Hidden Markov Models

Eklavya SARKAR
Biometrics Security and Privacy, Idiap

Table of Contents

Introduction

L. R. Rabiner

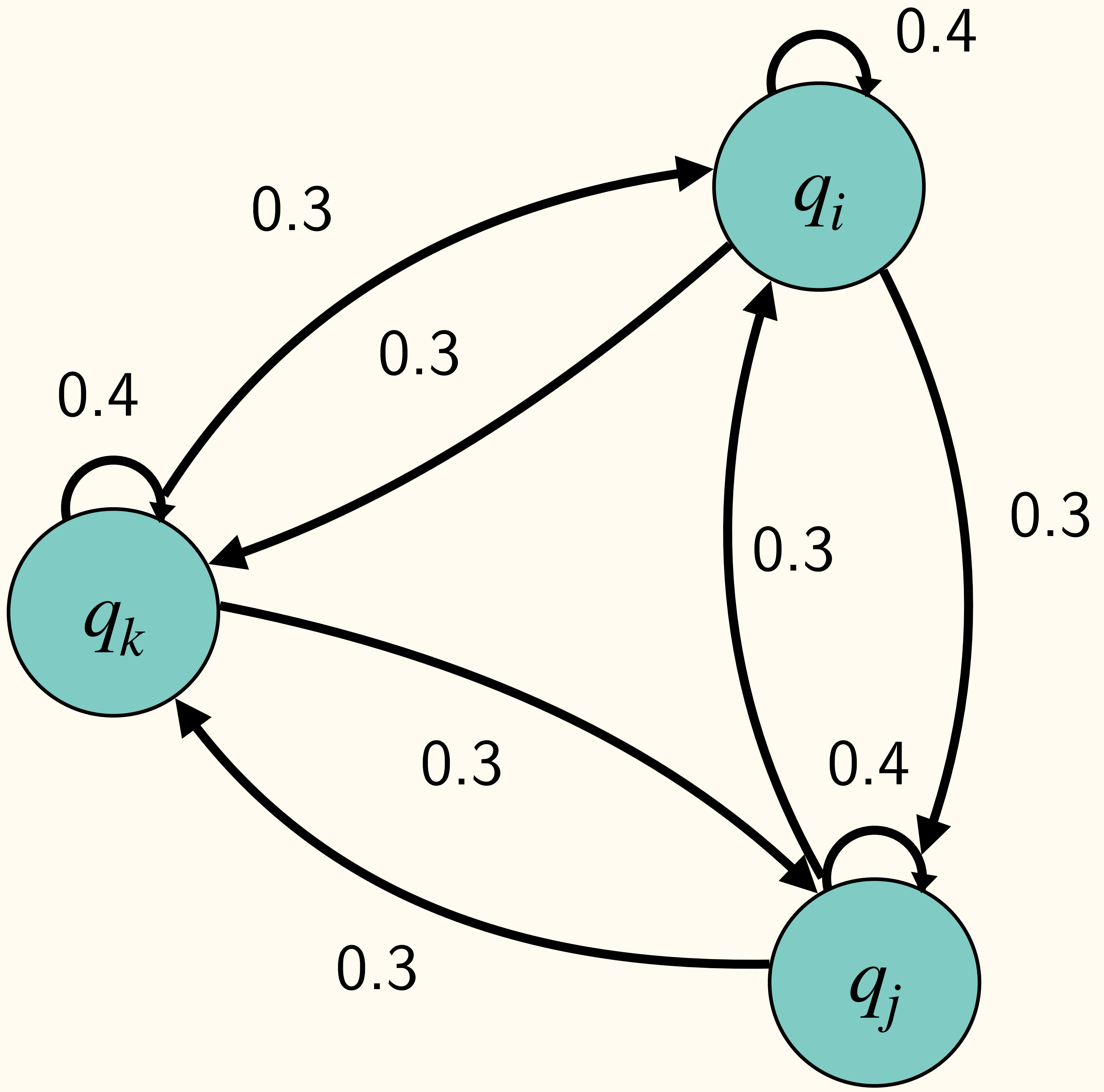
- Sequence processing:
  - Input: sequence X
  - Goal: estimate a sequence of outputs M
  - $P(M|X)$

- Tool: Hidden Markov Models (HMMs)
  - Introduced and studied in 1960-70s
  - Lawrence R. Rabiner. *A tutorial on Hidden Markov Models and selected applications in speech recognition.*

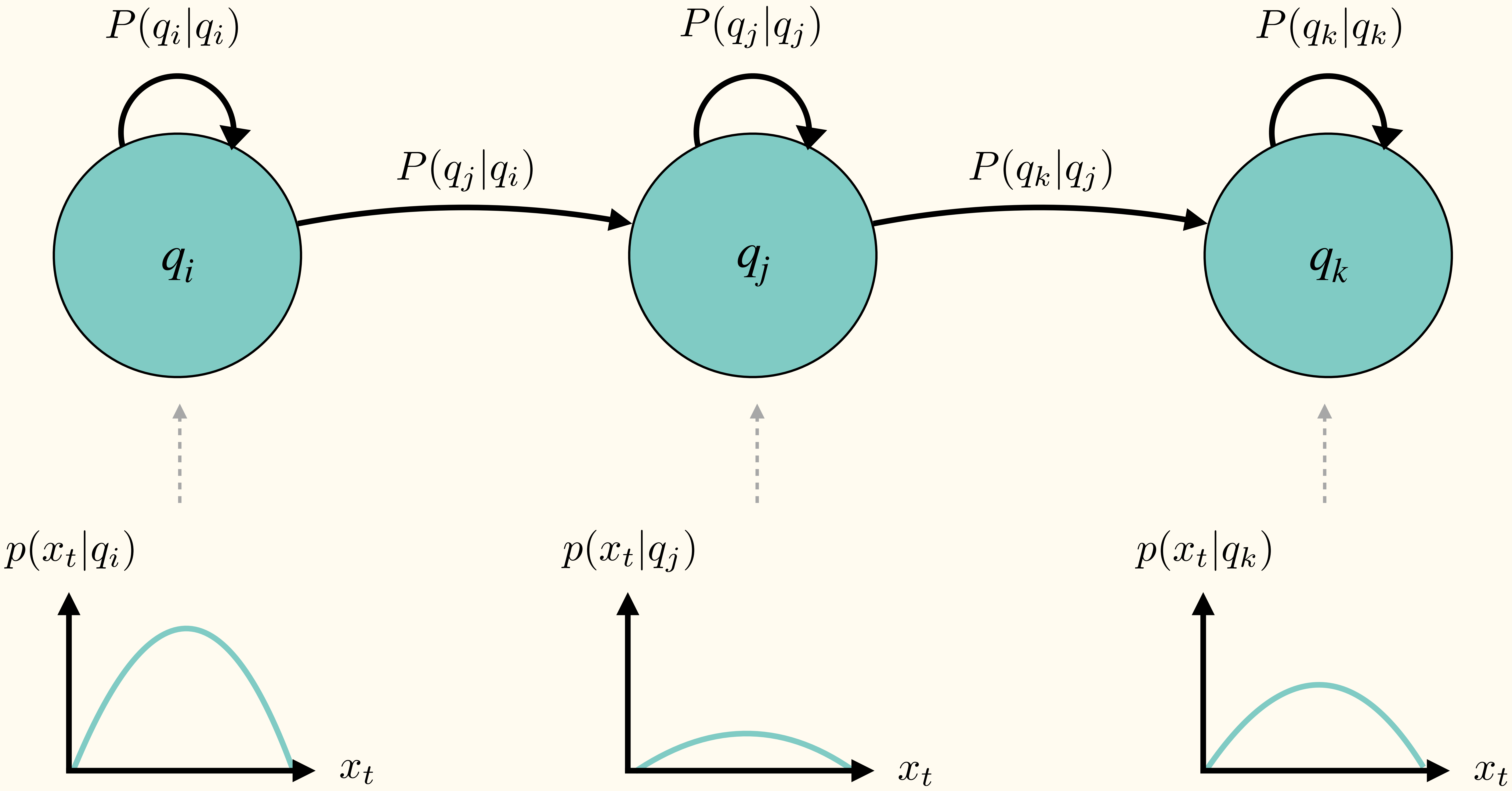
Discrete Markov Models (DMMs)

- Model M_k
- Composed of states $Q = \{q_1, \ldots, q_k, \ldots, q_K\}$
- q_j^t denotes state q_j at time t

- First-order Markov Models
- Time independent



Hidden Markov Models (HMMs)



HMMs

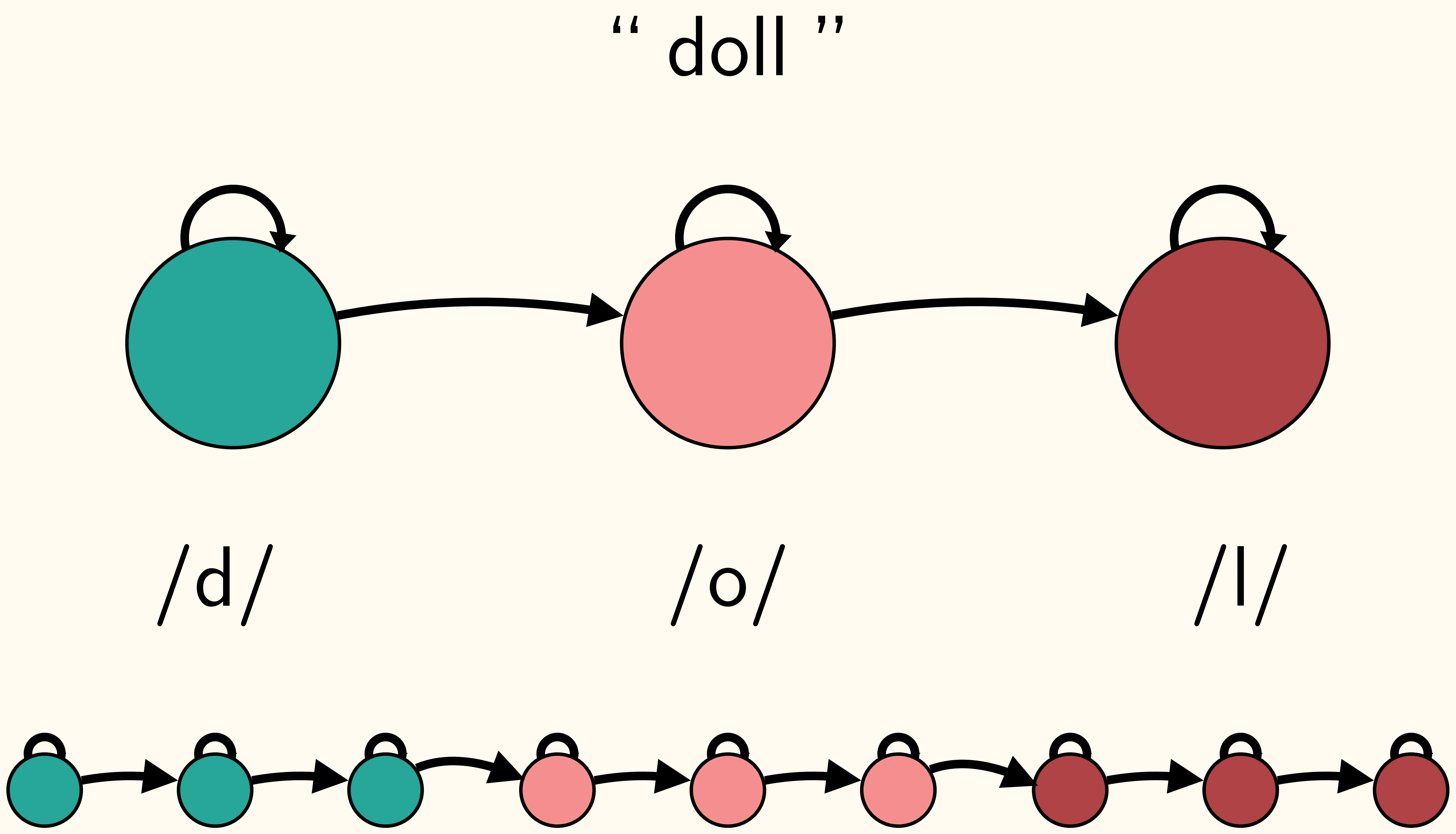
- Sequence of observations: $X = \{x_1, \ldots, x_t, \ldots, x_T\}$
- Sequence of states: $Q = \{q_1, \ldots, q_k, \ldots, q_K\}$, q_j^t is state a q_j at time t
- Transition probabilities: $A = \{a_{ij}\} : a_{ij} = P(q_j | q_i), \qquad 1 \leq i, j \leq K$
- Emission probabilities: $B = \{b_i(x)\} : b_i(x) = p(x | q_i), \qquad 1 \leq i \leq K$
- Initial state distribution: $\pi = \{\pi_i\} : \pi_i = P(I | q_j), \qquad 1 \leq j \leq K$

$$\Theta = \{\pi, A, B\}$$

- Observations now also described by emission probabilities, characterized by different stochastic distributions for each state q_i, $i \in [1, \ldots, K]$.
  - Discrete, Gaussians, GMMs, ANNs (MLPs, or RNNs).

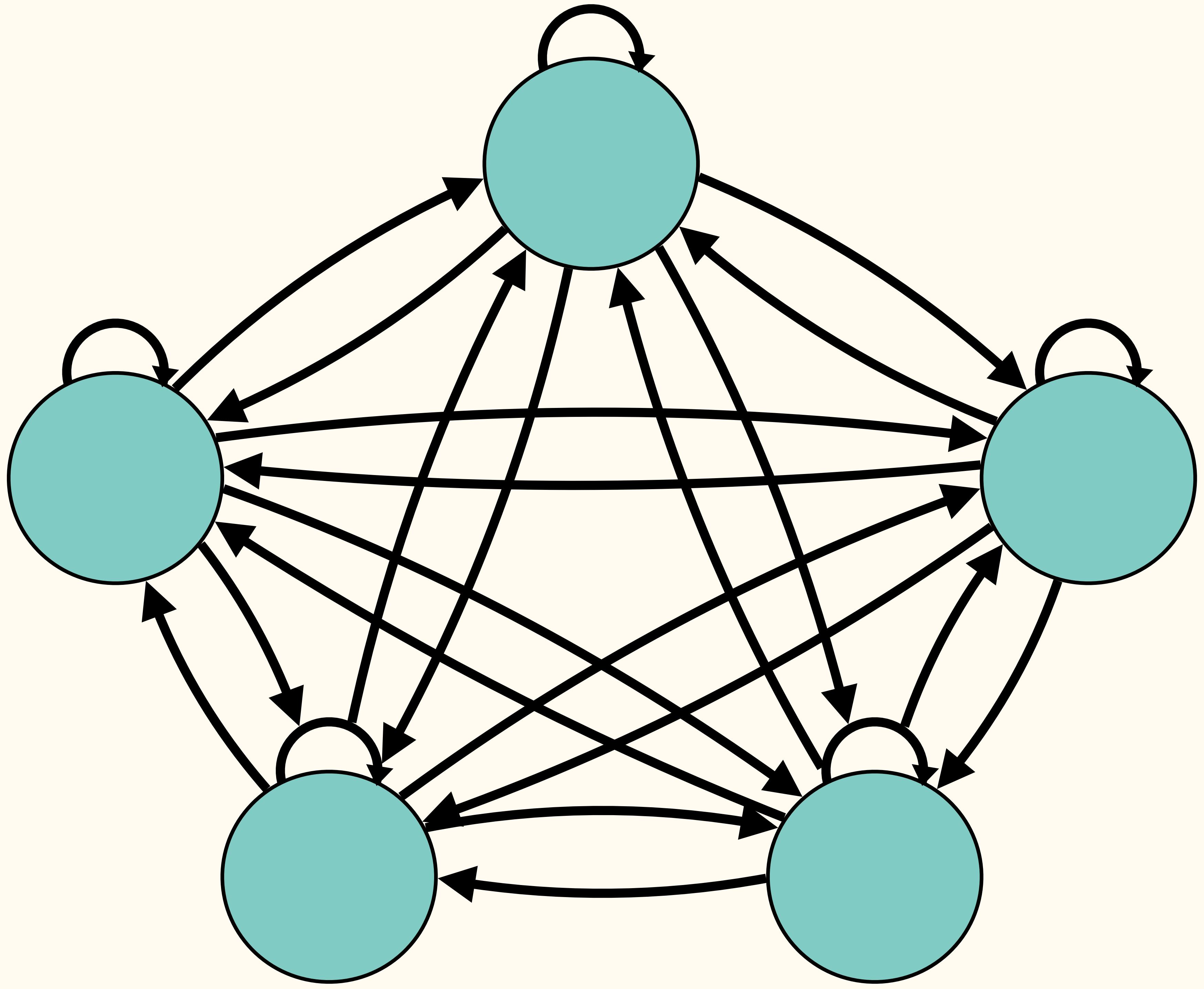
HMMs Topologies

Left-to-right model:



Speech recognition

Ergodic model:



Speaker identification

HMM-based Pattern Classification

Bayes Theorem

$$P(M|X,\Theta) = \frac{p(X|M,\Theta)\ P(M|\Theta)}{p(X|\Theta)}$$

- M: Sequential (sentence) model
- Θ: Model Parameters

- $P(X, M\,|\,\Theta)$: HMM (acoustic model)
- $P(X\,|\,\Theta)$: Assumed constant
- $P(M\,|\,\Theta)$: Prior knowledge (language model). $P(M|\Theta) \Rightarrow P(M|\Theta^*)$

Three HMM Problems

1. Definition and estimation of transition a_{ij} and emission $b_i(x)$ probabilities:
   - Computing likelihood $P(X|M, \Theta)$ for a given M_k and fixed Θ

2. Training a HMM:
   - Estimating Θ such that: $\operatorname{argmax}_{\Theta} \prod_{j=1}^{J} P(X_j|M_j, \Theta)$

3. Classification (decoding) of an observed sequence X:
   - $X \in M_j \quad \text{if} \quad M_j = \operatorname{argmax}_{M_k} P(X|M_k, \Theta)\, P(M_k)$

Likelihood Problem

Likelihood Estimation Problem

$$P(M|X,\Theta) = \frac{p(X|M,\Theta)\ P(M|\Theta)}{p(X|\Theta)}$$

- Computing $P(X|M,\Theta)$
- Fixed Θ
- Likelihood of a sequence of observations w.r.t. a HMM:
- Complexity: $\mathcal{O}(TK^T)$
  - Infeasible !

$$P(X|M) = \sum_{Q\in M} P(X,Q|M)$$

Bayes Theorem

$$= \sum_{Q\in M} P(X|Q,M)P(Q|M)$$

$$= \sum_{Q\in M} \prod_{t=1}^{T} p(x_t|q^t) \prod p_{q^{t-1},q^t}$$

$$= \sum_{Q\in M} \prod_{t=1}^{T} p(x_t|q^t)\, p_{q^{t-1},q^t}$$

Forward Recurrence

We define the following variable:

- $\alpha_t(i) = p(x_1, \ldots, x_t, q^t = q_i \,|\, \Theta)$

i.e. the probability of having observed the partial sequence $\{x_1, \ldots, x_t\}$ and being at state i at time t, given the parameters Θ.

- Complexity: $\mathcal{O}(TK^2)$
  - Bounded !

1. Initialization:
   - $\alpha_1(i) = \pi_i \, b_i(x_1), \quad 1 \leq i \leq K$

2. Recursion:
   - $$\alpha_{t+1}(j) = [\sum_{i=1}^{K} \alpha_t(i)\, a_{ij}]\, b_j(x_{t+1})$$

3. Termination:
   - $$P(X \,|\, \Theta) = \sum_{i=1}^{K} \alpha_T(i)$$

Forward Recurrence - Log Space

1. Initialization:
   - $$\alpha_1(i) = \pi_i \, b_i(x_1), \quad 1 \leq i \leq K$$

2. Recursion:
   - $$\alpha_{t+1}(j) = [\sum_{i=1}^{K} \alpha_t(i) \, a_{ij}] \, b_j(x_{t+1})$$

3. Termination:
   - $$P(X|M) = \sum_{i=1}^{K} \alpha_T(i)$$

1. Initialization:
   - $$\alpha_1^{(\log)}(i) = \log \pi_i + \log b_i(x_1)$$

2. Recursion:
   - $$\alpha_{t+1}^{(\log)}(j) = [\text{logsum}_{i=1}^{K}(\alpha_t^{(\log)}(i) + \log a_{ij})] + \log b_j(x_{t+1})$$

3. Termination:
   - $$\log P(X|M) = [\text{logsum}_{i=1}^{K} \alpha_T^{(\log)}(i)]$$

Training Problem

HMM Training Problem

- We want to accurately estimate parameters from the 'visible' sequence of observations.
- 'Training' an HMM means finding these parameters Θ.

- We use the Forward-Backward algorithm, with the following variables:
  - Forward variable $\alpha_t(i)$
  - Backward variable $\beta_t(i)$
  - Sequence of events $\xi_t(i, j)$
  - Gamma variable $\gamma_t(i)$

Backward Algorithm

We define the following variable:

- $\beta_t(i) = p(x_1, \ldots, x_t \,|\, q^t = q_i, \Theta)$

i.e. the probability of having observed the partial sequence $\{x_1, \ldots, x_t\}$, given the state i at time t and the parameters Θ.

- Complexity: $\mathcal{O}(TK^2)$

1. Initialization:
   - $\beta_T(i) = 1$
2. Recursion:
   - $\beta_t(j) = [\sum_{i=1}^{K} \beta_{t+1}(i)\, a_{ij}]\; b_j\;(x_{t+1})$
3. Termination:
   - $\beta_0 = P(X \,|\, \Theta) = \sum_{i=1}^{K} \pi_i\, b_i(x_1)\, \beta_1(i)$

Sequence of Events

Forward Backward

We define the following variable:

- $\xi_t(i,j) = P(q^t = q_i, q^{t+1} = q_j | X, \Theta)$

i.e. the probability of being in state i at time t and in state j at time $t+1$, given the observations and parameters Θ.

Can be expressed in terms of both forward and backward variables as:

$$\xi_t(i,j) = \frac{P(q_i^t, q_j^{t+1}, X | \Theta)}{P(X | \Theta)}$$

$$= \frac{a_t(i)\, a_{ij}\, b_j(x_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{K} \sum_{j=1}^{K} \alpha_t(i) a_{ij}\, b_j(x_{t+1}) \beta_{t+1}(j)}$$

Gamma Variable

Forward Backward

We define the following variable:

- $\gamma_t(i) = P(q^t = q_i | X, \Theta)$

i.e. the probability of being in state i at time t, given the observations and parameters Θ.

Can be expressed in terms of both forward and backward variables as:

$$\gamma_t(i) = \frac{P(q_i^t, X | \Theta)}{P(X | \Theta)} = \frac{\alpha_t(i)\,\beta_t(i)}{P(X | \Theta)}$$

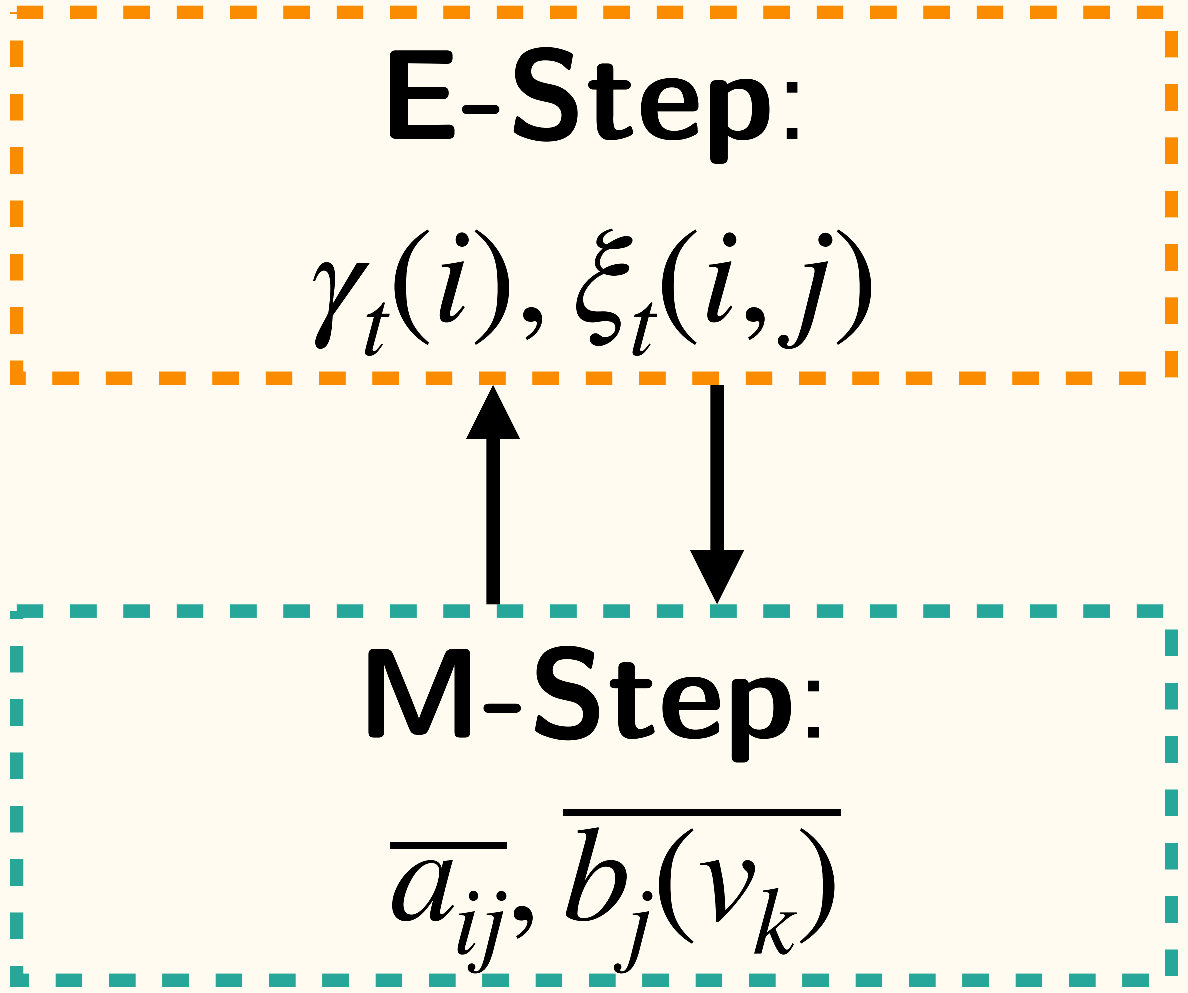
Estimator Formulas

We define the following formulas, as estimators for the:

- Initial state: $\overline{\pi_i} = \gamma_1(i)$ ← Expected frequency in state q_i at time $t = 1$
- Transition probabilities: $\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$
  - Numerator ← Expected number of transitions from state q_i to q_j
  - Denominator ← Expected number of transitions from state q_i
- Emission probabilities: $\overline{b_j(v_k)} = \frac{\sum_{t=1 \& x_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$
  - Numerator ← Expected number of times in state q_j and observing v_k
  - Denominator ← Expected number of times in state q_j

Baum-Welch Algorithm

- New values $\overline{a_{ij}}$ and $\overline{b_j(v_k)}$
  - Re-compute α_t, β_t, γ_t, ξ_t
    - New values $\overline{a_{ij}}$ and $\overline{b_j(v_k)}$
      - ...



- Iterate through this forward-backward (Baum-Welch) EM algorithm.
  - Until **convergence**.

Decoding Problem

Decoding Problem

- Estimating an optimal sequence of states given a sequence of observations and the parameters of a model.

    - Viterbi algorithm

Viterbi Algorithm

We define 2 variables:

1. $\delta_t(i)$: highest likelihood along a side path among all paths ending in state q_i at time t:
   - $\delta_t(i) = \max P[q^1, \ldots, q_i^t, x^1, \ldots, x^t \,|\, \Theta]$
   - Similar to the forward algorithm's $\alpha_t(i) = p(x_1, \ldots, x_t, q^t = q_i \,|\, \Theta)$
2. $\psi_t(i)$: variable to keep track of 'best path' ending in state q_i at time t:
   - $\psi_t(i) = \text{argmax } \; p(q^1, \ldots, q_i^t, x^1, \ldots, x^t \,|\, \Theta)$

Viterbi Algorithm

1. Initialization:
   - $\delta_1(i) = \pi_i \, b_i(x_1)$
   - $\psi_1(i) = 0$

2. Recursion:
   - $\delta_t(j) = \max_{1 \le i \le K} [\delta_{t-1}(i) \, a_{ij}] \, b_j(x_t)$
   - $\psi_t(j) = \text{argmax}_{1 \le i \le K} [\delta_{t-1}(i) \, a_{ij}]$

3. Termination:
   - $P^*(X \,|\, \Theta) = \max_{1 \le i \le K} \delta_T(i)$
   - $q_T^* = \text{argmax}_{1 \le i \le K} [\delta_T(i)]$

4. Backtracking:
   - $q^{t*} = \psi_{t+1}(q^{t+1*})$

Viterbi Algorithm

1. Initialization:
   - $\delta_1(i) = \pi_i b_i(x_1)$
   - $\psi_1(i) = 0$
2. Recursion:
   - $\delta_t(j) = \max_{1 \le i \le K} [\delta_{t-1}(i)\, a_{ij}]\, b_j(x_t)$
   - $\psi_t(j) = \text{argmax}_{1 \le i \le K} [\delta_{t-1}(i)\, a_{ij}]$
3. Termination:
   - $P^*(X \mid \Theta) = \max_{1 \le i \le K} \delta_T(i)$
   - $q_T^* = \text{argmax}_{1 \le i \le K} [\delta_T(i)]$
4. Backtracking:
   - $q^{t*} = \psi_{t+1}(q^{t+1*})$

Viterbi Algorithm - Log Space

1. Initialization:
   - $\delta_1^{(\log)}(i) = \log \pi_i + \log b_i(x_1)$
   - $\psi_1(i) = 0$

2. Recursion:
   - $\delta_t^{(\log)}(i) = \max\limits_{1 \le i \le K} [\delta_{t-1}^{(\log)}(i) + \log a_{ij}] + \log b_j(x_t)$
   - $\psi_t(j) = \text{argmax}_{1 \le i \le K}[\delta_{t-1}^{(\log)}(i) + \log a_{ij}]$

3. Termination:
   - $\log P^*(X \mid \Theta) = \max\limits_{1 \le i \le K} \delta_T^{(\log)}(i)$
   - $q_T^* = \text{argmax}_{1 \le i \le K}[\delta_T^{(\log)}(i)]$

4. Backtracking:
   - $q^{t^*} = \psi_{t+1}(q^{t+1^*})$

Viterbi Algorithm

In summary, given a:

- Sequence of observations $X = \{x_1, \ldots, x_n, \ldots x_T\}$
- Parameters Θ

The Viterbi algorithm returns the:

- Optimal path $Q* = \{q_1^*, \ldots, q_T^*\}$
- Likelihood along the best path $P*(X|\Theta)$

Solved !

Summary

Pros:

- Flexible topology.
- Rich mathematical framework.
- Wide range of applications.
- Powerful learning and decoding methods.
- Good abstraction for sequences, temporal aspects.

Cons:

- A priori selection of model topology and statistical distributions.
- First order Markov model for state transition.
- Lack of contextual information as correlation between successive acoustic vectors is ignored.
- Assumption of independence for computational efficiency.

Thank you !

Room 207-2, Idiap Research Institute

www.idiap.ch/~esarkar/

+41 78 82 50 754

eklavya.sarkar@idiap.ch