1, X_2 = x_2, \dots, X_T = x_T, \boldsymbol{\lambda})$

$$\gamma_t(i) = P(S_t = i \mid \boldsymbol{X}) = \frac{P(S_t = i, X_1, X_2, \dots, X_t,\ X_{t+1}, \dots, X_T)}{P(\boldsymbol{X})}$$

$$= \frac{P(S_t = i, X_1, X_2, \dots, X_t)\, P(X_{t+1}, \dots, X_T \mid S_t = i, X_1, X_2, \dots, X_t)}{P(\boldsymbol{X})}$$

$$= \frac{\alpha_t(i)\, \beta_t(i)}{P(\boldsymbol{X})}$$

- $\sum_{i=1}^{M} \gamma_t(i) = 1$ implies $P(\boldsymbol{X} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{M} \alpha_t(i)\, \beta_t(i)$ for any $t = 1, 2, \dots, T$ (joint pdf value)

2. Learning: to estimate transition probabilities

- Consider $\xi_t(i,j) \coloneqq pdf_{S_t,S_{t+1}|\,X_1,X_2,\ldots,X_T}(i,j\,|x_1,x_2,\ldots,x_T,\boldsymbol{\lambda}) = P(S_t = i, S_{t+1} = j \mid X_1,X_2,\ldots,X_T,\boldsymbol{\lambda})$
- $P(S_t = i, S_{t+1} = j, X_1, X_2, \ldots, X_t, X_{t+1}, \ldots, X_T)$

$= P(X_1,X_2,\ldots,X_t,S_t = i)\, P(X_{t+1},\ldots,X_T,S_{t+1} = j\,|X_1,X_2,\ldots,X_t,S_t = i)$

$= \alpha_t(i)\;\; P(X_{t+1},S_{t+1} = j\,|X_1,X_2,\ldots,X_t,S_t = i)\, P(X_{t+2},\ldots,X_T\,|X_1,X_2,\ldots,X_t,X_{t+1},\;S_t = i,S_{t+1} = j)$

$= \alpha_t(i)\;\; P(X_{t+1},S_{t+1} = j \mid X_t,S_t = i)\;\; P(X_{t+2},\ldots,X_T \mid X_{t+1},\;S_{t+1} = j)$

$= \alpha_t(i)\;\; P(S_{t+1} = j \mid X_t,S_t = i)\;\;\; P(X_{t+1} \mid S_{t+1} = j,S_t = i,X_t)\;\;\; \beta_{t+1}(j)$

$= \alpha_t(i)\;\; a_{ij}\;\; b_j(X_{t+1} \mid X_t)\;\; \beta_{t+1}(j)$

$$\xi_t(i,j) = \frac{\alpha_t(i)\;\; a_{ij}\;\; b_j(X_{t+1} \mid X_t)\;\; \beta_{t+1}(j)}{P(\boldsymbol{X} \mid \boldsymbol{\lambda})}$$

for $t = 1, 2, \ldots, T-1$.

- $\sum_{i=1}^{M}\sum_{j=1}^{M}\xi_t(i,j) = 1$ implies $P(\boldsymbol{X} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_t(i)\;\; a_{ij}\;\; b_j(X_{t+1} \mid X_t)\;\; \beta_{t+1}(j)$.

- $[\xi_t(i,j)]$ forms an array with the size of $((T-1) \times M \times M)$.

2. Learning: EM iteration

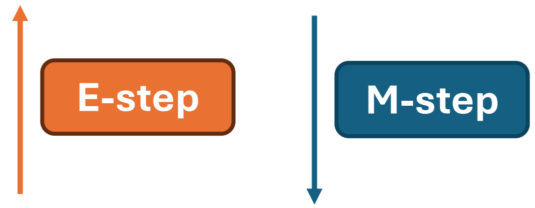
- Suppose parameter estimates at the iteration t are given: $\pi_i^{(t)}, a_{ij}^{(t)}, \phi_{0,i}^{(t)}, \phi_{1,i}^{(t)}, \sigma_i^{2\,(t)}$ for $i, j = 1, 2, \dots, M$

- (Initial probability) $\pi_i^{(t+1)} = \gamma_1(i)$
- (Transition probability)

$$a_{ij}^{(t+1)} = \frac{\hat{P}(S_t = i, S_{t+1} = j)}{\hat{P}(S_t = i)} = \frac{\frac{1}{T-1}\sum_{t=1}^{T-1} \xi_t(i,j)}{\frac{1}{T-1}\sum_{t=1}^{T-1} \gamma_t(i)}$$

- For initial parameter values ($t = 0$),
  - $\pi_i^{(0)}$: random probabilities form Uniform(0, 1) or $1/M$.
  - $a_{ij}^{(0)}$: random probabilities form Uniform(0, 1) with row sums being 1.
  - $\phi_{0,i}^{(0)}, \phi_{1,i}^{(0)}, \sigma_i^{2\,(0)}$: LSE estimates for all i.
  - Or any reasonable initial values.

- Transition probability
  - $a_{ij} = P(S_{t+1} = j \,|S_t = i)$
- Initial state probability
  - $\pi_i = P(S_1 = i)$



- Emission parameters for X_t

(eg) If $X_t \sim N(\mu_{S_t}, \sigma_{S_t}^2)$, then

$\boldsymbol{\theta} = (\mu_1, \dots, \mu_M, \sigma_1^2, \dots, \sigma_M^2)$

2. Learning: EM iteration

- LSE (MLE) for an AR(1) model
  - $\hat{\mu} \approx \bar{y}, \widehat{\mu_1} = \bar{x}$, $\widehat{\phi_1} = \frac{\sum(x_t - \hat{\mu})(x_{t-1} - \widehat{\mu_1})}{\sum(x_{t-1} - \widehat{\mu_1})^2}$, $\widehat{\phi_0} = \hat{\mu} - \widehat{\phi_1}\,\widehat{\mu_1}$

- AR estimates in M-step are the weighted averages.

$$\mu_i^{(t+1)} = \frac{\sum_{t=1}^{T} \gamma_t(i)\, x_t}{\sum_{t=1}^{T} \gamma_t(i)}, \qquad \mu_{1,i}^{(t+1)} = \frac{\sum_{t=1}^{T} \gamma_t(i)\, x_{t-1}}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\phi_{1,i}^{(t+1)} = \frac{\sum_{t=1}^{T} \gamma_t(i) \left(x_t - \mu_i^{(t+1)}\right)\left(x_{t-1} - \mu_{1,i}^{(t+1)}\right)}{\sum_{t=1}^{T} \gamma_t(i) \left(x_{t-1} - \mu_{1,i}^{(t+1)}\right)^2}, \qquad \phi_{0,i}^{(t+1)} = \mu_i^{(t+1)} - \phi_{1,i}^{(t+1)} \mu_{1,i}^{(t+1)}$$

$$\sigma_i^{2\,(t+1)} = \frac{\sum_{t=1}^{T} \gamma_t(i) \left(x_t - \phi_{0,i}^{(t+1)} - \phi_{1,i}^{(t+1)} x_{t-1}\right)^2}{\sum_{t=1}^{T} \gamma_t(i)}$$

3. Decoding: Viterbi algorithm

- Now optimize $argmax_{s_1,s_2,\dots,s_T} P(S_1 = s_1, \dots, S_T = s_T \mid X_1 = x_1, \dots, X_T = x_T, \boldsymbol{\lambda})$ with the converged EM estimates.

$$P(S_1 = s_1, \dots, S_T = s_T \mid X_1 = x_1, \dots, X_T = x_T, \boldsymbol{\lambda}) = pdf_{S_1,S_2,\dots,S_T|\boldsymbol{X}}(s_1, s_2, \dots, s_T \mid \boldsymbol{x}) = \frac{pdf_{\boldsymbol{S},\boldsymbol{X}}(\boldsymbol{s},\ \boldsymbol{x})}{pdf_{\boldsymbol{X}}(\boldsymbol{x})}$$

  - Sufficient to maximize $pdf_{\boldsymbol{S},\boldsymbol{X}}(\boldsymbol{s},\ \boldsymbol{x})$.

- Consider

$$\delta_t(i) = \max_{s_1,s_2,\dots,s_{t-1}} P(S_1 = s_1, S_2 = s_2, \dots, S_{t-1} = s_{t-1}, S_t = i, X_1, X_2, \dots, X_t \mid \boldsymbol{\lambda})$$

  - (Interpretation) For given observations up to time t, a probability that a state sequence up to time t ends with $S_t = i$ after passing through the most likely past states.

3. Decoding: Viterbi algorithm

- Consider

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} P(S_1 = s_1, S_2 = s_2, \dots, S_{t-1} = s_{t-1}, S_t = i, X_1, X_2, \dots, X_t \mid \boldsymbol{\lambda})$$

(1) Initialization

$$\delta_1(i) = P(S_1 = i, X_1) = \pi_i \, b_i(x_1 | x_0)$$

$$\psi_1(i) = 0$$

(2) Induction

$$\delta_t(j) = \max_i \max_{s_1, s_2, \dots, s_{t-2}} P(S_1 = s_1, \dots, S_{t-1} = i, S_t = j, X_1, X_2, \dots, X_t)$$

$$= \max_i \max_{s_1, s_2, \dots, s_{t-2}} P(S_1 = s_1, \dots, S_{t-1} = i,\ X_1, X_2, \dots, X_{t-1}) \, P(S_t = j,\ X_t | S_1 = s_1, \dots, S_{t-1} = i,\ X_1, X_2, \dots, X_{t-1})$$

$$= \max_i \delta_{t-1}(i) \, P(S_t = j \,| S_1 = s_1, \dots, S_{t-1} = i,\ X_1, X_2, \dots, X_{t-1}) \, P(X_t \mid S_t = j, S_1 = s_1, \dots, S_{t-1} = i,\ X_1, X_2, \dots, X_{t-1})$$

$$= \max_i \{\delta_{t-1}(i) \, a_{ij}\} \, b_j(X_t | X_{t-1})$$

$$\psi_t(j) = argmax_{\,1 \le i \le M} \{\delta_{t-1}(i) \, a_{ij}\}$$

(When $(S_1, \dots, S_t)$ ends with $S_t = j$, we only need to record which state is the best preceding one.)

(3) Termination

$$P^* = \max_i \delta_T(i), \quad s_T^* = \operatorname{argmax}_{1 \le i \le M} \{\delta_T(i)\}$$

(4) Path backtracking

$$s_t^* = \psi_{t+1}(s_{t+1}^*) \qquad \text{for } t = T-1, T-2, \dots, 1.$$

Summary: fitting AR(1)-HMMs

- AR(1)-Hidden Markov Model

S_t : a hidden state $\in \{1, \dots, M\}$

$X_t = \phi_{0,S_t} + \phi_{1,S_t} X_{t-1} + \epsilon_t,\ \epsilon_t \sim N(0, \sigma_{S_t}^2)$ for $t = 1, 2, \dots, T,\ X_0 = 0$

- Initials: $\pi_i^{(0)}, a_{ij}^{(0)}, \phi_{0,i}^{(0)}, \phi_{1,i}^{(0)}, \sigma_i^{2\,(0)}$

E-step: forward and backward equations

- $\alpha_t(j) = P(X_1 = x_1, \dots, X_t = x_t, S_t = j) = \{\sum_{i=1}^{M} \alpha_{t-1}(i) a_{ij}\} \cdot b_j(x_{t+1}|x_t)$
- $\beta_t(i) = P(X_{t+1} = x_{t+1}, \dots, X_T = x_T \mid S_t = i, X_t) = \sum_{j=1}^{M} a_{ij}\, b_j(x_{t+1}|x_t)\, \beta_{t+1}(j)$
- $\gamma_t(i) = P(S_t = i \mid X_1 = x_1, \dots, X_T = x_T) = \alpha_t(i)\beta_t(i) / \sum_{i=1}^{M} \alpha_t(i)\beta_t(i)$
- $\xi_t(i,j) = P(S_t = i, S_{t+1} = j \mid X_1, X_2, \dots, X_T, \boldsymbol{\lambda})$

M-step

- MLEs have the form of weighted averages

Viterbi algorithm

- $\delta_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} P(S_1 = s_1, S_2 = s_2, \dots, S_{t-1} = s_{t-1}, S_t = i, X_1, X_2, \dots, X_t \mid \boldsymbol{\lambda})$
- $s_t^* = \psi_{t+1}(s_{t+1}^*)$ (path backtracking)