

基于手牌预测的多人无限注德州扑克博弈方法

李 翔 姜晓红 陈英芝 包友军

(浙江大学计算机科学与技术学院 杭州 310027)

摘 要 作为非完备信息博弈的典型代表,德州扑克一直是人工智能领域内的难题.尤其在多人无限注德州扑克中,博弈策略的制定需要考虑诸多复杂因素,加上其解空间巨大,使问题极具挑战.一般有两种思路解决之:第一种是基于博弈论的方法,通过搜索博弈树、寻找纳什均衡点得到最佳策略;第二种是基于知识的方法,通过学习人类玩家的行动来制定博弈策略.该文的方法属于后者,提出了一种基于牌型预测的德州扑克博弈方法.该方法的基本思想是模拟人类玩家的“读牌”能力.读牌是德州扑克对抗中的重要部分,即根据对手表现出的即时信息及过往的行为习惯,判断对手手牌的大致分布甚至精确牌型.读牌之所以可行,是因为随着牌局发展,对手会进行多次行动,而这些行动往往体现了其手牌信息.文章从非完备信息博弈的角度出发,提出了一套完整的博弈框架,并讨论框架的适用性.随后,将该框架具体应用于德州扑克,将研究重点放在未知信息集的预测上,并采用蒙特卡洛方法计算胜率、得出决策.文章详细地阐述了该方法的设计思想和实现细节,为多人无限注德州扑克程序的设计提供了宝贵的参考.该文是首篇全面论述并设计实现了基于对手手牌预测的多人(超过三人)无限注德州扑克程序的论文.在对手牌型预测上,该文程序比马尔可夫模型的预测精度平均高出 6.65%.在博弈性能上,选择 2015 年华为软件精英挑战赛上的七个程序进行比较,采用锦标赛赛制(允许一次后续买入),两人局比赛的平均胜率为 89%,八人局比赛的平均名次为 1.74.同时在筹码胜负、坚持局数等多项指标上均取得最好成绩.

关键词 非完备信息博弈;人工智能;德州扑克;手牌预测;蒙特卡洛

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2018.00047

Game in Multiplayer No-Limit Texas Hold'em Based on Hands Prediction

LI Xiang JIANG Xiao-Hong CHEN Ying-Zhi BAO You-Jun

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027)

Abstract As a typical example of imperfect information game, Texas Hold'em poker has been an ongoing challenge in artificial intelligence for a long time. Especially in the case of multiplayer no-limit Texas Hold'em, a variety of factors should be considered, combined with its large solution space, making multiplayer no-limit Texas Hold'em AI more challenging. Generally, there are two approaches to handle this problem. The first is game-theoretic method, trying to search the game tree and find out the Nash Equilibrium points with a range of approaches. The second is termed knowledge-based method, learning human player's actions and betting patterns to provide AI with references and make corresponding decisions. In this paper, we propose a gaming policy in multiplayer no-limit Texas Hold'em based on hands prediction, which belongs to the latter category. The basic idea behind our approach is to imitate the process of “hands reading” taken by human beings. Hands reading is an essential skill in playing Texas Hold'em, which refers to predicting the rough probability distribution, or even the precise hands, of the opponents' cards. Hands reading is based on the actions taken by our opponents and the real-time

收稿日期:2015-12-21;在线出版日期:2017-04-06. 李 翔,男,1990 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为云计算、人工智能等. E-mail: lixiang2011@zju.edu.cn. 姜晓红(通信作者),女,1966 年生,博士,副教授,主要研究方向为计算机体系结构、分布式系统、云计算等. E-mail: jiangxh@zju.edu.cn. 陈英芝,女,1988 年生,硕士,主要研究方向为数据挖掘. 包友军,男,1991 年生,硕士研究生,主要研究方向为数据挖掘、模式识别.

information we perceived. As the development of gaming, players will take several action in each betting round (each player will normally take several actions in a single hand in total). The actions taken by players reveal the information of their hands, making hands reading possible. In this paper, we proposes a complete gaming framework from the perspective of imperfect information game. The applicable scope of this game framework is also discussed in the same section. Then we apply the proposed framework to Texas Hold'em. When designing our poker AI, we focus more on the prediction of the unknown information set. At each time point, our poker AI conducts Monte Carlo simulation to obtain the winning rate based on the current collected information and the predicted information. And winning rate is the most important factor our AI considers to make a decision. This paper presents a detailed description of the AI designation and corresponding implementation. We give the readers valuable references to implement their own multiplayer no-limit Texas Hold'em AI. To our best knowledge, this is the first paper that holistically designs and implements multiplayer (more than 3 players) no-limit Texas Hold'em AI based on hands prediction. Experiments are conducted to demonstrate the performance of our policy and the produced AI. In terms of predicting unknown information set, our approach increases the hands prediction accuracy by 6.65% than Markov model. To intuitively illustrate the strength of our AI, we select 7 programs from 2015 Huawei Software Elite Challenge for comparison. Experiments are conducted under various conditions, including heads-up (2 players) and multiplayer no-limit Texas Hold'em. We select tournament as the competition rule, with one extra buy-in and a constant blind. Our program gets a winning rate of 89% in heads-up poker, and places 1.74 in average in eight-player poker. Additionally, it gets the best performance in terms of lasting rounds, winning chips, etc.

Keywords imperfect information game; artificial intelligence; Texas Hold'em; hands prediction; Monte Carlo

1 引 言

人机博弈一直是人工智能领域的重要研究课题. 根据博弈信息是否可以完全可知, 可分为完备信息博弈和非完备信息博弈. 前者指博弈的所有参与者都可以完全地获得所有博弈信息. 例如, 在围棋、象棋和五子棋的对弈过程中, 双方对所有的棋子分布、对手的行动过程都可以完全获知. 非完备信息博弈指参与者无法获得全盘的信息, 仅拥有部分可见的信息集(information set). 例如在军旗或大量扑克游戏中, 任何选手都无法直接查看其它玩家的棋子分布或手牌, 而只能根据当前的局势进行估计, 从而做出决策.

在完备信息博弈中, 极大极小值算法^[1]获得了巨大成功. 该算法对博弈树的每一个节点进行评分, 在假定对手也是以最优策略进行博弈的情况下, 得到一条往后若干(k)步的最优路径. 该方法的主要局

限在于时间复杂度为 $O(x^k)$. 其中 x 是每一个状态下可能的决策数量, k 为算法往后推测的步数. 随着推算步数的增加, 计算时间呈指数级剧增. 为了减少计算时间, 出现了 α - β 剪枝、搜索顺序优化、直接进行状态查找匹配、利用对手思考时间计算、蒙特卡洛搜索^[2-3]等优化方法. 近来, Google 公司开发的 AlphaGo 分别击败欧洲围棋冠军及世界围棋冠军. AlphaGo 综合使用了蒙特卡洛搜索和深度神经网络方法, 基于围棋专家的对弈数据和自身对弈数据对神经网络进行有监督式学习. AlphaGo 的成功标志着人工智能在完备信息博弈领域取得了里程碑式的进展^[2].

在非完备信息的博弈中, 由于无法得知部分信息, 极大极小值中所需的评分过程便无从进行, 也难以推测对手的后续动作. 这使得原有的博弈算法不能直接应用于非完备信息的博弈中. 机器学习方法^[2,4]为非完备信息博弈提供了新的思路, 即先对局面进行特征提取, 再通过大量学习高水平人类玩

家的博弈进行建模. 然后部署到对弈环境中, 根据局势和所建的模型产生决策. 纳什均衡搜索^[5]是另一种常见的解决方法, 将博弈过程抽象成博弈树, 再通过搜索纳什均衡点给出一条决策路径. 纳什均衡点是指博弈双方都不能通过独自改变自身策略从而在博弈中获得更多利益的决策状态. 这在本质上和极大极小值算法类似.

作为非完备信息博弈的典型代表, 德州扑克是人工智能领域内极具挑战的难题. 其中尤以多人无限注^①(multiplayer no-limit)德州扑克最为困难, 具有易学难精、充满人生哲学与决策智慧的特点. 一个优秀的牌手善于发现和总结以下信息, 并综合起来做出决策.

(1) 位置. 一般而言, 后做出行动(in position)的选手比先做出行动(out of position)的选手更具优势. 这是因为后位行动的选手可以在观察前位选手行动之后再决策, 因此能收集到更多信息. 例如翻牌前, 位置越靠后, 公开加注(open-raise)所需牌力越小. 另外, 根据差距原理(gap theory)^[6], 后位跟注一个公开加注却需要更好的牌力. 其次, 在翻牌后, 位置的劣势迫使前位的选手更加谨慎.

(2) 筹码深浅(多少). 因为筹码的数量影响到隐含赔率和反向隐含赔率, 也直接影响到牌手后续的下注强度和操作空间. 根据哈林顿第一原则^[6], 筹码越深, 起手牌的价值差距越小. 所以通常在短筹码博弈中, 选手的起手牌范围较深筹码时小.

(3) 博弈规则. 德州扑克主要分锦标赛和现金局两大类. 两者有着显著的区别. 锦标赛后续买入次数有限或无法后续买入; 现金局不限买入次数. 其次, 锦标赛的盲注通常会不断上涨, 而现金局盲注固定. 另外, 现金局往往会在深筹码下互相对抗, 带来更高的隐含赔率. 这使得在现金局中可以用更多的投机牌入局. 另外, 由于锦标赛的筹码很难支持多次下注. 因此翻牌前和翻牌圈成为底池争夺的主要机会, 这也使得锦标赛选手起手牌范围更小.

(4) 对手的打牌风格. 按入局的手牌范围, 有松紧之分^②. 后者大部分情况只选择很强的手牌入局; 前者除了强牌以外还会掺杂投机牌甚至弱牌入局. 按下注风格, 有凶弱之分. 后者以防守居多, 较少加注; 前者常采取进攻态势, 伴随更多的加注和再加注.

(5) 对手在本局中的下注行为. 对手的下注行为很大程度体现了他的手牌情况. 例如对手进行一个超过底池(pot)的下注可能意味着他拿到了强牌

但又担心对手的反超; 半个底池左右的下注可能是拿到了强牌并希望对手跟注, 我们称之为价值注(value bet).

(6) 手牌和公共牌, 两者的组合可得到自己的牌力^③. 牌力的强弱主要由两方面进行表征: 当前牌力和后续潜力(如果翻牌未完全结束)^[7]. 例如手持两张红桃花色的小牌, 同时牌面也有两张红桃牌. 当前牌力很小, 但有较大可能形成同花, 即后续潜力较大.

德州扑克竞技所要考虑的要素远不止于此. 在这些要素的背后, 读牌能力贯穿始终, 是区分德州扑克竞技水平的关键指标. 理论上, 如果只有一位对手, 除去本方手持的两张手牌和已经发出的 ϵ 张公共牌, 对手手牌的可能性有 h_1 种.

$$h_1 = C_{52-2-\epsilon}^2 \quad (1)$$

在翻牌前, $\epsilon=0$, $h_1=1225$. 假定有 q 个对手, 则手牌可能性的数量 h_q 为

$$h_q = \prod_{i=1}^q C_{52-2i}^2 \quad (2)$$

对于计算机而言, 多人无限注德州扑克由于以下原因使其极富挑战. (1) 多个玩家的手牌, 再加上公共牌的多样性, 使得手牌与公共牌的组合局面极为复杂. 这意味着计算机需要承载巨大的计算量; (2) 加注量的不确定性, 加注量可以位于最低加注额与所有剩余筹码之间的任意值(与有限注德州扑克不同). 在不考虑浮点数情况下, 解空间依然很大. 综上所述, 计算机需要面对其他选手众多的手牌可能性, 利用纷繁的已知条件, 不断缩小对手手牌的范围, 并在巨大的解空间下找到决策方案. 本文致力于该目标, 试图根据对手的下注行为和历史信息推测对手的牌型分布. 然后通过蒙特卡洛方法得到胜率, 并做出决策, 本文的贡献及创新点总结如下.

(1) 提出了一个非完备信息博弈的决策框架. 该框架包括行为理解、对手建模、未知信息集预测和决策制定四个部分, 为未知信息集提供了推测方法. 同时, 文章讨论了该框架的适用范围和需要满足的三个条件: 对手行为可见的动态博弈、对手行为可完美回忆、对手行为依赖我方未知信息集.

(2) 将该非完备信息博弈框架具体应用于多人

① 通常而言, “多人”指两人以上(含两人). 由于以往的工作大多只关注两人和三人的德州扑克博弈, 而本文所提出的方法适用于多于三人的场合. 为将其区分开来, 本文的多人, 特指超过三人的情况.

② 本文涉及的德州扑克术语请参见附录 1.

③ 本文所指的牌力, 是一个相对概念. 我们使用胜率值对牌力进行衡量. 具体地, 我们在第 4 节中介绍胜率的计算过程.

无限注德州扑克中,在行为理解对手建模的基础上,详细介绍手牌的预测过程.在缺乏训练数据的情况下,提出了默认的对手模型以及相应的动态自适应方法.在借助蒙特卡洛方法计算胜率之后,根据具体的博弈环境进行策略选择.本文的牌型预测模型在河牌圈比马尔可夫模型高出 14.08%.

(3)将本文程序与 2015 年华为软件精英挑战赛上的七个风格各异的程序进行比较实验.在与其进行的八人局比赛中,平均名次为 1.74.一对一的比赛平均胜率为 89%.同时在筹码胜负、坚持局数等多项指标上均为最好成绩^①.据我们所知,本文是首篇全面论述、设计并实现了基于手牌预测的多人无限注德州扑克的论文.

本文第 2 节介绍相关工作和研究现状;第 3 节介绍非完备信息博弈的决策框架;第 4 节将框架具

体化到德州扑克博弈中,详细介绍决策流程;第 5 节介绍牌型预测、两人无限注德州扑克、多人无限注德州扑克的实验结果.

2 相关工作

德州扑克按人数可分为两人(heads-up)比赛和多人比赛(multiplayer);按下注限制分为有限注(limit)和无限注(no-limit).其中以两人有限注德州扑克最为简单.Bowling 等人^[8]在 2015 年宣布两人有限注德州扑克已经基本解决.而无限注德州扑克仍然无法完全解决.本节将相关工作分为基于博弈论(game-theoretic)的方法和基于知识(knowledge-based)的方法^[9]进行介绍,并将具有代表性的工作总结在表 1 中.

表 1 扑克 AI 相关工作总结

作者(程序名称)	研究场景	主要方法	实验方法/指标	实验效果
Van ^[3]	两人无限注	• 基于贝叶斯概率模型对手建模 • 基于蒙特卡洛的博弈树搜索	人机博弈	与低水平玩家持平 略差于高水平玩家
Brown (Tartanian) ^[10]	两人无限注	• 反遗憾最小化算法求解纳什均衡点 • 博弈树结点抽象化	ACPC2014 比赛结果	平均获益率为 479 mbb/h
Gilpin (GS3) ^[7]	两人有限注	• 对后续牌力发展敏感的抽象化方法 • 博弈树搜索	• 对比 vexbot、BluffBot 等七个程序 • 永远跟注、加注程序	平均收益 0.071 mbb/h
Bowling (CFR+) ^[8]	两人有限注	• 反遗憾最小化算法求解纳什均衡点	计算可利用系数 (exploitability)	0.986 mbb/h
Heinrich (Smooth UCT) ^[11]	两人有限注 三人有限注	平滑上置信区间搜索	ACPC2014 比赛结果	获得三个第二名
Brown ^[12]	Leduc 扑克	动态结点抽象化的博弈树搜索	计算可利用系数	约 20 mbb/h
Waugh ^[13]	Leduc 扑克	回归反遗憾最小化算法 求解纳什均衡点	计算可利用系数	约 2 mb/h
João ^[14]	两人有限注	• 真人博弈数据分析 • 基于知识的对手建模	无	无
Rubin (SartreNL) ^[15]	两人无限注	• 真人博弈数据分析 • 基于知识的对手建模	ACPC2010 比赛结果	ACPC2010 的第二名
Southey(Thompson, BBR,MAP) ^[16]	两人有限注	• 基于贝叶斯概率模型对手建模 • 分别实现了基于贝叶斯、MAP、 汤普森三种反馈模型用于决策	对比 Opti、Vexbot	水平介于 Opti、Vexbot 之间
Risk (IR16,EX16) ^[17]	三人有限注	反遗憾最小化算法求解纳什均衡点	ACPC2009 比赛结果	IR16 和 EX16 分获第一、第二
本文方法(FOLD)	多人无限注	• 基于条件概率的对手牌型预测 • 默认对手模型及模型动态调整 • 基于蒙特卡洛方法的胜率计算	对比 HC、FANDY 等七个程序	八人局平均名次 1.74

基于博弈论方法的本质是寻找博弈的纳什均衡点^[10].但由于无限注德州扑克手牌及公共牌可能性过多,导致博弈树的规模超出计算机所能承受的极限.即使最简单的两人有限注德州扑克,所有信息集的数量也达到 3.19×10^{14} ^[8].通常有两种简化方法:(1)使用蒙特卡洛树搜索^[3],即无需对整个完整的博弈树进行搜索,也不受博弈树深度的

影响,而是按照一定概率分布进行随机取样分析.该方法也同样应用于完备信息博弈中^[2];(2)对牌型进行抽象化^[12,18].例如按照胜率进行聚类.Andrew 将博弈过程分成两个部分:第一部分包括翻牌前、翻牌和转牌阶段;第二部分为河牌阶段,再引入状

① 基于本文方法实现的博弈程序获得该赛事杭厦赛区初赛第一名、复赛第五名、最后一次模拟赛全国总冠军等成绩.

态空间算法和整数规划对牌型进行抽象化^[19]。文献[7]引入了潜力的概念,并对上述抽象方法进行了优化。他们实现的 AI 程序名为 Tartanian,其后续版本见于文献[19-20]等。由于下注量变化很大,通常需要将下注方式进行离散化处理。例如将下注分为 1/2 池、满池及超池等。抽象化之后,需使用各种方法搜索该博弈问题的纳什均衡点。例如文献[10]使用涅斯捷罗夫的过量间隙技术(Nesterov's excessive gap technique)求解均衡点;文献[11]则使用平滑上置信区间搜索(smooth UCT search),并取得一定成功。纳什均衡点的其它相关工作可参见文献[12-13,21-22]。

基于博弈论方法的主要局限在于:(1)由于方法隶属于博弈论和计算机的交叉领域,因此在将纳什均衡的思想应用于具体的博弈规则中时,理论艰深、实现复杂;(2)博弈树需要同时将对手牌型、公共牌、行动三个彼此独立的对象进行建模,而且每个对象都属于博弈树单独的一层。即使有将其横向压缩的方法^[12,19],博弈树仍然存在计算规模、计算时间难以控制的问题。这是该方法至今没有广泛应用在多人无限注德州扑克的主要原因。

基于知识的方法主要是通过记录、学习职业选手的博弈数据,从中提取出固定的策略并指导决策^[4,23]。Billings 等人的做法是:在翻牌前通过计算胜率值,结合专家系统得出决策;翻牌后使用穷举法评测手牌强度,并加入了一定的高级打法(如陷阱和诈唬)。穷举法存在的问题是计算量过大,由第 4.4 节可知,蒙特卡洛模拟可以有效解决该问题。文献[14]从职业选手的博弈数据出发,重点放在翻牌前的策略上,探讨了若干因素(例如位置、牌型等)对决策的影响大小。文献[15]则提出了完整的基于知识学习的博弈系统。文章提取牌力、潜力、对手下注列表和套池率作为特征,对职业选手的博弈进行提取分析。AI 在对战中将实际情况与数据库记录做对比并计算相似度,依照相似度高的记录做出决策。文献[24]使用神经网络对这些特征进行建模、得出决策。与文献[15]不同的是,其模型的训练是基于遗传算法的搜索。其程序仅与最基本的 AI(如全是弃牌、全是加注的程序)进行比较,缺乏说服力。

基于知识的方法的主要困难是需要大量的文本化或结构化数据,而在部分博弈游戏中,甚至没有职业选手,也没有大量文本化信息可供分析和学习。所谓的文本化信息,是指记录选手(真实玩家或计算机

AI 程序)行为、牌局信息、筹码胜负,存储在数据库或文件,可供读取和解析的信息。例如每年的 ACPC (Annual Computer Poker Competition)都会将博弈数据公开。但由于 ACPC 是有限注博弈,且仅限两人和三人德州扑克,不能满足多人无限注德州扑克程序的学习进程。文献[25]使用 ACPC 的两人有限注德州扑克比赛的数据记录,对 slumbot 等程序进行建模,尝试预测玩家的手牌,达到了较理想的预测精度。近年来,随着服务器(集群)计算能力的提升和大数据的出现,深度学习引起业界重视。采用深度神经网络对职业选手的行为进行学习取得了令人瞩目的成就^[2]。而德州扑克领域缺乏大量的训练数据,使得该方法至今鲜见,也成为该方法的主要局限和亟待解决的问题之一。

此外,基于知识的方法需要在特征数量和数据量之间做权衡。特征越多,所需训练数据越多;特征越少,反映真实博弈情况的能力越弱,博弈水平越低。无论是基于知识的方法,还是基于博弈论的方法,都存在对手牌型预测的需求。而大多数工作都假定对手的牌型属于平均分布,或者穷举对手手牌的所有可能来构造博弈树。这将使问题规模陡增。文献[3]所述的手牌预测过程与本文接近,但文章并没有将对手手牌分布与胜率计算相结合,同时存在以下局限:(1)对手模型需要线下的大量学习,如果没有大量可供学习的数据则无法完成预测;(2)文章仅仅探讨了两人德州扑克,没有将手牌预测和多人德州扑克相结合。本文所述的方法隶属于基于知识的方法,因此仍然存在数据支持度的问题。例如在德州扑克中,每位选手的风格各异,在缺乏历史数据支持的情况下,对每位选手进行快速建模极具挑战。为解决该问题,本文提出了默认的对手模型,同时增加了动态的自适应方法增加模型的鲁棒性和实用性。另外,文章将牌型预测与基于蒙特卡洛方法的胜率计算相结合,在时间复杂度、可扩展性、可控性上具有明显优势。

3 非完备信息博弈框架

3.1 框架设计

如图 1 所示,框架包含四个主要模块:行为理解、对手建模、未知信息集预测和决策制定。这四个模块体现了非完备信息博弈的基本要素以及决策制定的大致流程。

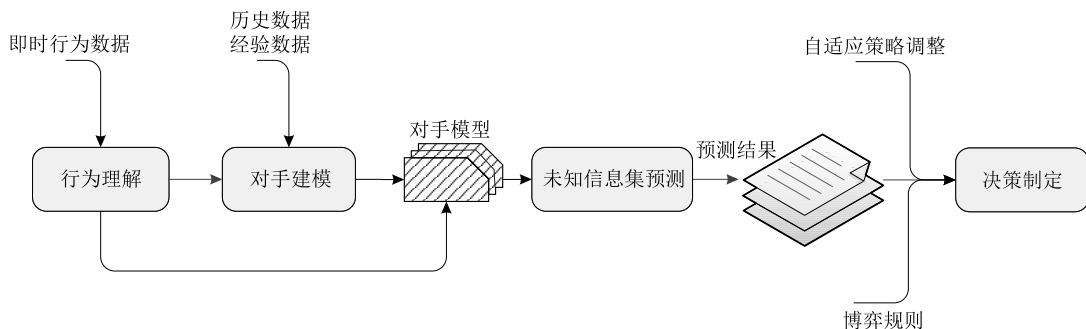


图 1 非完备信息博弈的决策框架

(1) 行为理解. 同样的行动, 在不同情境下有不同的理解. 例如在德州扑克中下注 500, 如下注前底池为 1000, 则只是中等额度下注; 如下注前底池只有 200, 则是非常重的超底池下注. 只有理解对手的行为之后, 才可能挖掘行为背后的意图. 例如, 一个加注频率极低的选手进行加注往往代表持有强牌. 而经常加注的较凶选手进行加注通常仅有中等牌力甚至是诈唬(bluff). 高水平的德州扑克选手会不断平衡自己的打法, 即让同样的行为具有多种解释.

(2) 对手建模. 我们将其概括为: 对手在特定环境下做出各种行为的概率分布, 也即博弈风格. 一般而言, 选手的博弈风格具有连续性. 比如相互熟悉的对手之间进行博弈, 更能预测对方的手牌. 德州扑克的在线统计软件 HM2^①, 则是根据扑克网站的数据, 对线上玩家的各项数据进行统计, 包括选手的翻牌前再加注(3-bet)概率、弃牌率或起手牌范围等等. 这些数据为线上用户提供了宝贵的信息, 对了解对手风格、理解对手行为、制定博弈策略提供了极大帮助.

(3) 未知信息集预测. 该模块的任务是以即时信息和对手模型为输入, 以未知信息集的预测结果或大致分布为输出. 由于每位对手风格不一, 因此手牌的预测依赖于对手历史行为的收集与建模. 我们假定任何对手都存在一个既定的对手模型. 如果我们无法从对手过往的历史数据中进行学习, 那我们则根据大致的经验来假定一个默认模型. 我们将在 4.2 节进行详细阐述.

(4) 决策制定. 对未知信息集的预测通常无法完全精确. 大多数时候只是得到一个大致概率分布. 该分布对决策制定有关键的参考意义. 例如, 在军旗博弈中, 知道对手某棋子的大致强弱可以推算我方棋子与对方棋子交锋时的胜率. 在德州扑克中可以根据对手手牌分布推算手牌的胜率. 除此之外, 影响决策制定的还包括其它因素. 例如在桥牌中选手的打牌风格, 在德州扑克中双方筹码的深浅等. 这

些因素我们统称为博弈环境. 一个优秀的博弈程序应该可以对不同的环境进行识别并自动调整博弈策略.

3.2 框架适用性

3.1 节提出的决策框架适用于诸多非完备信息博弈系统, 但需满足以下特征.

(1) 对手行为可见的动态博弈, 无论是对手建模阶段, 还是对手行为理解与分析阶段, 都以对手行为可见为前提. 所谓的动态博弈, 是指参与人需要进行阶段性、顺序性决策的博弈. 例如在囚徒困境(prisoner's dilemma)^②的博弈中, 囚徒双方都不能看到对方的决定, 且没有阶段性和顺序性, 便不满足该假定. 而在大量扑克游戏中, 例如奥马哈、桥牌、德州扑克以及军旗等博弈中, 由于选手是交替轮流行动, 并且行为可见, 所以满足该条件.

(2) 行为可完美回忆. 博弈的完美回忆(perfect recall)是指在一次博弈中, 我们对其他选手和自己的过往行为是可以完全记住的. 现实生活中, 我们进行扑克游戏时往往不能记住自己 and 他人出过的牌. 这是不具有完美回忆特点的典型例子. 对于动态博弈而言, 选手的历史数据是局面分析的关键.

(3) 对手行为依赖我方未知的信息集. 在德州扑克的博弈中, 未知信息分为两部分: 对手手牌和暂未发出的公共牌. 对于公共牌而言, 是从剩余牌中以完全随机的方式抽取的, 并不是对手行为的主要依据. 另一部分信息是对手的手牌. 根据引言部分的分析可知, 对方的行为依赖其手牌情况, 这是本文假定的前提之一. 也就是说, 对方行为必须在一定程度上体现其信息集, 才使得该信息集具有预测的可能性.

4 基于手牌预测的德州扑克博弈

在本节中, 我们将提出的非完备信息博弈框架

① <http://www.holdemmanager.com>.

② https://en.wikipedia.org/wiki/Prisoner%27s_dilemma

具体化到多人无限注德州扑克中,分行为理解、对手建模、牌型预测、胜率计算和决策制定五个部分介绍我们的系统.我们的方法对于博弈的人数没有限制,不仅适用于两人德州扑克,同时可以扩展到多人德州扑克.

4.1 行为理解

对手的行为有过牌、加注、跟注和弃牌(全押属于跟注或加注).在无限注德州扑克中,如式(3)所示,加注额 C_{raise} 不能低于本轮上一位选手的加注额 $C_{\text{last_raise}}$.如果率先下注,加注额不能低于大盲注 C_{BB} .

$$\max(C_{\text{BB}}, C_{\text{last_raise}}) \leq C_{\text{raise}} \leq C_{\text{total}} \quad (3)$$

其中, C_{total} 是在加注前剩余的筹码.例如某选手 $C_{\text{total}} = 1000$,大盲注为 20,且假定筹码取整数.他在翻牌圈率先加注的话,有 981 种合法加注额.如果进一步考虑其他选手的后续行为,会使得博弈树的节点数量极其庞大.通常的 AI 程序,都采用化零为整的下注模型^[22].在翻牌前,加注行为分为加注(bet)、再加注(3-bet)、更多次加注(4-bet 及以上).翻牌后将加注行为分为:1/2 底池、3/4 底池、满底池、超底池等.在现实的博弈甚至高水平博弈中,这种归类方法已经足够完备.我们现对翻牌后(包括转牌与河牌)的加注进行简要介绍.

(1) 1/2 底池(half a pot).在翻牌前,加注额一般以大盲注为基准.翻牌后则以底池为基准,这样做的原因是给对方相应的赔率(pot odds).例如两位选手正处在翻牌后博弈阶段,选手 A 率先下注 1/2 底池,则对手跟注的赔率为 1 : 3.

(2) 3/4 底池.这是一个较重的下注.给对方的赔率为 3 : 7,在翻牌圈,该赔率略大于两个高张的击中顶对的概率;在转牌圈,该赔率则大于绝大部分听牌(包括同时听顺子和同花)的成功率,因此是阻止对手听牌的有效手段.

(3) 满底池或超底池.在某些特殊情况下,选手会采取这样激进的博弈策略.例如己方有强力听牌;或手持边缘牌力的成牌且担心反超;或者已有强牌但面对对方加注,试图激进地取得更大价值.这种打法很容易牵涉到自己的全部筹码,某些情况下基本相当于全下.

$$\alpha = \begin{cases} 1/2 C_{\text{pot}}, & 0 \leq C_{\text{raise}} < 0.6 C_{\text{pot}} \\ 3/4 C_{\text{pot}}, & 0.6 C_{\text{pot}} \leq C_{\text{raise}} < C_{\text{pot}} \\ C_{\text{pot}}, & C_{\text{raise}} > C_{\text{pot}} \end{cases} \quad (4)$$

对手的下注不会总是按照上述的比例进行,因此我们需对其进行归类.如果对手的加注额为 C_{raise} ,则我们理解的行为 α 采用式(4)所示的映射关

系得到.其中, C_{pot} 是当前底池的筹码数量.一局博弈包括至多四个轮次,每个轮次又可能包含多次行动.为求简单化,我们将每个轮次的行动看成一个动作,取每一轮最激进的动作作为该轮的行动.例如“过牌-加注”被看成“加注”.

4.2 对手建模

我们将对手建模分为两部分:手牌范围建模和行为习惯建模.前者研究的是选手会手持什么样的牌入局;后者研究的是入局后在不同情况下会采取何种行动,即对手在特定的手牌下,记为 ϕ_i ,做出行为 α_i 的概率,记为 $P(\alpha_i | \phi_i)$.

所谓手牌范围,是指对手在翻牌前(pre-flop)选择起手牌的松紧程度.同一个选手在不同位置或筹码深度下会不断调整其手牌范围,但一般而言,该范围还是有规律可循的.决定手牌范围的最主要因素是位置.我们将位置分为:前位、中位、后位和盲注位四种.在不同情境中,位置分类的精细程度需要具体权衡.例如盲注位又可分为大盲位(big blind)和小盲位(small blind).分类越精细,往往效果越好,但对手建模所需的训练数据也越多.我们对每一位选手的每一种位置下的入场率进行统计,即入场率是一个四元组:

$$P_{\text{in}} = \{P_{\text{front}}, P_{\text{mid}}, P_{\text{rear}}, P_{\text{blind}}\} \quad (5)$$

入场率指的是选手在翻牌前通过加注、跟注或者过牌进入到翻牌圈的概率.德州扑克的起手牌可以通过胜率计算或实际经验进行强弱排序^①.例如在八人局中,某选手在后位进入翻牌圈的概率为 35%.那么在实际博弈中,该选手在后位再次进入翻牌圈时,我们会认为其手牌排在手牌序列前 35% 的可能性非常大.

对手建模的第二部分是行为习惯建模,是指对手在入局之后,持有特定牌时习惯做出的动作,或做出特定动作的概率.牌型大致分为五种:超强牌、强牌、中等牌、弱牌和听牌.其实,上述的手牌范围建模也可以归并到行为习惯建模中.例如,翻牌前手持 $\{K\spadesuit, K\clubsuit\}$ 可以看成是超强牌.为简便起见,本文统一采用“行为习惯建模”的方法,并应用于牌型预测中.

文献[3]先将牌力进行分类(bucketing),再通过机器学习的方法不断训练对手模型.该方法需要有足够多的训练数据.在数据量不足、对手陌生时,

① <http://www.preflophands.com/>

我们给出一个默认的基准模型 DFModel. 图 2 以两人底池为例进行说明. 假定对手手牌为 ψ_i , 对手的行为是跟注, 当前公共牌为 Δ_0 . 通过蒙特卡洛方法 (此处我们使用平均分布进行撒点) 得到其胜率为 $P_{\text{win}}(\psi_i)$. 如果 $P_{\text{win}}(\psi_i)$ 恰好落在图 2 下方的“跟注”区间内, 则概率相对值取 $P_{\text{rel}}(\psi_i) = \theta$ (例如为 50). 该区间如图中双箭头所示, 称为完美匹配区间, 用 $\text{domain}_{\text{call}}$ 表示. 此例中, $\text{domain}_{\text{call}} = [0.57, 0.68]$. 如果 $P_{\text{win}}(\psi_i) \notin \text{domain}_{\text{call}}$, 则将胜率 P_{win} 与 $\text{baseline}_{\text{call}}$ 比较 (本文取 0.61), 概率相对值 $P_{\text{rel}}(\psi_i)$ 如下计算:

$$P_{\text{rel}}(\psi_i) = \begin{cases} \frac{\text{baseline}_{\text{call}}}{\text{baseline}_{\text{call}} - P_{\text{win}}(\psi_i)}, & P_{\text{win}}(\psi_i) < \text{baseline}_{\text{call}} \\ \frac{1 - \text{baseline}_{\text{call}}}{P_{\text{win}}(\psi_i) - \text{baseline}_{\text{call}}}, & P_{\text{win}}(\psi_i) > \text{baseline}_{\text{call}} \end{cases} \quad (6)$$

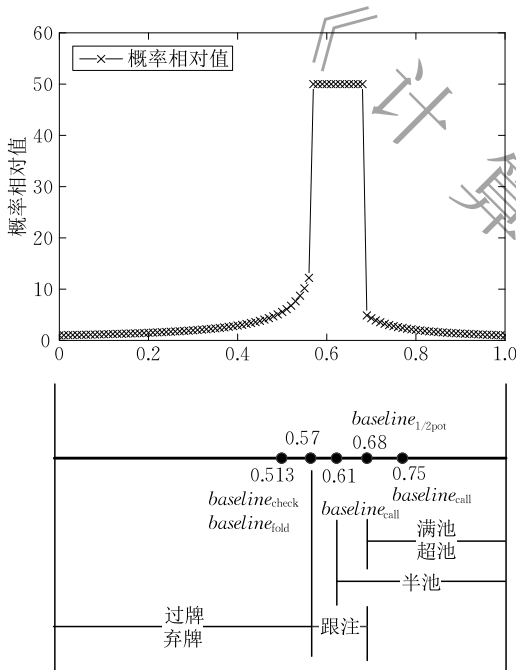


图 2 默认的对手模型 DFModel

也就是说, $P_{\text{win}}(\psi_i)$ 如果落在完美匹配区间, 相对值 $P_{\text{rel}}(\psi_i)$ 取最高值 θ , 否则, 胜率 $P_{\text{win}}(\psi_i)$ 越接近 baseline , 相对值 $P_{\text{rel}}(\psi_i)$ 越高. 图 2 上侧的曲线表示两人底池下, 对手行为是跟注时, 其各种手牌 (每一种手牌对应一个胜率 $P_{\text{win}}(\psi_i) \in [0, 1]$) 的概率相对值 $P_{\text{rel}}(\psi_i)$. 例如对手手牌胜率为 0.55 时, 对手跟注所对应的 $P_{\text{rel}}(\psi_i)$ 约为 10. 类似地, 如果对手行为是过牌, 我们同样按照上述方法得到相应的概率相对值. 最后, 我们将手牌 ψ_i 的所有可能的行为进行归一化处理, 即可得到 $P(\text{call}|\psi_i)$ 、 $P(\text{check}|\psi_i)$ 等等. 至此, 我们回答了本节开始所提出的问题, 即对手在特定手牌下做出各种行为的概率.

此外, 对于陌生的对手 (即没有历史数据可供训练), 我们将实时地对各项参数进行统计, 并纳入概率相对值计算的考虑范围. 例如, 当我们发现对手的跟注频率过高时, 我们会将该对手的 $\text{domain}_{\text{call}}$ 的下限降低. 这意味着对手同样的跟注行为, 所代表的牌力较弱. 相反地, 如果我们发现对手经常弃牌, 那我们会提高 $\text{domain}_{\text{call}}$ 的下限. 因为这样的对手一旦跟注, 说明他的牌力较强. 因此, 我们的程序可以动态地学习对手的下注模式, 优化程序自身的博弈水平.

4.3 牌型预测

假定某时刻, 已发出的公共牌 (community cards) 为 Δ_i , 是一个五元组, 每个元素代表一张扑克牌. 未发出的公共牌置为空. 例如在翻牌圈, δ_{i4} 和 δ_{i5} 便为空值.

$$\Delta_i = \{\delta_{i1}, \delta_{i2}, \delta_{i3}, \delta_{i4}, \delta_{i5}\} \quad (7)$$

假设对手手牌为 ψ_i (未知), 是一个二元组.

$$\psi_i = \{\varphi_{i1}, \varphi_{i2}\} \quad (8)$$

从 4.2 节的对手模型中, 我们可以知道对手在特定的手牌 ψ_i 下, 做出行为 α_i 的概率 $P(\alpha_i|\psi_i)$. 我们先考虑对手单一轮次行为与手牌概率分布的关系, 再拓展到对手多轮行为的牌型预测. 这里, 我们假定对手的行为只依赖于他的手牌 (公共牌是默认已知的信息). 这虽与实际情况有所出入, 但如将影响对手行为的诸多因素全部予以考虑, 会使得逻辑极为复杂甚至失去可行性.

我们的任务是对对手牌型进行预测, 即已知对手的行为是 α_i 的情况下 (这里所指的对对手行为, 是已经发生的行为, 不包括将来的行为, 因为对手行为可见, 故为已知条件), 预测手牌的概率分布. 如式 (9) 所示, $\mathbf{p}_{\text{distribution}}$ 包含 $C(50-\epsilon, 2)$ 个元素, 代表作出行为 α_i 的情况下, 手牌分别为 $\psi_1, \psi_2, \dots, \psi_{C(50-\epsilon, 2)}$ 的概率. $\mathbf{p}_{\text{distribution}}$ 的所有元素之和为 1.

$$\mathbf{p}_{\text{distribution}} = \{P_{\psi_1}, P_{\psi_2}, \dots, P_{\psi_{C(50-\epsilon, 2)}}\}, \quad \text{s.t.} \quad \sum_{i=1}^{C(50-\epsilon, 2)} P_{\psi_i} = 1 \quad (9)$$

其中任一元素 P_{ψ_i} 如下式所示, 代表对手的行为是 α_i 的情况下, 手牌为 ψ_i 的概率值.

$$P_{\psi_i} = P(\{\Delta_i, \psi_i\}|\alpha_i) \quad (10)$$

因为公共牌 Δ_i 是既定事实, 属于已知条件范畴, 故

$$P(\{\Delta_i, \psi_i\}|\alpha_i) = P(\psi_i|\alpha_i) \quad (11)$$

根据式 (10)、(11) 和条件概率原理有

$$P_{\psi_i} = \frac{P(\psi_i) \times P(\alpha_i | \psi_i)}{P(\alpha_i)} \quad (12)$$

其中 $P(\alpha_i | \psi_i)$ 由 0 节对手模型得到. $P(\psi_i)$ 是发牌时, 发到 ψ_i 的概率. 由于发牌完全随机, 故对于任何手牌 $\psi_1, \psi_2, \dots, \psi_{C(50-\epsilon, 2)}$, 该概率为固定值. $P(\alpha_i)$ 是在当前的公共牌下, 选手做出行为 α_i 的概率. 同样地, $P(\alpha_i)$ 对所有手牌而言完全一样. 因此有

$$P_{\psi_i} = H \times P(\alpha_i | \psi_i) \quad (13)$$

其中, H 是对任何手牌恒定的常量. 我们依据上述的方法, 对每一种可能的手牌 $\psi_k \in \{\psi_1, \psi_2, \dots, \psi_{C(50-\epsilon, 2)}\}$ 进行计算, 得到相应的 $P\psi_k$. 再根据式(9)的约束条件进行归一化处理, 得到我们的牌型预测结果. 由于进行了归一化操作, H 的值便无需计算.

$$P_{\psi_i} = \frac{1}{\sum_{k=1}^{C(50-\epsilon, 2)} P(\alpha_i | \psi_k)} \times P(\alpha_i | \psi_k) \quad (14)$$

至此, 我们已经根据所建立的对手模型和条件概率, 推导出对手所有可能手牌的概率分布. 但是, 该分布仅考虑单一轮次. 一局至多包括四个轮次. 在每一轮次中, 我们都可以根据上述方法计算单个轮次的概率分布 $p_{\text{distribution}}$. 假定目前为止博弈进行了 γ 轮, 则对这 γ 个概率分布进行点乘, 得到对手牌型的

最终概率分布的相对值. 具体如式(15)所示.

$$p_{\text{distribution}} = \prod_{k=1}^{\gamma} p_{\text{distribution}}^k \quad (15)$$

即

$$P_{\psi_i} = \prod_{k=1}^{\gamma} P_{\psi_i}^k \quad (16)$$

其中 $p_{\text{distribution}}^k$ 是第 k 轮的牌型概率分布, $P_{\psi_i}^k$ 是其内部元素的值. 由于对每个轮次进行了点乘, 便不再满足式(9)的约束条件. 也就是说, P_{ψ_i} 此时是每种手牌的概率相对值, 需对其进行归一化^①:

$$P_{\psi_i} = P_{\psi_i} / \sum_{k=1}^{C(50-\epsilon, 2)} P_{\psi_k} \quad (17)$$

为了更清晰地解释该预测方法, 我们给出一个具体实例. 如图 3 所示, 我们的手牌为 $\{A\heartsuit, J\heartsuit\}$. 对手在枪口位, 翻牌前加注; 我们在庄位跟注; 其他人均弃牌. 我们进入一个两人底池, 公共牌为

$$\Delta = \{A\clubsuit, T\spadesuit, 4\heartsuit, Q\clubsuit, 5\clubsuit\} \quad (18)$$

对手在翻牌后选择过牌-跟注(由于我们击中了顶对、后门同花、后门顺子, 翻牌有顺子听牌的可能性. 为了阻止对手可能存在的听牌, 我们进行下注); 转牌对手选择率先下注, 而我们则形成顶对以及顺子听牌, 因此选择跟注; 河牌对手再次下注.

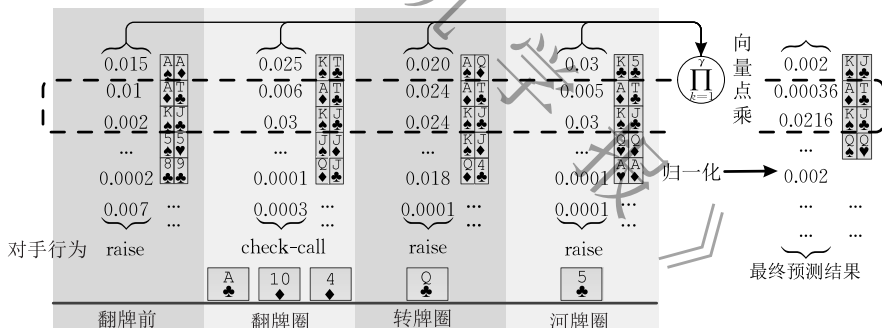


图 3 多个轮次的手牌概率分布预测样例

此时, 我们面对对手的河牌下注, 已经获得了对手四个轮次的行动信息. 下图描述了我们如何根据本文的方法, 使用四个轮次的信息, 不断地更新手牌分布向量并最终得到对手手牌的概率分布. 对手的手牌有 h_1 种可能性.

$$h_1 = C(50 - \epsilon, 2) |_{\epsilon=5} = 990 \quad (19)$$

由于发牌随机, 初始化时 $p_{\text{distribution}}$ 满足平均概率分布. 我们对每一种可能的牌型进行分析, 这里以对手手持以下牌型为例进行说明.

$$\begin{aligned} \psi_1 &= \{A\spadesuit, T\clubsuit\}, \\ \psi_2 &= \{K\spadesuit, J\clubsuit\} \end{aligned} \quad (20)$$

(1) 在翻牌前, 根据式(14)计算两手牌的概率, 假定为 0.01 和 0.002(因为翻牌前对手加注, 而手

持 ATo 比 KJo^② 更倾向于加注).

(2) 翻牌圈, ψ_1 形成两对. 假定由对手模型可知对手倾向于加注来保护成牌, 而不是过牌跟注. 而 ψ_2 形成坚果顺子听牌, 在这里做出过牌跟注非常合理. 因此在这一轮里, 根据式(14)算得两手牌的概率分别设为 0.006 和 0.03.

(3) 转牌圈, ψ_1 牌力没有提高, 但仍然属于强牌, 而 ψ_2 则形成了坚果顺子的超强牌. 因此对于本轮的加注, 两者概率接近, 假设分别为 0.024 和 0.024.

(4) 河牌圈, ψ_1 的牌力仍未能提高. 相反地, 公共

① 也可以略去式(14)的步骤直接在此一并进行归一化. 为求可读性, 本文并未略去.

② ATo(AT off suited)表示手牌一张为 A, 另一张为 10, 且不同色. 如同色, 则表示为 ATs(AT suited).

牌同时存在同花可能性和顺子可能性,对 ψ_1 而言具有极大威胁,但仍有很好的比牌价值.此时对手倾向于过牌以求最小代价比牌.对 ψ_2 而言虽然形成的顺子受到同花可能性的威胁,但是仍然是强牌.因此在河牌圈, ψ_2 比 ψ_1 更有理由加注.两者概率设为 0.005 和 0.03.

(5) 向量相乘和归一化.如式(16)所示,对每一轮的概率分布向量进行点乘.所得向量内的元素我们称之为概率的相对值.以 ψ_1 为例,对应的概率相对值为

$$P_{\psi_1} = \prod_{k=1}^4 P_{\psi_1}^k = 0.01 \times 0.006 \times 0.024 \times 0.005 = 7.2 \times 10^{-9} \quad (21)$$

同理, ψ_2 的概率相对值为 4.32×10^{-7} .

$$P_{\psi_2} = 4.32 \times 10^{-7} \quad (22)$$

在依次计算完所有手牌的相对概率值之后,我们采用如式(17)所示的方法,对所得的向量进行归一化操作,即可得到最终的牌型预测值.我们假设:

$$\sum_{k=1}^{C(50-\epsilon, 2)} P_{\psi_k} = 2 \times 10^{-5} \quad (23)$$

则两手手牌的最终预测概率分别如式(24)所示.由此可知, $\{K\spadesuit, J\clubsuit\}$ 的可能性远大于 $\{A\spadesuit, J\heartsuit\}$.

$$\begin{cases} P_{\psi_2} = 7.2 \times 10^{-9} / 2 \times 10^{-5} = 0.036\% \\ P_{\psi_2} = 4.32 \times 10^{-7} / 2 \times 10^{-5} = 2.16\% \end{cases} \quad (24)$$

4.4 胜率计算

本节介绍如何在现有的已知条件下,估算自己的胜率.我们将所得的胜率作为决策制定的重要参考.一般地,有两种方法计算胜率:穷举法和蒙特卡洛法.多人德州扑克中,多数情况下无法在可接受的时间内穷尽所有可能性.例如在一个四人底池的翻牌圈,后续的两张公共牌和另外三个对手的手牌未知,总共的可能性有

$$C_{50}^2 \times C_{48}^2 \times C_{46}^2 \times C_{44}^2 \approx 1.35 \times 10^{12} \quad (25)$$

随着人数的增加,这个数字也将迅速增加,并远超计算机的承受范围.因此,常使用蒙特卡洛方法来计算胜率.即随机地生成未知信息,并统计胜负关系.传统的蒙特卡洛方法认为未知信息的概率分布满足平均分布.对于“后续公共牌”而言,这是成立的.但对于对手的手牌而言,根据上一节的介绍,已经有了预测的概率分布 $p_{\text{distribution}}$.我们据此来随机生成对手的手牌.

具体方法如算法 1 所示.假设对手的手牌组合 all_hands 有 $C(52, 2) = 1326$ 种,若手牌与己方手牌(my_hands)或公共牌(Δ)冲突,则概率置为 0.上

节我们可以得到每位对手的牌型分布 $p_{\text{distribution}}$, 一共有 q 位对手,用二维数组 distribution 表示.

算法 1. 基于蒙特卡洛方法的胜率计算.

```

输入:  $\psi[1326]\text{all\_hands}$  //所有可能的手牌组合
INT  $q$  //对手数量
 $\psi[q][1326]\text{distribution}$  //对手手牌分布
 $\psi\text{my\_hands}$  //己方手牌
输出: DOUBLE  $\text{winPro}$  //己方获胜概率

 $\psi[q]\text{selected\_hands}$  //每位对手的手牌,暂时为空
INT  $\text{winCnt} = 0$  //记录胜率次数
WHILE ( $N > 0$ ) //N 为蒙特卡洛撒点次数
    FOR (each opponent  $opp$ )
        DOUBLE  $r = \text{random}(0, 1)$ 
        DOUBLE  $\text{sum} = 0$ 
        FOR (INT  $i = 0; i < 1326; i++$ )
            IF ( $r \in [\text{sum}, \text{sum} + \text{distribution}[opp][i]]$ )
                 $\text{selected\_hands}[opp] = \text{all\_hands}[i]$ 
            BREAK
         $\text{sum} += \text{distribution}[opp][i]$ 
    END FOR
END FOR
Card[ $\epsilon$ ] $\Delta =$ 
    deal cards without  $\text{my\_hands}$  and  $\text{selected\_hands}$ 
    BOOLEAN  $\text{isWin} =$ 
        judge( $\text{selected\_hands}, \Delta, \text{my\_hands}$ )
    IF ( $\text{isWin} == \text{true}$ )
         $\text{winCnt}++$ 
END WHILE
RETURN  $\text{winCnt} / N$ 

```

接下来我们进行 N 次循环,首先,程序为每位对手(opp)选择手牌(模拟发牌过程).所选手牌的概率分布满足 $\text{distribution}[opp]$,即选手 opp 的手牌分布,由上一节得出.在除去选出的手牌之后,我们随机发出剩余的公共牌 Δ .最后比较胜负关系^①.理论上说,迭代次数 N 越大,所得的胜率越接近收敛值.同时,运行时间也越长.为分析迭代次数 N 的最佳权衡值,我们选取了一个示例,以 N 为自变量,胜率值和消耗时间为因变量,绘制了图 4.由图可知,当迭代次数 N 达到一定值时(例如 500),胜率值变开始明显收敛.意味着再增加迭代次数,所得的胜率将不会有明显改变.

例如在该图中,我们取 $N \in [900, 1000]$ 范围内的胜率值的平均数 62.3% 作为收敛值的近似.当

① 牌力大小比较的代码参见:牌力大小比较的代码参见:
<http://www.cs.helsinki.fi/group/asdf/sources/Plugins/src/com/asdf/plugins/pokergames/>

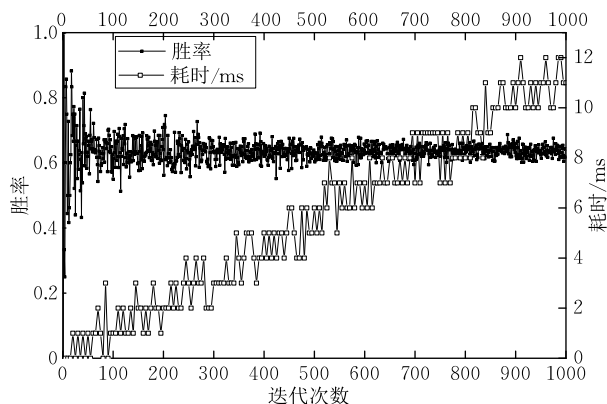


图 4 胜率计算时迭代次数对准确度、耗时的影响

$N \in [400, 500]$ 时, 胜率值为 $62.5\% \pm 3.3\%$. 意味着当 N 增加到 400 至 500 时, 胜率值计算的误差仅为 5.62%, 具体如式 (26) 所示. 同时, 随着 N 的增加, 运行时间也接近线性地增加. 作为耗时和准确率相互权衡的结果, 本文 N 取 500.

$$\max \left\{ \frac{(62.3\% - (62.5\% - 3.3\%)) / 62.3\%}{(62.5\% + 3.3\% - 62.3\%) / 62.3\%} \right\} = 5.62\% \quad (26)$$

由算法 1 及上述分析可知, 胜率的计算需要进行 N 次重复的发牌和比牌. 发牌过程中, 需要为 q 位对手按照所预测的概率分布选择手牌. 对手的手牌可能性至多有 $h = 1326$ 种, 因此发牌的时间复杂度为 $O(qh)$. 接下来算法将调用 `judge()` 函数进行比牌. 比牌的过程需要将己方手牌与 q 位对手逐一比较, 一旦发现对手的牌强于己方, 则停止比牌; 如果己方的牌强于所有对手, 则为获胜. 该过程的时间复杂度为 $O(q)$. 因此每次迭代的运行时间至多为 $qh + q$, 时间复杂度可表示为 $O(qh)$. 因此算法 1 的时间复杂度为 $O(Nqh)$.

相对于穷尽所有可能的胜率计算方法, 采用蒙特卡洛方法具有如下优势.

(1) 低时间复杂度, 由式 (25) 可知, 穷尽法在多人底池中几乎无法在合理时间内完成计算. 而由上述分析可知, 蒙特卡洛方法的时间复杂为 $O(Nq)$, 而由于德州扑克本身仅使用一副扑克牌进行博弈, 故 $q \leq 23$ (实际情况中, q 不超过 10). 由图 4 可知, 算法耗时仅在 10 ms 级.

(2) 高扩展性, 从算法 1 的分析可知, 蒙特卡洛的胜率计算采用“撒点-计算-统计”的模式, 当底池人数、博弈阶段、对手手牌发生变化时, 可以通过调整撒点方法高效地应对.

(3) 高可控性, 从时间复杂度和算法收敛的分析上可以看出, 我们可以通过调节 N 的大小, 灵活

地在耗时和准确度上进行调整. 也就是说, 我们可以通过控制撒点的数量, 对算法的性能和表现进行权衡和控制.

4.5 决策制定

算法 2 是我们决策制定的一般流程. 在算法内部, 我们设定了三个起始的阈值作为行动的参考. 我们使用了文献 [22] 的下注模型, 将下注量分为半个底池和一个底池两种. 首先, 我们调用 `AdjustThrAccordingToScenario()` 根据当前局势调整阈值. 例如当前是两人底池时, 本方的胜率为 70% 才可以跟注对手的加注, 但是, 三人底池时, 仅需要 55% 的胜率即可跟注对手的加注. 通常人数越多, 我方做出同样行为所需的胜率越低. 接下来, 如果遇到特殊情况时, 例如对手进行超高额下注, 算法会调用相应的函数 `getSpecialAction()` 返回特殊处理的结果. 最后, 我们会将 4.1~4.4 节所得的胜率 $winPro$ 与阈值进行比对, 返回相应的下注行为.

算法 2. 决策制定基本算法.

```

输入: DOUBLE call_thr      //跟注阈值
DOUBLE raise_half_thr     //加注半个底池阈值
DOUBLE raise_one_thr      //加注一个底池阈值
DOUBLE winPro             //预计胜率
输出: action              //决策

AdjustThrAccordingToScenario();
IF (isSpecial())
    RETURN getSpecialAction();
IF (winPro < call_thr)
    RETURN fold or check //如无法 check, 则 fold
ELSE IF (winPro < raise_half_thr)
    RETURN check or call //如无法 check, 则 call
ELSE IF (winPro < raise_one_thr)
    RETURN raise_half_pot //下注半个底池
ELSE
    RETURN raise_one_pot //下注一个底池

```

由于所得的胜率 $winPro$ 已经内在考虑了引言中的因素 (4)~(6); 对手的打牌风格、对手在本局中的下注行为、手牌和公共牌. 我们对因素 (1)~(3) 进行了如下优化.

(1) 筹码深浅. 具体地, 我们将筹码深度分为浅筹码 (低于 10 个大盲注)、中等筹码 (10~80 个大盲注) 和深筹码 (高于 80 个大盲注). 优化包括: 浅筹码在翻牌前对于任何手牌采取的行动只有全下和弃牌两种^[6]; 深筹码时在翻牌前的投机牌通常跟注而不会弃牌.

(2) 博弈规则. 在锦标赛中, 保证不被淘汰至关

重要. 我们在翻牌前提高阈值 `raise_call_thr` 从而提高入局手牌的平均牌力, 避免高风险. 同时我们会提高阈值 `raise_one_thr`, 从而减少大额的下注. 在现金局中, 策略则相反.

(3) 牌型. 牌型不同, 虽然具有同样胜率, 打法也不同. 本节就典型的听牌情况进行讨论. 我们比较两种情况:

① 形成两头听顺的牌型. 如果听中, 则确保获胜; 但当前牌力极弱, 如没听中则牌力落后. 在如此清晰的牌力情况下, 对手的牌型分布已不再重要. 胜率可以由补牌数进行估算. 两头听顺有 8 张补牌, 胜率为

$$p_{8\text{outs}} = 1 - \frac{42}{50} \times \frac{41}{49} = 29.7\%$$

(27)

② 形成中对, 且没有同花和顺子听牌. 即牌力可能暂时领先, 但后续提升空间小. 在当前情况下, 同样按照 4.1~4.4 节的胜率计算方法, 得到与情况①非常接近的胜率, 我们假设为 30%.

在上述两种情形中, 虽然胜率接近, 但是策略却不一样. 前者倾向于过牌跟注或者率先下注领打, 而很少弃牌. 因为即时当前赔率不合适, 但很多情况下有较大的隐藏赔率. 后者则常常会尝试下注赢取底池, 而在遭遇抵抗时选择立刻弃牌.

5 实验结果

5.1 实验参数

我们选择 2015 年华为精英软件挑战赛的七个参赛程序作为博弈对手. 这些程序分别由不同作者完成, 其中两个程序(XDDS、FANDY)进入复赛(256/3865), 四个程序(HC、LUO、HARRY1 和 HARRY2)进入全国总决赛(32/3865). 而且每个 AI 程序的松紧、凶弱风格各异. 图 5 为 HC、HARRY1 和本文程序(FOLD)的技术数据统计. 可以看出, 三个程序无论是加注、跟注、过牌还是弃牌的比例都相差很大. 本文分别采用如表 2 所示的赛制, 并分别进行两人

表 2 实验所用赛制

参赛人数	2~8 人
比赛局数	600 局(每场)
小/大盲注	20/40
是否升盲	否
优劣指标	排名、持续局数、筹码剩余、筹码波动
起始买入	2000(50 个大盲注)
后续买入	2000(仅限 1 次)
描述	多个程序进行若干场比赛, 每场最多 600 局. 每场比赛, 初始筹码和买入筹码都消耗完即为淘汰. 程序依次按坚持局数、筹码剩余进行排名.

赛和八人赛. 实验的物理机配置为 AMD Athlon II X4 535 CPU、4GB 内存. 使用 Oracle VirtualBox 作为虚拟化平台, 使用 Linux Ubuntu 12.04 虚拟机. 该虚拟机配置 1GB 内存和 2 个 VCPU.

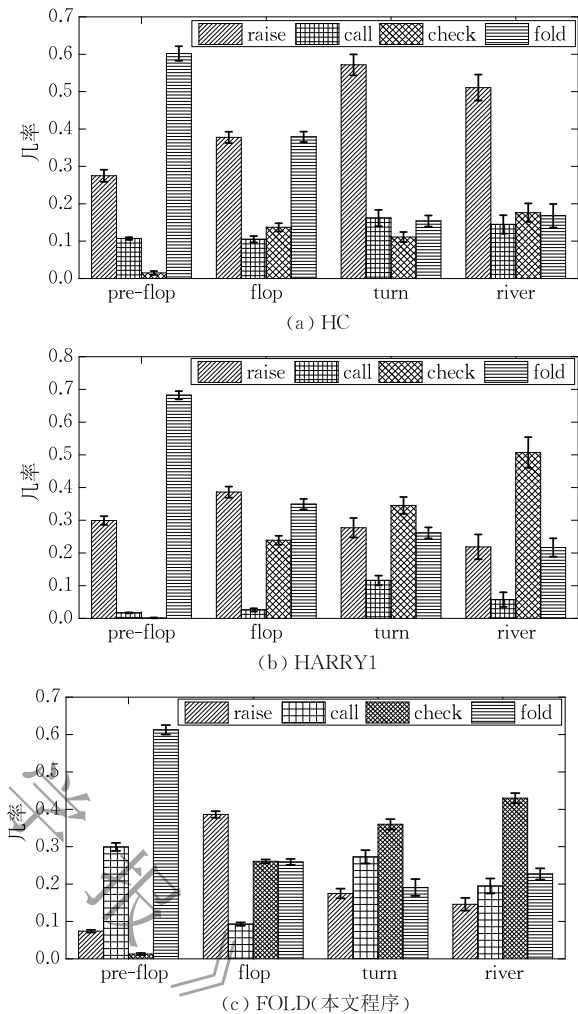


图 5 HC、HARRY1、FOLD(本文程序)的动作统计

5.2 牌型预测

文献[25]是使用马尔可夫模型进行对手建模的典型代表, 我们将其模型记为 Markov. 该文章选用 ACPC 的两人有限注德州扑克的博弈数据对其对手模型进行训练, 并得出对手牌型的预测精度. 为得到可方便比较的数据, 我们也选用了同样的数据集^①: 2012 年的两人有限注德州扑克中 Slumbot 的博弈数据. 该数据格式如图 6 所示, 包含了所有伦次的玩家行动、手牌、公共牌和筹码胜负数量.

由于本文程序适用于多人无限注德州扑克, 与该数据的博弈场景有所差异. 因此我们对模型做了如下调整: (1) 有限注德州扑克中加注量是固定的,

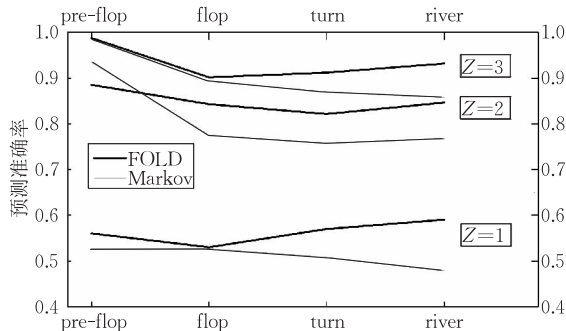
① 数据可由如下链接下载: <http://www.computerpokercompetition.org/downloads/competitions/2012>

为,因此轮次越后我方获得的信息越多.所以本模型可以在后续轮次中也能较准确地计算对手的牌力.实际上,后续轮次的牌型预测在总体的筹码胜负上更为关键.因为一局比赛中,轮次越往后底池越大,筹码胜负也越多.

我们先进行两人无限注德州扑克的比赛,再探讨多人的情况.为消除随机因素,需进行多次重复实验^[26].理论上,重复次数越多,所得结果越准确.在研究中我们发现,随着比赛次数的增加,所得结果会很快地收敛.图 8 显示随着比赛进行次数的不断增加,选手平均排名与进行 500 场比赛的平均排名的重合度.由图可知,在进行了 70 局左右比赛的时候,重合度便达到 95%,且标准误差(Standard Error, SE)低于 2.5%.因此,比赛仅需重复 70 场以上,即可得到具有统计意义的结果.在本文的实验中,我们将进行 100 次的重复实验,将实验的平均值和标准误差同时进行展示.



图 7 显示了 Z 分别为 1、2、3 时, Markov 和本实验的对手模型(计为 FOLD)的预测精度. 本文的对手模型在翻牌后、转牌、河牌的预测精度均高于 Markov. 在这个三个轮次中, 分别比 Markov 高出 3.4%、8.5% 和 14.1%. 一般地, 预测精度会在翻牌后有所下降. 这是因为翻牌前所能形成的牌型仅限于高张和一对. 牌型较少, 预测准确度高. 文献[25]将高张分为大高张(如 A9)和小高张(如 84); 将对子分为大对子(如 KK)和小对子(55). 因此翻牌前可组成总共四种牌型. 而翻牌后则有可能出现 11 种牌型中的任意一种, 使预测更困难. 但由于本文的模型综合考虑了对手在该牌局中之前所有轮次的行



这里的人数是指比赛开始时的人数. 因为在比赛过程中, 会陆续有选手输掉所有筹码而被淘汰. 例如在五人局中, 我们会随机选择 4 位其他对手并进行 100 场比赛, 得到一个平均值 (如平均排名、平均筹码剩余).

图 9(a)显示了博弈人数对筹码剩余量的影响. 由于人数越多,总筹码数也线性增加. 在两人局中,

表 3 两人局比赛成绩

	XDDS	XJX	FANDY	HC	HARRY1	LUO	HARRY2	FOLD	TOTAL
XDDS	—	0.90	0	0.95	0.90	0.55	1.00	0.05	0.62
	—	6607±210	−7968±1.46	7162±174	6374±240	207±257	7977±1.40	−6029±211	2047±612
XJX	0.10	—	0	0	0.40	0	0.55	0	0.15
	−6607±210	—	−7972±0.92	−7956±1.33	−1170±334	−7815±25.0	1527±282	−7859±11.3	−5412±437
FANDY	1.00	1.00	—	0	0.20	0	0.20	0.05	0.35
	7968±1.46	7972±0.92	—	−7855±21.4	−5543±246	−7859±20.9	−3865±229	−2351±194	−1648±592
HC	0.05	1.00	1.00	—	0.75	0.65	1.00	0.35	0.69
	−7162±174	7956±1.33	7855±21.4	—	3686±340	1752±350	7722±32.0	1510±212	3331±525
HARRY1	0.10	0.60	0.80	0.25	—	0.80	0.80	0.1	0.49
	−6374±240	1170±334	5543±246	−3686±340	—	3868±205	2522±135	−5328±195	−326.3±535
LUO	0.45	1.00	1.00	0.35	0.20	—	0.05	0.15	0.48
	−207±257	7815±25.0	7859±20.9	−1752±350	−3868±205	—	−4334±149	−1008±150	643.5±523
HARRY2	0	0.45	0.80	0	0.20	0.95	—	0.05	0.35
	−7977±1.40	−1527±282	3865±229	−7722±32.0	−2522±135	4334±149	—	−3628±120	−2168±475
FOLD	0.95	1.00	0.95	0.65	0.90	0.85	0.95	—	0.89
	6029±211	7859±11.3	2351±194	−1510±212	5328±195	1008±150	3628±120	—	3533±381

总筹码数为 $4000 \times 2 = 8000$ ，而在 3 人局中，总筹码数为 12000，以此类推. 为了更清晰显示博弈水平，我们在图 9(b)中展示了平均排名情况. 平均排名的最小值为 1(即每场比赛都是第 1 名)，最大值为 q (博弈人数). 本文程序的排名值始终保持在 1.11~1.74 以内，体现了非常好的稳定性. 接下来，本节将八人局比赛作为德州扑克多人局比赛的代表进行实验评估. 最后，文章对德州扑克两人局比赛的博弈成绩进行了展示.

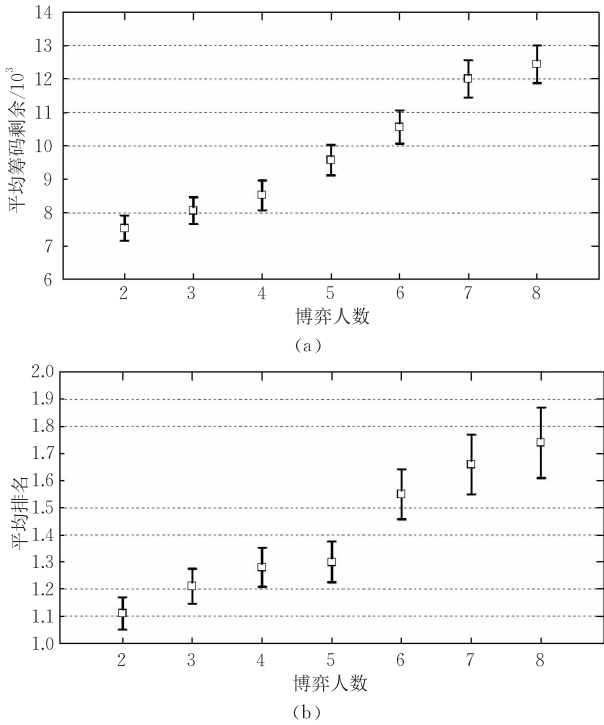


图 9 博弈人数对博弈效果的影响

图 10 为 8 个程序的筹码走势，其中深色线条为本文程序. 图10(a)为任取的一次实验的筹码走势

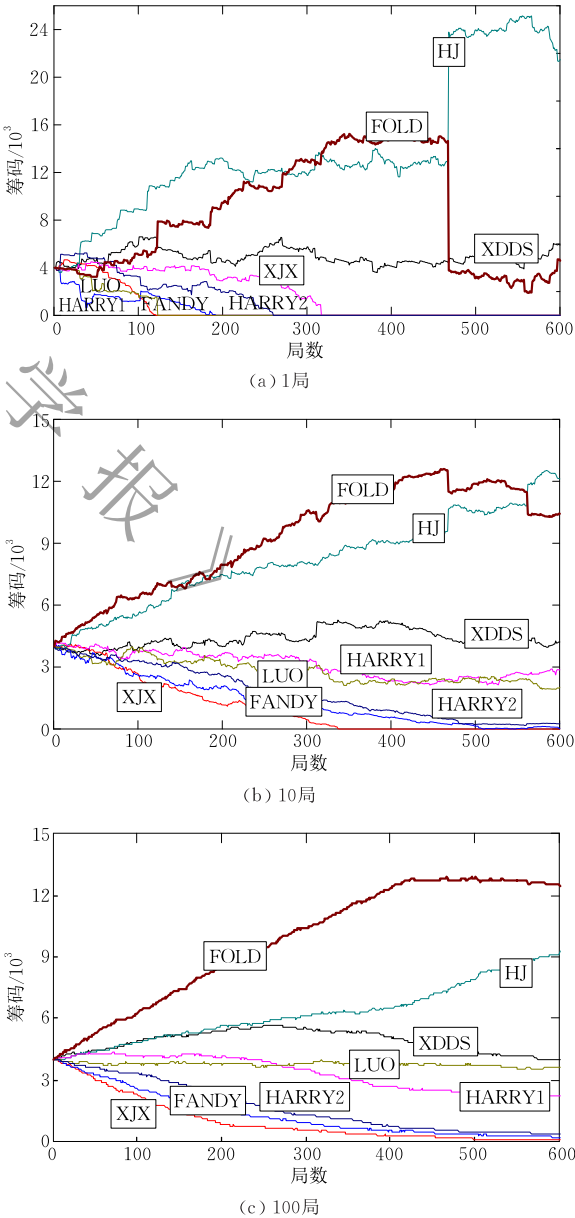


图 10 1 局、10 局、100 局的平均筹码走势图

图. 该图直观的反映了博弈过程中, 各选手的筹码数量. 筹码的巨幅波动说明在该局中, 博弈双方或多方进行了大幅下注和跟注. 为反映统计性结果, 图 10 (b)、(c) 分别为 10 次、100 次实验的平均筹码走势. 可见, 在大量重复实验的情况下, 本程序的表现趋于稳定.

5.5 分析和探讨

为更全面地评估程序性能, 我们综合测试了程序的名次、坚持局数、剩余筹码量和筹码波动. 本节以八人局为例进行评测. 如图 11 所示为 100 场比赛

的各项参数的平均值. 其中, 筹码波动指的是筹码在一场比赛内的波动幅度. 选手在考虑决策期望值的同时, 总是希望降低筹码波动. 我们提出以下方法计算波动.

$$F = \sum_{k=2}^G (C_{\text{left}}^k - C_{\text{left}}^{k-1})^2 \quad (28)$$

G 为程序坚持的局数 (上限为 600), C_{left}^k 和 C_{left}^{k-1} 为第 k 局和第 $k-1$ 局比赛后的筹码剩余量. 由图 11 可知, 本程序在平均排名、坚持局数、剩余筹码上均为最好成绩, 在筹码波动上略大于其它程序.

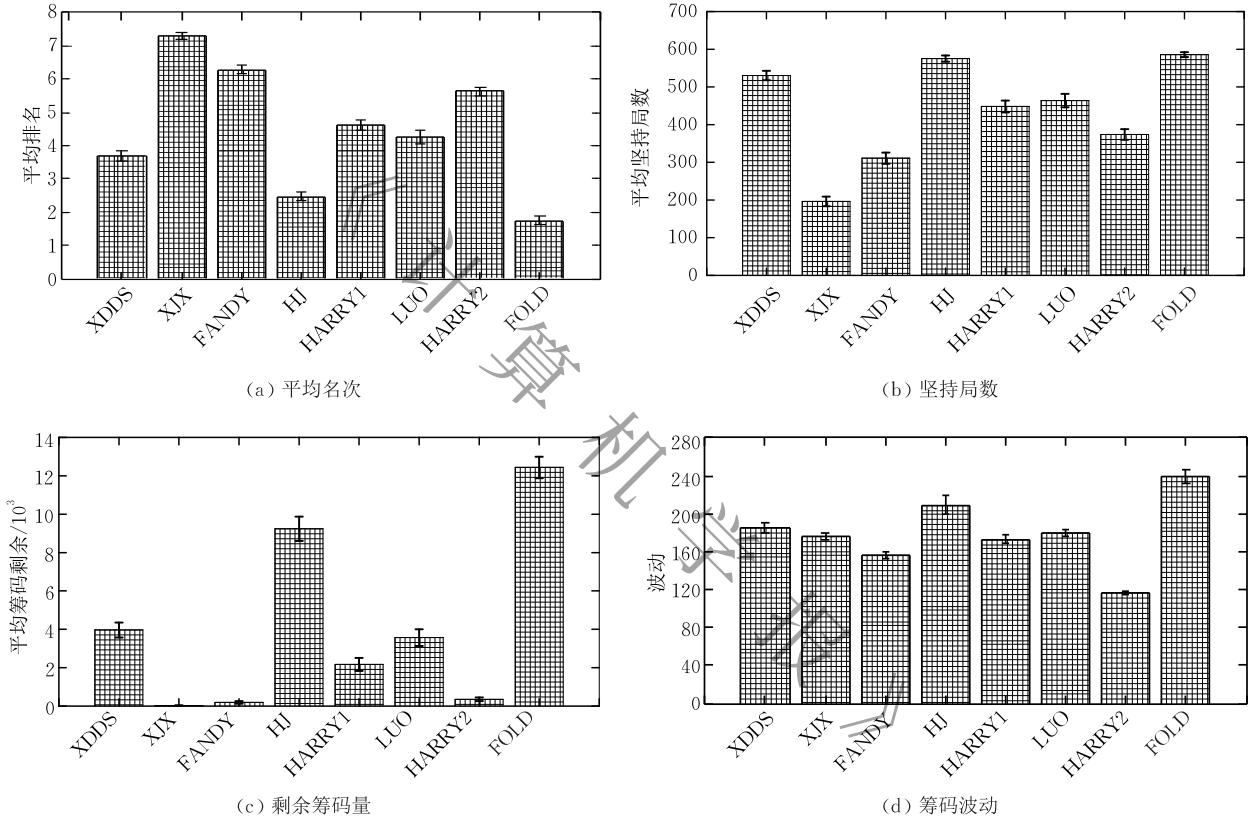


图 11 平均名次、坚持局数、剩余筹码量和筹码波动

本文的博弈方法依赖胜率计算, 但是如 4.5 节所述, 胜率一样, 本方的手牌和公共牌的结构可能完全不一样. 其中最重要的差别是听牌的处理, 不能仅考虑表明赔率, 而应该额外考虑隐含赔率. 因此在表面赔率不合适但有足够隐含赔率时, 应该进行跟注. 事实上任何手牌在公共牌没有发完之前, 都有牌力提升的可能, 即存在补牌. 在本程序中, 我们仅考虑补牌数量不小于 8 的情况. 这些情况往往代表同花和顺子的听牌. 由于德州扑克牌型极为复杂, 直接针对各种牌型 (例如带对听同花、三条听葫芦) 逐一穷举, 不仅带来大量的逻辑判断, 而且极易出错. 本文采用如下计算方法:

$\forall Card_i \in \text{undealt cards},$

if $P_{\langle \psi, \Delta_1 \rangle} - P_{\langle \psi, \Delta_0 \rangle} \geq 0.6(1 - P_{\langle \psi, \Delta_0 \rangle})$ (29)

then outs ++

即对于任何可能的卡牌 $Card_i$, 假定对手手牌随机分布, 如果在发出 $Card_i$ 前的胜率 $P_{\langle \psi, \Delta_0 \rangle}$ 和发出 $Card_i$ 后的胜率 $P_{\langle \psi, \Delta_1 \rangle}$ 满足式 (29), 即认为 $Card_i$ 是一张补牌. 其中 ψ 为己方手牌, Δ_0 为现有公共牌, Δ_1 为发出 $Card_i$ 之后的公共牌. 该式的含义表示: 发出的 $Card_i$ 使得己方胜率得到了显著提升.

图 12 以“已考虑隐含赔率”的博弈成绩作为基准值, 展示了程序是否对听牌进行特殊处理 (即是否考虑隐含赔率) 的对比结果. 可见, 对听牌进行特殊

处理明显提升了程序的水平. 例如考虑隐含赔率之后, 在平均排名上性能提升了 0.8%.

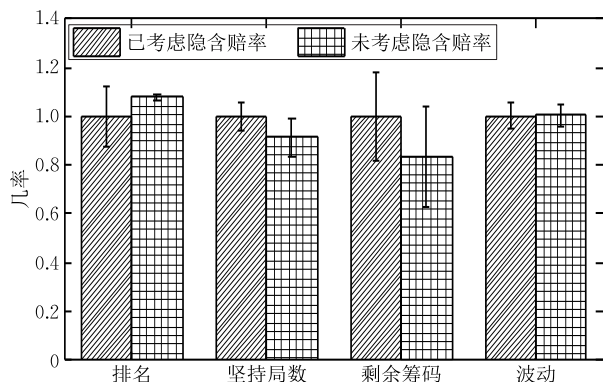


图 12 听牌时是否考虑隐含赔率的成绩对比

6 结束语

本文从非完备信息博弈的角度出发, 提出了一套完整的博弈框架, 并讨论了框架的适用性. 我们将该框架具体到德州扑克中, 提出了默认的对手模型, 有效地解决对手历史数据不足的问题. 该模型可以在博弈中根据对手的具体风格动态地进行自适应调整和优化. 随后, 文章详细阐述了“牌型预测-胜率计算-决策制定”的完整德州扑克博弈方法, 并辅以具体的样例加以说明. 在最后的决策制定部分, 我们讨论了影响决策的多项关键因素, 包括对手的筹码深浅、博弈规则以及牌型结构. 在实验部分, 我们对比了基于马尔可夫模型的手牌预测方法. 结果显示本文的模型可以有效地提升翻牌后的预测精度. 接下来我们采用 2015 年华为软件精英挑战赛所用的官方服务器, 并选用若干位进入复赛、全国总决赛的程序做对比实验, 采用多次重复的平行测试, 以期得到具有统计意义的结果. 数据显示, 基于本文方法实现的博弈程序几乎各项技术指标上都取得最佳成绩.

致 谢 在本论文的实验部分, 北京大学韩朝相、南京航空航天大学范壁健、厦门大学罗浩阳、河海大学程海粟、哈尔滨工业大学徐基鑫、西安电子科技大学王东东提供了测试程序; 华为公司提供了虚拟机、运行环境和德州扑克博弈服务器. 在此表示感谢!

参 考 文 献

- [1] Korf R E, Chickering D M. Best-first minimax search. *Artificial Intelligence*, 1996, 84(95): 299-337
- [2] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016, 529(7587): 484-489
- [3] Van der Kleij A A J. Monte Carlo Tree Search and Opponent Modeling Through Player Clustering in No-Limit Texas Hold'em Poker [M. S. dissertation]. University of Groningen, Groningen, The Netherlands, 2010
- [4] Teófilo L F, Passos N, Reis L P, Cardoso H L. Adapting strategies to opponent models in incomplete information games: A reinforcement learning approach for poker// *Proceedings of the Autonomous and Intelligent Systems*. Heidelberg, Berlin, 2012: 220-228
- [5] Nash J F. Equilibrium points in n -person games. *Proceedings of the National Academy of Sciences of the United States of America*, 1950, 36(1): 48-49
- [6] Harrington D. Harrington on Cash Games: How to Win at No-Limit Hold'em Money Games. Washington, USA: Two Plus Two Publishing LLC, 2008
- [7] Gilpin A, Sandholm T, Sørensen T B. Potential-aware automated abstraction of sequential games, and holistic equilibrium analysis of Texas Hold'em poker// *Proceedings of the AAAI Conference on Artificial Intelligence*. Cambridge, London, 2007: 50-57
- [8] Bowling M, Burch N, Johanson M, et al. Heads-up limit Hold'em poker is solved. *Science*, 2015, 347(6218): 145-149
- [9] Billings D, Burch N, Davidson A, et al. Approximating game-theoretic optimal strategies for full-scale poker// *Proceedings of the International Joint Conference on Artificial Intelligence*. Acapulco, Mexico, 2003: 661-668
- [10] Brown N, Ganzfried S, Sandholm T. Tartanian7: A champion two-player No-Limit Texas Hold'em poker-playing program// *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Austin, USA, 2015: 1-2
- [11] Heinrich J, Silver D. Smooth UCT search in computer poker// *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. Buenos Aires, Argentina, 2015: 1-7
- [12] Brown N, Sandholm T W. Simultaneous abstraction and equilibrium finding in games// *Proceedings of the AAAI Conference on Artificial Intelligence*. Austin, USA, 2015: 489-496
- [13] Waugh K, Morrill D, Bagnell J A, et al. Solving games with functional regret estimation// *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Austin, USA, 2015: 103-109
- [14] Ferreira J C. Opponent Modelling in Texas Hold'em: Learning Pre-Flop Strategies in Multiplayer Tables [M. S. dissertation]. University of Porto, Porto, Portugal, 2012
- [15] Rubin J, Watson I. Successful performance via decision generalisation in no limit Texas Hold'em// *Proceedings of the 19th International Conference on Case-Based Reasoning Research and Development*. London, UK, 2011: 467-481

[16] Southey F, Bowling M P, Larson B, et al. Bayes’ bluff: Opponent modelling in poker//Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence. Edinburgh, Scotland, 2005: 550-558

[17] Risk N A, Szafron D. Using counterfactual regret minimization to create competitive multiplayer poker agents//Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. Toronto, Canada, 2010: 159-166

[18] Sandholm T. Abstraction for solving large incomplete-information games//Proceedings of the AAAI Conference on Artificial Intelligence. Austin, USA, 2015: 1-5

[19] Gilpin A, Sandholm T. Better automated abstraction techniques for imperfect information games, with application to Texas Hold’em poker//Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. Hawaii, USA, 2007: 192-200

[20] Ganzfried S, Sandholm T. Tartanian5: A heads-up no-limit Texas Hold’em poker-playing program//Proceedings of the Computer Poker Symposium at the National Conference on Artificial Intelligence. Sydney, Australia, 2012: 1-6

[21] Miltersen P B, Sørensen T B. A near-optimal strategy for a heads-up no-limit Texas Hold’em poker tournament//Proceedings of the 6th International Joint Conference on Autonomous Agents and MultiAgent Systems. Hawaii, USA, 2007: 191-198

[22] Gilpin A, Sandholm T, Sørensen T B. A heads-up no-limit Texas Hold’em poker player: Discretized betting models and automatically generated equilibrium-finding programs//Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems. Estoril, Portugal, 2008: 911-918

[23] Billings D, Davidson A, Schaeffer J, Szafron D. The challenge of poker. Artificial Intelligence, 2002, 134(1-2): 201-240

[24] Nicolai G, Hilderman R J. No-Limit Texas Hold’em poker agents created with evolutionary neural networks//Proceedings of the IEEE Symposium on Computational Intelligence and Games. Nashville, USA, 2009: 125-131

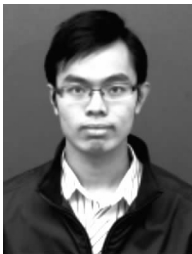
[25] Li Ruo-Bing. Study on Poker Opponent Modeling Based on Bayesian Network and Hidden Markov Models [M. S. dissertation]. Nanjing University, Nanjing, 2013(in Chinese) (李若冰. 基于贝叶斯网络和隐马尔可夫模型的扑克对手建模研究[硕士学位论文]. 南京大学, 南京, 2013)

[26] Billings D. Algorithms and Assessment in Computer Poker [Ph. D. dissertation]. University of Alberta, Edmonton, Canada, 2006

附录 1.

表 4 本文涉及的部分术语及释意

中文	英文	解释
下注轮	betting round	一局比赛包括四个下注轮: 翻牌前 (pre-flop)、翻牌圈 (flop)、转牌圈 (turn)、和河牌圈 (river)。
松/紧	tight/loose	相对概念, 松的选手特点为: 起手牌范围大, 常用边缘牌或弱牌跟注。紧的选手相反。
凶/弱	aggressive/weak	相对概念, 凶的选手特点为: 打法激进, 频繁加注从而迫使对手弃牌。弱的选手相反。
范围	range	翻牌前特定行为对应的可能手牌的组合。经常只能猜测获得大致的牌力情况, 故称范围。
枪口/庄位	UTG/dealer	大盲的下家称为枪口位 (under the gun, UTG), 是翻牌前最差位置。小盲的上家称为庄位。
差距理论	gap theory	该理论指出翻牌前用来跟注一个公开加注的牌力, 应强于自己在该位置进行公开加注的牌力。
公开加注	open (raise)	翻牌前的第一个加注。
再加注/3-bet	re-raise/3-bet	面对对手的加注, 再次加注。翻牌前的 re-raise 通常称为 3-bet, 通常是强力起手牌的表现。
投机牌	speculative hands	起手牌的一类, 特指中等的或者小的同花连张, 中等的或者小的对子。
平衡	balance	在最佳策略和亚策略之间选择性地切换, 或在不同意图时进行同样的动作。目的都是让对手无法根据行为读出自己的手牌。
赔率	pot odds	底池筹码与跟注所需筹码的比值, 是判断跟注与否的重要参数。
隐含赔率	implied odds	可能赢取的筹码值与跟注所需筹码的比值。显然, 当前底池筹码 $\leq$ 可能赢取的筹码 $\leq$ 双方筹码的最大值。由于对手的后续行为不确定, 因此隐含赔率仅是一个估计值。
听牌	drawing hands	暂时没有形成强牌, 但如果后续发出某些公共牌能使牌力极大增强。
听死牌	drawing dead	即使后续成功击中要听的牌, 但牌力仍弱于对手。
补牌	outs	当公共牌未发完时, 后续能够使牌力得到大幅提升的公共牌。
(半)诈唬	(semi-)bluff	诈唬是指没有形成强牌却扮演强牌进行下注, 以期对手弃牌。半诈唬是暂时没有强牌, 但有强牌的听牌, 在此情况下做率先下注。
坚果	nuts	无论对方手牌为任何牌, 本方的牌都强于对手。
深筹码/浅筹码	deep stack/short stack	相对概念, 相比于大盲注, 筹码的多少称为筹码的深浅。例如可认为超过 80 个和低于 20 个大盲注分别叫深筹码和浅筹码。
价值注	value bet	在确定自己牌力强于对手的情况下, 希望对手进行跟注, 从而获得更多回报的下注。



LI Xiang, born in 1990, Ph.D. candidate. His research interests focus on cloud computing and artificial intelligence, etc.

JIANG Xiao-Hong, born in 1966, Ph.D. , associate professor. Her research interests include computer architecture, distributed system, cloud computing, etc.

CHEN Ying-Zhi, born in 1988, M.S. Her research interests focuses on data mining, especially for Chinese traditional medicine.

BAO You-Jun, born in 1991, M.S. candidate. His research interests include data mining and pattern recognition.

Background

Multiplayer no-limit Texas Hold'em is a typical example of incomplete information game, and has been an ongoing challenge in artificial intelligence for many years. The following facts make it complicated: (1) enormous possibilities of hands and undealt community cards, leading to massive situations and computational overheads; (2) large scale of solution space because the bet can be any size between your amount of chips and the minimum bet.

Generally, there are two angles to handle this problem: (1) game-theoretic methods, trying to search the game tree and find out the Nash Equilibrium points with a range of approaches. (2) knowledge-based methods, learning human player's actions and betting patterns to provide AI with references. Our proposed approach belongs to the latter one. As the development of gaming, players will take several

actions. The actions can reflect the information of their hands, which makes hands reading possible.

Our method predicts others' hands through observing their actions in each betting round, combining the action models of our opponents. We leverage a vector to represent the relative probabilities of each hand in a single round. Then we multiply the vectors and normalize the multiplication to obtain the final prediction of probability distribution. Then, Monte Carlo is adopted to get our winning rate, based on which we can calculate the expected value of each action, and make the final action considering specific factors. To our best knowledge, this is the first paper that holistically handles, designs and implements computer multiplayer (more than 3 players) No-limit Texas Hold'em AI based on hands prediction.