Mathematical Formulation of the Input-Driven GLMHMM

1 Model Formulation

1.1 Input-Driven Transition Model

The probability of transitioning from state j to state k at time t, given the external input u_t, is modeled as:

$$P(z_t = k \mid z_{t-1} = j, u_t) = \frac{\exp\{\log P_{j,k} + \boldsymbol{w}_k^\top u_t\}}{\sum_{k'=1}^{K} \exp\{\log P_{j,k'} + \boldsymbol{w}_{k'}^\top u_t\}}. \tag{1}$$

where: - $P_{j,k}$ is the base transition probability. - $\boldsymbol{w}_k$ (or W_{in}) is the weight vector associated with state k. - u_t is the external input vector. - The denominator ensures the probabilities sum to 1.

1.2 Emission Model

The observed data y_t is modeled using a GLM-based Gaussian distribution:

$$P(y_t|X_t, z_t = k) \sim \mathcal{N}(\mu_k(X_t), \Sigma_k), \tag{2}$$

where: - The mean is given by:

$$\mu_k(X_t) = \frac{\tanh(X_t^\top W_k) + 1}{2}. \tag{3}$$

- The covariance matrix for state k:

$$\Sigma_k = \epsilon I_C. \tag{4}$$

2 Forward-Backward Algorithm (E-Step)

The forward-backward algorithm computes the posterior probabilities of the hidden states.

2.1 Forward Pass (Computing $\alpha_t(k)$)

The forward probabilities represent the probability of observing the sequence up to time t and ending in state k:

$$\alpha_t(k) = P(y_1, \ldots, y_t, z_t = k | X, U, \text{parameters}). \tag{5}$$

Initialization:

$$\alpha_1(k) = \pi_k P(y_1 | X_1, z_1 = k), \quad 1 \leq k \leq K. \tag{6}$$

Recursion:

$$\alpha_t(k) = \sum_{j=1}^{K} \alpha_{t-1}(j) P(z_t = k | z_{t-1} = j, u_t) P(y_t | X_t, z_t = k). \tag{7}$$

2.2 Backward Pass (Computing $\beta_t(k)$)

The backward probabilities represent the probability of observing future data given state k:

$$\beta_t(k) = P(y_{t+1}, \ldots, y_N | z_t = k, X, U, \text{parameters}). \tag{8}$$

Initialization:

$$\beta_N(k) = 1, \quad 1 \leq k \leq K. \tag{9}$$

Recursion:

$$\beta_t(j) = \sum_{k=1}^{K} P(z_{t+1} = k | z_t = j, u_t) P(y_{t+1} | X_{t+1}, z_{t+1} = k) \beta_{t+1}(k). \tag{10}$$

3 M-Step: Parameter Updates

3.1 Updating the Transition Model W_{in}

To optimize W_{in}, we maximize the transition log-likelihood:

$$\mathcal{L}_{\text{transition}} = \sum_{t=1}^{N-1} \sum_{j=1}^{K} \sum_{k=1}^{K} \xi_t(j,k) \log P(z_t = k \mid z_{t-1} = j, u_t). \tag{11}$$

Taking the gradient:

$$\frac{\partial \mathcal{L}_{\text{transition}}}{\partial \boldsymbol{w}_k} = \sum_{t=1}^{N-1} \sum_{j=1}^{K} \xi_t(j,k) \left[u_t - \sum_{k'} P(z_t = k' \mid z_{t-1} = j, u_t) u_t \right]. \tag{12}$$

Update rule:

$$\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k + \eta \sum_{t=1}^{N-1} \sum_{j=1}^{K} \xi_t(j,k) \left[u_t - \sum_{k'} P(z_t = k' \mid z_{t-1} = j, u_t) u_t \right]. \tag{13}$$

3.2 Updating the Base Transition Matrix

$$P_{j,k} = \frac{\sum_{t=1}^{N-1} \xi_t(j,k)}{\sum_{t=1}^{N-1} \sum_{k'=1}^{K} \xi_t(j,k')}. \tag{14}$$

3.3 Updating Emission Parameters

$$W_k \leftarrow \arg\min_{W_k} - \sum_t \gamma_t(k) \log P(y_t | X_t, z_t = k). \tag{15}$$

4 Viterbi Decoding

The most likely sequence of states z is found using:

$$\hat{z}_T = \arg\max_k \delta_T(k). \tag{16}$$

with recursion:

$$\delta_t(k) = \max_j \left[\delta_{t-1}(j) + \log P(z_t = k | z_{t-1} = j, u_t)\right] + \log P(y_t | X_t, z_t = k). \tag{17}$$

Backtracking:

$$\hat{z}_t = \psi_{t+1}(\hat{z}_{t+1}). \tag{18}$$

5 Data Generation

Autoregressive:

$$X_t^{(d)} = \sum_{i=0}^{p} a_i^{(d)} X_{t-i}^{(d)} + \epsilon \sim \mathcal{N}(0, 0.1) \tag{19}$$

where $\sum_{i=0}^{p} a_i^{(d)} = 1$.
d: feature of input X
p: the number of previously time points that determine X_t.
ϵ: Noise term that follows normal distribution

Sinusoidal:

$$X_t^{(d)} = X_1^{(d)} + a_{i1} \sin(\frac{1}{200}t) + a_{i2} \sin(\frac{1}{400}t) + a_{i3} \sin(\frac{1}{800}) \tag{20}$$

where $\sum_{j=1}^{3} a_{ij} = 1$
d: feature of input X
t: the current time point

$$z_t \sim z_t \sim P(z_t|z_{t-1}, u_t). \tag{21}$$

$$y_t \sim \mathcal{N}(\mu_{z_t}(X_t), \Sigma_{z_t}). \tag{22}$$