

Deep Generative Models: Hidden Markov Models

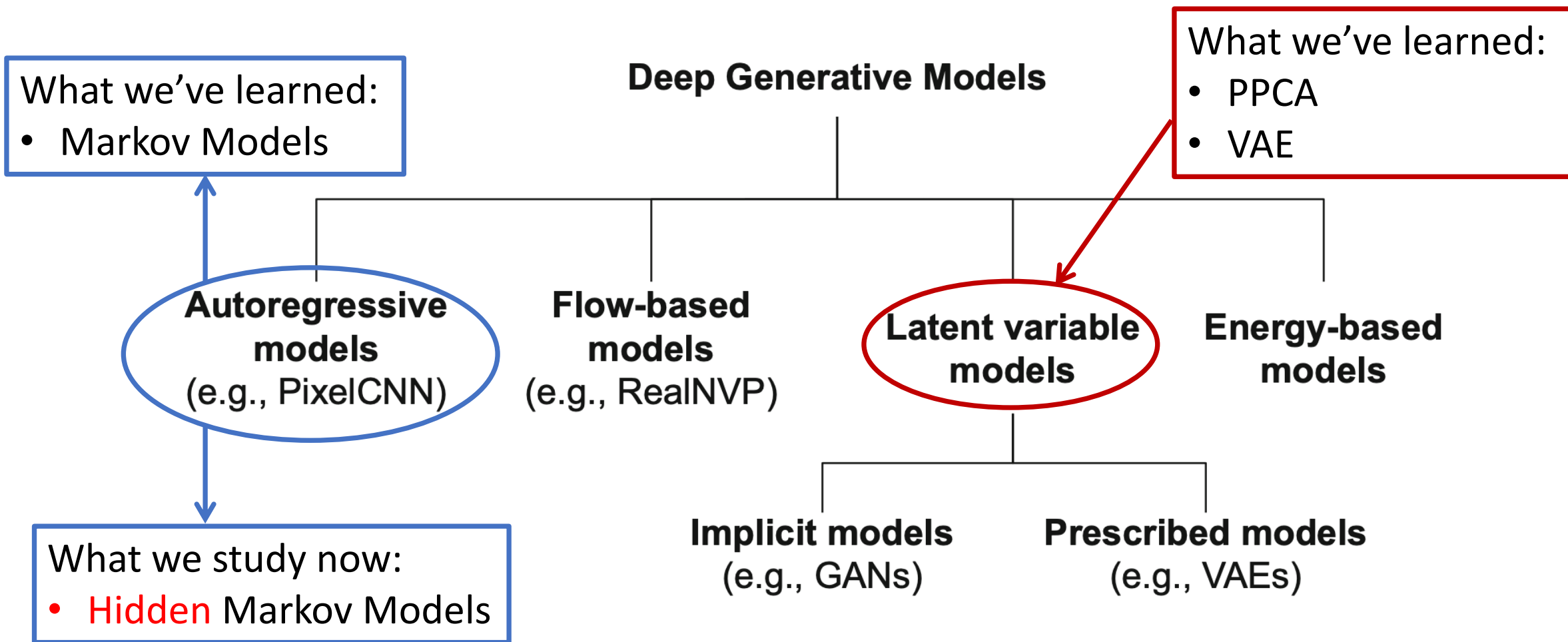
Fall Semester 2025

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS),
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE



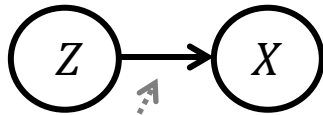
Taxonomy of Generative Models



Hidden Markov Models (Pictorial Definition)

Latent Variable Models

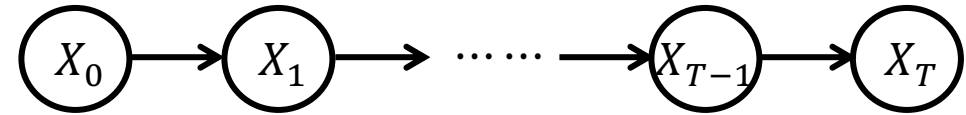
- Z : Latent variable
- X : Observation



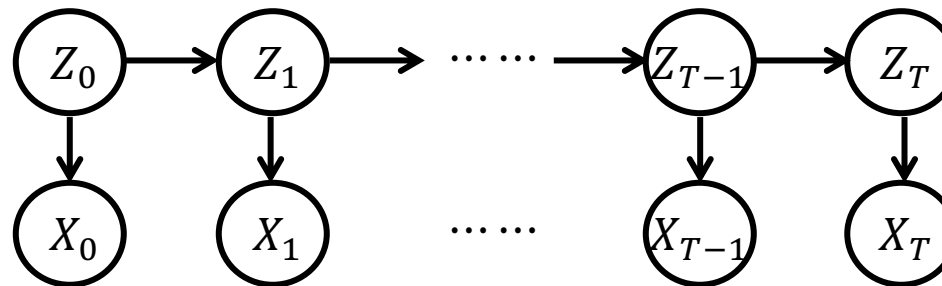
Read it: X depends on Z

Markov Models

- X_t : Observation (state)

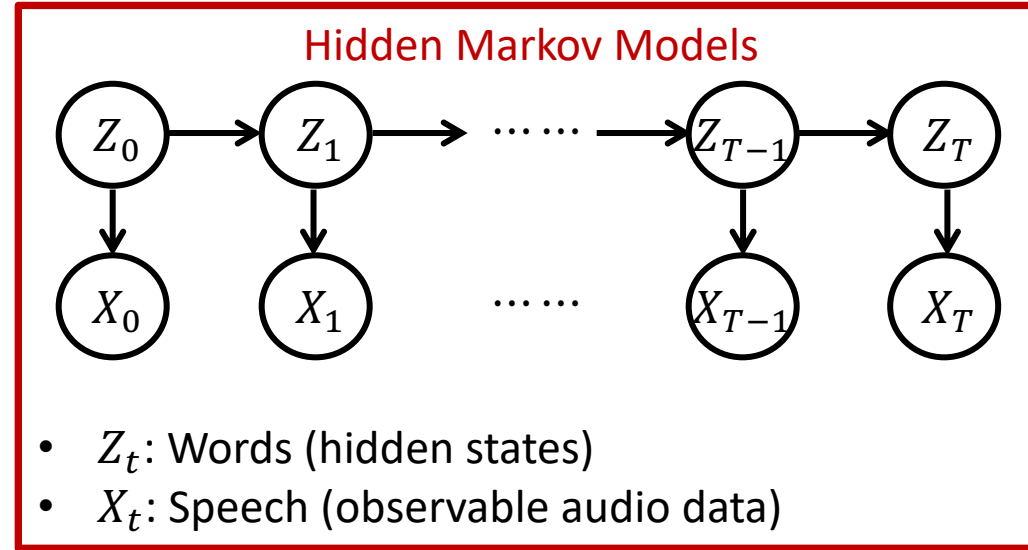


Hidden Markov Models

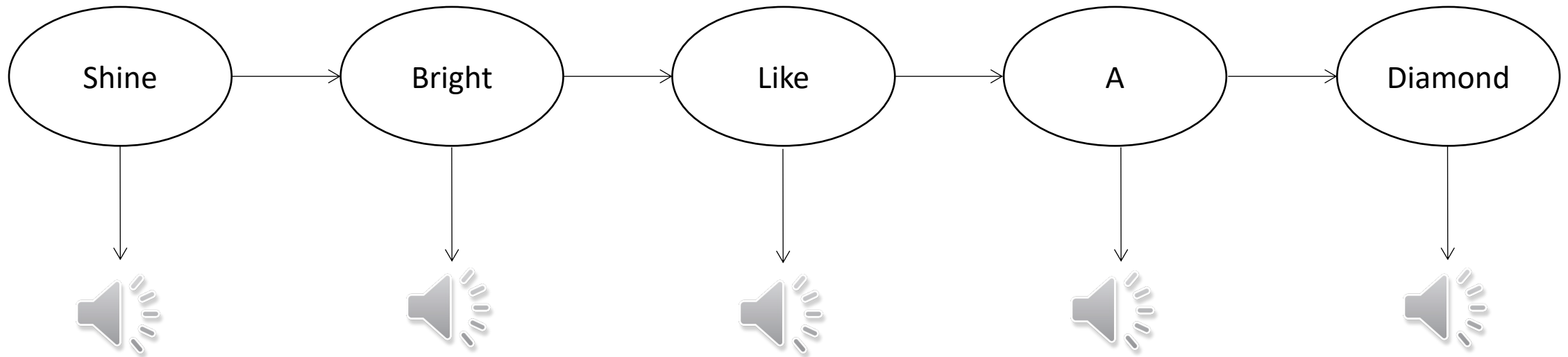


- Z_t : Hidden state (latent variable)
- X_t : Observation

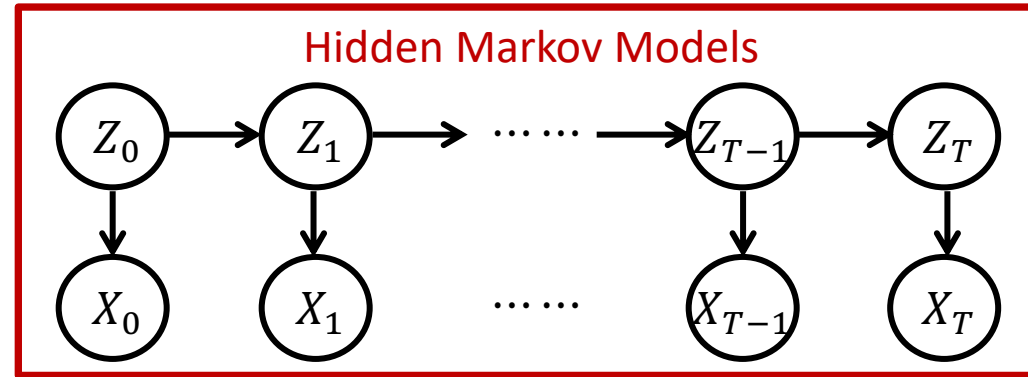
Example: Speech Recognition



Goal: Given observations (audio), discover the hidden states (words)



State Space and Observation Space



- In Markov models, we had state space Ω with K different states
 - So we labeled them as $\Omega = \{1, \dots, K\}$ without loss of generality
- In HMMs, we need to distinguish state space Ω and observation space Σ
 - So we define:

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:

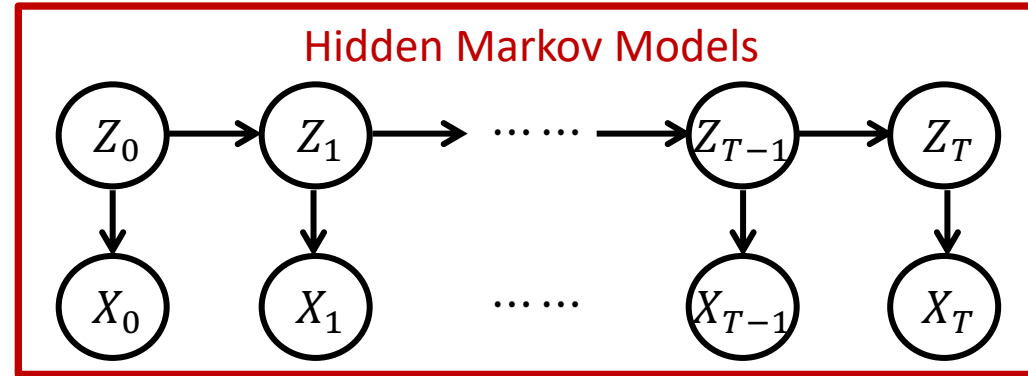
Each Z_t takes values in Ω

Observation Space $\Sigma = \{\sigma_1, \dots, \sigma_V\}$:

Each X_t takes values in Σ
 - But in words we might say:
 - “state i ” for ω_i (simpler than saying “state omega i ”)
 - “observation j ” for σ_j (simpler than saying “observation sigma j ”)

Formal Definition of Hidden Markov Models

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω



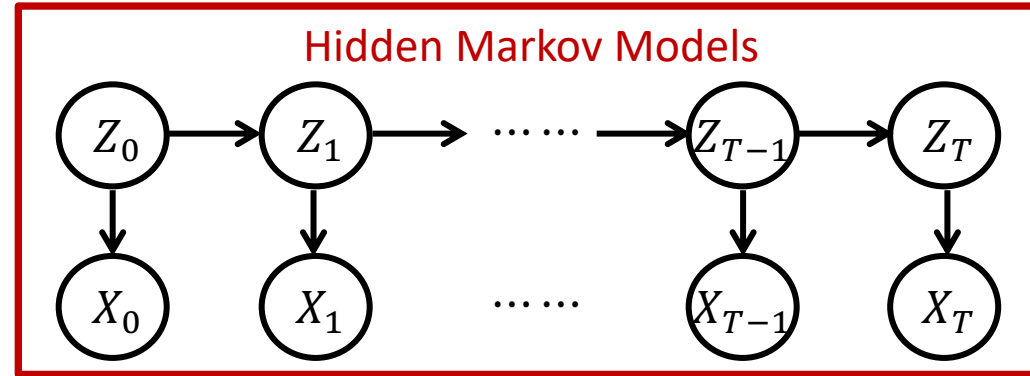
Observation Space $\Sigma = \{\sigma_1, \dots, \sigma_V\}$:
Each X_t takes values in Σ

- Initial Probability: $\pi_i = p(Z_0 = \omega_i), i = 1, \dots, K.$
- Transition Probability: $a_{ij} := p(Z_t = \omega_j \mid Z_{t-1} = \omega_i), i, j = 1, \dots, K.$
- Emission Probability: $c_{jr} := p(X_t = \sigma_r \mid Z_t = \omega_j), j = 1, \dots, K, r = 1, \dots, V.$

Matrix Notation: Transition Matrix $A \in \mathbb{R}^{K \times K}$, Emission Matrix $C \in \mathbb{R}^{K \times V}$, Initial distribution $\pi \in \mathbb{R}^K$

A hidden Markov model is fully specified by its parameters $\theta := (\pi, A, C)$

Assumptions



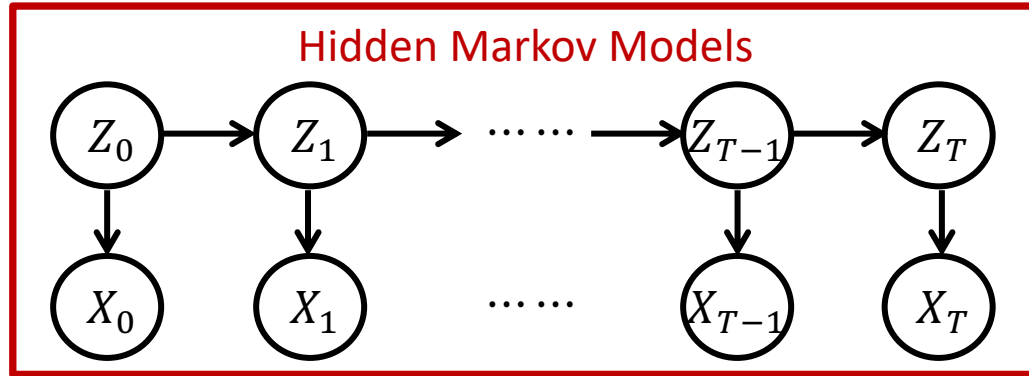
- Markov Property:

$$p_{\theta}(z_t \mid z_0, \dots, z_{t-1}) = p_{\theta}(z_t \mid z_{t-1})$$

- Output Independence Assumption:

$$p_{\theta}(x_t \mid z_0, \dots, z_T, x_0, \dots, x_{t-1}) = p_{\theta}(x_t \mid z_t)$$

Consequences of The Assumptions



Assumptions

A1: $p_{\theta}(z_t \mid z_0, \dots, z_{t-1}) = p_{\theta}(z_t \mid z_{t-1})$

A2: $p_{\theta}(x_t \mid z_0, \dots, z_T, x_0, \dots, x_{t-1}) = p_{\theta}(x_t \mid z_t)$

- **Transition Probability Factorization** (proof: factorize w.r.t. z_0, \dots, z_T , use A1):

- $p_{\theta}(z_0, \dots, z_T) = p_{\theta}(z_0) \prod_{t=1}^T p_{\theta}(z_t \mid z_{t-1})$

- **Emission Probability Factorization** (proof: factorize w.r.t. x_0, \dots, x_T , use A2):

- $p_{\theta}(x_0, \dots, x_T \mid z_0, \dots, z_T) = \prod_{t=0}^T p_{\theta}(x_t \mid z_t)$

- **Joint Probability Factorization** (proof: use the above factorizations):

$$p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{x} \mid \mathbf{z}) \cdot p_{\theta}(\mathbf{z}) = \prod_{t=0}^T p_{\theta}(x_t \mid z_t) \, p_{\theta}(z_0) \prod_{t=1}^T p_{\theta}(z_t \mid z_{t-1})$$

Inference, Decoding, and Learning

- **Inference**. Given θ , compute the likelihood $p_{\theta}(\mathbf{x})$ of observing \mathbf{x}
- **Decoding**. Given observation \mathbf{x} and parameters θ , find best states:

$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{z} \mid \mathbf{x}) = \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z}) / p_{\theta}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- **Learning**. Given N observations $\{\mathbf{x}^{(n)}\}_{n=1}^N$, find best θ :

$$\max_{\theta} \prod_{n=1}^N p_{\theta}(\mathbf{x}^{(n)})$$

- **Example** (Speech Recognition):
 - During training, we learn an HMM (i.e., parameter θ) from audio data $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$
 - At test time, we get an audio \mathbf{x} . We need to use θ and decode \mathbf{x} into words \mathbf{z}^* (states)

Decoding and Inference via “Brute-Force”

Exponential Time Complexity: $O(K^{T+1})$

- **Inference.** Given θ, \mathbf{x} , compute $p_{\theta}(\mathbf{x})$:

$$p_{\theta}(\mathbf{x}) = \sum_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} \prod_{t=0}^T p_{\theta}(x_t \mid z_t) p_{\theta}(z_0) \prod_{t=1}^T p_{\theta}(z_t \mid z_{t-1})$$

- **Decoding.** Given θ, \mathbf{x} , compute:

$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

1. For all possible states \mathbf{z} , compute the likelihood $p_{\theta}(\mathbf{x}, \mathbf{z})$
2. Output the state that gives the maximum likelihood

Faster Algorithms for Inference?

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

- **Inference.** Given θ, \mathbf{x} , compute $p_\theta(\mathbf{x})$
- If $T = 0$, then $p_\theta(\mathbf{x}) = \sum_{z_0} p_\theta(x_0, z_0) = \sum_{z_0} p_\theta(x_0 | z_0) \cdot p_\theta(z_0)$
 - We can compute $p_\theta(\mathbf{x})$ in $O(K)$ time (we need to sum over K possible values of z_0)
- If $T = 1$, then $p_\theta(\mathbf{x}) = \sum_{z_0} \sum_{z_1} p_\theta(x_0, x_1, z_0, z_1)$
 - Therefore, $p_\theta(\mathbf{x}) = \sum_{z_0} \sum_{z_1} p_\theta(x_1 | z_1) \cdot p_\theta(x_0 | z_0) \cdot p_\theta(z_1 | z_0) \cdot p_\theta(z_0)$
 - We can compute $p_\theta(\mathbf{x})$ in $O(K^2)$ time (need to sum over all possible z_1 and z_0)

Intuition. More generally, can we update $p_\theta(x_0, \dots, x_t, z_t)$ from $p_\theta(x_0, \dots, x_{t-1}, z_{t-1})$?
If so, then we might be able to derive a faster algorithm for inference.

Formalize the Intuition

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

- **Inference.** Given θ, \mathbf{x} , compute $p_\theta(\mathbf{x})$

Intuition. More generally, can we update $p_\theta(x_0, \dots, x_t, z_t)$ from $p_\theta(x_0, \dots, x_{t-1}, z_{t-1})$?
If so, then we might be able to derive a faster algorithm for inference.

- **Forward Probability:**

$$\alpha_j(t) := p_\theta(x_0, \dots, x_t, Z_t = \omega_j)$$

- By definition, we have

$$p_\theta(x_0, \dots, x_T) = \sum_{j=1}^K p_\theta(x_0, \dots, x_T, Z_T = \omega_j) = \sum_{j=1}^K \alpha_j(T)$$

- It remains to derive an update formula for $\alpha_j(t)$ for $t = 0, \dots, T$

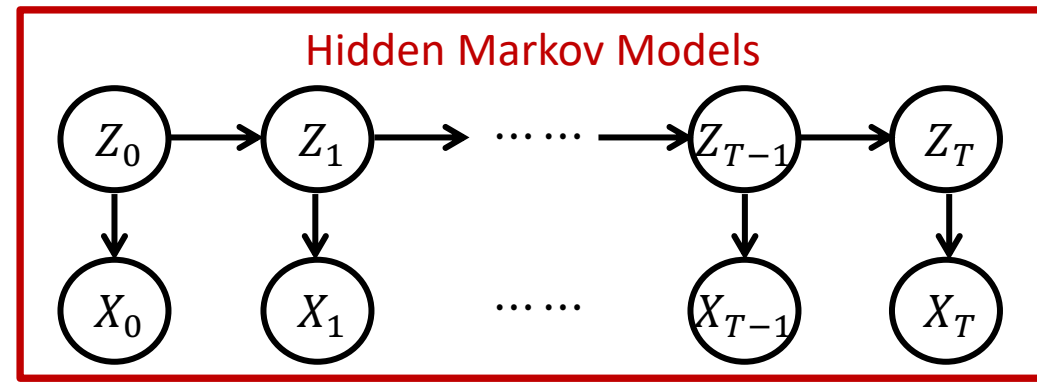
Formalize the Intuition

- **Inference.** Given θ, \mathbf{x} , compute $p_{\theta}(\mathbf{x})$
- **Forward Probability:**

$$\alpha_j(t) := p_{\theta}(x_0, \dots, x_t, Z_t = \omega_j)$$

- **Recurrence Relation:**

$$\begin{aligned}\alpha_j(t) &= p_{\theta}(x_t \mid x_0, \dots, x_{t-1}, Z_t = \omega_j) \cdot p_{\theta}(x_0, \dots, x_{t-1}, Z_t = \omega_j) \\ &= p_{\theta}(x_t \mid Z_t = \omega_j) \sum_{i=1, \dots, K} p_{\theta}(x_0, \dots, x_{t-1}, Z_t = \omega_j, Z_{t-1} = \omega_i) \\ &= c_j(x_t) \sum_{i=1}^K p_{\theta}(x_0, \dots, x_{t-1}, Z_{t-1} = \omega_i) \cdot p_{\theta}(Z_t = \omega_j \mid x_0, \dots, x_{t-1}, Z_{t-1} = \omega_i) \\ &= c_j(x_t) \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}\end{aligned}$$

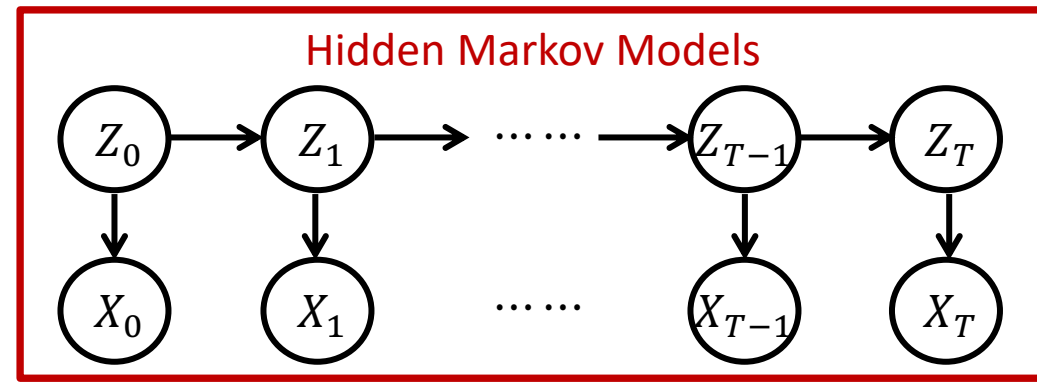


State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

$$c_j(x_t) := p_{\theta}(x_t \mid Z_t = \omega_j)$$

Faster Algorithm for Inference

Time Complexity: $O(TK^2)$



State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

$c_j(x_t) := p_\theta(x_t | Z_t = \omega_j)$

- Inference. Given θ, \mathbf{x} , compute $p_\theta(\mathbf{x})$
- Forward Probability: $\alpha_j(t) := p_\theta(x_0, \dots, x_t, Z_t = \omega_j)$
- Recurrence Relation: $\alpha_j(t) = c_j(x_t) \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}$

Algorithm:

- Initialization: $\alpha_j(0) = \pi_j \cdot c_j(x_0)$
- Recursion: $\alpha_j(t) = c_j(x_t) \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}$
- Termination: Output $\sum_{j=1}^K \alpha_j(T)$

$$\forall j = 1, \dots, K$$

$$\forall j = 1, \dots, K, \forall t = 1, \dots, T$$

What About Decoding?

- **Decoding**. Given θ, \mathbf{x} , compute:

$$\mathbf{z}^* \in \underset{\mathbf{z}}{\operatorname{argmax}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- We know how to decode in $O(K^{T+1})$ time
- Can we do it in $O(TK^2)$ time (similarly to inference)?
 - During **inference**, we need to calculate $\sum_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$
- Two differences from **inference**:
 - Need to calculate the **maximum** of $p_{\theta}(\mathbf{x}, \mathbf{z})$ over \mathbf{z} rather than **sum**
 - Need to find states \mathbf{z}^* attaining the maximum

Inference Versus Decoding

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

- **Inference**. Given θ, \mathbf{x} , compute:

$$p_{\theta}(\mathbf{x}) = \sum_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- **Recurrence Term**:

$$\alpha_j(t) := p_{\theta}(x_0, \dots, x_t, Z_t = \omega_j)$$

- **Marginalization**:

$$p_{\theta}(x_0, \dots, x_T) = \sum_{j=1}^K \alpha_j(T)$$

- **Recurrence Relation**:

$$\alpha_j(t) = c_j(x_t) \cdot \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}$$

$$c_j(x_t) := \mathbb{P}_{\theta}(x_t \mid Z_t = \omega_j)$$

- **Decoding**. Given θ, \mathbf{x} , compute:

$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- **Recurrence Term**:

$$v_j(t) := \max_{z_0, \dots, z_{t-1} \in \Omega} p_{\theta}(x_0, \dots, x_t, z_0, \dots, z_{t-1}, Z_t = \omega_j)$$

- **Maximization**:

$$\max_{z_0, \dots, z_T \in \Omega} p_{\theta}(x_0, \dots, x_T, z_0, \dots, z_T) = \max_{j=1, \dots, K} v_j(T)$$

- **Recurrence Relation**: (proof omitted)

$$v_j(t) = c_j(x_t) \cdot \max_{i=1, \dots, K} v_i(t-1) \cdot a_{ij}$$

Intuition. For decoding, just replace **sum** of inference with **max**

- Summation over z_0, \dots, z_{t-1} is implicit in the definition of $\alpha_j(t)$, and $v_j(t)$ is indeed obtained by replacing **sum** with **max**

Inference Versus Decoding

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

- **Inference**. Given θ, \mathbf{x} , compute:

$$p_{\theta}(\mathbf{x}) = \sum_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- **Initialization**: ($\forall j = 1, \dots, K$)

$$\alpha_j(0) = \pi_j \cdot c_j(x_0)$$

- **Recursion**: ($\forall j = 1, \dots, K, \forall t = 1, \dots, T$)

$$\alpha_j(t) = c_j(x_t) \cdot \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}$$

- **Output**:

$$\text{optimal value} = \sum_{j=1}^K \alpha_j(T)$$

$$c_j(x_t) := \mathbb{P}_{\theta}(x_t \mid Z_t = \omega_j)$$

- **Decoding**. Given θ, \mathbf{x} , compute:

$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$

- **Initialization**: ($\forall j = 1, \dots, K$)

$$v_j(0) = \pi_j \cdot c_j(x_0)$$

- **Recursion**: ($\forall j = 1, \dots, K, \forall t = 1, \dots, T$)

$$v_j(t) = c_j(x_t) \cdot \max_{i=1, \dots, K} v_i(t-1) \cdot a_{ij}$$

- **Output**:

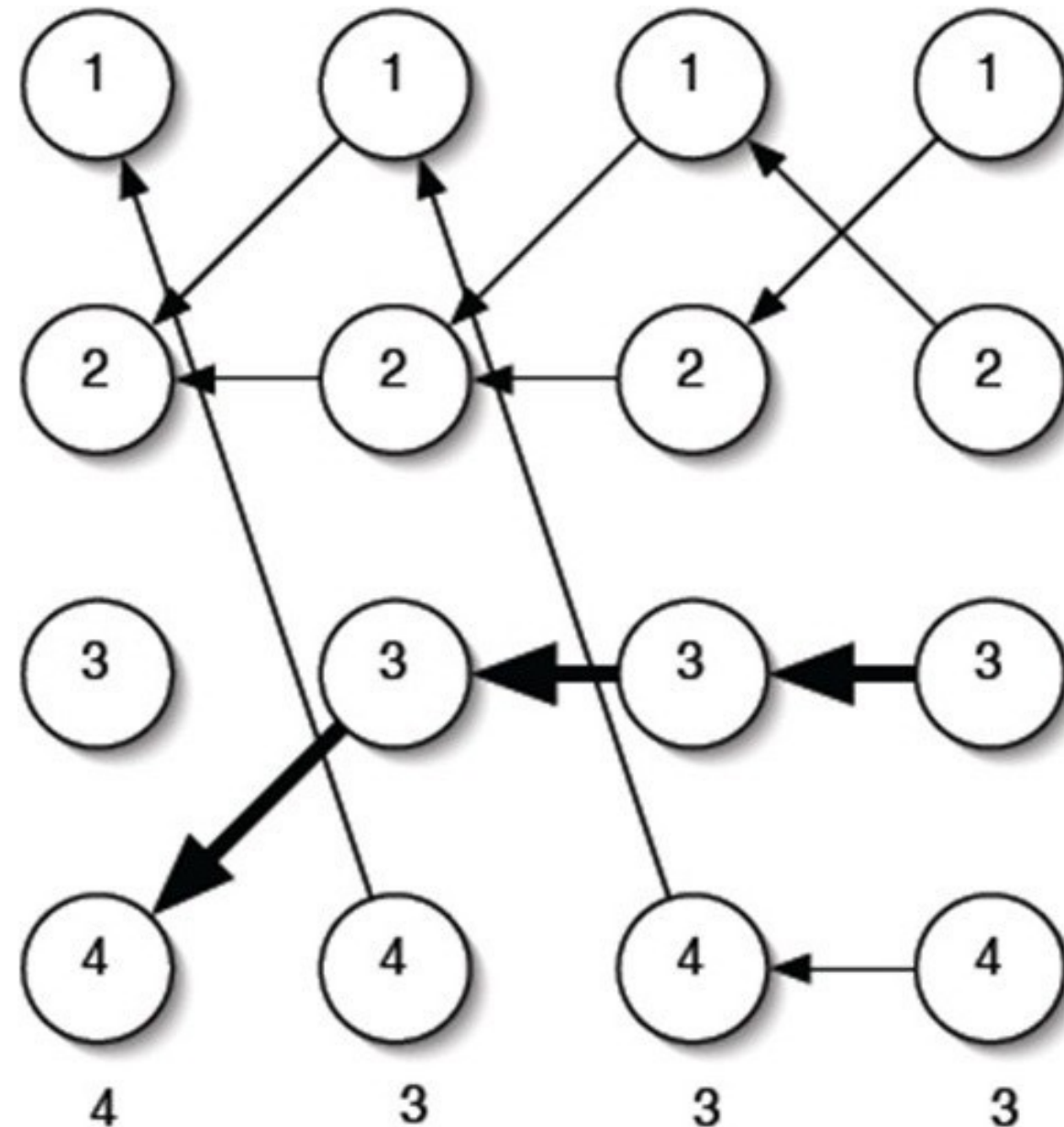
$$\text{optimal value} = \max_{j=1, \dots, K} v_j(T)$$

One more step needed: find \mathbf{z}^* that attains the optimum
Solution: “backtracking” (next page)

Backtracking for Decoding

$$c_j(x_t) := \mathbb{P}_\theta(x_t \mid Z_t = \omega_j)$$

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω



Algorithm (Viterbi)

- **Initialization:** ($\forall j = 1, \dots, K$)

$$v_j(0) = \pi_j \cdot c_j(x_0)$$

- **Recursion:** ($\forall j = 1, \dots, K, \forall t = 1, \dots, T$)

$$v_j(t) = c_j(x_t) \cdot \max_{i=1, \dots, K} \{v_i(t-1) \cdot a_{ij}\}$$

$$prev_j(t) = \operatorname{argmax}_{i=1, \dots, K} \{v_i(t-1) \cdot a_{ij}\}$$

- **Output:**

$$\text{optimal value} = \max_{j=1, \dots, K} v_j(T)$$

$$i^*(T) = \operatorname{argmax}_{j=1, \dots, K} v_j(T)$$

- **Backtracking:** ($\forall t = T, \dots, 1$)

$$i^*(t-1) = prev_{i^*(t)}(t)$$

REMARK. THE FINAL ALGORITHM IS KNOWN AS THE *VITERBI ALGORITHM*. IT IS AN INSTANCE OF *DYNAMIC PROGRAMMING*.

Inference, Decoding, and Learning

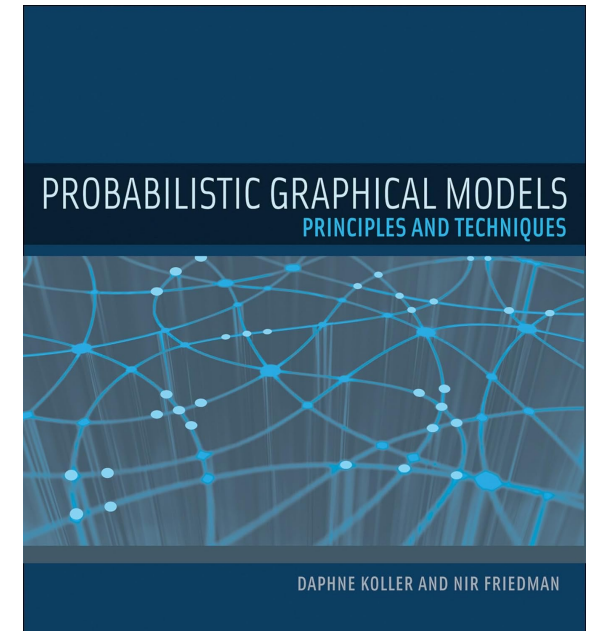
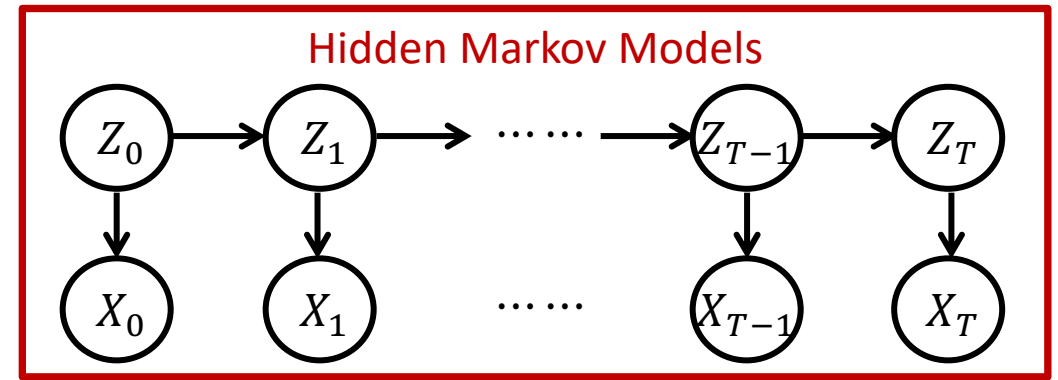
- **Inference.** Given θ, \mathbf{x} , compute $p_{\theta}(\mathbf{x})$
- **Decoding.** Given θ, \mathbf{x} , compute:
$$\mathbf{z}^* \in \operatorname{argmax}_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z})$$
- **Learning.** Given N observations $\{\mathbf{x}^{(n)}\}_{n=1}^N$, find best θ :

$$\max_{\theta} \prod_{n=1}^N p_{\theta}(\mathbf{x}^{(n)})$$

- **We've seen how to perform inference and decoding in $O(TK^2)$ time**
 - sum vs. max, dynamic programming, backtracking
- **We can perform learning via the EM algorithm**
 - derivations are algebraically involved and are given at the end of the slides.

Conclusion

- We have studied HMMs
 - Inference, Decoding, and Learning
 - Viterbi algorithm for decoding
- Further Study: Probabilistic Graphical Models (PGMs)
 - Graph can model more complicated dependency:
 - nodes denote random variables
 - edges denote the dependency between random variables
 - HMMs are examples of PGMs
 - We can extend the notions of inference and learning for PGMs



Appendix (Optional)

- In the subsequent slides, we derive EM for learning the parameters of HMMs.

Learning Hidden Markov Models

$$\mathbf{x}^{(n)} := (x_0^{(n)}, \dots, x_T^{(n)})$$
$$\mathbf{z} := (z_0, \dots, z_T)$$

- **Learning.** Given N observations $\{\mathbf{x}^{(n)}\}_{n=1}^N$, find best θ :

$$\max_{\theta} \prod_{n=1}^N p_{\theta}(\mathbf{x}^{(n)})$$

- **EM Algorithm.** Initialize θ^0 and alternate: (k = iteration counter)

E-step:

$$q^k(\mathbf{z} \mid \mathbf{x}^{(n)}) = p_{\theta^k}(\mathbf{z} \mid \mathbf{x}^{(n)})$$

M-step:

$$\theta^{k+1} = \operatorname{argmax}_{\theta} \sum_{n=1}^N \sum_{\mathbf{z}} q^k(\mathbf{z} \mid \mathbf{x}^{(n)}) \log p_{\theta}(\mathbf{x}^{(n)}, \mathbf{z})$$

Substitute E-step into M-step

$$\theta^{k+1} = \operatorname{argmax}_{\theta} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} \mid \mathbf{x}^{(n)}) \cdot \log p_{\theta}(\mathbf{x}^{(n)}, \mathbf{z})$$

- **We next instantiate the EM algorithm based on the hidden Markov model**
 - Caution: This involves multiple pages of derivations

EM for Mixture Models: A Basic Fact

- To proceed, we recall a basic fact we've seen multiple times

Fact. Consider the following optimization problem:

$$p^* = \operatorname{argmax}_p \sum_{i=1}^K s_i \log(p_i) \quad \text{subject to} \quad p_1 + \cdots + p_K = 1$$

This problem admits a closed-form solution for $p^* = [p_1^*, \dots, p_K^*]$:

$$p_i^* = \frac{s_i}{\sum_{i=1}^K s_i}$$

- In the sequel, we will use this fact often. When we use it, we write (concisely):

Fact: $\sum_{i=1}^K s_i \log(p_i)$ is maximized at $p_i = s_i / (\sum_{i=1}^K s_i)$

Instantiate EM for HMMs

$$\begin{aligned}\mathbf{x}^{(n)} &:= (x_0^{(n)}, \dots, x_T^{(n)}) \\ \mathbf{z} &:= (z_0, \dots, z_T)\end{aligned}$$

$$\theta^{k+1} = \operatorname{argmax}_{\theta} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\theta}(\mathbf{x}^{(n)}, \mathbf{z})$$

$$p_{\theta}(\mathbf{x}^{(n)}, \mathbf{z}) := \prod_{t=0}^T p_C(x_t^{(n)} | z_t) p_{\pi}(z_0) \prod_{t=1}^T p_A(z_t | z_{t-1})$$

$$(\pi^{k+1}, A^{k+1}, C^{k+1}) = \operatorname{argmax}_{\pi, A, C} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \left(\log p_{\pi}(z_0) + \sum_{t=0}^T \log p_C(x_t^{(n)} | z_t) + \sum_{t=1}^T \log p_A(z_t | z_{t-1}) \right)$$

The objective function is separable

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

$$C^{k+1} = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | z_t)$$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_A(z_t | z_{t-1})$$

Instantiate EM for HMMs

$$\begin{aligned}\mathbf{x}^{(n)} &:= (x_0^{(n)}, \dots, x_T^{(n)}) \\ \mathbf{z} &:= (z_0, \dots, z_T)\end{aligned}$$

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

$$C^{k+1} = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | z_t)$$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_A(z_t | z_{t-1})$$

Observation. The summation over \mathbf{z} in the above can be simplified, e.g.,

- In the update of π^{k+1} ,
 - $\log p_{\pi}(z_0)$ depends only on z_0
 - $p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)})$ depends on $\mathbf{z} := (z_0, \dots, z_T)$
- So we have

$$\sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0) = \sum_{z_0, \dots, z_T} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0) = \sum_{z_0} p_{\theta^k}(z_0 | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

- Similarly we can simplify the updates of A^{k+1} and C^{k+1}

Instantiate EM for HMMs

$$\begin{aligned}\mathbf{x}^{(n)} &:= (x_0^{(n)}, \dots, x_T^{(n)}) \\ \mathbf{z} &:= (z_0, \dots, z_T)\end{aligned}$$

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

$$C^{k+1} = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | z_t)$$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{\mathbf{z}} p_{\theta^k}(\mathbf{z} | \mathbf{x}^{(n)}) \log p_A(z_t | z_{t-1})$$

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{z_0} p_{\theta^k}(z_0 | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

$$C^{k+1} = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{z_t} p_{\theta^k}(z_t | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | z_t)$$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{z_t, z_{t-1}} p_{\theta^k}(z_t, z_{t-1} | \mathbf{x}^{(n)}) \log p_A(z_t | z_{t-1})$$

We will next handle the update of π^{k+1} , A^{k+1} , and C^{k+1} in succession

Updating π^{k+1}

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

$$\mathbf{x}^{(n)} := (x_0^{(n)}, \dots, x_T^{(n)})$$

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{z_0} p_{\theta^k}(z_0 | \mathbf{x}^{(n)}) \cdot \log p_{\pi}(z_0)$$

Rewrite the objective using the definition of π_i and add the constraints $\pi_1 + \dots + \pi_K = 1$

$$\pi^{k+1} = \operatorname{argmax}_{\pi} \sum_{n=1}^N \sum_{i=1}^K p_{\theta^k}(Z_0 = \omega_i | \mathbf{x}^{(n)}) \log(\pi_i) \quad \text{subject to} \quad \pi_1 + \dots + \pi_K = 1$$

Fact: $\sum_{i=1}^K s_i \log(p_i)$ is maximized at $p_i = s_i / (\sum_{i=1}^K s_i)$. But what is s_i ?

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N p_{\theta^k}(Z_0 = \omega_i | \mathbf{x}^{(n)})}{\sum_{n=1}^N \sum_{i=1}^K p_{\theta^k}(Z_0 = \omega_i | \mathbf{x}^{(n)})} = \frac{\sum_{n=1}^N p_{\theta^k}(Z_0 = \omega_i | \mathbf{x}^{(n)})}{N}, \quad \forall i = 1, \dots, K$$

Remark. We will calculate $p_{\theta^k}(Z_0 = \omega_i | \mathbf{x}^{(n)})$ later

Updating A^{k+1}

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

$$\mathbf{x}^{(n)} := (x_0^{(n)}, \dots, x_T^{(n)})$$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{z_t, z_{t-1}} p_{\theta^k}(z_t, z_{t-1} | \mathbf{x}^{(n)}) \log p_A(z_t | z_{t-1})$$

↓ Rewrite the objective using the definition of a_{ij} and add the constraints $\sum_{j=1}^K a_{ij} = 1 \ (\forall i)$

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{j=1}^K \sum_{i=1}^K p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)}) \log a_{ij} \quad \text{subject to } \sum_{j=1}^K a_{ij} = 1 \ (\forall i)$$

↓ The objective function and constraints are separable

$$(a_{i1}^{k+1}, \dots, a_{iK}^{k+1}) = \operatorname{argmax}_A \sum_{n=1}^N \sum_{t=1}^T \sum_{j=1}^K p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)}) \log a_{ij} \quad \text{subject to } \sum_{j=1}^K a_{ij} = 1$$

↓ Fact: $\sum_{j=1}^K s_j \log(p_j)$ is maximized at $p_j = s_j / (\sum_{j=1}^K s_j)$. But what is s_j ?

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)})}{\sum_{n=1}^N \sum_{t=1}^T \sum_{j=1}^K p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)})} = \frac{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)})}{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_{t-1} = \omega_i | \mathbf{x}^{(n)})} \quad (\forall i, j)$$

Remark. We will calculate $p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i | \mathbf{x}^{(n)})$ and $p_{\theta^k}(Z_{t-1} = \omega_i | \mathbf{x}^{(n)})$ later

Updating C^{k+1}

State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
Each Z_t takes values in Ω

Observation Space $\Sigma = \{\sigma_1, \dots, \sigma_V\}$:
Each X_t takes values in Σ

$$C^{k+1} = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{z_t} p_{\theta^k}(z_t | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | z_t)$$

$$\mathbf{x}^{(n)} := (x_0^{(n)}, \dots, x_T^{(n)})$$

Rewrite the objective using the definition of b_{jr} , indicator function $\mathbb{I}(\cdot)$, and add the constraints $\sum_{r=1}^V c_{jr} = 1 \ (\forall j)$

$$\begin{aligned} C^{k+1} &= \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{j=1}^K p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \log p_C(x_t^{(n)} | Z_t = \omega_j) \\ &= \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{j=1}^K p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \sum_{r=1}^V \mathbb{I}(x_t^{(n)} = \sigma_r) \log c_{jr} \end{aligned} \quad \text{s.t. } \sum_{r=1}^V c_{jr} = 1 \ (\forall j)$$

The objective function and constraints are separable

$$(c_{j1}^{k+1}, \dots, c_{jV}^{k+1}) = \operatorname{argmax}_C \sum_{n=1}^N \sum_{t=0}^T \sum_{r=1}^V p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r) \log c_{jr} \quad \text{s.t. } \sum_{r=1}^V c_{jr} = 1$$

Fact: $\sum_{r=1}^V s_r \log(p_r)$ is maximized at $p_r = s_r / (\sum_{r=1}^V s_r)$. But what is s_r ?

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T \sum_{r=1}^V p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r)} = \frac{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)})} \quad (\forall j, r)$$

Remark. We will calculate $p_{\theta^k}(Z_t = \omega_j | \mathbf{x}^{(n)})$ later

Updating $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N p_{\theta^k}(Z_0 = \omega_i \mid \mathbf{x}^{(n)})}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)})}{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)})} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)})} \quad (\forall j, r)$$

- **Question:** How to understand and interpret the above update formulas?
- **Hint:** Probability is expectation of indicator function, that is

$$\sum_{n=1}^N p_{\theta^k}(Z_0 = \omega_i \mid \mathbf{x}^{(n)}) = \sum_{n=1}^N \mathbb{E}_{\theta^k}[\mathbb{I}(Z_0 = \omega_i) \mid \mathbf{x}^{(n)}]$$

- **Answer:**

$$\pi_i^{k+1} = \frac{\text{Expected number of times that } Z_0 = \omega_i \text{ takes place, given } \{\mathbf{x}^{(n)}\}_{n=1}^N \text{ and } \theta^k}{N}$$

We can interpret a_{ij}^{k+1} and c_{jr}^{k+1} similarly (how?)

Updating $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N p_{\theta^k}(Z_0 = \omega_i \mid \mathbf{x}^{(n)})}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)})}{\sum_{n=1}^N \sum_{t=1}^T p_{\theta^k}(Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)})} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)}) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)})} \quad (\forall j, r)$$

- To proceed, it suffices to calculate the following “posteriors” $(\forall i, j)$:

- $\xi_{ij}^k(t, n) := p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)}) \quad (\forall t > 0)$
- $\gamma_j^k(t, n) := p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)}) \quad (\forall t \geq 0)$

- Indeed, we have:

Remark. Computing $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$ is the **E-step**!

- In general, the E-step requires computing the posterior $p_{\theta^k}(\mathbf{z} \mid \mathbf{x}^{(n)})$ for all possible \mathbf{z} .
- For HMMs, computing $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$ is enough

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N \gamma_i^k(0, n)}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T \xi_{ij}^k(t, n)}{\sum_{n=1}^N \sum_{t=1}^T \gamma_i^k(t-1, n)} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n)} \quad (\forall j, r)$$

Updating $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N \gamma_i^k(0, n)}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T \xi_{ij}^k(t, n)}{\sum_{n=1}^N \sum_{t=1}^T \gamma_i^k(t-1, n)} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n)} \quad (\forall j, r)$$

- $\xi_{ij}^k(t, n) := p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)}) \quad (\forall t > 0)$
- $\gamma_j^k(t, n) := p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)}) \quad (\forall t \geq 0)$

- **Question.** Wouldn't it be enough to compute $\xi_{ij}^k(t, n)$ only, as we obviously have

$$\gamma_j^k(t-1, n) = \sum_{i=1}^K \xi_{ij}^k(t, n)?$$

- **Answer.**

- $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$ are widely used notations elsewhere: Not possible to uproot them

Updating $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N \gamma_i^k(0, n)}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T \xi_{ij}^k(t, n)}{\sum_{n=1}^N \sum_{t=1}^T \gamma_i^k(t-1, n)} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n)} \quad (\forall j, r)$$

- $\xi_{ij}^k(t, n) := p_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \mathbf{x}^{(n)}) \quad (\forall t > 0)$
- $\gamma_j^k(t, n) := p_{\theta^k}(Z_t = \omega_j \mid \mathbf{x}^{(n)}) \quad (\forall t \geq 0)$

- Dropping indices n, k for clarity, we can reformulate our goal (**E-step**) as follows:

E-step. Given $\mathbf{y} = (x_1, \dots, x_T)$ and θ , compute $(\forall i, j)$

- $\xi_{ij}(t) = p_{\theta}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid x_0, \dots, x_T) \quad (t > 0)$
- $\gamma_j(t) = p_{\theta}(Z_t = \omega_j \mid x_0, \dots, x_T) \quad (t \geq 0)$

Computing $\xi_{ij}(t)$ and $\gamma_j(t)$

$$\alpha_j(t) := p_\theta(x_0, \dots, x_t, Z_t = \omega_j)$$
$$\beta_j(t) := p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_j)$$

E-step. Given $\mathbf{x} = (x_1, \dots, x_T)$ and θ , compute $(\forall i, j)$

- $\xi_{ij}(t) = p_\theta(Z_t = \omega_j, Z_{t-1} = \omega_i \mid x_0, \dots, x_T) \quad (t > 0)$
- $\gamma_j(t) = p_\theta(Z_t = \omega_j \mid x_0, \dots, x_T) \quad (t \geq 0)$

- **Intuition:** $\xi_{ij}(t)$ and $\gamma_j(t)$ are “posteriors”
 - To compute them, we need to convert them into likelihood via Bayes rule
- Applying the Bayes rule to $\gamma_j(t)$ gives

$$\begin{aligned} \gamma_j(t) &= \frac{p_\theta(x_0, \dots, x_T, Z_t = \omega_j)}{p_\theta(x_0, \dots, x_T)} = \frac{p_\theta(x_0, \dots, x_t, Z_t = \omega_j) \cdot p_\theta(x_{t+1}, \dots, x_T \mid x_0, \dots, x_t, Z_t = \omega_j)}{p_\theta(x_0, \dots, x_T)} \\ &= \frac{p_\theta(x_0, \dots, x_t, Z_t = \omega_j) \cdot p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_j)}{p_\theta(x_0, \dots, x_T)} = \frac{\alpha_j(t) \cdot p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_j)}{\sum_{i=1}^K \alpha_i(t) \cdot p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_i)} = \frac{\alpha_j(t) \beta_j(t)}{\sum_{i=1}^K \alpha_i(t) \beta_i(t)} \end{aligned}$$

- $\beta_j(t)$ is called “backward probability”
- To update $\gamma_j(t)$, it suffices to recursively update $\alpha_j(t)$ and $\beta_j(t)$

Computing $\xi_{ij}(t)$ and $\gamma_j(t)$

$$\alpha_j(t) := p_\theta(x_0, \dots, x_t, Z_t = \omega_j)$$

$$\beta_j(t) := p_\theta(x_{t+1}, \dots, x_T | Z_t = \omega_j)$$

E-step. Given $\mathbf{x} = (x_1, \dots, x_T)$ and θ , compute $(\forall i, j)$

- $\xi_{ij}(t) = p_\theta(Z_t = \omega_j, Z_{t-1} = \omega_i | x_0, \dots, x_T) \quad (t > 0)$
- $\gamma_j(t) = p_\theta(Z_t = \omega_j | x_0, \dots, x_T) \quad (t \geq 0)$

$$c_j(x_t) := p(x_t | Z_t = \omega_j)$$

$$\gamma_j(t) = \frac{p_\theta(x_0, \dots, x_T, Z_t = \omega_j)}{p_\theta(x_0, \dots, x_T)} = \frac{\alpha_j(t) \beta_j(t)}{\sum_{i=1}^K \alpha_i(t) \beta_i(t)}$$

Remark. Recall the recursion:

$$\alpha_j(t) = c_j(y_t) \cdot \sum_{i=1}^K a_{ij} \cdot \alpha_i(t-1)$$

- Similarly, applying the Bayes rule to $\xi_{ij}(t)$ gives: (details omitted, please verify)

$$\xi_{ij}(t) = \frac{p_\theta(x_0, \dots, x_T, Z_t = \omega_j, Z_{t-1} = \omega_i)}{p_\theta(x_0, \dots, x_T)} = \dots = \frac{c_j(y_t) \cdot a_{ij} \cdot \alpha_i(t-1) \beta_j(t)}{\sum_{i=1}^K \alpha_i(t) \beta_i(t)}$$

Remark. $\gamma_j(t)$ and $\xi_{ij}(t)$ have the same denominator.

Computing $\xi_{ij}(t)$ and $\gamma_j(t)$

$$\begin{aligned}\alpha_j(t) &:= p_\theta(x_0, \dots, x_t, Z_t = \omega_j) \\ \beta_j(t) &:= p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_j)\end{aligned}$$

E-step. Given $\mathbf{x} = (x_1, \dots, x_T)$ and θ , compute $(\forall i, j)$

- $\xi_{ij}(t) = p_\theta(Z_t = \omega_j, Z_{t-1} = \omega_i \mid x_0, \dots, x_T) \quad (t > 0)$
- $\gamma_j(t) = p_\theta(Z_t = \omega_j \mid x_0, \dots, x_T) \quad (t \geq 0)$

$$c_j(x_t) := p(x_t \mid Z_t = \omega_j)$$

- $\gamma_j(t) = \frac{p_\theta(x_0, \dots, x_T, Z_t = \omega_j)}{p_\theta(x_0, \dots, x_T)} = \frac{\alpha_j(t)\beta_j(t)}{\sum_{i=1}^K \alpha_i(t)\beta_i(t)}$
- $\xi_{ij}(t) = \frac{p_\theta(x_0, \dots, x_T, Z_t = \omega_j, Z_{t-1} = \omega_i)}{p_\theta(x_0, \dots, x_T)} = \dots = \frac{c_j(y_t) \cdot a_{ij} \cdot \alpha_i(t-1)\beta_j(t)}{\sum_{i=1}^K \alpha_i(t)\beta_i(t)}$

Remark.

- Now we only need to compute $\alpha_j(t)$ and $\beta_j(t)$
 - We've seen how to update $\alpha_j(t)$ via the recursion $\alpha_j(t) = c_j(y_t) \cdot \sum_{i=1}^K a_{ij} \cdot \alpha_i(t-1)$

Updating $\alpha_j(t)$ and $\beta_j(t)$

$$\begin{aligned}\alpha_j(t) &:= p_\theta(x_0, \dots, x_t, Z_t = \omega_j) \\ \beta_j(t) &:= p_\theta(x_{t+1}, \dots, x_T \mid Z_t = \omega_j)\end{aligned}$$

$$c_j(x_t) := p(x_t \mid Z_t = \omega_j)$$

- Compute **forward probability** $\alpha_j(t)$:

- Initialization: $\alpha_j(0) = \pi_j \cdot c_j(y_0)$
- Recursion: $\alpha_j(t) = c_j(x_t) \sum_{i=1}^K \alpha_i(t-1) \cdot a_{ij}$

$$\forall j = 1, \dots, K$$

$$\forall j = 1, \dots, K, \forall t = 1, \dots, T$$

- Compute **backward probability** $\beta_j(t)$: (details omitted, please verify)

- Initialization: $\beta_j(T) = 1$
- Recursion: $\beta_j(t-1) = \sum_{i=1}^K c_i(x_t) \cdot \beta_i(t) \cdot a_{ji}$

$$\forall j = 1, \dots, K$$

$$\forall j = 1, \dots, K, \forall t = T, \dots, 1$$

Summary of the EM Algorithm

1. We derived the formula of the EM algorithm for updating $\pi^{k+1}, A^{k+1}, C^{k+1}$ and showed that their updates depend on posteriors $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$
2. We dropped indices n, k and defined $\xi_{ij}(t)$ and $\gamma_j(t)$
3. We showed that $\xi_{ij}(t)$ and $\gamma_j(t)$ depend on the forward probability and backward probability, $\alpha_j(t)$ and $\beta_j(t)$, which can be updated recursively

$$\begin{aligned}\alpha_j(t) &:= p_\theta(x_0, \dots, x_t, Z_t = \omega_j) \\ \beta_j(t) &:= p_\theta(x_{t+1}, \dots, x_T | Z_t = \omega_j)\end{aligned}$$

Indices n, k added back

Remark. To implement the EM algorithm, we need to **go backwards**:

1. Compute $\alpha_j^k(t, n)$ and $\beta_j^k(t, n)$ (indices n, k added back)
2. **E-Step:** Compute $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$
3. **M-Step:** Compute $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\begin{aligned}\alpha_j^k(t, n) &:= p_{\theta^k}(x_0^{(n)}, \dots, x_t^{(n)}, Z_t = \omega_j) \\ \beta_j^k(t, n) &:= p_{\theta^k}(x_{t+1}^{(n)}, \dots, x_T^{(n)} | Z_t = \omega_j)\end{aligned}$$

Implement EM (Iteration k)

1. Compute $\alpha_j^k(t, n)$ and $\beta_j^k(t, n)$ (indices added back)
2. **E-Step:** Compute $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$
3. **M-Step:** Compute $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\alpha_j^k(t, n) := p_{\theta^k}(x_0^{(n)}, \dots, x_t^{(n)}, Z_t = \omega_j)$$

$$\beta_j^k(t, n) := p_{\theta^k}(x_{t+1}^{(n)}, \dots, x_T^{(n)} \mid Z_t = \omega_j)$$

$$c_j^k(x_t^{(n)}) := p_{\theta^k}(x_t^{(n)} \mid Z_t = \omega_j)$$

1. Compute forward probability $\alpha_j^k(t, n)$ and backward probability $\beta_j^k(t, n)$ ($\forall n$):

1.1. Initialization ($\forall j$)

$$\alpha_j^k(0, n) = \pi_j^k \cdot c_j^k(x_0^{(n)}), \quad \beta_j^k(T, n) = 1$$

1.2. Recursion ($\forall j, t$)

$$\alpha_j^k(t, n) = c_j^k(x_t^{(n)}) \sum_{i=1}^K \alpha_i^k(t-1, n) \cdot a_{ij}^k$$

$$\beta_j^k(t-1, n) = \sum_{i=1}^K c_i^k(x_t^{(n)}) \cdot \beta_i^k(t, n) \cdot a_{ji}^k$$

2. **E-Step:** Compute posteriors $\xi_{ij}^k(t, n)$ and $\gamma_j^k(t, n)$

$$\gamma_j^k(t, n) = \frac{\alpha_j^k(t, n) \beta_j^k(t, n)}{\sum_{i=1}^K \alpha_i^k(t, n) \beta_i^k(t, n)}$$

$$\xi_{ij}^k(t, n) = \frac{c_j^k(x_t^{(n)}) \cdot a_{ij}^k \cdot \alpha_i^k(t-1, n) \beta_j^k(t, n)}{\sum_{i=1}^K \alpha_i^k(t, n) \beta_i^k(t, n)}$$

Congrats! You have derived the “Baum-Welch algorithm” (Google search it)

3. **M-Step:** Compute parameters $\pi^{k+1}, A^{k+1}, C^{k+1}$

$$\pi_i^{k+1} = \frac{\sum_{n=1}^N \gamma_i^k(0, n)}{N} \quad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=1}^T \xi_{ij}^k(t, n)}{\sum_{n=1}^N \sum_{t=1}^T \gamma_i^k(t-1, n)} \quad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n) \mathbb{I}(x_t^{(n)} = \sigma_r)}{\sum_{n=1}^N \sum_{t=0}^T \gamma_j^k(t, n)} \quad (\forall j, r)$$