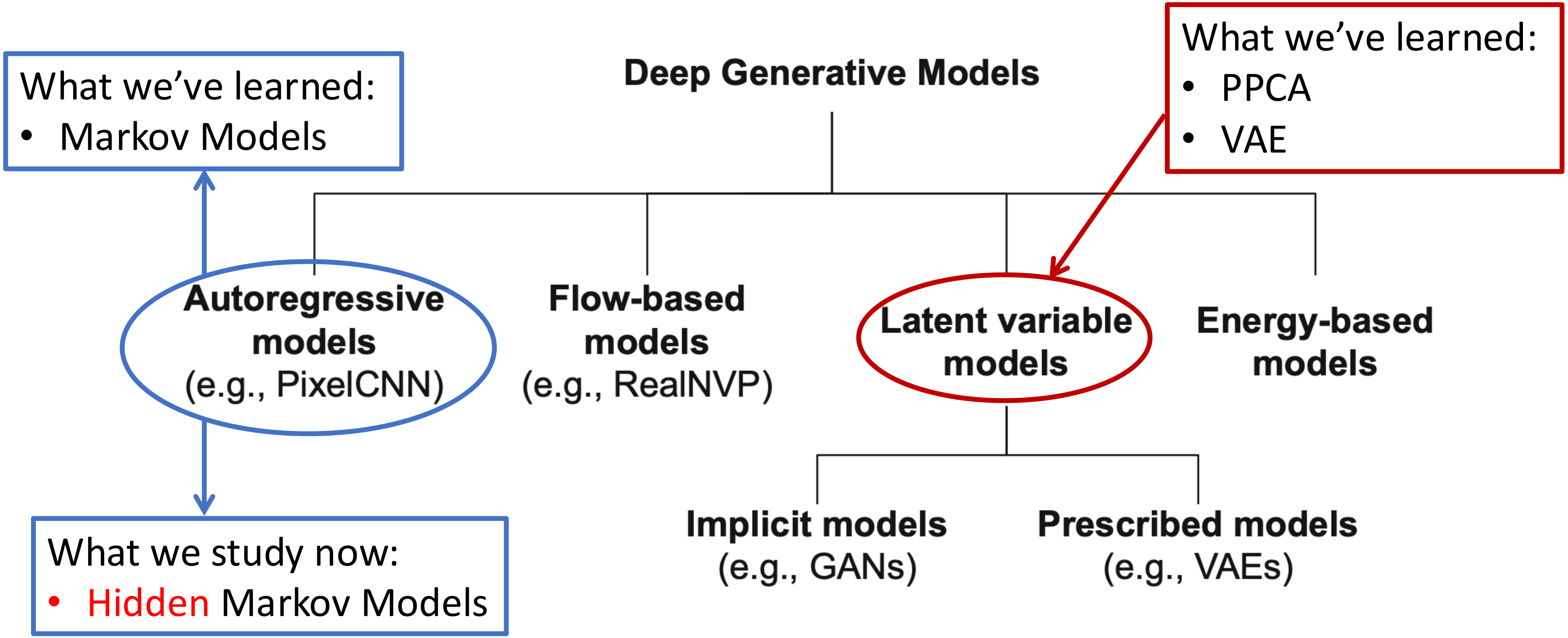
Deep Generative Models: Hidden Markov Models

Fall Semester 2024

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

Penn
UNIVERSITY of PENNSYLVANIA

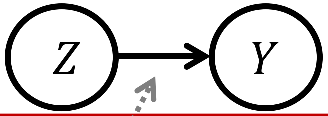
Taxonomy of Generative Models



Hidden Markov Models (Pictorial Definition)

Latent Variable Models

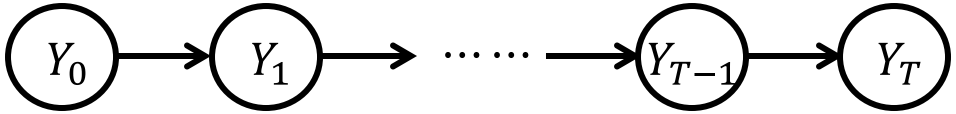
- Z: Latent variable
- Y: Observation



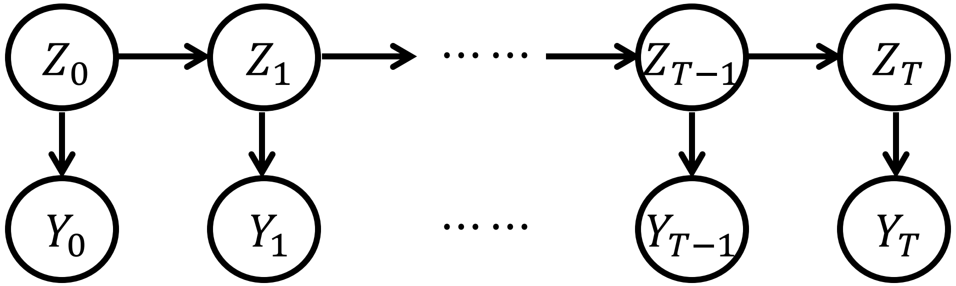
Read it: Y depends on Z

Markov Models

- Y_t: Observation (state)



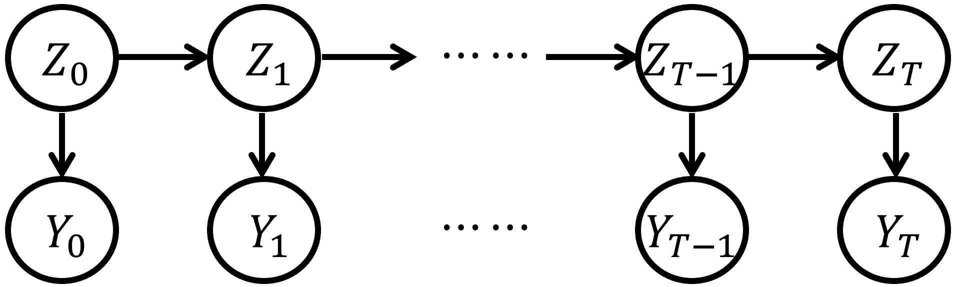
Hidden Markov Models



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

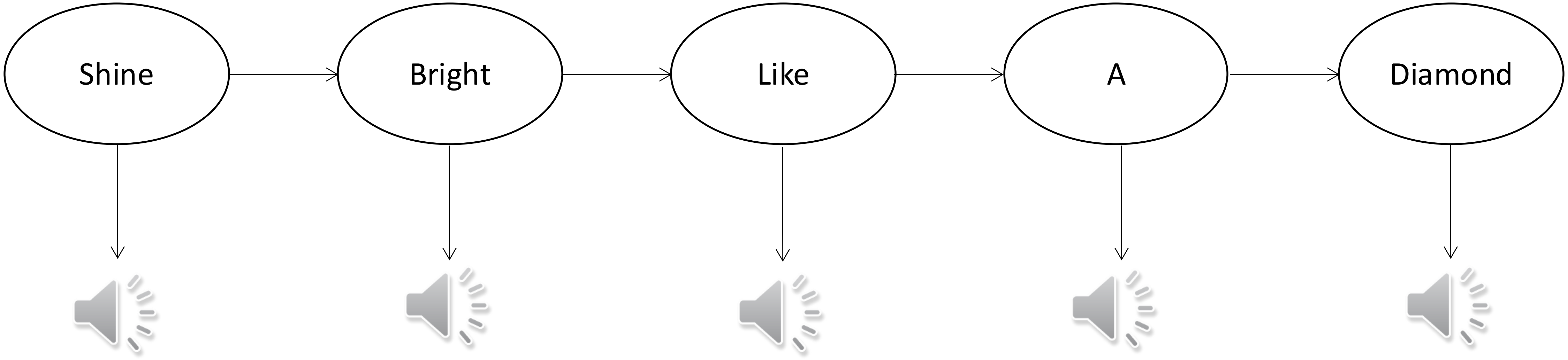
Example: Speech Recognition

Hidden Markov Models

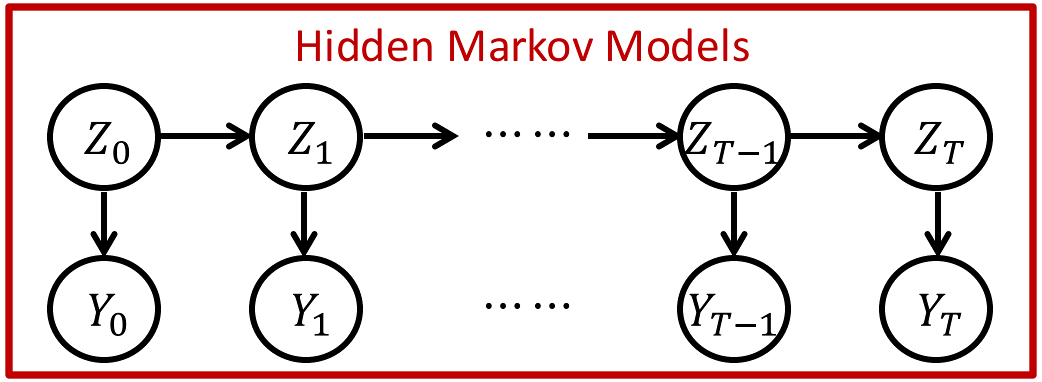


- Z_t: Words (hidden states)
- Y_t: Speech (observable audio data)

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:

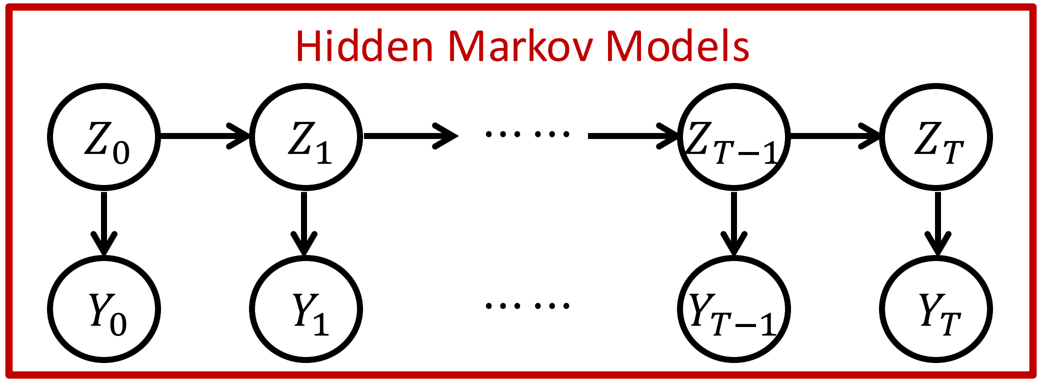
    State Space $\Omega = \{\omega_1, \dots, \omega_K\}$:
    Each Z_t takes values in Ω

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

- Initial Probability $\pi_1, \dots, \pi_K$: $\pi_i = \mathbb{P}(Z_0 = \omega_i)$
- Transition Probability a_{ij}:

$$a_{ij} := \mathbb{P}\left(Z_t = \omega_j \,\middle|\, Z_{t-1} = \omega_i\right)$$

- Emission Probability c_{jr}:

$$c_{jr} := \mathbb{P}\left(Y_t = \sigma_r \,\middle|\, Z_t = \omega_j\right)$$

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 $\theta := (\pi, A, C)$

Notations

- Studying HMMs need a lot of notations. We review some of them here and spell out the convention that we follow

  - $\boldsymbol{y} := [y_0, \ldots, y_T]^\top$ and similarly for $\boldsymbol{z} := [z_0, \ldots, z_T]^\top$

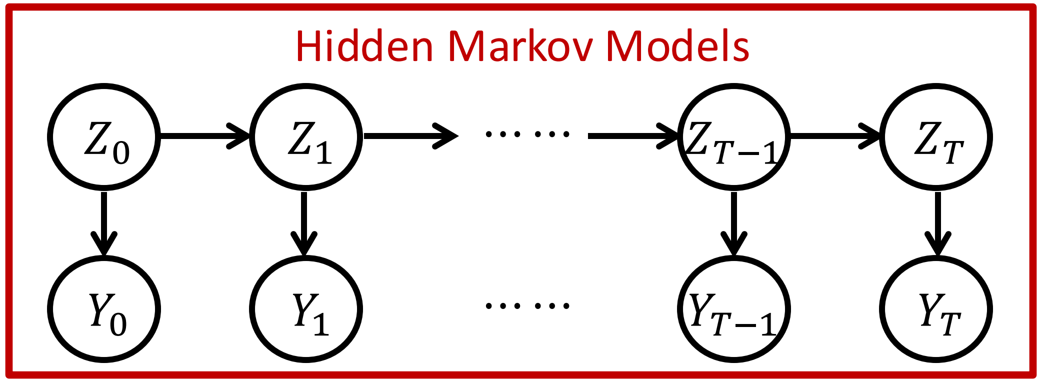
  - n-th observation $\boldsymbol{y}^{(n)} := \left[y_0^{(n)}, \ldots, y_T^{(n)}\right]^\top$

  - $\mathbb{P}(Z_0 = z_0, Z_1 = z_1)$: probability that $Z_0 = z_0$ and $Z_1 = z_1$

    - We might write $p(z_0, z_1)$ or $\mathbb{P}(z_0, Z_1 = z_1)$ for $\mathbb{P}(Z_0 = z_0, Z_1 = z_1)$ when there is no confusion

    - $\mathbb{P}_\theta(Z_0 = z_0, Z_1 = z_1)$ or $p_\theta(z_0, z_1)$ means the underlying probability distribution is parameterized by θ

Assumptions



- "Markov" Property:

$$p_\theta(z_t \mid z_0, \ldots, z_{t-1}, y_0 \ldots, y_{t-1}) = p_\theta(z_t | z_{t-1})$$

- Output Independence Assumption:

$$p_\theta(y_t | z_0, \ldots, z_T, y_0, \ldots, y_{t-1}, y_{t+1}, \ldots, y_T) = p_\theta(y_t | z_t)$$

- Consequences:

$$p_\theta(y_0, \ldots, y_T | z_0, \ldots, z_T) = \prod_{t=0}^{T} p_\theta(y_t | z_t)$$ Emission Probability

$$p_\theta(z_0, \ldots, z_T) = p_\theta(z_0) \prod_{t=1}^{T} p_\theta(z_t | z_{t-1})$$ Transition Probability

Inference, Decoding, and Learning

- Inference. Given θ, compute the likelihood $p_\theta(\boldsymbol{y})$ of observing $\boldsymbol{y}$

- Decoding. Given observation $\boldsymbol{y}$ and parameters θ, find best states:
$$\boldsymbol{z}^* \in \operatorname*{argmax}_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z})$$

- Learning. Given N observations $\left\{\boldsymbol{y}^{(n)}\right\}_{n=1}^{N}$, find best θ:
$$\max_\theta \prod_{n=1}^{N} p_\theta(\boldsymbol{y}^{(n)})$$

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

Decoding and Inference via "Brute-Force"

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

- Inference. Given $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$:

$$p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z}) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}|\boldsymbol{z}) p_\theta(\boldsymbol{z})$$

$$= \sum_{\boldsymbol{z}} \prod_{t=0}^{T} p_\theta(y_t|z_t)\, p_\theta(z_0) \prod_{t=1}^{T} p_\theta(z_t|z_{t-1})$$

- Decoding. Given $\theta, \boldsymbol{y}$, compute:

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

1. For all possible states $\boldsymbol{z}$, compute the likelihood $p_\theta(\boldsymbol{y}, \boldsymbol{z})$
2. Output the states 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 $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$
- If $T = 0$, then $p_\theta(\boldsymbol{y}) = \sum_{z_0} p_\theta(y_0, z_0) = \sum_{z_0} p_\theta(y_0|z_0) \cdot p(z_0)$
  - We can compute $p_\theta(\boldsymbol{y})$ in $O(K)$ time
- If $T = 1$, then $p_\theta(\boldsymbol{y}) = \sum_{z_1} p_\theta(y_0, y_1, z_1)$
  - Update $p_\theta(y_0, y_1, z_1)$ from $p_\theta(y_0, z_0)$:

$$\begin{aligned} p_\theta(y_0, y_1, z_1) &= \sum_{z_0} p_\theta(y_0, y_1, z_1|z_0) \cdot p_\theta(z_0) \\ &= \sum_{z_0} p_\theta(y_1|z_1) \cdot p_\theta(y_0|z_0) \cdot p_\theta(z_0) \cdot p_\theta(z_1|z_0) \\ &= p_\theta(y_1|z_1) \cdot \sum_{z_0} p_\theta(y_0, z_0) \cdot p_\theta(z_1|z_0) \end{aligned}$$

  - We can compute $p_\theta(\boldsymbol{y})$ in $O(K^2)$ time (need to sum over all possible z_1 and z_0)

Intuition. More generally, can we update $p_\theta(y_0, \dots, y_t, z_t)$ from $p_\theta(y_0, \dots, y_{t-1}, z_{t-1})$?
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 $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$

Intuition. More generally, can we update $p_\theta(y_0, \dots, y_t, z_t)$ from $p_\theta(y_0, \dots, y_{t-1}, z_{t-1})$?
If so, then we might be able to derive a faster algorithm for inference.

- Forward Probability:

$$\alpha_j(t) := \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$$

- By definition, we have

$$p_\theta(y_0, \dots, y_T) = \sum_{j=1}^K \mathbb{P}_\theta(y_0, \dots, y_T, Z_T = \omega_j) = \sum_{j=1}^K \alpha_j(T)$$

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

Formalize the Intuition

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

$c_j(y_t) \coloneqq \mathbb{P}_\theta(y_t | Z_t = \omega_j)$

- Inference. Given $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$
- Forward Probability:

$$\alpha_j(t) \coloneqq \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$$

- Recurrence Relation:

$$\alpha_j(t) = \mathbb{P}_\theta(y_t \mid y_0, \dots, y_{t-1}, Z_t = \omega_j) \cdot \mathbb{P}_\theta(y_0, \dots, y_{t-1}, Z_t = \omega_j)$$

$$= \mathbb{P}_\theta(y_t | Z_t = \omega_j) \sum_{i=1,\dots,K} \mathbb{P}_\theta(y_0, \dots, y_{t-1}, Z_t = \omega_j, Z_{t-1} = \omega_i)$$

$$= c_j(y_t) \sum_{i=1,\dots,K} \mathbb{P}_\theta(y_0, \dots, y_{t-1}, Z_{t-1} = \omega_i) \cdot \mathbb{P}_\theta(Z_t = \omega_j | y_0, \dots, y_{t-1}, Z_{t-1} = \omega_i)$$

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

Faster Algorithm for Inference

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

$c_j(y_t) \coloneqq \mathbb{P}_\theta(y_t | Z_t = \omega_j)$

Time Complexity: $O(TK^2)$

- Inference. Given $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$
- Forward Probability: $\alpha_j(t) \coloneqq \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$
- Recurrence Relation: $\alpha_j(t) = c_j(y_t) \sum_{i=1}^{K} \alpha_i(t-1) \cdot a_{ij}$

- Algorithm:
  - Initialization: $\alpha_j(0) = \pi_j \cdot c_j(y_0)$ $\quad \forall j = 1, \dots, K$
  - Recursion: $\alpha_j(t) = c_j(y_t) \sum_{i=1}^{K} \alpha_i(t-1) \cdot a_{ij}$ $\quad \forall j = 1, \dots, K, \forall t = 1, \dots, T$
  - Termination: Output $\sum_{j=1}^{K} \alpha_j(T)$

What About Decoding?

- Decoding. Given $\theta, \boldsymbol{y}$, compute:

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

- 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 $\sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z})$

- Two differences from inference:
  - Need to calculate the maximum of $p_\theta(\boldsymbol{y}, \boldsymbol{z})$ over $\boldsymbol{z}$ rather than sum

  - Need to find states $\boldsymbol{z}^*$ attaining the maximum

Inference Versus Decoding

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

- Inference. Given $\theta, \boldsymbol{y}$, compute:

$$p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z})$$

- Recurrence Term:

$$\alpha_j(t) \coloneqq \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$$

- Decomposition:

$$p_\theta(y_0, \dots, y_T) = \sum_{j=1}^K \alpha_j(T)$$

- Recurrence Relation:

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

$$\boxed{c_j(y_t) \coloneqq \mathbb{P}_\theta(y_t | Z_t = \omega_j)}$$

- Decoding. Given $\theta, \boldsymbol{y}$, compute:

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

- Recurrence Term:

$$v_j(t) \coloneqq \max_{z_0, \dots, z_{t-1} \in \Omega} \mathbb{P}_\theta(y_0, \dots, y_t, z_0, \dots, z_{t-1}, Z_t = \omega_j)$$

- Decomposition:

$$\max_{z_0, \dots, z_T \in \Omega} p_\theta(y_0, \dots, y_T, z_0, \dots, z_T) = \max_{j=1,\dots,K} v_j(T)$$

- Recurrence Relation: (proof omitted)

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

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 $\theta, \boldsymbol{y}$, compute:

$$p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}, \boldsymbol{z})$$

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

$$\alpha_j(0) = \pi_j \cdot c_j(y_0)$$

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

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

- Output:

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

$c_j(y_t) \coloneqq \mathbb{P}_\theta(y_t | Z_t = \omega_j)$

- Decoding. Given $\theta, \boldsymbol{y}$, compute:

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

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

$$v_j(0) = \pi_j \cdot c_j(y_0)$$

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

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

- Output:

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

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

Backtracing for Decoding

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

- How Does Backtracing Work:
  - z_T^* should maximize $v_j(T)$, i.e.,

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

    - $i^*(T)$: optimal index at time T
    - $v_{i^*(T)}(T)$: optimal value at time T
  - …
  - z_{t-1}^* should maximize $v_{i^*(t)}(t)$

    $i^*(t-1)$: optimal index at time $t-1$

    $v_{i^*(t-1)}(t-1)$: optimal value at time $t-1$

Remark. The final algorithm is known as *the Viterbi algorithm*. It is an instance of *dynamic programming*.

$$\boxed{c_j(y_t) \coloneqq \mathbb{P}_\theta(y_t | Z_t = \omega_j)}$$

Algorithm

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

  $$v_j(0) = \pi_j \cdot b_j(y_0)$$

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

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

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

- Output:

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

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

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

  $$i^*(t-1) = prev_{i^*(t)}(t) \qquad z_{t-1}^* = \omega_{i^*(t-1)}$$

Inference, Decoding, and Learning

- Inference. Given $\theta, \boldsymbol{y}$, compute $p_\theta(\boldsymbol{y})$
- Decoding. Given $\theta, \boldsymbol{y}$, compute:

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

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

$$\max_\theta \prod_{n=1}^{N} p_\theta(\boldsymbol{y}^{(n)})$$

- We've seen how to perform inference and decoding in $O(TK^2)$ time
  - sum vs. max, dynamic programming, backtracing
- We next study how to learn a hidden Markov model from data

Learning Hidden Markov Models

$$\boxed{\begin{aligned}\boldsymbol{y}^{(n)} &:= \left(y_0^{(n)}, \dots, y_T^{(n)}\right) \\ \boldsymbol{z} &:= (z_0, \dots, z_T)\end{aligned}}$$

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

$$\max_{\theta} \prod_{n=1}^{N} p_\theta\left(\boldsymbol{y}^{(n)}\right)$$

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

E-step:

$$q^k\left(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}\right) = p_{\theta^k}\left(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}\right)$$

M-step:

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

Substitute E-step into M-step

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

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

Instantiate EM for HMMs

$$\boldsymbol{y}^{(n)} := \left(y_0^{(n)}, \dots, y_T^{(n)}\right)$$
$$\boldsymbol{z} := (z_0, \dots, z_T)$$

$$\theta^{k+1} = \mathrm{argmax}_\theta \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \cdot \log p_\theta\left(\boldsymbol{y}^{(n)}, \boldsymbol{z}\right)$$

$$p_\theta\left(\boldsymbol{y}^{(n)}, \boldsymbol{z}\right) := \prod_{t=0}^{T} p_C(y_t^{(n)} | z_t)\, p_\pi(z_0) \prod_{t=1}^{T} p_A(z_t | z_{t-1})$$

$$(\pi^{k+1}, A^{k+1}, C^{k+1}) = \mathrm{argmax}_{\pi,A,C} \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \left(\log p_\pi(z_0) + \sum_{t=0}^{T} \log p_C\left(y_t^{(n)} \middle| z_t\right) + \sum_{t=1}^{T} \log p_A(z_t | z_{t-1}) \right)$$

The objective function is separable

$$\pi^{k+1} = \mathrm{argmax}_\pi \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0)$$

$$C^{k+1} = \mathrm{argmax}_C \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \sum_{t=0}^{T} \log p_C\left(y_t^{(n)} \middle| z_t\right)$$

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

Instantiate EM for HMMs

$$\boldsymbol{y}^{(n)} := \left(y_0^{(n)}, \dots, y_T^{(n)}\right)$$
$$\boldsymbol{z} := (z_0, \dots, z_T)$$

$$\pi^{k+1} = \text{argmax}_\pi \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0)$$

$$C^{k+1} = \text{argmax}_C \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \sum_{t=0}^{T} \log p_C\left(y_t^{(n)} \middle| z_t\right)$$

$$A^{k+1} = \text{argmax}_A \sum_{n=1}^{N} \sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \sum_{t=1}^{T} \log p_A(z_t \mid z_{t-1})$$

Note that the summation over $\boldsymbol{z}$ in the above can be simplified, e.g.,

$$\sum_{\boldsymbol{z}} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0) = \sum_{z_0,\dots,z_T} p_{\theta^k}(\boldsymbol{z} \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0) = \sum_{z_0} p_{\theta^k}(z_0 \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0)$$

This implies:

$$\pi^{k+1} = \text{argmax}_\pi \sum_{n=1}^{N} \sum_{z_0} p_{\theta^k}(z_0 \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0)$$

$$C^{k+1} = \text{argmax}_C \sum_{n=1}^{N} \sum_{z_t} p_{\theta^k}(z_t \mid \boldsymbol{y}^{(n)}) \sum_{t=0}^{T} \log p_C\left(y_t^{(n)} \middle| z_t\right)$$

$$A^{k+1} = \text{argmax}_A \sum_{n=1}^{N} \sum_{z_t, z_{t-1}} p_{\theta^k}(z_t, z_{t-1} \mid \boldsymbol{y}^{(n)}) \sum_{t=1}^{T} \log p_A(z_t \mid z_{t-1})$$

Updating π^{k+1}

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

$\boldsymbol{y}^{(n)} := \left(y_0^{(n)}, \dots, y_T^{(n)}\right)$

$$\pi^{k+1} = \operatorname{argmax}_\pi \sum_{n=1}^{N} \sum_{z_0} p_{\theta^k}(z_0 \mid \boldsymbol{y}^{(n)}) \cdot \log p_\pi(z_0)$$

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

$$\pi^{k+1} = \operatorname{argmax}_\pi \sum_{n=1}^{N} \sum_{i=1}^{K} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)}) \log(\pi_i) \qquad \text{subject to} \qquad \pi_1 + \cdots + \pi_K = 1$$

We know how to find its closed-form solution via the method of Lagrangian multipliers

$$\pi_i^{k+1} = \frac{\sum_{n=1}^{N} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})}{\sum_{n=1}^{N} \sum_{i=1}^{K} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})} = \frac{\sum_{n=1}^{N} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})}{N}, \qquad \forall i = 1, \dots, K$$

Remark. We will calculate $\mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})$ later

Updating A^{k+1}

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

$\boldsymbol{y}^{(n)} := \left(y_0^{(n)}, \dots, y_T^{(n)}\right)$

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

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

$$A^{k+1} = \operatorname{argmax}_A \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{j=1}^{K} \sum_{i=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right) \log a_{ij} \qquad \text{subject to} \sum_{j=1}^{K} a_{ij} = 1 \ (\forall i)$$

The objective function and constraints are separable

$$(a_{i1}^{k+1}, \dots, a_{iK}^{k+1}) = \operatorname{argmax}_A \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{j=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right) \log a_{ij} \qquad \text{subject to} \sum_{j=1}^{K} a_{ij} = 1$$

We know how to find its closed-form solution via the method of Lagrangian multipliers

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)}{\sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{j=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)}{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}\left(Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)} \qquad (\forall i, j)$$

Remark. We will calculate $\mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)$ and $\mathbb{P}_{\theta^k}\left(Z_{t-1} = \omega_i \middle| \boldsymbol{y}^{(n)}\right)$ 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 Y_t takes values in Σ

$$C^{k+1} = \mathrm{argmax}_C \sum_{n=1}^{N} \sum_{z_t} \mathbb{P}_{\theta^k}(z_t \,|\boldsymbol{y}^{(n)}) \sum_{t=0}^{T} \log p_C\left(y_t^{(n)} \middle| z_t\right)$$

$$\boldsymbol{y}^{(n)} \coloneqq \left(y_0^{(n)}, \dots, y_T^{(n)}\right)$$

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

$$\begin{aligned} C^{k+1} &= \mathrm{argmax}_C \textstyle\sum_{n=1}^{N} \sum_{t=0}^{T} \sum_{j=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \log p_B\left(y_t^{(n)} \middle| z_t = \omega_j\right) \\ &= \mathrm{argmax}_C \textstyle\sum_{n=1}^{N} \sum_{t=0}^{T} \sum_{j=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \sum_{r=1}^{K} \mathbb{I}\left(y_t^{(n)} = \sigma_r\right) \log c_{jr} \end{aligned} \quad \text{s.t. } \textstyle\sum_{r=1}^{K} c_{jr} = 1 \;(\forall j)$$

The objective function and constraints are separable

$$(c_{j1}^{k+1}, \dots, c_{jK}^{k+1}) = \mathrm{argmax}_C \sum_{n=1}^{N} \sum_{t=0}^{T} \sum_{r=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right) \log c_{jr} \quad \text{s.t. } \sum_{r=1}^{K} c_{jr} = 1$$

We know how to find its closed-form solution via the method of Lagrangian multipliers

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \sum_{r=1}^{K} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right)} \quad (\forall j, r)$$

Remark. We will calculate $\mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,|\boldsymbol{y}^{(n)}\right)$ later

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

$$\pi_i^{k+1} = \frac{\sum_{n=1}^{N} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})}{N} \qquad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \boldsymbol{y}^{(n)})}{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}(Z_{t-1} = \omega_i \mid \boldsymbol{y}^{(n)})} \qquad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j \mid \boldsymbol{y}^{(n)}) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j \mid \boldsymbol{y}^{(n)})} \qquad (\forall j, r)$$

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

$$\sum_{n=1}^{N} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)}) = \sum_{n=1}^{N} \mathbb{E}_{\theta^k}[\mathbb{I}(Z_0 = \omega_i) \mid \boldsymbol{y}^{(n)}]$$

- Answer:

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

We can interpret a_{ij}^{k+1} and c_{jr}^{k+1} similarly

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

$$\pi_i^{k+1} = \frac{\sum_{n=1}^{N} \mathbb{P}_{\theta^k}(Z_0 = \omega_i \mid \boldsymbol{y}^{(n)})}{N} \qquad (\forall i)$$

$$a_{ij}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \boldsymbol{y}^{(n)})}{\sum_{n=1}^{N} \sum_{t=1}^{T} \mathbb{P}_{\theta^k}(Z_{t-1} = \omega_i \mid \boldsymbol{y}^{(n)})} \qquad (\forall i, j)$$

$$c_{jr}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j \mid \boldsymbol{y}^{(n)}) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \mathbb{P}_{\theta^k}(Z_t = \omega_j \mid \boldsymbol{y}^{(n)})} \qquad (\forall j, r)$$

- To proceed, it suffices to calculate the following quantities $(\forall i, j)$:
  - $\xi_{ij}(t, n) \coloneqq \mathbb{P}_{\theta^k}(Z_t = \omega_j, Z_{t-1} = \omega_i \mid \boldsymbol{y}^{(n)})$ $(\forall t > 0)$
  - $\gamma_j(t, n) \coloneqq \mathbb{P}_{\theta^k}(Z_t = \omega_j \mid \boldsymbol{y}^{(n)})$ $(\forall t \geq 0)$
- Indeed, we have:

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

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

$$b_{jr}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \gamma_j(t, n) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \gamma_j(t, n)} \qquad (\forall j, r)$$

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

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

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

$$b_{jr}^{k+1} = \frac{\sum_{n=1}^{N} \sum_{t=0}^{T} \gamma_j(t,n) \mathbb{I}\left(y_t^{(n)} = \sigma_r\right)}{\sum_{n=1}^{N} \sum_{t=0}^{T} \gamma_j(t,n)} \qquad (\forall j,r)$$

- $\xi_{ij}(t,n) \coloneqq \mathbb{P}_{\theta^k}\left(Z_t = \omega_j, Z_{t-1} = \omega_i \,\middle|\, \boldsymbol{y}^{(n)}\right)$ $(\forall t > 0)$
- $\gamma_j(t,n) \coloneqq \mathbb{P}_{\theta^k}\left(Z_t = \omega_j \,\middle|\, \boldsymbol{y}^{(n)}\right)$ $(\forall t \geq 0)$

- Dropping indices n, k for clarity, we can reformulate our goal as follows:

Given $\boldsymbol{y} = (y_1, \ldots, y_T)$ and θ, compute $(\forall i, j)$

- $\xi_{ij}(t) = \mathbb{P}_\theta\left(Z_t = \omega_j, Z_{t-1} = \omega_i \,\middle|\, y_0, \ldots, y_T\right)$ $\qquad (t > 0)$
- $\gamma_j(t) = \mathbb{P}_\theta\left(Z_t = \omega_j \,\middle|\, y_0, \ldots, y_T\right)$ $\qquad (t \geq 0)$

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

$\alpha_j(t) \coloneqq \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$
$\beta_j(t) \coloneqq \mathbb{P}_\theta(y_{t+1}, \dots, y_T \mid Z_t = \omega_j)$

Given $\boldsymbol{y} = (y_1, \dots, y_T)$ and θ, compute $(\forall i, j)$

- $\xi_{ij}(t) = \mathbb{P}_\theta\left(Z_t = \omega_j, Z_{t-1} = \omega_i \,\middle|\, y_0, \dots, y_T\right)$ $\quad (t > 0)$
- $\gamma_j(t) = \mathbb{P}_\theta\left(Z_t = \omega_j \,\middle|\, y_0, \dots, y_T\right)$ $\quad (t \geq 0)$

- Intuition: $\xi_{ij}(t)$ and $\gamma_j(t)$ involve the posterior, but we have no formula for it
  - we need to convert them into likelihood via Bayes rule

- Applying the Bayes rule to $\gamma_j(t)$ gives

$$\gamma_j(t) = \frac{\mathbb{P}_\theta(y_0,\dots,y_t,Z_t=\omega_j)\cdot \mathbb{P}_\theta(y_{t+1},\dots,y_T \mid Z_t=\omega_j)}{\mathbb{P}_\theta(y_0,\dots,y_T)} = \frac{\alpha_j(t)\cdot \mathbb{P}_\theta(y_{t+1},\dots,y_T \mid Z_t=\omega_j)}{\sum_{i=1}^K \alpha_i(t)\cdot \mathbb{P}_\theta(y_{t+1},\dots,y_T \mid Z_t=\omega_i)} = \frac{\alpha_j(t)\beta_j(t)}{\sum_{i=1}^K \alpha_i(t)\beta_i(t)}$$

  - $\beta_j(t)$ 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 $\gamma_j(t)$

$\alpha_j(t) := \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$
$\beta_j(t) := \mathbb{P}_\theta(y_{t+1}, \dots, y_T \mid Z_t = \omega_j)$

$c_j(y_t) := \mathbb{P}_\theta(y_t | Z_t = \omega_j)$

Given $\boldsymbol{y} = (y_1, \dots, y_T)$ and θ, compute $(\forall i, j)$
- $\xi_{ij}(t) = \mathbb{P}_\theta\left(Z_t = \omega_j, Z_{t-1} = \omega_i \mid y_0, \dots, y_T\right) \qquad (t > 0)$
- $\gamma_j(t) = \mathbb{P}_\theta\left(Z_t = \omega_j \mid y_0, \dots, y_T\right) \qquad (t \geq 0)$

- $\gamma_j(t) = \dfrac{\alpha_j(t)\beta_j(t)}{\sum_{i=1}^K \alpha_i(t)\beta_i(t)}$

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

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

Remark. Now we only need to compute $\alpha_j(t)$ and $\beta_j(t)$
- We've seen how to update $\alpha_j(t)$

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

$\alpha_j(t) := \mathbb{P}_\theta(y_0, \dots, y_t, Z_t = \omega_j)$
$\beta_j(t) := \mathbb{P}_\theta(y_{t+1}, \dots, y_T \mid Z_t = \omega_j)$

$c_j(y_t) := \mathbb{P}_\theta(y_t | Z_t = \omega_j)$

- Compute forward probability $\alpha_j(t)$:
  - Initialization: $\alpha_j(1) = \pi_j \cdot b_j(x_1)$ $\quad\forall j = 1, \dots, K$
  - Recursion: $\alpha_j(t) = c_j(y_t) \sum_{i=1}^{K} \alpha_i(t-1) \cdot a_{ij}$ $\quad\forall j = 1, \dots, K, \forall t = 1, \dots, T$

- Compute backward probability $\beta_j(t)$: (details omitted, please verify)
  - Initialization: $\beta_j(T) = 1$ $\quad\forall j = 1, \dots, K$
  - Recursion: $\beta_j(t-1) = \sum_{i=1}^{K} c_i(y_t) \cdot \beta_i(t) \cdot a_{ji}$ $\quad\forall j = 1, \dots, K, \forall t = T, \dots, 1$

Put It Together

- Exercise. Put all the previous derivations together and you will get a concrete EM algorithm for learning HMMs from data samples

- Remark. This learning algorithm has a name: “Baum-Welch algorithm”