Mathematical Explanation of the GLMHMM Class

1 Problem Definition

The **Generalized Linear Model Hidden Markov Model (GLMHMM)** combines:

1. A **Hidden Markov Model (HMM)** to model sequences with latent discrete states.
2. A **Generalized Linear Model (GLM)** to parameterize the emission distributions conditioned on the observed inputs.

Key Variables

- N: Number of samples (time steps in the sequence).
- K: Number of hidden states.
- D: Number of input features (including bias).
- C: Number of output(observations) dimensions.
- $\mathbf{X} \in \mathbb{R}^{N \times D}$: Input data matrix.
- $\mathbf{Y} \in \mathbb{R}^{N \times C}$: Observed data (output).
- $\mathbf{z} \in \{1, \ldots, K\}^N$: Sequence of hidden states.

2 Model Components

2.1 Transition Model

The transitions between hidden states are represented by a transition probability matrix:

$$\mathbf{A} = a_{ij} \in \mathbb{R}^{K \times K}, \quad a_{ij} = P(z_t = j \mid z_{t-1} = i),$$

where:

$$a_{ij} \geq 0, \quad \sum_{j=1}^{K} a_{ij} = 1.$$

2.2 Emission Model

The emission model generates observations $\mathbf{Y}$ conditioned on the hidden state z_t and the input $\mathbf{x}_t$:

$$P(\mathbf{y}_t \mid \mathbf{x}_t, z_t = k) \sim \mathcal{N}(\boldsymbol{\mu}_k(\mathbf{x}_t), \boldsymbol{\Sigma}_k),$$

where:

- $\boldsymbol{\mu}_k(\mathbf{x}_t)$: Mean of the Gaussian distribution for state k, defined as:

$$\boldsymbol{\mu}_k(\mathbf{x}_t) = \tanh(\mathbf{x}_t^\top \mathbf{w}_k),$$

  where:

  - $\mathbf{x}_t \in \mathbb{R}^D$: Input at time t.
  - $\mathbf{w}_k \in \mathbb{R}^{D \times C}$: Weight matrix for state k.

- $\boldsymbol{\Sigma}_k$: Covariance matrix for state k, typically initialized as a small identity matrix:

$$\boldsymbol{\Sigma}_k = \epsilon \cdot I_C.$$

3 Expectation-Maximization Algorithm

The EM algorithm alternates between two steps:

3.1 E-Step (Expectation)

Using forward-backward algorithm, compute the forward path probability and backward path probability:

- **Forward pass:**

$$\alpha_t(k) = P(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = k \mid \mathbf{X}, \text{current parameters}).$$

  - **Initialization:**

$$\alpha_1(j) = \pi_j P(\mathbf{y}_1 \mid \mathbf{x}_1, z_1 = j), 1 \leq j \leq K$$

  - **Recursion:**

$$\alpha_t(j) = \sum_{i=1}^{K} \alpha_{t-1} a_{ij} P(\mathbf{y}_t \mid \mathbf{x}_t, z_t = j), 1 \leq j \leq K, 1 < t \leq N$$

- **Backward pass:**

$$\beta_t(k) = P(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_N \mid z_t = k, \mathbf{X}, \text{current parameters}).$$

  - **Initialization:**

$$\beta_N(i) = 1, 1 \leq i \leq K$$

  - **Recursion:**

$$\beta_t(i) = \sum_{j=1}^{K} a_{ij} P(\mathbf{y}_{t+1} \mid \mathbf{x}_{t+1}, z_{t+1} = j) \beta_{t+1}(j), 1 \leq i \leq K, 1 \leq t < N$$

Compute the posterior probabilities of the hidden states γ and the joint posterior probabilities for successive states ξ, given α, β and the current parameters:

$$\gamma_t(k) = P(z_t = k \mid \mathbf{Y}, \mathbf{X}, \text{current parameters}) = \frac{\alpha_t(k)\beta_t(k)}{\sum_{j=1}^{N} \alpha_t(k)\beta_t(k)},$$

$$\xi_t(i,j) = P(z_t = i, z_{t+1} = j \mid \mathbf{Y}, \mathbf{X}, \text{current parameters}) = \frac{\alpha_t(j) a_{ij} P(\mathbf{y}_{t+1} \mid \mathbf{x}_{t+1}, z_{t+1} = j)\beta_{t+1}(j))}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

3.2 M-Step

Update the model parameters to maximize the expected complete-data log-likelihood.

3.2.1 Transition Matrix Update

$$a_{ij} = \frac{\sum_{t=1}^{N-1} \xi_t(i,j)}{\sum_{t=1}^{N-1} \sum_{j=1}^{K} \xi_t(i,j)}.$$

3.2.2 Emission Parameters Update (Weights and Covariances)

For each state k, update the weights $\mathbf{w}_k$ by minimizing the negative log-likelihood:

$$\mathcal{L}_k = -\sum_{t=1}^{N} \gamma_t(k) \log P(\mathbf{y}_t \mid \mathbf{x}_t, z_t = k).$$

In other words,

$$\mathbf{w}_k \leftarrow \arg\min_{\mathbf{w}_k} -\sum_{t=1}^{N} \gamma_t(k) \log P(\mathbf{y}_t \mid \mathbf{x}_t, z_t = k).$$

Then the covariance matrix is updated as:

$$\mathbf{\Sigma}_k = \frac{1}{\sum_t \gamma_t(k)} \sum_t \gamma_t(k)(\mathbf{y}_t - \boldsymbol{\mu}_k(\mathbf{x}_t))(\mathbf{y}_t - \boldsymbol{\mu}_k(\mathbf{x}_t))^\top.$$

4 Viterbi Decoding

The most likely sequence of states $\mathbf{z}$ is computed using the Viterbi algorithm:

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} P(\mathbf{z} \mid \mathbf{Y}, \mathbf{X}, \text{current parameters}).$$

- **Initialization:**

$$v_1(k) = P(z_1 = k) P(\mathbf{y}_1 \mid \mathbf{x}_1, z_1 = k).$$

- **Recursion:**

$$v_t(k) = \max_{j=1,2,\ldots,K} v_{t-1}(j) a_{jk} P(\mathbf{y}_t \mid \mathbf{x}_t, z_t = k).$$

- **Backtracking:** Retrieve the most likely sequence from δ_t.

5 Data Generation

Synthetic data can be generated by:

1. Sampling the initial state:
$$z_1 \sim P(z_1).$$
2. Iteratively sampling:
   - The next state:
   $$z_t \sim P(z_t \mid z_{t-1}).$$
   - The observation:
   $$\mathbf{y}_t \sim \mathcal{N}(\boldsymbol{\mu}_{z_t}(\mathbf{x}_t), \boldsymbol{\Sigma}_{z_t}).$$