Powering Hidden Markov Model by Generative Models

Firsthand Scientists

February 25, 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): $\mathcal{H} := \{\boldsymbol{H} | \{\mathcal{S}, \boldsymbol{q}, A, p(\underline{\boldsymbol{x}}|\underline{\boldsymbol{s}}; \boldsymbol{\Phi})\}$,

- $\mathcal{S}$ 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 $p(\underline{\boldsymbol{x}}|\underline{\boldsymbol{s}}; \boldsymbol{\Phi})$ 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 $\underline{\boldsymbol{x}}$.

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

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

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

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

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.

3.1 Initial Probability Update

Equation 7 can be written as:

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

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:

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

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

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

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

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

is

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

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:

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

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.

3.3.1 Generator Mixed HMM (GenM-HMM)

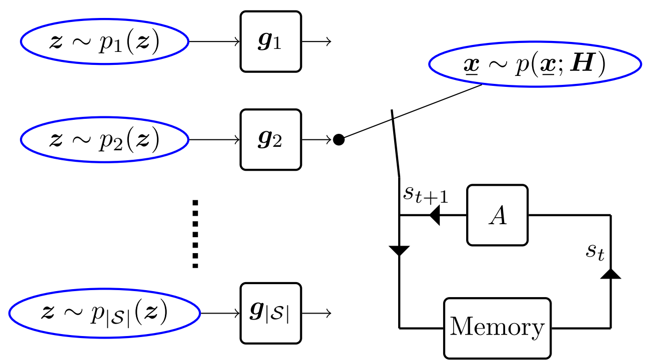

Figure 1: GenM-HMM Model defined by $\boldsymbol{H} = \{\mathcal{S}, \boldsymbol{q}, A, p(\boldsymbol{x}|\boldsymbol{s}; \boldsymbol{\Phi})\}$

For this proposal, we seek to use a generator mixed HMM scheme, termed as GenM-HMM. We define a set of generators for GenM-HMM:

$$\{\boldsymbol{g}_s | s \in \mathcal{S}, \boldsymbol{g}_s : \boldsymbol{z} \to \boldsymbol{x}, \boldsymbol{z} \sim p_s(\boldsymbol{z})\}. \tag{20}$$

Thus there are total $|\mathcal{S}|$ generators. $p(\boldsymbol{x}|s; \boldsymbol{\Phi})$ is induced as $\boldsymbol{g}_s(\boldsymbol{z}) \sim p(\boldsymbol{x}|s; \boldsymbol{\Phi})$ where $\boldsymbol{z} \sim p_s(\boldsymbol{z})$ for $s \in \mathcal{S}$. Let us denote the inverse of $\boldsymbol{g}_s$ as $\boldsymbol{f}_s = \boldsymbol{g}_s^{-1}$. We have the s-th component of the GenM-HMM model as

$$\begin{aligned} p(\boldsymbol{x}|s; \boldsymbol{\Phi}) &= p_s(\boldsymbol{z}) \left| \det \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right) \right| \\ &= p_s(\boldsymbol{f}_s(\boldsymbol{x})) \left| \det \left(\frac{\partial \boldsymbol{f}_s(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right| \end{aligned} \tag{21}$$

where $p_s(\boldsymbol{z})$ is the latent source distribution for $s = 1, 2, \cdots, |\mathcal{S}|$.

Let us denote the parameter set that defines latent distribution $p_s(z)$ by $\boldsymbol{\omega}_s$ and the parameter set that defines generator $\boldsymbol{g}_s$ by $\boldsymbol{\theta}_s$. Then $\boldsymbol{\Phi} = \{\boldsymbol{\theta}_s, \boldsymbol{\omega}_s, \forall s \in \mathcal{S}\}$. The problem in Equation 19 can be reformulated as:

$$\begin{aligned} &\max_{\boldsymbol{\Phi}} \mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}}) \\ =& \max_{\boldsymbol{\theta}_s, \boldsymbol{\omega}_s, \forall s \in \mathcal{S}} \sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}-1} \sum_{s_t^{(r)}=1}^{|\mathcal{S}|} p(s_t^{(r)} | \underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \left[\log p_{s_t^{(r)}}(\boldsymbol{f}_{s_t^{(r)}}(\boldsymbol{x}_t^{(r)})) + \log \left| \det \left(\frac{\partial \boldsymbol{f}_{s_t^{(r)}}(\boldsymbol{x}_t^{(r)})}{\partial \boldsymbol{x}_t^{(r)}} \right) \right| \right]. \end{aligned} \tag{22}$$

The diagram of GenM-HMM is shown as follows.

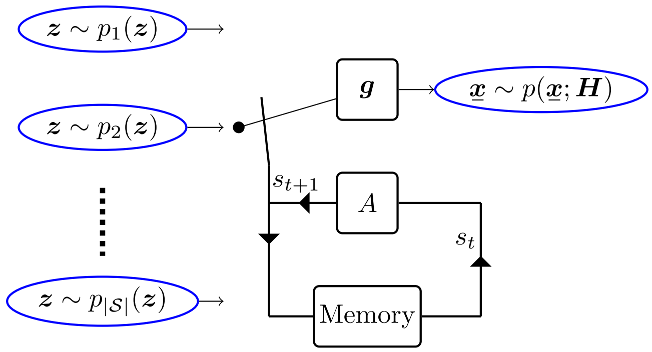

Figure 2: LatM-HMM Model defined by $\boldsymbol{H} = \{\mathcal{S}, \boldsymbol{q}, A, p(\boldsymbol{x}|\boldsymbol{s}; \boldsymbol{\Phi})\}$

3.3.2 Latent-source Mixed HMM (LatM-HMM)

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

$$\{\boldsymbol{g}|\boldsymbol{g} : \boldsymbol{z} \rightarrow \boldsymbol{x}, s \in \mathcal{S}, \boldsymbol{z} \sim p_s(\boldsymbol{z})\}. \tag{23}$$

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

$$\begin{aligned} p(\boldsymbol{x}|s; \boldsymbol{\Phi}) &= p_s(\boldsymbol{z}) \left| \det \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right) \right| \\ &= p_s(\boldsymbol{f}(\boldsymbol{x})) \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right| \end{aligned} \tag{24}$$

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:

$$\begin{aligned} &\max_{\boldsymbol{\Phi}} \mathcal{Q}(\boldsymbol{\Phi}; \boldsymbol{H}^{\text{old}}) \\ =&\max_{\boldsymbol{\theta}, \boldsymbol{\omega}_s, \forall s \in \mathcal{S}} \sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}-1} \sum_{s_t^{(r)}=1}^{|\mathcal{S}|} p(s_t^{(r)}|\underline{\boldsymbol{x}}^{(r)}; \boldsymbol{H}^{\text{old}}) \left[\log p_{s_t^{(r)}}(\boldsymbol{f}(\boldsymbol{x}_t^{(r)})) + \log \left| \det \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x}_t^{(r)})}{\partial \boldsymbol{x}_t^{(r)}} \right) \right| \right]. \end{aligned} \tag{25}$$

To be continued...

4 On Implementation

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.