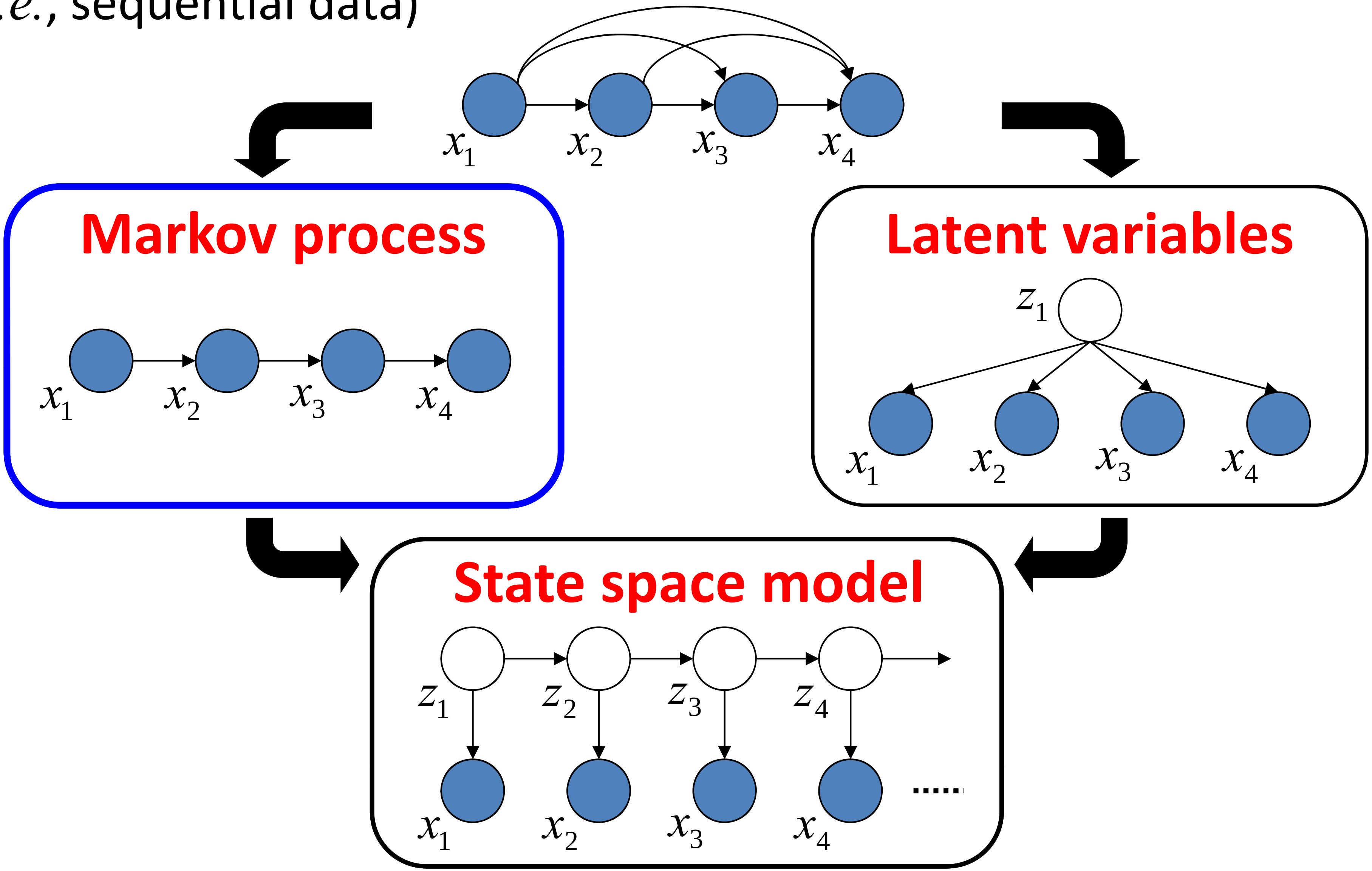
Sequential Data Modeling

Hidden Markov Models

Basic Approaches

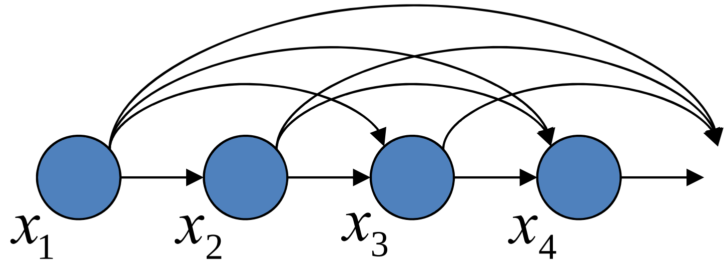
How to efficiently model joint probability of high-dimensional data ($i.e.$, sequential data)



Review: Markov Process

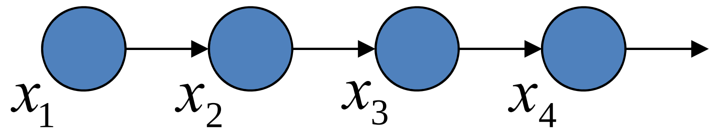
- Assume that the conditional probability distribution of the present states depends only on a few past states

$$p(x_1, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_1, \cdots, x_{n-1})$$



1st order Markov chain

$$p(x_1, \cdots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n | x_{n-1})$$



Show probabilistic graphical model

- Nodes: random variables
- Edges: causality relationship between variables

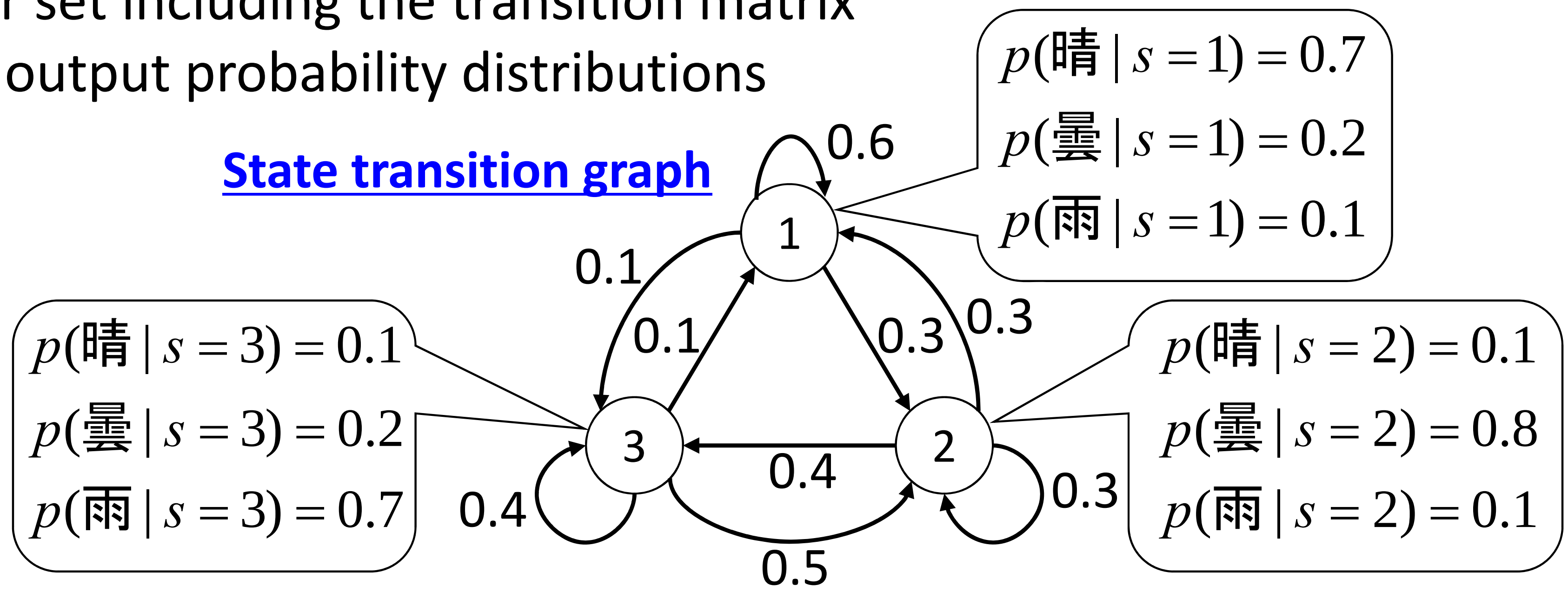
Easy to see conditional dependency between variables!

Hidden Markov Model (HMM)

- Use of discrete latent variables
- Parameter set including the transition matrix and state output probability distributions

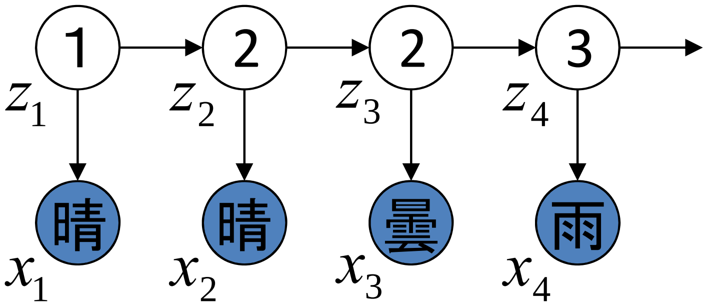
State transition graph

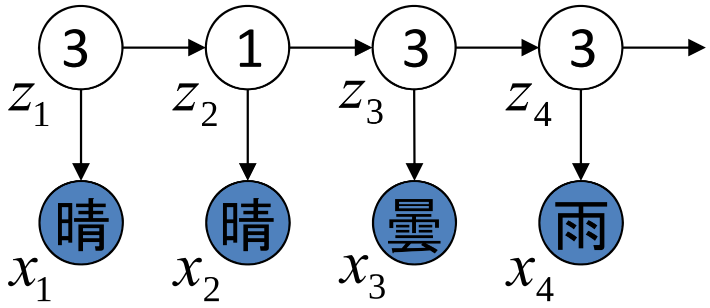
晴: fine
雨: rain
曇: cloud



Even if 晴 晴 曇 雨 … is observed, a state sequence is not observed (*i.e.*, **state sequence = latent variable**).

Possible state sequences…



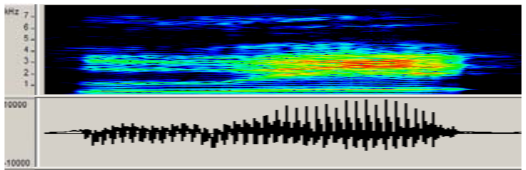


Example of Application

- Automatic speech recognition (*i.e.*, conversion from speech into text)

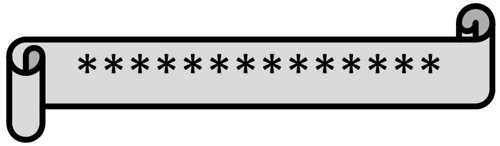
Speech parameter sequence → **Decoding** → **Word sequence**

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$$



$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} p(\boldsymbol{w} \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{w}} p(\boldsymbol{x} \mid \boldsymbol{w}) p(\boldsymbol{w})$$

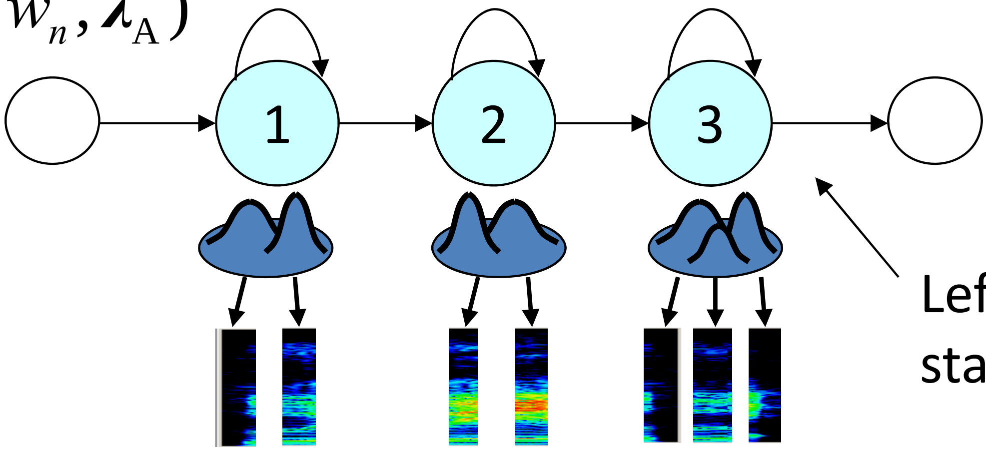
$$\hat{\boldsymbol{w}} = \{\hat{w}_1, \hat{w}_2, \cdots, \hat{w}_N\}$$



Acoustic model $p(\boldsymbol{x} \mid \boldsymbol{w}, \boldsymbol{\lambda}_{\mathrm{A}})$
e.g., **HMM** with **GMM**

Language model $p(\boldsymbol{w} \mid \boldsymbol{\lambda}_{\mathrm{L}})$
e.g., **Markov model**

$$p(\boldsymbol{x} \mid w_n, \boldsymbol{\lambda}_{\mathrm{A}})$$

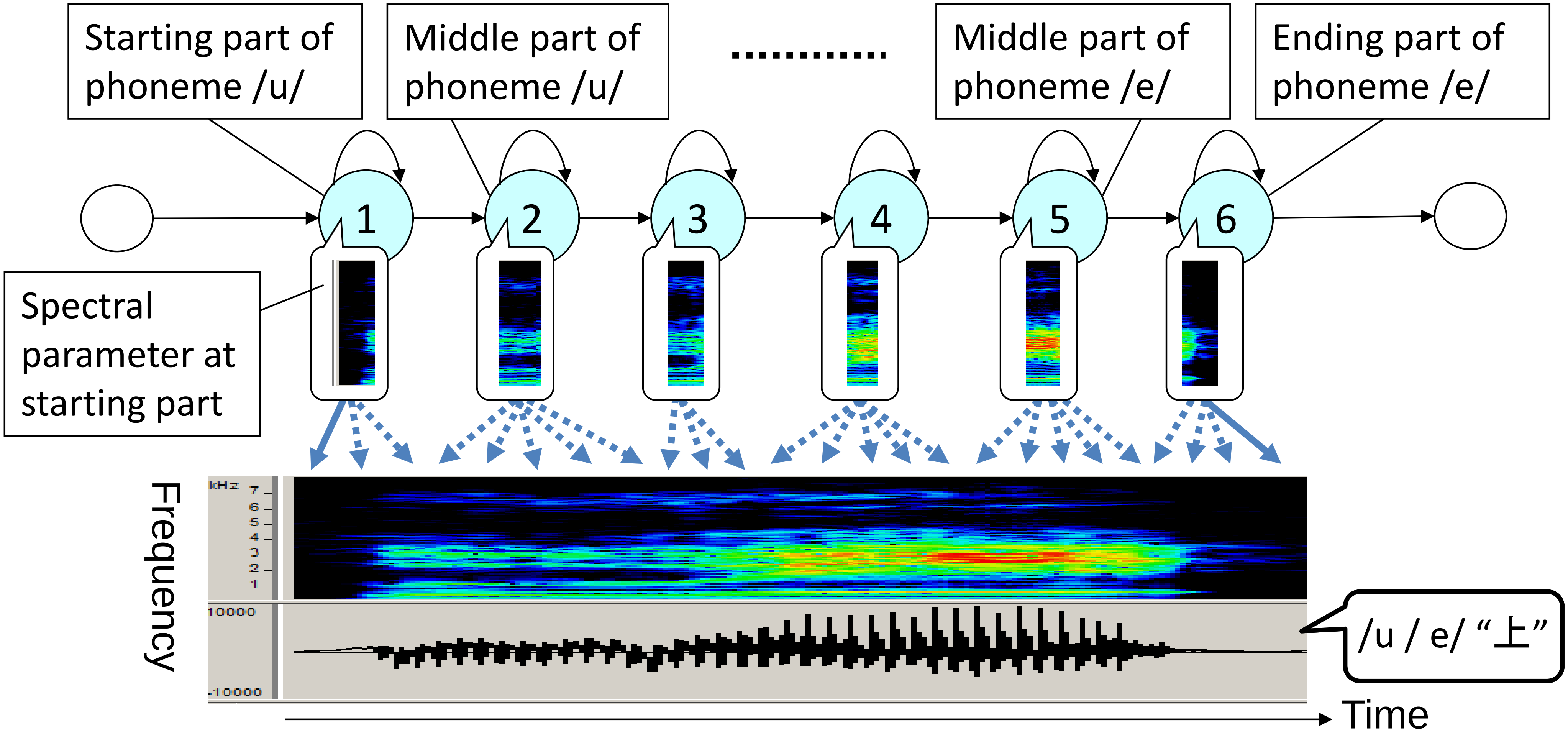


$$p(w_n \mid w_{n-1}, w_{n-2}, \boldsymbol{\lambda}_{\mathrm{L}})$$

Left-to-right HMM using GMM as state output probability density

Example: Acoustic Model

- HMM effectively models a time sequence of speech parameters
  - ✓ Handle fluctuation of speaking speed with self-loop transition
  - ✓ Handle fluctuation of articulation with output probability density function



Speech waveform and its spectral sequence

HMM as Mixture Model

- Observation sequence of length N

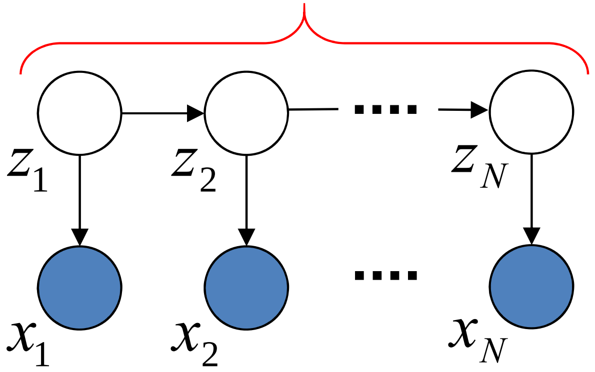
$$\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$$

- Corresponding state sequence of length N

$$\boldsymbol{z} = \{z_1, z_2, \cdots, z_N\}$$

- Likelihood of HMM

Markov chain of latent variables



x_4 depends on x_1 , x_2, and x_3.

$$p(\boldsymbol{x} \mid \boldsymbol{\lambda}) = \sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N, z_1, \cdots, z_N \mid \boldsymbol{\lambda})$$

$$= \sum_{\text{all } \boldsymbol{z}} \left\{ \underbrace{p(z_1 \mid \boldsymbol{\lambda}) \left[\prod_{n=2}^{N} p(z_n \mid z_{n-1}, \boldsymbol{\lambda}) \right]}_{\text{Prior probability of } \boldsymbol{z} \,:\, p(\boldsymbol{z})} \underbrace{\prod_{n=1}^{N} p(\boldsymbol{x}_n \mid z_n, \boldsymbol{\lambda})}_{\text{Output probability of } \boldsymbol{x} \text{ given } \boldsymbol{z} \,:\, p(\boldsymbol{x} \mid \boldsymbol{z})} \right\}$$

\# of mixture components : S^N

$$= \sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{z} \mid \boldsymbol{\lambda}) \, p(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\lambda})$$

Mixture model with discrete latent variables $\boldsymbol{z}$

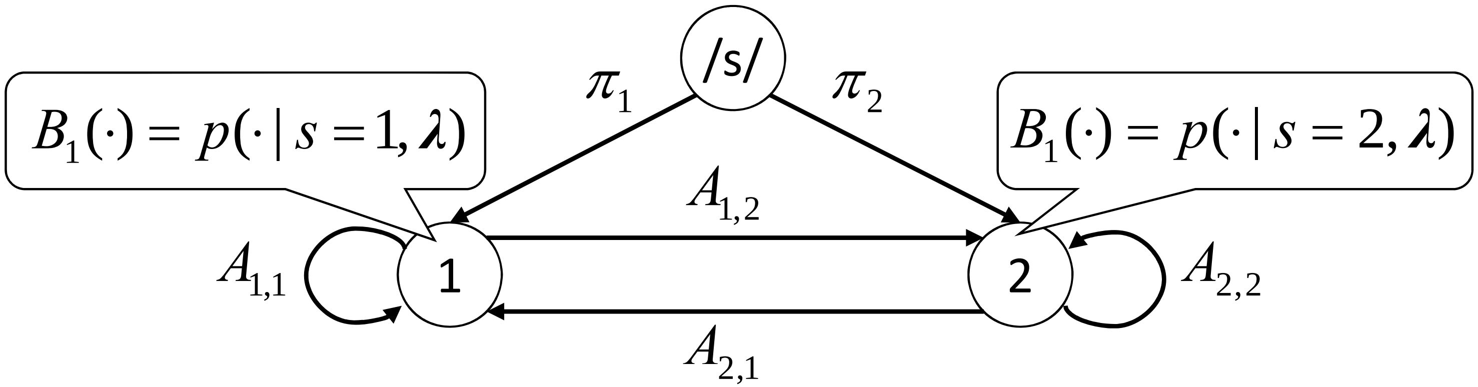
Elements of HMM

- HMM parameter set $\boldsymbol{\lambda} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}(\cdot)\}$
  - Set of S finite states: $s = \{1, 2, \cdots, S\}$
  - Initial state distribution

    $\boldsymbol{\pi} = \{\pi_1, \pi_2, \cdots, \pi_S\}$ $\qquad\qquad$ $\pi_i = p(s = i \mid \boldsymbol{\lambda})$
  - Transition probabilities

    $\boldsymbol{A} = \{A_{1,1}, A_{1,2}, \cdots, A_{S,S}\}$ $\qquad\qquad$ $A_{i,j} = p(s = j \mid s = i, \boldsymbol{\lambda})$
  - Output probabilities (or probability densities) of $\boldsymbol{x}_n$

    $\boldsymbol{B}(\boldsymbol{x}_n) = \{B_1(\boldsymbol{x}_n), B_2(\boldsymbol{x}_n), \cdots, B_S(\boldsymbol{x}_n)\}$ $\qquad$ $B_i(\boldsymbol{x}_n) = p(\boldsymbol{x}_n \mid s = i, \boldsymbol{\lambda})$



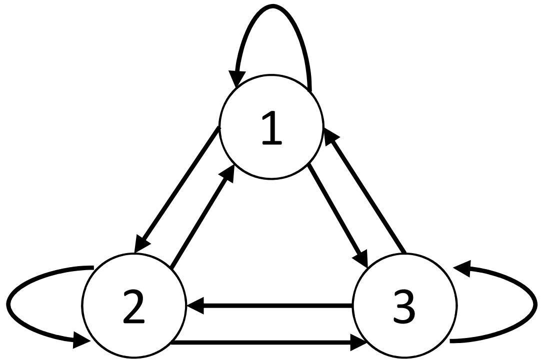
Type of HMM: Ergodic HMM

- Initial probabilities

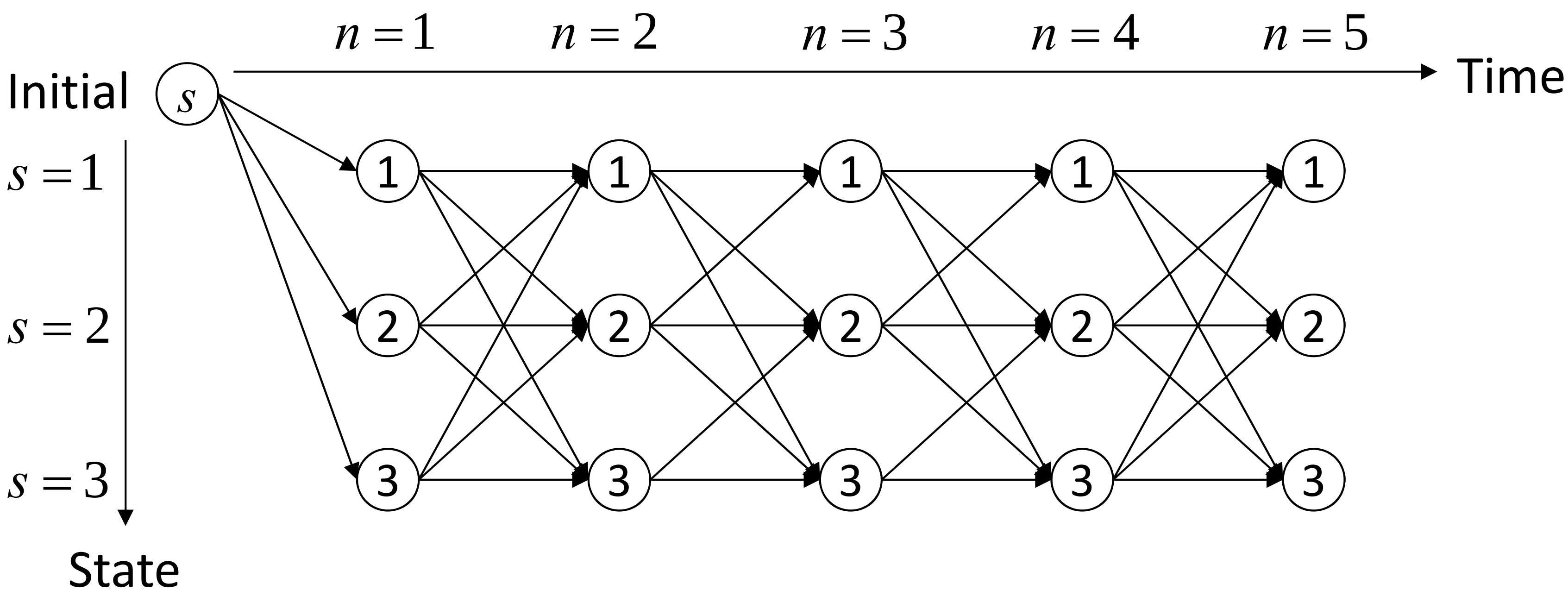
$$\pi_s = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix}$$

- Transition probabilities

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix}$$



- Trellis graph



Type of HMM: Left-to-Right HMM

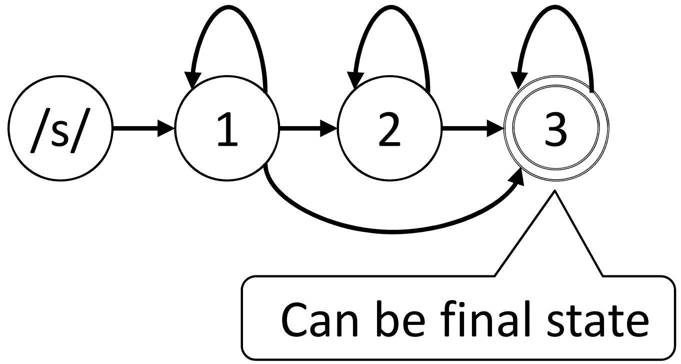
- Initial probabilities

$$\pi_s = \begin{cases} 1 & s = 1 \\ 0 & s \neq 1 \end{cases}$$

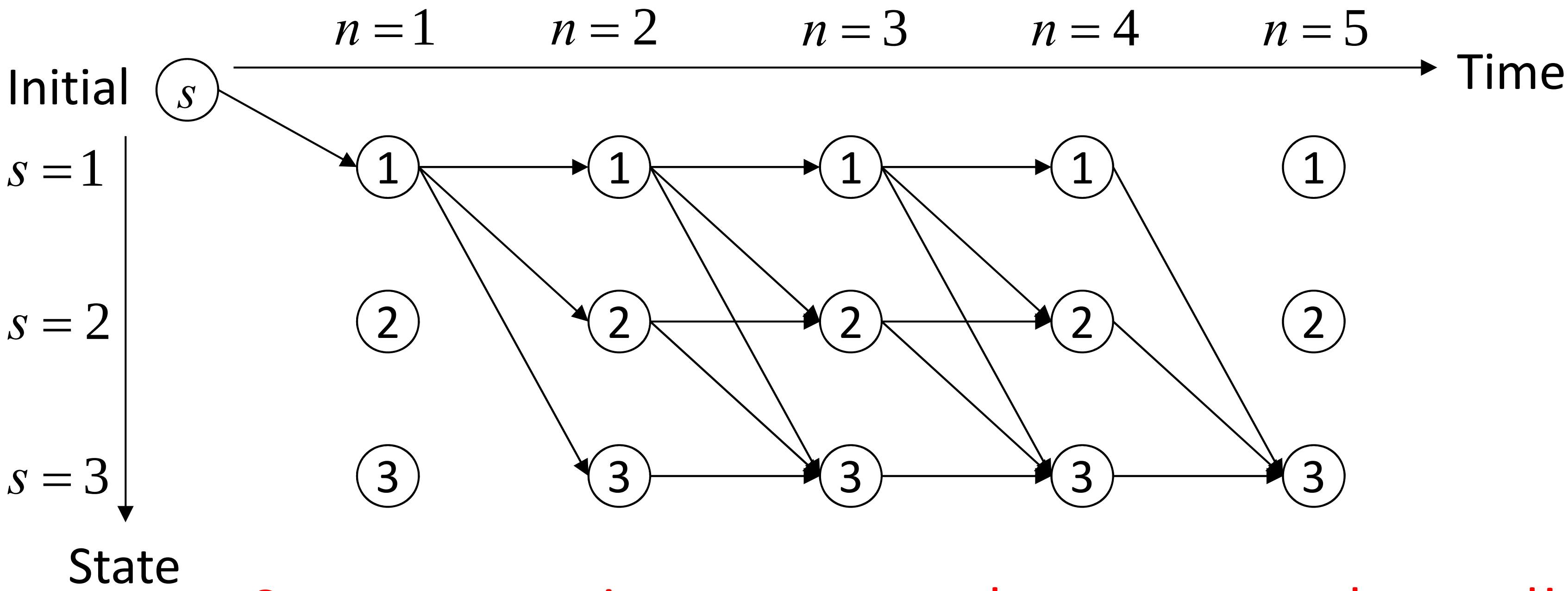
- Transition probabilities

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ 0 & A_{2,2} & A_{2,3} \\ 0 & 0 & A_{3,3} \end{bmatrix}$$

Suitable for speech model



- Trellis graph



Strong constraints on temporal structure can be used!

Evaluation/Alignment/Training

Evaluation

Model λ

Sequential data $\boldsymbol{x}$ → Forward/Backward algorithms → Likelihood

$$p(\boldsymbol{x} \mid \lambda) = \sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \lambda)$$

Alignment (Decoding)

Model λ

Sequential data $\boldsymbol{x}$ → Viterbi algorithm → State sequence

$$\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \lambda)$$

Training

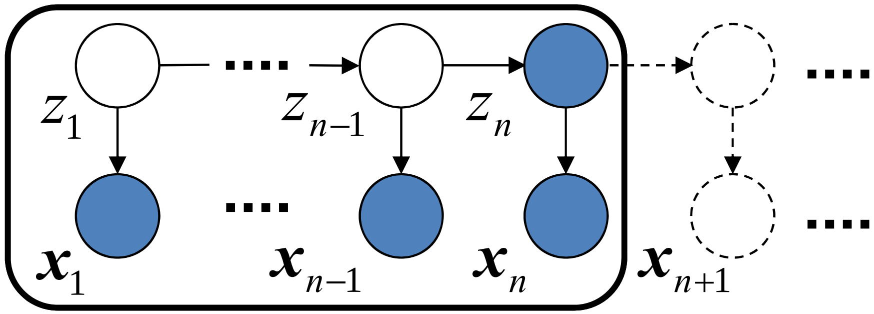
Sequential data $\boldsymbol{x}$ → Baum-Welch (i.e., EM) algorithm → Model parameter set

$$\hat{\lambda} = \arg\max_{\lambda} p(\boldsymbol{x} \mid \lambda)$$

Forward Algorithm

- Recursively calculate forward probabilities
  - Forward probability that HMM generates $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ and ends in state s

$$\alpha_n(s) = p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, z_n = s \mid \lambda)$$



Marginalizing out
all possible previous states!

  - Recursive calculation
    - Initialization ($n = 1; 1 \le s \le S$)

$$\alpha_1(s) = \pi_s B_s(\boldsymbol{x}_1)$$

    - Recursion ($2 \le n \le N; 1 \le s \le S$)

$$\alpha_n(s) = \left[\sum_{s'=1}^{S} \alpha_{n-1}(s') A_{s',s}\right] B_s(\boldsymbol{x}_n)$$

    - Termination

$$p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \mid \lambda) = \sum_{s=1}^{S} \alpha_N(s)$$

$$\begin{aligned}
\alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n) \\
&= p(\mathbf{x}_1, \ldots, \mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1} | \mathbf{z}_n) p(\mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1}) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1}) \\
&= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1})
\end{aligned}$$

ıg use of the definition (13.34) for $\alpha(\mathbf{z}_n)$, we then obtain

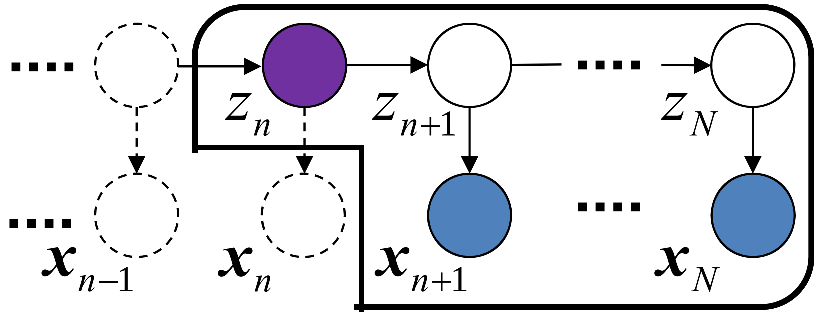
$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}).$$

Backward Algorithm

- Recursively calculate backward probabilities
  - Backward probability that HMM generates $\{x_{n+1},\cdots,x_N\}$ when starting in state s at time n

$$\beta_n(s) = p\left(x_{n+1},\cdots,x_N \mid z_n = s, \lambda\right)$$

Marginalizing out all possible succeeding states!



  - Recursive calculation
    - Initialization ($n = N; 1 \le s \le S$)

$$\beta_N(s) = 1 \qquad p(\mathbf{z}_N|\mathbf{X}) = \frac{p(\mathbf{X},\mathbf{z}_N)\beta(\mathbf{z}_N)}{p(\mathbf{X})}$$

    - Recursion ($1 \le n \le N-1; 1 \le s \le S$)

$$\beta_n(s) = \sum_{s'=1}^{S} A_{s,s'} B_{s'}(x_{n+1}) \beta_{n+1}(s')$$

$$\begin{aligned}
\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1},\ldots,\mathbf{x}_N|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1},\ldots,\mathbf{x}_N,\mathbf{z}_{n+1}|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1},\ldots,\mathbf{x}_N|\mathbf{z}_n,\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1},\ldots,\mathbf{x}_N|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n) \\
&= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2},\ldots,\mathbf{x}_N|\mathbf{z}_{n+1})p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n).
\end{aligned}$$

$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1})p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1})p(\mathbf{z}_{n+1}|\mathbf{z}_n).$$

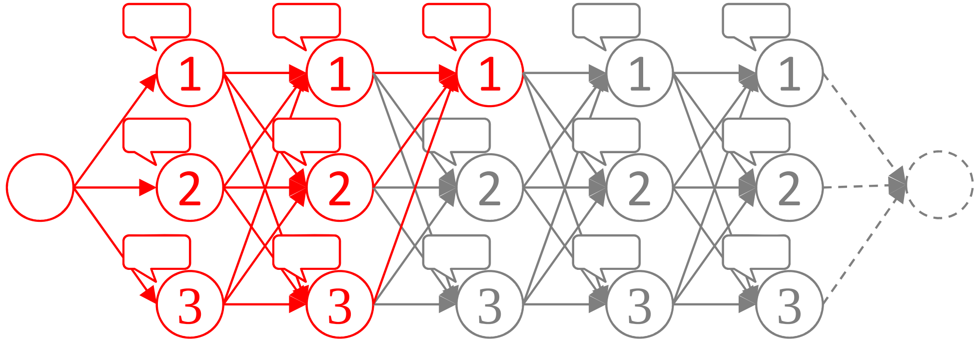
    - Termination

$$p\left(x_1,\cdots,x_N \mid \lambda\right) = \sum_{s=1}^{S} \pi_s B_s(x_1) \beta_1(s)$$

Product of Forward/Backward Probs

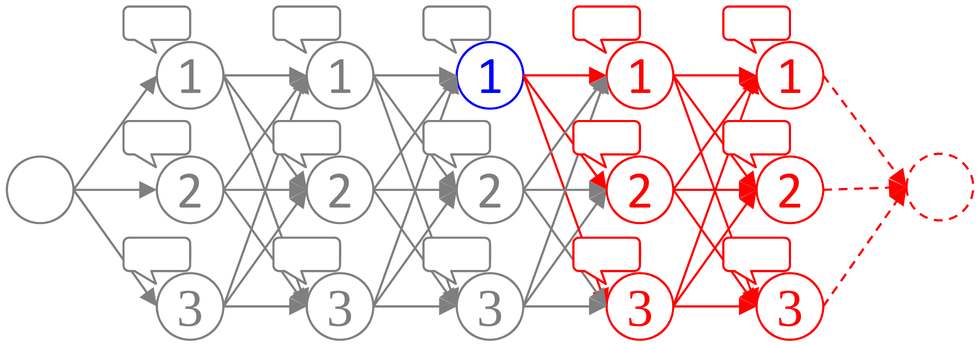
- Forward probability

$$\alpha_n(s) = p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, z_n = s \mid \boldsymbol{\lambda})$$



- Backward probability

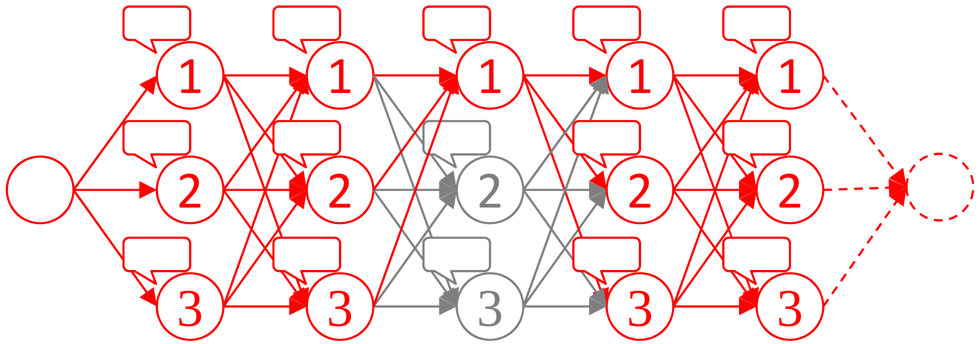
$$\beta_n(s) = p(\boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_N \mid z_n = s, \boldsymbol{\lambda})$$



- Their product

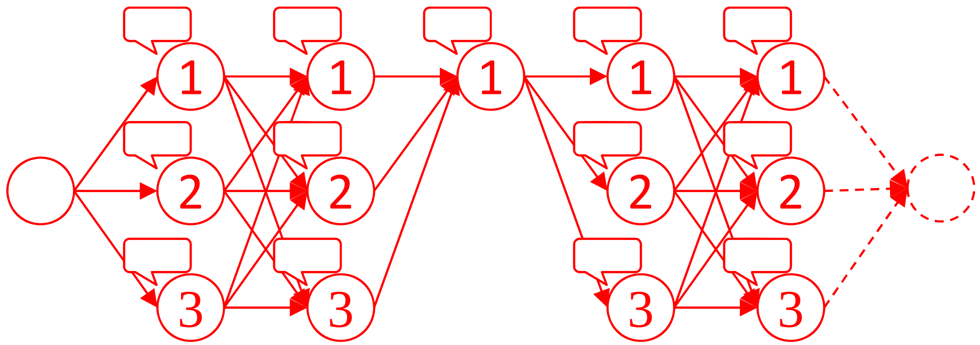
$$\alpha_n(s)\beta_n(s) = p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N, z_n = s \mid \boldsymbol{\lambda})$$

Considering all possible paths passing through state s at time n

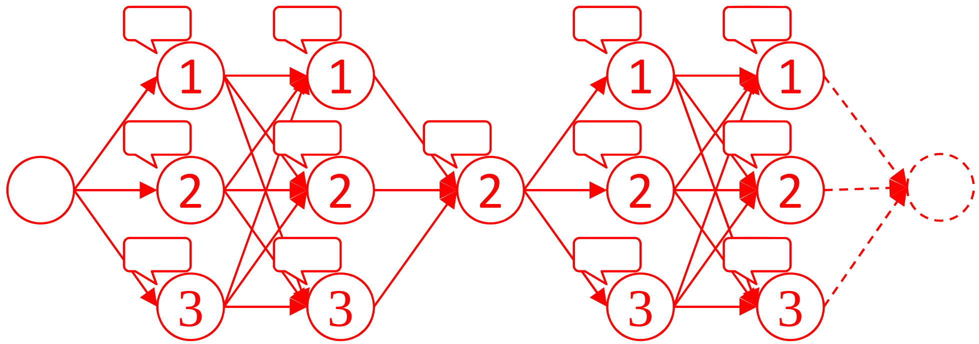


Likelihood from Forward/Backward Probs

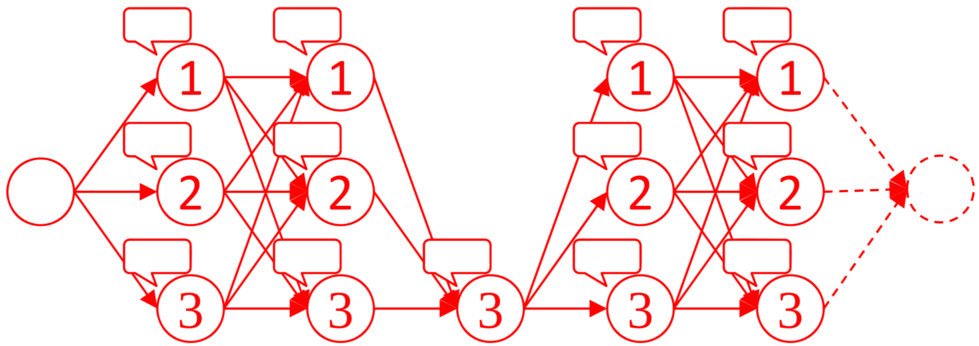
$$\alpha_n(1)\beta_n(1) = p\left(\boldsymbol{x}, z_n = 1 \mid \boldsymbol{\lambda}\right)$$



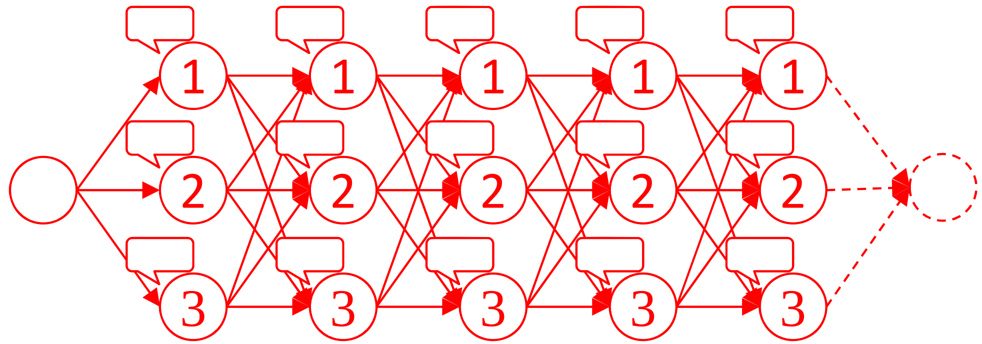
$$\alpha_n(2)\beta_n(2) = p\left(\boldsymbol{x}, z_n = 2 \mid \boldsymbol{\lambda}\right)$$



$$\alpha_n(3)\beta_n(3) = p\left(\boldsymbol{x}, z_n = 3 \mid \boldsymbol{\lambda}\right)$$



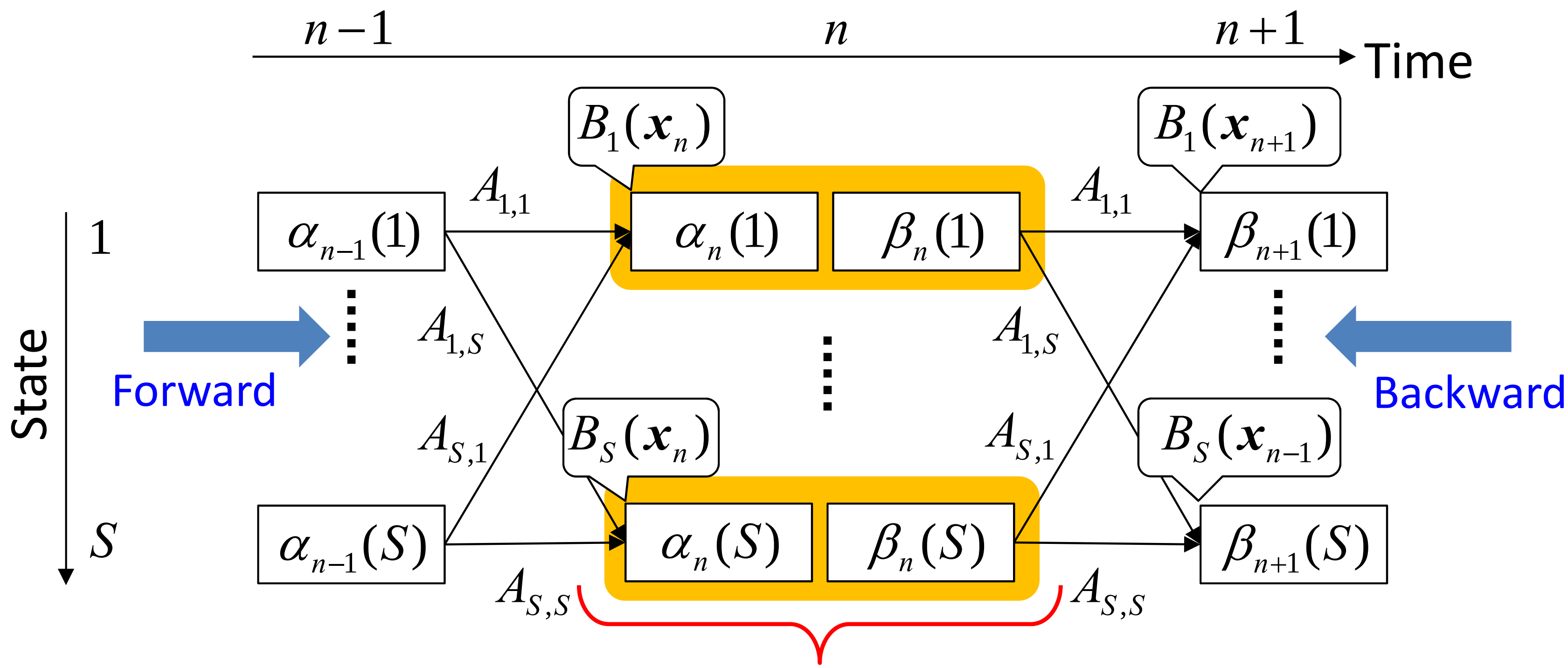
$$\sum_{s=1}^{S} \alpha_n(s)\beta_n(s) = \sum_{s=1}^{S} p\left(\boldsymbol{x}, z_n = s \mid \boldsymbol{\lambda}\right)$$
$$= p\left(\boldsymbol{x} \mid \boldsymbol{\lambda}\right)$$



Sum of the products of forward/backward probabilities at time n is equal to likelihood over all possible paths!

Likelihood Calculation

- Forward probability : $\alpha_n(s) = p\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, z_n = s \mid \boldsymbol{\lambda}\right)$
- Backward probability : $\beta_n(s) = p\left(\boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_N \mid z_n = s, \boldsymbol{\lambda}\right)$
- Their product: $\alpha_n(s)\beta_n(s) = p\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_N, z_n = s \mid \boldsymbol{\lambda}\right)$



Sum of their products at time n = Likelihood

$$\sum_{s=1}^{S} \alpha_n(s)\beta_n(s) = p\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \mid \boldsymbol{\lambda}\right)$$

App.: Derivation of Forward Algorithm

$$p(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N\mid\boldsymbol{\lambda})=\sum_{\text{all }\boldsymbol{z}}\left\{p(z_1\mid\boldsymbol{\lambda})\left[\prod_{n=2}^{N}p(z_n\mid z_{n-1},\boldsymbol{\lambda})\right]\prod_{n=1}^{N}p(\boldsymbol{x}_n\mid z_n,\boldsymbol{\lambda})\right\}$$

Factor out the common factor at $n=N$ in each state $s=s_N$

$$=\sum_{s_N=1}^{S}\left(B_{s_N}(\boldsymbol{x}_N)\underbrace{\sum_{\text{all }\boldsymbol{z}\backslash z_N}\left\{p(z_n=s_N\mid z_{N-1},\boldsymbol{\lambda})p(z_1\mid\boldsymbol{\lambda})\left[\prod_{n=2}^{N-1}p(z_n\mid z_{n-1},\boldsymbol{\lambda})\right]\prod_{n=1}^{N-1}p(\boldsymbol{x}_n\mid z_n,\boldsymbol{\lambda})\right\}}_{\text{All possible state sequences from } n=1 \text{ to } n=N-1}\right)$$

Iteratively factor out in the same manner from $n=N-1$ to $n=2$

$$=\sum_{s_N=1}^{S}\underbrace{\left(B_{s_N}(\boldsymbol{x}_N)\sum_{s_{N-1}=1}^{S}A_{s_{N-1},s_N}\underbrace{\left(B_{s_{N-1}}(\boldsymbol{x}_{N-1})\sum_{s_{N-2}=1}^{S}A_{s_{N-2},s_{N-1}}\cdots\underbrace{\left(B_{s_2}(\boldsymbol{x}_2)\sum_{s_1}^{S}A_{s_1,s_2}\underbrace{\left(\pi_{s_1}B_{s_1}(\boldsymbol{x}_1)\right)}_{\alpha_1(s_1)}\right)}_{\alpha_2(s_2)}\right)}_{\alpha_{N-1}(s_{N-1})}\right)}_{\alpha_N(s_N)}$$

$$=\sum_{s=1}^{S}\alpha_N(s)$$

App.: Derivation of Backward Algorithm

$$p(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N \mid \boldsymbol{\lambda}) = \sum_{\text{all } \boldsymbol{z}} \left\{ p(z_1 \mid \boldsymbol{\lambda}) \left[\prod_{n=2}^{N} p(z_n \mid z_{n-1}, \boldsymbol{\lambda}) \right] \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid z_n, \boldsymbol{\lambda}) \right\}$$

Factor out the common factor at $n=1$ in each state $s=s_1$

$$= \sum_{s_1=1}^{S} \pi_{s_1} B_{s_1}(\boldsymbol{x}_1) \underbrace{\sum_{\text{all } \boldsymbol{z} \setminus z_1} \left\{ p(z_2 \mid z_1 = s, \boldsymbol{\lambda}) \left[\prod_{n=3}^{N} p(z_n \mid z_{n-1}, \boldsymbol{\lambda}) \right] \prod_{n=2}^{N} p(\boldsymbol{x}_n \mid z_n, \boldsymbol{\lambda}) \right\}}_{\text{All possible state sequences from } n=2 \text{ to } n=N}$$

Iteratively factor out in the same manner from $n=2$ to $n=N-1$

$$= \sum_{s_1=1}^{S} \pi_{s_1} B_{s_1}(\boldsymbol{x}_1) \underbrace{\left(\sum_{s_2=1}^{S} A_{s_1,s_2} B_{s_2}(\boldsymbol{x}_2) \cdots \underbrace{\left(\sum_{s_{N-1}=1}^{S} A_{s_{N-2},s_{N-1}} B_{s_{N-1}}(\boldsymbol{x}_{N-1}) \underbrace{\left(\sum_{s_N=1}^{S} A_{s_{N-1},s_N} B_{s_N}(\boldsymbol{x}_N) \right)}_{\beta_{N-1}(s_{N-1})} \right)}_{\beta_{N-2}(s_{N-2})} \right)}_{\beta_1(s_1)}$$

$$= \sum_{s=1}^{S} \pi_s B_s(\boldsymbol{x}_1) \beta_1(s)$$

* Note that $\beta_N(s_N)=1$

Sequential Data Modeling

“Discrete Latent Variable Models 1”

Evaluation/Alignment/Training

Evaluation

Model λ

Sequential data $\boldsymbol{x}$ → Forward/Backward algorithms → Likelihood

$$p(\boldsymbol{x} \mid \lambda) = \sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \lambda)$$

Alignment (Decoding)

Model λ

Sequential data $\boldsymbol{x}$ → Viterbi algorithm → State sequence

$$\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \lambda)$$

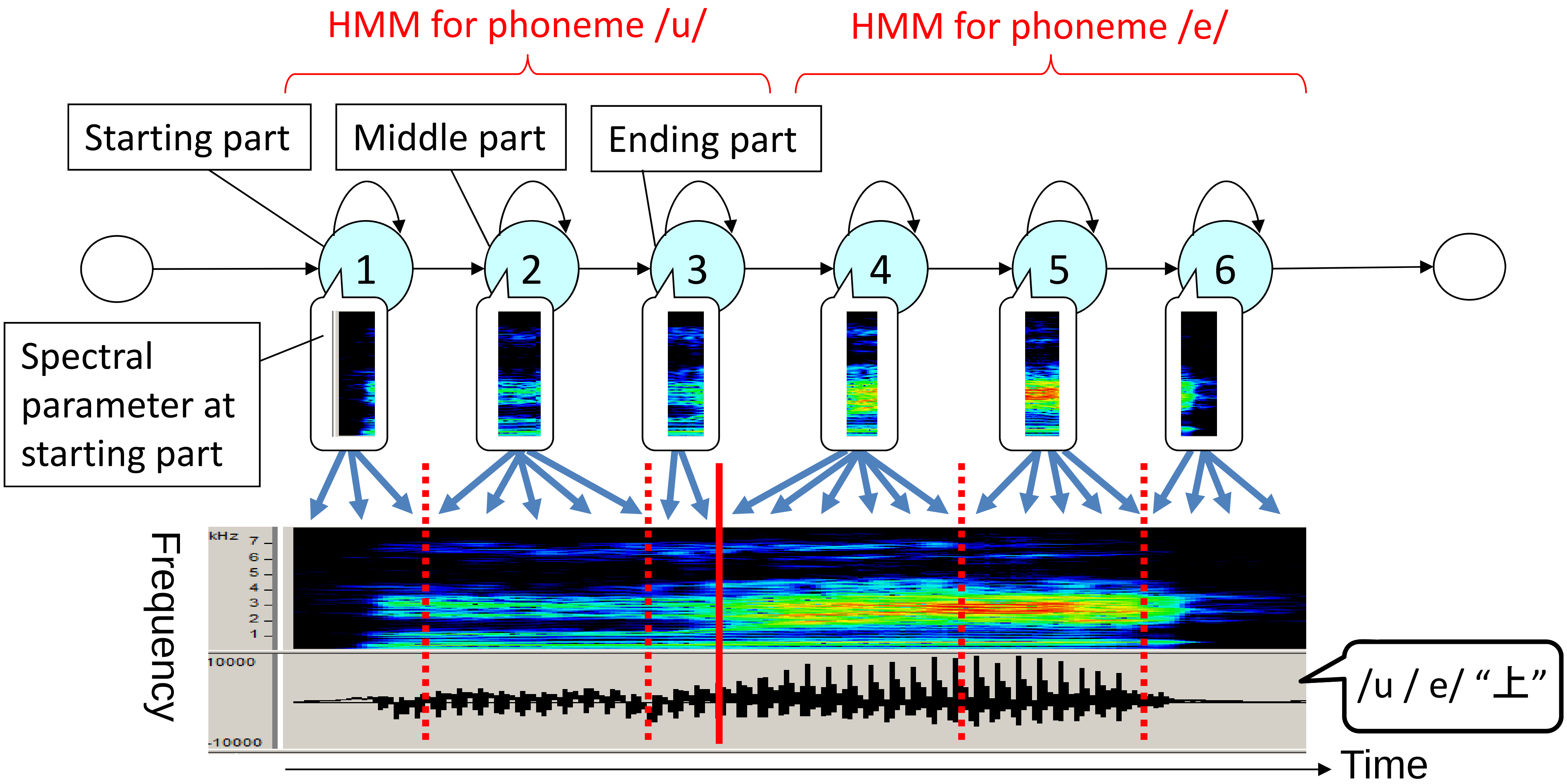
Training

Sequential data $\boldsymbol{x}$ → Baum-Welch (i.e., EM) algorithm → Model parameter set

$$\hat{\lambda} = \arg\max_{\lambda} p(\boldsymbol{x} \mid \lambda)$$

Alignment

- Find the best state sequence for a given sequential data sample
  e.g., find phoneme boundaries over a speech signal



Speech waveform and its spectral sequence

Viterbi Algorithm

- Recursively calculate the best path probability
  - The most likely path probability that generates $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ and ends in state s

$$\chi_n(s) = p\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \hat{z}_1, \cdots, \hat{z}_{n-1}, \hat{z}_n = s \mid \boldsymbol{\lambda}\right)$$

  - Recursive calculation

➢ **Initialization** ($n = 1; 1 \le s \le S$)

$$\chi_1(s) = \pi_s B_s(\boldsymbol{x}_1) \qquad \phi_1(s) = /\text{s}/$$

➢ **Recursion** ($2 \le n \le N; 1 \le s \le S$)

$$\chi_n(s) = B_s(\boldsymbol{x}_n) \max_{s'} \chi_{n-1}(s') A_{s',s}$$

$$\phi_n(s) = \arg\max_{s'} \chi_{n-1}(s') A_{s',s}$$

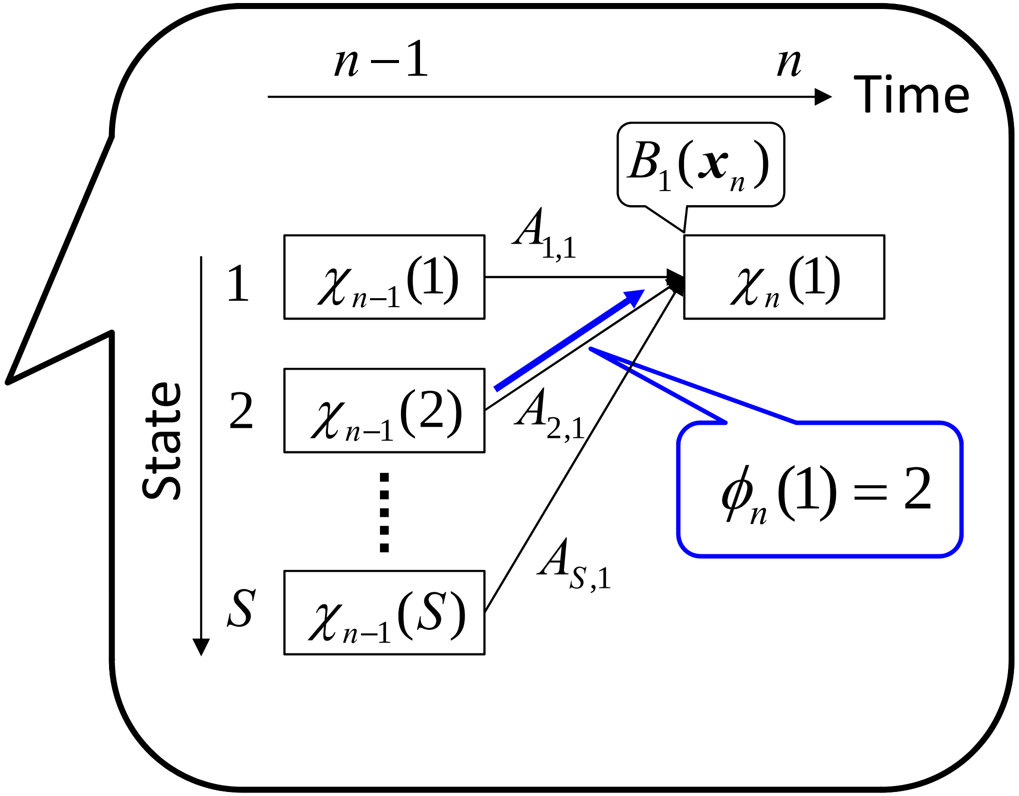
➢ **Termination**

$$p\left(\boldsymbol{x}, \hat{\boldsymbol{z}} \mid \boldsymbol{\lambda}\right) = \max_s \chi_N(s)$$

$$\hat{z}_N = \arg\max_s \chi_N(s)$$

➢ **Path backtracking** ($2 \le n \le N$)

$$\hat{z}_{n-1} = \phi_n(\hat{z}_n)$$



Evaluation/Alignment/Training

Evaluation

Model $\boldsymbol{\lambda}$

Sequential data $\boldsymbol{x}$ → Forward/Backward algorithms → Likelihood

$$p(\boldsymbol{x} \mid \boldsymbol{\lambda}) = \sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\lambda})$$

Alignment (Decoding)

Model $\boldsymbol{\lambda}$

Sequential data $\boldsymbol{x}$ → Viterbi algorithm → State sequence

$$\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\lambda})$$

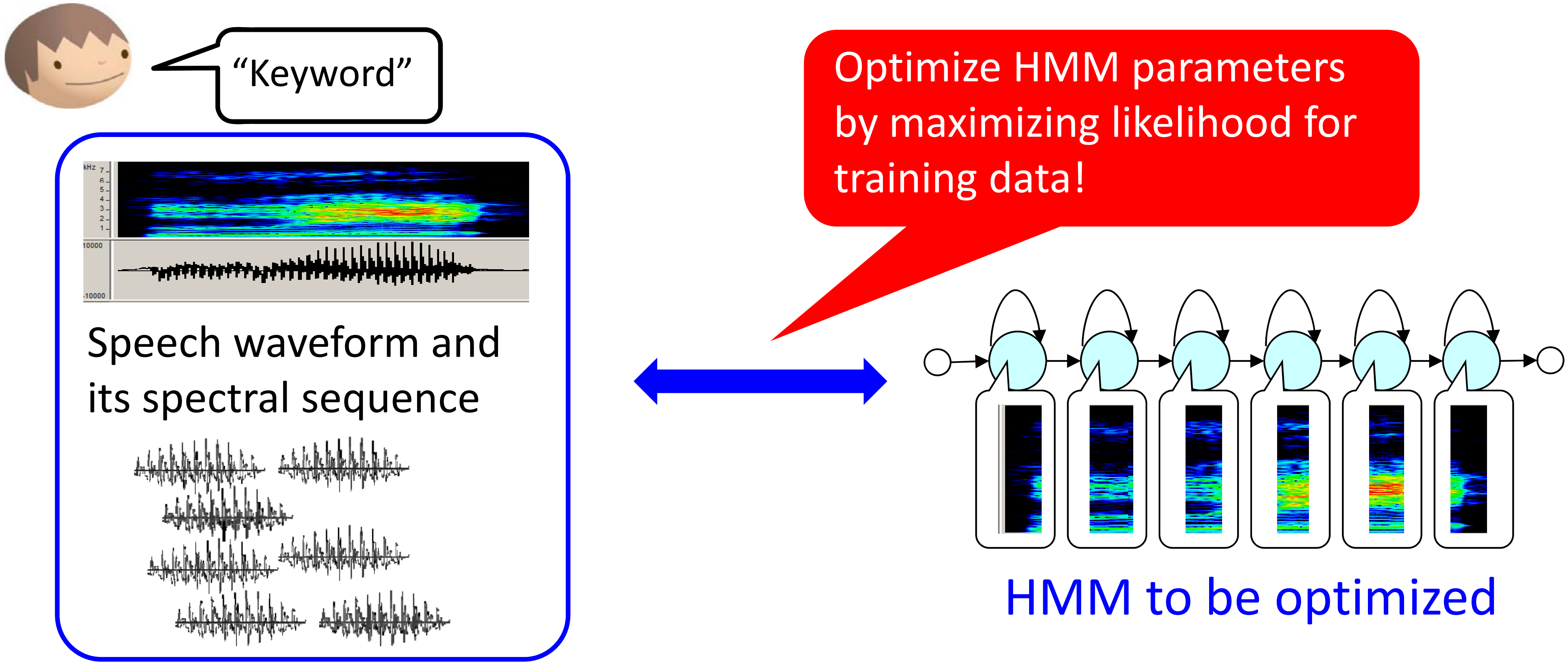
Training

Sequential data $\boldsymbol{x}$ → Baum-Welch (*i.e.*, EM) algorithm → Model parameter set

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{x} \mid \boldsymbol{\lambda})$$

Training

- Optimize HMM parameters so that the HMM appropriately models given observation sequences (*e.g.*, training the HMM from data)



Training data samples

Training with EM Algorithm

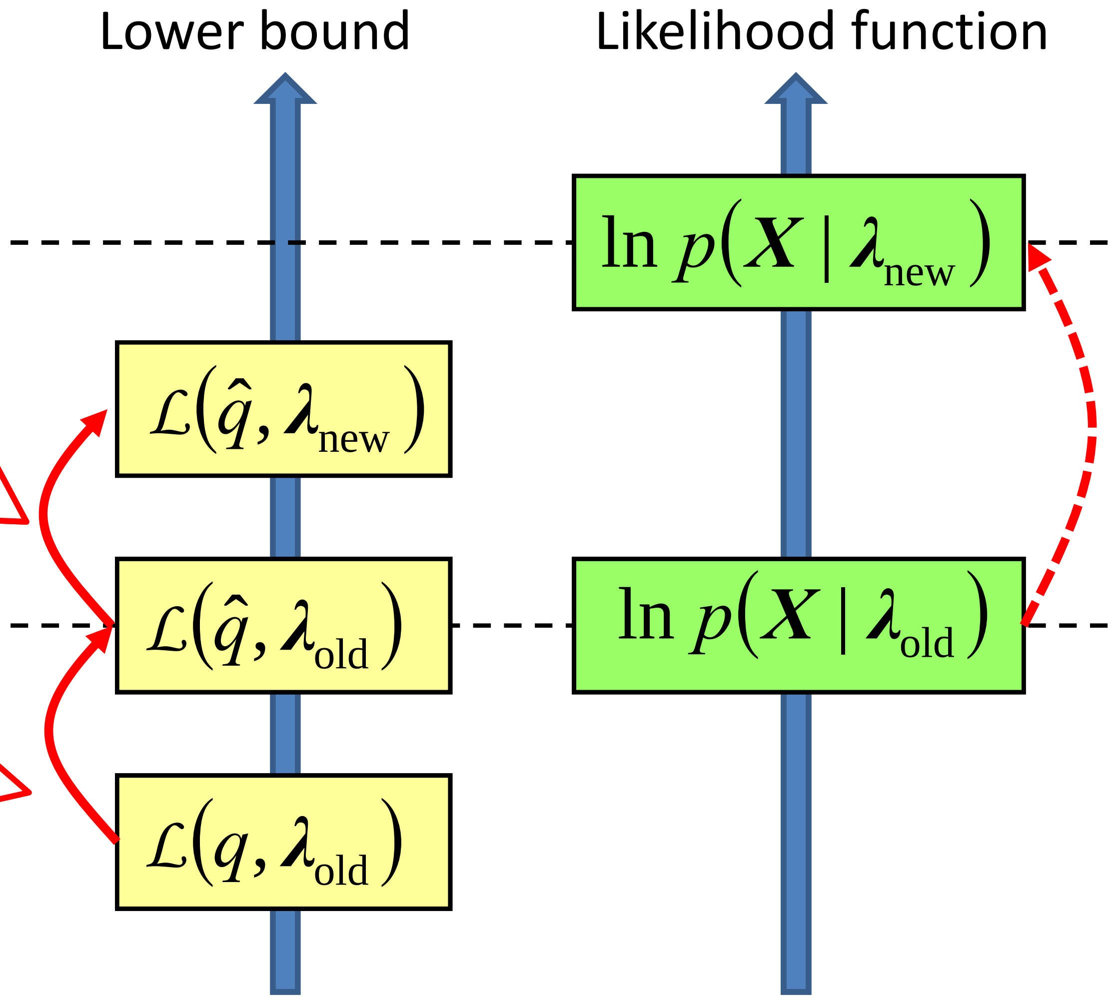
- Iteratively update lower bound of likelihood function through two steps: **E-step** and **M-step**

M-step: maximize lower bound with respect to λ while fixing q

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} \left\{ \hat{q}\left(\boldsymbol{z}^{(u)}\right) \cdot \ln p\left(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda}\right) \right\}$$

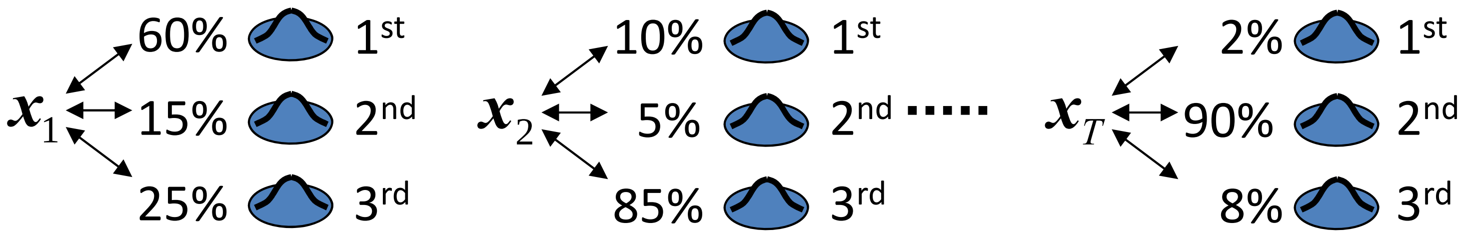
E-step: maximize lower bound with respect to q while fixing λ

$$\hat{q}\left(\boldsymbol{z}^{(u)}\right) = p\left(\boldsymbol{z}^{(u)} \mid \boldsymbol{x}^{(u)}, \boldsymbol{\lambda}\right)$$

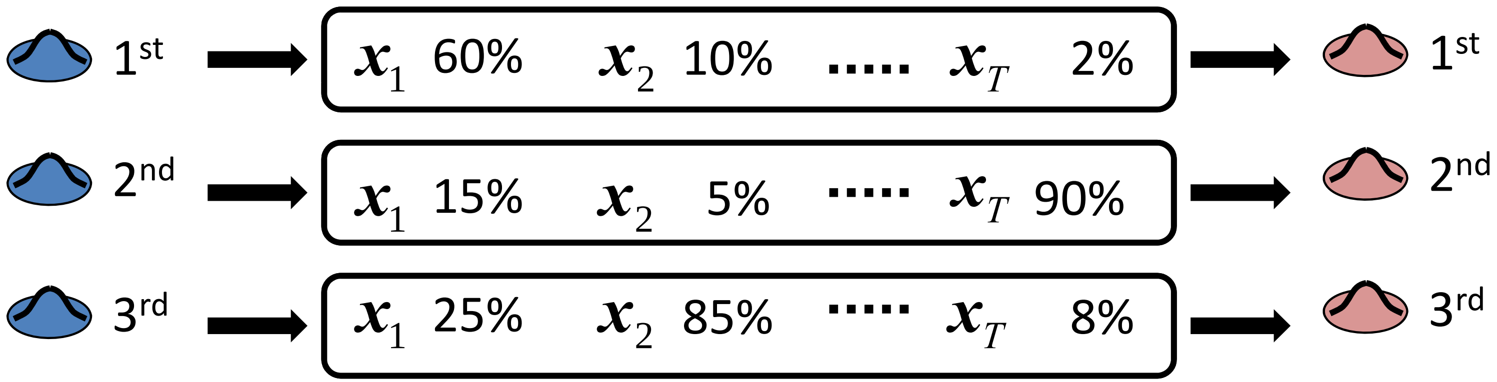


Review: Why Do We Use EM Algorithm?

- In a mixture model, we don't know which data can be used to update parameters of each mixture component due to latent variables.
- **EM algorithm** is capable of addressing this issue!
  - **E-step**: assign each data to individual mixture components (*i.e.*, estimate latent variables) based on current model parameters



  - **M-step**: update parameters of each mixture component using the assigned data



Lower Bound of HMM Likelihood

Log-scaled likelihood function for U samples of sequential data

$$\underline{\ln p\left(\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(U)} \mid \boldsymbol{\lambda}\right)} = \sum_{u=1}^{U} \ln \sum_{\text{all } \boldsymbol{z}^{(u)}} p\left(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda}\right)$$

$$\geq \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} q\left(\boldsymbol{z}^{(u)}\right) \ln \frac{p(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda})}{q(\boldsymbol{z}^{(u)})} = \underline{\mathcal{L}(q, \boldsymbol{\lambda})}$$

Lower bound

E-step: calculate posterior probabilities of latent variables ($i.e.,$ state sequences)

$$\hat{q}\left(\boldsymbol{z}^{(u)}\right) = p\left(\boldsymbol{z}^{(u)} \mid \boldsymbol{x}^{(u)}, \boldsymbol{\lambda}_{\text{old}}\right) = \frac{p(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda}_{\text{old}})}{\sum_{\text{all } \boldsymbol{z}^{(u)}} p(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda}_{\text{old}})}$$

M-step: maximize auxiliary function with respect to model parameters

$$Q\left(\boldsymbol{\lambda}_{\text{new}}, \boldsymbol{\lambda}_{\text{old}}\right) = \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}\left(\boldsymbol{z}^{(u)}\right) \ln p\left(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda}_{\text{new}}\right)$$

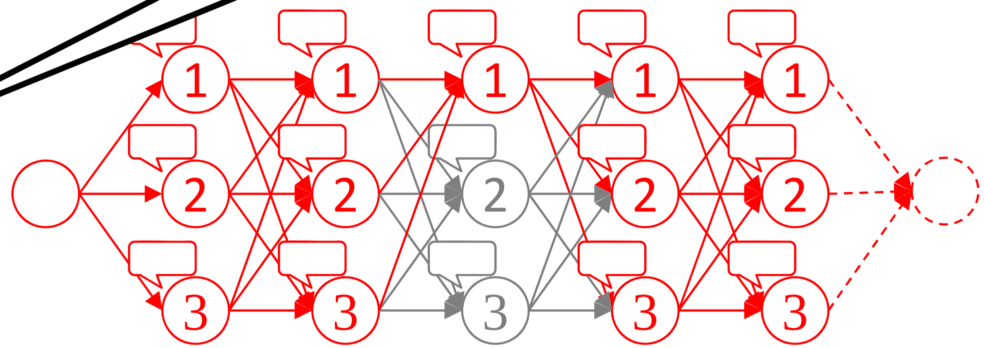
E-Step

- Calculate posterior probabilities of latent variables

Expected # of samples observed in state s at time n in sample u

$$\gamma_s^{(u)}(n) = \hat{q}\left(z_n^{(u)} = s\right) = p\left(z_n^{(u)} = s \mid \boldsymbol{x}^{(u)}, \boldsymbol{\lambda}_{\text{old}}\right) = \frac{p(\boldsymbol{x}^{(u)}, z_n^{(u)} = s \mid \boldsymbol{\lambda}_{\text{old}})}{p(\boldsymbol{x}^{(u)} \mid \boldsymbol{\lambda}_{\text{old}})}$$

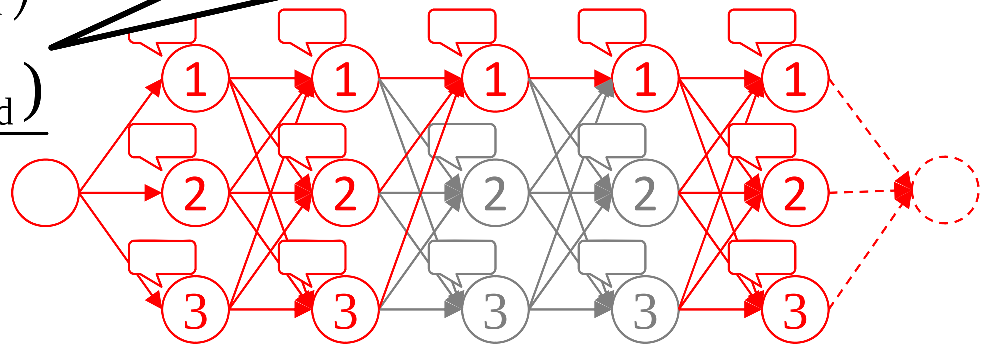
$$\alpha_n(s)\beta_n(s)$$



Expected # of samples from state s' at time $n-1$ to state s at time n in sample u

$$\xi_{s,s'}^{(u)}(n-1) = \hat{q}\left(z_{n-1}^{(u)} = s, z_n^{(u)} = s'\right) = p\left(z_{n-1}^{(u)} = s, z_n^{(u)} = s' \mid \boldsymbol{x}^{(u)}, \boldsymbol{\lambda}_{\text{old}}\right) = \frac{p(\boldsymbol{x}^{(u)}, z_{n-1}^{(u)} = s, z_n^{(u)} = s' \mid \boldsymbol{\lambda}_{\text{old}})}{p(\boldsymbol{x}^{(u)} \mid \boldsymbol{\lambda}_{\text{old}})}$$

$$\alpha_{n-1}(s) A_{s,s'} B_{s'}(\boldsymbol{x}_n) \beta_n(s')$$



Sufficient Statistics

Auxiliary function

$$Q(\lambda, \lambda_{\text{old}}) = \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}\left(\boldsymbol{z}^{(u)}\right) \ln p\left(\boldsymbol{x}^{(u)}, \boldsymbol{z}^{(u)} \mid \lambda\right)$$

$$= \sum_{s=1}^{S} \sum_{u=1}^{U} \gamma_s^{(u)}(n=1) \ln \pi_s + \sum_{s=1}^{S} \sum_{s'=1}^{S} \sum_{u=1}^{U} \sum_{n=2}^{N^{(u)}} \xi_{s,s'}^{(u)}(n-1) \ln A_{s,s'}$$

$$+ \sum_{s=1}^{S} \sum_{u=1}^{U} \sum_{\text{all o}} \sum_{n \in \{\boldsymbol{x}_n^{(u)} = \text{"o"}\}} \gamma_s^{(u)}(n) \ln B_s(\text{"o"})$$

Sum of posterior probabilities over all sequences

$$= \sum_{s=1}^{S} \gamma_s(n=1) \ln \pi_s + \sum_{s=1}^{S} \sum_{s'=1}^{S} \xi_{s,s'} \ln A_{s,s'} + \sum_{s=1}^{S} \sum_{\text{all "o"}} \gamma_s(\text{"o"}) \ln B_s(\text{"o"})$$

Sufficient statistics

Expected # of samples in state s at time n=1

$$\gamma_s(n=1) = \sum_{u=1}^{U} \gamma_s^{(u)}(n=1)$$

Expected # of samples from state s to state s'

$$\xi_{s,s'} = \sum_{u=1}^{U} \sum_{n=2}^{N^{(u)}} \xi_{s,s'}^{(u)}(n-1)$$

Expected # of samples of observing “o” in state s

$$\gamma_s(\text{"o"}) = \sum_{u=1}^{U} \sum_{n \in \{\boldsymbol{x}_n^{(u)} = \text{"o"}\}} \gamma_s^{(u)}(n)$$

M-Step

Auxiliary function

$$Q(\lambda, \lambda_{\text{old}}) = \sum_{s=1}^{S} \gamma_s(n=1) \ln \pi_s + \sum_{s=1}^{S} \sum_{s'=1}^{S} \xi_{s,s'} \ln A_{s,s'} + \sum_{s=1}^{S} \sum_{\text{all "o"}} \gamma_s(\text{"o"}) \ln B_s(\text{"o"})$$

ML estimates

$$\left. \frac{\partial \left\{ Q(\lambda, \lambda_{\text{old}}) + \varepsilon \left(1 - \sum_{s=1}^{S} \pi_s \right) \right\}}{\partial \pi_s} \right|_{\lambda = \hat{\lambda}} = 0$$

$$\left. \frac{\partial \left\{ Q(\lambda, \lambda_{\text{old}}) + \varepsilon \left(1 - \sum_{s'=1}^{S} A_{s,s'} \right) \right\}}{\partial A_{s,s'}} \right|_{\lambda = \hat{\lambda}} = 0$$

$$\left. \frac{\partial \left\{ Q(\lambda, \lambda_{\text{old}}) + \varepsilon \left(1 - \sum_{\text{all "o"}} B_s(\text{"o"}) \right) \right\}}{\partial B_s(\text{"o"})} \right|_{\lambda = \hat{\lambda}} = 0$$

For each state,

Initial state probability
$$\hat{\pi}_s = \frac{\gamma_s(n=1)}{\sum_{s=1}^{S} \gamma_s(n=1)}$$

Transition probability
$$\hat{A}_{s,s'} = \frac{\xi_{s,s'}}{\sum_{s'=1}^{S} \xi_{s,s'}}$$

Output probability
$$\hat{B}_s(\text{"o"}) = \frac{\gamma_s(\text{"o"})}{\sum_{\text{all "o"}} \gamma_s(\text{"o"})}$$

App.: Auxiliary Function

$$\sum_{\text{all } \boldsymbol{z}} p(\boldsymbol{z}) f(z_1) = \sum_{\text{all } \boldsymbol{z}} p(z_2,\cdots,z_N \mid z_1) p(z_1) f(z_1) = \sum_{\text{all } z_1} p(z_1) f(z_1)$$

$$Q(\boldsymbol{\lambda},\boldsymbol{\lambda}_{\text{old}}) = \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}(\boldsymbol{z}^{(u)}) \ln p(\boldsymbol{x}^{(u)},\boldsymbol{z}^{(u)} \mid \boldsymbol{\lambda})$$

$$= \sum_{u=1}^{U} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}(\boldsymbol{z}^{(u)}) \ln \left\{ p(z_1^{(u)} \mid \boldsymbol{\lambda}) \left[\prod_{n=2}^{N^{(u)}} p(z_n^{(u)} \mid z_{n-1}^{(u)},\boldsymbol{\lambda}) \right] \prod_{n=1}^{N^{(u)}} p(\boldsymbol{x}_n^{(u)} \mid z_n^{(u)},\boldsymbol{\lambda}) \right\}$$

$$= \sum_{u=1}^{U} \left\{ \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}(\boldsymbol{z}^{(u)}) \ln p(z_1^{(u)} \mid \boldsymbol{\lambda}) + \sum_{n=2}^{N^{(u)}} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}(\boldsymbol{z}^{(u)}) \ln p(z_n^{(u)} \mid z_{n-1}^{(u)},\boldsymbol{\lambda}) \right.$$

$$\left. + \sum_{n=1}^{N^{(u)}} \sum_{\text{all } \boldsymbol{z}^{(u)}} \hat{q}(\boldsymbol{z}^{(u)}) \ln p(\boldsymbol{x}_n^{(u)} \mid z_n^{(u)},\boldsymbol{\lambda}) \right\}$$

$$\underbrace{\sum_{s=1}^{S} \hat{q}(z_1^{(u)}=s)}_{=\gamma_s^{(u)}(n=1)} \underbrace{\ln p(z_1^{(u)}=s \mid \boldsymbol{\lambda})}_{=\pi_s}$$

$$\underbrace{\sum_{n=2}^{N^{(u)}} \sum_{s=1}^{S} \sum_{s'=1}^{S} \hat{q}(z_n^{(u)}=s, z_{n-1}^{(u)}=s')}_{=\xi_{s,s'}^{(u)}(n-1)} \cdot \underbrace{\ln p(z_n^{(u)}=s \mid z_{n-1}^{(u)}=s',\boldsymbol{\lambda})}_{=A_{s,s'}}$$

$$\sum_{n=1}^{N^{(u)}} \sum_{s=1}^{S} \underbrace{\hat{q}(z_n^{(u)}=s)}_{=\gamma_s^{(u)}(n)} \ln p(\boldsymbol{x}_n^{(u)} \mid z_n^{(u)}=s,\boldsymbol{\lambda}) = \sum_{\text{all "o"}} \sum_{n\in\{\boldsymbol{x}_n^{(u)}=\text{"o"}\}} \ln B_s(\text{"o"})$$

Hidden Markov Model Summary

Expectation Maximization

Maximum likelihood for the HMM

- Marginal likelihood

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

$$\begin{aligned} \gamma(\mathbf{z}_n) &= p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \end{aligned}$$

- EM
  - max $\quad Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) &= \sum_{k=1}^{K} \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \\ &\quad + \boxed{\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \ln p(\mathbf{x}_n|\boldsymbol{\phi}_k)}. \qquad (\end{aligned}$$

the same as GMM

Calculation

- E-step

$$\boxed{\begin{aligned}\gamma(\mathbf{z}_n) &= p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})\end{aligned}}$$

- M-step

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^{K} \gamma(z_{1j})} \qquad A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \qquad \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

Key quantities

$$\boxed{\begin{aligned} \gamma(\mathbf{z}_n) &= p(\mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \end{aligned}}$$

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{z}_n)p(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_N|\mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n)\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = \frac{\alpha(\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{z}_{n-1})\beta(\mathbf{z}_n)}{p(\mathbf{X})}$$

$$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n)\beta(\mathbf{z}_n) \quad p(\mathbf{X}) = \sum_{\mathbf{z}_N} \alpha(\mathbf{z}_N)$$

Inference

- Marginal likelihood (Forward-Backward / sum-product): $P(X)$
- Best state path (Viterbi / max-sum): $\max_Z p(X,Z)$
- Prediction: $p(x_{N+1}|X)$

Prediction

$$
\begin{aligned}
p(\mathbf{x}_{N+1}|\mathbf{X}) &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}, \mathbf{z}_{N+1}|\mathbf{X}) \\
&= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) p(\mathbf{z}_{N+1}|\mathbf{X}) \\
&= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}, \mathbf{z}_N|\mathbf{X}) \\
&= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N) p(\mathbf{z}_N|\mathbf{X}) \\
&= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N) \frac{p(\mathbf{z}_N, \mathbf{X})}{p(\mathbf{X})} \\
&= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N) \alpha(\mathbf{z}_N)
\end{aligned}
$$

Thinking 

- Recover: $p(x_n | x_1,\ldots,x_{n-1}, x_{n+1},\ldots,x_N)$

Scaling factors for HMM

- $\alpha(z_n)$ is obtained from the previous value $\alpha(z_n-1)$ by multiplying by quantities $p(z_n/z_n-1)$ and $p(x_n/z_n)$.
- $p(z_n/z_n-1)<1$ =>the values of $\alpha(z_n)$ can go to zero exponentially quickly.
- Rescale $\alpha(z_n)$ as

$$\widehat{\alpha}(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1,\ldots,\mathbf{x}_n) = \frac{\alpha(\mathbf{z}_n)}{p(\mathbf{x}_1,\ldots,\mathbf{x}_n)} \qquad c_n = p(\mathbf{x}_n|\mathbf{x}_1,\ldots,\mathbf{x}_{n-1})$$

- Then

$$\alpha(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_1,\ldots,\mathbf{x}_n)p(\mathbf{x}_1,\ldots,\mathbf{x}_n) = \left(\prod_{m=1}^{n} c_m\right)\widehat{\alpha}(\mathbf{z}_n)$$

Rewrite forward-backward recursion

- Forward (c_n the coefficient that normalizes the right side)

$$c_n \widehat{\alpha}(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \widehat{\alpha}(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1}) \qquad \alpha(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n|\mathbf{z}_{n-1})$$

- Backward

$$c_{n+1} \widehat{\beta}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \widehat{\beta}(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n) \qquad \beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{z}_{n+1}) p(\mathbf{z}_{n+1}|\mathbf{z}_n)$$

- Marginal

$$\begin{aligned} \gamma(\mathbf{z}_n) &= \widehat{\alpha}(\mathbf{z}_n) \widehat{\beta}(\mathbf{z}_n) \\ \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= c_n \widehat{\alpha}(\mathbf{z}_{n-1}) p(\mathbf{x}_n|\mathbf{z}_n) p(\mathbf{z}_n|\mathbf{z}_{-1}) \widehat{\beta}(\mathbf{z}_n) \end{aligned}$$

HMM Extensions

1. Generalize model of state duration:

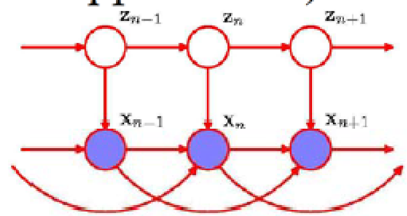
   - vanilla HMM restricted in model of how long stay in state - prob. that model will spend D steps in state k and then transition out:

$$P(D) = (A_{kk})^D (1 - A_{kk}) \propto \exp(-D \log A_{kk})$$

   - instead associate distribution with time spent in state k: P($t|k$) (see *semi-Markov* models for sequence segmentation applications)

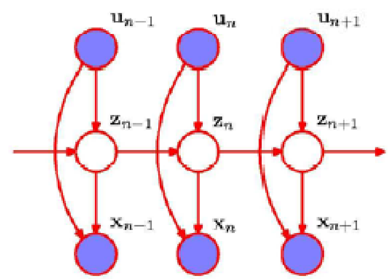
2. Combine with auto-regressive Markov model:

   - include long-range relationships
   - directly model relations between observations



3. Supervised setting:

   - include additional observations
   - *input-output* HMM



$$z_{n+1} \perp\!\!\!\perp z_{n-1} \mid z_n$$

4 *factorial hidden Markov model* (Ghahramani and Jordan, 1997)

- reduce latent states
- add Markov chains
- no conditional independence