隐马尔可夫模型(Hidden Markov Model，HMM).

SYSU-SPF 机器学习研修班讲义.　付星宇. 中山大学数学学院 && 逸仙学院

本讲义分为以下几个部分:

①- 例子引入.

②- HMM定义与经典问题.

③- Forward 算法 : $P(O \mid \lambda)$.

④- Viterbi 算法 : $I^* = \underset{I}{\operatorname{argmax}}\, P(I \mid O, \lambda)$

⑤- Baum-welch 算法 : $\lambda^* = \underset{\lambda}{\operatorname{argmax}}\, P(O \mid \lambda)$

①-例子引入.

考虑一个人，他每天有两种状态(state)，状态集合 I = {开心，不开心}.
在每一个状态下，他会做出一些可被人们察觉的行为(observation)，行为集 O = {睡觉，玩，工作}. 现在开始，我们每天都观测这个人的行为，假设观测了100天，那么我们会得到这个人100天的行为序列，但是这100天此人每天的状态是不可直接测得的.

HMM便是对这个例子的数学建模，我们希望通过对行为序列 $(O_1\ O_2 \cdots O_{100})$ 的研究，找出可能的隐藏状态序列.

↳注：状态序列不可测，故 Hidden

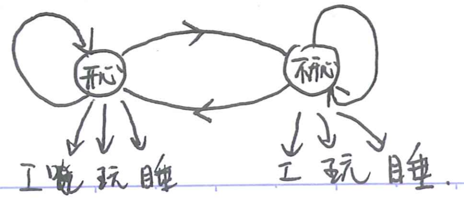
进一步假设：

- 齐次马氏性假设. 即 $\forall t \in N^+$，$t \geq 2$，第t天的状态只与第(t-1)天状态有关，i.e.

$$P(i_t = i \mid i_{t-1} = j, i_{t-2} = \cdots, \cdots, i_1 = \cdots)$$
$$= P(i_t = i \mid i_{t-1} = j) = a_{ji} \text{ (一个不依赖t的常数)}.$$

- 发射(emission)无关假设. 即 $\forall t \in N^+$，第t天的行为只与第t天的状态有关，i.e.

$$P(O_t = o \mid i_1\ O_1\ i_2\ O_2 \cdots \underset{=j}{i_t}\ i_{t+1}\ O_{t+1} \cdots i_T\ O_T)$$
$$= P(O_t = o \mid \underset{=j}{i_t}) = b_{jo} \quad \text{(一个不依赖t的常数)}.$$



可以用一张图来描述以上的模型：

HMM广范运用在交易择时，语音识别，基因测段等方向，据说HMM是美国著名对冲基金文艺复兴科技公司的大奖章基金背后的模型.

马氏链(Markov chain).

②一HMM定义与经典问题.

设 $\{I_t \mid t\in N^+\}$ 是一离散时间的马尔可夫过程，$I_t \in I$ (状态空间).
且 $|I| = n < +\infty$. 该马尔可夫过程的转移矩阵为 A，i.e.

$$A_{ij} = P(I_t=j \mid I_{t-1}=i), \quad \forall i,j \in I.$$

且初始分布为 π. i.e.

$$\pi_i = P(I_1 = i) \quad \forall i\in I.$$

由随机过程的知识知 (A,π) 完全决定了 $\{I_t \mid t\in N^+\}$.
此外在每一个时刻 t 下，状态 I_t 会发射(emit)一个观测 (observation) O_t.
设 $O_t \in O$ (观测空间)，且有发射矩阵 B，i.e.
$|O| = m$.

$$B_{io} = P(O_t = o \mid I_t = i)$$

其中 $\{O_t \mid t\in N^+\}$ 观测序列可见，而 $\{I_t \mid t\in N^+\}$ 是Hidden的.
令 $\lambda=(A,B,\pi)$，则 λ 完全决定一个HMM.

HMM有以下三个重要问题：

- 给定 $\lambda=(A,B,\pi)$，给定观测序列 $O_T=(O_1\ O_2 \cdots O_T)$，计算这一观测的可能性. i.e. $P(O_T \mid \lambda)$. (用Forward算法求解).
- 给定 $\lambda=(A,B,\pi)$，给定观测序列 $O_T=(O_1 \cdots O_T)$ 计算最可能导致这种观测的状态链 I_T^*. i.e. $I_T^* = \arg\max_I P(I \mid \lambda, O_T)$. (用Viterbi算法).
- 给定 $O_T=(O_1,O_2,\cdots,O_T)$，给定 $|I|=n$，$|O|=m$，找最可能的系统参数 $\lambda=(A,B,\pi)$. i.e. $\lambda^* = \arg\max_\lambda P(O \mid \lambda)$. (用Baum-welch算法).

③ - Forward 算法.

给定 $\lambda=(A,B,\pi)$ 给定 $O_T=(o_1,\cdots,o_T)$.

令 $\alpha_t(i)=P((o_1,o_2\cdots o_t), i_t=q_i \mid \lambda)$ （称为前向概率）.

由全概率公式 $P(O_T \mid \lambda) = \sum_{i_T\in I} P(O_T, i_T=i \mid \lambda)$

$$= \sum_{i_T\in I} \alpha_T(i)$$

且注意到 $\alpha_{t+1}(i) = P((o_1\ o_2\cdots o_{t+1}), i_{t+1}=i \mid \lambda)$.

$$= P(o_{t+1} \mid (o_1\ o_2\cdots o_t), i_{t+1}=i, \lambda) \times P((o_1,\cdots,o_t), i_{t+1}=i \mid \lambda)$$

（发射无关假设）$= b_{i o_{t+1}} \times P((o_1\cdots o_t)\ i_{t+1}=i \mid \lambda)$.

（全概） $= b_i o_{t+1} \times \sum_{j=1}^{N} P((o_1\cdots o_t)\ i_t=j, i_{t+1}=i \mid \lambda)$

$$= b_i o_{t+1} \sum_{j=1}^{N} P(i_{t+1}=i \mid i_t=j\ (o_1\cdots o_t), \lambda) \times P(i_t=j\ (o_1\cdots o_t) \mid \lambda)$$

（马氏性） $= b_{i o_{t+1}} \sum_{j=1}^{N} a_{ji}\, \alpha_t(j)$.

故我们可以一层一层，从下往上，递推计算 α 矩阵 $=[\alpha_t(i)]_{T\times N}$.

其中第一行 $\alpha_1(i) = P(o_1, i_1=i \mid \lambda)$

$$= P(o_1 \mid i_1=i, \lambda) \times P(i_1=i \mid \lambda)$$

$$= b_{i o_1} \times \pi_i.$$

这便是 Forward 算法. 比传统的计算方法快很多.

④- Viterbi算法

给定 $\lambda=(A,B,\pi)$ 给定 $O_T=(o_1\cdots o_T)$.

定义 $\delta_t(i)=\max\limits_{(i_1\cdots i_{t-1})} p((o_1\cdots o_t),i_t=i,(i_1\cdots i_{t-1})\mid\lambda)$.

显然有 $\max\limits_{I} p(I\mid O_T,\lambda)=\max\limits_{I}\dfrac{p(I,O_T\mid\lambda)}{p(O_T\mid\lambda)}$

$$=\frac{\max\limits_{I} p(I,O_T\mid\lambda)}{p(O_T\mid\lambda)}=\frac{\max\limits_{j\in I}\delta_T(j)}{p(O_T\mid\lambda)}$$ (动态规划)

而 $\delta_{t+1}(i)=\max\limits_{(i_1\cdots i_t)} p\{(o_1\cdots o_{t+1})$ 且 $i_{t+1}=i$ 且 $(i_1\cdots i_t)\mid\lambda)$

$$=\max_{(i_1\cdots i_t)} b_{io_{t+1}}\times p((o_1\cdots o_t)\ i_{t+1}=i\ (i_1\cdots i_t)\mid\lambda)$$

$$=\max_{(i_1\cdots i_t)} b_{io_{t+1}}\times a_{i_t i}\ p((o_1\cdots o_t)\ (i_1\cdots i_t)\mid\lambda).$$

(动态规划) $=\max\limits_{j}\left(\max\limits_{(i_1\cdots i_{t-1})} a_{ji}\times b_{io_{t+1}}\ p((o_1\cdots o_t)\ (i_1\cdots i_{t-1}\ j)\mid\lambda)\right)$

$$=\max_{j} b_{io_{t+1}}\,a_{ji}\,\delta_t(j)$$

故我们可以递推地计算最大似然状态列.

⑤ - Baum-welch 算法.

We want to maximize the log-likelihood function $L(\lambda) = \log P(O \mid \lambda)$. By EM Algorithm, we can maximize over $Q(\lambda, \bar{\lambda}) = \sum_I P(O, I \mid \bar{\lambda}) \times \log P(O, I \mid \lambda)$

while $P(O, I \mid \lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T)$

E Step:

$$\Rightarrow Q(\lambda, \bar{\lambda}) = \sum_I \log \pi_{i_1} P(O, I \mid \bar{\lambda}) + \sum_I \left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}} \right) P(O, I \mid \bar{\lambda})$$

$$+ \sum_I \left(\sum_{t=1}^{T} \log b_{i_t}(o_t) \right) P(O, \lambda, \mid \bar{\lambda})$$

M step:

Use Lagrange multiplier and set partial derivative to zero.

Details in P182 of '统计学习方法'.