Hidden Markov Model

> 參考演算法筆記 - HMM實作的ＨＭＭ模型
>
> 我實作的程式碼
>
> 以下是一些筆記～～

HMM 的基本架構

- 全部有State $S = \{s_0, ..., s_{len\_s-1}\}, len(S) = len\_s$
- 全部有Observe value : $O = \{o_0, ..., o_{len\_o-1}\}, len(O) = len\_o$

每個state有機率可以走到其他state，且每個state有機率可以突出一個Observe value

HMM的參數

- 全部的參數 $\lambda = \{\Pi, A, B\}$:
  - init probability : $\Pi = [\pi_i]_{len\_s}$ ，初始時由每個state開始的機率
  - State translation matric : $A = [a_{ij}]_{len\_s \times len\_s}$ ，a_{ij} 為 state i 轉到 state j 的機率
  - Observe matric : $B = [b_j(k)]_{len\_s \times len\_o}$ ，$b_j(k)$為state j 吐出 observe value k 的機率

HMM的假設：

- 假設 - 當前狀態只與前一個狀態有關 $P(X_{t_{n+1}}|X_{t_n}, ... X_{t_1}) = P(X_{t_n+1}|X_{t_n})$
- 假設 - 狀態轉移與時間無關 $P(X_{t_{i+1}}|X_{t_i}) = P(X_{t_{j+1}}|X_{t_j}), \forall i, j$

假設我們全部觀測了T個time step，$T = \{0, ..., len\_t - 1\}, len(T) = len\_t$

每個Time Step都可以觀測到(x_{t_i}, y_{t_i})，每個x_{t_i} 為在time $\{t_i\}$下的State，y_{t_i} 為該state吐出的Observe Value

因此$X = o_{t_1}, ..., o_{t_{len_t-1}}$ 為在T個time step觀察到的觀測序列，$Y = \{s_{t_1}, ..., s_{t_{len_t-1}}\}$為T個time step的狀態序列的轉換

程式實作

HMM的三個問題：

- evaluate.py:
  - The Evaluation Problem : 用dynamic programming，有forward與backward兩種算法，細節推倒可以看程式碼上的註解
  - The Decoding Problem：Viterbi Algorithm，跟forward很像，換成取最大值
- Train.py
  - The Learning Problem：EM Algorithm，基本的思路就是取機率的最大值
    - Pi => 更新成在 T 個steps中可以觀測到符合的觀測序列的機率的最大值，(在t_0時穿越s_i的機率路徑和) / 在t_0時穿越所有state的機率總和
    - a_{ij} => (s_i 在所有step中轉換到s_j的機率) / (s_i 在所有step中轉換到所有state的機率)
    - $b_j(k)$ => (在所有step中s_i吐出k的機率) / (在所有step穿越s_i的機率)