

标题 title

作者 *author*

2024 年 8 月 11 日

前言

目录

前言	i
第一部分 AI 的逻辑	1
第一章 合情推理	2
§1.1 命题逻辑的演绎推理	3
§1.2 合情推理的数学模型	8
1.2.1 合情推理的基本假设, 似然	9
1.2.2 似然与概率	12
1.2.3 先验与基率谬误	14
§1.3 合情推理的归纳强论证	15
1.3.1 归纳强论证	15
1.3.2 有效论证和归纳强论证的比较	18
§1.4 先验模型的存在性	21
§1.5 章末注记	23
第二章 Markov 链与决策	24
§2.1 Markov 链	24
§2.2 Markov 奖励过程 (MRP)	32
§2.3 Markov 决策过程 (MDP)	36
§2.4 隐 Markov 模型 (HMM)	43
2.4.1 评估问题	45
2.4.2 解释问题	46
§2.5 扩散模型	48
2.5.1 采样逆向过程	51

2.5.2 训练逆向过程	52
第二部分 信息与数据	54
第三章 信息论基础	55
§3.1 熵	55
3.1.1 概念的导出	55
3.1.2 概念与性质	58
3.1.3 熵与通信理论	63
§3.2 Kullback-Leibler 散度	66
3.2.1 定义	66
3.2.2 两个关于信息的不等式	68
3.2.3 在机器学习中的应用：语言生成模型	69
§3.3 附录：Shannon 定理的证明	70
§3.4 习题	72
§3.5 章末注记	73
第四章 Johnson-Lindenstrauss 引理	75
§4.1 机器学习中的数据	75
§4.2 矩法与集中不等式	76
§4.3 J-L 引理的陈述与证明	80
§4.4 J-L 引理的应用	84
§4.5 习题	85
§4.6 章末注记	85
第五章 差分隐私	86
§5.1 数据隐私问题	86
§5.2 差分隐私的定义与性质	88
§5.3 差分隐私的应用	92
5.3.1 随机反应算法	92
5.3.2 全局灵敏度与 Laplace 机制	93
5.3.3 DP 版本 Llyod 算法	95
§5.4 差分隐私与信息论	96
§5.5 习题	97

§5.6 章末注记	97
第三部分 决策与优化	98
第六章 凸分析	99
§6.1 决策与优化的基本原理	99
6.1.1 统计决策理论	99
6.1.2 优化问题	100
6.1.3 例子：网格搜索算法	103
§6.2 凸函数	105
§6.3 凸集	108
6.3.1 基本定义和性质	108
6.3.2 分离超平面定理	110
第七章 对偶理论	112
§7.1 条件极值与 Lagrange 乘子法	113
§7.2 Karush–Kuhn–Tucker 条件	116
§7.3 Lagrange 对偶	119
7.3.1 Lagrange 定理	119
7.3.2 弱对偶定理，强对偶定理	123
§7.4 应用：支持向量机 (SVM)	127
第八章 不动点理论	130
§8.1 Banach 不动点定理	130
§8.2 Brouwer 不动点定理	133
§8.3 不动点的一般视角	136
第四部分 逻辑与博弈	137
第九章 动态博弈	138
§9.1 输赢博弈	138
§9.2 随机博弈 (Markov 博弈)	143

第十章 静态博弈	149
§10.1 正则形式博弈	149
10.1.1 生成对抗网络	150
10.1.2 混合策略	152
§10.2 不完全信息博弈 (Bayes 博弈)	153
第五部分 认知逻辑	158
第十一章 模态逻辑基础	159
§11.1 模态逻辑的起源	159
11.1.1 三段论	159
11.1.2 非经典逻辑	160
§11.2 模态语言	161
§11.3 Kripke 语义与框架语义	165
§11.4 模态可定义性	169
第十二章 认知逻辑与共同知识	172
§12.1 “泥泞的儿童”谜题	172
§12.2 认知逻辑的基本模型与性质	174
12.2.1 “泥泞的儿童”再回顾	178
12.2.2 Aumann 结构	179
§12.3 对不一致达成一致	180
§12.4 Rubinstein 电子邮件博弈	183
第六部分 附录：预备知识	187
附录 A 线性代数基础	188
§A.1 线性空间	188
§A.2 线性映射	192
§A.3 矩阵	197
§A.4 双线性型与二次型	203
§A.5 带内积的线性空间	207
§A.6 行列式	213
§A.7 算子范数与谱理论	216

附录 B 微分学基础	222
§B.1 点集拓扑	222
B.1.1 度量空间, 范数	222
B.1.2 开集与闭集	225
B.1.3 紧致性, 收敛性, 完备性	228
B.1.4 连续映射	231
B.1.5 与实数序有关的性质	234
§B.2 一元函数的微分学	236
B.2.1 导数与微分的定义	237
B.2.2 微分学基本定理	240
§B.3 多元函数的微分学	242
B.3.1 微分、偏导数与导数的定义	242
B.3.2 微分学基本定理	248
B.3.3 隐函数定理	250
附录 C 概率论基础	254
§C.1 从朴素概率论到公理化概率论	254
C.1.1 Kolmogorov 概率论	254
C.1.2 条件概率, 独立性	258
§C.2 随机变量, 分布函数	262
C.2.1 基本定义	262
C.2.2 离散型随机变量	266
C.2.3 连续型随机变量	266
C.2.4 随机向量, 条件分布, 独立性	270
C.2.5 随机变量(向量)的函数	274
§C.3 随机变量的数字特征, 条件数学期望	277
C.3.1 数学期望, Lebesgue 积分	277
C.3.2 数学期望的性质	281
C.3.3 随机变量的内积空间	284
C.3.4 特征函数	286
C.3.5 条件数学期望	287
§C.4 多元正态分布 (Gauss 向量)	291

第一部分

AI 的逻辑

第二章 Markov 链与决策

我们都知道，人的推理和决策会受到时间的影响：我们宁愿要现在的十块钱也不要十年后的二十块钱。然而，人和人的耐心程度是不一样的。2010 年，来自意大利博洛尼亚大学 Manuela Sellitto 和她的同事进行了一项实验。他们的实验对象是一组大脑正常的人（对照组）、一组非眶额叶区受损的病人（实验组 1）和一组眶额叶周围受损的病人（实验组 2）。实验人员对被试进行提问，典型的问题如下：“你想要现在拿到 10 美元还是一个 月后拿到 12 美元？”。他们对食物和货币形式的奖励都进行了测试。

结果显示，眶额叶周围受损的病人更倾向于选择即时的奖励，而其他组的人更倾向于选择未来的奖励。如果货币奖励翻倍（例如，从 50 美元变为 100 美元），对照组和实验组 1 的人愿意等待 4 到 6 个月，而眶额叶周围受损的人（实验组 2）甚至不愿意等待 3 周。换句话说，人的耐心程度是大脑结构决定的，而不是自己可以轻易改变的！

以上例子不仅说明时间在决策中的重要性，更说明人脑中有特殊的结构和机制处理时间相关的决策问题。这样的观察对于人工智能也是同样重要的。本章我们将介绍 Markov 链，这是一种带有时间的概率模型，它是应用最为成功的含时决策模型之一。我们还将介绍人工智能领域基于 Markov 链的各种应用，包括 Markov 决策模型与强化学习、隐 Markov 模型以及扩散模型。

§2.1 Markov 链

在第一章中，我们说明了合情推理是人和 AI 非常重要的推理方式，这一推理模式基于 Bayes 概率论和似然。然而，这一模型对推理的假设是逻辑的、静态的，时间的概念并不出现在似然里面。例如，考虑一个罐子，里面有除颜色之外不可区分的 N 个球，有 n 个白球，剩下的是黑球。顺序从中拿出 N 个球，第 k 次拿出的球颜色是 W_k 或 B_k 。

用 Bayes 定理很容易证明，对任意 $i \neq j$ ， $\Pr(W_i|W_j) = \Pr(W_j|W_i)$ 。也就是说， $\Pr(W_i|W_j)$ 和 $\Pr(W_j|W_i)$ 不仅是可计算的，而且是相等的。然而，如果从推理的角度来说，我们基于时间更晚的状态推理时间更早的状态，这样的推理需要我们能够有对未来

的模型. 因此, 我们需要引入一个带有时间的推理模型, 这就是 Markov 链.

定义 2.1 (Markov 链) Markov 链 (马氏链) 是一个随机变量序列 $\{X_t\}_{t=0}^{\infty}$. 包含如下概念:

- 状态空间 \mathcal{S} : X_t 所有可能值构成的集合, 有限或者可数.
- 转移矩阵 \mathcal{P} (转移核): 下一时刻系统状态之间转移的概率. $\mathcal{P} = (p_{ij})_{i,j \in \mathcal{S}}$, p_{ij} 是从 i 状态转移到 j 状态的概率.
- Markov 性: 对任意时刻 $t = 1, \dots, n$ 和任意状态 $j, k, j_0, \dots, j_{t-1} \in \mathcal{S}$, 如下等式成立

$$\Pr(X_{t+1} = j | X_t = k, X_{t-1} = j_{t-1}, \dots, X_0 = j_0) = \Pr(X_{t+1} = j | X_t = k) = p_{kj}.$$

有时候也会考虑带初态的 Markov 链, 此时 X_0 服从分布 $\lambda = (\lambda_s)_{s \in \mathcal{S}}$.

我们给出的定义是简化的 Markov 链, 每个时刻之间的转移都是一样的转移矩阵, 这样的 Markov 链被称为时齐的. 有时候也会考虑非时齐的 Markov 链 (例如扩散模型), 即每个时刻之间的转移矩阵不一样, 这样的 Markov 链被称为非时齐的, 此时 t 时刻的转移矩阵是 $\mathcal{P}^{(t)}$, 定义中 Markov 性的转移概率是 $p_{kj}^{(t)}$.

Markov 链是一种简化的带时间的概率模型, 它最重要的性质是 Markov 性, 即在固定现在的情况下, 过去与未来相互独立. 这一性质的数学表述为:

命题 2.1 (Markov 性) 条件在 $X_n = i$ 下, $\{Y_m\}_{m=0}^{\infty} := \{X_{m+n}\}_{m=0}^{\infty}$ 是一个转移矩阵为 P 的 Markov 链, 并且与 (X_0, \dots, X_{n-1}) 相互独立.

证明留做习题.

我们考虑的 Markov 链还有时齐性, 即状态的转移不依赖当前时间, 只和当前的状态有关. 时齐性的数学表述为:

命题 2.2 设 $\{X_t\}_{t=0}^{\infty}$ 是一个 Markov 链, 那么对任意的 $t, m, n \in \mathbb{N}$ 和 $i, j, k \in \mathcal{S}$, 有 $\Pr(X_{m+n} = j | X_n = k) = \Pr(X_m = j | X_0 = k)$.

我们来看一个 Markov 链的例子.

例 2.1 (赌徒模型) 考虑公平对赌. 玩家 A 和 B 抛硬币来赌钱, A 赌正面, B 赌反面. 每一轮独立地抛硬币, 正面朝上的概率和反面朝上的概率相等, 都是 $1/2$. 赢的一方给输的一方一块钱. A 输 a 块钱破产, B 输 b 块钱破产, Z_i 是第 i 轮 A 的收入, $Z_0 = X_0 = 0$ 是 A 初始的收入. $X_n = Z_0 + \dots + Z_n$ 是 A 的累计收入. 那么, $\{X_n\}_{n \geq 0}$ 是一个 Markov 链.

- 状态空间: $\mathcal{S} = \{-a, -a+1, \dots, 0, 1, \dots, b\}$.
- 转移概率: 对 $-a < i < b-1$, $p_{i,i+1} = p_{i+1,i} = 1/2$; $p_{-a+1,-a} = p_{b-1,b} = 1/2$, $p_{-a,-a} = p_{b,b} = 1$; 其他值为 0. \square

转移矩阵可以画成图 2.1 所示的形式.

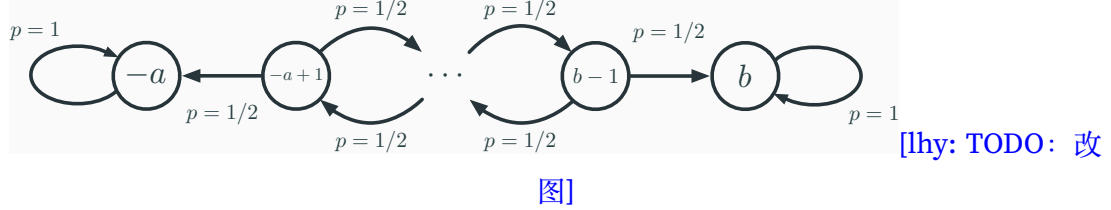


图 2.1: 赌徒模型的转移矩阵

接下来, 我们来看一个非时齐的 Markov 链的例子.

例 2.2 (Pólya 的坛子) 想象现在有一个坛子. 开始的时候, 坛子里有 1 个白球和 1 个黑球. 每一轮, 我们从坛子里拿出一个球, 观察颜色, 然后放回去, 再加入一个和刚才拿出的球颜色相同的球. 例如, 如果第一轮我们拿出一个白球, 那么第二轮开始时坛子里有 2 个白球和 1 个黑球.

设 X_t 是第 t 轮之后白球的数量, 我们会发现, 它是一个非时齐的 Markov 链, 转移概率为

- $\Pr(X_{t+1} = i+1 | X_t = i) = i/(t+2)$,
- $\Pr(X_{t+1} = i | X_t = i) = (t+1-i)/(t+2)$,
- 其他值的概率为 0.

这个 Markov 链的一个重要性质是当 $t \rightarrow \infty$ 时, X_t/t 趋于均匀分布 $\mathcal{U}[0,1]$.

这个性质可以通过对 t 进行归纳证明, 即 X_t 是 $\{1, \dots, t+1\}$ 上的均匀分布. $t=0$ 时, $X_0=1$, 显然成立. 假设对 t 成立, 即 X_t 是 $\{1, \dots, t+1\}$ 上的均匀分布. 那么对 $t+1$ 以及 $1 \leq j \leq t+2$, 根据全概率公式, 我们有

$$\begin{aligned}
 \Pr(X_{t+1} = j) &= \sum_{k=1}^{t+1} \Pr(X_{t+1} = j | X_t = k) \Pr(X_t = k) \\
 &= \frac{1}{t+1} (\Pr(X_{t+1} = j | X_t = j-1) + \Pr(X_{t+1} = j | X_t = j)) \\
 &= \frac{1}{t+1} \left(\frac{j-1}{t+2} + \frac{t+2-j}{t+2} \right) = \frac{1}{t+2}.
 \end{aligned}$$

于是，归纳成立。

Pólya 的坛子还有一种有趣的解读：如果我们把人生的每一次选择都看成放球，我们经常会基于现在好的东西（看到的黑球还是白球）投资自己的人生（放一个黑球还是白球），这样的结果是我们的人生会多姿多彩，百花齐放。□

我们回到赌徒模型。A 的累计收入 $\{X_n\}_{n \geq 0}$ 形成了 Markov 链。根据 Markov 性，未来双方的收入变化只取决于现在，而和过去运气无关。与之相关的一个现象是赌徒谬误，即认为过去的运气会影响未来的运气。例如，如果一个人连续输了很多次，那么他会认为自己未来运气会变好，赢的概率更大。但是，根据 Markov 性，过去的运气不会影响未来的运气，因此这种想法是错误的。“风水轮流转”在一场公平对赌中是不正确的认知。那么，如何评估赌局的公平性？

如果对赌是公平的，那么我们应该认为两个人每一轮的累计收入分布都是一样的，即

$$\Pr(X_n = i | X_0 = 0) = \Pr(X_n = -i | X_0 = 0).$$

因此，我们需要能够计算多步转移的概率。设 $p_{ij}^{(k)}$ 表示从状态 i 用 k 步转移到状态 j 的概率。 k 步转移概率形成了一个矩阵 $\mathcal{P}^{(k)}$ 。下面的定理给出了计算多步转移概率的方法。

定理 2.1 (Kolmogorov-Chapman 方程) $\mathcal{P}^{(k+l)} = \mathcal{P}^{(k)}\mathcal{P}^{(l)}$.

证明。 由 Markov 性、时齐性和全概率公式，

$$\begin{aligned} p_{ij}^{(k+l)} &= \Pr(X_{k+l} = j | X_0 = i) \\ &= \sum_{\alpha} \Pr(X_{k+l} = j, X_k = \alpha | X_0 = i) \\ &= \sum_{\alpha} \Pr(X_k = \alpha | X_0 = i) \Pr(X_{k+l} = j | X_k = \alpha) \\ &= \sum_{\alpha} p_{i\alpha}^{(k)} p_{\alpha j}^{(l)}. \end{aligned} \quad \square$$

Kolmogorov-Chapman 方程有两个重要的特例，前向方程： $\mathcal{P}^{(k+1)} = \mathcal{P}^{(k)}\mathcal{P}$ ，以及后向方程： $\mathcal{P}^{(l+1)} = \mathcal{P}\mathcal{P}^{(l)}$ 。见图 2.2 和图 2.3。

此外，利用归纳法，我们还有如下推论：

推论 2.1 $\mathcal{P}^{(k)} = \mathcal{P}^k$.

若已知初始分布向量为 λ ，利用这一推论，我们可以计算它随时间的演化：

$$\lambda^\top, \lambda^\top \mathcal{P}, \dots, \lambda^\top \mathcal{P}^n, \dots$$

[lhy: TODO: 重 画]

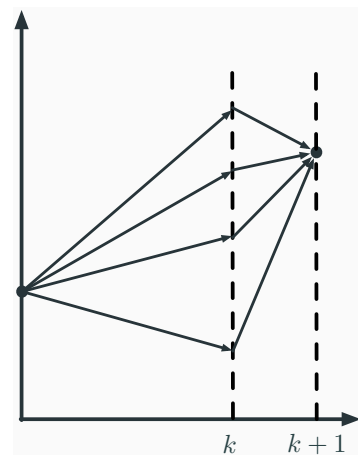


图 2.2: 前向方程 (往前一步)

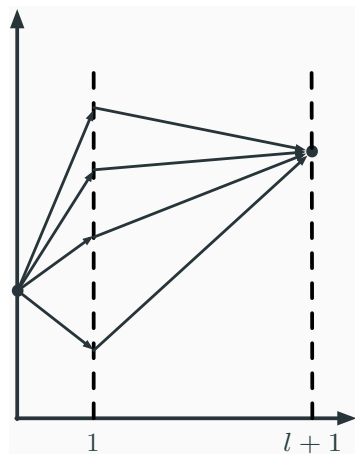


图 2.3: 后向方程 (往回一步)

回到赌徒模型，如何计算公平对赌中 X_n 的概率分布？我们先来看一个简化的例子。考虑只有两个状态 0,1，转移矩阵为

$$\mathcal{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}.$$

或者画成图 2.4 的形式。

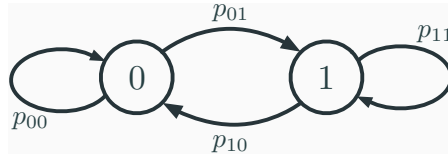


图 2.4: 只有两个状态的 Markov 链

可以归纳证明:

$$\mathcal{P}^n = \frac{1}{2 - p_{00} - p_{11}} \begin{pmatrix} 1 - p_{11} & 1 - p_{00} \\ 1 - p_{11} & 1 - p_{00} \end{pmatrix} + \frac{(p_{00} + p_{11} - 1)^n}{2 - p_{00} - p_{11}} \begin{pmatrix} 1 - p_{00} & -(1 - p_{00}) \\ -(1 - p_{11}) & 1 - p_{11} \end{pmatrix}.$$

假设 $|p_{00} + p_{11} - 1| < 1$, 等价地, p_{00} 和 p_{11} 不同时为 0 或同时为 1, 那么

- $\lim_{n \rightarrow \infty} p_{i0}^{(n)} = (1 - p_{11}) / (2 - p_{00} - p_{11}),$
- $\lim_{n \rightarrow \infty} p_{i1}^{(n)} = (1 - p_{00}) / (2 - p_{00} - p_{11}).$

因此, 无论初始分布是什么, 随着时间的推移, Markov 链的分布会收敛到一个同一个稳定的分布. 这个例子是否具有普遍性?

我们再考虑一个例子. 一个六元环 $\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$ 上的 Markov 链 X_k , 转移矩阵为 $p(i, i+1) = p(i, i-1) = 1/2$, 这里的表达式将 -1 和 5 等同, 6 和 0 等同. 如果初始分布集中在 0 上, 那么我们会发现以概率 1 有 $X_{2k} \in \{0, 2, 4\}$, $X_{2k+1} \in \{1, 3, 5\}$. 这种情况下, 最终不会趋于一个稳定的分布! 如果初始分布等概率地分布在 $\{0, 1\}$ 上, 最终会趋于一个 \mathbb{Z}_6 上的均匀分布. 因此, 并不是所有 Markov 链都具有这样的性质. 然而, 对于相当广泛的一类 Markov 链, 这一结论成立, 这就是遍历定理.

定理 2.2 (遍历定理) 设 Markov 链的状态空间为 $S = \{1, \dots, N\}$, 转移矩阵为 $\mathcal{P} = (p_{ij})$. 如果对于某一个 n_0 有

$$\min_{ij} p_{ij}^{(n_0)} > 0, \quad (2.1)$$

那么存在分布 $\lambda = (\lambda_1, \dots, \lambda_N)$ 使得

$$\lambda_i > 0, \quad \sum_i \lambda_i = 1, \quad (2.2)$$

并且对于每一个 $j \in S$ 和任意 $i \in S$ 都有

$$p_{ij}^{(n)} \rightarrow \lambda_j, \quad n \rightarrow \infty. \quad (2.3)$$

反之，如果存在满足 (2.2) 和 (2.3) 的 λ ，则存在满足 (2.1) 的 n_0 .

最后，在以上条件下，(2.2) 的 λ 满足

$$\lambda^\top = \lambda^\top \mathcal{P}. \quad (2.4)$$

条件 (2.1) 表明超过某个步数 n_0 之后，从 i 出发到达 j 的概率总是正的，这个条件被称为遍历。条件 (2.2) 表明每一个状态被访问到的概率都是正的，没有“死状态”。遍历定理表明遍历的 Markov 链从任何状态出发都是不可逆的，最终会把每个状态都走过一遍（遍历），变成一个混合均匀的状态。这可以用来解释物理学中的扩散现象，也是扩散模型的基础。

下面我们证明遍历定理。

证明。 首先证明从 (2.1) 到 (2.2) 和 (2.3) 的过程。定义序列

$$m_j^{(n)} = \min_i p_{ij}^{(n)}, \quad M_j^{(n)} = \max_i p_{ij}^{(n)}.$$

我们先证明这两个序列是单调的。由于

$$p_{ij}^{(n+1)} = \sum_{\alpha} p_{i\alpha} p_{\alpha j}^{(n)},$$

可见

$$m_j^{(n+1)} = \min_i p_{ij}^{(n+1)} \geq \min_i \sum_{\alpha} p_{i\alpha} \min_{\alpha} p_{\alpha j}^{(n)} = m_j^{(n)},$$

因此 $m_j^{(n)} \leq m_j^{(n+1)}$.

类似地 $M_j^{(n)} \geq M_j^{(n+1)}$.

接下来我们说明， $M_j^{(n)} - m_j^{(n)}$ 会趋于零。设 $\varepsilon = \min_{i,j} p_{ij}^{(n_0)} > 0$ ，由 Kolmogorov-Chapman 方程可得

$$\begin{aligned} p_{ij}^{(n_0+n)} &= \sum_{\alpha} p_{i\alpha}^{(n_0)} p_{\alpha j}^{(n)} = \sum_{\alpha} \left[p_{i\alpha}^{(n_0)} - \varepsilon p_{j\alpha}^{(n)} \right] p_{\alpha j}^{(n)} + \varepsilon \sum_{\alpha} p_{j\alpha}^{(n)} p_{\alpha j}^{(n)} \\ &= \sum_{\alpha} \left[p_{i\alpha}^{(n_0)} - \varepsilon p_{j\alpha}^{(n)} \right] p_{\alpha j}^{(n)} + \varepsilon p_{jj}^{(2n)} \end{aligned}$$

而由于 $p_{i\alpha}^{(n_0)} - \varepsilon p_{j\alpha}^{(n)} \geq p_{i\alpha}^{(n_0)} - \varepsilon \geq 0$ ，可见

$$p_{ij}^{(n_0+n)} \geq m_j^{(n)} \sum_{\alpha} \left[p_{i\alpha}^{(n_0)} - \varepsilon p_{j\alpha}^{(n)} \right] + \varepsilon p_{jj}^{(2n)} = m_j^{(n)} (1 - \varepsilon) + \varepsilon p_{jj}^{(2n)},$$

最后一个等式是因为概率求和为 1。由 i 的任意性，左边的不等式对所有 i 都成立，所以对最小的也成立：

$$m_j^{(n_0+n)} \geq m_j^{(n)} (1 - \varepsilon) + \varepsilon p_{jj}^{(2n)}.$$

同理，考虑 $M_j^{(n)}$ ，有

$$p_{ij}^{(n_0+n)} \leq M_j^{(n)} \sum_{\alpha} [p_{i\alpha}^{(n_0)} - \varepsilon p_{j\alpha}^{(n)}] + \varepsilon p_{jj}^{(2n)} = M_j^{(n)}(1 - \varepsilon) + \varepsilon p_{jj}^{(2n)},$$

类似可得

$$M_j^{(n_0+n)} \leq M_j^{(n)}(1 - \varepsilon) + \varepsilon p_{jj}^{(2n)}.$$

从而

$$M_j^{(n_0+n)} - m_j^{(n_0+n)} \leq (M_j^{(n)} - m_j^{(n)})(1 - \varepsilon),$$

说明当 $n \rightarrow \infty$ ， $M_j^{(n)} - m_j^{(n)} \rightarrow 0$ ， $M^{(n)}$ 和 $m^{(n)}$ 趋于同一个极限。

若记 $\pi_j = \lim_n m_j^{(n)}$ ，则

$$|p_{ij}^{(n)} - \pi_j| \leq M_j^{(n)} - m_j^{(n)} \leq (1 - \varepsilon)^{[n/n_0]-1},$$

即 $p_{ij}^{(n)}$ 以几何速度收敛于极限值 π_j 。

因为 $m_j^{(n)} \geq m_j^{(n_0)} \geq \varepsilon > 0, n \geq n_0$ ，所以 $\pi_j > 0$ 。这就推出了 (2.2) 和 (2.3)。

接下来，我们说明 (2.2) 和 (2.3) 可以推出 (2.1)。因为状态数有限，所以对任意充分小的 $\varepsilon > 0$ ，存在 n_0 使得对任意 i, j ，

$$|p_{ij}^{(n)} - \pi_j| \leq \varepsilon, n \geq n_0.$$

因此

$$p_{ij}^{(n)} \geq \pi_j - \varepsilon > 0, n \geq n_0.$$

最后我们说明 (2.3) 可以推出 (2.4)。注意到

$$\lim_{n \rightarrow \infty} \lambda^\top \mathcal{P}^n = \left(\sum_i \lambda_i \pi_1, \sum_i \lambda_i \pi_2, \dots, \sum_i \lambda_i \pi_N \right) = \lambda^\top.$$

等式两边同时右乘 \mathcal{P} ，左边的极限不变，右边变成 $\lambda^\top \mathcal{P}$ ，所以 $\lambda^\top \mathcal{P} = \lambda^\top$ 。 \square

满足条件 (2.4) 的分布被称为平稳分布。用性质 $\lambda^\top \mathcal{P} = \lambda^\top$ 很容易说明，平稳分布为初始状态时，Markov 链的演化与时间无关：

命题 2.3 设 $\{X_n\}$ 是 Markov 链，如果 X_0 是平稳分布，那么 (X_k, \dots, X_{k+l}) 的联合分布不依赖于 k 。

如果 Markov 链是遍历的，那么平稳分布是唯一的：

命题 2.4 设 $\{X_n\}$ 是遍历的 Markov 链，那么它有唯一平稳分布 μ 。

证明. 假设 μ 是另外一个平稳分布, 那么 $\mu_j = \sum_{\alpha} \mu_{\alpha} p_{\alpha j} = \cdots = \sum_{\alpha} \mu_{\alpha} p_{\alpha j}^{(n)}$. 因为 $p_{\alpha j}^{(n)} \rightarrow \lambda_j$, 所以 $\mu_j = \sum_{\alpha} (\mu_{\alpha} \lambda_j) = \lambda_j$. \square

非遍历 Markov 链也可能存在 (唯一) 平稳分布, 考虑如下转移矩阵:

$$\mathcal{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

它有唯一平稳分布 $\lambda = (1/2, 1/2)^T$.

§2.2 Markov 奖励过程 (MRP)

我们接下来的目标就是在 Markov 链上建立决策理论. 很多认知科学的研究证明, 人和动物在做决策的时候, 脑中会有一个价值系统, 用来评估每一个行动的可能产生的价值/奖励. 俗话说, 最美味的食物是饿了一整天之后的白米饭; 然而, 如果我们每一顿饭都是山珍海味, 就算是龙虾也会变得索然无味. 这说明, 我们的价值系统会随着时间和自身的状态而发生改变. 另一方面, 本章开篇所讲的故事表明, 我们的价值系统会对不同时期的奖励产生不同的反应, 一般来说, 我们会对即时奖励更加敏感.

总结上面的观察, 我们可以在 Markov 链上引入类似的价值系统, 这就是 Markov 奖励过程 (MRP). 在进入正式定义之前, 我们先来看一个例子.

例 2.3 (李二的 MRP) 在一个学期中, 学生李二可能处于几种状态: 在教室 1 中、刷手机、在教室 2 中、约会、睡觉、考试通过、考试挂科.

学生在不同的状态下会有不同的奖励, 例如, 李二总是进入教室 1 逼迫自己学习, 因为不情愿, 所以奖励是 -2 ; 但如果他被某个姑娘邀请去约会, 他会很激动, 所以奖励是 $+5$.

当处于某个状态时, 李二会有一定的概率转移到另一个状态. 例如, 在教室 1 中, 因为李二并不情愿学习, 所以他会有 0.5 的概率开始刷手机, 还有另外 0.5 的概率, 他发现差不多该去上课了, 于是进入了教室 2. 简化起见, 在状态转移中, 我们考虑抽象的时间单位, 对于李二来说, 时刻只会有 $t = 0, 1, 2, \dots$.

李二的人生就在这些状态之间循环往复, 当他进入某个状态之后, 他就会获得相应的奖励. 这个过程可以被图 2.5 描述. [lhy: 重画]

李二除了会获得即时奖励, 他还会对未来有预期. 例如, 尽管李二不愿意学习, 但是考试通过的奖励是 $+10$, 为了这么大的奖励, 现在遭罪一些是值得的, 所以他会愿意坐在教室 1 里自习. 假如下一刻李二就要考试, 这一刻他开始在教室 1 中自习, 他对于下一

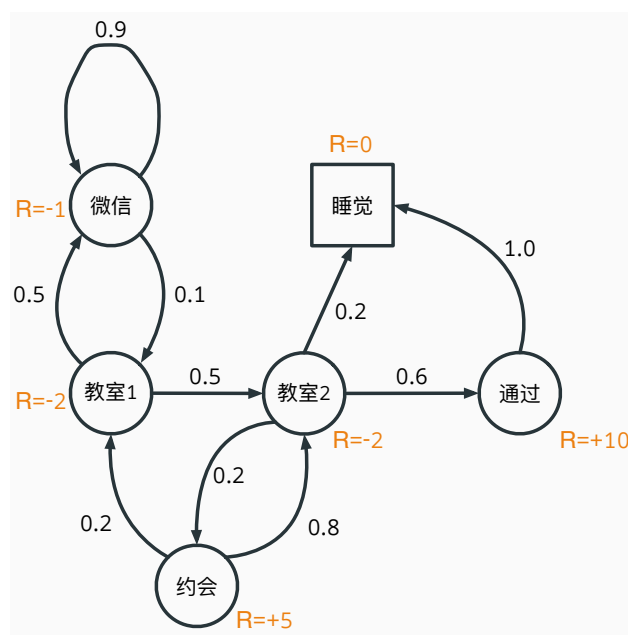


图 2.5: 学生 MRP

刻考试通过的奖励预期是 +9. 也就是说, 李二对未来的奖励会有一个折扣, 这个折扣系数就是 $\gamma = 9/10 = 0.9$. \square

更一般地, 我们可以形式上定义 MRP.

定义 2.2 (Markov 奖励过程, MRP) 一个 **Markov 奖励过程** (**Markov 奖励模型**, **MRP**) 由四元组 $\langle S, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 构成:

- S 是一个有穷的状态集合.
- \mathcal{P} 是一个状态转移矩阵, 从 i 转移到 j 的概率记为 $\mathcal{P}_{i,j}$. 根据这一转移矩阵可以产生一个状态转移的 Markov 链 $\{S_t\}_{t \geq 0}$.
- \mathcal{R} 是 (单步期望) 奖励函数, 定义为 $\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$, 其中, 随机变量 R_{t+1} 表示下一阶段所处状态的奖励. 也就是说, 当 t 时刻位于状态 s 时, \mathcal{R}_s 是下一时刻获得的奖励的期望.
- γ 是一个折扣系数, $\gamma \in [0, 1]$.

在 MRP 中, 我们最关心的并不是实时奖励, 而是综合来看整个过程的奖励. 为了描述这一点, 我们引入回报的概念.

定义 2.3 (回报) MRP 中, t 时刻以后的总回报 G_t 定义为

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.$$

从定义中可以看到, $\gamma \in [0, 1]$ 衡量了未来下一时段单位奖励在当前时刻的价值, γ 越小, 我们更偏好即时奖励, 因而更“短视”; γ 越大, 我们更偏好未来奖励, 因而更有“远见”.

我们这里对折扣做了极大的简化. 折扣系数和时间、状态都有可能存在依赖关系.

- 在李二的例子中, 他可能更愿意十天后有的一次约会, 而不是通过考试, 换言之, 约会对他的折扣系数可能是 0.99, 而考试通过的折扣系数可能是 0.9.
- 如果老师给李二布置的作业难度波动很大, 那么李二做完作业之后对通过考试的折扣系数就会发生变化: 作业简单, 李二觉得自己已经掌握了知识, 对通过考试的折扣系数就会降低; 作业困难, 李二觉得自己可能很难通过考试, 对通过考试的折扣系数就会提高.

然而, 在 MRP 的定义中, 我们假设了一个固定的折扣系数, 并且规定了一个简单的方法计算折扣: t 时刻后的奖励是即时奖励的 γ^t 倍. 这一定义体现了折衷的思想: 如果折扣系数太符合实际, 那么这个模型就不太实用. 在我们的定义之下, 随着时间改变, 折扣系数会指数衰减, 这不仅和实验结果比较符合, 而且也使得 Markov 性能被很好利用.

除了回报之外, 在 MRP 中, 我们更关心的是价值函数, 即处于某个状态时候预期的回报是多少:

定义 2.4 (价值函数) 在 MRP 中, 状态-价值函数 (或价值函数) 定义为

$$v(s) = \mathbb{E}[G_t | S_t = s].$$

注意, 等式右边有 t 但左边没有, 所以我们需要说明这个定义对任意 t 都是成立的. 我们只需要说明对于任意的 t , $t+1$ 和 t 定义了同一个 $v(s)$. 注意, 随机变量 R_{t+k} 只依赖于 S_{t+k} , 即 $R_{t+k} = R(S_{t+k})$, 所以

$$\begin{aligned} \mathbb{E}[G_t | S_t = s] &= \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right] \\ &= \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k R(S_{t+k+1}) \middle| S_t = s \right] \\ &= \sum_{k=0}^{\infty} \gamma^k \mathbb{E}[R(S_{t+k+1}) | S_t = s]. \end{aligned}$$

注意，项 $\mathbb{E}[R(S_{t+k+1})|S_t = s]$ 的定义满足 Markov 性，即

$$\mathbb{E}[R(S_{t+k+1})|S_t = s] = \mathbb{E}[R(S_{t+k+2})|S_{t+1} = s].$$

因而对于任意的 t , $t+1$ 和 t 定义了同一个 $v(s)$ 。

因此，这一定义蕴含了 Markov 性：只从当前起考虑未来收益，不考虑历史收益（沉没成本）的影响；也蕴含了时齐性：价值函数的定义不依赖于时刻 t （无穷阶段情形）。我们在后面要各种相关概念的定义都需要用到这个性质，证明都是类似的，不再赘述。

接下来我们展示价值函数的计算方法。直观上说，回报应该被分解为两部分：即时回报 R_{t+1} 以及未来的回报 $\gamma v(S_{t+1})$ ，也就是下一个状态期望回报再做折扣。具体来说，我们有

$$\begin{aligned} v(s) &= \mathbb{E}(G_t | S_t = s) \\ &= \mathbb{E}(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s) \\ &= \mathbb{E}(R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots) | S_t = s) \\ &= \mathbb{E}(R_{t+1} + \gamma G_{t+1} | S_t = s) \\ &= \mathbb{E}(R_{t+1} | S_t = s) + \gamma \mathbb{E}(G_{t+1} | S_t = s) \\ &= \mathbb{E}(R_{t+1} | S_t = s) + \mathbb{E}[\gamma v(S_{t+1}) | S_t = s] \\ &= \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'} v(s'). \end{aligned}$$

其中倒数第二行是因为

$$\begin{aligned} \mathbb{E}(G_{t+1} | S_t = s) &= \sum_{s' \in \mathcal{S}} \mathbb{E}(G_{t+1} | S_{t+1} = s', S_t = s) \Pr(S_{t+1} = s' | S_t = s) \\ &= \sum_{s' \in \mathcal{S}} \mathbb{E}(G_{t+1} | S_{t+1} = s') \Pr(S_{t+1} = s' | S_t = s) \\ &= \sum_{s' \in \mathcal{S}} v(s') \Pr(S_{t+1} = s' | S_t = s) \\ &= \mathbb{E}[v(S_{t+1}) | S_t = s]. \end{aligned}$$

我们因此得到了 **Bellman 方程**：

定理 2.3 (Bellman 方程) $v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'} v(s')$ 。

在后面，我们将不断看到这样形式的方程，它将一个和 Markov 链有关的计算分解为当前的部分和未来的部分。我们将这样形式的方程统称 Bellman 方程。

Bellman 方程可以用矩阵形式表达：

$$v = \mathcal{R} + \gamma \mathcal{P}v.$$

这里 v 是列向量 $v = (v(s))_{s \in \mathcal{S}}$.

写成矩阵形式之后，我们可以看到，Bellman 方程其实是一个线性方程，可以被直接解：

$$v = \mathcal{R} + \gamma \mathcal{P}v \implies (I - \gamma \mathcal{P})v = \mathcal{R} \implies v = (I - \gamma \mathcal{P})^{-1} \mathcal{R}.$$

我们还可以有另一种观点，考虑一个映射 $f: \mathcal{S} \rightarrow \mathcal{S}$, $f(v) = \mathcal{R} + \gamma \mathcal{P}v$ ，那么 Bellman 方程可以被写作

$$v = f(v).$$

因此， v 是 f 的不动点。在第八章，我们将系统地讨论不动点的性质以及它对于 Markov 链相关模型的重要性。

回到解 Bellman 方程，对于 n 个状态的 Markov 链，用线性方程组法的计算复杂度为 $\mathcal{O}(n^3)$ 。对于较小的 MRP 可以直接解，太大的 MRP 开销太大。对于大型 MRP，可以采用迭代算法，例如：

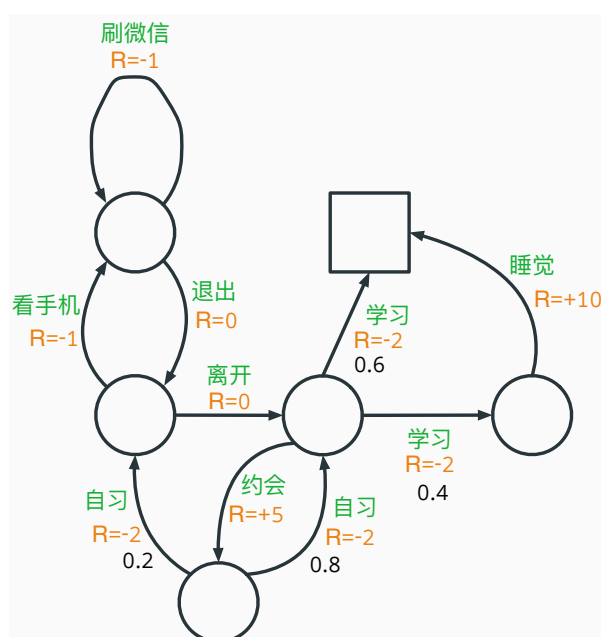
- 动态规划
- Monte-Carlo 评估
- 时序差分学习

动态规划法的思路我们将在第 2.4 节中介绍。

§2.3 Markov 决策过程 (MDP)

上一节中我们建模了 Markov 链上的价值系统，即 MRP，接下来我们进入建模决策的环节。回顾李二的例子（例 2.3），我们在 MRP 中忽略了一个非常重要的考虑：当李二坐在教室 1 中的时候，李二不是随机地开始刷手机的，他需要选择开始刷手机。换言之，在 MRP 中，在当前状态可以做什么行动是缺失的。

此外，李二的奖励其实不是由状态决定的，而是由他做了什么行动决定的，这样，李二才能评估做什么行动可以最大化自己的奖励，然后选择奖励高的那个行动。实际上，认知神经科学的研究表明，人体的运动控制就是由类似的机制完成的：首先，大脑皮层会提出若干不同的运动计划，这些运动计划对应了不同的奖励（由多巴胺浓度来表征），人



[lhy: 重画]

图 2.6: 学生 MDP

脑中有一个被称作基底神经节的结构，它负责“放行”奖励高于某个阈值的运动计划，于是人就可以产生这个动作了。

综合以上这些思考，我们就可以给出 *Markov* 决策过程 (MDP) 的模型。我们还是先看李二的例子，然后再推广到一般的情况。

例 2.4 (李二 MDP) 李二的 MDP 见图 2.6。在图中，状态依然是之前状态，但是状态的转移被给上了两个标签：(1) 这个转移是什么动作引起的（紫色），(2) 这个动作可以带来多少的奖励（蓝色）。例如，当李二在教室 1 的时候，她如果选择看手机，那么就会有 -1 的奖励，并且进入刷手机的状态。

接下来，我们给出一般的 MDP 的定义。

定义 2.5 (Markov 决策过程, MDP) *Markov* 决策过程 (MDP) 是一个 MDP 是五元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ ，其中

- \mathcal{S} 是一个有限的状态集合。
- \mathcal{A} 是一个有限的行动 (action) 集合。
- \mathcal{P} 是状态转移概率矩阵，

$$\mathcal{P}_{ss'}^a = \Pr(S_{t+1} = s' | S_t = s, A_t = a).$$

- \mathcal{R} 是一个奖励函数, $\mathcal{R}_s^a = \mathbb{E}(R_{t+1} | S_t = s, A_t = a)$, 随机变量 R_{t+1} 是进行某一行动到达某一状态后的奖励.
- γ 是一个折扣系数 $\gamma \in [0, 1]$.

现在我们对 MDP 和 MRP. MDP 中, 状态转移矩阵依赖动作, 奖励函数也依赖动作. 在李二的例子中, 在教室 2 如果选择学习, 尽管都是在学习, 但是学习并不一定会产生一个确定的结果: 他会有 0.4 的概率睡着, 0.6 的概率继续学习. 自然, 睡着的奖励和继续学习的奖励是不同的.

定义 2.6 (策略) 一个策略 π 是给定状态下行动的分布, 即

$$\pi(a|s) = \Pr(A_t = a | S_t = s).$$

一个策略完全决定了一个智能体在 MDP 环境中的行为. 它的定义蕴含着 Markov 性: MDP 的策略取决于当前状态, 而非历史状态; 也蕴含着时齐性: MDP 的策略不依赖于时刻 t . 这样的定义会方便我们讨论价值函数以及决策的问题.

MDP 与 MDP 的关系由策略给出.

命题 2.5 给定一个 MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ 和一个策略 π , $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$ 是一个 Markov 链, $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$ 是一个 MRP, 其中

$$\begin{aligned} \mathcal{P}_{s,s'}^\pi &= \mathbb{E}_{a \sim \pi(\cdot|s)}(\mathcal{P}_{s,s'}^a) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_{s,s'}^a, \\ \mathcal{R}_s^\pi &= \mathbb{E}_{a \sim \pi(\cdot|s)}(\mathcal{R}_s^a) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_s^a. \end{aligned}$$

证明. 对行动利用全概率公式. □

现在我们有三个概念, MDP, MRP, 以及 Markov 链, 命题 2.6 给了他们三者的关系, 我们可以总结到图 2.7.



图 2.7: MDP, MRP 和 Markov 链的关系

对于李二来说, 选择什么样的策略很大程度取决于他能从中获得多少奖励. 同样, 在 MDP 中, 我们可以定义回报的概念.

定义 2.7 (回报) 在 MDP 中, t 时刻以后的总回报 G_t 定义为

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.$$

类似 MRP, 我们需要定义相应的价值函数. 在 MDP 中, 状态-价值函数和行动-价值函数是两个重要的价值函数, 它们分别描述了从某一状态出发, 遵从某一策略的期望回报.

定义 2.8 (价值函数) 状态-价值函数 $v_{\pi}(s)$ 是从状态 s 出发, 遵从策略 π 的期望回报

$$v_{\pi}(s) = \mathbb{E}_{\pi}(G_t | S_t = s).$$

行动-价值函数 $q_{\pi}(s, a)$ 是从状态 s 出发, 采取行动 a , 遵从策略 π 的期望回报

$$q_{\pi}(s, a) = \mathbb{E}_{\pi}(G_t | S_t = s, A_t = a).$$

注意, 类似 MRP 中的价值函数, MDP 中的价值函数定义也不依赖 t 的选择, 这是因为 MDP 的 Markov 性和时齐性.

行动-价值函数比起状态-价值函数更加具体, 它可以帮助李二评判在当前选择每一个行动的回报, 从而选择最优的行动. 而状态-价值函数则是对行动-价值函数的一个期望, 因而是李二预期他从这个状态出发的回报. 具体来说, 这两个价值函数有如下关系:

命题 2.6 状态-价值函数 $v_{\pi}(s)$ 和行动-价值函数 $q_{\pi}(s, a)$ 之间有如下关系:

$$v_{\pi}(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}(q_{\pi}(s, a)) = \sum_{a \in \mathcal{A}} \pi(a|s) q_{\pi}(s, a),$$

$$q_{\pi}(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{s,s'}^a v_{\pi}(s').$$

证明. 用 $q_{\pi}(s, a)$ 来表示 $v_{\pi}(s)$, 对行动 a 用全概率公式即可得到.

另一方面, 用 $v_{\pi}(s)$ 来表示 $q_{\pi}(s, a)$ 也是全概率公式. 具体来说, 在状态 s 采取行动 a 之后, 期望上有 \mathcal{R}_s^a 的即时奖励, 然后以 $\mathcal{P}_{s,s'}^a$ 的概率转移到状态 s' , 在状态 s' 的期望回报是 $\gamma v_{\pi}(s')$. 按照 s' 用全概率公式即可得到 $q_{\pi}(s, a)$. \square

下面我们给出状态-价值函数和行动-价值函数的 Bellman 方程. 首先, 价值函数可以被分解为即时回报加未来的折扣回报, 具体来说

- 状态-价值函数可以被分解为:

$$v_{\pi}(s) = \mathbb{E}_{\pi}[\mathcal{R}_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = s].$$

- 行动-价值函数可以被类似地分解,

$$q_{\pi}(s, a) = \mathbb{E}_{\pi}[\mathcal{R}_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a].$$

他们的计算方法和定理 2.3 中的方法类似. 继续仿照定理 2.3 的证明, 我们可以得到 MDP 的 Bellman 方程:

定理 2.4 (Bellman 方程)

$$\begin{aligned} v_{\pi}(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a v_{\pi}(s') \right), \\ q_{\pi}(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_{\pi}(s', a'). \end{aligned}$$

状态-价值函数的 Bellman 方程可以被写成矩阵形式:

$$v_{\pi} = \mathcal{R}^{\pi} + \gamma \mathcal{P}^{\pi} v_{\pi} = (I - \gamma \mathcal{P}^{\pi})^{-1} \mathcal{R}^{\pi}.$$

注意, MDP 的 Bellman 方程只能告诉我们给定策略 π 的价值函数, 它并不能告诉智能体 (也就是李二) 要如何行动, 所以, 接下来我们要讨论最优策略和最优价值函数.

定义 2.9 (最优价值函数) 最优状态-价值函数 $v_{\star}(s)$ 是所有策略中最大的状态-价值函数

$$v_{\star}(s) = \max_{\pi} v_{\pi}(s).$$

最优行动-价值函数 $q_{\star}(s, a)$ 是所有策略中最大的行动-价值函数

$$q_{\star}(s, a) = \max_{\pi} q_{\pi}(s, a).$$

相应地, 最优价值函数确定了智能体在 MDP 中的最佳收益, 解 MDP 即确定达到最优价值函数的策略.

在以上定义中可能会遇到这样的问题: 对两个状态 s_1 和 s_2 , 存在两个不同的策略 π_1 和 π_2 , 使得 $v_{\star}(s_1) = v_{\pi_1}(s_1)$, $v_{\star}(s_2) = v_{\pi_2}(s_2)$. 此时, 每个状态取到最大价值的策略 π 可能并不是同一个, 因此 v_{\star} 并不是某个特定策略可以实现的值. 所以, 我们需要证明, 存在一个策略 π_{\star} , 使得对于任意的策略 π , π_{\star} 都取得最大价值函数.

我们有如下定理, 说明了这样策略的存在性, 因而也证明了 MDP 解的存在性.

定理 2.5 (MDP 解的存在性) 对任意 MDP, 存在一个策略 π_{\star} ,

- 对任意状态 s , $v_{\pi_{\star}}(s) = v_{\star}(s)$.

- 对任意状态 s 和行动 a , $q_{\pi_*}(s, a) = q_*(s, a)$.

证明. 我们给出一个构造性证明, 即找出最优策略. 我们先找到一个 π_* 最大化 q , 然后说明这个 π_* 也可以最大化 v .

直观上说, 我们只要对每个状态都选择最好的行动, 这就是一个最优策略. 具体来说, 我们可以通过如下步骤找到 π_* :

- 固定 s ,
- 找到一个 a_* 最大化 $q_*(s, \cdot)$, 即 $q_*(s, a_*) = \max_a q_*(s, a)$, 令 $\pi_*(a_*|s) = 1$.
- 对其他 $a \neq a_*$, 令 $\pi_*(a|s) = 0$.

首先, 根据选法, π_* 取得最优行动-价值函数. 接下来我们说明, 它也取得了最优状态-价值函数. 任意策略 π , 给定状态 s , 我们有如下计算:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_{a \sim \pi(\cdot|s)}[q_\pi(s, a)] \\ &\leq \mathbb{E}_{a \sim \pi(\cdot|s)}[q_*(s, a)] \\ &\leq q_*(s, a_*) \\ &= v_{\pi_*}(s). \end{aligned} \quad \square$$

这个证明还有一个推论:

推论 2.2 对任意 MDP, 总存在一个非随机的最优策略, 即对任意状态 s , $\pi_*(a|s) \in \{0, 1\}$.

两个最优价值函数之间有如下关系:

命题 2.7

$$\begin{aligned} v_*(s) &= \max_a q_*(s, a), \\ q_*(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s, s'}^a v_*(s'). \end{aligned}$$

证明. 在定理 2.5 的证明中, 我们将 π 取为 π_* , 于是有

$$v_*(s) = v_{\pi_*}(s) = q_*(s, a_*) = \max_a q_*(s, a).$$

对于第二个等式, 根据定理 2.5,

$$v_*(s) = v_{\pi_*}(s), \quad q_*(s, a) = q_{\pi_*}(s, a),$$

在命题 2.6 取 $\pi = \pi_*$ 即可得到. \square

根据上面结论, 如果我们知道 $q_*(s, a)$ 或者 $v_*(s)$, 我们就能获得最优策略. 这一计算同样依赖 Bellman 方程¹:

定理 2.6 (Bellman 方程)

$$v_*(s) = \max_a \left\{ \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a v_*(s') \right\},$$

$$q_*(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \max_{a'} q_*(s', a').$$

证明. 对于第一个方程, 将命题 2.7 中第二个等式代入第一个等式即可得到.

对于第二个方程, 将命题 2.7 中第一个等式代入第二个等式即可得到. □

Bellman 不是线性的. 因此很难有解析形式的解. 但是 MDP 的数值解是可以多项式时间求出来的. 我们一般采用迭代算法求解:

- 价值迭代
- 策略迭代
- Q-learning
- Sarsa

求解 MDP 的过程实际上就是人工智能中强化学习的核心步骤. 在本节的开头, 我们说过人脑运动控制机制. 实际上, 绝大部分的哺乳动物都具备类似这样的学习机制: 通过奖励的反馈, 动物可以学会如何行动, 从而获得最大的奖励. 这样的学习机制被称为强化学习. 强化学习在很多复杂交互的环境中都有广泛的应用, 例如围棋的 AlphaGo、星际争霸的 AlphaStar 等.

注. Bellman 方程是强化学习、经济学动态优化的核心. Bellman 方程的推导是 Markov 链中最为常用的技巧: 考虑从当前状态转移到下一状态, 利用全概率公式, 一步转移会将两个状态之间的概率 (期望) 用递推公式联系起来. 在随机过程中, 有大量这样的例子: 前向方程、Wald 等式、调和函数. 后面的 HMM 也是类似的例子.

最后, 我们谈谈深度强化学习. 在一些非常复杂的情况下 (例如下围棋), 使用经典的迭代算法并不容易求解 MDP. 深度强化学习是一种结合了深度学习和强化学习的方法. 在深度强化学习中, 用神经网络来表示 π 和 v , 之后, 用某种学习算法训练神经网络.

¹在文献中, 这一 Bellman 方程被称为 Bellman 最优性方程, 而之前推导的 Bellman 方程被称为 Bellman 期望方程. 为了不过度引入术语, 我们这里不做这种区分.

在一些深度强化学习模型中（例如 AlphaZero），状态空间 \mathcal{S} 也用一个神经网络表示。用神经网络来表示状态空间的好处是可以减少人类的特征工程，让神经网络充分发掘状态空间好的表示方法。

在很多深度强化学习模型中（例如 AlphaGo），MDP 的策略是基于过去 k 期的状态做当前的决策，这样可以更好地利用状态的历史信息。这样的决策模型等价于一个利用一期信息决策的 MDP（见习题[[lhy: 出一下](#)]），因此依然可以用同样的深度学习算法来求解。

§2.4 隐 Markov 模型（HMM）

在本节，我们考虑 Markov 链上的另一种应用。在统计学和机器学习中，我们有时候要处理一类含时间的数据。例如，如果我们希望利用机器的力量帮助我们炒股赚钱，就要考虑如何预测股价然后做出相应的决策，这样的投资模式被称为量化投资。在 1989 年到 2009 年间，量化界的传奇人物 James Simons 操盘大奖章基金，平均年回报率高达 35%，即便是在次贷危机爆发的 2007 年，该基金的回报率仍高达 85%。据说，让 Simons 成功的秘诀是隐 Markov 模型，这正是本节的主题。

我们先给出隐 Markov 模型的定义。

定义 2.10 (隐 Markov 模型, HMM) 一个隐 Markov 模型 (HMM) 是两列随机变量（被称为观测序列） X_1, X_2, \dots ，和 z_1, z_2, \dots ，（被称为隐状态序列）的序列，满足：

- $\{Z_t\}$ 构成一条 Markov 链。
- 对任意 t ， X_t 的分布仅依赖于 Z_t 。
- 对于任意 t ， $\Pr(X_t|Z_t)$ 服从分布 $F(Z_t)$ 。

示意图图 2.8。

为了理解这一概念，我们可以考虑炒股的例子。

例 2.5 (美为 HMM) 假设我们要投资美为的股票，第 t 天的股票价格是 X_t 。我们很希望理解整个 X_t 的变化趋势。然而，美为股价的背后有一个神秘势力操控。我们并不清楚这个神秘势力每天决策的具体细节，只知道他们的决策非常健忘，即他们的决策只依赖于前一天的决策。他们会决定一个明天的预计股价 Z_t ，这构成了一个 Markov 链 $\{Z_t\}$ 。

因为市场和汇率的波动，这个神秘势力无法完全决定股票的价格 X_t ，然而，他们每一天所决定的预期价格 Z_t 会导致股票价格的变化，我们可以认为 X_t 仅依赖于 Z_t ，但依

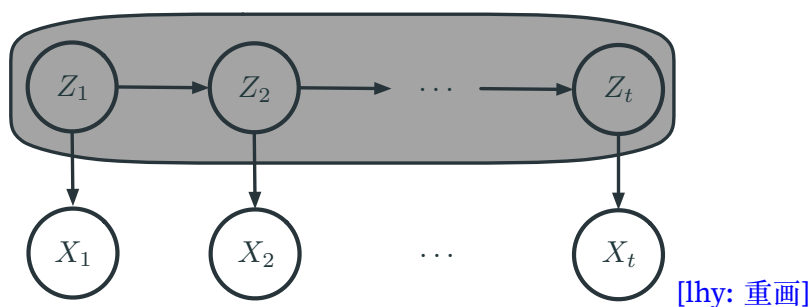


图 2.8: 隐 Markov 模型

然受到一些随机因素的影响. 作为简化, 我们假设每天的随机影响是独立且一致的, 服从分布 $F(Z_t)$. 这样的得到的模型就是一个 HMM. \square

接下来, 我们给 HMM 引入一些记号. 为简便起见, 我们只研究有限离散的 HMM. 设 HMM \mathcal{M} 所形成的概率测度是 $\Pr_{\mathcal{M}}$. 我们假设模型中以下的量都是已知的:

- \mathcal{Z} : 有限的状态集合, 即 Z_t 的取值集合.
- \mathcal{X} : 有限的观测集合, 即 X_t 的取值集合.
- T : Markov 链 $\{Z_t\}$ 的转移矩阵, $T_{i,j} = \Pr_{\mathcal{M}}(Z_{t+1} = j | Z_t = i)$.
- M : 给定隐状态时的观测概率, $M_{i,k} = \Pr(X_t = k | Z_t = i)$.
- λ : 隐状态的初始分布.

我们会把序列 A_i, \dots, A_j 记作 $A_{i:j}$, 然后将它看作一个向量. 例如 $X = X_{1:t}$ 就是描述从时刻 1 到 t 的观测序列的随机向量, 而 $Z = Z_{1:t}$ 就是描述从时刻 1 到 t 的隐状态序列的随机向量.

为了理解 HMM 的任务, 我们接着看美的例子. 假设我们已经知道一个 HMM \mathcal{M} 可以预测美的股价 X_t , 作为一个使用者, 我们希望 \mathcal{M} 确实有用. 因此, 我们要评估这个模型的表现. 具体来说, 我们希望知道, 给定观测历史 $x = (x_1, x_2, \dots, x_t)$, 如何计算 $\Pr_{\mathcal{M}}(X = x)$. 这一方法也可以用来做预测: 利用全概率公式, 我们可以计算 $\Pr_{\mathcal{M}}(X_{t+1} = x_{t+1} | X = x)$, 从而预测未来的股价.

评估一个 HMM 固然重要, 但是评估依然是把 HMM 当成一个黑盒来使用. 我们还希望知道这个模型背后到底发生了什么, 这就是解释问题. 具体来说, 我们希望知道, 给定观测历史 $x = (x_1, x_2, \dots, x_t)$ 以及一个时刻 k , 如何计算 $\Pr_{\mathcal{M}}(Z_k | X = x)$. 这一分布表明了, 在 k 时刻神秘势力更有可能做了什么样的决策, 从而帮助我们更好理解股价的波动.

接下来我们将分别阐述如何解决评估问题和解释问题，因为我们只讨论某一个具体的 HMM，为简便起见，我们此后都将概率测度 $\Pr_{\mathcal{M}}$ 简记为 \Pr 。

2.4.1 评估问题

我们引入记号随机向量 $X = (X_1, \dots, X_t)$, $Z = (Z_1, \dots, Z_t)$ 。我们考虑 HMM 的评估问题：给定一个 HMM \mathcal{M} ，以及它的观测历史 $x = (x_1, x_2, \dots, x_t)$ ，计算 $\Pr(X = x)$ 。

关键困难是我们不知道隐状态历史 $Z = (z_1, z_2, \dots, z_t)$ ，因此我们需要利用全概率公式将隐状态消除掉，即：

$$\Pr(X = x) = \sum_{Z=(z_1, \dots, z_t) \in \mathcal{Z}^t} \Pr(X = x | Z = z) \Pr(Z = z).$$

接下来我们分别计算 $\Pr(X = x | Z = z)$ 和 $\Pr(Z = z)$ 。对于前者，因为每一个观测值 X_i 仅依赖于 Z_i ，我们有

$$\Pr(X = x | Z = z) = \prod_{i=1}^t \Pr(X_i = x_i | Z_i = z_i) = M_{z_1, x_1} \cdot M_{z_2, x_2} \dots M_{z_t, x_t},$$

对于后者，因为 Z 是一个 Markov 链，我们有

$$\begin{aligned} \Pr(Z = z) &= \Pr(Z_1 = z_1) \prod_{i=2}^t \Pr(Z_i = z_i | Z_{i-1} = z_{i-1}) \\ &= \lambda_{z_1} \cdot T_{z_1, z_2} \cdot T_{z_2, z_3} \dots T_{z_{t-1}, z_t}. \end{aligned}$$

以上的量都是已知的，所以我们已经可以计算评估问题了。然而，这一方法的需要计算的乘法次数是 $\mathcal{O}(t|\mathcal{Z}|^t)$ ，对于很大的 t 和 \mathcal{Z} ，这是不可接受的计算量，我们需要更好的计算方法。

接下来，我们采用前向方程（见定理 2.1）的思路，从前 k 步的结果推出前 $k+1$ 步的结果，然后据此列出递推方程。在第 $k+1$ 步，Markov 链的状态发生了转移，按照从哪个状态转移到了哪个状态，我们可以拆分概率：

$$\begin{aligned} &\Pr(X_{1:k+1} = x_{1:k+1}) \\ &= \sum_{z \in \mathcal{Z}} \Pr(X_{1:k} = x_{1:k}, Z_k = z) \Pr(X_{k+1} = x_{k+1} | Z_k = z) \\ &= \sum_{z \in \mathcal{Z}} \Pr(X_{1:k} = x_{1:k}, Z_k = z) \sum_{z' \in \mathcal{Z}} \Pr(Z_{k+1} = z' | Z_k = z) \Pr(X_{k+1} = x_{k+1} | Z_{k+1} = z') \\ &= \sum_{z \in \mathcal{Z}} \Pr(X_{1:k} = x_{1:k}, Z_k = z) \sum_{z' \in \mathcal{Z}} T_{z, z'} M_{z', x_{k+1}}. \end{aligned}$$

如果把左边按照 Z_{k+1} 拆分, 我们有

$$\sum_{z \in \mathcal{Z}} \Pr(X_{1:k+1} = x_{1:k+1}, Z_{k+1} = z) = \sum_{z \in \mathcal{Z}} \Pr(X_{1:k} = x_{1:k}, Z_k = z) \sum_{z' \in \mathcal{Z}} T_{z,z'} M_{z', x_{k+1}}.$$

如果令 $\alpha_k(z) := \Pr(X_{1:k} = x_{1:k}, Z_k = z)$, 用类似的计算, 我们有递推:

- 当 $k = 1$, $\alpha_k(z) = \lambda(z) M_{z, x_k}$.
- 当 $k > 1$, $\alpha_{k+1}(z) = \sum_{z' \in \mathcal{Z}} \alpha_k(z') T_{z', z} M_{z, x_{k+1}}$.

最后, $\Pr(X = x) = \sum_{z \in \mathcal{Z}} \alpha_t(z)$. 这一方法需要算的乘法次数是 $\mathcal{O}(t|\mathcal{Z}|^2)$, 比前一种计算方法要快很多.

镜像地, 我们可以使用后向方程的思路, 从前 $k+1$ 步的结果推出前 k 步的结果. 同样可以列出递推方程. 定义 $\beta_k(z) := \Pr(X_{k+1:t} = x_{k+1:t} | Z_k = z)$, 我们有递推:

- 当 $k = t$, $\beta_k(z) = 1$.
- 当 $1 \leq k < t$, $\beta_k(z) = \sum_{z' \in \mathcal{Z}} T_{z, z'} M_{z', x_{k+1}} \beta_{k+1}(z')$.

于是, $\Pr(X = x) = \sum_{z \in \mathcal{Z}} \lambda(z) M_{z, x_1} \beta_1(z)$. 这一方法需要算的乘法次数是 $\mathcal{O}(t|\mathcal{Z}|^2)$.

2.4.2 解释问题

接下来我们讨论 HMM 的解释问题. 给定一个 HMM $\mathcal{M} = (\mathcal{Z}, \mathcal{X}, T, M, \lambda)$, 一列观测历史 $x = (x_1, x_2, \dots, x_t)$, 解释问题旨在寻找一个状态序列, 能最好地解释这些历史观察. 具体来说我们考虑如下四个问题:

1. 过滤: 计算 $\Pr(Z_k = s | X_{1:k} = x_{1:k})$.
2. 平滑: 计算 $\Pr(Z_k = s | X = x)$, $k < t$.
3. 预测: 计算 $\Pr(Z_k = s | X = x)$, $k > t$.
4. 解码: 找到最有可能的状态序列 $z = (z_1, z_2, \dots, z_t)$.

首先考虑过滤: $\Pr(Z_k = s | X_{1:k} = x_{1:k})$. 回顾记号 $\alpha_k(s) = \Pr(X_{1:k} = x_{1:k}, Z_k = s)$, 这其实已经足够我们计算过滤了. 根据条件概率的定义, 我们有

$$\begin{aligned} \Pr(Z_k = s | X_{1:k} = x_{1:k}) &= \frac{\Pr(X_{1:k} = x_{1:k}, Z_k = s)}{\Pr(X_{1:k} = x_{1:k})} \\ &= \frac{\alpha_k(s)}{\sum_{z \in \mathcal{Z}} \alpha_k(z)}. \end{aligned}$$

我们已经知道如何计算 $\alpha_k(s)$ ，所以这就可以用来计算过滤了。

然后是平滑： $\Pr(Z_k = s|X = x)$ ， $k < t$ 。回顾记号 $\alpha_k(s) = \Pr(X_{1:k} = x_{1:k}, Z_k = s)$ ，以及 $\beta_k(s) = \Pr(X_{k+1:t} = x_{k+1:t}|Z_k = s)$ 。可以证明（见习题[thy: 出一下]）：

$$\Pr(Z_k = s|X = x) = \frac{\beta_k(s)\alpha_k(s)}{\sum_{z \in \mathcal{Z}} \alpha_t(z)}. \quad (2.5)$$

同样，我们已经知道如何计算 $\alpha_k(s)$ 和 $\beta_k(s)$ ，所以这就可以用来计算平滑了。

之后是预测： $\Pr(Z_k = s|X = x)$ ， $k > t$ 。首先用过滤计算 $\lambda = \Pr(Z_t = s|X = x)$ 。从 t 之后的隐状态都只依赖于 t 时刻的隐状态 Z_t ，因此，条件在 $X = x$ 下， Z_t, Z_{t+1}, \dots, Z_k 构成了一个 Markov 链，它的初始分布为 λ ，转移矩阵为 T 。于是我们利用定理 2.1 来计算该 Markov 链第 $k - t$ 步的分布。

最后是解码：求 $z = (z_1, z_2, \dots, z_t)$ ，使得 $\Pr(Z = z|X = x)$ 最大。注意，这一概率最大等价于 $\Pr(Z = z, X = x)$ 最大。我们也使用递归的想法来解决这个问题。同样，在前 $k - 1$ 个状态已经选好之后，考虑最后一个状态应该选哪个。具体来说，定义

$$\delta_k(s) = \max_{z_{1:k-1}} \Pr(Z_{1:k} = (z_{1:k-1}, s), X_{1:k} = x_{1:k}).$$

于是

$$\begin{aligned} & \delta_{k+1}(s) \\ &= \max_{z_{1:k}} \Pr(Z_{1:k+1} = (z_{1:k}, s), X_{1:k+1} = x_{1:k+1}) \\ &= \max_{z_{1:k}} \{\Pr(Z_{1:k} = z_{1:k}, X_{1:k} = x_{1:k}) \Pr(Z_{k+1} = s|Z_k = z_k) \Pr(X_{k+1} = x_{k+1}|Z_{k+1} = s)\} \\ &= \max_q \{\max_{z_{1:k-1}} \Pr(Z_{1:k} = (z_{1:k-1}, q), X_{1:k} = x_{1:k}) T_{q,s}\} M_{s, x_{k+1}} \\ &= \max_q \{\delta_k(q) T_{q,s}\} M_{s, x_{k+1}}. \end{aligned}$$

这就给出了从 k 推导到 $k + 1$ 的递推方程。这一递推的初始状态是

$$\delta_1(s) = \Pr(Z_1 = s, X_1 = x_1) = \Pr(Z_1 = s) \Pr(X_1 = x_1|Z_1 = s) = \lambda(s) M_{s, x_1}.$$

利用这一递推，我们就可以解决解码问题了。具体算法在下面给出。

1. 利用递推公式，逐层用第 k 层的 δ_k 计算第 $k + 1$ 层的 δ_{k+1} ，最后得到 δ_t 。
2. 求一个 z_t^* 使得 $\delta_t(z_t^*)$ 最大，根据定义，这个 z_t^* 也使得 $\Pr(Z = z, X = x)$ 最大，把这个最大值记为 δ_t^* 。

3. 接下来，逐层用第 $k+1$ 层的 δ_{k+1}^* 计算第 k 层的 δ_k^* 和 z_k^* . 已知

$$\delta_{k+1}^* = \delta_{k+1}(z_{k+1}^*) = \max_q \{ \delta_k(q) T_{q, z_{k+1}^*} \} M_{z_{k+1}^*, x_{k+1}},$$

从中找到一个 q 使得 $\delta_k(q) T_{q, z_{k+1}^*}$ 最大，记此时的 $\delta_k(q)$ 为 δ_k^* , $z_k^* = q$.

4. 最后，我们就得到了最优的状态序列 $z = (z_1^*, z_2^*, \dots, z_t^*)$ ，使得 $\Pr(Z = z, X = x)$ 最大.

以上算法被称为 *Viterbi* 算法，是解码问题的一个高效算法，它需要计算的乘法次数是 $\mathcal{O}(t|Z|^2)$. 这一算法采用了动态规划的思想，实际上大部分和 Bellman 方程有关的问题（特别是最优化的问题）都可以用这一方法解决.

§2.5 扩散模型

本节我们讨论基于 Markov 链的另一种模型，即扩散模型. 不同于之前的模型，扩散模型的启发来自物理学. 滴一滴墨水到水中，最后墨会均匀地在水中分布，这一过程不会反过来，即墨水不会自动聚集到一起. 同样的现象在热力学中也有体现：如果把金属勺子的一端放在热水中，热量会从热水中通过勺子传导到勺子的另一端，最终整个勺子会均匀地变热，而不会反过来. 这些现象都是扩散过程. 本节的主要任务就是为这一过程建立数学模型，并讨论它的应用.

我们先看墨水的例子.

例 2.6 (墨水的扩散) 墨水的扩散有一个具体的数学模型，即 *Ehrenfest* 模型.

模型如图 2.9 所示. 在这个模型中，一共有两个箱子 A 和 B，我们可以想象成这是水杯的上半部分和下半部分. 两个箱子里一共 N 个球，我们可以想象成这是墨水分子. 每个时刻，有很小的概率 $\alpha > 0$ ，盒子 A 和 B 中的球会保持不动；另外的 $1 - \alpha$ 的概率，我们均匀随机选择一个球，从一个箱子跳到另一个箱子，这就是扩散的过程.

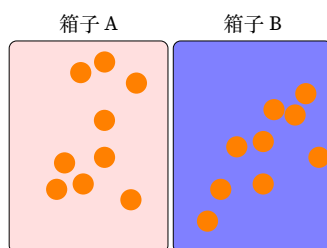


图 2.9: Ehrenfest 模型示意图

设 X_k 是盒子 A 中球的数目，那么 $\{X_k\}$ 是一个 Markov 链，它的转移核²是

$$\Pr(X_{k+1} = i | X_k = j) = \begin{cases} \alpha, & i = j, \\ (1 - \alpha)j/N, & i = j - 1, \\ (1 - \alpha)(N - j)/N, & i = j + 1, \\ 0, & \text{其他.} \end{cases}$$

下面我们来解释为什么墨水最终会均匀分布在水中。从遍历定理（定理 2.2）我们知道，对于一个遍历的 Markov 链，无论初始分布是什么，Markov 链最终都会收敛到唯一平稳分布。可以证明（见习题[hy: 出一下]）， $\{X_k\}$ 是一个遍历的 Markov 链，所以它会收敛到一个平稳分布 $\pi \sim B(N, 1/2)$ 。

根据二项式的性质，该分布在 $N/2$ 附近取到最大值，因此，这是墨分子最有可能的分布情况。更精细的结论是（见习题[hy: 出一下]），对任意 $\epsilon > 0$ ，当 N 充分大的时候，

$$\Pr \left(\limsup_{k \rightarrow \infty} \left| \frac{X_k}{N} - \frac{1}{2} \right| < \epsilon \right) \approx 1.$$

换言之，几乎以概率 1，分子会均匀分布在两个箱子中，即墨水会均匀地分布在水中。□

以上例子告诉我们，物理过程几乎是不可逆的。然而，如果我们仔细记录每一步扩散的过程，我们可以尝试倒推 X_0 的分布。这就是扩散模型的基本思想。下面给出扩散模型的定义。

定义 2.11 (扩散模型) 扩散模型由随机变量 x_0, \dots, x_T 给出，它包含两个过程：

- **扩散过程**：从 x_0 到 x_T 的 Markov 链，它的核是 $q(x_{k+1}|x_k)$ ，也叫**正向过程**。通常，扩散模型是连续型随机变量，所以，核是一个转移密度函数。
- **逆向过程**：从 x_T 到 x_0 的 Markov 链，它的核是 $p(x_{k-1}|x_k)$ 。通常，扩散模型是连续型随机变量，所以，核是一个转移密度函数。

注意，扩散过程和逆向过程都是非时齐的 Markov 链。假设扩散过程是一个 Markov 链，那么逆向过程必须是一个 Markov 链，所以上述定义是良定义的（见习题[hy: 出一下]）。

²我们在前面都说的是转移矩阵，然而，在扩散模型这一部分，转移核是一个更恰当的表述。这是因为，我们之前处理的大部分 Markov 链都是离散的状态空间，而这一部分处理的状态空间既有离散的也有连续的，所以“矩阵”并不是一个恰当的词汇。

除了物理过程，还有很多其他的场景也可以被看成扩散模型。比如，我们可以将一张图片看成刚滴入墨水的水杯：它有结构，不是混乱无序的。³ 而水则可以看作一张由 Gauss 噪声生成的图片：它是随机的，缺乏结构。于是，墨分子扩散就可以被理解为一张有结构的图片逐渐变成噪声的过程。

在这种理解下，扩散过程就是图片加噪声的过程，逆向过程就是去噪的过程。尽管通过加噪，所有的图片都会变成噪声，然而，噪声与噪声之间细微的区别仍然是可以被区分的。于是，我们可以通过逆向过程，从噪声中恢复出原始的图片。

如果我们通过大量的图片训练了一个扩散模型，那么我们就可以用它来凭空生成图片。首先，我们随机生成一个噪声图片 x_T ，然后通过逆向过程，我们就可以回到有结构的原始图片 x_0 。因此，扩散模型可以被用作生成模型。

我们接下来都会以去噪扩散概率模型（DDPM）为例来讨论扩散模型，它是一个用来生成图片的神经网络模型。DDPM 的过程如图 2.10 所示。

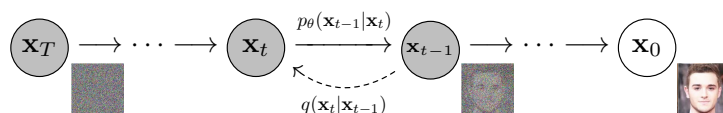


图 2.10: DDPM 示意图

DDPM 中的 Markov 核都是 Gauss 分布，具体如下：

- 扩散过程：假设真实图片的分布是 $q(x_0)$ ，这就应该 x_0 的初始分布。扩散过程从 x_{t-1} 到 x_t 就是在 x_{t-1} 上加上一个小的 Gauss 噪声扰动，因此我们可以写出它的转移核和初始分布：

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t\mathbf{I}), \quad x_0 \sim q(x_0).$$

这里 β_t 是一个超参数， $\mathcal{N}(x; \mu, \Sigma)$ 是 Gauss 分布 $\mathcal{N}(\mu, \Sigma)$ 在点 x 的概率密度。

- 逆向过程：扩散的末端 x_T 是一个完全的 Gauss 噪声，我们不妨设它是标准 Gauss 分布，即 $x_T \sim \mathcal{N}(0, \mathbf{I})$ 。逆向过程希望从 x_t 恢复到 x_{t-1} ，因此我们应该用 x_t 和 t 来预测 x_{t-1} 的期望，我们用一个神经网络 $\mu_\theta(x_t, t)$ 来表示这个期望。于是，逆向过程的转移核和初始分布为：

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2\mathbf{I}), \quad p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I}).$$

这里 σ_t 是一个超参数。

³有趣的是，古代中国和西方真的有在水面上撒颜料来作画的技法，也就是在水上撒颜料，然后用纸张把颜料印下来。在中国，这种技法叫湿拓画。在西方，这种技法叫 *marbling* 或者 *ebru*（来自土耳其语）。

根据我们的上面的讨论，扩散模型有两个任务：

1. 训练一个逆向过程，使得 $p_\theta(x_0)$ 尽可能接近数据的分布 $q(x_0)$.
2. 采样一个逆向过程，对一个训练好的扩散模型，这个步骤可以从噪声 x_T 出发生成一张有结构的图片 x_0 .

对于第一步，我们需要将训练的损失函数写出. 对于第二步，我们需要一个采样的方法. 由于采样比训练更简单，而且训练依赖采样，所以我们会先讨论采样.

2.5.1 采样逆向过程

因为逆向过程是一个 Markov 链，我们需要能够采样一个 Markov 链的状态序列. 因为随机变量之间有相互依赖关系，Markov 链的采样不是那么平凡的. 但是，Markov 性给了我们一种采样的方法.

考虑 Markov 链 X_k ，转移核为 $p(i, j)$ ，初始分布为 λ . 我们可以通过以下方法采样 X_k :

1. 从初始分布 λ 采样 X_0 .
2. 对于 $k = 1, 2, \dots$ ，从 $p(X_{k-1}, \cdot)$ 采样 X_k .

根据 Markov 性，在给定 X_{k-1} 的情况下， X_k 的分布只依赖于 X_{k-1} ，所以这一采样方法可以正确地采样 X_k . 根据这一原则，我们可以很具体地将逆向过程的采样算法写出.

1. $x_T \sim \mathcal{N}(0, \mathbf{I})$
2. 对 $t = T, \dots, 1$ ，重复以下步骤采样 x_{t-1} :
 - (1) 如果 $t \geq 1$ ，采样 $z \sim \mathcal{N}(0, \mathbf{I})$ ；否则， $z = 0$.
 - (2) $x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z$.
3. 输出 x_0 .

为了和后面训练部分对应，我们需要将 2.(2) 中的步骤重写为另一个等价的形式，现在可以先不管为什么要这么做. 令 $\alpha_t = 1 - \beta_t$ ， $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ ， $\epsilon_\theta(x_t, t)$ 是 $\mu_\theta(x_t, t)$ 的重参数化：

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right).$$

于是，我们可以将 2.(2) 重写为：

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z.$$

2.5.2 训练逆向过程

接下来我们讨论训练逆向过程的损失函数. 我们需要衡量 $p(x_0)$ 和 $q(x_0)$ 的差异, 一个常用的选择是交叉熵, 我们将会在下一章中详细讨论这一概念. 此时, 我们先接受交叉熵的概念, 将损失函数写为:

$$\mathbb{E}_q[-\log p_\theta(x_0)] = - \int q(x_0) \log p_\theta(x_0) dx_0.$$

为计算这一表达式, 我们需要将 $p_\theta(x_0)$ 的表达式算出. 类似 HMM, 我们用 $x_{i:j}$ 表示 x_i, \dots, x_j .

根据全概率公式:

$$\begin{aligned} p_\theta(x_0) &= \int p_\theta(x_{0:T}) dx_{1:T} \\ &= \int p_\theta(x_{0:T}) \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} dx_{1:T} \\ &= \int q(x_{1:T}|x_0) \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} dx_{1:T} \\ &= \int q(x_{1:T}|x_0) p_\theta(x_T) \prod_{t=1}^T \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} dx_{1:T}. \end{aligned}$$

接下来, 我们计算并放缩损失函数, 最终得到容易利用梯度进行优化⁴的形式:

$$\begin{aligned} K &= - \int q(x_0) \log p(x_0) dx_0 \\ &= - \int q(x_0) \cdot \log \left[\int q(x_{1:T}|x_0) \cdot p(x_T) \prod_{t=1}^T \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})} dx_{1:T} \right] dx_0. \end{aligned}$$

利用 Jensen 不等式 (定理 C.17), $\log(\int q f dx) \geq \int q \log(f) dx$, 这可以被放缩为:

$$K \leq - \int q(x_{0:T}) \log \left[p(x_T) \prod_{t=1}^T \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \right] dx_{0:T} := L.$$

可以证明 (留到下章作业), L 可以被写作:

$$\begin{aligned} &\mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(x_T|x_0) \| p(x_T))}_{L_T} + \right. \\ &\quad \left. \sum_{t>1} \underbrace{D_{\text{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))}_{L_{t-1}} - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} \right], \end{aligned}$$

其中 $D_{\text{KL}}(f\|g) = \int f \log(f/g) dx$ 是 K-L 散度.

接下来我们分别讨论每一部分的计算.

⁴对于优化相关的详细讨论, 见第三部分.

- L_T 不含参数，所以可以丢掉。
- L_0 和输出的数据格式有关（例如图片如何编码），需要具体问题具体处理，所以这里不讨论。
- 唯一需要处理的是 L_1 到 L_{T-1} 的计算。

由于 $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$ ，可以算得（见习题[thy: 出一下]）：

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] + C,$$

其中 C 是一个不含 θ 的常数，并且

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t.$$

因为 C 是一个常数，我们可以丢弃它而不影响最优值。接着，利用重参数化

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right),$$

引入随机变量 $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ ，我们可以将 L_{t-1} 写作：

$$\mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t \right) \right\|^2 \right].$$

如果我们想用基于梯度方法对它进行优化，我们需要计算它的梯度。然而，这是一个期望，没有办法直接求梯度，所以我们用采样到的 x_0 和 ϵ 代替期望中的 x_0 和 ϵ ，把它当成期望来求梯度。

另外，如果把所有 L_{t-1} 都用来训练，当总时长 T 非常大的时候，训练也会非常困难，所以在实际训练中，我们会均匀随机选择一个 t ，对 L_{t-1} 进行优化。

最终，我们得到了扩散模型的训练算法：

- 重复以下步骤直到收敛：
 - 采样 $x_0 \sim q(x_0)$ 。
 - 采样 $t \sim \mathcal{U}(\{1, \dots, T\})$ 。
 - 采样 $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ 。
 - 用下面的梯度做一次优化：

$$\nabla_\theta \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t \right) \right\|^2.$$

第二部分

信息与数据

第三部分

决策与优化

第四部分

逻辑与博弈

第五部分

认知逻辑

第六部分

附录：预备知识

参考文献

- [Bre57] Leo Breiman. The Individual Ergodic Theorem of Information Theory. *The Annals of Mathematical Statistics*, 28(3):809–811, 1957.
- [CT12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.
- [Huf52] David A. Huffman. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9):1098–1101, September 1952.
- [Inf] Information | Etymology, origin and meaning of information by etymonline. <https://www.etymonline.com/word/information>.
- [Jay02] Edwin T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2002.
- [KL51] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [LLG⁺19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, October 2019.
- [McM53] Brockway McMillan. The Basic Theorems of Information Theory. *The Annals of Mathematical Statistics*, 24(2):196–219, June 1953.
- [RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, USA, January 1986.

- [Rob49] Robert M. Fano. *The Transmission of Information*. March 1949.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.
- [Shi96] A. N. Shiryaev. *Probability*, volume 95 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1996.
- [Tin62] Hu Kuo Ting. On the Amount of Information. *Theory of Probability & Its Applications*, 7(4):439–447, January 1962.
- [Uff22] Jos Uffink. Boltzmann’s Work in Statistical Physics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2022 edition, 2022.
- [李 10] 李贤平. 概率论基础. 高等教育出版社, 2010.

索引

- AlphaGo, 42, 43
- AlphaStar, 42
- AlphaZero, 43
- Bellman 方程, 35, 40, 42
- DDPM, 50
- Ehrenfest 模型, 48
- HMM, 42, 43, 52
- Jensen 不等式, 52
- K-L 散度, 52
- Kolmogorov-Chapman 方程, 27
- Markov 决策过程, 36, 37
- Markov 奖励模型, 33
- Markov 奖励过程, 32, 33
- Markov 性, 25
- Markov 链, 25
- MDP, 36, 37
- Monte-Carlo 评估, 36
- MRP, 32, 33
- Pólya 的坛子, 26
- Q-learning, 42
- Sarsa, 42
- Viterbi 算法, 48
- Wald 等式, 42
- 不动点, 36
- 交叉熵, 52
- 价值函数, 34, 39
 - 状态-~, 39
 - 行动-~, 39
- 价值系统, 32
- 价值迭代, 42
- 前向方程, 42
- 动态优化, 42
- 动态规划, 36, 48
- 去噪扩散概率模型, 50
- 回报, 34, 39
- 平滑, 46
- 平稳分布, 31
- 强化学习, 42
 - 深度~, 42
- 扩散模型, 25, 48, 49
- 扩散过程, 48, 49
- 时序差分学习, 36
- 时齐的, 25
- 最优状态-价值函数, 40
- 最优行动-价值函数, 40
- 梯度, 52, 53

正向过程, 49

状态-价值函数, 34

状态空间, 25

生成模型, 50

策略, 38

策略迭代, 42

解码, 46

解释, 44, 46

评估, 44, 45

调和函数, 42

赌徒模型, 25

赌徒谬误, 27

转移核, 25

转移矩阵, 25

过滤, 46

逆向过程, 49

遍历, 30

遍历定理, 29

隐 Markov 模型, 43

预测, 46

马氏链, 25