El Farol 酒吧问题和少数派博弈

1. 项目简介：

酒吧问题是美国人阿瑟(W.B.Arthur)教授在 1994 提出的一个经济学上的问题（http://en.wikipedia.org/wiki/El_Farol_Bar_problem）。这个问题是“归纳论证和有界理性”的一个具体例子。该问题中，根据每个人有限的知识和分析能力,通过归纳推理生成一个“反馈循环”：每个人会根据他对其他人行为的预期来做出自己的行为。每个人对他人的行为的预期是基于对他人在过去一段时间的行为分析得来的。 归纳推理会假设在这种反馈的帮助下，这些参与者可以最终有足够知识了解这种博弈，并到达稳定状态。

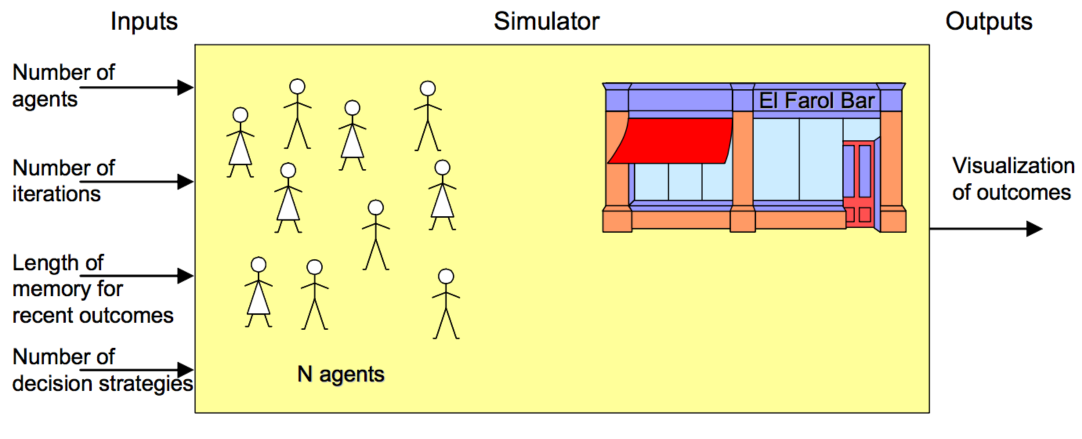

Figure 1: El Farol Bar problem simulation.

该问题的详细阐述（Figure1）：

有 N 个人，他们每个周末都要决定是否去酒吧玩。但是酒吧的座位是有限的，人太多会让酒吧很拥挤，因此大家会觉得玩的不够愉快。我们定义，如果这 N 个人中至多只有 60%的人来了酒吧，那么来了酒吧的人就会玩的很愉快。在去酒吧之前，这群人相互间并没有交流，他们所知道的信息只有过去的几周中，每周都有多少人来了酒吧。每个人做出本周的决定之前是不知道本周有谁会去，也不能看到本周有多少人去了再下决定，因此，我们直接规定，大家的决定是同一时间一起下的。

因为参与者所得到的信息只有最近几周每周来酒吧的人数，所以参与者们不能通过演绎推理来确定他们本周是否去酒吧。有很多合理的数学期望模型可能被应用来对该问题进行建模。由于每个参与者并不知道其他参与者会用什么期望模型，因此一个参与者也没有很好的手段来选择他自己的模型。如果所有人都觉得大部分人都会去酒吧，那么大家都不会去酒吧了，同样的如果大家觉得这周没什么人会去酒吧，那酒吧这周就得爆满了。参与者们没必要知道一共有多少人参与了这场博弈，他们只要知道每周来酒吧的人数的百分比就可以了。现在我们假设一共 100 人，N=100,然后近几周来酒吧的人数如下（按时间线，左边为较早时间）：

……63, 42, 72, 53, 49, 36, 70, 39, 51, 40, 44, 84, 35, 19, 47, 54, 41（当前时间）。以下是几种可能的预测方案：

- 和三周之前一样：47
- 拿上周的数据来和 50 做个镜面对称：59
- 最近 5 周的最小值：19
- 最近三周的人数求个平均取整数近似：48

一个参与者为他的每个预测方案设置一个积分，根据方案是否给出本周正确的决策而给方案一定的打分。

每周一个参与者，用累计得分最高的预测方案来决定他本周是否去酒吧。虽然，这个问题讨论的是非市场环境，但是它提供了一个很好的框架来建立一个简单的市场模型。酒吧问题可以很容易的被扩展到市场场景中。即：每一步中，一个参与者可以购买或者卖出一个资产。在一定时间之后，这种资产的价格就由简单的供需规则所确

定。如果求大于供，则价格会走高，反之，如果供大于求，价格就会走低。如果价格高，那么卖方获胜，如果价格低则买方获胜。因此少数方总能获得胜利。

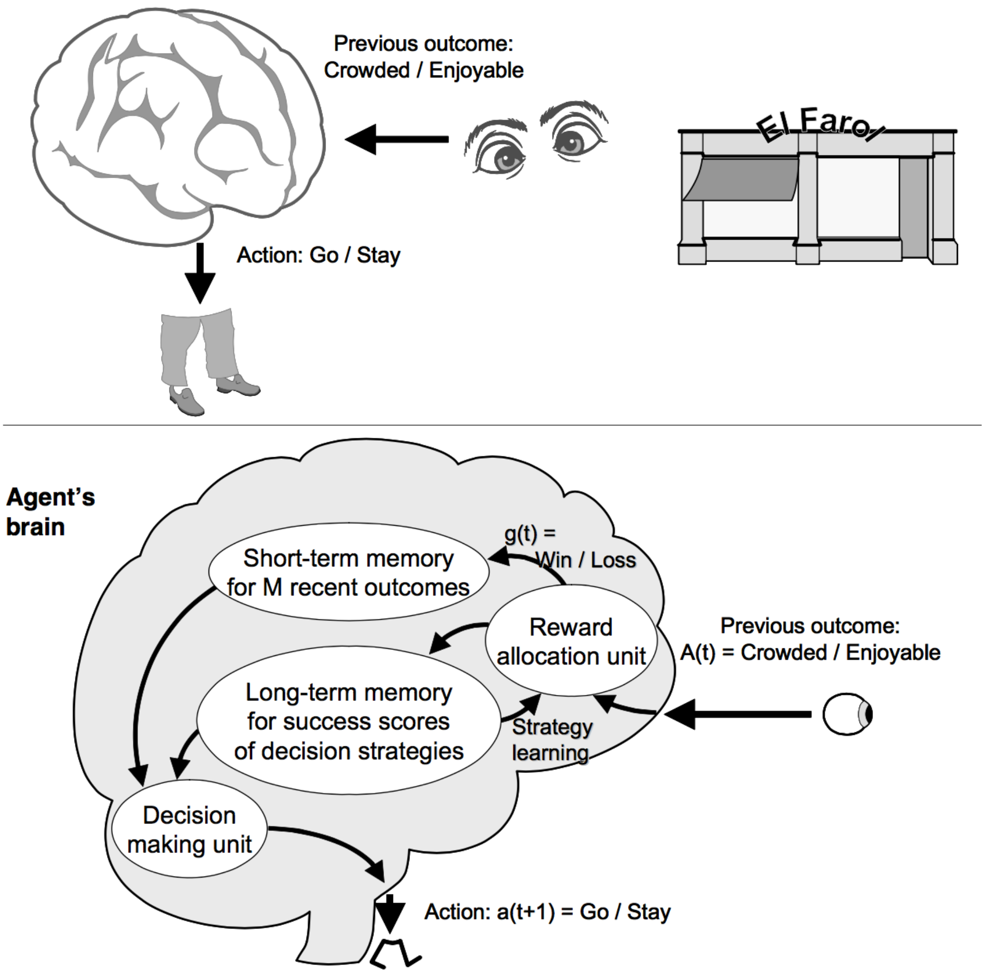

Figure 2: Minority game model of an agent playing the game.

1.1. 少数者博弈（Minority game）

少数者博弈问题是酒吧问题的一种变形，由张翼成和他的团队提出。在少数者博弈中，存在奇数 N 个玩家，每个人必须独立的在每一轮中在两个选项中选择一个。如 Figure 2 的第一行， 每一轮中，每个玩家必须做出决定去酒吧（记作：$a_i(t)=+1$) 还是在家呆着(记作：$a_i(t)=+1$)。博弈的结果是少数一方的行为胜利，而多数的一方失败。我们用下述公式计算在酒吧中的人员差分：

$$A(t)=\sum_{j=1}^{N}a_j(t) \tag{1}$$

由于 N 为奇数，总和 A(t) 不可能会等于 0。A(t)<0 表示，在 t 周，大部分的人呆在家里，因此酒吧里面的人就玩的很“愉快”。反之 A(t)>0 时，在 t 周，少部分的人呆在家里，所以酒吧非常拥挤，大家玩的很“不愉快”。简便其见，我们不去关心到底多少人去了酒吧，只关注哪一方获胜（呆在家里的一方或者是去酒吧的一方）。

第 i 个参与者的结果可以由下述公式来表示：

$$g_i(t) = -a_i(t) \cdot A(t) \tag{2}$$

函数 表示在该轮（周）中，第 i 个参加者行动的奖励，并且保证了只有少数派的行动才会得到奖励。也就是说当 的时候，第 i 个参加者在当前这轮获胜，当 时第 i 个参加者输掉了本轮博弈。 的绝对值表示第 i 个参加者胜利或者失败的程度。我们不用去关心 的具体值，只关心参加者们在博弈中的两种结果，“胜利”（用 1 表示）或者“失败”（用 0 表示）。即：

$$g_i'(t) = \begin{cases} 0 & \text{if} \quad g_i(t) < 0 \\ 1 & \text{if} \quad g_i(t) > 0 \end{cases} \tag{2’}$$

我们假设每个参加者计算能力有限，因此每个人只能记得最近的 M 周的博弈结果，即他们对博弈结果的记忆是“短期记忆”的（Figure2 中下半部分）。当参与者们知道了最新的博弈结果之后，他们所记得离当前时间最久的那周的记录就会忘记，从而保证每个人脑脑子里只记得最近的 M 周的结果，即类似一个移位寄存器，新的一位比特进入之后，最老的一位就会被弹出。 然后大家通过这 M 周结果来对下一周的情况做出决策。

给定一个序列来表示最近的 M 周的博弈情况, 则对于参加者要做的决策，一共有 2^M 种可能的输入。再给定一个策略，用该策略来分析每个包含 M 周纪录的序列，并为这些序列输出一个决策（是“去酒吧”，还是“呆家里”）。比如：M＝3 时，一个历史纪录序列是“赢”，“输”，“赢”，即：101，那么有一个策略给出的该局的决策是“呆家里”。则该策略就用下述形式来表示：

（“赢”，“输”，“赢”）→ “呆家里” 或者 （101） → -1

当然一个完整的策略应该可以对每一种输入的序列都做出该轮行为的决策。因此，当参与者们要分析近 M 周的博弈结果时，他们的策略应该包含 2^M 个条目。每一个条目，包含了一个特定的输入序列（近 M 周的博弈结果）和一个对应的行动决策。Figure 3 是一个例子。

在有 M 周记录的情况下，一共包含了 2^M 种可能的输入，则可能的策略一共有 2^{2^M} 个。当 M＝3，策略一共有 256 种，当 M＝5，策略一共有 4,294,967,296 种。显然，当策略数量过大的时候，参与者们的表现会很糟糕。我们认为，当决策数量 S 大于 8 的时候，参与者们的能力会退化的十分显著。然后，策略数目对在市场模型中的所有操作不会有太大的影响。发生这种情况的原因是，由于备选策略太多，参与者们很可能会非常困惑，因为他们会因为其他的策略看上去比他们当前选的策略稍微多一

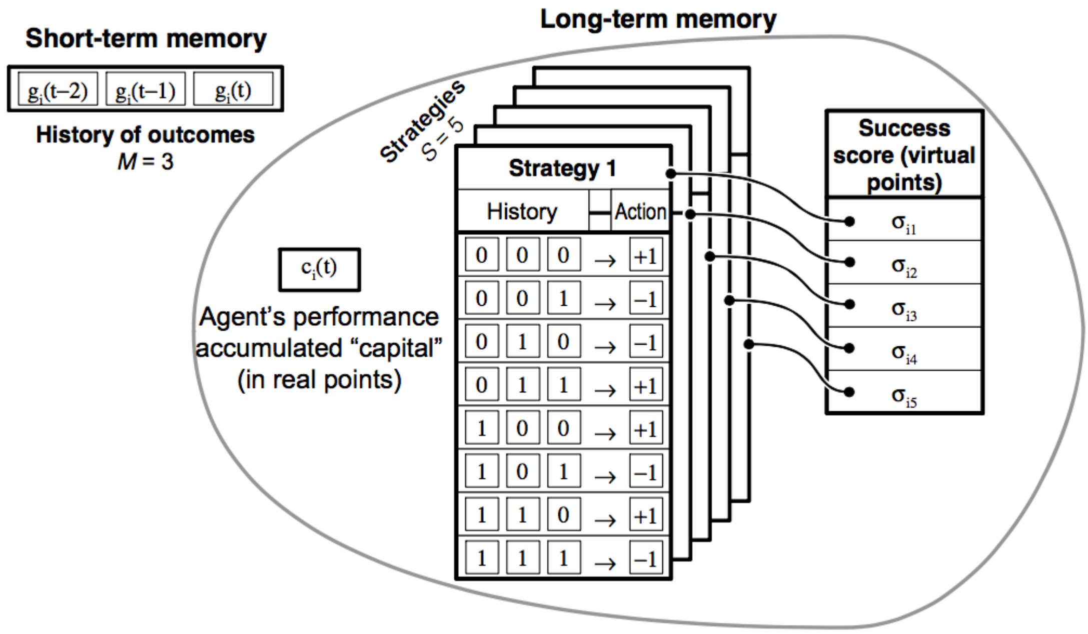

Figure 3: Data memorized by each agent in their short-term and long-term memories. Five strategies are initially selected randomly from the strategy space of 256 possible strategies.

点地优势而频繁的切换策略。设置一个阈值来帮助切换策略可以改善这种情况。

我们假设在博弈的一开始，每个参与者在大小为 2^{2^M} 的策略池中随机的选择了S 种策略，S 远小于 2^{2^M}。举例来说，在 Figure3 中，我们假设每个参与者有 S＝5 中策略。因为每个参与者都是在独立不和他人沟通的情况下选择了他们自己的策略，因此可能出现也可能不出现多人用同样策略的情况。这 S 种策略是被参与者们持久记忆下来的，只要整个博弈还在继续，这 S 种策略就不会被忘记。当参与者进行博弈的时候，每个策略会被打分，打分代表策略对于获得胜利的保障程度。为了尝试从过去的失误中学习，每一轮中，每个参与者会为那些本可以正确预测结果的策略（即那些能把参与者放到少数派的策略）分配一个虚拟的点数。因此，在参与者选择的 S 种策略中，参与者不仅仅会回顾他刚用过的策略，也会回顾那些产生了该轮正确答案的策略。考虑一个参与者 i，和当前第 t 周的情况，用 A(t)表示该周的结果（就是公式1）。对于每个策略 s_{ij} , 1 δ j δ S, 该参与者如下分配虚拟点数。如果策略 s_{ij} 给出的决策是呆在家里 (a_{ij} = 1)并且事实上酒吧非常拥挤(A(t) > 0)，则该策略得到奖励，加上一个虚拟点数。相反，如果这个策略给出的决策是呆在家里，但是酒吧非常空 (A(t) < 0)，则这个策略得分不变（另一种选择是减掉一定的分数）。同样的对于给出决策时去酒吧的情况下（a_{ij} = 1）,根据酒吧最后的情况，为这类策略分配不同的分数。我们将之记为：

$$\sigma_{ij}(t)=\begin{cases}\sigma_{ij}(t-1) & , \quad \text{if} \quad \left(a_{ij}\cdot A(t)\right)<0 \\ \sigma_{ij}(t-1)+1 & , \quad \text{if} \quad \left(a_{ij}\cdot A(t)\right)>0\end{cases}$$

(3)

一开始，所有的策略的打分都被初始化为 0，即：对于所有的 i，j，$\sigma_{ij} = 0$。在第一轮，每个参与者随机选择一个他们持久记忆的策略。比如在 Figure 3 种，这个参与者随机选择了五个策略中的一个。在随后的几轮中，参与者们选择得分最高的策略，并用其做出决策（僵局被随机打破）。由于参与者们记录着他们的策略的表现，为策略打分，并且选用得分最高的策略，因此参与者们也在持续的适应中。

每一个参与者同时累计“资本”来反映总得分 $c_i(t)$, 只有当参与者所选择的策略赢得了新一轮的博弈时，参与者才能获得真实点数。每个参与者都希望最大化自己的“资本”（累计点数）并且他们的表现只通过时间平均获得“资本”量来评估。

总结而言，少数派模拟器如下工作：

1. 初始化，每个参与者初始随机从 2^{2^M} 个可能的策略中选择 S 个策略并对其进行持久记忆。每个策略的得分置为 0，即：对所有的 i,j $\sigma_{ij} = 0$ 。短期记忆序列也进行初始化，可以全部初始化为一个固定的值，如全部初始化为“输”，也可以对序列里面的每一个元素进行随机初始化。

2. 每一个参与者从他的短期记忆中获取最近的历史博弈结果，并且利用这个序列去他的策略集中查找符合该序列的条目。参与者选择当前得分 σ_{ij} 最高的策略(僵局被随机打破)。第 i 个参与者决定在当前这轮（第 t 轮）中采纳他得分最高的策略给出的决策，即： $j = \max\arg_j(\sigma_{ij})$. ,其中, .

3. 每个参与者根据公式（1）接受到当前这轮 t 的结果，A(t)。根据这个结果，参与者计算公式（2）中的奖励函数 $g_i(t)$ 计算奖励，并且记忆到他们的短期记忆区。短期记忆区进行位移，最老的一个值 $g_i(t-(M-1))$ 被从短期记忆去删去

4. 每个参与者根据公式（3）学习并更新他们的策略打分。同时更新其累计 $c_i(t) = c_i(t-1) + g_i(t)$ 。

5. 每个参与者回到步骤 2,，进行新的迭代，除非达到迭代的最大数。

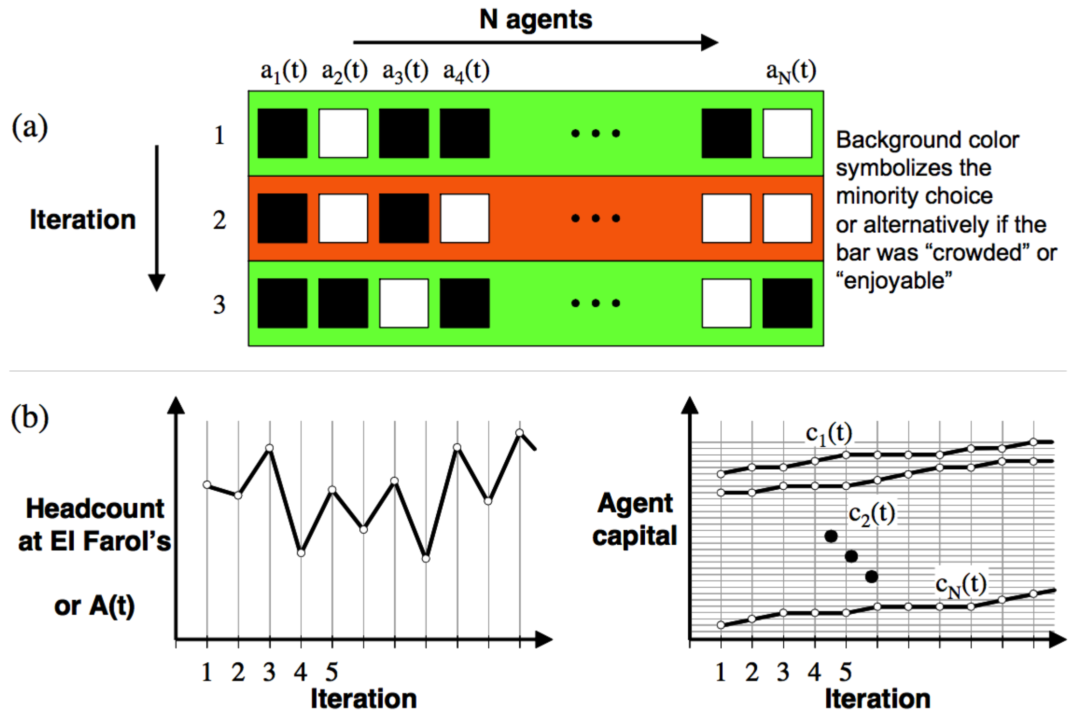

Figure 4: Ideas for output visualization of the outcomes of game turns.

2. 项目要求：

实现一个少数派问题的博弈模拟器，要求如下：

1.1 功能需求

（1）**在系统实现中，提供可视化的交互和输出。**Figure 4 给出了一些对游戏输出可视化的样例。可以考虑设计类似的可视化展示，当然也可以自行设计合理的展现方式。

（2）用户与该博弈模拟器的交互中，应该要包含一个可以打断博弈过程，并且检查每个参与者当前记忆状况（长期记忆部分和短期记忆部分）的功能。

（3）该模拟器应该可以让用户来扮演一个参与者。即：有 N-1 个电脑参与者和 1 个用户扮演的参与者。用户界面上应该要有一些按钮来帮助用户告诉模拟器每一轮中用户的决策（“呆家里”或者“去酒吧”）。该系统同时也应该可以展现酒吧的状态（“拥挤的”或者“愉快的”），（Figure4 可能已经足够完成这些任务）。此外，用户的累积资产应该被高亮从而区分用户和其他电脑参与者。

1.2 其他要求

（1） 使用面向对象分析方法和统一建模语言（UML），编写该模拟器的需求规格说明书。

（2） 使用面向对象设计方法和统一建模语言（UML），编写该模拟器的设计规格说明书。

（3） 在设计的时候，选择并应用合适的设计模式（至少 3 个）。

（4） 编写该模拟器的测试计划和测试用例。

（5） 提交该编译器的源代码（要有合理的注释），以及可执行程序。编程语言可用 C++或 Java 等面向对象程序设计语言。