Powering Hidden Markov Model by Generative Models

Firsthand Scientists

March 5, 2019

1 Notation

Time is indexed by subscript and sequence is denoted by underline. $\boldsymbol{x}_t$ is signal at time t. The sequential time is denoted by $\underline{\boldsymbol{x}} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T]^\intercal$, where $[\cdot]^\intercal$ means transpose and T is the length of the sequence. Sequential signal or clip uses underline notation and is indexed by superscript, for instance $\underline{\boldsymbol{x}}^{(r)}$ means the r-th sequential signal, where $r = 1, 2, \cdots, R.$, and $\underline{\boldsymbol{x}}^{(r)} = \left[\boldsymbol{x}_1^{(r)}, \boldsymbol{x}_2^{(r)}, \cdots, \boldsymbol{x}_{T^{(r)}}^{(r)}\right]$ with length $T^{(r)}$. Note different sequential signal $\underline{\boldsymbol{x}}^{(r)}$ could have different lengths.

The hypothesis of Hidden Markov Model (HMM): $\mathcal{H} := \{\boldsymbol{H} | \{\mathcal{S}, \boldsymbol{q}, A, p(\boldsymbol{x}|s; \boldsymbol{\Phi})\}$,

- $\mathcal{S}$ is the set of states of HMM $\boldsymbol{H}$;
- $\boldsymbol{q} = \left[q_1, q_2, \cdots, q_{|\mathcal{S}|}\right]^\intercal$ initial distribution of HMM $\boldsymbol{H}$ with $|\mathcal{S}|$ is cardinality of $\mathcal{S}$, $q_k = p(s = k)$ for random state variable s.
- A is the transition matrix for the HMM $\boldsymbol{H}$ of size $|\mathcal{S}| \times |\mathcal{S}|$.
- Observable signal density $p(\boldsymbol{x}|s; \boldsymbol{\Phi})$ given hidden state sequence, where $\boldsymbol{\Phi}$ is the parameter set that defines this conditional probabilistic model.

2 Problem Statement

Given a empirical distribution $\hat{p}(\underline{\boldsymbol{x}}) = \frac{1}{R}\sum_{r=1}^{R} \delta_{\underline{\boldsymbol{x}}^{(r)}}(\underline{\boldsymbol{x}})$. We want to find a probabilistic model such that:

$$\min KL(\hat{p}(\underline{\boldsymbol{x}}) \| p(\underline{\boldsymbol{x}})) \tag{1}$$

where $KL(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence.

When we use HMM to model the empirical distribution and approach the unknown true distribution, the problem boils down to:

$$\underset{\boldsymbol{H}\in\mathcal{H}}{\operatorname{argmax}}\, p(\underline{\boldsymbol{X}}; \boldsymbol{H}) \tag{2}$$

where $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}}^{(1)}, \underline{\boldsymbol{x}}^{(2)}, \cdots, \underline{\boldsymbol{x}}^{(R)}\right]$

The problem can be reformulated as

$$\underset{\boldsymbol{H}\in\mathcal{H}}{\operatorname{argmax}} \sum_{r=1}^{R} \log\, p(\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}) \tag{3}$$

for independent identical distributed assumption of $\underline{\boldsymbol{x}}$.

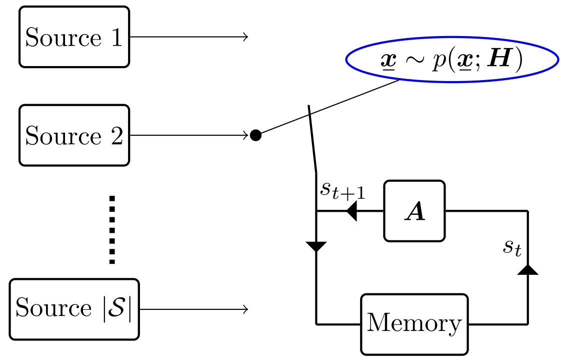

Figure 1: HMM Model defined by $\boldsymbol{H} = \{\mathcal{S}, \boldsymbol{q}, \boldsymbol{A}, p(\boldsymbol{x}|\boldsymbol{s}; \boldsymbol{\Phi})\}$

3 Proposal

Since model $\boldsymbol{H}$ contains hidden sequential variable $\underline{\boldsymbol{s}}$, we can not directly solve the maximum likelihood problem in Equation 3. We use expectation maximization (EM) to address the hidden variable problem by

- E-step: The "expected likelihood" function:

$$\mathcal{Q}(\boldsymbol{H}; \boldsymbol{H}^{\text{old}}) = \mathbb{E}_{p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})} \left[\sum_{r=1}^{R} \log\, p(\underline{\boldsymbol{x}}^{(r)}, \underline{\boldsymbol{s}}^{(r)}; \boldsymbol{H}) \right] \tag{4}$$

- M-step: the optimization step:

$$\max_{\boldsymbol{H}}\, \mathcal{Q}(\boldsymbol{H}; \boldsymbol{H}^{\text{old}}) \tag{5}$$

The Equation 5 can be reformulated as:

$$\max_{\boldsymbol{H}}\, \mathcal{Q}(\boldsymbol{H}; \boldsymbol{H}^{\text{old}}) = \max_{\boldsymbol{q}}\, \mathcal{Q}(\boldsymbol{q}; \boldsymbol{H}^{\text{old}}) + \max_{A}\, \mathcal{Q}(A; \boldsymbol{H}^{\text{old}}) + \max_{\boldsymbol{\Phi}}\, \mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}}) \tag{6}$$

where

$$\mathcal{Q}(\boldsymbol{q}; \boldsymbol{H}^{\text{old}}) = \sum_{r=1}^{R} \mathbb{E}_{p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})} \left[\log\, p(s_1^{(r)}; \boldsymbol{q}) \right] \tag{7}$$

$$\mathcal{Q}(\boldsymbol{A}; \boldsymbol{H}^{\text{old}}) = \sum_{r=1}^{R} \mathbb{E}_{p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})} \left[\log \sum_{t=1}^{T^{(r)}-1} p(s_{t+1}^{(r)}|s_t^{(r)}; A) \right] \tag{8}$$

$$\mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}}) = \sum_{r=1}^{R} \mathbb{E}_{p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})} \left[\log\, p(\underline{\boldsymbol{x}}^{(r)}|\underline{\boldsymbol{s}}^{(r)}; \boldsymbol{\Phi}) \right] \tag{9}$$

We can see that the solution of H depends on the posterior probability $p(\underline{\boldsymbol{s}}|\underline{\boldsymbol{x}}; \boldsymbol{H})$. Though the evaluation of posterior according to Bayesian theorem is simple, the computation complexity of $p(\underline{\boldsymbol{s}}|\underline{\boldsymbol{x}}; \boldsymbol{H})$ grows exponentially with the length of $\underline{\boldsymbol{s}}$. Therefore, we would employ Forward/Backward algorithm [] to do the posterior computation efficiently. The marginal $p(s_t|\underline{\boldsymbol{x}}; \boldsymbol{H})$ is also efficiently computed as the joint posterior.

We summarize the optimization algorithm as:

Algorithm 1 Meta algorithm for HMM powered by Generative Models

Input: Building $\boldsymbol{H}^{\text{old}}, \boldsymbol{H} \in \mathcal{H}$ gives:
$\boldsymbol{H}^{\text{old}} = \{\mathcal{S}, \boldsymbol{q}^{\text{old}}, A^{\text{old}}, p(\boldsymbol{x}|s; \boldsymbol{\Phi}^{\text{old}})\}$, $\boldsymbol{H} = \{\mathcal{S}, \boldsymbol{q}, A, p(\boldsymbol{x}|s; \boldsymbol{\Phi})\}$,
Initialize $\boldsymbol{H}$
$\boldsymbol{H}^{\text{old}} \leftarrow \boldsymbol{H}$
for $\boldsymbol{H}$ not converge **do**
  Sample a batch of data $\left\{\underline{\boldsymbol{x}}^{(r)}\right\}_{r=1}^{R_b}$ from the dataset $\hat{p}(\underline{\boldsymbol{x}})$
  Compute $p(s_t^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})$, $p(s_t^{(r)}, s_{t+1}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})$ by forward/backward algorithm;
  $\boldsymbol{q} \leftarrow \underset{\boldsymbol{q}}{\text{argmin}}\ \mathcal{Q}(\boldsymbol{q}; \boldsymbol{H}^{\text{old}})$ by Equation 13;
  $\boldsymbol{A} \leftarrow \underset{\boldsymbol{A}}{\text{argmin}}\ \mathcal{Q}(\boldsymbol{A}; \boldsymbol{H}^{\text{old}})$ by Equation 15;
  $\boldsymbol{\Phi} \leftarrow \underset{\boldsymbol{\Phi}}{\text{argmin}}\ \mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}})$ by calling Algorithm 2 or 3;
  $\boldsymbol{H}^{\text{old}} \leftarrow \boldsymbol{H}$
end for

3.1 Initial Probability Update

Equation 7 can be written as:

$$
\begin{aligned}
\mathcal{Q}(\boldsymbol{q}; \boldsymbol{H}^{\text{old}}) &= \sum_{r=1}^{R} \sum_{\underline{\boldsymbol{s}}^{(r)}} p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \log p(s_1^{(r)}; \boldsymbol{q}) \\
&= \sum_{r=1}^{R} \sum_{s_1^{(r)}=1}^{|\mathcal{S}|} \sum_{s_2^{(r)}=1}^{|\mathcal{S}|} \cdots \sum_{s_{T^r}^{(r)}}^{|\mathcal{S}|} p(s_1^{(r)}, s_2^{(r)}, \cdots, s_{T^r}^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \log p(s_1^{(r)}; \boldsymbol{q}) && (10) \\
&= \sum_{r=1}^{R} \sum_{s_1^{(r)}=1}^{|\mathcal{S}|} p(s_1^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \log p(s_1^{(r)}; \boldsymbol{q}) && (11)
\end{aligned}
$$

Since $p(s_1^{(r)}; \boldsymbol{H})$ is the probability of initial state of HMM $\boldsymbol{H}$ for r-th sequential, actually $q_i = p(s_1^{(r)} = i; \boldsymbol{H})$ for $i = 1, 2, \cdots, |\mathcal{S}|$. Solution to problem:

$$
\begin{aligned}
\boldsymbol{q}^{\text{new}} &= \underset{\boldsymbol{q}}{\text{argmax}}\ \mathcal{Q}(\boldsymbol{q}; \boldsymbol{H}^{\text{old}}), \\
&\text{s.t.} \sum_{i=1}^{|\mathcal{S}|} q_i = 1 \\
&q_i \geqslant 0, \forall s. && (12)
\end{aligned}
$$

is

$$
q_i = \frac{1}{R} \sum_{r=1}^{R} p(s_1^{(r)} = i|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}), \forall\, i = 1, 2, \cdots, |\mathcal{S}|. \tag{13}
$$

3.2 Transition Probability Update

Equation 8 can be written as

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{A};\boldsymbol{H}^{\text{old}}) &= \sum_{r=1}^{R}\mathbb{E}_{p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})}\left[\log\sum_{t=1}^{T^{(r)}-1}p(s_{t+1}^{(r)}|s_t^{(r)};A)\right] \\
&= \sum_{r=1}^{R}\sum_{\underline{\boldsymbol{s}}^{(r)}}p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})\log\sum_{t=1}^{T^{(r)}-1}p(s_{t+1}^{(r)}|s_t^{(r)};A) \\
&= \sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1}\sum_{s_t^{(r)}=1}^{|\mathcal{S}|}\sum_{s_{t+1}^{(r)}=1}^{|\mathcal{S}|}p(s_t^{(r)},s_{t+1}^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})\log p(s_{t+1}^{(r)}|s_t^{(r)};A)
\end{aligned}\tag{14}$$

Since $\boldsymbol{A}_{i,j}=p(s_{t+1}^{(r)}=j|s_t^{(r)}=i;A)$ where $A_{i,j}$ is the element of transition matrix A, the solution to problem:

$$\begin{aligned}
\boldsymbol{A}^{\text{new}} =& \operatorname*{argmax}_{\boldsymbol{A}}\ \mathcal{Q}(\boldsymbol{A};\boldsymbol{H}^{\text{old}}), \\
\text{s.t.}\ \ & \boldsymbol{A}\cdot\mathbf{1}=\mathbf{1} \\
& \boldsymbol{A}^{\intercal}\cdot\mathbf{1}=\mathbf{1} \\
& \boldsymbol{A}_{i,j}\geqslant 0.
\end{aligned}\tag{15}$$

is

$$\boldsymbol{A}_{i,j}^{\text{new}}=\frac{\bar{\xi}_{i,j}}{\sum_{k=1}^{|\mathcal{S}|}\bar{\xi}_{i,k}},\tag{16}$$

where

$$\bar{\xi}_{i,j}=\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1}p(s_t^{(r)}=i,s_{t+1}^{(r)}=j|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})\tag{17}$$

3.3 Generative Model Update

Equation 9 can be rewritten as

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\Phi};\boldsymbol{H}^{\text{old}}) &= \sum_{r=1}^{R}\sum_{\underline{\boldsymbol{s}}^{(r)}}p(\underline{\boldsymbol{s}}^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})\log p(\underline{\boldsymbol{x}}^{(r)}|\underline{\boldsymbol{s}}^{(r)};\boldsymbol{\Phi}) \\
&= \sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1}\sum_{s_t^{(r)}=1}^{|\mathcal{S}|}p(s_t^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})\log p(\boldsymbol{x}_t^{(r)}|s_t^{(r)};\boldsymbol{\Phi}).
\end{aligned}\tag{18}$$

Then the third subproblem of Equation 6 becomes:

$$\begin{aligned}
& \operatorname*{argmax}_{\boldsymbol{\Phi}}\ \mathcal{Q}(\boldsymbol{\Phi};\boldsymbol{H}^{\text{old}}), \\
\text{s.t.}\ \ & p(\boldsymbol{x}|s;\boldsymbol{\Phi})\ \text{is our general model}
\end{aligned}\tag{19}$$

It could be seen from Equation 18 that the key to update generate model is to evaluate $p(\boldsymbol{x}|s;\boldsymbol{\Phi})$ for all $s\in\mathcal{S}$. In Forward/Backward algorithm, evaluation of $p(\boldsymbol{x}|s;\boldsymbol{\Phi})$ is also all what is needed to compute $p(s|\boldsymbol{x};\boldsymbol{\Phi})$. In the following two subsections, we will provide two neural network based generative models that fulfill this requirement and also have high capability for complex signal modeling.

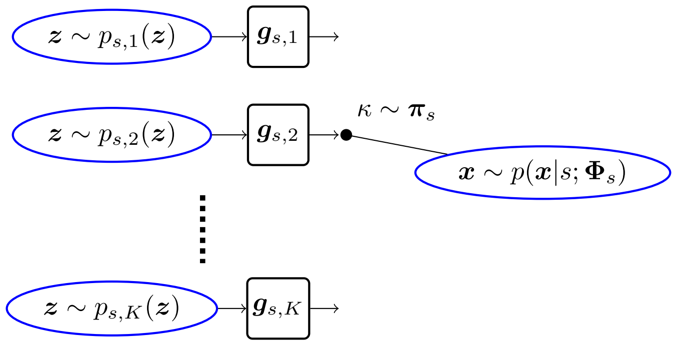

Figure 2: Source s powered by generator mixed generative model (GenHMM).

3.3.1 Generator Mixed HMM (GenHMM)

Base on the generative model assumption:

$$p(\boldsymbol{x}|s;\boldsymbol{\Phi}_s) = \sum_{\kappa=1}^{K} \pi_{s,\kappa} p(\boldsymbol{x}|s,\kappa;\boldsymbol{\Phi}_{s,\kappa}) \tag{20}$$

where $\boldsymbol{\Phi}_s = \{\boldsymbol{\Phi}_{s,\kappa}|\kappa = 1, 2, \cdots, K.\}$, and

$$\sum_{\kappa=1}^{K} \pi_{s,\kappa} = 1 \tag{21}$$

The help function should be revised to deal with the new latent variable κ into:

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\Phi};\boldsymbol{H}^{\text{old}}) =& \sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1}\sum_{s_t^{(r)}=1}^{|\mathcal{S}|}\sum_{\kappa_t^{(r)}=1}^{K} p(s_t^{(r)},\kappa_t^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}}) \log p(\kappa_t^{(r)},\boldsymbol{x}_t^{(r)}|s_t^{(r)};\boldsymbol{\Phi}) \\
=& \sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1}\sum_{s_t^{(r)}=1}^{|\mathcal{S}|}\sum_{\kappa_t^{(r)}=1}^{K} p(s_t^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}}) p(\kappa_t^{(r)}|s_t^{(r)},\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}}) \Big[\log \pi_{s_t^{(r)},\kappa_t^{(r)}} \\
&+ \log p(\boldsymbol{x}_t^{(r)}|s_t^{(r)},\kappa_t^{(r)};\boldsymbol{\Phi})\Big]
\end{aligned} \tag{22}$$

In Equation 22, $p(s_t|\underline{\boldsymbol{x}},\boldsymbol{H}^{\text{old}})$ is computed by forward/backward algorithm. The posterior of κ is:

$$\begin{aligned}
p(\kappa|s,\underline{\boldsymbol{x}};\boldsymbol{H}^{\text{old}}) &= \frac{p(\kappa,\underline{\boldsymbol{x}}|s;\boldsymbol{H}^{\text{old}})}{p(\underline{\boldsymbol{x}}|s,\boldsymbol{H}^{\text{old}})} \\
&= \frac{\pi_{s,\kappa} p(\underline{\boldsymbol{x}}|s,\kappa,\boldsymbol{H}^{\text{old}})}{\sum_{\kappa=1}^{K}\pi_{s,\kappa} p(\underline{\boldsymbol{x}}|s,\kappa,\boldsymbol{H}^{\text{old}})} \\
&= \frac{\pi_{s,\kappa} p(\boldsymbol{x}|s,\kappa,\boldsymbol{H}^{\text{old}})}{\sum_{\kappa=1}^{K}\pi_{s,\kappa} p(\boldsymbol{x}|s,\kappa,\boldsymbol{H}^{\text{old}})}
\end{aligned} \tag{23}$$

where the last equation is due to the fact that only $\boldsymbol{x}_t$ among sequence $\underline{\boldsymbol{x}}$ depends on s_t, κ_t.

The latent prior for mixture of each source s is obtained by solving the following problem:

$$\pi_{s,\kappa} = \underset{pi_{s,\kappa}}{\text{argmax}}\ \mathcal{Q}(\boldsymbol{\Phi};\boldsymbol{H}^{\text{old}}) \tag{24}$$

$$s.t. \sum_{\kappa=1}^{K} \pi_{s,\kappa} = 1$$

which gives the solution:

$$\pi_{s,\kappa} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1} p(s_t^{(r)} = s, \kappa_t^{(r)} = \kappa | \underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})}{\sum_{k=1}^{K}\sum_{r=1}^{R}\sum_{t=1}^{T^{(r)}-1} p(s_t^{(r)} = s, \kappa_t^{(r)} = k | \underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})} \tag{25}$$

where $p(s, \kappa|\underline{\boldsymbol{x}}; \boldsymbol{H}^{\text{old}}) = p(s|\underline{\boldsymbol{x}}; \boldsymbol{H}^{\text{old}})p(\kappa|s, \underline{\boldsymbol{x}}; \boldsymbol{H}^{\text{old}})$ that can be computed by results of forward/backward and Equation 23.

In implementation, GenHMM uses the generator mixed emission model with κ-th component as:

$$\begin{aligned} & p(\boldsymbol{x}|s, \kappa; \boldsymbol{\Phi}_{s,\kappa}) \\ =& p_{s,\kappa}(\boldsymbol{z}) \left| \det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}\right) \right| \\ =& \mathcal{N}\left(\boldsymbol{f}_{s,\kappa}(\boldsymbol{x}); \boldsymbol{\mu}_{s,\kappa}, \boldsymbol{C}_{s,\kappa}\right) \left| \det\left(\frac{\partial \boldsymbol{f}_{s,\kappa}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \right| \end{aligned} \tag{26}$$

where $\boldsymbol{f}_{s,\kappa} = \boldsymbol{g}_{\boldsymbol{s},\boldsymbol{\kappa}}^{-\boldsymbol{1}}$ is defined by parameter set $\boldsymbol{\theta}_{s,\kappa}$, and $\boldsymbol{\Phi}_{s,\kappa} = \{\boldsymbol{\mu}_{s,\kappa}, \boldsymbol{C}_{s,\kappa}, \boldsymbol{\theta}_{s,\kappa}\}, \kappa = 1, 2, \cdots, K$. We can start by setting $\boldsymbol{\mu}_{s,\kappa} = \boldsymbol{0}$ and $\boldsymbol{C}_{s,\kappa} = diag(\boldsymbol{1})$.

Algorithm 2 M-step w.r.t. $\boldsymbol{\Phi}$ powered by GenMM

Input: Latent mixture distribution: $\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \text{diag}(\boldsymbol{1})), \forall\, s \in \mathcal{S}, \kappa = 1, 2, \cdots, K$;
Empirical distribution $\hat{p}(\boldsymbol{x})$ of dataset;
Set a total number of epochs T of training as stop criterion. A learning rate η.
for epoch $t < T$ **do**
  Sample a batch of data $\left\{\underline{\boldsymbol{x}}^{(r)}\right\}_{r=1}^{R_b}$ from dataset $P_d(\underline{\boldsymbol{x}})$
  Compute $p(s_t^{(r)}, \kappa_t^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}})$ by Forward/backward and Equation 23.
  Compute loss $\mathcal{Q}\left(\boldsymbol{\Phi}, \boldsymbol{H}^{\text{old}}\right)$ in Equation 22
  $\partial\boldsymbol{\theta}_s \leftarrow \nabla_{\boldsymbol{\theta}} - \frac{1}{R_b}\mathcal{Q}\left(\boldsymbol{\Phi}, \boldsymbol{H}^{\text{old}}\right)$
  $\boldsymbol{\theta}_s \leftarrow \boldsymbol{\theta}_s - \eta \cdot \partial\boldsymbol{\theta}_s, \forall s \in \mathcal{S}$
end for
Update $\pi_{s,\kappa}, \forall, s \in \mathcal{S}, \kappa = 1, 2, \cdots, K$, according to Equation 25
Assemble $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_s | s \in \mathcal{S}\}$ for $\boldsymbol{\Phi}_s = \{\boldsymbol{\theta}_{s,\kappa}, \boldsymbol{\mu}_{s,\kappa}, \boldsymbol{C}_{s,\kappa} | \kappa = 1, 2, \cdots, K\}$.

3.3.2 Latent-source Mixed HMM (LatMM)

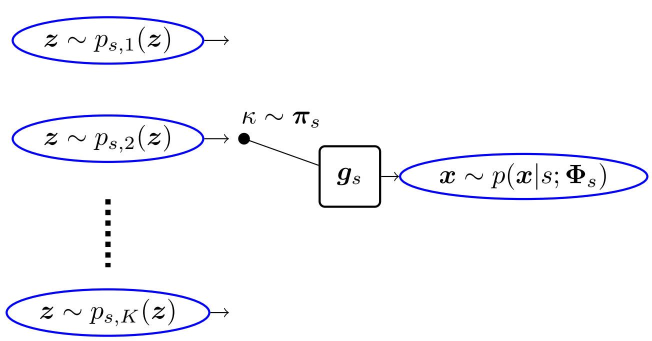

Figure 3: Source s powered by Latent mixed generative model (LatHMM).

LatHMM, (**benching marking**), the latent source mixed emission model is built by allowing different latent source sharing the same generator $\boldsymbol{g}_s$:

$$\begin{aligned}
&p(\boldsymbol{x}|s,\kappa;\boldsymbol{\Phi}_s)\\
=&p_{s,\kappa}(\boldsymbol{z})\left|\det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}\right)\right|\\
=&\mathcal{N}\left(\boldsymbol{f}_s(\boldsymbol{x});\boldsymbol{\mu}_{s,\kappa},\boldsymbol{C}_{s,\kappa}\right)\left|\det\left(\frac{\partial \boldsymbol{f}_s(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|
\end{aligned}\tag{27}$$

where $\boldsymbol{f}_s = \boldsymbol{g}_s^{\boldsymbol{-1}}$ is defined by $\boldsymbol{\theta}$, and $\boldsymbol{\Phi}_s = \{\boldsymbol{\theta}_s, \boldsymbol{\mu}_{s,\kappa}, \boldsymbol{C}_{s,\kappa}|\kappa = 1, 2, \cdots, K\}$.

This emission generator sharing makes the posterior computation w.r.t. of κ easier:

$$p(\kappa|s,\underline{\boldsymbol{x}};\boldsymbol{H}^{\text{old}}) = \frac{\pi_{s,\kappa}p(\boldsymbol{x}|s,\kappa,\boldsymbol{H}^{\text{old}})}{\sum_{\kappa=1}^{K}\pi_{s,\kappa}p(\boldsymbol{x}|s,\kappa,\boldsymbol{H}^{\text{old}})} = \left.\frac{\pi_{s,\kappa}\mathcal{N}\left(\boldsymbol{z};\boldsymbol{\mu}_{s,\kappa},\boldsymbol{C}_{s,\kappa}\right)}{\sum_{\kappa=1}^{K}\pi_{s,\kappa}\mathcal{N}\left(\boldsymbol{z};\boldsymbol{\mu}_{s,\kappa},\boldsymbol{C}_{s,\kappa}\right)}\right|_{\boldsymbol{z}=\boldsymbol{f}_s(\boldsymbol{x})}. \tag{28}$$

Computation of $p(s_t|\underline{\boldsymbol{x}},\boldsymbol{H}^{\text{old}})$ remains the same, relying on forward/backward algorithm.

In implementation, $\boldsymbol{C}_{s,\kappa} = diag\left(\boldsymbol{\sigma}_{s,\kappa}\right)$ for simplicity. To avoid the singularity problem of Gaussian mixture, we put a Gamma distribution as prior for $\boldsymbol{\sigma}_{s,\kappa}$, i.e. $\Gamma(\boldsymbol{\sigma}_{s,\kappa}^{-1}, a, b)$ where a and b are hyperparameter for Gamma distribution. Then the problem can be reformulated as:

$$\underset{\boldsymbol{\Phi}}{\text{argmax}}\ \mathcal{Q}(\boldsymbol{\Phi};\boldsymbol{H}^{\text{old}}) + \frac{1}{K}\log\prod_{k=1}^{K}\Gamma(\boldsymbol{\sigma}_{s,k}^{-1};a,b) \tag{29}$$

Apart from the emission prboability model difference, rest computation is the same as subsubsection 3.3.1. We summarize the algorithm as:

Algorithm 3 M-step w.r.t. $\boldsymbol{\Phi}$ powered by LatMM

1: **Input:** Latent mixture distribution: $\sum_{k=1}^{K}\pi_{s,\kappa}\mathcal{N}\left(\boldsymbol{z};\boldsymbol{\mu}_{s,\kappa},\text{diag}(\boldsymbol{\sigma}_{s,\kappa}^2)\right)$
Empirical distribution $\hat{p}(\boldsymbol{x})$ of dataset;
2: Set a total number of epochs T of training as stop criterion. A learning rate η. Set hyperparameter a , b for prior of $\boldsymbol{\sigma}_{s,\kappa}^{-1}, \forall k$.
3: **for** epoch $t < T$ **do**
4: Sample a batch of data $\left\{\underline{\boldsymbol{x}}^{(r)}\right\}_{r=1}^{R_b}$ from dataset $P_d(\underline{\boldsymbol{x}})$
5: Compute $p(s_t^{(r)}, \kappa_t^{(r)}|\underline{\boldsymbol{x}}^{(r)};\boldsymbol{H}^{\text{old}})$ by Forward/backward and Equation 28.
6: Compute loss in Equation 29
7: $\partial\boldsymbol{\theta}_s, \partial\boldsymbol{\mu}_{s,\kappa}, \partial\boldsymbol{\sigma}_{s,\kappa} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\mu}_k,\boldsymbol{\sigma}_k} - \frac{1}{R_b}\mathcal{Q}\left(\boldsymbol{\Phi}, \underline{\boldsymbol{H}}^{\text{old}}\right) - \frac{1}{K}\sum_{k=1}^{K}\log\Gamma(\boldsymbol{\sigma}_{s,\kappa}^{-1};a,b)$
8: $\boldsymbol{\theta}_s \leftarrow \boldsymbol{\theta}_s - \eta\cdot\partial\boldsymbol{\theta}_s, \forall s \in \mathcal{S}$
9: $\boldsymbol{\mu}_{s,\kappa} \leftarrow \boldsymbol{\mu}_{s,\kappa} - \eta\cdot\partial\boldsymbol{\mu}_{s,\kappa}, \forall\kappa, s$
10: $\boldsymbol{\sigma}_{s,\kappa} \leftarrow \boldsymbol{\sigma}_{s,\kappa} - \eta\cdot\partial\boldsymbol{\sigma}_{s,\kappa}, \forall\kappa, s$
11: **end for**
12: Update π_k according to Equation 25
13: Assemble $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_s|s \in \mathcal{S}\}$ for $\boldsymbol{\Phi}_s = \{\boldsymbol{\theta}_s, \boldsymbol{\mu}_{s,\kappa}, \boldsymbol{C}_{s,\kappa}|\kappa = 1, 2, \cdots, K\}$.

3.3.3 Generator-shared HMM (GSHMM)...to be continued

to be continued...

Alternatively, we can use a latent-source mixed HMM (LatM-HMM) where different latent source share the same generator functioning as feature mapping. Then the generator of the LatM-HMM is defined as

$$\{\boldsymbol{g}|\boldsymbol{g}:\boldsymbol{z}\rightarrow\boldsymbol{x}, s\in\mathcal{S}, \boldsymbol{z}\sim p_s(\boldsymbol{z})\}. \tag{30}$$

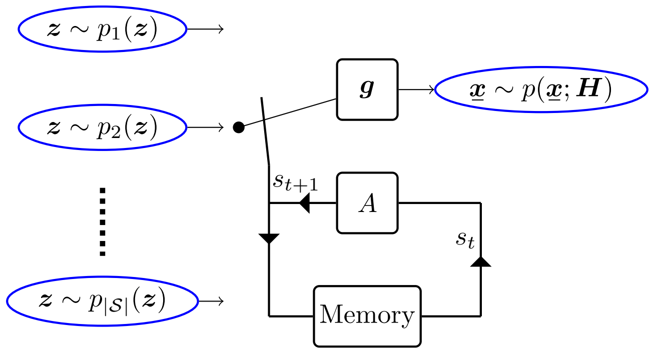

Figure 4: LatM-HMM Model defined by $\boldsymbol{H} = \{\mathcal{S}, \boldsymbol{q}, A, p(\boldsymbol{x}|\boldsymbol{s}; \boldsymbol{\Phi})\}$

We use $\boldsymbol{f} = \boldsymbol{g}^{-1}$ to denote inverse of $\boldsymbol{g}$ and use $\boldsymbol{\theta}$ to denote the parameter set of $\boldsymbol{g}$. Then the conditional probability for LatM-HMM is modeled as

$$\begin{aligned} p(\boldsymbol{x}|s; \boldsymbol{\Phi}) &= p_s(\boldsymbol{z}) \left| \det \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right) \right| \\ &= p_s(\boldsymbol{f}(\boldsymbol{x})) \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right| \end{aligned} \tag{31}$$

The parameter set for this model to be decide is $\boldsymbol{\Phi} = \{\boldsymbol{\theta}, \boldsymbol{\omega}_s, \forall s \in \mathcal{S}\}$. Then the problem in Equation 19 can be reformulated as:

$$\begin{aligned} &\max_{\boldsymbol{\Phi}} \mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}}) \\ =& \max_{\boldsymbol{\theta}, \boldsymbol{\omega}_s, \forall s \in \mathcal{S}} \sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}-1} \sum_{s_t^{(r)}=1}^{|\mathcal{S}|} p(s_t^{(r)} | \underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \left[\log p_{s_t^{(r)}}(\boldsymbol{f}(\boldsymbol{x}_t^{(r)})) + \log \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x}_t^{(r)})}{\partial \boldsymbol{x}_t^{(r)}} \right) \right| \right]. \end{aligned} \tag{32}$$

3.3.4 Generator Shared HMM (GSHMM)

To be continued...

4 On Implementation of acoustic signal

Found a HMM python lib that basics provide needed API for us, see hmmlearn. Saikat also has suggestion.

For problem Equation 19 we are going to use our generative models to solve. I have the following consideration to revised our LatMM and GenMM for this application:

- Use factorized model instead of additive mixture model, to make likelihood computation logarithm domain compatible;
- Use full EM fashion instead of mini-batch fashion for training: store generative model as old for EM, there are always two neural networks working, one old for probability evaluation and one new for optimization.

5 On Implementation of Planning

Refer to [1] and its experiments.

References

[1] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J. Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *CoRR*, abs/1807.09341, 2018.