

隐马尔科夫模型

深度学习专业班

欧阳若飞



全球人工智能学院

国内首家专注于AI技术职业化教育平台



主要内容

- ☐ 无监督学习
- ☐ 时间序列
- ☐ 隐马模型
- ☐ 前后向算法
- ☐ Viterbi算法
- ☐ Baum Welch算法



无监督学习

$x, y \rightarrow f$ 有监督学习

$x \rightarrow f$ 无监督学习

$x, z \rightarrow f$ z 是一个隐含变量

分类：有监督 y 看得见

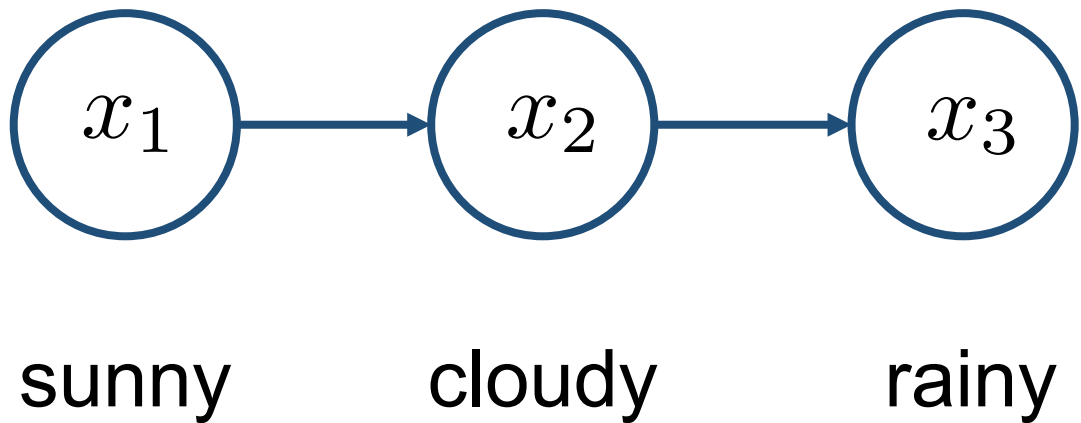
聚类：无监督 z 看不见

HMM是有监督还是无监督呢？

一般认为是无监督，因为 z 看不到
但也有 z 看得到的情况，此时为有监督



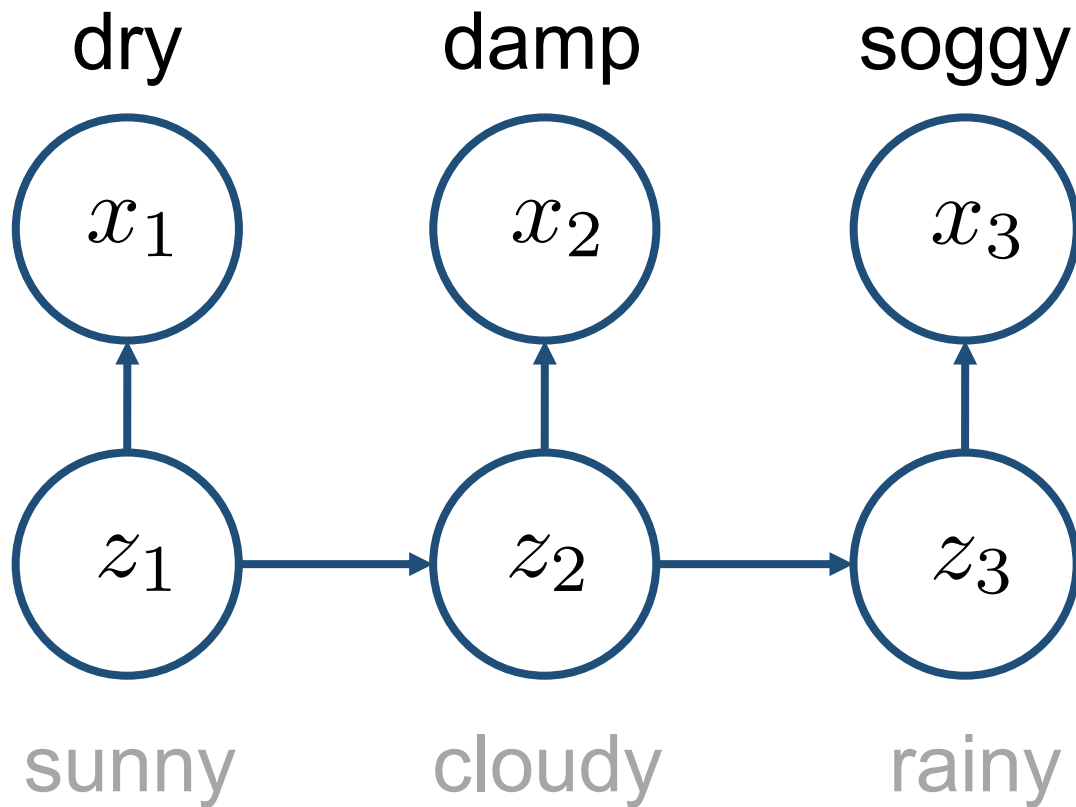
时间序列



马尔科夫过程: $p(x_t | x_{1:t-1}) = p(x_t | x_{t-1})$
 $p(x_3 | x_2, x_1) = p(x_3 | x_2)$



时间序列



隐马尔科夫过程：天气本身不能直接观测，而是看豆子的干燥程度去推测



HMM模型

一个地区的天气有自己的特定分布

	sunny	cloudy	rainy
冬天的北京	100%	0%	0%
新加坡	50%	0%	50%

这是我们对天气这个隐变量的最初感觉到的概率分布 π



HMM模型

给定昨天的天气，今天是什么天气呢？

旱季几乎天天晴天

雨季几乎天天下雨

	sunny	cloudy	rainy
sunny	90%	0%	10%
cloudy	40%	10%	40%
rainy	10%	0%	90%

天气这个隐变量的状态转移矩阵 A



HMM模型

假使超人每天夜里到新加坡看豆子来猜当天白天的天气。。。

晴天的话豆子比较干燥

雨天的话豆子比较潮湿

	dry	dryish	damp	soggy
sunny	0.6	0.2	0.15	0.05
cloudy	0.25	0.25	0.25	0.25
rainy	0.05	0.1	0.35	0.5

豆子干燥度这个观测变量的观测矩阵 B

为了方便理解后续推导，我们认为观测的概率满足multinomial distribution



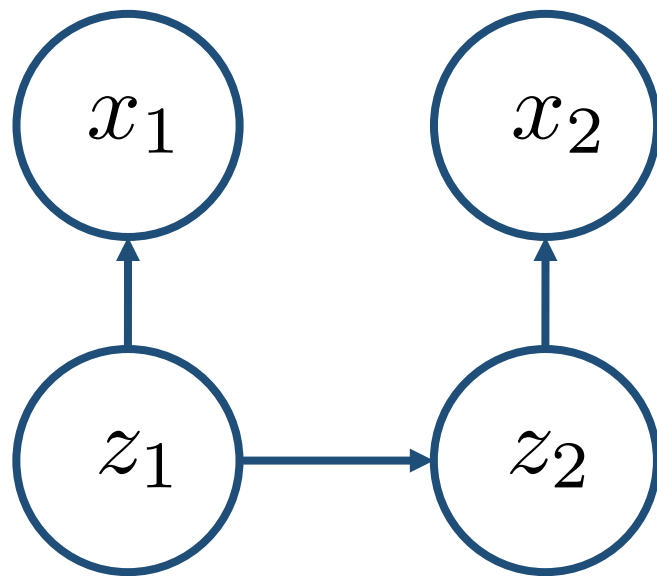
HMM模型

那么这个无聊的超人可以用HMM干下面三件事情:

- 1) 根据n天观测结果计算出现该观测结果的概率
(天天豆子都是潮湿的概率)
(forward-backward algorithm)
- 2) 根据n天观测结果估计最可能的天气序列
(豆子都是潮湿的, 那么最近可能都是雨天)
(Viterbi algorithm)
- 3) 根据n天观测结果学习转移矩阵和观测矩阵
(看了一百年新加坡豆子的干燥程度, 学习到了天气和农业的关系 π, A, B)
(Baum-Welch algorithm)



简单的概率图模型



$$p(x_1, x_2, z_1, z_2) = p(z_1)p(x_1|z_1)p(z_2|z_1)p(x_2|z_2)$$

按概率公式和马尔科夫性质慢慢推也可以，但很麻烦
直接按概率图模型的结构写比较快



前后向算法

$$p(x) = \int_z p(x, z) dz = \sum_z p(x, z)$$

$$p(x_1, x_2) = \sum_{z_1} \sum_{z_2} p(x_1, x_2, z_1, z_2)$$

$$p(x_{1:t}) = \sum_{z_{1:t}} p(x_{1:t}, z_{1:t})$$

$$\sum_{z_1=1}^K \sum_{z_2=1}^K \cdots \sum_{z_T=1}^K \quad \text{计算复杂度: } \mathcal{O}(K^T)$$



前向算法

$$\alpha_{z_t=k} = p(x_{1:t}, z_t = k)$$

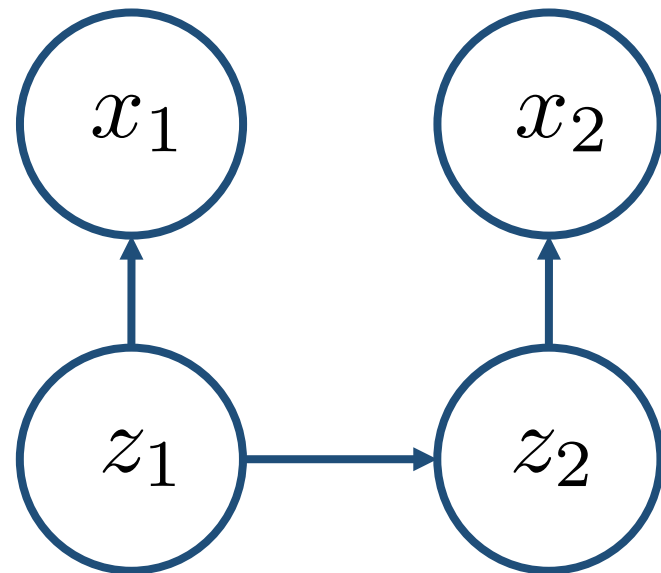
$$\alpha_{z_1=k} = p(x_1 | z_1 = k) p(z_1)$$

$$\alpha_{z_2=k} = p(x_1, x_2, z_2 = k)$$

$$= \sum_{k'=1}^K p(x_1, x_2, z_1 = k', z_2 = k)$$

$$= \sum_{k'=1}^K p(x_2 | z_2 = k) p(z_2 = k | z_1 = k') p(x_1, z_1 = k')$$

$$= p(x_2 | z_2 = k) \sum_{k'=1}^K p(z_2 = k | z_1 = k') \alpha_{z_1=k'}$$





前向算法

$$\alpha_{z_t=k} = p(x_{1:t}, z_t = k)$$

如何用这个记录的结果求 $p(x_{1:t})$

$$p(x_{1:t}) = \sum_{k=1}^K p(x_{1:t}, z_t = k) = \sum_{k=1}^K \alpha(z_t = k)$$

前向算法就是一个动态规划，把指数型复杂度降低到多项式级别 $\mathcal{O}(KT)$
这里的alpha就是动态规划里的备忘录



前向算法

观测第一个数据 x_0 , 穷举所有隐状态 $s=0, s=1$

$$\alpha_{t=0, s=0} = \pi_0 B_{s=0, x_0}$$

$$\alpha_{t=0, s=1} = \pi_1 B_{s=1, x_0}$$

$$\begin{bmatrix} \alpha_{t=0, s=0} \\ \alpha_{t=0, s=1} \end{bmatrix} = \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} \odot \begin{bmatrix} B_{s=0, x_0} \\ B_{s=1, x_0} \end{bmatrix}$$



前向算法

观测第二个数据 x_1 , 穷举所有隐状态 $s=0, s=1$

$$\alpha_{t=1,s=0} = \pi_0 B_{s'=0,x_0} A_{s'=0,s=0} B_{s=0,x_1} + \pi_1 B_{s'=1,x_0} A_{s'=1,s=0} B_{s=0,x_1}$$

$$\alpha_{t=1,s=1} = \pi_0 B_{s'=0,x_0} A_{s'=0,s=1} B_{s=1,x_1} + \pi_1 B_{s'=1,x_0} A_{s'=1,s=1} B_{s=1,x_1}$$

$$\alpha_{t=1,s=0} = \alpha_{t=0,s'=0} A_{s'=0,s=0} B_{s=0,x_1} + \alpha_{t=0,s'=1} A_{s'=1,s=0} B_{s=0,x_1}$$

$$\alpha_{t=1,s=1} = \alpha_{t=0,s'=0} A_{s'=0,s=1} B_{s=1,x_1} + \alpha_{t=0,s'=1} A_{s'=1,s=1} B_{s=1,x_1}$$

$$\begin{bmatrix} \alpha_{t=1,s=0} \\ \alpha_{t=1,s=1} \end{bmatrix} = \begin{bmatrix} \alpha_{t=0,s'=0} A_{s'=0,s=0} + \alpha_{t=0,s'=1} A_{s'=1,s=0} \\ \alpha_{t=0,s'=0} A_{s'=0,s=1} + \alpha_{t=0,s'=1} A_{s'=1,s=1} \end{bmatrix} \odot \begin{bmatrix} B_{s=0,x_1} \\ B_{s=1,x_1} \end{bmatrix}$$

$$\alpha_t = \alpha_{t-1}^\top A \odot B_{x_t}$$



后向算法

$$\beta(z_t = k) = p(x_{t+1:T} | z_t = k)$$

$$\beta(z_T = k) = 1$$

$$\beta(z_t = k) = \sum_{k'=1}^K \beta(z_{t+1} = k') p(x_{t+1} | z_{t+1}) p(z_{t+1} | z_t)$$



后向算法

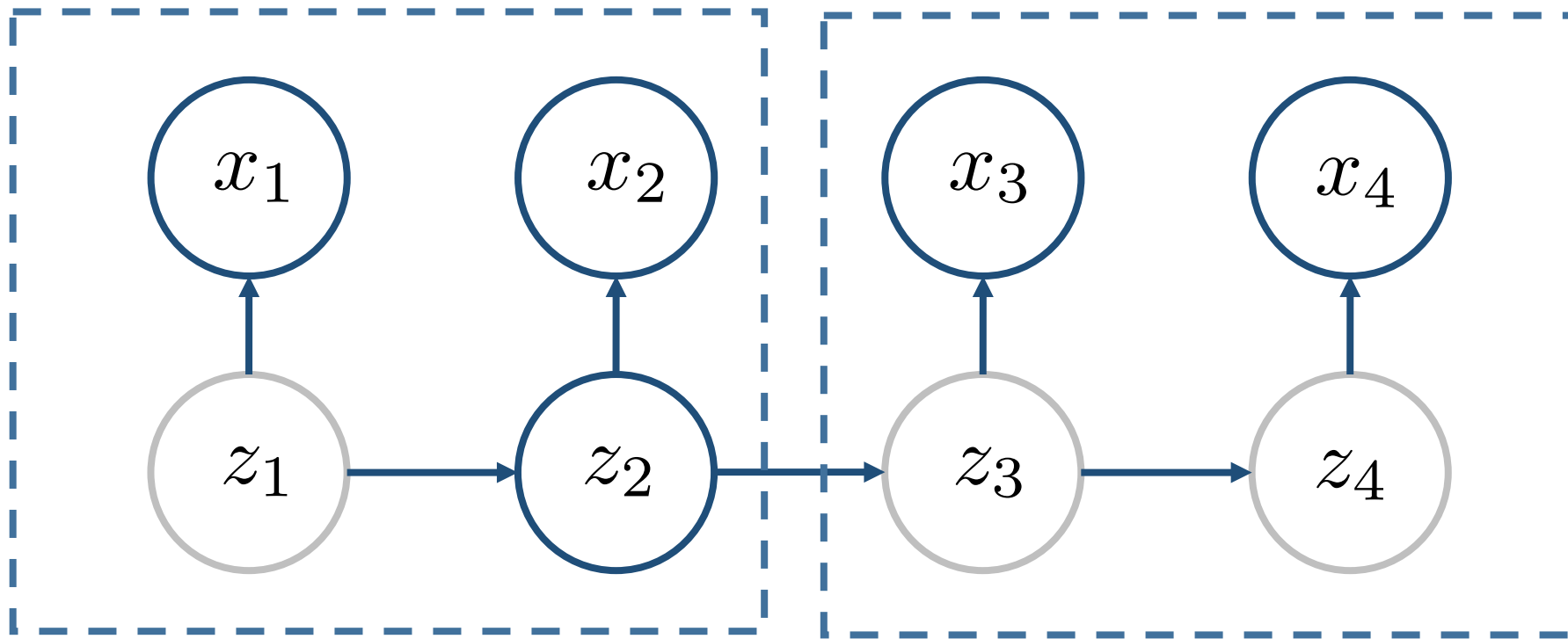
$$\beta_{z_1=k} = p(x_{1:T} | z_1 = k)$$

如何用这个记录的结果求 $p(x_{1:T})$

$$\begin{aligned} p(x_{1:T}) &= \sum_{k=1}^K p(x_{1:T}, z_1 = k) \\ &= \sum_{k=1}^K p(x_{1:T} | z_1 = k) p(z_1 = k) \\ &= \sum_{k=1}^K \beta(z_1 = k) p(z_1 = k) \end{aligned}$$



后向算法



$$\begin{aligned} p(x_{1:T}, z_t = k) &= p(x_{1:T} | z_t = k) p(z_t = k) \\ &= p(x_{1:t} | z_t = k) p(x_{t+1:T} | z_t = k) p(z_t = k) \\ &= p(x_{1:t}, z_t = k) p(x_{t+1:T} | z_t = k) = \alpha(z_t = k) \beta(z_t = k) \end{aligned}$$



后向算法

穷举T时刻所有隐状态 $s=0, s=1$

$$\beta_{t=T, s=0} = 1$$

$$\beta_{t=T, s=1} = 1$$



后向算法

观测最后一个数据 x_T ,穷举所有隐状态 $s=0,s=1$

$$\beta_{t=T-1,s=0} = A_{s=0,s'=0} B_{s'=0,x_T} \beta_{t=T,s'=0} + A_{s=0,s'=1} B_{s'=1,x_T} \beta_{t=T,s'=1}$$

$$\beta_{t=T-1,s=1} = A_{s=1,s'=0} B_{s'=0,x_T} \beta_{t=T,s'=0} + A_{s=1,s'=1} B_{s'=1,x_T} \beta_{t=T,s'=1}$$

$$\begin{bmatrix} \beta_{t=T-1,s=0} \\ \beta_{t=T-1,s=1} \end{bmatrix} = \begin{bmatrix} A_{s=0,s'=0} & A_{s=0,s'=1} \\ A_{s=1,s'=0} & A_{s=1,s'=1} \end{bmatrix} \begin{bmatrix} B_{s'=0,x_T} \beta_{t=T,s'=0} \\ B_{s'=1,x_T} \beta_{t=T,s'=1} \end{bmatrix}$$

$$\beta_t = A(B_{x_{t+1}} \odot \beta_{t+1})$$



Viterbi算法

求最可能的隐变量序列

算法和前向算法类似，只是：
备忘录不是记录当前的概率
而是当前的最大值



Viterbi算法

$$\begin{aligned} & \max_{z_{1:t-1}} p(x_{1:t}, z_{1:t}) \\ &= \max_{z_{1:t-1}} \underbrace{p(x_t | z_t) p(z_t | z_{t-1})}_{\text{blue underline}} \underbrace{p(x_{1:t-1}, z_{1:t-1})}_{\text{blue underline}} \\ &= \max_{z_{t-1}} \{ p(x_t | z_t) p(z_t | z_{t-1}) \max_{z_{1:t-2}} p(x_{1:t-1}, z_{1:t-1}) \} \end{aligned}$$

最终结果为: $\max_{z_t} \{ \max_{z_{t-1}} p(x_{1:t}, z_{1:t}) \}$

$$f(a) \geq 0 \text{ for } \forall a \quad g(a, b) \geq 0 \text{ for } \forall a, b$$

$$\max_{a, b} f(a)g(a, b) = \max_a \{ f(a) \max_b g(a, b) \}$$



Baum Welch算法

这个推导过于繁琐，而且实际使用并不需要了解细节

只需要知道用了EM参数估计的方法和拉格朗日算子

Baum Welch干的事情就是：

根据很多的观测序列估计 π, A, B



HMM实际使用

我们讨论两种情况:

- 1 隐变量实际上能观测到 (POS-TAG, 有监督学习)
- 2 隐变量观测不到 (股市行情, 无监督学习)



HMM实际使用

隐变量实际上能观测到 (POS-TAG)

根据 X 和 Z 直接计算 π, A, B

在需要预测的序列 x 上使用Viterbi算法得到 z

观测值: 单词

隐变量: 词性

状态分布: 词性的频率

转移矩阵: 词性转移频率

观测矩阵: 该词性出现的单词词频



HMM实际使用

隐变量观测不到 (股市行情)

根据 X 用Baum-Welch计算 π, A, B

在需要预测的序列 x 上使用Viterbi算法得到 z

观测值: 跳楼人数 (无, 少, 多)

隐变量: 总体行情

注意: 隐变量的含义是需要我们人工赋予的

$A \rightarrow$ 少 $B \rightarrow$ 多 A : 牛市, B : 熊市

但我们并不能在训练前就指定 A =牛市 B =熊市 (无监督学习)



HMM实例

观测值: dry dryish damp soggy

隐状态: sunny cloudy rainy

sunny	0.63
cloudy	0.17
rainy	0.20

	sunny	cloudy	rainy
sunny	0.5	0.375	0.125
cloudy	0.25	0.125	0.625
rainy	0.25	0.375	0.375

	dry	dryish	damp	soggy
sunny	0.6	0.2	0.15	0.05
cloudy	0.25	0.25	0.25	0.25
rainy	0.05	0.1	0.35	0.5



谢谢大家



助教微信: aischool1007