Probabilistic Graphical Model (PGM)

1 Probabilistic Graphical Model (PGM)

The PGM uses the graphic to represent the probability distribution. Firstly, some rules for (continuous random variable),

$$
\begin{aligned}
&\text{sum rule: } p(x_1) = \int_{x_2} p(x_1, x_2) dx_2 \\
&\text{product rule: } p(x_1, x_2) = p(x_1|x_2)p(x_2) \\
&\text{chain rule: } p(x_1, x_2, \cdots, x_n) = p(x_1) \prod_{i=2}^{p} p(x_i|x_{i+1}, x_{i+2}, \cdots, x_p) \\
&\text{bayesian rule: } p(x_1|x_2) = \frac{p(x_2|x_1)p(x_1)}{p(x_2)}
\end{aligned}
\tag{1}
$$

The PGM comprises three theoretical parts:

- representation:
  - directed graphical model (Bayesian network)
  - undirected graphical model (Markov network)
- inference:
  - precise inference (Variational Inference)
  - approximated inference (MCMC)
- learning:
  - parameters learning (EM)

1.1 Bayesian Network

The joint probability can be obtained through factorization.

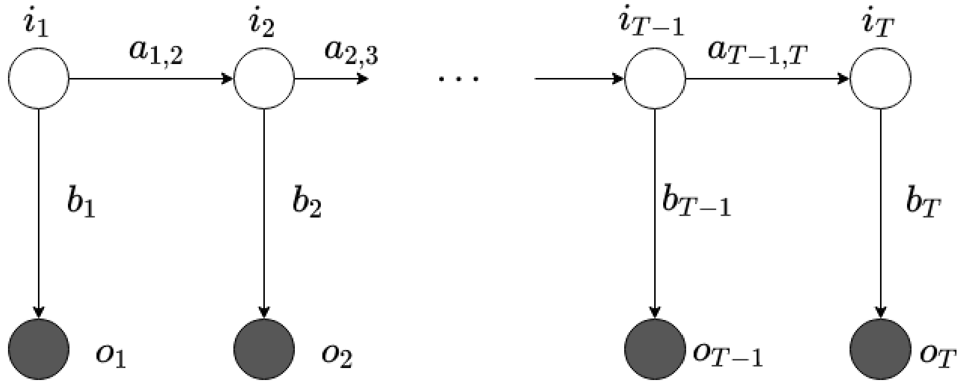

Fig. 1. HMM Model.

2 Hidden Markov Model (HMM)

2.1 Components and Problems

Components:

- Initial state $\lambda = (\pi, A, B)$, where π is the initialize probability distribution, A is the state transfer matrix, and B is the launch matrix.
- $I = (i_1, i_2, \cdots, i_T)$ state sequence, $i_{t,t\in T} = (q_1, q_2, \cdots, q_N) = Q$.
- $O = (o_1, o_2, \cdots, o_T)$ observe sequence, $o_{t,t\in T} = (v_1, v_2, \cdots, v_N) = V$.
- $a_{i,j} = P(i_{t+1} = q_j | i_t = q_i)$, $b_j(k) = P(o_t = v_k | i_t = q_j)$.
- $p(i_{t+1}|i_t, i_{t-1}, \cdots, i_1, o_t, o_{t-1}, \cdots, o_1) = p(i_{t+1}|i_t)$.
- $p(o_t|i_t, i_{t-1}, \cdots, i_1, o_{t-1}, \cdots, o_1) = p(o_t|i_t)$

Problem

- Calculate probability. Known $\lambda = (A, B, \pi)$ and observe sequence $O = (o_1, o_2, \cdots, o_T)$, compute the probability $P(O|\lambda)$ of O appearance of model λ.
- Learning. $\lambda = \arg\max_{\lambda} p(O|\lambda)$, EM algorithm.
-

2.2 Compute the probability of $P(O|\lambda)$ (Forward-Backward Algorithm)

Forward algorithm.

$$\begin{aligned}
p(O|\lambda) &= \sum_I p(O, I|\lambda) = \sum_I P(I|\lambda)P(O|I, \lambda) \\
p(I|\lambda) &= p(i_1, i_2, \cdots, i_t|\lambda) \\
&= p(i_t|i_{t-1}, i_{t-2}, \cdots, i_1, \lambda)p(i_1, i_2, \cdots, i_{t-1}|\lambda) \\
&= p(i_t|i_{t-1})p(i_1, i_2, \cdots, i_{t-1}|\lambda) \\
&= a_{t-1,t}p(i_1, i_2, \cdots, i_{t-1}|\lambda) \\
&= \pi_1 \prod_{t=2}^{T} a_{i_{t-1},i_t} \\
p(O|I, \lambda) &= p(o_1, o_2, \cdots, o_t|I, \lambda) \\
&= p(o_t|o_{t-1}, o_{t-2}, \cdots, o_1, I, \lambda)p(i_1, i_2, \cdots, i_{t-1}|I, \lambda) \\
&= p(o_t|i_t = q_t)p(i_1, i_2, \cdots, i_{t-1}|I, \lambda) \\
&= \prod_{t=1}^{T} b_{i_t}(o_t)
\end{aligned} \tag{2}$$

Therefore,

$$p(O|\lambda) = \sum_I \pi_1 \prod_{t=2}^{T} a_{i_{t-1},i_t} \prod_{t=1}^{T} b_{i_t}(o_t) \tag{3}$$

Time complexity is $O(N^T)$.

$$\begin{aligned} \text{Let} \quad & a_t(i) = p(o_1, o_2, \cdots, o_T, i_t = q_i, |\lambda) \\ \therefore \quad & a_T(i) = p(O, i_t = q_i|\lambda) \\ & p(O|\lambda) = \sum_{i=1}^{N} p(O, i_T = q_i|\lambda) = \sum_{i=1}^{N} a_T(i) \end{aligned} \tag{4}$$

$$\begin{aligned} a_{t+1}(j) &= p(o_1, o_2, \cdots, o_{t+1}, i_{t+1} = q_j, |\lambda) \\ &= \sum_{i=1}^{N} p(o_1, o_2, \cdots, o_{t+1}, i_{t+1} = q_j, i_t = q_i|\lambda) \\ &= \sum_{i=1}^{N} p(o_{t+1}|o_1, o_2, \cdots, o_t, i_{t+1} = q_j, i_t = q_i, \lambda) p(o_1, o_2, \cdots, o_t, i_{t+1} = q_j, i_t = q_i|\lambda) \\ &= \sum_{i=1}^{N} p(o_{t+1}|i_{t+1} = q_j) p(o_1, o_2, \cdots, o_t, i_{t+1} = q_j, i_t = q_i|\lambda) \\ &= \sum_{i=1}^{N} b_j(o_{t+1}) p(i_{t+1} = q_j|o_1, o_2, \cdots, o_t, i_t = q_i, \lambda) p(o_1, o_2, \cdots, o_t, i_t = q_i|\lambda) \\ &= \sum_{i=1}^{N} b_j(o_{t+1}) p(i_{t+1} = q_j|i_t = q_i, \lambda) p(o_1, o_2, \cdots, o_t, i_t = q_i|\lambda) \\ &= \sum_{i=1}^{N} b_j(o_{t+1}) a_{i,j} a_t(i) \\ &= b_j(o_{t+1}) \sum_{i=1}^{N} a_{i,j} a_t(i) \end{aligned} \tag{5}$$

Backward algorithm.

$$\begin{aligned} \text{Let} \;\; \beta_t(i) &= p(o_{t+1}, o_t, \cdots, o_1|i_t = q_i, \lambda) \\ p(O|\lambda) &= p(o_1, o_2, \cdots, o_t|\lambda) \\ &= \sum_{i=1}^{N} p(o_1, o_2, \cdots, o_t, i_1 = q_i|\lambda)(joint \;\; distribution) \\ &= \sum_{i=1}^{N} p(o_1, o_2, \cdots, o_t|i_1 = q_i, \lambda) p(i_1 = q_i|o_1, o_2, \cdots, o_t) \\ &= \sum_{i=1}^{N} \pi_i p(o_1, o_2, \cdots, o_t|i_1 = q_i, \lambda) \\ &= \sum_{i=1}^{N} \pi_i p(o_1|o_2, \cdots, o_t, i_1 = q_i, \lambda) p(o_2, \cdots, o_t, i_1 = q_i|\lambda) \\ &= \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) \end{aligned} \tag{6}$$

$$
\begin{aligned}
\beta_t(i) &= p(o_{t+1}, o_{t+2}, \cdots, o_T | i_t = q_i, \lambda) \\
&= \sum_{j=1}^{N} p(o_{t+1}, o_{t+2}, \cdots, o_T, i_{t+1} = q_j | i_t = q_i, \lambda) \\
&= \sum_{j=1}^{N} p(o_{t+1}, o_{t+2}, \cdots, o_T | i_{t+1} = q_j, i_t = q_i, \lambda) p(i_{t+1} = q_j | i_t = q_i) \\
&\because \;\; i_t \;\rightarrow\; i_{t+1} \text{ and } i_{t+1} \;\rightarrow\; o_{t+1}, \text{ if } i_{t+1} \text{ is known} \\
&\therefore \;\; i_t \text{ and } i_{t+1} \text{ are independent} \\
&\therefore \;\; p(o_{t+1}, o_{t+2}, \cdots, o_T | i_{t+1} = q_j, i_t = q_i, \lambda) = p(o_{t+1}, o_{t+2}, \cdots, o_T | i_{t+1} = q_j, \lambda) \\
&= \sum_{j=1}^{N} p(o_{t+1}, o_{t+2}, \cdots, o_T | i_{t+1} = q_j, \lambda) a_{i,j} \\
&= \sum_{j=1}^{N} p(o_{t+1} | o_{t+2}, \cdots, o_T, i_{t+1} = q_j, \lambda) p(o_{t+2}, o_{t+3}, \cdots, o_T, | i_{t+1} = q_j, \lambda) a_{i,j} \\
&= \sum_{j=1}^{N} b_j(o_{t+1}) \beta_{t+1}(j) a_{i,j}
\end{aligned} \tag{7}
$$

Conclusion, forward algorithm:

Algorithm 1: HMM Forward Algorithm

Result: the probability $P(O|\lambda)$ of observed sequence O.
Known $\lambda = (\pi, A, B)$;

1. For $t = 1, 2, \cdots, T-1$, $a_{t+1} = b_j(o_{t+1}) \sum_{i=1}^{N} a_{i,j} a_t(i)$;
2. $P(O|\lambda) = \sum_{i=1}^{N} a_T(i)$;

backward algorithm:

Algorithm 2: HMM Backward Algorithm

Result: the probability $P(O|\lambda)$ of observed sequence O.
Known $\lambda = (\pi, A, B)$;

1. For $t = T-1, T-2, \cdots, 1$, $\beta_t(i) = \sum_{j=1}^{N} b_j(o_{t+1}) \beta_{t+1}(j) a_{i,j}$;
2. $P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)$

2.3 Learning

Target:

$$\lambda_{MLE} = \arg\max_{\lambda} p(O|\lambda) \tag{8}$$

General EM algorithm:

$$\lambda^{t+1} = \arg\max_{\lambda} \sum_{I} p(O, I|\lambda) p(I|O, \lambda^t) \tag{9}$$

The learning target is

$$
\begin{aligned}
&\because \;\; p(I|O, \lambda^t) = \frac{p(O, I|\lambda^t)}{p(O|\lambda^t)} \\
&\therefore \;\; p(O|\lambda^t) \text{ is not relevant to } \lambda \\
&\quad = \arg\max_{\lambda} \sum_{I} p(O, I|\lambda) p(I, O|\lambda^t)
\end{aligned} \tag{10}
$$

2.3.1 Learning for π

E-step:

$$\begin{aligned}
\because\ & p(O,I|\lambda^t)=\sum_I \pi_1 \prod_{i=2}^{T} a_{i_{t-1},i_t} \prod_{i=1}^{T} b_{i_t}(o_t) \\
\therefore\ & Q(\lambda,\lambda^t)=\sum_I log\left[p(O,I|\lambda)p(O,I|\lambda^t)\right] \\
& =\sum_I log\left[\left[log\pi_i+\sum_{i=2}^{T} log(a_{i_{t-1},i_t})+\sum_{i=1}^{T} log(b_{i_t}(o_t))\right]p(O,I|\lambda^t)\right] \\
\because\ & \sum_{i=2}^{T} log(a_{i_{t-1},i_t}) \text{ and } \sum_{i=1}^{T} log(b_{i_t}(o_t)) \text{ and are not related to } \pi \\
\therefore\ & =\sum_I log\left[\pi_{i_1}p(O,I|\lambda^t)\right]
\end{aligned} \tag{11}$$

M-step:

$$\begin{aligned}
\pi^{t+1} &= \arg\max_{\pi} \sum_I \left[log\pi_{i_1}p(O,I|\lambda^t)\right] \\
&= \arg\max_{\pi} \sum_I \left[log\pi_{i_1}p(O,i_1,i_2,\cdots,i_t|\lambda^t)\right] \\
&= \arg\max_{\pi} \sum_{i_1}\cdots\sum_{i_t} \left[log\pi_{i_1}p(O,i_1,i_2,\cdots,i_t|\lambda^t)\right] \\
&\because\ \sum_{i_t} p(O,i_1,i_2,\cdots,i_t|\lambda^t) \text{ is to compute the marginal probability of } i_t \\
\therefore\ \pi^{t+1} &= \arg\max_{\pi} \sum_{i_1} \left[log\pi_{i_1}p(O,i_1|\lambda^t)\right]
\end{aligned} \tag{12}$$

Build the Lagrange function

$$\begin{aligned}
& L(\pi_i,\eta)=\sum_{i_1}\left[log\pi_{i_1}p(O,i_1=q_i|\lambda^t)\right]+\eta\left(\sum_{i=1}^{N}\pi_i-1\right) \\
& \frac{\partial L}{\partial \pi_i}=\frac{1}{\pi_i}p(O,i_1=q_i|\lambda^t)+\eta=0 \\
& p(O,i_1=q_i|\lambda^t)+\pi_i\eta=0 \quad A \\
& \because\ \sum_{i=1}^{N}\pi_i=1 \\
& \therefore\ \sum_{i=1}^{N}\left[p(O,i_1=q_i|\lambda^t)+\pi_i\eta\right]=0 \\
& \sum_{i=1}^{N}\left[p(O,i_1=q_i|\lambda^t)\right]=-\eta \\
& \eta=-p(O|\lambda^t) \qquad B \\
& \text{use B in function A} \\
& \therefore\ \pi_i=\frac{p(O,i_1=q_i|\lambda^t)}{p(O|\lambda^t)}
\end{aligned} \tag{13}$$

2.3.2 Learning for a_{ij}

E-step:

$$\begin{aligned}
\because\ & p(O,I|\lambda^t)=\sum_I \pi_1 \prod_{i=2}^{T} a_{i_{t-1},i_t} \prod_{i=1}^{T} b_{i_t}(o_t) \\
\therefore\ & Q(\lambda,\lambda^t)=\sum_I log\left[p(O,I|\lambda)p(O,I|\lambda^t)\right] \\
& =\sum_I log\left[\left[log\pi_i+\sum_{i=2}^{T} log(a_{i_{t-1},i_t})+\sum_{i=1}^{T} log(b_{i_t}(o_t))\right]p(O,I|\lambda^t)\right]
\end{aligned} \tag{14}$$

M-step:

$$
\begin{aligned}
a_{ij}^{t+1} &= \arg\max_{a} \sum_{t=2}^{T} log(a_{i_t,i_{t+1}}) p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t) \\
&= \arg\max_{a} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} log(a_{i_t,i_{t+1}}) p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t) \\
\because \ & \sum_{j=1}^{N} a_{ij} = 1 \\
\therefore \ J(a) &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} log(a_{i_t,i_{t+1}}) p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t) + \eta(\sum_{j=1}^{N} a_{ij} - 1) \\
\frac{\partial J(a)}{a_{i_t,i_{t+1}}} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} \frac{p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t)}{a_{i_t,i_{t+1}}} + \sum_{j=1}^{N} \eta_j = 0 \\
& \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t) + a_{i_t,i_{t+1}} \sum_{j=1}^{N} \eta_j = 0 \\
& a_{i_t,i_{t+1}} = -\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} p(O, i_{t+1} = q_i, i_t = q_j|\lambda^t)}{\sum_{j=1}^{N} \eta_j}
\end{aligned}
\tag{15}
$$