人工智能基础大作业

徐恒　王陆航　郭朝晨

2022 年 12 月 20 日

目录

1 基本概念

1.1 通过文献调研为下列名词下定义

通过文献调研为下列名词下定义：二人零和博弈、扩展型博弈、博弈树、子博弈、动态博弈、重复博弈、合作博弈、不完全信息博弈、不完美信息博弈、贝叶斯博弈、弱劣势策略、纯策略、混合策略、最优反应（best response）、最大最小策略、鞍点策略、ε-纳什均衡。

二人零和博弈（two-player zero-sum game） 有两个参与人，不管双方采取什么策略，最终双方的收益之和总是为0，即一方的收益必然意味着另一方的损失。事实上，零和博弈与常和博弈本质相同。

扩展型博弈（extensive game） 扩展型博弈，又称动态博弈。通常通过树的形式来描述博弈，博弈从唯一一个初始节点开始，通过由参与者决定的路径到达终端节点，则博弈结束，参与人获得相应的收益。相比于策略型博弈，扩展型博弈的每个参与人可以在博弈中多次行动，且中途每一步都没有收益，只有最后一步才计算收益。

博弈树（game tree） 博弈树是扩展型博弈的一种形象化表达。它是基于扩展型博弈参与人的行动有先后次序，因此可以将参与人的行动展开成一个树状图形。博弈从唯一一个初始节点开始，双方通过轮流决策轮流扩展节点，最终到达终端节点，则博弈结束，参与人获得相应的收益。博弈树较为形象，能提供有限博弈的几乎所有信息，基本构建材料包括结、枝和信息集。

子博弈（subgame） 在动态博弈中，所有参与人先后都采取了一次行动后所构成的一组新的博弈，这组博弈中的每一个都称为子博弈。子博弈也可以用博弈树的形式进行表示，必须始于单个节点，且包含原博弈当中的所有后续节点，信息集必须仍然保持完整。事实上，子博弈是原博弈的一部分，它本身可以看作一个独立的博弈进行分析。

动态博弈（dynamic game） 动态博弈是指参与人的行动有先后顺序，后行动者可以观察到先行动者的决策并依次做出自己的决策。在决定动态

博弈中的战略时，要考虑到他人的决策，在博弈开始前就应当抉择每一个决策点上的选择。

重复博弈（repeated games） 重复博弈是指同样结构的博弈重复若干次，其中的每次博弈称为“阶段博弈”（stage games）。在重复博弈中，每次博弈的条件、规则和内容都是相同的，但由于各个参与人需要追求长远利益，因此需要考虑当前阶段的决策是否会导致后续阶段对方决策的改变而产生的利益损失，即在单次博弈当中也许需要考虑其他参与人的决策和利益。

合作博弈（cooperative game） 合作博弈，亦称正和博弈，博弈的参与人可以因合作而使得各个参与人的收益都有所增加，或者至少一方的收益得以增加、其他参与人的收益至少不受损害，因而合作可以使得整个社会的利益有所增加。合作博弈存在有两个基本条件，其一是合作联盟的整体收益大于其每个成员单独经营时的收益之和，其二是对联盟内部而言每个成员都能获得不少于不加入联盟时的收益。

不完全信息博弈（incomplete information game） 不完全信息博弈，也称贝叶斯博弈（Bayesian game），是指参与人对其他参与人的特征、策略空间及收益函数信息了解不够准确或者不是对所有参与人的特征、策略空间及收益函数都有准确的信息。博弈参与者对于其他参与者的收益函数没有完全信息。在分析不完全信息博弈时，通常采用海萨尼转换（the Harsanyi transformation），引入“自然”（nature）作为一个参与人引入博弈，其他参与人不知道自然的具体选择，仅知道各种选择的概率分布，这样就将不完全信息博弈转化为不完美信息博弈。

不完美信息博弈（imperfect information game） 不完美信息博弈是指没有参与人能够获得其他参与人的行动信息的博弈，也即参与人做选择时不知道其他参与人的选择。例如同时行动的博弈就是一种不完美信息博弈。

贝叶斯博弈（Bayesian game） 贝叶斯博弈是指参与人对于对手的收益函数没有完全信息的博弈，因此也被称为不完全信息博弈。

纯策略（pure strategy） 在完全信息博弈中，如果在每个给定信息下，只能选择一种特定策略，这个策略为纯策略。纯策略是混合策略的特例。纯策略的收益可以用效用表示。

混合策略（mixed strategy） 在完全信息博弈中，如果在每个给定信息下只以某种概率选择不同的策略，则称为混合策略。混合策略是纯策略在空间上的概率分布。混合策略的收益只能用预期效用表示。

最优反应（best response） 在某个博弈中，对于某个参与人而言，假如其他人所采取的行动是已知或者能被预测的，根据这个已知的或可预测的行动而采取的能使自己的收益最大化的战略，称为这个参与人的最优反应。

最大最小策略（maximin strategy） 最大最小策略，也称为最小最大化策略（minmax strategy），是指博弈的参与人选择使得自己可能获得的最小收益最大化的一种策略。最大最小策略是一种保守的策略，而非利润最大化的策略。博弈的参与人通常只有在信息不完全的情况下才采取最大最小策略。

鞍点策略（saddle strategy） 在所有参与人绝对理性的博弈当中，所有参与人倾向于选择一个平衡的策略，在这个策略下，若某个参与人试图通过修改自己的策略来使自己收益扩大，其他参与人都有相应的反制手段，这样的策略使得所有参与人之间达到了一个平衡，这个策略就成为鞍点策略。

ε-纳什均衡（ε-Nash equilibrium） 纳什均衡，又称为非合作博弈均衡。在一个博弈过程中，对于每个参与人来说，无论其他参与人的策略选择如何，他都会选择某个确定的策略（优势策略），所有参与人都选择各自的优势策略的组合称为纳什均衡。也即如果在一个策略组合上，当所有其他参与人都不改变策略时，没有人会改变自己的策略，则该策略组合就是一个纳什均衡。

1.2 通过文献调研重新复述下列理论

通过文献调研重新复述下列理论：冯·诺依曼-摩根斯坦定理、Sperner引理、布劳威尔不动点定理、最小最大定理、纳什均衡定理。

冯·诺依曼-摩根斯坦定理（Von Neumann-Morgenstern utilities theorem) 冯·诺依曼和摩根斯坦曾经创建期望效用函数即VNM效用函数理论，建立了对不确定条件下对理性人的选择进行分析的框架。在博弈的结果集上定义一个满足完备性、传递性、可代替性、可分解性、单调性和连续性的偏好关系“$\succ$”，则存在一个效用函数，使得在偏好关系上呈现“$\succ$”关系的二者在效用函数上呈现大于等于关系，且对一个不确定性下的收益集合，期望效用函数就是它作为数值函数在该随机变量上取值的数学期望。这提供了一个很好的判断有风险的收益的方法和工具。

Sperner引理（Sperner theorem） 在介绍此定理之前，首先介绍几个概念。若$X_0, ..., X_m$线性无关，则$X_1 - X_0, ..., X_m - X_0$便是仿射无关的。$n$维单纯形就是$(n+1)$个仿射无关点的凸包。对$n$维单纯形的$(n+1)$个顶点赋以标号使得各顶点的标号各不相同。对某个母单纯形进行剖分，规定母单纯形各边上的子单纯形节点不能使用对面顶点上的数字，但使用两端中的哪个都可以。若原先的母单纯形内部存在一个剖出的小单纯形，其各顶点恰好使用了所有标号，则这个小单纯形称为好子形。Sperner引理指出对任何母单纯形，不管怎么进行剖分，其内部一定存在奇数个好子形。

布劳威尔不动点定理（Brouwer fixed point theorem） 对于拓扑空间上的一个凸的有界闭集M，若f是$M \longrightarrow M$的连续映射，则一定存在某个点$x \in M$，使得$f(x) = x$。这个定理可以使用Sperner引理进行证明。

最小最大定理（maximin theorem） 例如在二人零和博弈中，设$N = \{A, B\}$，A采取将自己最小收益最大化的策略，而B采取将A的最大收益最小化的策略是最稳妥的选择。这时双方的策略中一方的最大值与另一方的最小值保持一致，这就是所谓的最小最大定理。

纳什均衡定理（Nash equilibrium theorem） n人有限非合作博弈条件下，纳什均衡必定存在。纳什均衡存在性定理可以使用布劳威尔不动点定理进行证明。

1.3

将石头-剪刀-布二人零和博弈形式化为规范型博弈和扩展型博弈；找到其纯策略均衡和混合策略均衡；当对手的混合策略为$\left(\frac{1}{2},\frac{1}{3},\frac{1}{6}\right)$时，你的最优反应策略是什么,收益是多少？

规范型博弈 设参与人集合$N=\{A,B\}$，对于参与人A和B而言，A可采取的策略集合S_1和B可采取的策略集合S_2，下用r表示石头（rock)，用p表示布（paper)，用s表示剪刀（scissors)，$S_1=S_2=\{r,p,s\}$。则A和B各自的效用函数可以如下表示：

$$
\begin{array}{lll}
u_1(r,r)=0 & u_1(r,p)=-1 & u_1(r,s)=1\\
u_1(p,r)=1 & u_1(p,p)=0 & u_1(p,s)=-1\\
u_1(s,r)=-1 & u_1(s,p)=1 & u_1(s,s)=0\\
u_2(r,r)=0 & u_2(r,p)=1 & u_2(r,s)=-1\\
u_2(p,r)=-1 & u_2(p,p)=0 & u_2(p,s)=1\\
u_2(s,r)=1 & u_2(s,p)=-1 & u_2(s,s)=0
\end{array}
$$

这就是所谓的二人零和博弈，对于两个参与人而言，无论他们采取怎样的策略组合，即$\forall cl\in S_1\times S_2$，总有$u_1(cl)+u_2(cl)\equiv 0$。用表格表示石头剪刀布的二人零和博弈如下：

(A,B)	r	p	s
r	$(0,0)$	$(-1,1)$	$(1,-1)$
p	$(1,-1)$	$(0,0)$	$(-1,1)$
s	$(-1,1)$	$(1,-1)$	$(0,0)$

扩展型博弈 参与人集合仍然是$N=\{A,B\}$，整个扩展型博弈G表示为$G=\{N,S,H,\{u_i\}\}$，其中S表示所有可能的策略集合（但在某些节点上可能只有一部分可以选择的策略)，H表示用来表示策略的序列所构成集合的历史集。下用树的形式具体地表示石头剪刀布的博弈。

实际上，图3和图1的博弈树在形状方面完全相同，只是在参与人B的三个决策节点之间被虚线连接起来，被虚线连接在一起的节点集合称为信息

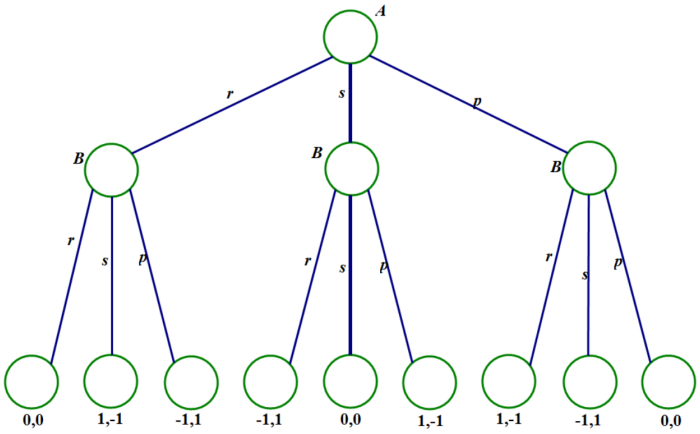

图 1: 可观察情形下（可能是A的意图较为明显或者已经暴露）参与人A先行动的剪刀石头布博弈。

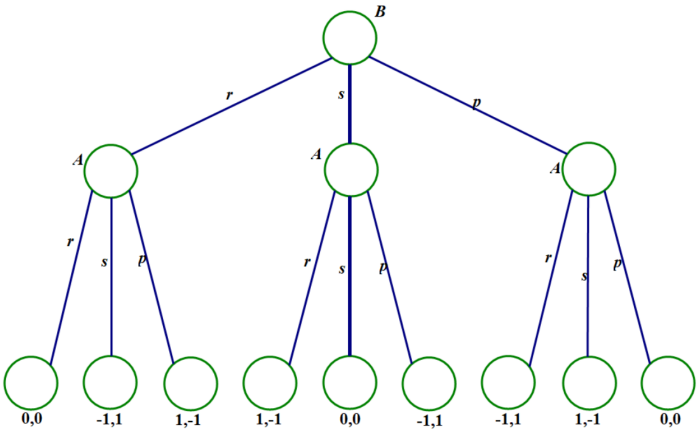

图 2: 可观察情形下（可能是B的意图较为明显或者已经暴露）参与人B先行动的剪刀石头布博弈。

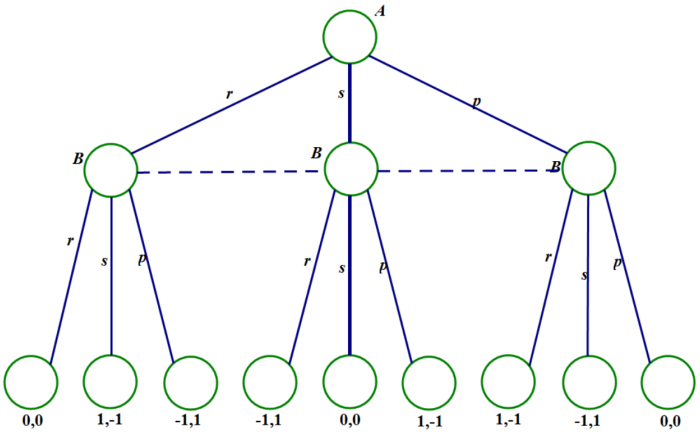

图 3: 不可观察情形下（实际上石头剪刀布的常规情形就是不可观察情形）参与人A先行动的剪刀石头布博弈。

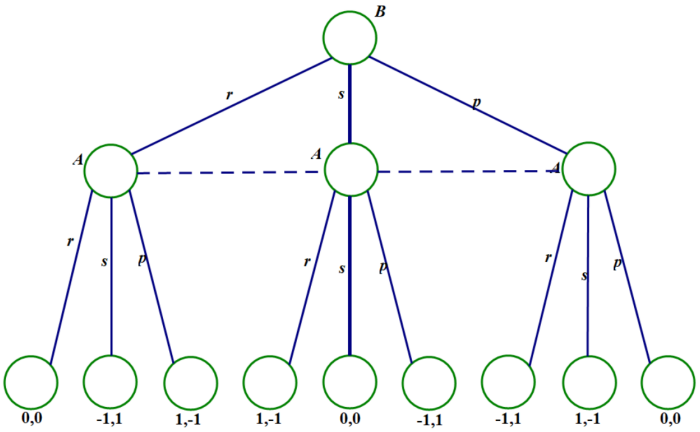

图 4: 不可观察情形下（实际上石头剪刀布的常规情形就是不可观察情形）参与人B先行动的剪刀石头布博弈。

集（information set）。当博弈进行到信息集当中的某个决策节点且轮到某参与人行动时，该参与人不知道自己位于哪个节点上，原因在于他没有观察到前面发生了什么事情。而上述提及的参与人A或B先行动指的是对应的参与人先确定好自己接下来要做的决策。同样地，图4和图2也只存在信息集上的不同。

纯策略均衡 参与人集$N=\{A,B\}$，不妨设参与人A采取纯策略，始终选择石头（r），则 B 观察到此信息可以始终选择布（p），此时A的效用函数为-1，而B的效用函数为1。在这样的博弈重复下，A会发现自己的策略无法获得较高的收益，因此二者相关的策略发生变化。参与人A和B将永远围绕着策略圈相互追逐，因此这样的纯策略是不存在平衡点的。

混合策略均衡 参与人集$N=\{A,B\}$，$x_1=\{x_{11},x_{12},x_{13}\}\in S_1$是参与人$A$所采用的混合策略，其中$x_{11}$表示他采用石头（r）的概率，$x_{12}$表示他采用剪刀（s）的概率，$x_{13}$表示他采用布（p）的概率。同理 $x_2=\{x_{21},x_{22},x_{23}\}\in S_2$ 描述参与人B所采取的混合策略。则可以用效用函数表示参与人A的预期收益：

$$E(u_1)=x_{11}x_{22}+x_{12}x_{23}+x_{13}x_{21}-x_{11}x_{23}-x_{12}x_{21}-x_{13}x_{22}$$

由于是二人零和博弈，因此可以表示出参与人的预期收益：

$$E(u_2)=-E(u_1)=-x_{11}x_{22}-x_{12}x_{23}-x_{13}x_{21}+x_{11}x_{23}+x_{12}x_{21}+x_{13}x_{22}$$

其中$x_{11}+x_{12}+x_{13}=x_{21}+x_{22}+x_{23}=1$,$A$和$B$都可以不断调整自己的混合策略。对上述两式求偏导可以得到结论：

$x_1=\left\{\frac{1}{3},\frac{1}{3},\frac{1}{3}\right\}$,$x_2=\left\{\frac{1}{3},\frac{1}{3},\frac{1}{3}\right\}$是这个博弈的混合策略均衡，两个参与人都会平均地选用三种不同的策略，而两个参与人的预期收益也都为0。

例如当已知参与人A的混合策略为$\left(\frac{1}{2},\frac{1}{3},\frac{1}{6}\right)$时，考虑参与人$B$采用怎样的策略。设$B$采取混合策略 $x_2=\{x_{21},x_{22},x_{23}\}\in S_2$ 。则

$$E(u_2)=-\frac{1}{2}x_{22}-\frac{1}{3}x_{23}-\frac{1}{6}x_{21}+\frac{1}{2}x_{23}+\frac{1}{3}x_{21}+\frac{1}{6}x_{22}$$

通过数学处理可知B只要采用混合策略$x_2=\{t,0,1-t\},\forall t\in[0,1]$就可获得最大收益。

2 基础算法

2.1 遗憾最小化算法

采用遗憾最小化算法求解石头-剪刀-布二人零和博弈的均衡（编程+分析）。

以石头-剪刀-布游戏为例，假设赢得比赛收益为1，输掉比赛收益为-1，平局收益0。显然，最优策略是赢得比赛。当对手出石头时，我们自己出石头，剪刀，布的收益和遗憾值如下表所示：

对手	自己	收益	遗憾值=最优策略收益-当前动作的收益
石头	剪刀	-1	2
石头	石头	0	1
石头	剪刀	1	0

在石头-剪刀-布游戏中，我们并不知道对方接下来会出什么，但是仍然通过历史对局的累计遗憾值来选择动作，累计遗憾值反映了对手历史的策略。假设对手则采取以1/3概率出石头、剪刀和布，而我们自己是一个总结和善于改进的智能体——游戏过程中记录遗憾值，并选择遗憾值最低的动作。随着博弈次数的增加，我们也会逐渐采用1/3的概率选择石头、剪刀和布。此时达到了纳什均衡。代码见附录$A.1$。

2.2 虚拟博弈算法

采用虚拟博弈（Fictitious Play, FP, Brown 1951）算法求解石头-剪刀-布二人零和博弈的均衡（编程+分析）。

虚拟博弈是博弈论中一种传统的方法，其历史真的非常久远，于1951年被Brown, George W 提出。其核心思想非常简单，就是利用博弈论中常用的反应函数思想。使每个智能体拥有两个策略集。一个是最优策略集，一个是历史平均策略集。在每一轮博弈的开始，每个均智能体根据对手的历史平均策略集，找到一个最优的针对策略。然后根据历史平均策略和本轮最优策略更新自己的历史平均策略。

拿石头剪刀布举例子：首先第一轮随机出拳，如果P1石头，P2剪刀。在第二轮时，P1根据P2的历史数据（P2只出了剪刀）得出自己应该出石头，则P1还是出石头，P2根据P1历史数据（P1只出了石头）得出自己应该

出布。所以第二轮P1石头，P2布；玩家更新自己的历史策略集P1还是只出了石头，P2有50%的情况出了剪刀，50%的情况出布，以此类推.....随着迭代的继续，策略会慢慢收敛到纳什均衡。代码见附录$A.2$。

3 算法原理

3.1 对偶线性规划

在众多博弈模型中，占有重要地位的是二人有限零个对策，又称为矩阵博弈，这对策是目前为止在理论研究和求解方法方面都比较完善的一个博弈。矩阵博弈就是二人有限零和博弈，或有限二人零和博弈；在众多博弈模型中占有重要地位，是到目前为止，在理论研究和求解方法方面都比较完整的一类博弈。下面给出求解矩阵博弈的原理：

定义 1 策略型博弈G形如$G=(N,S,u)$，其中$N=\{1,2,...,n\}$为一非空集合，$S=S_1\times S_2\times ...\times S_n$，$u=(u_1,u_2,...,u_n),\forall i\in N$。这里$N$为博弈$G$的局中人的集合，任取局中人$i$，$S_i$为$i$的可用的（纯）策略集合。当进行策略型博弈$G$时，每个局中人$i$需选$S_i$里一策略，$N$里每个局中人所选的策略组成一个策略组合。任取$S$里的策略组合$\alpha=(\alpha_1,\alpha_2,...,\alpha_n)$，数$u_i(\alpha)$表示当$\alpha$为全体局中人实现的策略组合时，局中人$i$在这博弈里得到的收益。

定义 2 考虑$N=\{1,2\}$，设局中人1有m个策略，局中人2有n个策略，则$G=\{S_1,S_2;A\}$，其中$S_1=\{\alpha_1,\alpha_2,...,\alpha_m\}$，$S_2=\{\beta_1,\beta_2,...,\beta_n\}$,构建一个$m*n$的矩阵，对应值为局中人1的收益，记作收益矩阵$A$，表达式为：

$$A=\begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{bmatrix}$$

因为是零和博弈，显然局中人2的收益矩阵$B=-A$。

定义 3 对$\forall G(N,S,u)$，如$\exists(\alpha_1*,\alpha_2*,...,\alpha_m*)$，使得$\forall i$，$\forall\alpha_{ij}\in S_i$，有 $u_i(\alpha_i*,\alpha_{-i}*)\geq u_i(\alpha_{ij}*,\alpha_{-i}*)$，则称$(\alpha_1*,\alpha_2*,...,\alpha_n*)$为$G\{S_1,S_2;A\}$的一个纳什均衡。

定理 1 设有两个矩阵博弈$G_1=\left\{S_1,S_2;A_1=(a_{ij})_{m\times n}\right\}$，$G_2=\left\{S_1,S_2;A_2=(a_{ij}+d)_{m\times n}\right\}$，$d$为常数，则$G_1$与$G_2$有相同的解。

定理 2 任意矩阵博弈$G=\{S_1,S_2;A\}$的求解均等价于一对互为对偶的线性规划问题。设$x^*\in S_1*$，$y^*\in S_2*$，则(x^*,y^*)为矩阵博弈$G=\{S_1,S_2;A\}$的

纳什均衡的充要条件是：存在数V，使得x^*，y^*分别满足：

$$\begin{cases} \sum_i a_{ij}x_i \geq V, j = 1, 2, ..., n \\ \sum_i x_i = 1 \\ x_i \geq 0, i = 1, 2, ..., m \end{cases}$$

$$\begin{cases} \sum_j a_{ij}y_j \leq V, i = 1, 2, ..., n \\ \sum_j y_j = 1 \\ y_j \geq 0, j = 1, 2, ..., m \end{cases}$$

推论 1 局中人1的最优策略等价于线性规划问题：

$$\begin{aligned} \min \quad & Z = \sum_i {x_i}' \\ s.t. \quad & \sum_i a_{ij}{x_i}' \geq 1, \quad j = 1, 2, ..., n \\ & {x_i}' \geq 0, \qquad\quad i = 1, 2, ..., m \end{aligned}$$

局中人2的最优策略等价于线性规划问题：

$$\begin{aligned} \max \quad & W = \sum_j {y_j}' \\ s.t. \quad & \sum_j a_{ij}{y_j}' \leq 1, \quad i = 1, 2, ..., n \\ & {y_j}' \geq 0, \qquad\quad j = 1, 2, ..., m \end{aligned}$$

采用以上线性规划法求解的必要条件是$V \geq 0$。如何判断$V \geq 0$，可以证明，当$a_{ij} \geq 0$时，$V \geq 0$。若某个$a_{ij} \leq 0$，可对A的各元素加上适当的数$d \leq 0$，使所有的$a_{ij} \geq 0$。

由1.3可知石头剪刀布的二人零和博弈,因此可以得到A的收益矩阵

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

为了让线性规划有解，由定理2知，可以令

$$A^{'} = A + E = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{bmatrix}$$

根据对偶线性规划的性质可知

$$B^{'} = A^{T} = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

接着利用线性规划法求解上述矩阵对策，可将其化成两个互为对偶的线性规划问题：

$$\begin{aligned} \min \quad & z = x_1 + x_2 + x_3 \\ \text{s.t.} \quad & x_1 + x_2 \geq 1 \\ & x_2 + 3x_3 \geq 1 \\ & 2x_1 + x_3 \geq 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad & w = y_1 + y_2 + y_3 \\ \text{s.t.} \quad & y_1 + 2x_3 \leq 1 \\ & 2y_1 + y_2 \leq 1 \\ & 2y_2 + y_3 \leq 1 \\ & y_1, y_2 y x_3 \geq 0 \end{aligned}$$

利用单纯形法求解可得最优解：

$$X = (0.3333, 0.3333, 0.3333)$$

$$Y = (0.3333, 0.3333, 0.3333)$$

代码见附录$A.3$。

联系 在石头剪刀布这样的二人零和博弈当中，可以通过对偶线性规划法确定应对对手历史策略的最优的混合策略，也即通过对偶线性规划法得到的结果可以用于确定FP算法当中的迭代方式，如此迭代下去最终可以得到博弈的纳什均衡。

3.2 FP算法的收敛性

利用FP算法分析石头剪刀布博弈，可以最终收敛到其纳什均衡。可以证明FP算法运用到零和博弈或纯策略纳什均衡解的常和博弈都可以收敛，下面证明其收敛性。

由于第一轮为随机决策，不妨假设A出石头，B出剪刀。第二轮决策时，A会根据B的历史数据（B只出了剪刀），得出A自己应当出石头；而B会根据A的历史数据（A只出了石头），得出B自己应当出布。因此第二轮A出石头，B出布。这样历史策略集又得到了更新，A仍然以100%的情况出了石头，而B有50%的情况出了剪刀、50%的情况出了布。以此类推。随着迭代的继续，最终会收敛到纳什均衡。

设在某一轮A的历史策略集以混合策略的形式表示为(x_{11}, x_{12}, x_{13})，其中$x_{11}+x_{12}+x_{13}=1$，B依据A的历史策略决定采取混合策略(x_{21}, x_{22}, x_{23})，其中$x_{21}+x_{22}+x_{23}=1$。下面衡量B的期望收益：

$$\begin{aligned}
E(u_2) &= -x_{11}x_{22} - x_{12}x_{23} - x_{13}x_{21} + x_{11}x_{23} + x_{12}x_{21} + x_{13}x_{22} \\
&= x_{21}(x_{12} - x_{13}) + x_{22}(x_{13} - x_{11}) + x_{23}(x_{11} - x_{12}) \\
&= x_{21}(x_{12} - x_{13}) + x_{22}(x_{13} - x_{11}) + (1 - x_{21} - x_{22})(x_{11} - x_{12}) \\
&= x_{21}(x_{12} - x_{13} - x_{11} + x_{12}) + x_{22}(x_{13} - x_{11} - x_{11} + x_{12}) + (x_{11} - x_{12})
\end{aligned}$$

其中x_{11}、x_{12}视为常数，x_{13}可以用x_{11}和x_{12}表示，而上式中的变量也只有x_{21}和x_{22}，x_{23}已用x_{21}和x_{22}进行表示。B选择使得$E(u_2)$最大的混合策略做出决策。同样地，A依据B此前的历史策略集决定A自己的策略。这样的迭代可以使得最终参与人A和B的混合策略趋向于各自的纳什均衡$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$。

利用[7]当中的方法，石头剪刀布实际上运用的离散情形下的FP算法，即适用于DFP（Discrete-time fictitious play）的情形。设$p(t)$为第t轮A的决策，而$q(t)$为第t轮B的决策，其中$p(0)$和$q(0)$表示初始状态A和B的决策，也就是开始A和B进行的一次随机决策。则这样的DFP表示为$(p(0), q(0)) \in \triangle_m \times \triangle_n$，采用如下迭代方式，可使$A$和$B$的决策趋向于纳什均衡：

$$\begin{aligned}
p(t+1) &\in \frac{t}{t+1}p(t) + \frac{1}{t+1}BR_1(q(t)) \\
q(t+1) &\in \frac{t}{t+1}q(t) + \frac{1}{t+1}BR_2(p(t))
\end{aligned}$$

4 上手科研

4.1

阅读文献[1,2]，将石头-剪刀-布二人零和博弈分解为Transitive Game和Non-Transitive Game，并体会Non-Transitivity的作用。

可传递性博弈（Transitive Game） 单调的博弈（Monotonic games）可以产生可传递性博弈。在可传递性博弈当中引入FFG（Functional-form games）当中的单调函数σ，例如定义可传递性博弈当中的等级函数（rating function），$\phi(v,w)=\sigma(f(v)-f(w)v)$，则对于理性参与人$v$而言，可以根据$w$此前的表现调整接下来的策略以使自己的等级函数尽可能最大化。

非传递性博弈（Non-Transitive Game） 对石头剪刀布博弈，可以得到

$$r_\epsilon=\frac{\sqrt{3}\epsilon}{2}(\cos 0,\sin 0)$$

$$p_\epsilon=\frac{\sqrt{3}\epsilon}{2}\left(\cos\frac{2\pi}{3},\sin\frac{2\pi}{3}\right)$$

$$s_\epsilon=\frac{\sqrt{3}\epsilon}{2}\left(\cos\frac{4\pi}{3},\sin\frac{4\pi}{3}\right)$$

由此可以获得状态转移矩阵

$$A_{\{r_\epsilon,p_\epsilon,s_\epsilon\}}=\begin{bmatrix}0 & \epsilon^2 & -\epsilon^2\\ -\epsilon^2 & 0 & \epsilon^2\\ \epsilon^2 & -\epsilon^2 & 0\end{bmatrix}$$

依据此矩阵可以通过有限次左乘（或右乘）使得博弈回到之前的状态，即可以将重复博弈看成多次甚至无限次的循环。

4.2

阅读文献[3]，通过石头-剪刀-布二人零和博弈的例子复现DO算法。

DO（Double Oracle）算法 Double Oracle算法主要针对有限博弈，在一个有限但可能较大的集合（finite though possibly large set）上进行考虑用于解决一系列双人博弈问题。该算法分别维护双方的策略池，通过目前

已有的信息针对对方的策略池中所有策略做出最优反应，并加入自己新的策略池当中。

Algorithm 1 Double Oracle算法

1: 初始建模和求解均衡：将原博弈以正则博弈的形式建模，DO算法在每一次迭代中求解一个子博弈的均衡；
2: BR：在下一轮迭代中为每个玩家计算针对1中均衡的最优反应策略（由于石头剪刀布博弈是二人零和博弈，对于玩家A，可以用$-A$表示除他之外的其他玩家）；
3: 检验当前是否结束迭代：若计算出的最优反应策略无法进一步提升策略收益，则停止优化，输出当前的子博弈均衡。否则，将求解到的最优反应策略分别扩张到对应不同玩家的策略池当中；
4: 回到步骤2，继续进行。

DO算法在双人博弈分析当中可以保证最终收敛到纳什均衡，但最坏的情况下需要遍历整个策略空间，无法处理复杂的非完美信息博弈。

PSRO（Policy-Space Response Oracles）算法 PSRO算法维护一个所有玩家策略的策略池，之后循环地选定玩家，从他的策略集当中选择出一个策略，固定其他所有玩家此时的策略，之后不断地训练这个策略使得该策略成为一个在其他玩家策略不变的情况下的近似的最优反应，然后将其加入策略集合。如此不断训练，直到达到纳什均衡。

Algorithm 2 Policy-Space Response Oracles算法

1: 初始建模和策略维护：将原博弈以正则博弈的形式建模，维护一个历史的策略池；
2: 求解均衡：基于历史策略池，求解一个混合的历史策略，找到这个策略的纳什均衡；
3: BR：学习一个新的针对均衡策略的最优反应，并加入历史策略池中；
4: 检验和迭代：若当前策略均衡已达到目标，则输出当前的历史策略均衡；否则，回到步骤2和3，继续进行。

在理想情况下，PSRO算法总能找到一个不存在于当前策略池中的新策略用于扩张现有策略池。也即，新的策略总是可以被扩展的，这保证了PSRO算法能够使得理性参与人总是不断地学习更优质的新策略。

注1：DO算法其实可以理解为PSRO算法的表格形式。
注2：DO算法和PSRO算法针对石头剪刀布博弈的博弈都会有较大的过拟合的问题，但由于“$P = NP$”命题的未定性，过拟合是无法避免的。

A 代码附录

A.1 遗憾最小化算法

```
import numpy as np
#定义每局游戏的遗憾值
def yihanzhi(action_opponent,action_agent):
    return 1-payoff[action_agent][action_opponent]
payoff = [#石头剪刀布
            [0,   1,  -1], # 石头
            [-1,  0,   1], # 剪刀
            [1,   -1,  0]  # 布
         ]
#假设对手采取以1/3概率出石头、剪刀和布
p1=(1/3,1/3,1/3)
#定义遗憾矩阵
A=np.array([0, 0, 0])
for i in range(1000):
    action_agent=np.argmin(A)
    action_opponent=np.random.choice(3, 1, p1).item(0)
    A[action_agent]=yihanzhi(action_opponent,action_agent)
                    +A[action_agent]
ans=sum(A)
A=A/ans
print("采取石头、剪刀、布的概率依次为：")
print(A)
```

A.2 虚拟博弈算法

```
import numpy as np
#定义玩家1与玩家2的收益矩阵分别为u0,u1
u0=np.array( [# 石头   剪刀   布
                [0,    1,   -1], # 石头
                [-1,    0,   1], # 剪刀
                [1,    -1,   0]  # 布       ])
u1=-u0
A0=np.array([0,0,0]) #用于存放玩家1的历史策略
A1=np.array([0,0,0]) #用于存放玩家2的历史策略
a0=np.random.choice(3, 1).item(0)
a1=np.random.choice(3, 1).item(0)
A0[a0]=1      A1[a1]=1
B0=A0 #用于存放玩家1的历史平均策略
B1=A1 #用于存放玩家2的历史平均策略
#进行虚拟博弈
for i in range(10000):
    #根据2的历史平均策略，计算出玩家1当前最优的决策
    earn = B1*np.reshape(u0,(3,3))
    money_sum = np.sum(earn, axis=1)
    a0= np.argmax(money_sum)
    #根据1的历史平均策略，计算出玩家2当前最优的决策
    earn = B0*u1.T
    money_sum = np.sum(earn, axis=1)
    a1= np.argmax(money_sum)
    #对玩家1、2的策略集进行更新
    A0[a0]=A0[a0]+1
    A1[a1]=A1[a1]+1
    B0=A0/sum(A0)
    B1=A1/sum(A1)
print("10000次虚拟博弈后，玩家1、2的选择石头、剪刀、布的概率为")
print(B0)      print(B1)
```

A.3 单纯形法

```
from pulp import *
# 1. 建立问题
prob = LpProblem("Problem", LpMaximize)
# 2. 建立变量
x1 = LpVariable("x1",lowBound=0)
x2 = LpVariable("x2",lowBound=0)
x3 = LpVariable("x3",lowBound=0)
# 3. 设置目标函数 z
prob += x1 + x2 + x3
# 4. 施加约束
prob += x1 + 2*x3 <= 1, "constraint1"
prob += 2*x1 + x2 <= 1, "constraint2"
prob += x2 + 2*x3 <= 1, "constraint3"
# 5. 求解
prob.solve()
# 6. 打印求解状态
print("Status:", LpStatus[prob.status])
# 8. 打印最优解的目标函数值
print("z=", value(prob.objective))
# 7. 打印出每个变量的最优值
for v in prob.variables():
    print(v.name, "=", v.varValue)
```

参考文献

[1] Nachbar, J.H. “Evolutionary” selection dynamics in games: Convergence and limit properties. Int J Game Theory 19, 59–89 (1990).

[2] Lanctot, Marc et al. “A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning.” NIPS (2017).

[3] Robinson J . An Iterative Method of Solving a Game[J]. Annals of Mathematics, 1951, 54(2):296-301

[4] Brandt, F., Fischer, F., Harrenstein, P. (2010). On the Rate of Convergence of Fictitious Play. In: Kontogiannis, S., Koutsoupias, E., Spirakis, P.G. (eds) Algorithmic Game Theory. SAGT 2010. Lecture Notes in Computer Science, vol 6386. Springer, Berlin, Heidelberg.

[5] Balduzzi, David et al. “Open-ended Learning in Symmetric Zero-sum Games.” ICML (2019).

[6] Czarnecki, Wojciech M. et al. “Real World Games Look Like Spinning Tops.” NeurIPS (2020).

[7] Chen, Yurong et al. “On the Convergence of Fictitious Play: A Decomposition Approach.” ArXiv abs/2205.01469 (2022): n. pag.

[8] McMahan, H. B. et al. “Planning in the Presence of Cost Functions Controlled by an Adversary.” ICML (2003).

[9] Shoham Y, Leyton-Brown K. Multiagent Systems: Algorithmic,Game-Theoretic, and Logical Foundations. Cambridge, UK:Cambridge University Press, 2008

[10] 周雷,尹奇跃,黄凯奇.人机对抗中的博弈学习方法[J].计算机学报,2022,45(09):1859-1876.

[11] 代佳宁. 基于虚拟遗憾最小化算法的非完备信息机器博弈研究[D].哈尔滨工业大学,2017.