SDS 383C: Statistical Modeling I
Fall 2022, Module VIII

Abhra Sarkar

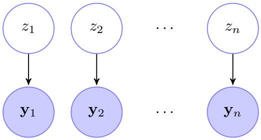
Department of Statistics and Data Sciences
The University of Texas at Austin

"All models are wrong, but some are useful."- George E. P. Box

Hidden Markov Models

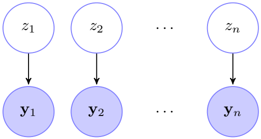
$$(z_i \mid \boldsymbol{\pi}) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi})$$

$$(\mathbf{y}_i \mid z_i = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_i \mid \boldsymbol{\xi}_k)$$

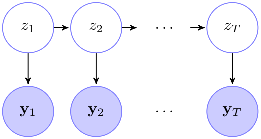


Hidden Markov Models

$$(z_i \mid \boldsymbol{\pi}) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi})$$
$$(\mathbf{y}_i \mid z_i = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_i \mid \boldsymbol{\xi}_k)$$



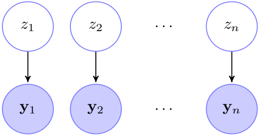
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

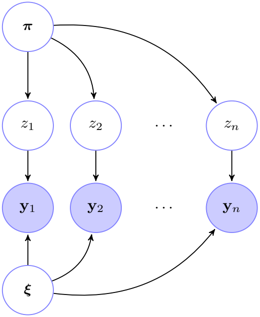


Hidden Markov Models

$$(z_i \mid \boldsymbol{\pi}) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi})$$

$$(\mathbf{y}_i \mid z_i = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_i \mid \boldsymbol{\xi}_k)$$

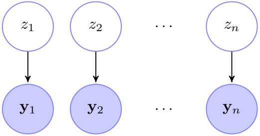


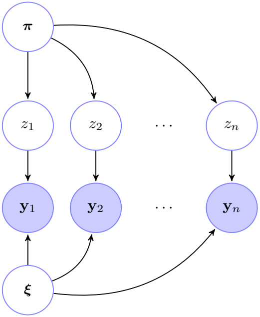


Hidden Markov Models

$$(z_i \mid \boldsymbol{\pi}) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi})$$

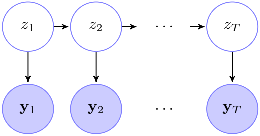
$$(\mathbf{y}_i \mid z_i = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_i \mid \boldsymbol{\xi}_k)$$

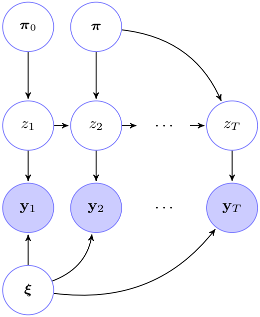




$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

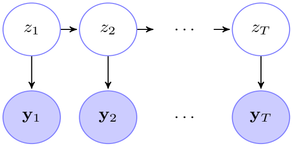




Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

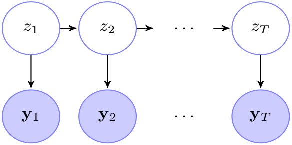


- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- Three main components:

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

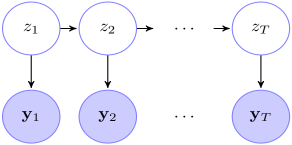


- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

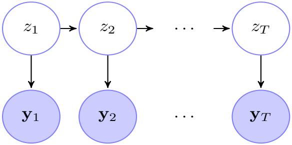


- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

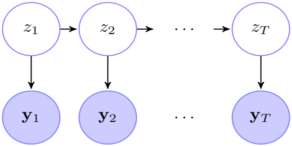


- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
    - **Initial distribution:** $p(z_1)$
    - **Transition distribution:** $p(z_t \mid z_{t-1})$
    - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

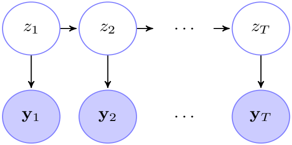


- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$
- **Bayesian formulation:**

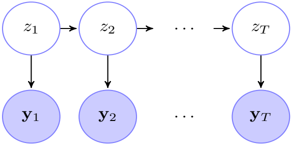
$$(\mathbf{y}_t \mid \boldsymbol{\psi}, \{\boldsymbol{\xi}_j\}_{j=1}^{K}, z_t = k) \sim p(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_k), \qquad p(z_t = k \mid z_{t-1} = j) = \pi_{j,k}$$

$$\boldsymbol{\xi}_k \overset{iid}{\sim} p_0(\boldsymbol{\xi}_k), \qquad \boldsymbol{\psi} \sim p_0(\boldsymbol{\psi}), \qquad \boldsymbol{\pi}_j \overset{ind}{\sim} p_0(\boldsymbol{\pi}_j)$$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

▶ **HMM with Normal location emissions:**

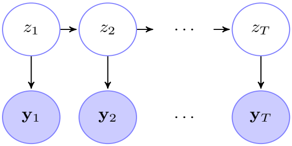
$$(\mathbf{y}_t \mid \sigma^2, \{\mu_j\}_{j=1}^{K}, z_t = k) \sim \text{Normal}(\mathbf{y}_t \mid \mu_k, \sigma^2), \qquad p(z_t = k \mid z_{t-1} = j) = \pi_{j,k}$$

$$\mu_k \overset{iid}{\sim} \text{Normal}(\mu_0, \sigma_0^2), \qquad \sigma^2 \sim \text{Inv-Ga}(a_0, b_0), \qquad \boldsymbol{\pi}_j \overset{ind}{\sim} \text{Dir}(\alpha_{j,1}, \ldots, \alpha_{j,K})$$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

▶ **HMM with Normal location-scale emissions:**

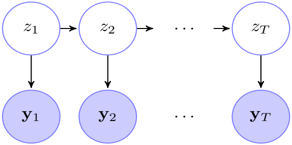
$$(y_t \mid \{(\mu_j, \sigma_j^2)\}_{j=1}^K, z_t = k) \sim \text{Normal}(\mathbf{y}_t \mid \mu_k, \sigma_k^2), \qquad p(z_t = k \mid z_{t-1} = j) = \pi_{j,k}$$

$$(\mu_k, \sigma_k^2) \overset{iid}{\sim} \text{Normal}(\mu_0, \sigma_0^2) \cdot \text{Inv-Ga}(a_0, b_0), \qquad \boldsymbol{\pi}_j \overset{ind}{\sim} \text{Dir}(\alpha_{j,1}, \ldots, \alpha_{j,K})$$

Likelihood Factorization

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \stackrel{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \stackrel{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- $p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T}) = p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\}$
- **Three main components:**
  - **Initial distribution:** $p(z_1)$
  - **Transition distribution:** $p(z_t \mid z_{t-1})$
  - **Emission distribution:** $p(\mathbf{y}_t \mid z_t)$

▶ **HMM with Poisson emissions:**

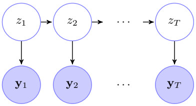
$$(y_t \mid \{\lambda_j\}_{j=1}^K, z_t = k) \sim \text{Poisson}(\mathbf{y}_t \mid \lambda_k), \qquad p(z_t = k \mid z_{t-1} = j) = \pi_{j,k}$$

$$\lambda_k \stackrel{iid}{\sim} \text{Ga}(a, b), \qquad \boldsymbol{\pi}_j \stackrel{ind}{\sim} \text{Dir}(\alpha_{j,1}, \ldots, \alpha_{j,K})$$

Marginal Likelihood

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



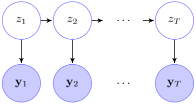
- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:**
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Marginal Likelihood

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

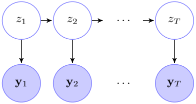


- **Marginal likelihood:**
$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$
- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:**
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

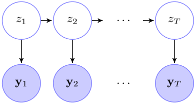
$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:**
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:**

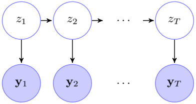
$$\begin{aligned} \alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_1, \ldots, \mathbf{y}_{t+1}, z_{t+1}) &= \textstyle\sum_{z_t} p(\mathbf{y}_1, \ldots, \mathbf{y}_{t+1}, z_{t+1}, z_t) \\ &= p(\mathbf{y}_{t+1} \mid z_{t+1}) \textstyle\sum_{z_t} p(\mathbf{y}_1, \ldots, \mathbf{y}_t \mid z_t) p(z_{t+1} \mid z_t) p(z_t) \\ &= p(\mathbf{y}_{t+1} \mid z_{t+1}) \textstyle\sum_{z_t} p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t) p(z_{t+1} \mid z_t) \\ &= p(\mathbf{y}_{t+1} \mid z_{t+1}) \textstyle\sum_{z_t} p(z_{t+1} \mid z_t) \alpha_t(z_t) \end{aligned}$$

- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



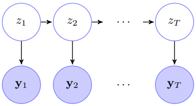
- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t) \alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



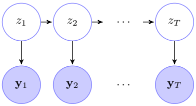
- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t) \alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

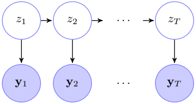
$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \left\{ \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) \right\} p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t) \alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1, z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$
- Complexity of the algorithm is $K + K^2(T-1) + K \approx TK^2$.

Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

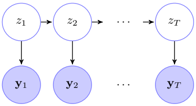
$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:**
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$

Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T, z_{t+1} \mid z_t)$

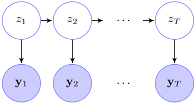
$$= \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(\mathbf{y}_{t+2}, \ldots, \mathbf{y}_T \mid z_{t+1}) p(z_{t+1} \mid z_t)$$

$$= \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$$

- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$

Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

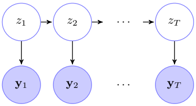


- **Marginal likelihood:**
$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) p(\mathbf{z}_{1:T})$$
- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$

Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) p(\mathbf{z}_{1:T})$$

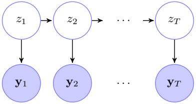
- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_1} p(\mathbf{y}_{1:T} \mid z_1) p(z_1)$

$$= \sum_{z_1} p(\mathbf{y}_{2:T} \mid z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$$

$$= \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$$

Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

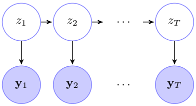


- **Marginal likelihood:**

$$p(\mathbf{y}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} p(\mathbf{y}_{1:T} \mid \mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) = \sum_{z_1} \cdots \sum_{z_T} \prod_{t=1}^{T} p(\mathbf{y}_t \mid z_t) p(\mathbf{z}_{1:T})$$

- Complexity of a bruce-force algorithm is $\approx TK^T$.
- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:** $p(\mathbf{y}_{1:T}) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$
- Complexity of the algorithm is $K^2(T-1) + K \approx TK^2$.

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

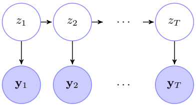


- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t) \alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1 \mid z_1) p(z_1)$
- **Marginal likelihood:** $L = p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$ $\rightarrow$ numerically unstable!

Forward Algorithm - Numerical Issues

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t)\alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1 \mid z_1)p(z_1)$
- **Marginal likelihood:** $L = p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$ $\rightarrow$ numerically unstable!
- **Normalized forward messages:** $\phi_{\alpha,t}(z_t) = \frac{\alpha_t(z_t)}{\sum_z \alpha_t(z)} = \frac{\alpha_t(z_t)}{w_{\alpha,t}}$
- $\phi_{\alpha,T}(z_T) = \frac{\alpha_T(z_T)}{w_{\alpha,T}} = \frac{w_{\alpha,T-1}}{w_{\alpha,T}} \frac{w_{\alpha,T-2}}{w_{\alpha,T-1}} \cdots \frac{w_{\alpha,2}}{w_{\alpha,3}} \frac{w_{\alpha,1}}{w_{\alpha,2}} \frac{\alpha_T(z_T)}{w_{\alpha,1}}$

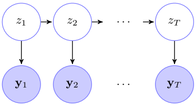
$$\Rightarrow \sum_{z_T} \phi_{\alpha,T}(z_T) = \frac{w_{\alpha,T-1}}{w_{\alpha,T}} \frac{w_{\alpha,T-2}}{w_{\alpha,T-1}} \cdots \frac{w_{\alpha,2}}{w_{\alpha,3}} \frac{w_{\alpha,1}}{w_{\alpha,2}} \frac{1}{w_{\alpha,1}} \left\{ \sum_{z_T} \alpha_T(z_T) \right\} = 1$$

$$\Rightarrow \sum_{z_T} \alpha_T(z_T) = \frac{w_{\alpha,T}}{w_{\alpha,T-1}} \frac{w_{\alpha,T-1}}{w_{\alpha,T-2}} \cdots \frac{w_{\alpha,3}}{w_{\alpha,2}} \frac{w_{\alpha,2}}{w_{\alpha,1}} w_{\alpha,1} = L$$

$$\Rightarrow \log L = \log w_{\alpha,1} + \sum_{t=2}^{T} \log \left(\frac{w_{\alpha,t}}{w_{\alpha,t-1}} \right)$$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Forward messages:** $\alpha_t(z_t) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t)$
- **Recursion:** $\alpha_{t+1}(z_{t+1}) = p(\mathbf{y}_{t+1} \mid z_{t+1}) \sum_{z_t} p(z_{t+1} \mid z_t)\alpha_t(z_t)$
- **Initial condition:** $\alpha_1(z_1) = p(y_1 \mid z_1)p(z_1)$
- **Marginal likelihood:** $L = p(\mathbf{y}_{1:T}) = \sum_{z_T} \alpha_T(z_T)$ $\rightarrow$ numerically unstable!
- **Normalized forward messages:** $\phi_{\alpha,t}(z_t) = \frac{\alpha_t(z_t)}{\sum_z \alpha_t(z)} = \frac{\alpha_t(z_t)}{w_{\alpha,t}}$
- $\phi_{\alpha,T}(z_T) = \frac{\alpha_T(z_T)}{w_{\alpha,T}} = \frac{w_{\alpha,T-1}}{w_{\alpha,T}} \frac{w_{\alpha,T-2}}{w_{\alpha,T-1}} \cdots \frac{w_{\alpha,2}}{w_{\alpha,3}} \frac{w_{\alpha,1}}{w_{\alpha,2}} \frac{\alpha_T(z_T)}{w_{\alpha,1}}$

  $\Rightarrow \sum_{z_T} \phi_{\alpha,T}(z_T) = \frac{w_{\alpha,T-1}}{w_{\alpha,T}} \frac{w_{\alpha,T-2}}{w_{\alpha,T-1}} \cdots \frac{w_{\alpha,2}}{w_{\alpha,3}} \frac{w_{\alpha,1}}{w_{\alpha,2}} \frac{1}{w_{\alpha,1}} \left\{ \sum_{z_T} \alpha_T(z_T) \right\} = 1$

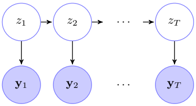
  $\Rightarrow \sum_{z_T} \alpha_T(z_T) = \frac{w_{\alpha,T}}{w_{\alpha,T-1}} \frac{w_{\alpha,T-1}}{w_{\alpha,T-2}} \cdots \frac{w_{\alpha,3}}{w_{\alpha,2}} \frac{w_{\alpha,2}}{w_{\alpha,1}} w_{\alpha,1} = L$

  $\Rightarrow \log L = \log w_{\alpha,1} + \sum_{t=2}^{T} \log \left(\frac{w_{\alpha,t}}{w_{\alpha,t-1}} \right)$ $\rightarrow$ numerically stable!

Backward Algorithm - Numerical Issues

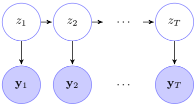
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:**
  $L = \sum_{z_1} \widetilde{\beta}_1(z_1) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$ $\rightarrow$ numerically unstable!

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:**
  $L = \sum_{z_1} \widetilde{\beta}_1(z_1) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$ $\rightarrow$ numerically unstable!
- **Normalized backward messages:** $\phi_{\beta,t}(z_t) = \frac{\beta_t(z_t)}{\sum_z \beta_t(z)} = \frac{\beta_t(z_t)}{w_{\beta,t}}$
- $\widetilde{\phi}_{\beta,1}(z_1) = \frac{\widetilde{\beta}_1(z_1)}{\widetilde{w}_{\beta,1}} = \frac{1}{w_{\beta,T}} \frac{w_{\beta,T}}{w_{\beta,T-1}} \cdots \frac{w_{\beta,2}}{w_{\beta,1}} \frac{w_{\beta,1}}{\widetilde{w}_{\beta,1}} \widetilde{\beta}_1(z_1)$

  $\Rightarrow \sum_{z_1} \widetilde{\phi}_{\beta,1}(z_1) = \frac{1}{w_{\beta,T}} \frac{w_{\beta,T}}{w_{\beta,T-1}} \cdots \frac{w_{\beta,2}}{w_{\beta,1}} \frac{w_{\beta,1}}{\widetilde{w}_{\beta,1}} \left\{ \sum_{z_1} \widetilde{\beta}_1(z_1) \right\} = 1$

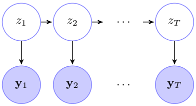
  $\Rightarrow \sum_{z_1} \widetilde{\beta}_1(z_1) = w_{\beta,T} \frac{w_{\beta,T-1}}{w_{\beta,T}} \cdots \frac{w_{\beta,1}}{w_{\beta,2}} \frac{\widetilde{w}_{\beta,1}}{w_{\beta,1}} = L$

  $\Rightarrow \log L = \log w_{\beta,T} + \sum_{t=2}^{T} \log\left(\frac{w_{\beta,t-1}}{w_{\beta,t}}\right) + \log\left(\frac{\widetilde{w}_{\beta,1}}{w_{\beta,1}}\right)$

Backward Algorithm - Numerical Issues

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Backward messages:** $\beta_t(z_t) = p(\mathbf{y}_{t+1}, \dots, \mathbf{y}_T \mid z_t)$
- **Recursion:** $\beta_t(z_t) = \sum_{z_{t+1}} p(\mathbf{y}_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \beta_{t+1}(z_{t+1})$
- **Final condition:** $\beta_T(z_T) = 1$
- **Marginal likelihood:**
  $L = \sum_{z_1} \widetilde{\beta}_1(z_1) = \sum_{z_1} \beta_1(z_1) p(\mathbf{y}_1 \mid z_1) p(z_1)$ $\rightarrow$ numerically unstable!
- **Normalized backward messages:** $\phi_{\beta,t}(z_t) = \frac{\beta_t(z_t)}{\sum_z \beta_t(z)} = \frac{\beta_t(z_t)}{w_{\beta,t}}$
- $\widetilde{\phi}_{\beta,1}(z_1) = \frac{\widetilde{\beta}_1(z_1)}{\widetilde{w}_{\beta,1}} = \frac{1}{w_{\beta,T}} \frac{w_{\beta,T}}{w_{\beta,T-1}} \dots \frac{w_{\beta,2}}{w_{\beta,1}} \frac{w_{\beta,1}}{\widetilde{w}_{\beta,1}} \widetilde{\beta}_1(z_1)$

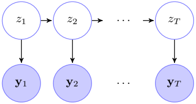
  $\Rightarrow \sum_{z_1} \widetilde{\phi}_{\beta,1}(z_1) = \frac{1}{w_{\beta,T}} \frac{w_{\beta,T}}{w_{\beta,T-1}} \dots \frac{w_{\beta,2}}{w_{\beta,1}} \frac{w_{\beta,1}}{\widetilde{w}_{\beta,1}} \left\{ \sum_{z_1} \widetilde{\beta}_1(z_1) \right\} = 1$

  $\Rightarrow \sum_{z_1} \widetilde{\beta}_1(z_1) = w_{\beta,T} \frac{w_{\beta,T-1}}{w_{\beta,T}} \dots \frac{w_{\beta,1}}{w_{\beta,2}} \frac{\widetilde{w}_{\beta,1}}{w_{\beta,1}} = L$

  $\Rightarrow \log L = \log w_{\beta,T} + \sum_{t=2}^{T} \log \left(\frac{w_{\beta,t-1}}{w_{\beta,t}} \right) + \log \left(\frac{\widetilde{w}_{\beta,1}}{w_{\beta,1}} \right)$ $\rightarrow$ numerically stable!

Three Main Problems

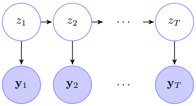
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t)$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
- **Prediction:** $p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

Three Main Problems

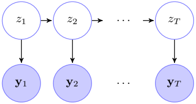
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t)$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
- **Prediction:** $p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

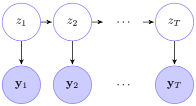
Three Main Problems

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t)$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
- **Prediction:** $p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \frac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
- **Prediction:**

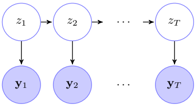
$$p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$
$$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) p(z_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$
$$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) \frac{\alpha_T(z_T)}{\sum_z \alpha_T(z)}$$

$$p(\mathbf{y}_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \sum_{z_{T+m}} p(\mathbf{y}_{T+m} \mid z_{T+m}) p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

Filtering, Smoothing & Prediction

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
- **Prediction:**

$p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

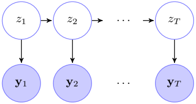
$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) p(z_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) \frac{\alpha_T(z_T)}{\sum_z \alpha_T(z)}$

$p(\mathbf{y}_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \sum_{z_{T+m}} p(\mathbf{y}_{T+m} \mid z_{T+m}) p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

Filtering, Smoothing & Prediction

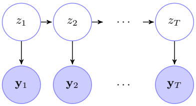
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $$\begin{aligned}\gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) &= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_T \mid z_t) p(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)} \\ &= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_t \mid z_t) p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t) p(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)} \\ &= \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t) p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_T \mid z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)} \\ &= \frac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}\end{aligned}$$
  - $\Rightarrow \alpha_t(z_t)\beta_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) p(\mathbf{y}_1, \ldots, \mathbf{y}_T) = p(z_t, \mathbf{y}_1, \ldots, \mathbf{y}_T)$
    $\Rightarrow \sum_z \alpha_t(z)\beta_t(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T)$

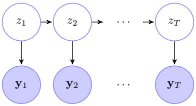
Filtering, Smoothing & Prediction

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



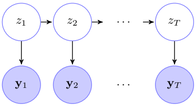
- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}$
  - $\Rightarrow \alpha_t(z_t)\beta_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)p(\mathbf{y}_1, \ldots, \mathbf{y}_T) = p(z_t, \mathbf{y}_1, \ldots, \mathbf{y}_T)$
    $\Rightarrow \sum_z \alpha_t(z)\beta_t(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T)$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}$
  - $\Rightarrow \alpha_t(z_t)\beta_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)p(\mathbf{y}_1, \ldots, \mathbf{y}_T) = p(z_t, \mathbf{y}_1, \ldots, \mathbf{y}_T)$
    $\Rightarrow \sum_z \alpha_t(z)\beta_t(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ $\rightarrow$ Free of t!

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

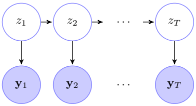


- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}$
  - $\Rightarrow \alpha_t(z_t)\beta_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)p(\mathbf{y}_1, \ldots, \mathbf{y}_T) = p(z_t, \mathbf{y}_1, \ldots, \mathbf{y}_T)$
    $\Rightarrow \sum_z \alpha_t(z)\beta_t(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\Rightarrow \sum_z \alpha_s(z)\beta_s(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T) \quad \forall s$
  - $\Rightarrow \gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)} = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{\sum_z \alpha_s(z)\beta_s(z)}$
- **Prediction:**
  $p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  $= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) p(z_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

Filtering, Smoothing & Prediction

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



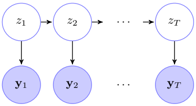
- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)}$
  - $\Rightarrow \alpha_t(z_t)\beta_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)p(\mathbf{y}_1, \ldots, \mathbf{y}_T) = p(z_t, \mathbf{y}_1, \ldots, \mathbf{y}_T)$
    $\Rightarrow \sum_z \alpha_t(z)\beta_t(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T)$
  - $\Rightarrow \sum_z \alpha_s(z)\beta_s(z) = \sum_z p(z, \mathbf{y}_1, \ldots, \mathbf{y}_T) = p(\mathbf{y}_1, \ldots, \mathbf{y}_T) \quad \forall s$
  - $\Rightarrow \gamma_t(z_t) = p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_T)} = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{\sum_z \alpha_s(z)\beta_s(z)}$
- **Prediction:**

$$p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

$$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) p(z_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$$

Filtering, Smoothing & Prediction

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

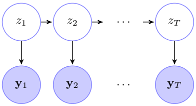


- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{\sum_z \alpha_s(z)\beta_s(z)}$
- **Prediction:**

$$\begin{aligned}
&p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) \\
&\quad = \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) p(z_T \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) \\
&\quad = \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) \tfrac{\alpha_T(z_T)}{\sum_z \alpha_T(z)} \\
&p(\mathbf{y}_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \sum_{z_{T+m}} p(\mathbf{y}_{T+m} \mid z_{T+m}) p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)
\end{aligned}$$

Filtering, Smoothing & Prediction - Summary

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \stackrel{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \stackrel{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Filtering:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_t) = \dfrac{\alpha_t(z_t)}{\sum_z \alpha_t(z)}$
- **Smoothing:** $p(z_t \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \dfrac{\alpha_t(z_t)\beta_t(z_t)}{\sum_z \alpha_s(z)\beta_s(z)}$
- **Prediction:** $p(z_{T+m} \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$

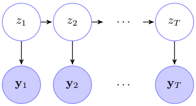
$$= \sum_{z_{T+m-1}} p(z_{T+m} \mid z_{T+m-1}) \cdots \sum_{z_T} p(z_{T+1} \mid z_T) \frac{\alpha_T(z_T)}{\sum_z \alpha_T(z)}$$

EM Algorithm - the General Case

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}), \quad \mathbf{y} \to \text{observed}, \ \mathbf{z} \to \text{latent}$$
$$p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}) = p(\mathbf{z}, \mathbf{y} \mid \boldsymbol{\theta}) / p(\mathbf{y} \mid \boldsymbol{\theta}) \ \Rightarrow \ p(\mathbf{y} \mid \boldsymbol{\theta}) = p(\mathbf{z}, \mathbf{y} \mid \boldsymbol{\theta}) / p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta})$$

- **Log-likelihood:** $\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{y} \mid \boldsymbol{\theta}) = \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) - \log p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta})$
- $\mathcal{L}(\boldsymbol{\theta}) = \log p(\mathbf{y} \mid \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} \log p(\mathbf{y} \mid \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} \left\{ \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) - \log p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}) \right\}$
- $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \left\{ \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) - \log p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}) \right\}$
  $= \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)}) \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) - \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)}) \log p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta})$
  $= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$
- $\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta}^{(m)}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) - Q(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(m)})$
- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $\mathcal{L}(\boldsymbol{\theta}^{(m+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(m)}) \geq Q(\boldsymbol{\theta}^{(m+1)}, \boldsymbol{\theta}^{(m)}) - Q(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(m)}) \geq 0$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \{p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta}) p(z_t \mid z_{t-1}, \boldsymbol{\theta})\}$

$$= p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid z_t = k, \boldsymbol{\theta}) p(z_t = k \mid z_{t-1} = j, \boldsymbol{\theta})\}^{1(z_{t-1}=j, z_t=k)}$$

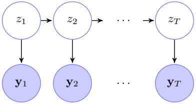
$$= p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$$

- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$

EM Algorithm - HMM

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

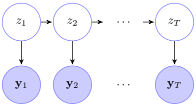


- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$
- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = \mathcal{L}_1 + \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$
  $\approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$

EM Algorithm - HMM

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Iterative algorithm:**

  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.

  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.

  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.

- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$

- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$

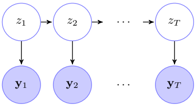
- $\pi_{t,j,k} = \mathbb{E}\{1(z_{t-1} = j, z_t = k) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}\} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta})$

  $\propto p(z_{t-1} = j, \mathbf{y}_{1:(t-1)} \mid \boldsymbol{\theta}) p(z_t = k \mid z_{t-1} = j, \boldsymbol{\theta}) p(\mathbf{y}_t \mid z_t = k, \boldsymbol{\theta}) p(\mathbf{y}_{(t+1):T} \mid z_t = k, \boldsymbol{\theta})$

  $\propto \alpha_{t-1}(j)\ \pi(j,k)\ p(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_k)\ \beta_t(k)$

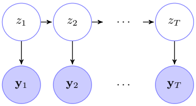
EM Algorithm - HMM

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$
- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$
- $\pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)})$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

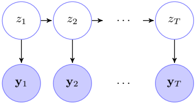


- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$
- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$
- $\pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)})$
- **E-step:**
  $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$
  $= \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$
  $= \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \{\log p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) + \log \pi_{j,k}\}$

EM Algorithm - HMM

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$

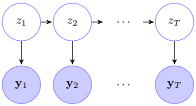


- **Iterative algorithm:**
  Starting with some $\boldsymbol{\theta}^{(0)}$, iteratively update $\boldsymbol{\theta}^{(m)}$ until convergence.
  (a) **E-step:** Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)})} \log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta})$.
  (b) **M-step:** Compute $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$.
- $p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) = p(\mathbf{y}_1 \mid z_1, \boldsymbol{\theta}) p(z_1 \mid \boldsymbol{\theta}) \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}^{1(z_{t-1}=j, z_t=k)}$
- $\log p(\mathbf{y}, \mathbf{z} \mid \boldsymbol{\theta}) \approx \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} 1(z_{t-1} = j, z_t = k) \log \{p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \pi_{j,k}\}$
- $\pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)})$
- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \{\log p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) + \log \pi_{j,k}\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = (\boldsymbol{\pi}^{(m+1)}, \boldsymbol{\xi}^{(m+1)}) = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

EM Algorithm - Normal Location-Scale Mixtures

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$



- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ \log p(y_t \mid \mu_k, \sigma_k^2) + \log \pi_{j,k} \right\}$

$$= \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ -\frac{1}{2} \log \sigma_k^2 - \frac{(y_t - \mu_k)^2}{2\sigma_k^2} + \log \pi_{j,k} \right\}$$

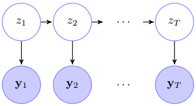
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j \left(\sum_{k=1}^{K} \pi_{j,k} - 1 \right) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \mu_k} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \sigma_k^2} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with} \quad \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

$$\Rightarrow \mu_k^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)} y_t}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}, \quad \sigma_k^{2(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)} (y_t - \mu_k^{(m+1)})^2}{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)}}.$$

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$



- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ -\frac{1}{2} \log \sigma_k^2 - \frac{(y_t - \mu_k)^2}{2\sigma_k^2} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j (\sum_{k=1}^{K} \pi_{j,k} - 1) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \mu_k} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \sigma_k^2} = 0$$

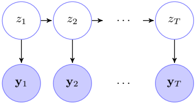
$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with} \quad \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

$$\Rightarrow \mu_k^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)} y_t}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}, \quad \sigma_k^{2(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)} (y_t - \mu_k^{(m+1)})^2}{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)}}.$$

EM Algorithm - Normal Location-Scale Mixtures

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$



- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ -\frac{1}{2} \log \sigma_k^2 - \frac{(y_t - \mu_k)^2}{2\sigma_k^2} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

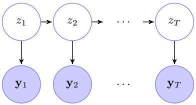
$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j \left(\sum_{k=1}^{K} \pi_{j,k} - 1 \right) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \mu_k} = 0, \quad \frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})}{\partial \sigma_k^2} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \text{ with } \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

$$\Rightarrow \mu_k^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)} y_t}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}, \quad \sigma_k^{2(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)} (y_t - \mu_k^{(m+1)})^2}{\sum_{t=2}^{T} \sum_{j=1}^{K} \widetilde{\pi}_{t,j,k}^{(m+1)}}.$$

Baum-Welch Algorithm - EM for Multinomial Emissions

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \equiv \text{Mult}(1, \mathbf{b}_k)$$



- $p(y_t = y \mid z_t = k, \mathbf{b}_k) = b_{k,y}, \quad y \in \{1, \ldots, M\}$
- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ \sum_{y=1}^{M} 1(y_t = y) \log b_{k,y} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j (\sum_{k=1}^{K} \pi_{j,k} - 1) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with} \quad \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

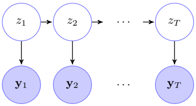
$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{k=1}^{K} \widetilde{\lambda}_k (\sum_{y=1}^{M} b_{k,y} - 1) \right\}}{\partial b_{k,y} / \widetilde{\lambda}_k} = 0$$

$$\Rightarrow b_{k,y}^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} 1(y_t = y) \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}.$$

Baum-Welch Algorithm - EM for Multinomial Emissions

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \equiv \text{Mult}(1, \mathbf{b}_k)$$



- $p(y_t = y \mid z_t = k, \mathbf{b}_k) = b_{k,y}, \quad y \in \{1, \ldots, M\}$
- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ \sum_{y=1}^{M} 1(y_t = y) \log b_{k,y} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j (\sum_{k=1}^{K} \pi_{j,k} - 1) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with} \quad \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

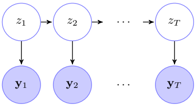
$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{k=1}^{K} \widetilde{\lambda}_k (\sum_{y=1}^{M} b_{k,y} - 1) \right\}}{\partial b_{k,y} / \widetilde{\lambda}_k} = 0$$

$$\Rightarrow b_{k,y}^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} 1(y_t = y) \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}.$$

Baum-Welch Algorithm - EM for Multinomial Emissions

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \equiv \text{Mult}(1, \mathbf{b}_k)$$



- $p(y_t = y \mid z_t = k, \mathbf{b}_k) = b_{k,y}, \quad y \in \{1, \ldots, M\}$
- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ \sum_{y=1}^{M} 1(y_t = y) \log b_{k,y} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

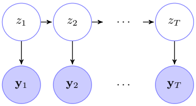
$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j (\sum_{k=1}^{K} \pi_{j,k} - 1) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with} \quad \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{k=1}^{K} \widetilde{\lambda}_k (\sum_{y=1}^{M} b_{k,y} - 1) \right\}}{\partial b_{k,y} / \widetilde{\lambda}_k} = 0$$

$$\Rightarrow b_{k,y}^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} 1(y_t = y) \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}.$$

Baum-Welch Algorithm - EM for Multinomial Emissions



$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \mathrm{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k) \equiv \mathrm{Mult}(1, \mathbf{b}_k)$$

- $p(y_t = y \mid z_t = k, \mathbf{b}_k) = b_{k,y}, \quad y \in \{1, \ldots, M\}$
- **E-step:** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) = \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)} \left\{ \sum_{y=1}^{M} 1(y_t = y) \log b_{k,y} + \log \pi_{j,k} \right\}$
- **M-step:** $\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$

$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{j=1}^{K} \lambda_j (\sum_{k=1}^{K} \pi_{j,k} - 1) \right\}}{\partial \pi_{j,k} / \partial \lambda_j} = 0$$

$$\Rightarrow \pi_{j,k}^{(m+1)} = \frac{\sum_{t=2}^{T} \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{k=1}^{K} \pi_{t,j,k}^{(m)}} \quad \text{with } \pi_{t,j,k}^{(m)} = p(z_{t-1} = j, z_t = k \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(m)}),$$

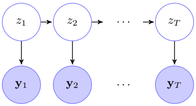
$$\frac{\partial \left\{ Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)}) + \sum_{k=1}^{K} \widetilde{\lambda}_k (\sum_{y=1}^{M} b_{k,y} - 1) \right\}}{\partial b_{k,y} / \widetilde{\lambda}_k} = 0$$

$$\Rightarrow b_{k,y}^{(m+1)} = \frac{\sum_{t=2}^{T} \sum_{j=1}^{K} 1(y_t = y) \pi_{t,j,k}^{(m)}}{\sum_{t=2}^{T} \sum_{j=1}^{K} \pi_{t,j,k}^{(m)}}.$$

Viterbi Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- Most likely path of the latent sequence:

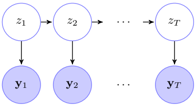
$$\widehat{\mathbf{z}}_{1:T} = \max_{\mathbf{z}_{1:T}} p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T})$$
$$= \max_{\mathbf{z}_{1:T}} \left[p(z_1)p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1})p(\mathbf{y}_t \mid z_t)\} \right]$$

- **Algorithm:**
  (a) For each t and each k, calculate
  $$v_{1,k} = p(\mathbf{y}_1 \mid z_1 = k)p(z_1 = k), \quad \text{and}$$
  $$v_{t,k} = \max_j \{p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j}\},$$
  $$z^*_{t,k} = \arg\max_j \{p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j}\}.$$
  (b) Iteratively set $\quad \widehat{z}_T = \arg\max_k v_{T,k}$ and
  $$\widehat{z}_{t-1} = z^*_{t,\widehat{z}_t}.$$

- Complexity of the algorithm is $\approx TK^2$.

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- Most likely path of the latent sequence:

$$\widehat{\mathbf{z}}_{1:T} = \max_{\mathbf{z}_{1:T}} p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T})$$
$$= \max_{\mathbf{z}_{1:T}} \left[p(z_1) p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{ p(z_t \mid z_{t-1}) p(\mathbf{y}_t \mid z_t) \} \right]$$

- **Algorithm:**

  (a) For each t and each k, calculate

$$v_{1,k} = p(\mathbf{y}_1 \mid z_1 = k) p(z_1 = k), \quad \text{and}$$
$$v_{t,k} = \max_j \left\{ p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j} \right\},$$
$$z^\star_{t,k} = \arg\max_j \left\{ p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j} \right\}.$$

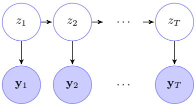
  (b) Iteratively set $\widehat{z}_T = \arg\max_k v_{T,k}$, and

$$\widehat{z}_{t-1} = z^\star_{t,\widehat{z}_t}.$$

- Complexity of the algorithm is $\approx TK^2$.

Viterbi Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- Most likely path of the latent sequence:

$$\widehat{\mathbf{z}}_{1:T} = \max_{\mathbf{z}_{1:T}} p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T})$$
$$= \max_{\mathbf{z}_{1:T}} \left[p(z_1) p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{ p(z_t \mid z_{t-1}) p(\mathbf{y}_t \mid z_t) \} \right]$$

- **Algorithm:**

  (a) For each t and each k, calculate

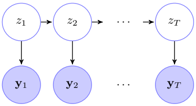
$$v_{1,k} = p(\mathbf{y}_1 \mid z_1 = k) p(z_1 = k), \quad \text{and}$$
$$v_{t,k} = \max_j \left\{ p(\mathbf{y}_t \mid z_t = k) \, \pi_{j,k} \, v_{t-1,j} \right\},$$
$$z^\star_{t,k} = \arg\max_j \left\{ p(\mathbf{y}_t \mid z_t = k) \, \pi_{j,k} \, v_{t-1,j} \right\}.$$

  (b) Iteratively set $\widehat{z}_T = \arg\max_k v_{T,k}$, and

$$\widehat{z}_{t-1} = z^\star_{t, \widehat{z}_t}.$$

- Complexity of the algorithm is $\approx TK^2$.

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- Most likely path of the latent sequence:

$$\widehat{\mathbf{z}}_{1:T} = \max_{\mathbf{z}_{1:T}} p(\mathbf{y}_{1:T}, \mathbf{z}_{1:T})$$
$$= \max_{\mathbf{z}_{1:T}} \left[p(z_1) p(\mathbf{y}_1 \mid z_1) \prod_{t=2}^{T} \{p(z_t \mid z_{t-1}) p(\mathbf{y}_t \mid z_t)\} \right]$$

- **Algorithm:**

  (a) For each t and each k, calculate

$$v_{1,k} = p(\mathbf{y}_1 \mid z_1 = k) p(z_1 = k), \quad \text{and}$$
$$v_{t,k} = \max_j \left\{ p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j} \right\},$$
$$z^\star_{t,k} = \arg\max_j \left\{ p(\mathbf{y}_t \mid z_t = k)\, \pi_{j,k}\, v_{t-1,j} \right\}.$$

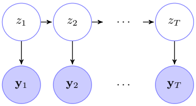
  (b) Iteratively set $\widehat{z}_T = \arg\max_k v_{T,k}$, and

$$\widehat{z}_{t-1} = z^\star_{t, \widehat{z}_t}.$$

- Complexity of the algorithm is $\approx TK^2$.

MCMC

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Updating the Parameters of the Transition Distribution:**
  - The conditional posterior of $\boldsymbol{\pi}_j$ is

$$p(\boldsymbol{\pi}_j \mid \mathbf{z}_{1:T}) \propto p_0(\boldsymbol{\pi}_j) \prod_k \pi_{j,k}^{n_{j,k}},$$

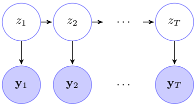
where $n_{j,k} = \sum_{t=2}^{T} 1\{z_{t-1} = j, z_t = k\}$ is the number of transitions from j to k.

  - With $p_0(\boldsymbol{\pi}_j) = \text{Dir}(\alpha_{j,1}, \ldots, \alpha_{j,K})$, the full conditional is simplified as

$$p(\boldsymbol{\pi}_j \mid \mathbf{z}_{1:T}) = \text{Dir}(\alpha_{j,1} + n_{j,1}, \ldots, \alpha_{j,K} + n_{j,K}).$$

MCMC

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k)$$



- **Updating the Parameters of the Transition Distribution:**
  - The conditional posterior of $\boldsymbol{\pi}_j$ is

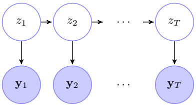
$$p(\boldsymbol{\pi}_j \mid \mathbf{z}_{1:T}) \propto p_0(\boldsymbol{\pi}_j) \prod_k \pi_{j,k}^{n_{j,k}},$$

    where $n_{j,k} = \sum_{t=2}^{T} 1\{z_{t-1} = j, z_t = k\}$ is the number of transitions from j to k.
  - With $p_0(\boldsymbol{\pi}_j) = \text{Dir}(\alpha_{j,1}, \ldots, \alpha_{j,K})$, the full conditional is simplified as

$$p(\boldsymbol{\pi}_j \mid \mathbf{z}_{1:T}) = \text{Dir}(\alpha_{j,1} + n_{j,1}, \ldots, \alpha_{j,K} + n_{j,K}).$$

MCMC

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the Parameters of the Emission Distribution:**
  - The conditional posterior of $\boldsymbol{\psi}$ is

$$p(\boldsymbol{\psi} \mid \mathbf{y}_{1:T}, \boldsymbol{\xi}) \propto p_0(\boldsymbol{\psi}) \prod_{t=1}^{T} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_{z_t}).$$

  - The conditional posterior of $\boldsymbol{\xi}_k$ is

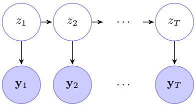
$$p(\boldsymbol{\xi}_k \mid \mathbf{z}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\psi}) \propto p_0(\boldsymbol{\xi}_k) \prod_{\{t: z_t = k\}} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_k).$$

  - The full conditionals are simplified if $p_0(\boldsymbol{\psi})$ and $p_0(\boldsymbol{\xi})$ are conjugate to $f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi})$.

MCMC

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the Parameters of the Emission Distribution:**
  - The conditional posterior of $\boldsymbol{\psi}$ is

$$p(\boldsymbol{\psi} \mid \mathbf{y}_{1:T}, \boldsymbol{\xi}) \propto p_0(\boldsymbol{\psi}) \prod_{t=1}^{T} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_{z_t}).$$

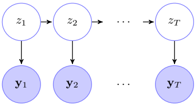
  - The conditional posterior of $\boldsymbol{\xi}_k$ is

$$p(\boldsymbol{\xi}_k \mid \mathbf{z}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\psi}) \propto p_0(\boldsymbol{\xi}_k) \prod_{\{t:z_t=k\}} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_k).$$

  - The full conditionals are simplified if $p_0(\boldsymbol{\psi})$ and $p_0(\boldsymbol{\xi})$ are conjugate to $f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi})$.

MCMC

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the Parameters of the Emission Distribution:**
  - The conditional posterior of $\boldsymbol{\psi}$ is

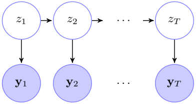
$$p(\boldsymbol{\psi} \mid \mathbf{y}_{1:T}, \boldsymbol{\xi}) \propto p_0(\boldsymbol{\psi}) \prod_{t=1}^{T} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_{z_t}).$$

  - The conditional posterior of $\boldsymbol{\xi}_k$ is

$$p(\boldsymbol{\xi}_k \mid \mathbf{z}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{\psi}) \propto p_0(\boldsymbol{\xi}_k) \prod_{\{t: z_t = k\}} f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi}_k).$$

  - The full conditionals are simplified if $p_0(\boldsymbol{\psi})$ and $p_0(\boldsymbol{\xi})$ are conjugate to $f(\mathbf{y}_t \mid \boldsymbol{\psi}, \boldsymbol{\xi})$.

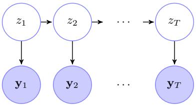
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
    - Define forward messages $\alpha_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t \mid \boldsymbol{\theta})$, with boundary condition $\alpha_1(z_1 \mid \boldsymbol{\theta}) = p(y_1 \mid z_1, \boldsymbol{\theta})\pi_0(z_1 \mid \boldsymbol{\theta})$. The following recursion holds
    $$\alpha_{t+1}(z_{t+1} \mid \boldsymbol{\theta}) = p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}) \textstyle\sum_{z_t} p(z_{t+1} \mid z_t, \boldsymbol{\theta})\alpha_t(z_t \mid \boldsymbol{\theta}).$$
    - The joint conditional posterior of the latent states factorizes as
    $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{y}_{1:T}, \boldsymbol{\theta})\, p(z_{T-1} \mid z_T, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \ \ldots \ p(z_1 \mid \mathbf{z}_{2:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$
    where $p(z_t \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{p(\mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})$
    $$\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(\mathbf{z}_{(t+1):T} \mid z_t, \mathbf{y}_{1:t}, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta})$$
    $$\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(z_{(t+1)} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta})$$
    $$= \alpha_t(z_t \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta}).$$
    - To sample $\mathbf{z}_{1:T}$, first pass messages $\alpha_t(z_t \mid \boldsymbol{\theta})$ forwards, then sample backwards.

MCMC - The Forward-Backward Algorithm

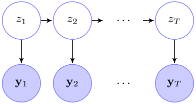
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
  - Define forward messages $\alpha_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t \mid \boldsymbol{\theta})$, with boundary condition $\alpha_1(z_1 \mid \boldsymbol{\theta}) = p(y_1 \mid z_1, \boldsymbol{\theta})\pi_0(z_1 \mid \boldsymbol{\theta})$. The following recursion holds
  $$\alpha_{t+1}(z_{t+1} \mid \boldsymbol{\theta}) = p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}) \textstyle\sum_{z_t} p(z_{t+1} \mid z_t, \boldsymbol{\theta})\alpha_t(z_t \mid \boldsymbol{\theta}).$$
  - The joint conditional posterior of the latent states factorizes as
  $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{y}_{1:T}, \boldsymbol{\theta})\, p(z_{T-1} \mid z_T, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \;\ldots\; p(z_1 \mid \mathbf{z}_{2:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$
  where $p(z_t \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{p(\mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})$
  $$\begin{aligned}
  &\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(\mathbf{z}_{(t+1):T} \mid z_t, \mathbf{y}_{1:t}, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta}) \\
  &\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(z_{(t+1)} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta}) \\
  &= \alpha_t(z_t \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta}).
  \end{aligned}$$
  - To sample $\mathbf{z}_{1:T}$, first pass messages $\alpha_t(z_t \mid \theta)$ forwards, then sample backwards.

MCMC - The Forward-Backward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
  - Define forward messages $\alpha_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t \mid \boldsymbol{\theta})$, with boundary condition $\alpha_1(z_1 \mid \boldsymbol{\theta}) = p(y_1 \mid z_1, \boldsymbol{\theta})\pi_0(z_1 \mid \boldsymbol{\theta})$. The following recursion holds

    $$\alpha_{t+1}(z_{t+1} \mid \boldsymbol{\theta}) = p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}) \textstyle\sum_{z_t} p(z_{t+1} \mid z_t, \boldsymbol{\theta})\alpha_t(z_t \mid \boldsymbol{\theta}).$$

  - The joint conditional posterior of the latent states factorizes as

    $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{y}_{1:T}, \boldsymbol{\theta})\, p(z_{T-1} \mid z_T, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \ \ldots \ p(z_1 \mid \mathbf{z}_{2:T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$

    where $p(z_t \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{p(\mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})$

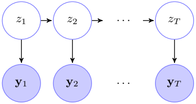
    $$\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(\mathbf{z}_{(t+1):T} \mid z_t, \mathbf{y}_{1:t}, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid z_t, \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta})$$
    $$\propto p(\mathbf{y}_{1:t}, z_t \mid \boldsymbol{\theta})\, p(z_{(t+1)} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{(t+1):T} \mid \mathbf{z}_{(t+1):T}, \mathbf{y}_{1:t}, \boldsymbol{\theta})$$
    $$= \alpha_t(z_t \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta}).$$

  - To sample $\mathbf{z}_{1:T}$, first pass messages $\alpha_t(z_t \mid \boldsymbol{\theta})$ forwards, then sample backwards.

MCMC - The Backward-Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
  - Define backward messages $\beta_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})$, with boundary condition $\beta_T(z_T) = 1$. The following recursion holds

    $$\beta_t(z_t \mid \boldsymbol{\theta}) = \textstyle\sum_{z_{t+1}} \beta_{t+1}(z_{t+1} \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}).$$

  - The joint conditional posterior of the latent states factorizes as

    $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{z}_{1:(T-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \ \ldots \ p(z_2 \mid z_1, \mathbf{y}_{1:T}, \boldsymbol{\theta}) p(z_1 \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$

    where $p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})}{p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})$

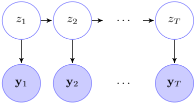
    $$\propto p(\mathbf{y}_{t:T} \mid \mathbf{z}_{1:t}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})$$
    $$\propto p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta})$$
    $$= \beta_t(z_t \mid \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta}).$$

  - To sample $\mathbf{z}_{1:T}$, first pass messages $\beta_t(z_t \mid \boldsymbol{\theta})$ backwards, then sample forwards.

MCMC - The Backward-Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$



- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
  - Define backward messages $\beta_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})$, with boundary condition $\beta_T(z_T) = 1$. The following recursion holds

    $$\beta_t(z_t \mid \boldsymbol{\theta}) = \textstyle\sum_{z_{t+1}} \beta_{t+1}(z_{t+1} \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}).$$

  - The joint conditional posterior of the latent states factorizes as

    $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{z}_{1:(T-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \ \ldots \ p(z_2 \mid z_1, \mathbf{y}_{1:T}, \boldsymbol{\theta}) p(z_1 \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$

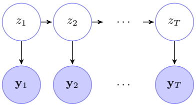
    where $p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})}{p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})$

    $$\begin{aligned}
    &\propto p(\mathbf{y}_{t:T} \mid \mathbf{z}_{1:t}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta}) \\
    &\propto p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta}) \\
    &= \beta_t(z_t \mid \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta}).
    \end{aligned}$$

  - To sample $\mathbf{z}_{1:T}$, first pass messages $\beta_t(z_t \mid \boldsymbol{\theta})$ backwards, then sample forwards.

MCMC - The Backward-Forward Algorithm

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}, \boldsymbol{\psi}) \overset{ind}{\sim} p(\mathbf{y}_t \mid \boldsymbol{\xi}_k, \boldsymbol{\psi})$$

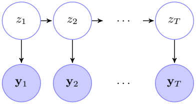


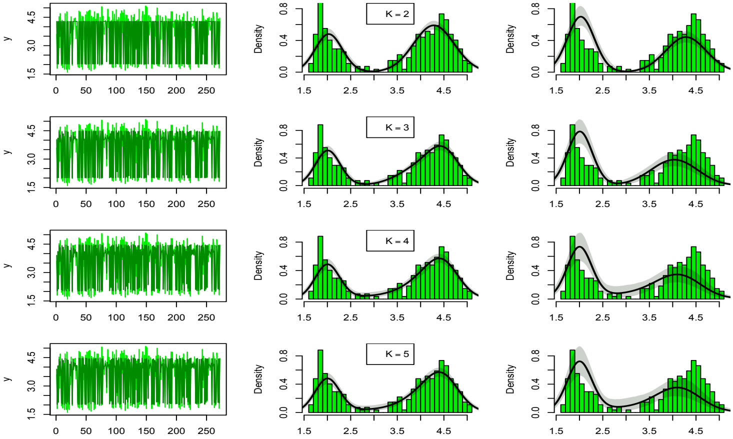
- **Updating the latent sequence $\mathbf{z}_{1:T}$:**
  - Define backward messages $\beta_t(z_t \mid \boldsymbol{\theta}) = p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})$, with boundary condition $\beta_T(z_T) = 1$. The following recursion holds
    $$\beta_t(z_t \mid \boldsymbol{\theta}) = \sum_{z_{t+1}} \beta_{t+1}(z_{t+1} \mid \boldsymbol{\theta})\, p(z_{t+1} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_{t+1} \mid z_{t+1}, \boldsymbol{\theta}).$$
  - The joint conditional posterior of the latent states factorizes as
    $$p(\mathbf{z}_{1:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) = p(z_T \mid \mathbf{z}_{1:(T-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \;\ldots\; p(z_2 \mid z_1, \mathbf{y}_{1:T}, \boldsymbol{\theta}) p(z_1 \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}),$$
    where $p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta}) = \dfrac{p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})}{p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})} \propto p(z_t, \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:T}, \boldsymbol{\theta})$
    $$\propto p(\mathbf{y}_{t:T} \mid \mathbf{z}_{1:t}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(z_t \mid \mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})\, p(\mathbf{z}_{1:(t-1)}, \mathbf{y}_{1:(t-1)}, \boldsymbol{\theta})$$
    $$\propto p(\mathbf{y}_{(t+1):T} \mid z_t, \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta})$$
    $$= \beta_t(z_t \mid \boldsymbol{\theta})\, p(\mathbf{y}_t \mid z_t, \boldsymbol{\theta})\, p(z_t \mid z_{t-1}, \boldsymbol{\theta}).$$
  - To sample $\mathbf{z}_{1:T}$, first pass messages $\beta_t(z_t \mid \boldsymbol{\theta})$ backwards, then sample forwards.

faithful$eruptions - Normal Location-Scale HMM - Gibbs Sampling

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$

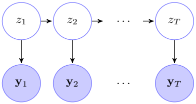


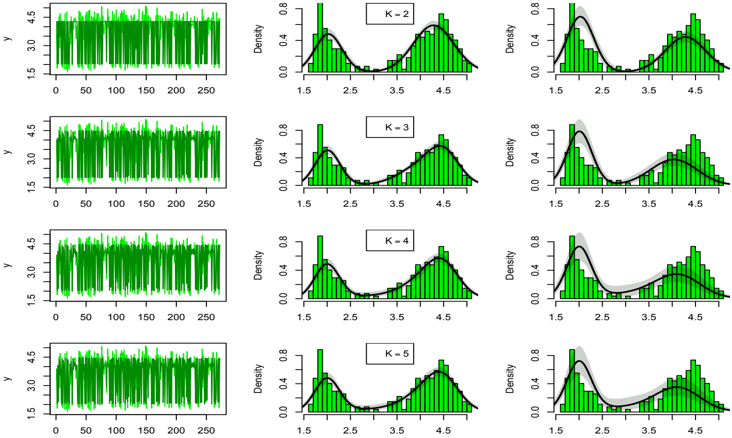


faithful$eruptions - Normal Location-Scale HMM - Gibbs Sampling

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$

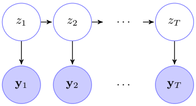


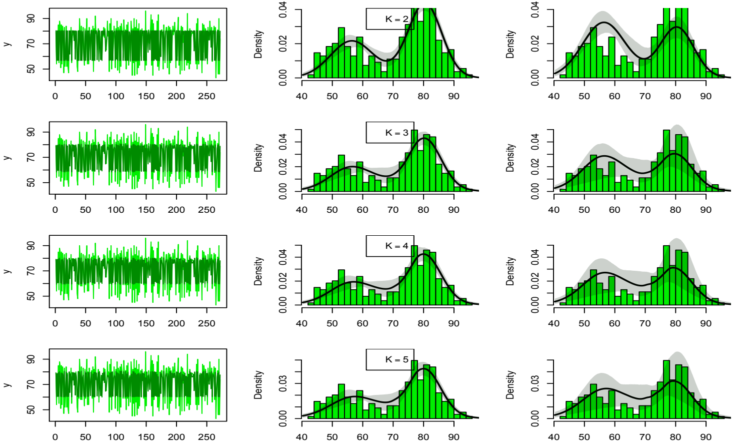


- Stationary distribution expected to match well with the histogram, predictive density NOT!

faithful$waiting - Normal Location-Scale HMM - Gibbs Sampling

$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \overset{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$
$$(y_t \mid z_t = k, \boldsymbol{\xi}) \overset{ind}{\sim} \text{Normal}(y_t \mid \mu_k, \sigma_k^2)$$

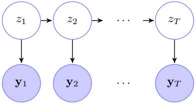


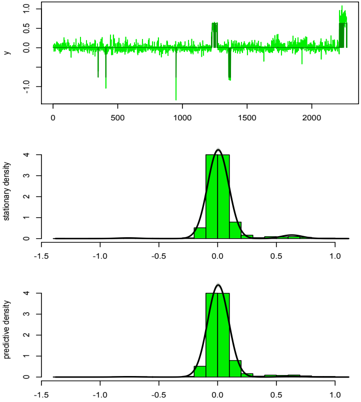


coriel aCGH data - Normal Location HMM - Gibbs Sampling

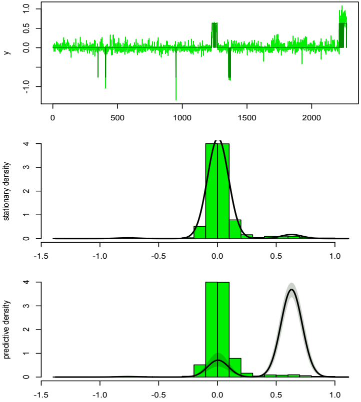
$$(z_t \mid \boldsymbol{\pi}, z_{t-1} = j) \stackrel{iid}{\sim} \text{Mult}(1, \boldsymbol{\pi}_j)$$

$$(\mathbf{y}_t \mid z_t = k, \boldsymbol{\xi}) \stackrel{ind}{\sim} \text{Normal}(\mathbf{y}_t \mid \mu_k, \sigma^2)$$





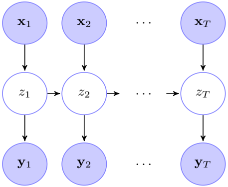
$T = 2271, \ y_T \approx 0.004$

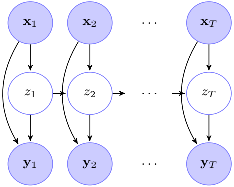


$T = 2270, \ y_T \approx 0.654$

Extensions

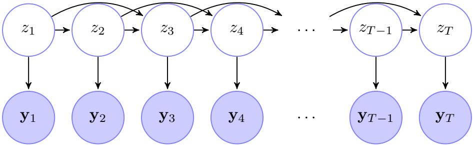
- Nonhomogeneous HMM
- Higher order HMM
- Switching vector autoregressive process





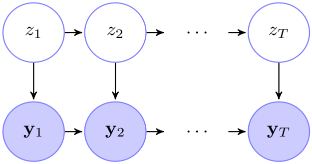
Extensions

- Nonhomogeneous HMM
- Higher order HMM
- Switching vector autoregressive process



Extensions

- Nonhomogeneous HMM
- Higher order HMM
- Switching vector autoregressive process



Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and [illegible]
    - Backward algorithm - define final messages and [illegible]
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages

Module VIII Summary

- Hidden Markov models
  - The three components that define an HMM are
    - Initial distribution $p(z_1)$
    - Transition distributions $p(z_t \mid z_{t-1})$
    - Emission distributions $p(\mathbf{y}_t \mid z_t)$
  - Brute force algorithms for estimation and inference are computationally infeasible.
  - Message passing algorithms provide computationally efficient alternatives.
    - Forward algorithm - define initial messages and recursively move forwards
    - Backward algorithm - define final messages and recursively move backwards
  - EM algorithms can be designed using forward/backward messages
  - Baum-Welch algorithm is EM for HMMs with multinomial emissions
  - Viterbi algorithm estimates the most likely path of the latent sequence
  - Efficient MCMC algorithms can be designed using forward/backward messages