强化学习与预测论

1900010681 李东晨

2023 年 6 月 1 日

致谢：本讲义参考了邓小铁老师主讲的课程《AI 中的数学》第一章讲义. 特别感谢李翰禹同学慷慨提供讲义的 latex 代码.

1 Markov 链

1.1 引言

Markov 链 (Markov Chain，又译马氏链、马尔可夫链) 是含时的随机现象最简单的数学模型. 在简单的框架之下，我们得以对其行为做大量的分析. 同时，它又具有丰富的应用. 这使得 Markov 链成为随机过程的第一个，也是最重要的例子. 事实上，随机过程论的整个数学研究都可以被视作关于 Markov 链的理论的推广.

——Norris, *Markov Chains*

1.2 基础概念

Markov 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 $\mathcal{P}$：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. [1]

Markov 性（Markov property）：对任意时刻 $t = 1, \ldots, n$ 和任意状态 $j, k, j_0, \ldots, j_{t-1} \in \mathcal{S}$，要求如下等式成立：

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

其意义为：在固定现在的情况下，未来与过去独立. 例如，可以证明 Markov 性可以推导出下面的性质：

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

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

[1] 注：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（time-homogeneous）.

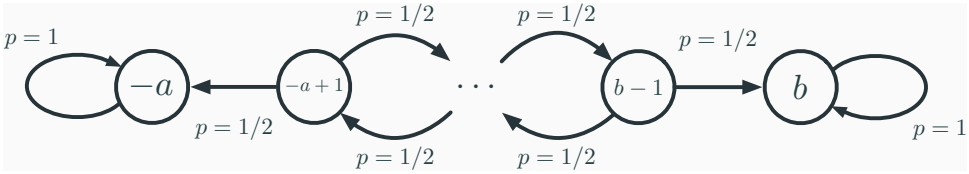

图 1: 公平对赌马氏链图示

1.3 举例（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 块钱时破产，B 输 b 块钱时破产.

我们记 $Z_i (i \geq 1)$ 是第 i 轮 A 的收入, 则 $\Pr[Z_i = 1] = Pr[Z_i = -1] = 1/2$. 记 $X_0 = 0$ 是 A 初始的收入，$X_n = X_0 + Z_1 + \cdots + Z_n$ 是 A 的累计收入. 那么，$\{X_n\}_{n\geq 0}$ 是一个 Markov 链：

- 状态空间：$\mathcal{S} = \{-a, -a+1, \ldots, 0, 1, \ldots, b\}$.
- 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.

从这个例子，我们可以反驳一个常见的谬误：赌徒谬误. 人们总是认为自己现在的“运气”和过去相关，特别是过去运气差未来就会变好.

从这个例子出发，除了基础的性质之外，我们还想进一步了解这个模型的其他性质，例如，A 和 B 分别输掉的概率，赌博结束的期望时间等等，这就首先需要考虑多步转移的概率，即从某一个状态出发，走若干步之后落在各个状态的概率分布.

设 $p_{ij}^{(k)}$ 表示从状态 i 用 k 步转移到状态 j 的概率，那么 k 步转移概率形成了一个矩阵 $\mathcal{P}^{(k)}$. 一个比较显然的结果是 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，$p_{ij}^{(k+l)} = \Pr(X_{k+l} = j | X_0 = i) = \sum_\alpha \Pr(X_{k+l} = j, X_k = \alpha | X_0 = i) = \sum_\alpha \Pr(X_k = \alpha | X_0 = i) \Pr(X_{k+l} = j | X_k = \alpha) = \sum_\alpha p_{i\alpha}^{(k)} p_{\alpha j}^{(l)}$.

特例：前向方程 (forward equation)：$\mathcal{P}^{(k+1)} = \mathcal{P}^{(k)}\mathcal{P}$，后向方程 (backward equation)：$\mathcal{P}^{(l+1)} = \mathcal{P}\mathcal{P}^{(l)}$.

推论：$\mathcal{P}^{(k)} = \mathcal{P}^k$. 若已知初始分布（列）向量为 λ，我们可以计算它随时间的演化：

$$\lambda^\intercal, \lambda^\intercal \mathcal{P}, \ldots, \lambda^\intercal \mathcal{P}^n, \ldots$$

1.4 遍历定理

现在我们考虑若干步之后 Markov 链的概率分布：一个直觉是，上面的演化过程如果收敛到一个分布 λ^*，那么需要满足 $\mathcal{P}^T \lambda^* = \lambda^*$，即 λ^* 为 $(I - \mathcal{P}^T)x = 0$ 的解，与初始状态无关. 这就导出了下面的遍历定理：

Theorem 1.1 (遍历定理，Ergodic theorem)**.** *Markov 链的状态空间为 $\mathcal{S} = \{1, \ldots, N\}$，转移矩阵为 $\mathcal{P} = (p_{ij})$. 如果存在 n_0，使得 $\min_{ij} p_{ij}^{(n_0)} > 0$ （称为遍历性），那么存在分布 $\lambda = (\lambda_1, \ldots, \lambda_N)$ 使得*

$$\lambda_i > 0, \quad \sum_i \lambda_i = 1, \tag{1.1}$$

并且对于每一个 $j \in \mathcal{S}$ 和任意 $i \in \mathcal{S}$ 都有

$$\lim_{n\to\infty} p_{ij}^{(n)} \to \lambda_j. \tag{1.2}$$

反之，如果存在满足 (1.1) 和 (1.2) 的 λ，则该 *Markov* 链具有遍历性. 式 (1.1) 的 λ 满足

$$\lambda^{\mathsf{T}} = \lambda^{\mathsf{T}}\mathcal{P}. \tag{1.3}$$

条件 (1.1) 表明每一个状态被访问到的概率都是正的，因此，遍历定理表明，遍历的 Markov 链最终趋向于一个混合均匀的状态.

满足条件 (1.3) 的分布被称为平稳分布（stationary distribution）.

- 平稳分布为初始状态时，Markov 链的演化与时间无关：$(X_k, \dots, X_{k+l})$ 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.
- 非遍历 Markov 链可能不存在平稳分布，也可能存在（唯一）平稳分布，例如，考虑 $\mathcal{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ 和 $\lambda = (1/2, 1/2)^{\mathsf{T}}$.

Remark 1.1. Markov 链是概率论的核心模型之一，但是受到课程内容限制，不可能对 Markov 链过于深入。关于 Markov 链更深入的讨论，例如，对遍历定理、停时、随机游走、连续时间 Markov 过程（跳过程）等的讨论，可以参考章复熹老师的《应用随机过程》课程。
课程主页：https://www.math.pku.edu.cn/teachers/zhangfxi/homepage/slidesSPA.htm

Exercise 1.1.

1. 证明：如果 *Markov* 链是遍历的，那么平稳分布是唯一的。

2. *(Ehrenfest* 模型*) Ehrenfest* 于 *1906* 年在讨论统计力学的循环问题时提出了如下的试验设想*:* 设有 N 个分子分布在两个容器 A 和 B 中. 在时刻 n 随机地选取一个分子并把它从原来所在的容器移到另一个容器中. 系统的状态用容器 A 中的分子数目 X_n 表示. 解答如下问题*:*

   (1) 证明*:* 马氏链 $\{X_n\}_{n=1}^{\infty}$ 是非遍历的.

   (2) 计算马氏链 $\{X_n\}_{n=1}^{\infty}$ 的平稳分布 π.

3. *(*赌徒模型的 *Wald* 等式*)* 假设甲乙两人参加公平对赌（每局独立以 $1/2$ 概率赢或输*),* 甲最初有 a 元，乙最初有 b 元 *(*a, b 都是正整数*),* 每局双方各下注 *1* 元，赢家可以获得本局所有的下注. 当其中一方赌资为 *0* 时赌博结束. 设 X_k 表示第 k 轮甲得到的钱. $X_0 = 0$, 设 $S_n = X_0 + \cdots + X_n, \tau_A = \min\{n \geq 0 : S_n = -a\}$ 表示甲输光的时间, $\tau_B = \min\{n \geq 0 : S_n = b\}$ 表示乙输光的时间. $\tau = \min\{\tau_A, \tau_B\}$ 即为赌博结束的时间.

   (1) 证明：赌博以概率 *1* 结束，即：$\Pr(\tau < \infty) = 1$. 提示：思考出现何种移动时赌博一定会结束，然后证明这种移动以概率 *1* 出现，从而推出：$\lim_{N\to\infty} \Pr \tau \geq N = 0$.

   (2) 计算甲输的概率。提示：所求概率等价于 $\Pr(S_\tau = -a \mid S_0 = 0)$. 考虑 $\Pr(S_\tau = -a \mid S_0 = x)$，利用 *Markov* 链的性质列出关于 x 的递推公式，解得 $x = 0$ 的情形.

   (3) 计算赌博结束的期望时间。提示：同理，考虑 $\mathsf{E}(\tau \mid S_0 = x)$，列出关于 x 的递推公式，解得 $x = 0$ 的情形.

2 Markov 奖励过程

2.1 引言

下面，我们对 Markov 链做一些推广. 注意到，在实际应用 Markov 链的过程中，不同的状态可能有不同的“价值”. 例如，在上面的公平对赌游戏中，对玩家来说，自己赢得越多，这个状态就对自己越有价值. 因此，我们希望在 Markov 链的状态上添加若干“奖励”，从而计算玩家需要的一些信息，例如游戏结束时的期望收益等.

2.2 基本概念

一个 Markov 奖励过程（Markov reward process，MRP）是四元组 $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$：

- $\mathcal{S}$ 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 $\mathcal{P}_{i,j}$.
- $\mathcal{R}$ 是一个奖励函数，$\mathcal{R}_s = \mathsf{E}[R_{t+1}|S_t = s]$：当 t 时刻位于状态 s 时下一时刻获得的奖励的期望，R_{t+1} 是下一阶段所处状态的奖励.（简单起见可以理解为任意时刻“停”在状态 s 的奖励是 R_s）
- γ 是一个折扣系数，$\gamma \in [0, 1]$，衡量未来的收益对玩家的价值.

注：条件数学期望 $\mathsf{E}[\cdot|S_t = s]$ 可以理解为条件在 $\{S_t = s\}$ 下定义的概率所求的期望.

MRP 中，t 时刻以后的总回报（Return）G_t 定义为

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

$\gamma \in [0, 1]$ 衡量了未来下一时段的 1 单元奖励在当前时刻的价值. 若 $\gamma \to 0$，表示对奖励进行“短视”的评估；反之更“远见”. 使用折扣系数是因为未来具有不确定性（风险），人们往往更重视即时回报.

现在，我们可以计算玩家在 Markov 奖励过程中的收益：我们定义价值函数（value function）$v(s)$ 表示从状态 s 出发的期望回报

$$v(s) = \mathsf{E}(G_t|S_t = s).$$

根据 Markov 性，这样定义的期望回报与出发的时刻无关（进行无穷轮），因此价值函数 $v(s)$ 衡量了状态 s 的长期效益.

2.3 Bellman 方程

那么，如何计算价值函数 $v(s)$ 呢？注意到，价值与时间无关，因此一个基本想法是据此构造一个递推式，称为 Bellman 方程.

价值函数可以被分解为两部分：

- 即时回报 R_{t+1}
- 下一个状态开始的折扣价值 $\gamma v(S_{t+1})$

据此，我们可以导出 Bellman 方程：

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

也可以将其简写为矩阵形式：

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

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

Bellman 方程是一个线性方程，可以直接求解:

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

使用简单的矩阵求逆，n 个状态的 Markov 链的计算复杂度为 $\mathcal{O}(n^3)$. 因此，较小的 MRP 可以直接求解，但较大的 MRP 开销太大，一般采用迭代算法，如：

- 动态规划（dynamic programming）
- Monte-Carlo 评估（Monte-Carlo evaluation)
- 时序差分学习（temporal-difference learning）

Exercise 2.1.

1. *考虑下面的 Markov 奖励过程：一名同学入学后是大一年级，每年有 1/2 的概率留级，1/2 的概率进入下一年级. 大学中共有四个年级，停留在第 i 年级的奖励为 +i. 进入大四后，再进入下一年级相当于毕业，毕业后不再获得奖励. 假设这名同学每年的折扣率是 $\gamma, 0 < \gamma < 1$，计算这名同学在大学的期望收益.*

3 Markov 决策过程

3.1 引言

当然，为了建模人的行为，有时候只引入奖励是不够的，还要考虑人自身的主观能动性. 我们希望智能体能够模仿人类，以某种局部最优的行为追求某个既定目标，如打游戏、下围棋等. 我们为自己构建的类比就是 Markov 决策过程.

在我们的一生中，我们会采取许多行动来追求我们的梦想. 其中一些会给我们带来丰厚的回报，而另一些则不会. 一路上，我们不断探索不同的路径，并试图找出哪种行动可能会带来更好的回报. 我们努力实现我们的梦想，利用我们根据我们的行动获得的反馈来改进我们的战略.

——Sayak Paul

3.2 基本概念

Markov 决策过程（Markov decision process, MDP）是一个定义了决策的 MRP. 它可以看做一个任意状态都具有 Markov 性的环境. 一个 MDP 是五元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$.

- $\mathcal{S}$ 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，受到当前状态和行动的影响：

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

- $\mathcal{R}$ 是一个奖励函数，$\mathcal{R}^a_s = \mathsf{E}(R_{t+1} | S_t = s, A_t = a)$，$R_{t+1}$ 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 $\gamma \in [0, 1]$.

在 MDP 中，一个策略（policy）π 是给定状态下行动的概率分布，给出采取各个行动的概率：

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

一个策略完全决定了一个智能体在 MDP 环境中的行为. 同样地，根据 Markov 性，策略与当前时刻无关.

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：

$$\mathcal{P}^\pi_{s,s'} = \mathsf{E}_{a \sim \pi(\cdot|s)}(\mathcal{P}^a_{s,s'}) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{s,s'},$$

$$\mathcal{R}^\pi_s = \mathsf{E}_{a \sim \pi(\cdot|s)}(\mathcal{R}^a_s) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s.$$

因此，我们同样可以定义价值函数：在 MDP 中，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

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

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

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

注意，以上定义都与时刻无关.

3.3 Bellman 方程

同理，我们可以给出 Bellman 方程：状态-价值函数可以被分解为即时回报加后续状态的折扣价值：

$$v_\pi(s) = \mathsf{E}_\pi(R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s).$$

行动-价值函数可以被类似地分解：

$$q_\pi(s, a) = \mathsf{E}_\pi(R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a).$$

二者之间的关系（全概率公式、一步转移概率）：

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

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

因此，我们得到 MDP 的 Bellman 期望方程：

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

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

其矩阵形式为：

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

但是，Bellman 方程现在只告诉我们对给定的策略如何求价值函数，并没有告诉我们怎么求出最优的策略. 为此，我们需要考虑什么策略下状态-价值函数达到最大.

定义 $v_\star(s)$ 是在状态 s 时所有决策中最大的状态-价值函数

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

定义 $q_\star(s,a)$ 是所有决策中最大的行动-价值函数

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

那么，最优价值函数确定了 MDP 中的最佳可能收益.

问题在于，每个状态对应的最优策略可能不是同一个. 不过，MDP 的解的存在性定理保证了，存在一个在每个状态上都最优的策略：

Theorem 3.1 (MDP 解的存在性)**.** *对任意 MDP，存在一个最优策略 $\pi_\star$ 使得 $\forall \pi\ \pi_\star \geq \pi$：*

- *最优策略取得最优状态-价值函数：$v_{\pi_\star}(s) = v_\star(s)$.*
- *最优策略取得最优行动-价值函数：$q_{\pi_\star}(s,a) = q_\star(s,a)$.*

在价值函数 v_π 之外计算 q_π 是为了找到最优策略 π：对给定 s，选取 $a_\star = \text{argmax}_a\, q_\star(s,a)$，令 $\pi_\star(a_\star|s) = 1$，且对 $\forall a \neq a_\star$，$\pi_\star(a|s) = 0$.

容易证明 $\pi_\star$ 取得最优行动-价值函数：

由 $v_\pi(s) = \mathsf{E}_{a\sim\pi(\cdot|s)}(q_\pi(s,a)) \leq \mathsf{E}_{a\sim\pi(\cdot|s)}(q_\star(s,a)) \leq q_\star(s,a_\star) = v_{\pi_\star}(s)$ 知 $\pi_\star$ 取得最优状态-价值函数.

推论：对任意 MDP，总存在一个非随机的最优决策. 如果我们知道 $q_\star(s,a)$，我们就能获得最优决策.

根据这个性质，我们可以把上面的 Bellman 期望方程化简为 Bellman 最优性方程：

$$v_\star(s) = \max_a q_\star(s,a),$$

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

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

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

因此，要求最优策略，只需要求解 Bellman 最优性方程. 然而，该方程非线性，故不一定有解析解. 不过，我们可以求出其数值近似解. 常用的算法有下列迭代算法：

- 价值迭代（value iteration）
- 策略迭代（policy iteration）
- Q-learning
- Sarsa

3.4 Q-learning

这一节，我们主要介绍 Bellman 最优性方程的 Q-learning 算法. 这个算法在 1989 年由 Chris Watkins 提出，形式非常简洁，而且可以证明其收敛性.

在考虑正式的 Q-learning 算法前，我们先考虑更简单的价值迭代算法. 由于内存限制，我们要求所求 MDP 的状态和行动数量有限.

我们对所有状态和行动初始化一个行动-价值函数 $q_0(s,a)$，一般就全部初始化为 0. 之后，在每一轮迭代中，我们对每个 s,a 的行动-价值函数更新如下：

$$q_{t+1}(s,a) = \sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a \left[\mathcal{R}_s^a + \gamma \max_{a'} q_t(s',a') \right]$$

.

注意到，这是一个确定性算法，我们证明其必然收敛到最优行动价值函数 q^*：

定义作用于函数族 $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 的算子 $\mathbf{H}$：

$$(\mathbf{H}q)(s,a) = \sum_{s' \in S} \mathcal{P}_{s,s'}^a \left[\mathcal{R}_s^a + \gamma \max_{a'} q(s',a') \right]$$

注意到 $\sum_{s' \in \mathcal{S}} \mathcal{P}_{s,s'}^a = 1$，根据 Bellman 最优性方程，我们有 $\mathbf{H}q^* = q^*$. 而且，我们实际上有下面的定理：

Theorem 3.2. *在无穷范数下 $\mathbf{H}$ 是一个收缩系数为 γ 的收缩算子：即 $||\mathbf{H}q_1 - \mathbf{H}q_2||_\infty \leq \gamma ||q_1 - q_2||_\infty$.*

证明：注意到：

$$
\begin{aligned}
&\|\mathbf{H}q_1 - \mathbf{H}q_2\|_\infty = \\
&= \max_{s,a} \left| \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s,s'} \left[\mathcal{R}^a_s + \gamma \max_{a'} q_1(s', a') - \mathcal{R}^a_s - \gamma \max_{a'} q_2(s', a') \right] \right| = \\
&= \max_{s,a} \gamma \left| \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s,s'} \left[\max_{a'} q_1(s', a') - \max_{a'} q_2(s', a') \right] \right| \leq \\
&= \max_{s,a} \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s,s'} \left| \max_{a'} q_1(s', a') - \max_{a'} q_2(s', a') \right| \leq \\
&= \max_{s,a} \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s,s'} \max_{s',a'} |q_1(s', a') - q_2(s', a')| = \\
&= \max_{s,a} \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s,s'} \|q_1 - q_2\|_\infty = \\
&= \gamma \|q_1 - q_2\|_\infty .
\end{aligned}
$$

命题得证！

从而，迭代过程相当于计算函数 $q_{t+1} = \mathbf{H}q_t$. 故有：

$$||q_{t+1} - q^*||_\infty = ||\mathbf{H}q_t - q^*||_\infty = ||\mathbf{H}q_t - \mathbf{H}q^*||_\infty \leq \gamma ||q_t - q^*||_\infty$$

因此，q_t 以指数速度趋向于 q^*，命题得证！

但是，这个算法的问题在于，强化学习的应用中行动和状态的空间可能非常大，而我们的每一步迭代就要对所有的 s, a 更新行动-价值函数，需要的计算量和内存是很大的. 而且，就像现实中我们很难看清楚事物的全貌，在第一次做出一个行为之前很难评估它的后果，在强化学习中，智能体一般也只能得到在某个顶点 s 采取某个行动 a 的回报 R_{t+1} 和结果 s'，而很难得到整个状态空间和行为结果的完全信息，如行动的期望回报 $\mathcal{R}^a_s$ 和状态转移矩阵 $\mathcal{P}^a_{s,s'}$. 所以，我们希望寻找一个逐渐“探索”的算法，其在每一步更新时只能知道局部的信息，但最后却能找到全局的最优解.

现在，我们仍初始化一个行动-价值函数 $q_0(s, a)$（例如初始化为 0）. 但是，我们每步只更新一个 s, a 的 q 值. 随机选择一个满足 $\pi(a|s) > 0, \forall s, a$ 的策略 π（保证以非零概率探索所有可能行动）. 现在，假设我们是使用这个策略的智能体. 假设在时刻 t，我们位于 s_t，用策略 π 产生的行动是 a_t，做行动 a_t 后达到的状态是 s_{t+1}，获得的收益是 R_{t+1}. 那么我们如下更新行动-价值函数 a：

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \alpha_t(s_t, a_t) \left[R_{t+1} + \gamma \max_{a'} q_t(s_{t+1}, a') - q_t(s_t, a_t) \right].$$

算法中，第一项 $q_t(s_t, a_t)$ 表示过往经验，第二项则表示从相邻节点新学习到的信息. 因此，参数 $\alpha_t(s, a)$ 称为学习率.

这里，我们提供一个 Q-learning 收敛的直觉：虽然我们的迭代方程与 Bellman 方程差异很大，但是，实际上每一步都是对 Bellman 方程的一个无偏估计（期望等于 Bellman 方程）. 注意到：根据定义，有：

$$\mathsf{E}[R_t|s_t, a_t] = \mathcal{R}^{a_t}_{s_t}, \mathsf{E}[\max_{a'} q_t(s_{t+1}, a')|s_t, a_t] = \sum_{s' \in \mathcal{S}} \mathcal{P}^{a_t}_{s_t,s'} \max_{a'} q_t(s', a')$$

下面我们证明：如果学习率满足：对任意 s, a，

$$\sum_t \alpha_t(s, a) = \infty \quad \sum_t \alpha^2_t(s, a) < \infty,$$

那么随机算法 Q-learning 以概率 1 收敛到 Bellman 方程的解.

引理：(证明见 [1])

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\,\Delta_t(x) + \alpha_t(x)F_t(x)$$

满足下列假设时概率 1 收敛到零.

- $0 \le \alpha_t \le 1, \sum_t \alpha_t(x) = \infty, \sum_t \alpha_t^2(x) < \infty$;
- $\|\mathsf{E}\left[F_t(x) \mid \mathcal{F}_t\right]\|_W \le \gamma \|\Delta_t\|_W, \gamma < 1$;（这里 $\mathcal{F}_t$ 是指 t 步之前的历史信息，W 是任意度量）
- 存在 $C > 0$，使得 $\mathsf{Var}\left[F_t(x) \mid \mathcal{F}_t\right] \le C\left(1 + \|\Delta_t\|_W^2\right)$

下面证明 Q-learning 的收敛性：首先将迭代公式重写为：

$$q_{t+1}\left(s_t, a_t\right) = (1 - \alpha_t(s_t, a_t))q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left[R_{t+1} + \gamma \max_{a'} q_t(s_{t+1}, a')\right]$$

仍记最优行动价值函数为 $q^*(s,a)$，在两边同时减去 $q^*(s_t,a_t)$，定义 $\Delta_t(s,a) = q_t(s,a) - q^*(s,a)$，则有：

$$\begin{aligned}\Delta_{t+1}(s_t, a_t) &= (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t)\left[\mathcal{R}_{t+1} + \gamma \max_{a'} q_t(s_t, a') - q^*(s_t, a_t)\right] \\ &=\triangleq (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t)F_t(s_t, a_t)\end{aligned}$$

根据上面的无偏估计的推导，借用简单价值迭代中定义的算子 $\mathbf{H}$，有

$$\begin{aligned}\mathsf{E}[F_t(s,a)|s,a] &= \sum_{s' \in \mathcal{S}} P_{s,s'}^a \left[\mathcal{R}_s^a + \gamma \max_{a'} q_t(s', a') - q^*(s,a)\right] \\ &= (\mathbf{H}q_t)(s,a) - q^*(s,a) = (\mathbf{H}q_t)(s,a) - (\mathbf{H}q^*)(s,a)\end{aligned}$$

从而，$||\mathsf{E}\left[F_t(x) \mid \mathcal{F}_t\right]||_\infty \le \gamma ||q_t - q^*||_\infty = \gamma ||\Delta_t||_\infty$.

最后，把上面求出的期望代入，即可求出 F_t 的方差：

$$\begin{aligned}\mathsf{Var}\left[F_t(s,a) \mid \mathcal{F}_t\right] =& \mathsf{E}\left[\left(\mathcal{R}_{t+1} + \gamma \max_{a'} q_t(s,a') - q^*(s,a) - (\mathbf{H}q_t)(s,a) + q^*(s,a)\right)^2\right] \\ =& \mathsf{Var}\left[\mathcal{R}_{t+1} + \gamma \max_{a'} q_t(s,a') | \mathcal{F}_t\right]\end{aligned}$$

注意到在有限 MDP 中 R 有界，而 $|\max_{a'} q_t(s,a) - q^*(s,a)| \le ||\Delta_t||_\infty$，从而存在常数 C，使得 $\mathsf{Var}\left[F_t(s,a) \mid \mathcal{F}_t\right] \le C(1 + ||\Delta_t||_\infty^2)$.

综上，使用引理，命题得证！

Exercise 3.1.

1. 一个同学有两种状态，躺平和学习. 在躺平状态下，他有两个行动，一个是继续躺平，仍然保持在躺平状态，奖励 *+0*，另一个是开始学习，奖励*-1*，但不一定能成功，有 *1/2* 的概率保持在躺平状态，有 *1/2* 的概率进入学习状态；在进入学习状态后他只有继续学习一个行动，奖励 *+1*. 假设同学的折扣率是 $\gamma \in (0,1)$.

   (1) 分类讨论何时同学在躺平状态下的最优策略是继续躺平，进而利用 *Bellman* 方程直接计算 γ 在不同条件下同学的最优策略 π^* 和两个状态的价值 v.

   (2) 模拟价值迭代算法的过程，初始行动-价值函数为 *0*，计算 q^*，进而计算 γ 在不同条件下同学的最优策略和两个状态的价值（详细写出模拟过程）.

4 预测论

4.1 引言

在 MDP 和强化学习中，智能体需要依靠自己的探索来获取信息，进而计算自己的最佳决策。但是，日常生活中，我们往往可以从外界获得某些（不一定正确的）信息，然后依据这些信息去做出决策。这就是不完全信息人类决策的另一个模型，即预测论。

4.2 基本概念

预测论试图通过对某个序列的结果进行预测。亦即：对于在时刻 $t = 1, 2, ...$ 不断生成的序列 $\{y_t\}_{t=1}^n$，在每个时刻 t，预测者要基于之前的观察 $\{y_1, ..., y_{t-1}\}$ 和某些信息来预测 $\{y_t\}$，并试图达到最大的正确率。

预测论听起来很像统计学。但是，统计学往往假设 $\{y_t\}$ 产生自一个固定的分布族 $\mathcal{F}$，并且希望通过若干统计量（如样本期望，样本方差等）来确定数据具体来源于哪个分布。与其不同，预测论不对序列 $\{y_t\}$ 做任何假设，而是假设存在若干个“专家”，他们在每个时刻会提出各自的建议。预测者的目的是基于这些建议做出最正确的预测。我们通过下面的例子展示预测论的一些基本概念和思想：

一个典型的例子：从若干个基金经理的建议中选择是否买当前股票

Example 4.1. 假设预测者需要预测一个序列 $\{y_t\}_{t=1}^{\infty}, y_t \in \{0, 1\}$。从时刻 1 开始，在每个时刻 t，N 个专家会分别给他一个建议 $f_{t,i}, i = 1, \ldots, N$，他需要基于这些建议给出一个预测 $\hat{p}_t \in \{0, 1\}$。做出预测后，他会被告知 y_t 的真实值。他的目的是给出某个决策程序，使得自己犯错的总次数尽可能小。考虑下面的两种情况：

realizing, halving and agnostic

(1) 若已知至少有一名专家的预测是一定正确的，给出一个犯错总次数有限的决策程序。

(2) 若已知存在一名专家，在前 m 次的预测中保证至多犯 km 次错，k 为某个小于 1 的常数，给出一个在前 m 次犯错总比例仅与 k 相关的决策程序。

证明. (1) 一个很自然的想法是维护一个“可信专家集”，并且决策仅依赖他们的建议，例如，在他们的建议里面随机选择一个采纳。一开始，这个集合包含所有的专家，但是某个专家一旦犯错就将他移出这个集合。这样，就能保证这位预测一定正确的专家总是留在可信集内。采用这种策略，可以得到一个自然的错误上界 $N - 1$：因为预测者每次犯错就意味着至少有一个专家变得不可信，而一开始 N 个专家都可信，最终至少还剩一个专家，所以犯错次数至多为 $N - 1$。

但是，我们其实可以稍微改进一下决策过程，以大幅降低犯错误的次数。我们考虑“少数服从多数”的原则，每次从可信专家集的建议中采纳出现次数较多的建议。这样，我们一旦犯错，则至少有一半的专家都变得不可信。从而，犯错的次数至多是 $\lfloor \log_2 N \rfloor + 1$ 次。

(2) 由于我们现在无法保证一定有一个专家是正确的了，所以不能再采用之前一旦错误就不信任的方法。考虑对 N 个专家各自维护一个权重 w_i，并且专家若犯错一次，则将其权重更新为 βw_i $(\beta < 1)$，即减少他的策略的权重。(1) 中的方法正是 $\beta = 0$ 的特殊情形。

我们构造的策略是：在第一步时对所有专家都赋予权重 1，之后每一轮将上一次预测错误的专家的权重更新为 βw_i。在做决策时，我们比较所有给出预测 1 的专家的总权重和给出所有预测为 0 的专家的总权重，哪个总权重更高，我们就使用哪个预测。这样，我们可以保证，我们一旦犯错，那么至少有占据一半权重的专家犯错。

因此，假设我们在前 m 次中总共犯了 m^* 次错，那么，每次犯错会导致总权重从 W 至少减少到 $W/2+\beta W/2$，而一开始的总权重为 N，那么最终的总权重至多为：

$$\left(\frac{1+\beta}{2}\right)^{m^*} N$$

但是，根据条件，至少有一名专家，他至多会犯 km 次错，故权重至多会被减少 km 次，故他的权重至少是 β^{km}. 综合以上两个式子，有：

$$\beta^{km} \leq \left(\tfrac{1+\beta}{2}\right)^{m^*} N$$
$$\text{解得：}\ \frac{m^*}{m} \leq \left\lfloor \frac{\log_2 N}{m \log_2 \frac{2}{1+\beta}} + k\frac{\log_2(1/\beta)}{\log_2 \frac{2}{1+\beta}} \right\rfloor$$

从而得到了一个错误比例仅与 k 有关的决策。特别的，当决策总次数 $m \to \infty$ 时，该比例趋近于 $k\frac{\log(1/\beta)}{\log(2/(1+\beta))}$，$\beta \to 1$ 时取到最小值 $2k$. 而且，即使是 m 比较小的情况，我们的决策也能保证受到专家总数量 N 的影响很小。 □

从这个例子里，我们可以看到很多决策论的基本思想，例如，为不同的建议加权，总是选择能够引入最多信息的决策等等。下面，我们给出专家建议决策的一个正式的数学模型。这个模型和上面例子的区别主要是引入了损失函数。因为，在决策空间比 $\{0,1\}$ 更大的情况下，不再能够使用“对/错”简单地描述决策的结果。

Definition 4.1. 专家建议决策：
参数：决策空间 D，D 是凸向量空间，结果空间 Y，损失函数 $l: D \times Y \to \mathbb{R}$，专家集合 E.
对于每一轮 $t = 1, 2, ...$:

（1）专家们提供建议 $\{f_{e,t} \in D : i \in E\}$.

（2）预测者选择预测 $\hat{p}_t \in D$.

（3）环境揭示结果 $y_t \in Y$.

（4）预测者遭受损失 $l(\hat{p}_t, y_t)$，每个专家 e 遭受损失 $l(f_{e,t}, y_t)$.

从而可以定义：

玩家的累积损失：$\hat{L}_n \triangleq \sum_{t=1}^{n} l\left(\hat{p}_t, y_t\right)$.

专家 e 的累积损失：$L_{e,n} \triangleq \sum_{t=1}^{n} l\left(f_{e,t}, y_t\right)$

预测者的目标：

最小化自己的累积损失和每个专家 e 的累积损失之差，即没有遵守专家的建议导致的损失，称为后悔 (regret)，即：

$$\text{Minimize: } R_{e,n} \triangleq \hat{L}_n - L_{e,n}$$

定义第 t 轮中对专家 e 的后悔为 $r_{e,t} = l\left(\hat{p}_t, y_t\right) - l\left(f_{e,t}, y_t\right)$.

那么，将上面例题的思想略作推广，就可以得到在关于专家建议决策问题的一个决策机制：

我们只考虑专家总数有限的情况，因此不妨设共 N 个专家，编号为 $1, 2, \ldots, N$。在时刻 t，假设这些专家的后悔分别为 $R_{1,t}, \ldots, R_{N,t}$。根据定义，一个专家的后悔值越高，说明他的建议越本该被采纳。仿照上面的思想，我们希望构造关于后悔值的非负权重函数 w，使得后悔值越高，权重就越大。这样，在第 t 轮时，这些专家的权重分别为 $(w(R_{1,t}, \ldots, R_{N,t}))$，因此预测者在这一轮的预测应该为：

$$\widehat{p}_t = \frac{\sum_{i=1}^{N} w(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^{N} w(R_{j,t-1})}$$

Lemma 4.1. 如果损失函数 l 对于第一个参数是凸的，那么：

$$\sup_{y_t \in Y} \sum_{i=1}^{N} r_{i,t} w\left(R_{i,t-1}\right) \leq 0$$

证明. 使用 Jensen 不等式，有：

$$\begin{aligned} l(\widehat{p}_t, y) = \quad & l\left(\frac{\sum_{i=1}^{N} w(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^{N} w(R_{j,t-1})}, y\right) \leq \frac{\sum_{i=1}^{N} w(R_{i,t-1}) l(f_{i,t}, y)}{\sum_{j=1}^{N} w(R_{j,t-1})} \\ \text{即：} & \sum_{i=1}^{N} w\left(R_{i,t-1}\right)\left(l\left(\hat{p}_t, y\right) - l\left(f_{i,t}, y\right)\right) = \sum_{i=1}^{N} w\left(R_{i,t-1}\right) r_{i,t} \leq 0 \end{aligned}$$

□

这个引理的给出了这种加权结构的一个优良性质，即：不论序列中下一个出现的元素是什么，采用加权方法给出的新策略都能保证这一轮中所有的后悔值的加权和的上界为 0。而且，更重要的是，这种形式和方向导数很接近。我们下面说明：通过构造特定势函数，那么这种结构可以进一步转换为更加有利于分析的形式。

将时刻 t 对所有专家的后悔值记为一个向量 $r_t = (r_{1,t}, \ldots, r_{N,t})$，该向量在时间 t 上的累积为 $R_t = \sum_{j=1}^{t} r_j$，那么，定义关于 n 维向量 u 的势函数 $\Phi(u) = \psi\left(\sum_{i=1}^{N} \phi(u_i)\right)$，如果我们取 $w = \phi'$，则：

$$\nabla\Phi(u) = ((\psi'(\sum_{i=1}^{N} \phi(u_i))\phi'(u_1), \ldots, (\psi'(\sum_{i=1}^{N} \phi(u_i))\phi'(u_N))$$

从而，上面引理的等价形式为：$\sup_{y_t \in Y} \sum_{i=1}^{N} r_t \cdot \nabla\Phi(R_{t-1}) \leq 0$.（Blackwell 条件）

为了保证现在的权重函数 ϕ' 满足一开始的三个条件，即：非负、单调递增，我们需要 ϕ 是非负单调递增的凸函数。

根据方向导数小于等于 0 的条件，利用泰勒展开，我们立即得到：

$$\begin{aligned} \Phi\left(\mathbf{R}_t\right) =& \Phi\left(\mathbf{R}_{t-1} + \mathbf{r}_t\right) \\ =& \Phi\left(\mathbf{R}_{t-1}\right) + \nabla\Phi\left(\mathbf{R}_{t-1}\right) \cdot \mathbf{r}_t + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left.\frac{\partial^2 \Phi}{\partial u_i \partial u_j}\right|_{\xi} r_{i,t} r_{j,t} \leq \quad \Phi\left(\mathbf{R}_{t-1}\right) + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left.\frac{\partial^2 \Phi}{\partial u_i \partial u_j}\right|_{\xi} r_{i,t} r_{j,t} \end{aligned}$$

我们对余项进行估计：

$$\begin{aligned} & \sum_{i=1}^{N} \sum_{j=1}^{N} \left.\frac{\partial^2 \Phi}{\partial u_i \partial u_j}\right|_{\xi} r_{i,t} r_{j,t} \\ & = \psi''\left(\sum_{i=1}^{N} \phi\left(\xi_i\right)\right) \sum_{i=1}^{N} \sum_{j=1}^{N} \phi'\left(\xi_i\right) \phi'\left(\xi_j\right) r_{i,t} r_{j,t} \\ & + \psi'\left(\sum_{i=1}^{N} \phi\left(\xi_i\right)\right) \sum_{i=1}^{N} \phi''\left(\xi_i\right) r_{i,t}^2 \\ & = \psi''\left(\sum_{i=1}^{N} \phi\left(\xi_i\right)\right)\left(\sum_{i=1}^{N} \phi'\left(\xi_i\right) r_{i,t}\right)^2 + \psi'\left(\sum_{i=1}^{N} \phi\left(\xi_i\right)\right) \sum_{i=1}^{N} \phi''\left(\xi_i\right) r_{i,t}^2 \\ & \leq \psi'\left(\sum_{i=1}^{N} \phi\left(\xi_i\right)\right) \sum_{i=1}^{N} \phi''\left(\xi_i\right) r_{i,t}^2 \quad \text{（假设 } \psi \text{ 是凹函数）} \end{aligned}$$

迭代地使用这个结果，即可得到定理：

Theorem 4.1. 若 ψ 为二阶可导凹函数，ϕ 为非负单调递增的二阶可导凸函数，则其定义的函数 $\Phi(u)=\psi\left(\sum_{i=1}^{N}\phi(u_i)\right)$ 使得：

$$\Phi(R_t)\leq\Phi(0)+\tfrac{1}{2}\sum_{j=1}^{t}C(r_j)$$
$$\text{其中，}\ C\left(\mathbf{r}_t\right)=\sup_{\mathbf{u}\in\mathbb{R}^N}\psi'\left(\sum_{i=1}^{N}\phi\left(u_i\right)\right)\sum_{i=1}^{N}\phi''\left(u_i\right)r_{i,t}^2$$

使用这个定理，我们可以对许多不同类型的加权函数进行分析，如使用 p-范数加权，指数加权等。这里我们给出指数加权的分析。

注意到：$\ln(x)/\lambda$ 是二阶可导凹函数，而 $e^{\lambda}x$ 是非负单调递增的二阶可导凸函数，我们可以给出一个满足上面定理条件的势函数

$$\Phi_\eta(\mathbf{u})=\tfrac{1}{\eta}\ln\left(\sum_{i=1}^{N}e^{\eta u_i}\right)$$

从而，有 $w_{i,t-1}=\nabla\Phi_\eta\left(\mathbf{R}_{t-1}\right)_i=\frac{e^{\eta R_{i,t-1}}}{\sum_{j=1}^{N}e^{\eta R_{j,t-1}}}$，预测者的预测为：

$$\widehat{p}_t=\frac{\sum_{i=1}^{N}\exp\left(\eta\left(\widehat{L}_{t-1}-L_{i,t-1}\right)\right)f_{i,t}}{\sum_{j=1}^{N}\exp\left(\eta\left(\widehat{L}_{t-1}-L_{j,t-1}\right)\right)}=\frac{\sum_{i=1}^{N}e^{-\eta L_{i,t-1}}f_{i,t}}{\sum_{j=1}^{N}e^{-\eta L_{j,t-1}}}$$

从而，权重函数可以化简为 $w_{i,t-1}=e^{-\eta L_{i,t-1}}$.

这表明指数加权方法有着非常好的性质。一方面，预测者的预测仅与各个专家的总后悔值有关，和预测者自己的后悔值无关；另一方面，两轮之间权重的变化更新非常容易，因为根据定义，$L_{i,t}=L_{i,t-1}+r_{i,t}$，因此，在新的轮次中，我们只需要将每个专家的权重更新为原来权重和 $e^{-\eta r_{i,t}}$ 的乘积。

下面我们用上面的估计定理对玩家的累计后悔值给出一个上界：

Corollary 4.1. 指数加权上界：
假设损失函数关于第一个参数凸，并且取值范围为 $[0,1]$，则对任意 n 和 $\eta>0$：

$$\widehat{L}_n-\min_{i=1,\ldots,N}L_{i,n}\leq\frac{\ln N}{\eta}+\frac{n\eta}{2}$$

证明. 代入 $\phi(x)=e^{\eta x},\psi(x)=(1/\eta)\ln x$，有：

$$\psi'\left(\sum_{i=1}^{N}\phi\left(u_i\right)\right)\sum_{i=1}^{N}\phi''\left(u_i\right)r_{i,t}^2\leq\eta\max_{i=1,\ldots,N}r_{i,t}^2\leq\eta.$$

从而，$C\left(\mathbf{r}_t\right)\leq\eta$，则

$$\max_{i=1,\ldots,N}R_{i,n}\leq\Phi_\eta\left(\mathbf{R}_n\right)\leq\frac{\ln N}{\eta}+\frac{n\eta}{2}$$

其中第一个不等式是由于琴生不等式：$\phi_\eta(R_n)=\frac{\ln(\sum_{i=1}^{N}e^{\eta R_{i,n}})}{\eta}\geq\frac{\sum_{i=1}^{N}\eta R_{i,n}}{\eta}$ □

Remark 4.1. 同样，对于预测论的讨论也不止于此。本节讲义主要参考了 N Cesa-Bianchi, G Lugosi 的《Prediction, Learning, and Games》第一章，有兴趣的同学可以深入阅读.

Exercise 4.1.

1. 利用定理 *2.2* 证明以下结论：如果损失函数对一个参数凸，且取值范围为 $[0,1]$，选取势函数

$$\Phi(u) = ||u_+||_p^2 \triangleq \left(\sum_{u_i>0, 1\le i\le N} (u_i)^p \right)^{2/p}$$

，则

$$\widehat{L}_n - \min_i L_{i,n} \le \sqrt{n(p-1)N^{2/p}}$$

2. 为了展示势函数的作用，定理 *2.2* 的证明思路和例题并不相同。事实上，用例题的证明思路，定义 $W_t = \sum_{i=1}^N w_{i,t}$，利用 $\frac{W_n}{W_0}$ 的上下界，可以证明定理 *2.2* 的强化版本。按照如下提示，给出证明：

（*1*）证明：

$$\ln \frac{W_n}{W_0} \ge -\eta \min_{i=1,\dots,N} L_{i,n} - \ln N.$$

（*2*）利用 *Hoeffding* 不等式：对任意随机变量 X 取值范围 $[a,b]$，$\ln \mathbb{E}\left[e^{sX}\right] \le s\mathbb{E}X + \frac{s^2(b-a)^2}{8}$，证明 $\ln \frac{W_t}{W_{t-1}} \le -\eta \ell\left(\widehat{p}_t, y_t\right) + \frac{\eta^2}{8}$

（*3*）结合以上两式，可得：$\widehat{L}_n - \min_{i=1,\dots,N} L_{i,n} \le \frac{\ln N}{\eta} + \frac{n\eta}{8}$

参考文献

[1] Tommi Jaakkola, Michael Jordan, and Satinder Singh. Convergence of stochastic iterative dynamic programming algorithms. *Advances in neural information processing systems*, 6, 1993. 3.4