(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112639841 A
(43)申请公布日 2021.04.09

(21)申请号 201980034738.0
(22)申请日 2019.01.17
(85)PCT国际申请进入国家阶段日
2020.11.23
(86)PCT国际申请的申请数据
PCT/CN2019/072200 2019.01.17
(87)PCT国际申请的公布数据
WO2020/147074 EN 2020.07.23
(71)申请人 创新先进技术有限公司
地址 开曼群岛大开曼岛乔治镇医院路27号开曼企业中心
(72)发明人 李辉 胡开亮 宋乐
(74)专利代理机构 北京博思佳知识产权代理有限公司 11415
代理人 周嗣勇
(51)Int.Cl.
G06N 20/00(2006.01)

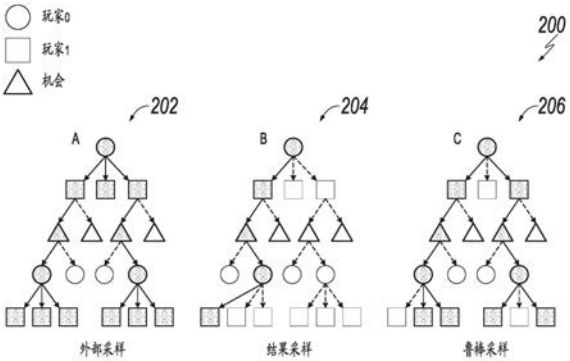
权利要求书3页 说明书29页 附图11页

(54)发明名称

用于在多方策略互动中进行策略搜索的采样方案

(57)摘要

本文公开的方法、系统和装置，包括在计算机存储介质上编码的计算机程序，用于执行反事实遗憾最小化(CRF)以在多方策略互动中进行策略搜索。所述方法之一包括：识别第一方在所述第一方的第一状态下的N1个可能动作；以第一采样概率从所述第一方在第一状态下的所述N1个可能动作中采样出可能动作；识别所述第一方在所述第一方的第二状态下的N2个可能动作，其中，所述第一方的第一状态比所述第一方的第二状态更接近IIG的开始状态；以第二采样概率从所述第一方在第二状态下的所述N2个可能动作中采样出可能动作，其中，所述第一采样概率小于所述第二采样概率。



CN 112639841 A

1.一种计算机实现的方法,用于执行反事实遗憾最小化CRF,以在两方或更多方之间的策略互动中进行策略搜索,所述方法包括:

识别第一方在所述第一方的第一状态下的N1个可能动作;

以第一采样概率从所述第一方在所述第一状态下的所述N1个可能动作中采样出可能动作;

识别所述第一方在所述第一方的第二状态下的N2个可能动作,其中,所述第一方的所述第一状态比所述第一方的所述第二状态更接近不完美信息博弈IIG的开始状态;

以第二采样概率从所述第一方在所述第二状态下的所述N2个可能动作中采样出可能动作,其中,所述第一采样概率小于所述第二采样概率;以及

基于所述第一方在所述第一状态下的所述N1个可能动作中的所述可能动作和所述第一方在所述第二状态下的所述N2个可能动作中的所述可能动作,执行所述CRF。

2.如权利要求1所述的方法,其中,基于所述第一方在所述第一状态下的所述N1个可能动作中的所述可能动作和所述第一方在第二状态下的所述N2个可能动作中的所述可能动作,执行所述CRF,包括:

计算所述第一方在所述第一状态下的所述N1个可能动作中的所述可能动作的遗憾值;

计算所述第一方在所述第二状态下的所述N2个可能动作中的所述可能动作的遗憾值;

基于所述N1个可能动作中的所述可能动作的所述遗憾值,更新所述第一方在所述第一状态下的第一策略;以及

基于所述N2个可能动作中的所述可能动作的所述遗憾值,更新所述第一方在所述第二状态下的第二策略。

3.如权利要求1所述的方法,还包括:

从所述第一方在所述第一状态下的所述N1个可能动作中采样出k1个可能动作,其中,所述k1个可能动作中的每个可能动作都以相同的所述第一采样概率被采样;以及

从所述第一方在所述第二状态下的所述N2个可能动作中采样出k2个可能动作,其中,所述k2个可能动作中的每个可能动作都以相同的所述第二采样概率被采样。

4.如权利要求3所述的方法,还包括:

基于所述第一方在所述第一状态下的所述N1个可能动作中的所述k1个可能动作和所述第一方在所述第二状态下的所述N2个可能动作中的所述k2个可能动作,执行所述CRF。

5.如权利要求3所述的方法,其中,所述第一采样概率是k1/N1,并且所述第二采样概率是k2/N2。

6.如权利要求3所述的方法,其中,$2 \leqslant k1 \leqslant N1$,并且$2 \leqslant k2 \leqslant N2$。

7.如权利要求3所述的方法,其中,$k1 = k2$。

8.如权利要求1所述的方法,还包括:

识别第二方在所述第二方的第一状态下的M1个可能动作;

以第三采样概率从所述第二方在所述第一状态下的所述M1个可能动作中采样出可能动作;

识别所述第二方在所述第二方的第二状态下的M2个可能动作,其中,所述第二方的所述第一状态比所述第二方的所述第二状态更接近所述IIG的所述开始状态;以及

以第四采样概率从所述第一方在所述第二状态下的M2个可能动作中采样出可能动作,

其中,所述第三采样概率小于所述第四采样概率。

9.如权利要求1所述的方法,还包括:

识别第二方在所述第二方的所述第一状态下的M1个可能动作,其中,所述第一方的所述第一状态比所述第二方的所述第一状态更接近所述IIG的所述开始状态;以及

以第三采样概率从所述第二方在所述第一状态下的所述M1个可能动作中采样出可能动作,其中,所述第三采样概率大于所述第一采样概率。

10.一种装置,用于执行反事实遗憾最小化CRF,以在两方或更多方之间的策略互动中进行策略搜索,所述装置包括:

第一识别模块,用于识别第一方在所述第一方的第一状态下的N1个可能动作;

第一采样模块,用于以第一采样概率从所述第一方在第一状态下的所述N1个可能动作中采样出可能动作;

第二识别模块,用于识别所述第一方在所述第一方的第二状态下的N2个可能动作,其中,所述第一方的第一状态比所述第一方的第二状态更接近不完美信息博弈IIG的开始状态;

第二采样模块,用于以第二采样概率从所述第一方在第二状态下的所述N2个可能动作中采样可能动作,其中,所述第一采样概率小于所述第二采样概率;以及

处理模块,用于基于所述第一方在第一状态下的所述N1个可能动作中的可能动作和所述第一方在第二状态下的所述N2个可能动作中的可能动作,执行所述CRF。

11.如权利要求10所述的装置,其中,所述处理模块包括:

第一计算模块,用于计算所述第一方在第一状态下的所述N1个可能动作中的可能动作的遗憾值;

第二计算模块,用于计算所述第一方在第二状态下的所述N2个可能动作中的可能动作的遗憾值;

第一更新模块,用于基于所述N1个可能动作中的可能动作的遗憾值,更新所述第一方在第一状态下的第一策略;以及

第二更新模块,用于基于所述N2个可能动作中的可能动作的所述遗憾值,更新所述第一方在第二状态下的第二策略。

12.如权利要求10所述的装置,其中:

所述第一采样模块从所述第一方在第一状态下的所述N1个可能动作中采样出k1个可能动作,其中,所述k1个可能动作中的每个可能动作都以相同的所述第一采样概率被采样;以及

所述第二采样模块从所述第一方在第二状态下的所述N2个可能动作中采样出k2个可能动作,其中,所述k2个可能动作中的每个可能动作以相同的所述第二采样概率被采样。

13.如权利要求12所述的装置,其中,所述处理模块基于所述第一方在第一状态下的所述N1个可能动作中的所述k1个可能动作和所述第一方在第二状态下的所述N2个可能动作中的所述k2个可能动作,执行所述CRF。

14.如权利要求12所述的装置,其中,所述第一采样概率是k1/N1,并且所述第二采样概率是k2/N2。

15.如权利要求12所述的装置,其中,$2 \leqslant k1 \leqslant N1$,并且$2 \leqslant k2 \leqslant N2$。

16.如权利要求12所述的装置,其中,k1＝k2。

17.如权利要求10所述的装置,还包括:

第三识别模块,用于识别第二方在所述第二方的第一状态下的M1个可能动作;

第三采样模块,用于以第三采样概率从所述第二方在第一状态下的所述M1个可能动作中采样出可能动作;

第四识别模块,用于识别所述第二方在所述第二方的第二状态下的M2个可能动作,其中,所述第二方的第一状态比所述第二方的第二状态更接近所述IIG的所述开始状态;以及

第四采样模块,用于以第四采样概率从所述第一方在第二状态下的M2个可能动作中采样出可能动作,其中,所述第三采样概率小于所述第四采样概率。

18.如权利要求10所述的装置,还包括:

第五识别模块,用于识别所述第二方在所述第二方的第一状态下的M1个可能动作,其中,所述第一方的第一状态比所述第二方的第一状态更接近所述IIG的所述开始状态;以及

第五采样模块,用于以第三采样概率从所述第二方在第一状态下的所述M1个可能动作中采样出可能动作,其中,所述第三采样概率大于所述第一采样概率。

19.一种系统,包括:

一个或多个处理器;以及

耦接到所述一个或多个处理器并其上存储有指令的一个或多个计算机可读存储器,所述指令能由所述一个或多个处理器执行以执行权利要求1至9中任一项所述的方法。

20.一种计算机实现的方法,用于执行反事实遗憾最小化CRF,以求解涉及两方或更多方的不完美信息博弈IIG,所述IIG表示涉及至少第一方和第二方的合作产品服务推荐服务,所述第一方具有对所述第二方的信息的有限访问权,所述方法包括:

识别第一方在所述第一方的第一状态下的N1个可能动作,其中,所述第一方的第一状态包括由所述第二方提供的信息的历史,所述第一方的所述N1个可能动作包括:响应于由所述第二方提供的所述信息的所述历史的N1个可能动作,以向客户提供产品服务推荐;

以第一采样概率从所述第一方在第一状态下的所述N1个可能动作中采样出可能动作;

识别所述第一方在所述第一方的第二状态下的N2个可能动作,其中,所述第一方的第一状态比所述第一方的第二状态更接近IIG的开始状态;

以第二采样概率从所述第一方在第二状态下的所述N2个可能动作中采样可能动作,其中,所述第一采样概率小于所述第二采样概率;以及

基于所述第一方在第一状态下的所述N1个可能动作中的可能动作和所述第一方在第二状态下的所述N2个可能动作中的可能动作,执行所述CRF。

用于在多方策略互动中进行策略搜索的采样方案

技术领域

[0001] 本文涉及在两方或更多方之间的策略互动中进行策略搜索。

背景技术

[0002] 两方或更多方之间的策略互动可以通过涉及两方或更多方(也称为玩家)的博弈来建模。在涉及两个或更多个玩家的不完美信息博弈(imperfect information games，IIG)中，玩家在做出决策之前只能部分地了解其对手。这类似于现实场景，例如贸易、流量交通路线规划和公开拍卖。许多现实生活场景可以表示为IIG，例如不同公司之间的商业竞争、拍卖场景中的竞标关系、欺诈方和反欺诈方之间的博弈关系。

[0003] 求解IIG的方法具有很大的经济和社会效益。由于信息隐藏，因此玩家必须在对其对手信息不确定的情况下进行推理，并且玩家还需要采取行动以利用其对手对其自己的信息的不确定的优势。

发明内容

[0004] 本文的实施方式包括计算机实现的用于在多方策略互动中进行策略搜索的方法。更具体地，本文描述了用于在求解不完美信息博弈(IIG)时执行反事实遗憾最小化(CRF)算法的采样方案的示例，该方法可以降低计算的复杂度和方差同时提高CRF算法的收敛速度。本文还描述了用神经网络执行反事实遗憾最小化(CRF)的技术，由于神经网络的泛化能力，它可以节省存储空间并提供更快的收敛。

[0005] 本文中描述的在特定实施例中实现的主题实现了以下技术效果和优势中的一个或多个。在一些实施例中，所描述的采样技术可以以更有效的方式帮助寻找例如资源分配、产品/服务推荐、网络攻击预测和/或预防、流量交通路线规划、欺诈管理等的现实场景的更好策略，这些现实场景可被建模或表示为多方策略互动、例如涉及两方或更多方的IIG。在一些实施例中，所描述的技术可以提高反事实遗憾最小化(CRF)算法在寻找由IIG建模的现实场景的最佳策略时的计算效率并降低计算负荷。在一些实施例中，所描述的采样技术可以提供比结果采样更低的方差，同时比外部采样具有更高的存储效率。在一些实施例中，所描述的技术可以提高CRF算法在寻找纳什(Nash)均衡以求解代表一个或多个现实场景的博弈时的收敛速度。在一些实施例中，所描述的技术提供表示IIG的博弈树的更平衡和全面的信息，使得CRF算法可以具有更小的方差和更快的收敛速度。在一些实施例中，所描述的技术通过使用神经网络结合CRF算法，节省了存储空间并提供了更快的收敛。在一些实施例中，所描述的技术可以仅需要少量存储空间用于CRF算法的每次迭代。

[0006] 本文还提供了耦接一个或多个处理器并且其上存储有指令的一个或多个非暂时性计算机可读存储介质，所述指令当由所述一个或多个处理器执行时，促使所述一个或多个处理器按照本文提供的方法的实施例执行操作。

[0007] 本文还提供了用于实施本文提供的所述方法的系统。所述系统包括一个或多个处理器以及耦接到所述一个或多个处理器并且其上存储有指令的计算机可读存储介质，所述

指令当由所述一个或多个处理器执行时，促使所述一个或多个处理器按照本文提供的方法的实施例执行操作。

[0008] 应了解，依据本文的方法可以包括本文描述的方面和特征的任意组合。也就是说，根据本文的方法不限于本文具体描述的方面和特征的组合，还包括所提供的方面和特征的任意组合。

[0009] 以下在附图和描述中阐述了本文的一个或多个实施例的细节。根据说明书和附图以及权利要求，本文的其他特征和优点将显现。

附图说明

[0010] 图1是根据本文的实施例示出的单牌扑克中的部分博弈树的示例的图。

[0011] 图2是根据本文的实施例示出的不同采样方案的示例的图。

[0012] 图3是根据本文的实施例的鲁棒采样蒙特卡罗CFR(MCCFR)的示例的伪代码。

[0013] 图4是根据本文的实施例示出的应用于博弈树的双神经CFR算法的示例的图。

[0014] 图5是根据本文的实施例的双神经CFR算法的示例的伪代码。

[0015] 图6是根据本文的实施例的用于优化与双神经CFR算法相关的神经网络的算法的示例的伪代码。

[0016] 图7是根据本文的实施例的小批量MCCFR算法的示例的伪代码。

[0017] 图8是根据本文的实施例示出的用于执行MCCFR的采样处理的示例的流程图。

[0018] 图9是根据本文的实施例示出的双神经CFR算法的示例的流程图。

[0019] 图10描绘了根据本文的实施例示出的计算机实现的用于提供与所描述的算法、方法、功能、处理、流程和程序相关联的计算功能的系统的示例的框图。

[0020] 图11描绘了根据本文的实施例的装置的模块的示例。

[0021] 图12描绘了根据本文的实施例的另一装置的模块的示例。

[0022] 各附图中相同的附图标记和名称表示相同的元件。

具体实施方式

[0023] 本文的实施方式包括计算机实现的例如通过求解不完美信息博弈(IIG)在多方策略互动中进行策略搜索的方法。IIG可以代表例如资源分配、产品/服务推荐、网络攻击预测和/或预防、流量交通路线规划、欺诈管理等一个或多个现实场景，其中，涉及两方或更多方(也称为玩家)，且每方可以具有关于他方的决策的不完全或不完美信息。更具体地，本文描述了用于在求解IIG时执行反事实遗憾最小化(counterfactual regret minimization，CRF)算法的采样方案的示例，可以降低计算的复杂度和方差同时提高CRF算法的收敛速度。本文还描述了用神经网络执行反事实遗憾最小化(CRF)的技术，由于神经网络的泛化能力，它可以节省存储空间并提供更快的收敛。

[0024] 纳什均衡是涉及两个或更多个玩家的IIG的典型解法。反事实遗憾最小化(CRF)是一种旨在近似找到大型博弈的纳什均衡的算法。CRF试图最小化整体的反事实遗憾。事实证明，所有迭代中的多个策略的平均值会收敛到纳什均衡。在求解博弈时，原始形式的CRF(也称为原始CRF、标准CRF、普通CFR或简称CRF)在每次迭代中遍历整个博弈树。因此，原始CFR需要大容量存储器处理大型零和扩展式博弈，例如单挑无限注德州扑克。在某些情况下，原

始CFR可能无法用有限存储器处理大型博弈。

[0025] 引入蒙特卡洛CFR(Monte Carlo CFR,MCCFR)以使反事实遗憾最小化。MCCFR可以计算反事实值(counterfactual value)的无偏估计,并避免遍历整个博弈树。由于在每次迭代中仅访问所有信息集的子集,因此MCCFR所需的存储空间比原始CFR少。

[0026] 可以使用结果采样(outcome sampling)算法或外部采样(external sampling)算法来执行MCCFR。MCCFR中的结果采样算法的方差大,并且难以在较少的迭代步骤中收敛到近似纳什均衡解。MCCFR中的外部采样算法的方差比结果采样算法的方差小,但该方法的缺点类似于CFR的缺点。当博弈树很大时,它需要非常大的存储空间,并且不能扩展到复杂的大规模IIG。

[0027] 本文公开了一种鲁棒采样(robust sampling)方案。在所述鲁棒采样方案中,每个玩家在当前决策点使用均匀采样(uniform sampling)方法进行采样,而他方根据相应的策略进行采样。对应于不同迭代的到达概率可以是固定的。可以证明,该鲁棒采样方案的方差比MCCFR中的结果采样方案的方差小,同时比外部采样的存储效率高。在一些实施例中,该鲁棒采样方案可以使MCCFR以更快的收敛求解纳什均衡。

[0028] 本文公开了一种深度依赖采样方案。所述深度依赖采样方案可以将更高的采样概率分配给更接近终点(terminal)状态的状态,而不是与终点状态更远(或更接近初始状态或开始状态)的另一状态。在一些实施例中,所述深度依赖采样方案可以允许更接近终点状态的状态更多地被采样,从而提供IIG的更全面(well-around)信息,因此与现有采样方案相比提高了MCCFR的收敛速率。

[0029] 本文还公开了双神经CFR算法。现有的CFR方法,如CFR和MCCFR,使用两个基于表格的大容量存储器来记录所有信息集的累积遗憾和平均策略。这种表格表示形式使得这些方法难以以有限的时间和空间应用于大型扩展式的博弈。

[0030] 相比之下,双神经CFR算法使用两个神经网络来计算IIG的近似纳什均衡。例如,其中一个神经网络可用于学习累积遗憾,另一个神经网络可用于学习平均策略分布(strategy profile)的累积分子。在这两个网络的帮助下,双神经CFR算法不需要使用基于表格的两个大容量存储器。基于该紧凑神经网络的泛化能力,累积遗憾和平均策略可以被学习和生成。所公开的双神经CFR算法可以保持MCCFR需要较少的计算负荷的益处还不需要两个大容量表格式存储器。即使存在存储器限制,所公开的双神经CFR算法也可用于大型博弈中。在一些实施例中,相比于现有技术,双神经方法可以以更少的迭代实现更低的可利用性。另外,在一些实施例中,在从不良表格策略初始化之后,双神经CFR还可以持续改进。

[0031] 在一些实施例中,所描述的技术可以用于例如AI扑克、推荐平台和许多其他AI和机器学习的应用。所描述的技术使用蒙特卡罗方法,并且不需要针对整个博弈树的变量。

[0032] 在一些实施例中,具有玩家有限集合N＝{0,1,...,n-1}的扩展式博弈可以表示如下。定义h^v_i为IIG中玩家i的隐藏变量。例如,在扑克游戏中,h^v_i可以指玩家i的私人牌。H指的是历史的有限集合。H的每个成员$h = (h_i^v)_{i=0,1,\dots,n-1}(a_l)_{l=0,\dots,L-1} = h_0^v h_1^v \dots h_{n-1}^v a_0 a_1 \dots a_{L-1}$表示可能的历史(或状态),其包括每个玩家的隐藏变量和包括机会的玩家所采取的L个动作。对于玩家i,h也可以表示为$h_i^v h_{-i}^v a_0 a_1 \dots a_{L-1}$,其中$h_{-i}^v$指的是对手的隐藏变量。空序列$\emptyset$是H的成员。表达式$h_j \sqsubseteq h$表示$h_j$是h的前提(prefix),其中

$h_j = (h_i^v)_{i=0,1,\ldots,n-1}(a_l)_{l=1,\ldots L'-1}$且0<L'<L。$Z \subseteq H$表示终点历史，并且任何成员z∈Z不是任何其他序列的前提。A(h)＝{a:ha∈H}是非终点历史h∈H\Z之后的可用动作集合。玩家函数P将N∪{c}的成员分配给每个非终点历史，其中c表示机会玩家标识符(ID)，其通常可以是例如-1。P(h)是在历史h之后采取动作的玩家。

[0033] 历史{h∈H:P(h)＝i}的I_i是玩家i的信息分划。集合$I_i \in I_i$是玩家i的信息集。I_i(h)指的是状态h下的信息集I_i。在一些实施例中，I_i只能记住玩家i观察到的信息，包括玩家i的隐藏变量和公开动作。因此，I_i表示IIG中的序列，即$h^v_i a_0 a_2 \ldots a_{L-1}$。在一些实施例中，对于$I_i \in I_i$和对于任何$h \in I_i$，集合A(h)可以由$A(I_i)$表示，并且玩家P(h)由$P(I_i)$表示。对于每个玩家i∈N，效用函数$u_i(z)$定义终点状态z的收益。下面包括图1中所示的示例将讨论这些符号和定义的更详细解释。

[0034] 图1是根据本文的实施例示出的单牌扑克中的部分博弈树102和部分博弈树104的示例的图100。单牌扑克是双玩家IIG扑克。单牌扑克是扩展式博弈的示例。

[0035] 博弈树是有向图。博弈树的节点表示博弈中的位置(或玩家状态)，并且博弈树表示的节点可以表示博弈的玩家的行动或动作。在图1中，z_i表示终点节点，表示终点状态，h_i表示非终点节点。部分博弈树102和104各自具有表示机会的根节点h_0。在第一部分树102中存在19个不同的节点，对应于左侧树中的包括机会h_0的9个非终点节点h_i和10个终点节点z_i。

[0036] 在第一部分树102中，两个玩家(玩家0和玩家1)被发牌(Q,J)，如左子树中的“0:Q 1:J”和(Q,J)，如右子树中所示的“0:Q 1:K”。

[0037] 从根节点到每个节点的轨迹是动作的历史。动作由博弈树的边(用箭头表示)旁边的字母(例如，F、C、P和B)或表达式(例如，“0:Q 1:J”)表示。

[0038] 在扩展式博弈中，h_i指的是动作的历史。例如，如第一部分树102中所示，h_3包括动作0:Q 1:J和P。h_7包括动作0:Q 1:J、P和B。h_8包括动作0:Q1:K、P和B。在第一部分树102中，$h_3 \sqsubseteq h_7$，即h_3是h_7的前提。$A(h_3)$＝{P,B}表示非终点历史h_7之后的可用动作组是P和B。$P(h_3)$＝1表示在历史h_3之后采取动作的玩家是玩家1。

[0039] 在该IIG中，玩家1的私人牌对于玩家0是不可见的，因此对于玩家0来说h_7和h_8实际上是相同的。信息集可用于表示这些不可区分状态的集合。类似地，h_1和h_2在同一信息集中。对于右部分树104，h'_3和h'_5在相同的信息集中；h'_4和h'_6在相同的信息集中。

[0040] 通常，任何$I_i \in I$只能记住玩家i观察到的信息，包括玩家i的隐藏变量和公开动作。例如，如第一部分树102中所示，h_7和h_8的信息集表示0:Q、P和B的序列。由于h_7和h_8未被IIG中的玩家0区分，如果I_0是h_7和h_8的信息集，则$I_0 = I_0(h_7) = I_0(h_8)$。

[0041] 策略分布$\sigma = \{\sigma_i | \sigma_i \in \Sigma_i, i \in N\}$是所有玩家策略的集合，其中$\Sigma_i$是玩家i的所有可能策略的集合。$\sigma_{-i}$指的是除了玩家i之外的所有玩家的策略。对于玩家i∈N，策略$\sigma_i(I_i)$是函数，它将$A(I_i)$上的动作分布分配给信息集I_i。$\sigma_i(a|h)$表示在状态h由玩家i∈N∪{c}采取动作a的概率。在IIG中，如果两个或更多个状态具有相同的信息集，则这两个或更多个状态具有相同的策略。即，$\forall h_1, h_2 \in I_i$，$I_i = I_i(h_1) = I_i(h_2)$，$\sigma_i(I_i) = \sigma_i(h_1) = \sigma_i(h_2)$，$\sigma_i(a|I_i) = \sigma_i(a|h_1) = \sigma_i(a|h_2)$。例如，$I_0$是$h_7$和$h_8$的信息集，则$I_0 = I_0(h_7) = I_0(h_8)$，$\sigma_0(I_0) = \sigma_0(h_7) = \sigma_0(h_8)$，$\sigma_0(a|I_0) = \sigma_0(a|h_7) = \sigma_0(a|h_8)$。在图1中，针对每个状态，除灰色之外的相同颜色在

相同信息集中。

[0042] 对于玩家i，策略分布σ的预期博弈效用被表示为$u_i^{\sigma} = \sum_{z \in Z} \pi^{\sigma}(z) u_i(z)$，其是所有可能终点节点的预期收益。给定固定策略分布σ_{-i}，玩家i的针对π_{-i}^{σ}实现收益最大化的任何策略$\sigma_i^* = \arg\max_{\sigma_i' \in \sum_i} u_i^{(\sigma_i', \sigma - i)}$是最佳回应。对于双玩家的扩展式博弈，纳什均衡是策略分布$\sigma^* = (\sigma_0^*, \sigma_1^*)$，以使每个玩家的策略是对对手的最佳回应。∈-纳什均衡是纳什均衡的近似，其策略分布σ*满足：$\forall_i \in N, u_i^{\sigma_i} + \epsilon \geq max_{\sigma_i' \in \sum_i} u_i^{(\sigma_i', \sigma - i)}$。

[0043] 策略σ_i的可利用性可以定义为$\epsilon_i(\sigma_i) = u_i^{\sigma^*} - u_i^{(\sigma_i, \sigma_{-i}^*)}$。如果$\in_i(\sigma_i) = 0$，则策略是不可利用的。在像扑克这样的大型双玩家零和游戏中，$u_i^{\sigma^*}$很难计算。然而，如果玩家交换他们的位置，则一对游戏的值为零，即$u_0^{\sigma^*} + u_1^{\sigma^*} = 0$。策略分布σ的可利用性可以定义为$\epsilon(\sigma) = (u_1^{(\sigma_0, \sigma_1^*)} + u_0^{(\sigma_0^*, \sigma_1)})/2$。

[0044] 对于诸如CFR的迭代方法，σ^t指的是第t次迭代的策略分布。如果玩家根据σ采取动作，则历史h的状态到达概率可以用$\pi^{\sigma}(h)$表示。对于空序列，$\pi^{\sigma}(\phi) = 1$。可以根据每个玩家的贡献将到达概率分解为$\pi^{\sigma}(h) = \prod_{i \in N \cup \{c\}} \pi_i^{\sigma}(h) = \pi_i^{\sigma}(h) \pi_{-i}^{\sigma}(h)$，其中$\pi_i^{\sigma}(h) = \prod_{h' a \sqsubseteq h, P(h') = P(h')} \sigma_i(a|h')$和$\pi_{-i}^{\sigma}(h) = \prod_{h' a \sqsubseteq h, P(h') \neq P(h)} \sigma_{-i}(a|h')$。

[0045] 信息集I_i的到达概率（也称为信息集到达概率）可以定义为$\pi^{\sigma}(I_i) = \sum_{h \in I_i} \pi^{\sigma}(h)$。如果$h' \sqsubseteq h$，从状态h′到h的间隔状态到达概率可以定义为$\pi^{\sigma}(h', h)$，则$\pi^{\sigma}(h', h) = \pi^{\sigma}(h) / \pi^{\sigma}(h')$。到达概率$\pi_i^{\sigma}(I_i)$、$\pi_{-i}^{\sigma}(I_i)$、$\pi_i^{\sigma}(h', h)$和$\pi_{-i}^{\sigma}(h', h)$可以被类似地定义。

[0046] 在大型零和IIG中，CFR被证明是用于计算纳什均衡的有效方法。可以证明，一个玩家的状态到达概率与对手的隐藏变量的后验概率成比例，即，$p(h_{-i}^v | I_i) \propto \pi_{-i}^{\sigma}(h)$，其中$h^v_i$和$I_i$表示特定h。

[0047] 对于玩家i和策略分布σ，状态h下的反事实值(CFV)$v_i^{\sigma}(h)$可以定义为

[0048] $$v_i^{\sigma}(h) = \sum_{h \sqsubseteq z, z \in Z} \pi_{-i}^{\sigma}(h) \pi^{\sigma}(h, z) u_i(z) = \sum_{h \sqsubseteq z, z \in Z} \pi_i^{\sigma}(h, z) u'_i(z) \quad (1)$$

[0049] 其中$u'_i(z) = \pi_{-i}^{\sigma}(z) u_i(z)$是玩家i相对于对手的隐藏变量的近似后验分布的预期奖励。采取动作a的动作反事实值可以表示为$v_i^{\sigma}(a|h) = v_i^{\sigma}(ha)$，并且采取该动作的遗憾是$r_i^{\sigma}(a|h) = v_i^{\sigma}(a|h) - v_i^{\sigma}(h)$。

[0050] 类似地，信息集I_i的CFV可以定义为$v_i^{\sigma}(I_i) = \sum_{h \in I_i} v_i^{\sigma}(h)$，给定信息集$I_i$，动作a的遗憾可以定义为

[0051] $$r_i^{\sigma}(a|I_i) = \sum_{z \in Z, ha \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(ha, z) u'_i(z) - \sum_{z \in Z, h \sqsubseteq z, h \in I_i} \pi_i^{\sigma}(h, z) u'_i(z) \quad (1a)$$

[0052] 然后,可以根据式(2)计算动作a在T次迭代之后的累积遗憾:

[0053] $$R_i^T(a|I_i) = \sum_{t=1}^{T}\left(v_i^{\sigma^t}(a|I_i) - v_i^{\sigma^t}(I_i)\right) = R_i^{T-1}(a|I_i) + r_i^{\sigma^T}(a|I_i) \quad (2)$$

[0054] 其中$R_i^0(a|I_i)=0$。定义$R_i^{T,+}(a|I_i)=\max(R_i^T(a|I_i),0)$,可以例如基于遗憾匹配,根据如下式(3)更新T+1迭代处的当前策略(或行为策略):

[0055] $$\sigma_i^{T+1}(a|I_i) = \begin{cases} \dfrac{R_i^{T,+}(a|I_i)}{\sum_{a\in A(I_i)} R_i^{T,+}(a|I_i)}, & \text{如果 } \sum_{a\in A(I_i)} R_i^{T,+}(a|I_i) > 0 \\ \dfrac{1}{|A(I_i)|}, & \text{否则} \end{cases} \quad (3)。$$

[0056] 从迭代1到T的平均策略$\bar{\sigma}_i^T$可以定义为:

[0057] $$\bar{\sigma}_i^T(a|I_i) = \frac{\sum_{t=1}^{T}\pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i)}{\sum_{t=1}^{T}\pi_i^{\sigma^t}(I_i)} \quad (4)$$

[0058] 其中,$\pi_i^{\sigma^t}(I_i)$表示在第t次迭代时I_i的信息集到达概率,并用于对相应的当前策略$\sigma_i^t(a|I_i)$进行加权。

[0059] 在迭代t中定义$s^t(a|I_i) = \pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i)$为附加分子,然后可以将平均策略$\bar{\sigma}_i^T$的累积分子定义为

[0060] $$S^T(a|I_i) = \sum_{t=1}^{T}\pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i) = S^{T-1}(a|I_i) + s_i^T(a|I_i) \ , \quad (5)$$

[0061] 其中$S^0(a|I_i)=0$。

[0062] 在求解博弈时,原始CFR在每次迭代中遍历整个博弈树。因此,原始CFR可能无法用有限的存储器处理大型博弈。引入蒙特卡洛CFR(MCCFR)以使反事实遗憾最小化。MCCFR可以计算反事实值的无偏估计并避免遍历整个博弈树。由于在每次迭代中仅访问所有信息集的子集,因此MCCFR所需的存储空间比原始CFR少。

[0063] 例如,定义$Q=\{Q_1,Q_2,\ldots,Q_m\}$,其中$Q_j\in Z$是每次迭代中的采样终点历史的块,使得Q_j涵盖集合Z。通常,根据指定的采样方案,不同的Q_j可能会重叠。可以使用几种采样方案。

[0064] 图2是根据本文的实施例示出的不同采样方案的示例的图200。具体地,子图A示出了博弈树的外部采样方案202的示例;子图B示出了博弈树的结果采样方案204的示例,子图C示出了博弈树的鲁棒采样方案206的示例。

[0065] 如图2中所示,圆圈表示玩家0节点,矩形表示玩家1节点,三角形表示机会节点。实线边或箭头表示被采样的动作,而虚线边或箭头表示未被采样的动作。阴影节点表示被采样的节点,而空白节点表示未被采样的节点。

[0066] 以玩家0的更新为例,用如子图A所示的外部采样方案202,玩家0节点遍历玩家0节点的所有分支,非玩家0节点(例如,玩家1节点和机会节点)根据相应采样策略对分支进行随机采样。

[0067] 结果采样方案不区分不同玩家。如子图B所示,结果采样方案204根据相应采样策略随机地针对所有玩家对一个分支进行采样。因此,在结果采样方案下只会对一个轨迹进行采样。

[0068] 如子图C所示，鲁棒采样方案206根据玩家0的均匀分布随机选择k个分支，并根据相应采样策略对非玩家0节点的一个分支进行随机采样。通过改变k的值，鲁棒采样方案可以对多个路径或单个路径进行采样，例如，取决于实际存储器需求或系统规范。与外部采样方案不同，鲁棒采样方案不需要每次都了解在当前玩家i的决策点的所有可能动作和变量。

[0069] 在一些实施例中，在外部采样和结果采样方案中，每个块$Q_j \in Q$表示Z的分划。定义q_{Q_j}为考虑块Q_j的概率，其中$\sum_{j=1}^{m} q_{Q_j} = 1$。定义$q(z) = \sum_{j:z\in Q_j} q_{Q_j}$为考虑特定终点历史z的概率。在一些实施例中，普通CFR可以被认为是MCCFR的特殊情况，其中$Q=\{Z\}$仅包含一个块并且$q_{Q1}=1$。

[0070] 在结果采样方案中，将仅对一个轨迹进行采样，使得$\forall Q_j \in Q$，$|Q_j|=1$且$|Q_j|=|Z|$。对于信息集I_i，反事实值的样本估计是

[0071] $$\tilde{v}_i^{\sigma}(I_i|Q_j) = \sum_{h\in I_i, z\in Q_j, h\sqsubseteq z} \frac{1}{q(z)} \pi_{-i}^{\sigma}(z)\pi_i^{\sigma}(h,z)u_i(z)。$$

[0072] 证明了MCCFR中的采样反事实值是CFR中实际反事实值的无偏估计：$E_{j\sim q_{Q_j}}\left[\bar{v}_i^{\sigma}(I_i|Q_j)\right] = v_i^{\sigma}(I_i)$。

[0073] 定义σ^{rs}为采样策略分布，其中σ^{rs}是玩家i的采样策略，并且σ_{-i}^{rs}是除玩家i之外的玩家的采样策略。在一些实施例中，对于外部采样和结果采样，$\sigma_{-i}^{rs} = \sigma_{-i}$。采样动作$a\in A(I_i)$的遗憾可以定义为：

[0074] $$\tilde{r}_i^{\sigma}\left((a|I_i)\middle|Q_j\right) = \sum_{z\in Q_j, ha\sqsubseteq z, h\in I_i} \pi_i^{\sigma}(ha,z)u_i^{rs}(z) - \sum_{z\in Q_j, h\sqsubseteq z, h\in I_i} \pi_i^{\sigma}(h,z)u_i^{rs}(z) \quad (6)$$

[0075] 其中$u_i^{rs}(z) = \frac{u_i(z)}{\pi_i^{\sigma^{rs}}(z)}$是由$\frac{1}{\pi_i^{\sigma^{rs}}(z)}$加权的新效用。可以将T次迭代后的动作a的累积遗憾的样本估计定义为$\tilde{R}_i^{T}\left((a|I_i)\middle|Q_j\right) = \tilde{R}_i^{T-1}\left((a|I_i)\middle|Q_j\right) + \tilde{r}_i^{\sigma^T}((a|I_i)|Q_j)$且$\tilde{R}_i^{0}((a|I_i)|Q_j) = 0$。

[0076] 对于鲁棒采样，可以将采样简档定义为$\sigma^{rs(k)} = (\sigma_i^{rs(k)}, \sigma_{-i})$，其中玩家i可以根据针对每个信息集$I_i$的采样策略$\sigma_i^{rs(k)}(I_i)$随机选择k个动作，并且其他玩家可以根据策略$\sigma_{-i}$随机选择一个动作。

[0077] 在一些实施例中，如果玩家i根据在信息集I_i的离散均匀分布unif(0,$|A(I_i)|$)随机选择min(k,$|A(I_i)|$)个动作。即$\sigma_i^{rs(k)}(a|I_i) = \frac{\min(k,|A(I_i)|)}{|A(I_i)|}$，那么在玩家i根据采样策略或简档$\sigma_i^{rs(k)}$采取动作的情况下，信息集$I_i$的到达概率可以通过下式计算：

[0078] $$\pi_i^{\sigma^{rs(k)}}(I_i) = \prod_{h\in I_i, h'\sqsubseteq h, h'a\sqsubseteq h, h'\in I_i'} \frac{\min(k, A(I_i'))|}{|A(I_i')|},$$

[0079] 并且在给定采样简档$\sigma^{rs(k)}$的情况下，加权效用$u_i^{rs(k)}(z)$在每次迭代中可以是常数，其具有低方差。此外，由于加权效用不再需要明确了解对手的策略，因此鲁棒采样可以用于在线遗憾最小化。

[0080] 为简化符号，令k＝max表示$k=\max_{I_i\in I}|A(I_i)|$。如果k＝max和玩家i根据信息集$I_i$的离散均匀分布unif(0,|A($I_i$)|)随机选择k个动作，$\forall_i\in N,\ \forall I_i\in L_i,\ \forall_a\in A(I_i),\ \sigma_i^{rs(k)}(a|I_i)\sim unif(0,|A(I_i)|)$，则鲁棒采样可以类似于$k=\max_{I_i\in I}|A(I_i)|$时的外部采样。

[0081] 如果k＝1且$\sigma_i^{rs(k)}=\sigma_i$，在这种情况下只对一个历史z进行采样，那么对于$a\in A^{rs(k)}(I_i)$，$u_i^{rs(1)}(z)=\frac{u_i(z)}{\pi^{\sigma^{rs(k)}}(z)},\ \exists h\in I_i,$

[0082]
$$\begin{aligned}\tilde{r}_i^{\sigma}\left((a|I_i)\middle|Q_j\right)&=\tilde{r}_i^{\sigma}\left((a|h)\middle|Q_j\right)\\&=\sum_{z\in Q_j,ha\sqsubseteq z,h\in I_i}\pi_i^{\sigma}(ha,z)u_i^{rs}(z)-\sum_{z\in Q_j,h\sqsubseteq z,h\in I_i}\pi_i^{\sigma}(h,z)u_i^{rs}(z)\\&=\frac{(1-\sigma_i(a|h))u_i(z)}{\pi_i^{\sigma}(ha)}\end{aligned}$$

[0083] 如果在状态h没有对动作a进行采样，即$a\notin A^{rs(k)}(I_i)$，则遗憾是$\tilde{r}_i^{\sigma}((a|h)|j)=0-\tilde{v}_i^{\sigma}(h|j)$。在这种情况下，鲁棒采样类似于k＝1且$\sigma_i^{rs(k)}=\sigma_i$时的结果采样。

[0084] 如果k＝1，且玩家i根据信息集I_i的离散均匀分布unif(0,|A(I_i)|)随机选择一个动作，则鲁棒采样可以与结果采样类似。例如，如果k＝1，并且玩家i根据信息集I_i的离散均匀分布unif(0,|A(I_i)|)随机选择一个动作，则

[0085] 对于$a\in A^{rs(k)}(I_i)$，$u_i^{rs(1)}(z)=\frac{u_i(z)}{\pi^{\sigma^{rs(k)}}(z)}$是常数，$\exists h\in I_i,$

[0086]
$$\begin{aligned}\tilde{r}_i^{\sigma}\left((a|I_i)\middle|Q_j\right)&=\sum_{z\in Q_j,ha\sqsubseteq z,h\in I_i}\pi_i^{\sigma}(ha,z)u_i^{rs}(z)-\sum_{z\in Q_j,h\sqsubseteq z,h\in I_i}\pi_i^{\sigma}(h,z)u_i^{rs}(z)\\&=(1-\sigma_i(a|h))\pi_i^{\sigma}(ha,z)u_i^{rs(1)}(z)。\end{aligned}$$

[0087] 如果在状态h没有对动作a进行采样，即$a\notin A^{rs(k)}(I_i)$，则遗憾是$\tilde{r}_i^{\sigma}((a|h)|j)=0-\tilde{v}_i^{\sigma}(h|j)$。与结果采样相比，在这种情况下的鲁棒采样的方差由于常数$u_i^{rs(1)}(z)$而较小。

[0088] 图3是根据本文的实施例的鲁棒采样MCCFR的示例的伪代码300。如伪代码300的第1-5行所示，整体鲁棒采样MCCFR是输入为总迭代次数t的迭代算法。在每次迭代t内，针对玩

家0和玩家1调用鲁棒采样MCCFR(RS-MCCFR)函数(如第3和第4行所示),以更新累积遗憾$R^t(\cdot|I_i)$和平均策略分子$S^t(a|I_i)$。函数RS-MCCFR可以定义为如伪代码300的第6-30行中所示。函数RS-MCCFR返回每个信息集的反事实值作为输出。在一些实施例中,反事实值可用于计算反事实遗憾。因此,可以相应地获得累积遗憾和时间平均策略。

[0089] 具体地,函数RS-MCCFR根据如上结合图2所述的鲁棒采样方案对多个动作进行采样。如伪码300的第16行所示,可以根据鲁棒采样策略$\sigma_i^{rs(k)}$对k个不同的动作进行采样并收集为$A^{rs(k)}(I_i)$。

[0090] 在一些实施例中,可以使用深度依赖采样方案来提供由博弈树表示的博弈的更平衡或全面的信息。例如,采样策略$\sigma_i^{rs}(a|h)$可以是博弈树中状态h的深度的函数。例如,可以设计采样策略$\sigma_i^{rs}(a|h)$,使得更接近终点状态的状态的采样概率比接近初始状态的状态(例如,由博弈树的根表示的状态)的采样概率高。作为示例,可以通过将不同的权重赋予具有不同深度的不同状态的采样概率来实现深度依赖采样方案。这种深度依赖采样可以帮助提供关于更靠近终点状态的状态的更多信息,这可以是有利的,因为通常更接近终点状态的状态比更接近初始状态的状态要多(例如,由于博弈树的分支结构),并且在现有的采样方案下,这些节点的被采样的可能性低于采样轨迹中更接近根节点的节点的被采样的可能性。

[0091] 在一些实施例中,深度依赖采样方案可以与鲁棒采样、结果采样、外部采样或任何其他合适的采样算法结合使用。例如,深度依赖采样方案可以进一步改善鲁棒采样、结果采样和外部采样中任一项的方差和收敛速度,因为后三种采样方案更多地关注于玩家状态博弈树的不同动作之间的水平采样(例如,表示为博弈树的一个节点的不同分支)。

[0092] 图4是根据本文的实施例示出的应用于博弈树410的双神经CFR算法的示例400的图。双神经CFR算法400使用两个神经网络420和430来计算IIG,例如由博弈树410表示,的近似纳什均衡。如图4中所示,一个神经网络420用于获得累积遗憾并且被称为RegretSumNetwork(RSN)。另一神经网络430用于获得平均策略,并且被称为AveStrategyNetwork(ASN)。

[0093] 在一些实施例中,CFR算法的迭代更新维持两种策略:当前策略$\sigma_i^t(a|I_i)$以及针对$\forall i\in N, \forall I_i\in I_i, \forall a\in A(I_i), \forall t\in\{1,...,T\}$的平均策略$\bar{\sigma}_i^t(a|I_i)$。因此,两个神经网络420和430可以被设计为分别以迭代方式记录这两个策略。在一些实施例中,双神经CFR算法的示例400可以被称为双增量CFR算法,因为是基于每次迭代中新的、额外的样本来训练或优化该神经网络的。

[0094] 根据公式(3),可以通过累积遗憾$R^t(a|I_i)$来计算当前策略$\sigma^{t+1}(a|I_i)$。在一些实施例中,仅公式(3)中的分子被跟踪,因为当使用该策略时,可以容易地计算分母的归一化。给定信息集I_i和动作a,表示为$\mathcal{R}(a,I_i|\theta_R^t)$的神经网络RSN 420可以用于学习$R^t(a|I_i)$,其中$\theta_R^t$是在第t次迭代时RSN 420中的参数。

[0095] 如图4所示,存储器M_R^t可以定义为$M_R^t=\{(I_i,\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j))|\forall i\in N,\forall a\in A(I_i),h\in I_i,h\vee z,z\in Q_j\}$。$M_S^t$的每个成员可以包括所访问的信息集$I_i$和相应遗憾

$\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j)$，其中$Q_j$是第t次迭代中的采样块。根据式(2)，$\mathcal{R}(a,I_i|\theta_R^{t+1})$可以使用以下优化进行估计：

[0096] $$\theta_R^{t+1} \leftarrow \underset{\theta_R^{t+1}}{\operatorname{argmin}} \sum_{(I_i,\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j))\in M_R^t} (\mathcal{R}(a,I_i|\theta_R^t) + \tilde{r}_i^{\sigma^t}((a|I_i)|Q_j) - \mathcal{R}(a,I_i|\theta_R^{t+1}))^2, \quad (7)$$

[0097] 根据式(4)，近似纳什均衡是T次迭代中所有先前策略的加权平均。类似于累积遗憾，被表示为$S(a,I_i|\theta_S^t)$的另一神经网络ASN 430可用于学习平均策略分子。将另一存储器M_S^t定义为

[0098] $$M_S^t = \{(I_i, s^t(a|I_i))|\forall_i \in N, \forall_a \in A(I_i), h \in I_i, h \sqsubseteq z, z \in Q_j\} = \{(I_i, \pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i))|\forall_i \in N, \forall_a \in A(I_i), h \in I_i, h \sqsubseteq z, z \in Q_j\}。$$

[0099] M_S^t的每个成员可以包括访问信息集I_i和$\pi_i^{\sigma t}(I_i)\sigma_i^t(a|I_i)$的值，其中$Q_j$是第t次迭代中的采样块。然后可以通过下式估计ASN 430的参数θ_S^{t+1}：

[0100] $$\theta_S^{t+1} \leftarrow \underset{\theta_S^{t+1}}{\operatorname{argmin}} \sum_{(I_i, s_i^t(a|I_i))\in M_S^t} (S(a,I_i|\theta_S^t) + s_i^t(a|I_i) - S(a,I_i|\theta_S^{t+1}))^2 \quad (8)$$

[0101] 在一些实施例中，在每次迭代中，$\mathcal{R}(a,I_i|\theta_R^t)$和$S(a,I_i|\theta_S^t)$两者都可以根据梯度下降法通过优化公式(7)和式(8)中的目标进行优化，例如相对于以下图7描述的小批量随机梯度下降法。

[0102] 在一些实施例中，如果大容量存储器可用于在多次迭代内汇总并保存$s_i^t(a|I_i)$，则平均策略不需要在每次迭代中更新。如果存储器M_S^t已满，则可以通过优化公式(8)来学习增量值。

[0103] 在一些实施例中，在每次迭代中，仅对信息集的小的子集进行采样，这可能导致神经网络RSN 420和ASN 430忘记那些未观察到的或未采样的信息集的值。为了解决这个问题，来自先前迭代的神经网络参数可以用作当前迭代的初始值，为更新提供了在线学习/适应特色。此外，由于神经网络的泛化能力，即使来自信息集的少量样本也可用于更新新神经网络，并且新更新的神经网络可以为累积遗憾和平均策略产生良好的值。

[0104] 在一些实施例中，随着迭代次数t增加，$R_i^t(a|I_i)$的值可能变得越来越大，可能使得神经网络难以学习。为了解决这个问题，累积遗憾可以通过$\sqrt{t}$的因子进行归一化来使其范围更稳定。这可以根据在线学习的遗憾约束来理解。更具体地，令$\Delta = \max_{Ii,a,t}|R^t(a|I_i) - R^{t-1}(a|I_i)|$，$\forall I_i \in \mathrm{I}, a \in A(I_i), t \in \{1,\ldots,T\}$。$R_i^t(a|I_i) \leq \Delta\sqrt{|A|t}$，其中$|A| = \max_{Ii\in I}|A(I_i)|$。在一些实施例中，神经网络RSN 420可用于跟踪$\hat{R}_i^t(a|I_i) = R_i^t(a|I_i)/\sqrt{t}$并通过下式更新它：

[0105] $$\hat{R}_i^t(a|I_i) = \frac{\sqrt{t-1}\hat{R}_i^{t-1}(a|I_i)}{\sqrt{t}} + \frac{r_i^{\sigma^t}(a|I_i)}{\sqrt{t}}\ , \qquad (9)$$

[0106] 其中$\hat{R}_i^0(a|I_i)=0$。

[0107] 在一些实施例中，例如，由于存储器M_R^t和M_S^t的容量有限，在双增量CFR算法中，存储器M_R^t和M_S^t可以在每次迭代之后被清除。在一些实施例中，例如在大型博弈中，即使是使用神经网络来学习累积遗憾和平均策略的双增量CFR算法，存储器M_R^t和M_S^t的容量仍然可能需要非常大才能记录每次迭代的累积遗憾和平均策略。

[0108] 在一些实施例中，为了用有限存储器但无限迭代来持续改进平均策略和/或进一步减轻对存储器容量的要求，可以使用利用两个贮存器(reservoir)M_R和M_S的双贮存器CFR算法贮存器来保存不同迭代中的采样累积遗憾以及平均策略，以动态学习累积遗憾和平均策略。

[0109] 在一些实施例中，可以根据重写公式(2)的公式(10)获得T次迭代之后的平均累积遗憾：

[0110] $$\bar{R}_i^T(a|I_i) = R_i^T(a|I_i)/T \qquad (10)$$

[0111] 类似地，平均策略可以是累积策略的归一化，如公式(4)所示，是由其到达概率$\pi_i^{\sigma^t}(I_i)$进行加权的平均策略$\sigma_i^t(a|I_i)$ 。

[0112] 在一些实施例中，两个一样的贮存器M_R和M_S可用于分别保存采样的$\tilde{r}_i^{\sigma^t}((a|I_i)|Q_j)$和$s^t(a|I_i)$。具体而言，$M_R$可以保存$M_R^t$中的样本的贮存器，而$M_S$可以保存$M_S^t$中的样本的贮存器。可以通过贮存器采样算法将新样本插入到贮存器中。贮存器采样包括一系列随机算法，用于从包含n个项目的列表中随机选择k个项目。例如，如果贮存器未满，则可以将新样本直接添加到贮存器中。如果贮存器已满，则新样本可以根据先进先出(FIFO)原则或根据均匀随机分布或其他分布替换旧样本。

[0113] 请注意，双增量CFR算法和双贮存器CFR算法均采用在线学习的思想，并分别使用两个神经网络来学习更新遗憾和平均策略。在一些实施例中，不需要在每次迭代中更新ASN，而在蒙特卡洛采样之后可能需要对RSN进行优化以产生新的行为策略。例如，当使用新的行为策略遍历博弈树时，可能需要在每次迭代时更新RSN。另一方面，ASN可以用作最终的近似纳什均衡，其是行为策略的加权平均。ASN可以用作双神经CFR算法的输出。如果有足够大的数据存储设备来保存所有样本，则只需要在最后一步优化平均策略。实际上，对于大型博弈，大容量数据存储设备可能非常昂贵。这样，如果数据存储设备(例如，贮存器M_S^t)已满，则可以逐步优化平均策略。这样，双神经CFR算法可以包括两个变体。在双增量算法中，仅通过增量样本对神经网络(例如，RSN和/或ASN)进行优化，而在双贮存器算法中，可以通过贮存器中的所有样本对神经网络(例如，RSN和/或ASN)进行优化。

[0114] 双神经CFR算法和双贮存器CFR算法具有不同的样本集。对于双增量CFR，基于新增的多个样本对神经网络进行优化。对于双贮存器CFR，基于固定大小的贮存器中的多个样本对神经网络进行优化。另外，在双贮存器法中，平均策略可以通过最大对数似然而不是最小

均方误差来优化。

[0115] 图5是根据本文的实施例的双神经CFR算法的示例的伪代码500。该双神经CFR算法的示例包括使用双神经CFR算法或双贮存器CFR算法的选项。

[0116] 伪代码500的第3-7行显示了第一次迭代中初始化策略的示例。例如，如果系统根据现有CFR方法(例如基于表格的CFR或MCCFR方法或该双神经CFR方法)热启动，则可以根据现有策略分布初始化神经网络以克隆累积遗憾和策略。如果没有热启动初始化，则双神经CFR算法可以在迭代t＝1时通过随机初始化RSN和ASN中的参数来启动。

[0117] 在一些实施例中，如果使用双增量CFR算法，如伪代码500的第8行所示，则可以使用采样方法来返回该迭代中被采样信息集的反事实遗憾和平均策略分子。该迭代中被采样信息集的反事实遗憾和平均策略分子可以分别保存在存储器M_R^t和M_S^t中。采样方法可以包括例如如图7描述的小批量鲁棒采样方法算法。在一些实施例中，例如，根据下式(11)和如图7描述的小批量MCCFR算法，此次迭代中多个被采样信息集的多个反事实遗憾可以通过信息集进行累加，以在M_R^t中提供总值。在一些实施例中，可以删除M_S^t中的重复记录。

[0118] 在一些实施例中，如果使用双贮存器CFR算法，则该迭代中被采样信息集的反事实遗憾和平均策略分子(例如在双增量CFR算法中保存在存储器M_R^t和M_S^t中的那些)可以分别被保存在贮存器M_R和M_S中。如果贮存器M_R和/或M_S已满，则可以使用贮存器采样。

[0119] 如伪代码500的第13-15行所示，该迭代中被采样信息集的这些反事实遗憾和平均策略分子可以由图6所示的NeuralAgent算法使用，以优化两个神经网络RSN和ASN，并返回RSN和ASN(例如，$\mathcal{R}(a,I_i|\theta_R^t)$和$S(a,I_i|\theta_S^t)$)的参数(例如$\theta_R^t$和$\theta_S^t$)。

[0120] 图6是根据本文的实施例的用于优化与双神经CFR算法相关的神经网络的算法的示例的伪代码600。该算法的示例称为NeuralAgent算法。所描述的双神经CFR算法可以使用其他算法来优化在双神经CFR算法中使用的一个或两个神经网络。

[0121] 将β_{epoch}定义为训练回合(epoch)，β_{lr}定义为学习率，β_{loss}定义为提前停止或终止的标准，β_{re}定义为自上次获得最小损失以来的迭代次数的上限，θ^{t-1}定义为要优化的参数，$f(\cdot|\theta^{t-1})$定义为神经网络，M定义为训练样本，由信息集和相应目标组成。为了简化表示，使用β^*表示神经网络的参数集。例如，$\beta_{\mathcal{R}}^*$和β_S^*分别表示RSN和ASN中的参数集。实验表明，经过精心设计的NeuralAgent算法在优化神经网络RSN和ASN方面可以获得相对较高的可利用性收敛速度。伪代码600以解释性注释示出了NeuralAgent算法的细节。

[0122] 在一些实施例中，现有的优化器可能因潜在的鞍点或局部最小值而不会返回足够低的损失。为了获得相对较高的精度和较低的优化损失，专门设计了一种调度器，以在损失已停止减少时降低学习率。具体地，调度器读取度量值量，例如均方误差，并且如果若干回合内没有看到改善，则学习率被降低了一个因子。另外，一旦损失在β_{re}回合停止减少，就可以在优化器和调度器中重置学习率。可以使用梯度裁剪机制来限制参数梯度的大小，并使优化器在陡崖附近表现更好。在每个回合之后，最佳参数将被更新。一旦最低损失小于规定的标准β_{loss}，就使用提前停止机制。

[0123] 在实验中，可以如下设置神经网络的超参数。例如，对于RSN，神经批量大小(batch size)为256，学习率β_{lr}＝0.001。调度器将基于回合的数量和损失的收敛速率降低学习率，

帮助神经智能体获得高精度。当损失在10个回合后停止改善时，学习率可以降低0.5。该调度器中所有参数的学习率的下限是10^{-6}。为了避免算法收敛到潜在局部最小值或鞍点，可以将学习率重置为，例如，0.001，并帮助优化器学习更好的性能。${}^{0}best^{T}$是在T个回合后实现最低损耗的最佳参数。如果回合t的平均损失小于指定的标准$\beta_{loss}=10^{-4}$，则优化器可以提前停止。例如，设置$\beta_{epoch}=2000$并更新优化器最多2000个回合。

[0124] 对于ASN，提前停止标准的损失可以设置为10^{-5}。当损失在15个回合后停止改善时，学习率可以降低0.7。ASN中的其他超参数可以类似于RSN的那些超参数。

[0125] 图7是根据本文的实施例的小批量MCCFR算法的示例的伪代码700。小批量MCCFR算法(表示为Mini-Batch-MCCFR-NN)包括用于获取博弈的被采样信息集的反事实遗憾和平均策略分子的采样算法。与仅在一次迭代中对一个块进行采样并提供一个原始CFV的无偏估计量的传统的结果采样和外部采样不同，小批量采样技术可以在一次迭代中随机采样b个块。伪代码700中所示的小批量MCCFR算法的示例基于上述鲁棒采样。在一些实施例中，小批量MCCFR算法可以与诸如深度依赖采样方案的其他采样方案结合使用。请注意，小批量MCCFR算法是用于获取博弈的被采样信息集的反事实遗憾和平均策略分子的算法的示例。双神经CFR算法可以使用其他算法来获取博弈的被采样信息集的反事实遗憾和平均策略分子。

[0126] 令Q^{j}表示在第j次根据鲁棒采样方案采样的终点块，然后对于信息集I_i，具有b个小批量的小批量CFV可定义为：

[0127] $$\tilde{v}_i^{\sigma}(I_i|b)=\frac{1}{b}\sum_{j=1}^{b}\left(\sum_{h\in I_i,z\in Q^j,h\sqsubseteq z}\frac{\pi_{-i}^{\sigma}(z)\pi_i^{\sigma}(h,z)u_i(z)}{q(z)}\right)=\sum_{j=1}^{b}\frac{\tilde{v}_i^{\sigma}(I_i|Q^j)}{b}。\quad(11)$$

[0128] 此外，可以表明$\tilde{v}_i^{\sigma}(I_i|b)$是$I_i$的反事实值的无偏估计量：$E_{Q^j\sim\text{鲁棒采样}}[\tilde{v}_i^{\sigma}(I_i|b)]=v_i^{\sigma}(I_i)$。

[0129] 同样，动作a的累积小批量遗憾是

[0130] $$\tilde{R}_i^{T}((a|I_i)|b)=\tilde{R}_i^{T-1}((a|I_i)|b)+\tilde{v}_i^{\sigma^T}((a|I_i)|b)-\tilde{v}_i^{\sigma^T}(I_i|b)\ ,\quad(12)$$

[0131] 其中$\tilde{R}_i^{0}((a|I_i)|b)=0$。在一些实施例中，小批量MCCFR可以并行地对b个块进行采样，并且帮助MCCFR更快地收敛。

[0132] 注意，小批量MCCFR使用遗憾匹配算法来更新累积小批量遗憾$\tilde{R}^{T,+}((a|I_i)|b)$。在一些实施例中，作为小批量MCCFR的变型，可以通过下式使用小批量MCCFR+算法来更新累积小批量遗憾$\tilde{R}^{T,+}((a|I_i)|b)$直到迭代T：

[0133] $$\tilde{R}^{T,+}((a|I_i)|b)=\begin{cases}\left(\tilde{v}_i^{\sigma^T}((a|I_i)|b)-\tilde{v}_i^{\sigma^T}(I_i|b)\right)^{+} & \text{如果}T=0\\ \left(\tilde{R}_i^{T-1,+}((a|I_i)|b)+\tilde{v}_i^{\sigma^T}((a|I_i)|b)-\tilde{v}_i^{\sigma^T}(I_i|b)\right)^{+} & \text{如果}T>0\end{cases}\quad(13)$$

[0134] 其中，$(x)^{+}=\max(x,0)$。在一些实施例中，发现当指定合适的小批量大小时，小批量MCCFR+的收敛比小批量MCCFR的收敛快。伪代码700中所示的函数Mini-Batch-MCCFR-NN提出了一种小批量采样方法，其中b个块将被并行采样。这种小批量方法可以帮助MCCFR实现对CFV的更准确地估计。并行采样使该方法在实践中有效。

[0135] 如伪代码700的第1-6行所示，Mini-Batch-MCCFR-NN是输入为总迭代次数t的迭代算法。在每次迭代中，MCCFR-NN函数分别被调用用于玩家0和玩家1(如第4和5行所示)，并且该迭代中的被采样信息集的反事实遗憾和平均策略分子分别被返回并被保存在存储器M_R^t和M_S^t中。

[0136] 函数MCCFR-NN可以定义为如伪代码700的第8-33行所示的。函数MCCFR-NN像表格MCCFR一样遍历博弈树，它从根历史$h=\emptyset$开始。将I_i定义为h的信息集。假设玩家i将根据鲁棒采样对k个动作进行采样。然后函数可以被定义为如下。(1)如果历史是终点(例如$h\in Z$)，则该函数返回加权效用。(2)如果历史是机会玩家(例如$P(I_i)=-1$)，则可以根据策略$\sigma_{-i}(I_i)$对一个动作$a\in A(I_i)$进行采样。然后，此动作将被添加到历史中，即$h\leftarrow ha$。(3)如果$P(I_i)=i$，则可以通过RSN预测的累积遗憾更新当前策略。然后根据指定采样策略分布$\sigma_i^{rs(k)}$(例如，利用或不利用深度依赖采样的鲁棒采样)对k个动作进行采样。在进行递归更新后，可以获得在I_i处每个动作的反事实值和遗憾。对于被访问的节点，可以将其反事实遗憾和相应平均策略分子分别存储在M_R^t和M_S^t中。(4)如果$P(I_i)$是对手，则将根据策略$\sigma_{-i}(I_i)$仅对一个动作进行采样。

[0137] 图8是根据本文的实施例的用于执行MCCFR的采样处理800的示例的流程图。采样处理800可以是上述用于执行反事实遗憾最小化(CRF)以在两方或更多方之间的策略互动中进行策略搜索的深度依赖采样方案的示例。在一些实施例中，两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。IIG可以代表涉及两方或更多方的一个或多个现实场景，例如资源分配、产品/服务推荐、网络攻击预测和/或预防、流量交通路线规划、欺诈管理等，其中，每方可以具有关于他方的决策的不完全或不完美信息。作为示例，IIG可以表示至少涉及第一玩家和第二玩家的合作产品/服务的推荐服务。第一玩家可以是，例如，具有客户(或用户)信息、产品和服务信息、客户的购买历史等的在线零售商。第二玩家可以是，例如，具有该客户的社交网络数据的社交网络平台、具有该客户的财务信息的银行或其他金融机构、汽车经销商或在预测和向该客户推荐产品和服务时具有关于该客户的偏好、需求、财务状况、位置等客户信息的任何其他方。第一玩家和第二玩家可能各自具有不希望与他人共享的专有数据。第二玩家可以在不同时间向第一玩家仅提供部分信息。这样，第一玩家对第二玩家的信息只有有限访问权。为方便起见，处理800将被描述为数据处理装置，如，位于一个或多个位置并根据本文被适当地编程的一个或多个计算机的系统。例如，被适当地编程的图10的计算机系统1000可以执行处理800。

[0138] 在810处，数据处理装置识别第一玩家在第一玩家的第一状态下的N1个可能动作。在一些实施例中，IIG可以由博弈树(例如，博弈树102、104、202、204或206)表示。第一玩家的第一状态可以由博弈树的第一节点(例如，博弈树102中的玩家0的节点h1)表示，并且N1个可能动作可以是博弈树的第一节点的边或分支(例如，博弈树102中玩家0的节点h1的P和B边)。在合作产品/服务的推荐服务的示例中，第一玩家的第一状态包括第二玩家所提供信息的历史，并且第一玩家的N1个可能动作包括响应于第二玩家所提供信息的历史的N1个可能动作，以向客户提供产品/服务推荐。第一玩家的第一状态和可能动作可以包括由IIG建模的其他现实场景中的其他特征。

[0139] 在820处，数据处理装置以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作。在一些实施例中，数据处理装置可以从第一玩家在第一状态下的N1个可能动作中采样出k1个可能动作，其中，k1个可能动作中的每个动作都以相同的第一采样概率被采样。

[0140] 在830处，数据处理装置识别第一玩家在第一玩家的第二状态下的N2个可能动作，其中，第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态。在博弈树102的示例中，第一玩家的第二状态可以是例如h7节点，其比第一玩家的第一状态(例如，博弈树102中的玩家0的节点h1)距离博弈树102的开始状态(例如，h0节点)更远。

[0141] 在840处，数据处理装置以第二采样概率从第一玩家在第二状态下的N2个可能动作中采样出可能动作，其中，第一采样概率小于第二采样概率。在一些实施例中，数据处理装置从第一玩家在第二状态下的N2个可能动作中采样出k2个可能动作，其中，k2个可能动作中的每个动作都以相同的第二采样概率被采样。

[0142] 在850处，数据处理装置基于从第一玩家在第一状态下的N1个可能动作中采样出的可能动作和从第一玩家在第二状态下的N2个可能动作中的采样出可能动作执行CRF。在一些实施例中，可以根据相对于图3和/或图7描述的示例性技术来执行CRF。

[0143] 在一些实施例中，输出是通过求解IIG产生的第一玩家的策略。该策略可以包括由IIG建模的现实场景中第一玩家的一系列动作。例如，在合作产品/服务的推荐场景中，通过求解IIG产生的第一玩家的策略可以包括，如，响应于第二玩家提供的信息的一系列动作、基于第一玩家的信息给客户的相应产品/服务的推荐、以及由第二玩家提供的信息。通过求解IIG产生的第一玩家的输出策略可以包括由IIG建模的其他现实场景中的其他信息。

[0144] 在一些实施例中，基于从第一玩家在第一状态下的N1个可能动作中采样出的可能动作和从第一玩家在第二状态下的N2个可能动作中采样出的可能动作执行CRF，包括：计算从第一玩家在第一状态下的N1个可能动作中采样出的可能动作的遗憾值(例如，根据公式(1a)和/或公式(2))；计算从第一玩家在第二状态下的N2个可能动作中采样出的可能动作的遗憾值(例如，根据公式(1a)和/或公式(2))；基于N1个可能动作中采样出的可能动作的该遗憾值更新第一玩家在第一状态下的第一策略(例如，根据公式(3))；以及基于N2个可能动作中采样出的可能动作的该遗憾值更新第一玩家在第二状态下的第二策略(例如，根据公式(3))。

[0145] 在一些实施例中，数据处理装置基于从第一玩家在第一状态下的N1个可能动作中采样出的k1个可能动作和从第一玩家在第二状态下的N2个可能动作中采样出的k2个可能动作执行CRF。

[0146] 在一些实施例中，可以结合深度依赖采样进行鲁棒采样。例如，第一采样概率是k1/N1，第二采样概率是k2/N2。这样，根据均匀分布该可能动作被采样。

[0147] 在一些实施例中，2<＝k1<＝N1且2<＝k2<＝N2，以便玩家的每个状态都会有一个以上的可能动作被使用。

[0148] 在一些实施例中，k1＝k2，因此在第一玩家的第一状态和第二状态下选择或访问相等数量的样本。

[0149] 类似地，可以结合第二玩家进行深度依赖采样。例如，数据处理装置识别第二玩家在第二玩家的第一状态下的M1个可能动作。数据处理装置以第三采样概率从第二玩家在第

一状态下的M1个可能动作中采样出可能动作。数据处理装置识别第二玩家在第二玩家的第二状态下的M2个可能动作，其中，第二玩家的第一状态比第二玩家的第二状态更接近IIG的开始状态。数据处理装置以第四采样概率从第一玩家在第二状态下的M2个可能动作中采样出可能动作，其中，第三采样概率小于第四采样概率。

[0150] 在一些实施例中，可以结合第一玩家和第二玩家两者进行深度依赖采样。在一些实施例中，数据处理装置识别第二玩家在第二玩家的第一状态下的M1个可能动作，其中，第一玩家的第一状态(例如，博弈树102中的玩家0的状态h1)比第二玩家的第一状态(例如，博弈树102中的玩家1的状态h4)更接近IIG的开始状态(例如，h0状态)。数据处理装置以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作，其中，第三采样概率大于第一采样概率。

[0151] 图9是根据本文的实施例的用于执行MCCFR的双神经CFR算法900的示例的流程图。采样处理900可以是如上针对图4至图7描述的用于执行反事实遗憾最小化(CRF)以在两个或更多个玩家之间的策略互动中进行策略搜索的双贮存器CFR算法的示例。在一些实施例中，两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。IIG可以代表涉及两方或更多方的一个或多个现实场景，例如资源分配、产品/服务推荐、网络攻击预测和/或预防、流量交通路线规划、欺诈管理等，其中，每方可以具有关于他方的决策的不完全或不完美信息。作为示例，IIG可以表示至少涉及第一玩家和第二玩家的合作产品/服务的推荐服务。第一玩家可以是，例如，具有客户(或用户)信息、产品和服务信息、客户的购买历史等的在线零售商。第二玩家可以是，例如，具有该客户的社交网络数据的社交网络平台、具有该客户的财务信息的银行或其他金融机构、汽车经销商或在预测和向该客户推荐产品和服务时可能具有关于该客户的偏好、需求、财务状况、位置等客户信息的任何其他方。第一玩家和第二玩家可能各自具有不希望与他人共享的专有数据。第二玩家可以在不同时间向第一玩家仅提供部分信息。这样，第一玩家对第二玩家的信息只有有限访问权。为方便起见，处理900将被描述为数据处理装置，如，位于一个或多个位置并根据本文被适当地编程的一个或多个计算机的系统。例如，被适当地编程的图10的计算机系统1000可以执行处理900。

[0152] 在910处，数据处理装置初始化第一神经网络的参数和第二神经网络的参数。第一神经网络(例如，RegretSumNetwork(RSN)420)可用于预测玩家状态下的可能动作的遗憾值。在合作产品/服务的推荐服务的示例中，玩家状态包括第二玩家所提供信息的历史，并且玩家的可能动作包括响应于第二玩家所提供信息的历史的可能动作，以向客户提供产品/服务的推荐。第二神经网络(例如，AveStrategyNetwork(ASN)430)可用于预测玩家状态下的可能动作的策略值。在一些实施例中，数据处理装置根据热启动分别初始化参数，例如基于在先前迭代中或基于现有CRF算法获得的第一神经网络的参数和第二神经网络的参数分别初始化参数。在一些实施例中，数据处理装置随机地初始化第一神经网络的参数和第二神经网络的参数。

[0153] 在920处，数据处理装置在第一数据存储设备(例如，贮存器M_R)中存储多个遗憾样本，其中，多个遗憾样本各自包括玩家状态和该玩家状态下的可能动作的遗憾值。在一些实施例中，玩家状态下的可能动作的遗憾值包括基于该玩家状态下的该可能动作的反事实值计算出的该玩家状态下的该可能动作的反事实遗憾值。例如，每个遗憾样本可以包括遗憾

元组(I_i,$r\sim_i^{\sigma t}(a|I_i)$)。在一些实施例中,在该玩家与至少另一玩家之间的策略互动中进行策略搜索时,通过反事实遗憾最小化(CRF)算法的两次或更多次迭代获得多个遗憾样本。在一些实施例中,CRF算法包括鲁棒采样CRF算法。

[0154] 在一些实施例中,在求解IIG的CRF算法的两次或更多次迭代的每次迭代中,数据处理装置根据采样方案从玩家在第二状态下的多个可能动作中采样出可能动作;计算玩家在第二状态下的该可能动作的反事实值(例如,根据公式(1));基于玩家在第二状态下的该可能动作的该反事实值,计算玩家在第二状态下的该可能动作的遗憾值(例如,根据公式(1a)和/或公式(2));根据遗憾匹配算法,基于玩家在第二状态下的该可能动作的该遗憾值,计算玩家在第二状态下的该可能动作的更新策略(例如,根据公式(3));以及基于玩家在第二状态下的该可能动作的该更新策略,计算玩家在第二状态下的该可能动作的策略值(例如,根据公式(4)和/或公式(5))。

[0155] 在一些实施例中,数据处理装置可以获得新遗憾样本(例如,通过执行MCCFR的另一次迭代)。所述数据处理装置可以根据贮存器采样算法将新遗憾样本存储到第一数据存储设备中。例如,根据贮存器采样算法将新遗憾样本存储到第一数据存储设备中,包括:确定第一数据存储设备是否已满;当确定第一数据存储设备已满,用新遗憾样本替换第一数据存储设备中的多个遗憾样本中的一个。

[0156] 在930处,数据处理装置在第二数据存储设备(例如,贮存器M_S)中存储多个策略样本,其中,多个策略样本各自包括玩家状态和该玩家状态下的可能动作的策略值。在一些实施例中,玩家状态下的可能动作的策略值包括平均策略分子。例如,多个策略样本各自可以包含策略元组(I_i,$s^t(a|I_i)$)。

[0157] 在940处,例如,根据公式(7),数据处理装置基于第一数据存储设备中的多个遗憾样本更新用于预测玩家状态下的可能动作的遗憾值的第一神经网络的参数。在一些实施例中,可以根据图6所示的NeuralAgent算法或任何其他用于优化神经网络的算法更新第一神经网络的参数。

[0158] 在950处,例如,根据公式(8),数据处理装置基于第二数据存储设备中的多个策略样本更新用于预测玩家状态下的可能动作的策略值的第二神经网络的参数。在一些实施例中,可以根据图6所示的NeuralAgent算法或任何其他用于优化神经网络的算法更新第二神经网络的参数。

[0159] 在960处,数据处理装置识别玩家的第一状态以及玩家在第一状态下的第一可能动作。

[0160] 在970处,数据处理装置使用第一神经网络的参数预测玩家在第一状态下的第一可能动作的第一遗憾值。在一些实施例中,玩家在第一状态下的第一可能动作的预测的第一遗憾值可以被用于CFR算法的下一次迭代中。

[0161] 在980处,数据处理装置使用第二神经网络的参数预测玩家在第一状态下的第一可能动作的第一策略值。在一些实施例中,玩家在第一状态下的第一可能动作的预测的第一策略值可以被用于CFR算法的下一次迭代中。在一些实施例中,玩家在第一状态下的第一可能动作的预测的第一策略值可以被用于计算近似纳什均衡并作为CFR算法的输出。在一些实施例中,玩家在第一状态下的第一可能动作的预测的第一策略值可以包括在由IIG建模的现实场景中的第一玩家的一系列动作。例如,在合作产品/服务的推荐场景中,玩家在

第一状态下的第一可能动作的预测的第一策略值可以包括，例如，响应于由第二玩家提供的信息的一系列动作、基于第一玩家的信息给客户的相应产品/服务的推荐、以及由第二玩家提供的信息。玩家在第一状态下的第一可能动作的预测的第一策略值可以包括由IIG建模的其他现实场景中的其他信息。

[0162] 图10描绘了示出根据本文的实施例的计算机实现的用于提供与所描述的算法、方法、功能、处理、流程和程序相关联的计算功能的系统的示例的框图。图10是示出根据本公开的实施例的计算机实现的用于提供与所描述的算法、方法、功能、处理、流程和程序相关联的计算功能的系统1000的示例的框图。在所示的实施例中，系统1000包括计算机1002和网络1030。

[0163] 所示的计算机1002旨在包含任何计算设备，例如服务器、台式计算机、膝上型计算机/笔记本计算机、无线数据端口、智能电话、个人数据助理(PDA)、平板计算机、这些设备中的一个或多个处理器、另一计算设备或计算设备的组合，包括计算设备的物理或反事实实例、或计算设备的物理或反事实实例的组合。另外，计算机1002可以包括输入设备，例如小键盘、键盘、触摸屏，另一输入设备或可以接受用户信息的输入设备的组合，以及传达与计算机1002的操作相关联的信息的输出设备，包括图形类型的用户界面(UI)(或GUI)或其他UI上的数字数据、视觉、音频、另一种类型的信息或各种类型的信息的组合。

[0164] 计算机1002可以在分布式计算系统中充当客户端、网络组件、服务器、数据库的角色或另一持续性设备、另一角色或用于执行本公开中描述的主题的角色的组合。所示的计算机1002可通信地与网络1030耦接。在一些实施例中，计算机1002的一个或多个组件可以被配置为在包括基于云计算的环境、本地环境、全局环境、另一环境或环境组合的环境中操作。

[0165] 在高级别上，计算机1002是可操作用于接收、发送、处理、存储或管理与所描述的主题相关联的数据和信息的电子计算设备。根据一些实施例，计算机1002还可包括服务器或与服务器可通信地耦接，包括应用服务器、电子邮件服务器、网络服务器、高速缓存服务器、流数据服务器、另一服务器或服务器的组合。

[0166] 计算机1002可以通过网络1030(例如，来自另一计算机1002上执行的客户端软件应用)接收请求，并通过使用软件应用或软件应用的组合处理接收的请求来响应接收的请求。另外，还可以从内部用户(例如，从命令控制台或通过另一内部访问方法)、外部或第三方或者其他实体、个人、系统或计算机向计算机1002发送请求。

[0167] 计算机1002的每个组件可以使用系统总线1003进行通信。在一些实施例中，计算机1002的任何或所有组件，包括硬件、软件或者硬件和软件的组合，可以使用应用编程接口(API)1012、服务层1013、或者API 1012和服务层1013的组合通过系统总线1003进行接口连接。API 1012可以包括用于例程、数据结构和对象类的规范。API 1012可以是独立于计算机语言的或依赖于计算机语言的，并且是指完整的接口、单个函数或甚至一组API。服务层1013向计算机1002或可通信地耦接到计算机1002的其他组件(无论是否示出)提供软件服务。使用服务层1013的所有服务客户可访问计算机1002的功能诸如由服务层1013提供的软件服务通过定义的接口提供可重用的、定义的功能。例如，该接口可以是用以JAVA、C++、另一种计算语言、或以可扩展标记语言(XML)格式、另一种格式或多种格式的组合提供数据的计算机语言的组合编写的软件。虽然示出为计算机1002的集成组件，但是替代实施例可以

将API 1012或服务层1013示出为与计算机1002的其他组件有关的独立组件或可通信地耦接到计算机1002的其他组件(无论是否示出)。此外,在不脱离本公开的范围的情况下,API 1012或服务层1013的任何或所有部分可以被实现为另一软件模块、企业应用或硬件模块的子模块(a child or a sub-module)。

[0168] 计算机1002包括接口1004。尽管示出为单个接口1004,但是可以根据计算机1002的特定需要、期望或特定实施例使用两个或更多个接口1004。接口1004被计算机1002用来与在分布式环境中通信地链接到网络1030的另一计算系统(无论是否示出)进行通信。通常,接口1004可操作地与网络1030通信,并且包括以软件、硬件或者软件和硬件的组合编码的逻辑。更具体地说,接口1004可以包括支持与通信相关联的一个或多个通信协议的软件,以使得网络1030或接口1004的硬件可操作地在所示计算机1002之内和所示计算机1002之外通信物理信号。

[0169] 计算机1002包括处理器1005。尽管示出为单个处理器1005,但是可以根据计算机1002的特定需要、期望或特定实施例使用两个或更多个处理器1005。通常,处理器1005执行指令并操纵数据以执行计算机1002的操作以及本公开中所描述的任何算法、方法、功能、处理、流程和程序。

[0170] 计算机1002还包括数据库1006,该数据库1006可以保存用于计算机1002的数据、通信地链接到网络1030的另一组件(无论是否示出)的数据、或者计算机1002和另一组件的组合的数据。例如,数据库1006可以是存储与本公开一致的存储数据的内存、常规数据库或另一类型的数据库。在一些实施例中,根据计算机1002的特定需要、期望或特定实施例以及所描述的功能,数据库1006可以是两个或更多个不同数据库类型的组合(例如,混合内存和常规数据库)。尽管被示为单个数据库1006,但是可以根据计算机1002的特定需求、期望或特定实施例以及所描述的功能来使用相似或不同类型的两个或更多个数据库。尽管数据库1006被示为计算机1002的集成组件,但在替代实施例中,数据库1006可以在计算机1002的外部。作为示例,数据库1006可以包括上述存储遗憾样本1026的贮存器M_R 1016和存储策略样本1028的存储库M_S 1018。

[0171] 计算机1002还包括存储器1007,该存储器1007可以保存用于计算机1002的数据、通信地链接到网络1030的另一组件(无论是否示出)的数据、或者计算机1002和另一组件的组合的数据。存储器1007可以存储与本公开一致的任何数据。在一些实施例中,根据计算机1002的特定需要、期望或特定实施例以及所描述的功能,存储器1007可以是两种或更多种不同类型的存储器的组合(例如,半导体和磁存储设备的组合)。尽管被示为单个存储器1007,但是可以根据计算机1002的特定需求、期望或特定实施例以及所描述的功能来使用相似或不同类型的两个或更多个存储器1007。尽管存储器1007被示为计算机1002的集成组件,但在替代实施例中,存储器1007可以在计算机1002的外部。

[0172] 应用1008是算法软件引擎,其提供根据计算机1002的特定需要、期望或特定实施例的功能,特别是关于本公开中描述的功能。例如,应用1008可以用作一个或多个组件、模块或应用。此外,尽管被示为单个应用1008,但是应用1008可以被实现为计算机1002上的多个应用1008。另外,尽管被示为与计算机1002集成,但是在替代实施例中,应用1008可以在计算机1002的外部。

[0173] 计算机1002还可以包括电源1014。电源1014可包括可被配置为用户可更换的可充

电电池或用户不可更换的不可充电电池。在一些实施例中，电源1014可以包括功率转换或管理电路(包括充电、备用或另一电源管理功能)。在一些实施例中，电源1014可以包括电源插头，以允许将计算机1002插入壁式插座或另一电源中，从而例如为计算机1002供电或为可充电电池充电。

[0174] 可以存在与包含计算机1002的计算机系统关联或在其外部的任何数量的计算机1002，每个计算机1002通过网络1030进行通信。此外，在不脱离本公开的范围的情况下，术语“客户端”、“用户”或其他适当的术语可以适当地互换使用。此外，本公开预期许多用户可以使用一个计算机1002，或者一个用户可以使用多个计算机1002。

[0175] 图11是根据本文的实施例的装置1100的模块的示例的图。装置1100可以是数据处理装置的示例性实施例，该数据处理装置用于执行反事实遗憾最小化(CRF)以在两个或更多个玩家之间的策略互动中进行策略搜索。在一些实施例中，两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。作为示例，IIG表示至少涉及第一玩家和第二玩家的合作产品服务推荐服务，第一玩家对第二玩家的信息具有有限访问权。装置1100可以对应于上述实施例，并且装置1100包括：第一识别模块1101，用于识别第一玩家在第一玩家的第一状态下的N1个可能动作；第一采样模块1102，用于以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作；第二识别模块1103，用于识别第一玩家在第一玩家的第二状态下的N2个可能动作，其中，第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态；第二采样模块1104，以第二采样概率从所述第一玩家在第二状态下的N2个可能动作中采样出可能动作，其中，所述第一采样概率小于所述第二采样概率；处理模块1105，用于基于第一玩家在第一状态下的N1个可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中的可能动作执行CRF。在一些实施例中，第一玩家的第一状态包括第二玩家所提供信息的历史，并且第一玩家的N1个可能动作包括响应于第二玩家所提供信息的历史的N1个可能动作，以向客户提供产品服务推荐。

[0176] 在可选实施例中，处理模块包括：第一计算模块，用于计算第一玩家在第一状态下的N1个可能动作中的可能动作的遗憾值；第二计算模块，用于计算所述第一玩家在第二状态下的所述N2个可能动作中的可能动作的遗憾值；第一更新模块，用于基于所述N1个可能动作中的可能动作的遗憾值更新所述第一玩家在所述第一状态下的第一策略；第二更新模块，用于基于所述N2个可能动作中的可能动作的遗憾值更新第一玩家在第二状态下的第二策略。

[0177] 在可选实施例中，第一采样模块从第一玩家在第一状态下的N1个可能动作中采样k1个可能动作，其中，k1个可能动作中的每个可能动作都以相同的第一采样概率被采样。第二采样模块从第一玩家在第二状态下的N2个可能动作中采样出k2个可能动作，其中，k2个可能动作中的每个可能动作以相同的第二采样概率被采样。

[0178] 在可选实施例中，处理模块基于第一玩家在第一状态下的N1个可能动作中的k1个可能动作和第一玩家在第二状态下的N2个可能动作中的k2个可能动作执行CRF。

[0179] 在可选实施例中，第一采样概率为k1/N1，第二采样概率为k2/N2。

[0180] 在一个可选的实施例中，2<＝k1<＝N1和2<＝k2<＝N2。

[0181] 在可选的实施例中，k1＝k2。

[0182] 在可选实施例中，装置1100还包括：第三识别模块，用于识别第二玩家在第二玩家的第一状态下的M1个可能动作；第三采样模块，用于以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作；第四识别模块，用于识别第二玩家在第二玩家的第二状态下的M2个可能动作，其中，第二玩家的第一状态比第二玩家的第二状态更接近IIG的开始状态；第四采样模块，用于以第四采样概率从第一玩家在第二状态下的M2个可能动作中采样出可能动作，其中，第三采样概率小于第四采样概率。

[0183] 在可选实施例中，装置1100还包括：第五识别模块，用于识别第二玩家在第二玩家的第一状态下的M1个可能动作，其中，第一玩家的第一状态比第二玩家的第一状态更接近IIG的开始状态；第五采样模块，用于以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作，其中，第三采样概率大于第一采样概率。

[0184] 在先前实施例中示出的系统、装置、模块或单元可以通过使用计算机芯片或实体来实现，或者可以通过使用具有特定功能的产品来实现。典型的实施设备是计算机，计算机可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或这些设备的任意组合。

[0185] 对于装置中每个模块的功能和角色的实施过程，可以参考前一方法中相应步骤的实施过程。为简单起见，这里省略了细节。

[0186] 由于装置实施基本上对应于方法实施，对于相关部分，可以参考方法实施中的相关描述。先前描述的装置实施仅是示例。被描述为单独部分的模块可以是或不是物理上分离的，并且显示为模块的部分可以是或不是物理模块，可以位于一个位置，或者可以分布在多个网络模块上。可以基于实际需求来选择一些或所有模块，以实现本文方案的目标。本领域普通技术人员无需付出创造性劳动就能理解和实现本申请的实施方式。

[0187] 再次参考图11，可以将其解释为示出了内部功能模块和数据处理装置的结构，该数据处理装置用于执行反事实遗憾最小化(CRF)以在两方或更多方之间的策略互动中进行策略搜索。在一些实施例中，两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。执行主体本质上可以是电子设备，并且所述电子设备包括：一个或多个处理器；被配置为存储一个或多个处理器的可执行指令的存储器。

[0188] 所述一个或多个处理器被配置为识别第一玩家在第一玩家的第一状态下的N1个可能动作；以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作；识别第一玩家在第一玩家的第二状态下的N2个可能动作，其中，第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态；以第二采样概率从第一玩家在第二状态下的N2个可能动作中采样出可能动作，其中，第一采样概率小于第二采样概率；并基于第一玩家在第一状态下的N1个可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中的可能动作执行CRF。

[0189] 可选地，所述一个或多个处理器被配置用于计算所述第一玩家在第一状态下的N1个可能动作中的可能动作的遗憾值；计算第一玩家在第二状态下的N2个可能动作中的可能动作的遗憾值；基于N1个可能动作中的可能动作的遗憾值更新第一玩家在第一状态下的第一策略；并基于N2个可能动作中的可能动作的遗憾值更新第一玩家在第二状态下的第二策略。

[0190] 可选地,所述一个或多个处理器被配置用于从第一玩家在第一状态下的N1个可能动作中采样出k1个可能动作,其中,k1个可能动作中的每个可能动作以相同的第一采样概率被采样;从第一玩家在第二状态下的N2个可能动作中采样出k2个可能动作,其中,k2个可能动作中的每个可能动作以相同的第二采样概率被采样。

[0191] 可选地,一个或多个处理器被配置为基于第一玩家在第一状态下的N1个可能动作中的k1个可能动作和第一玩家在第二状态下的N2个可能动作中的k2个动作执行CRF。

[0192] 可选地,第一采样概率是k1/N1,第二采样概率是k2/N2。

[0193] 可选地,2<=k1<=N1和2<=k2<=N2。

[0194] 可选地,k1=k2。

[0195] 可选地,一个或多个处理器被配置为识别第二玩家在第二玩家的第一状态下的M1个可能动作;以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作;识别第二玩家在第二玩家的第二状态下的M2个可能动作,其中,第二玩家的第一状态比第二玩家的第二状态更接近IIG的开始状态;并以第四采样概率从第一玩家在第二状态下的M2个可能动作中采样出可能动作,其中,第三采样概率小于第四采样概率。

[0196] 可选地,一个或多个处理器被配置为识别第二玩家在第二玩家的第一状态下的M1个可能动作,其中,第一玩家的第一状态比第二个玩家的第一状态更接近IIG的开始状态;并以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作,其中,第三采样概率大于第一采样概率。

[0197] 图12是根据本文的实施例的另一装置1200的模块的示例的图。装置1200可以是数据处理装置的示例性实施例,该数据处理装置用于执行反事实遗憾最小化(CRF)以在两个或更多个玩家之间的策略互动中进行策略搜索。装置1200可以对应于上述实施例,并且装置1200包括以下:第一存储模块1201,用于将多个遗憾样本存储在第一数据存储设备中,其中,多个遗憾样本各自包括玩家状态和玩家状态下的可能动作的遗憾值,其中,多个遗憾样本是在玩家与至少另一玩家之间的策略互动中进行策略搜索时,通过反事实遗憾最小化(CRF)算法的两次或更多次迭代获得的;第二存储模块1202,用于将多个策略样本存储在第二数据存储设备中,其中,多个策略样本各自包括玩家状态和玩家状态下的可能动作的策略值;第一更新模块1203,用于基于第一数据存储设备中的多个遗憾样本,更新用于预测玩家状态下的可能动作的遗憾值的第一神经网络的参数;第二更新模块1204,用于基于第二数据存储设备中的多个策略样本更新用于预测玩家状态下的可能动作的策略值的第二神经网络的参数。在一些实施例中,两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。作为示例,IIG表示至少涉及玩家和第二玩家的合作产品服务推荐服务,所述玩家对第二玩家的信息具有有限的访问权,其中,玩家状态包括第二玩家所提供信息的历史,其中,玩家的可能动作包括响应于第二玩家所提供信息的历史的可能动作以向顾客提供产品服务推荐。

[0198] 前述和其他描述的实施例可以各自可选地包括以下特征中的一个或多个:

[0199] 在可选的实施例中,装置1200还包括:识别模块,用于识别玩家的第一状态和玩家在第一状态下的第一可能动作;第一预测模块,用于使用第一神经网络的参数预测玩家在第一状态下的第一可能动作的第一遗憾值;第二预测模块,用于使用第二神经网络的参数预测玩家在第一状态下的第一可能动作的第一策略值。

[0200] 在可选的实施例中，其中，所述第一存储模块能够获取新遗憾样本；并且根据贮存器采样算法将所述新遗憾样本存储到第一数据存储设备中。

[0201] 在可选的实施例中，其中，根据贮存器采样算法将新遗憾样本存储到第一数据存储设备中，包括：确定第一数据存储设备是否已满；并响应于确定第一数据存储设备已满，用新遗憾样本替换第一数据存储设备中的多个遗憾样本中的一个。

[0202] 在可选的实施例中，其中，CRF算法包括鲁棒采样CRF算法。

[0203] 在可选的实施例中，其中，玩家状态下的可能动作的策略值包括平均策略分子。

[0204] 在可选的实施例中，其中，玩家状态下的可能动作的遗憾值包括基于玩家状态下的可能动作的反事实值计算出的玩家状态下的可能动作的反事实遗憾值。

[0205] 在可选的实施例中，装置1200还包括：在玩家与至少另一玩家之间的策略互动中进行策略搜索时，在反事实遗憾最小化(CRF)算法的两次或更多次迭代的每次迭代中，采样模块，用于根据采样方案从玩家在第二状态下的多个可能动作中采样出可能动作；第一计算模块，用于计算玩家在第二状态下的可能动作的反事实值；第二计算模块，用于基于玩家在第二状态下的可能动作的反事实值计算玩家在第二状态下的可能动作的遗憾值；第三计算模块，用于根据遗憾匹配算法基于玩家在第二状态下的可能动作的遗憾值计算玩家在第二状态下的可能动作的更新策略；第四计算模块，用于基于玩家在第二状态下的可能动作的更新策略计算玩家在第二状态下的可能动作的策略值。

[0206] 在可选的实施例中，装置1200还包括：第一初始化模块，用于基于先前迭代中的第一神经网络的参数对第一神经网络的参数进行初始化；第二初始化模块，用于基于先前迭代中的第二神经网络的参数对第二神经网络的参数进行初始化。

[0207] 在先前实施例中示出的系统、装置、模块可以通过使用计算机芯片或实体来实现，或者可以通过使用具有特定功能的产品来实现。典型的实施设备是计算机，计算机可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或这些设备的任意组合。

[0208] 对于装置中每个模块的功能和角色的实施过程，可以参考前一方法中相应步骤的实施过程。为简单起见，这里省略了细节。

[0209] 由于装置实施基本上对应于方法实施，对于相关部分，可以参考方法实施中的相关描述。先前描述的装置实施仅是示例。被描述为单独部分的模块可以是或不是物理上分离的，并且显示为模块的部分可以是或不是物理模块，可以位于一个位置，或者可以分布在多个网络模块上。可以基于实际需求来选择一些或所有模块，以实现本文方案的目标。本领域普通技术人员无需付出创造性劳动就能理解和实现本申请的实施方式。

[0210] 再次参考图12，可以将其解释为示出了内部功能模块和数据处理装置的结构，该数据处理装置用于执行反事实遗憾最小化(CRF)以在两方或更多方之间的策略互动中进行策略搜索。本质上，执行主体可以是电子设备，电子设备包括：一个或多个处理器；被配置为存储一个或多个处理器的可执行指令的存储器。

[0211] 所述一个或多个处理器被配置为将多个遗憾样本存储在第一数据存储设备中，其中，多个遗憾样本各自包括玩家状态和玩家状态下的可能动作的遗憾值，其中，多个遗憾样本是在玩家与至少一个其他玩家之间的策略互动中进行策略搜索时，通过反事实遗憾最小化(CRF)算法的两次或更多次迭代获得的；将多个策略样本存储在第二数据存储设备中，其

中,所述多个策略样本各自包括玩家状态和玩家状态下的可能动作的策略值;基于第一数据存储设备中的多个遗憾样本更新用于预测玩家状态下的可能动作的遗憾值的第一神经网络的参数;基于第二数据存储设备中的多个策略样本更新用于预测玩家状态下的可能动作的策略值的第二神经网络的参数。在一些实施例中,两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。作为示例,IIG表示至少涉及玩家和第二玩家的合作产品服务推荐服务,所述玩家对第二玩家的信息具有有限的访问权,其中,玩家状态包括第二玩家所提供信息的历史,其中,玩家的可能动作包括响应于第二玩家所提供信息的历史的可能动作以向顾客提供产品服务推荐。

[0212] 可选地,所述一个或多个处理器被配置为识别玩家的第一状态以及玩家在第一状态下的第一可能动作;使用第一神经网络的参数预测玩家在第一状态下的第一可能动作的第一遗憾值;并使用第二神经网络的参数预测玩家在第一状态下的第一可能动作的第一策略值。

[0213] 可选地,所述一个或多个处理器被配置为:获得新遗憾样本;以及根据贮存器采样算法将新遗憾样本存储到第一数据存储设备中。

[0214] 可选地,一个或多个处理器被配置为:确定第一数据存储器是否已满;并响应于确定第一数据存储设备已满,用新遗憾样本替换第一数据存储设备中的多个遗憾样本中的一个。

[0215] 可选地,CRF算法包括鲁棒采样CRF算法。

[0216] 可选地,玩家状态下的可能动作的策略值包括平均策略分子。

[0217] 可选地,玩家状态下的可能动作的遗憾值包括基于玩家状态下的可能动作的反事实值计算出的玩家状态下的可能动作的反事实遗憾值。

[0218] 可选地,所述一个或多个处理器被配置为:在玩家与至少另一玩家之间的策略互动中进行策略搜索时,在反事实遗憾最小化(CRF)算法的两次或更多次迭代的每次迭代中,根据采样方案从玩家在第二状态下的多个可能动作中采样出可能动作;计算玩家在第二状态下的可能动作的反事实值;基于玩家在第二状态下的可能动作的反事实值计算玩家在第二状态下的可能动作的遗憾值;根据遗憾匹配算法基于玩家在第二状态下的可能动作的遗憾值计算玩家在第二状态下的可能动作的更新策略;并基于玩家在第二状态下的可能动作的更新策略计算玩家在第二状态下的可能动作的策略值。

[0219] 可选地,所述一个或多个处理器被配置为基于先前迭代中的第一神经网络的参数对第一神经网络的参数进行初始化;并基于先前迭代中的第二神经网络的参数对第二神经网络的参数进行初始化。

[0220] 所描述的主题的实施例可以包括单独或组合的一个或多个特征。例如,在第一实施例中,一种计算机实现的用于执行反事实遗憾最小化(CRF)以在两方或更多方之间的策略互动中进行策略搜索的方法。所述方法包括识别第一玩家在第一玩家的第一状态下的N1个可能动作;以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作;识别第一玩家在第一玩家的第二状态下的N2个可能动作,其中,第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态;以第二采样概率从第一玩家在第二状态下的N2个可能动作中采样出可能动作,其中,第一采样概率小于第二采样概率;并基于第一玩家在第一状态下的N1个可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中

的可能动作执行CRF。在一些实施例中，两个或更多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。作为示例，IIG表示指示涉及第一玩家和第二玩家的合作产品服务推荐服务，所述第一玩家对第二玩家的信息具有有限的访问权，其中，第一玩家状态包括由第二玩家提供的信息的历史，其中，第一玩家的N1个可能动作包括响应于由第二玩家提供的信息的历史的N1个可能动作以向顾客提供产品服务推荐。

[0221] 前述和其他描述的实施例可以各自可选地包括以下特征中的一个或多个：

[0222] 第一特性，可与以下任何特性组合，其中，基于第一玩家在第一状态下的N1可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中的可能动作执行CRF包括：计算所述第一玩家在第一状态下的N1个可能动作中的可能动作的遗憾值；计算第一玩家在第二状态下的N2个可能动作中的可能动作的遗憾值；基于N1个可能动作中的可能动作的遗憾值，更新第一玩家在第一状态下的第一策略；并基于N2个可能动作中的可能动作的遗憾值更新第一玩家在第二状态下的第二策略。

[0223] 第二特征，可与以下任何特征组合，进一步包括：从第一玩家在第一状态下的N1个可能动作中采样出k1个可能动作，其中，k1个可能动作中的每个可能动作以相同的第一采样概率被采样；从第一玩家在第二状态下的N2个可能动作中采样出k2个可能动作，其中，k2个可能动作中的每个可能动作以相同的第二采样概率被采样。

[0224] 第三特征，可与以下任何特征组合，进一步包括：基于第一玩家在第一状态下的N1个可能动作中的k1个可能动作和在第一玩家在第二状态下的N2个可能动作中的k2个可能动作执行CRF。

[0225] 第四特征，可与以下任何特征组合，其中，第一采样概率是k1/N1，第二采样概率是k2/N2。

[0226] 第五特征，可与以下任何特征组合，其中，2<＝k1<＝N1并且2<＝k2<＝N2。

[0227] 第六特征，可与以下任何特征组合，其中，k1＝k2。

[0228] 第七特征，可与以下任何特征组合，进一步包括：识别第二玩家在第二玩家的第一状态下的M1个可能动作；以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作；识别第二玩家在第二玩家的第二状态下的M2个可能动作，其中，第二玩家的第一状态比第二玩家的第二状态更接近IIG的开始状态；并以第四采样概率从第一玩家在第二状态下的M2个可能动作中采样出可能动作，其中，第三采样概率小于第四采样概率。

[0229] 第八特征，可与以下任何特征组合，进一步包括：识别第二玩家在第二玩家的第一状态下的M1个可能动作，其中，第一玩家的第一状态比第二个玩家的第一状态更接近IIG的开始状态；并以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作，其中，第三采样概率大于第一采样概率。

[0230] 在第二实施例中，一种系统，包括：一个或多个处理器；一个或多个计算机可读存储器，其耦接到所述一个或多个处理器并其上存储有指令，所述指令可由所述一个或多个处理器执行以执行第一实施例中的任一方法及其如上所述一个或多个特征的可选组合。

[0231] 在第三实施例中，一种用于执行反事实遗憾最小化CFR以在两方或更多方之间的策略互动中进行策略搜索的装置。所述装置包括：第一识别模块，用于识别第一玩家在第一玩家的第一状态下的N1个可能动作；第一采样模块，用于以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作；第二识别模块，用于识别第一玩家在第一玩

家的第二状态下的N2个可能动作，其中，第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态；第二采样模块，以第二采样概率从所述第一玩家在第二状态下的N2个可能动作中采样出可能动作，其中，所述第一采样概率小于所述第二采样概率；处理模块，用于基于第一玩家在第一状态下的N1个可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中的可能动作执行CRF。在一些实施例中，两个或多个玩家之间的策略互动可以通过涉及两个或更多个玩家的不完美信息博弈(IIG)来建模。作为示例，IIG表示至少涉及第一玩家和第二玩家的合作产品服务推荐服务，所述第一玩家对第二玩家的信息具有有限的访问权，其中，第一玩家的第一状态包括由第二玩家提供的信息的历史，其中第一玩家的N1个可能动作包括响应于由第二玩家提供的信息的历史的N1个可能动作以向顾客提供产品服务推荐。

[0232] 前述和其他描述的实施例可以各自可选地包括以下特征中的一个或多个：

[0233] 第一特性，可与以下任何特性组合，其中，所述处理模块包括：第一计算模块，用于计算第一玩家在第一状态下的N1个可能动作中的可能动作的遗憾值；第二计算模块，用于计算所述第一玩家在第二状态下的所述N2个可能动作中的可能动作的遗憾值；第一更新模块，用于基于所述N1个可能动作中的可能动作的遗憾值更新所述第一玩家在所述第一状态下的第一策略；第二更新模块，用于基于所述N2个可能动作中的可能动作的遗憾值更新第一玩家在第二状态下的第二策略。

[0234] 第二特征，可与以下任何特征组合，其中，第一采样模块从第一玩家在第一状态下的N1个可能动作中采样出k1个可能动作，其中，k1个可能动作中的每个可能动作以相同的第一采样概率被采样；以及第二采样模块从第一玩家在第二状态下的N2个可能动作中采样出k2个可能动作，其中，k2个可能动作中的每个可能动作以相同的第二采样概率被采样。

[0235] 第三特征，可与以下任何特征组合，其中，所述处理模块基于第一玩家在第一状态下的N1个可能动作中的k1个可能动作和第一玩家在第二状态下的N2个可能动作中的k2个可能动作执行CRF。

[0236] 第四特征，可与以下任何特征组合，其中，第一采样概率是k1/N1，第二采样概率是k2/N2。

[0237] 第五特征，可与以下任何特征组合，其中，2<＝k1<＝N1并且2<＝k2<＝N2。

[0238] 第六特征，可与以下任何特征组合，其中，k1＝k2。

[0239] 第七特性，可与以下任何特性组合，进一步包括：第三识别模块，用于识别第二玩家在第二玩家的第一状态下的M1个可能动作；第三采样模块，用于以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作；第四识别模块，用于识别第二玩家在第二玩家的第二状态下的M2个可能动作，其中，第二玩家的第一状态比第二玩家的第二状态更接近IIG的开始状态；第四采样模块，用于以第四采样概率从第一玩家在第二状态下的M2个可能动作中采样出可能动作，其中，第三采样概率小于第四采样概率。

[0240] 第八特征，可与以下任何特征组合，进一步包括：第五识别模块，用于识别第二玩家在第二玩家的第一状态下的M1个可能动作，其中，第一玩家的第一状态比第二个玩家的第一状态更接近IIG的开始状态；和第五采样模块，用于以第三采样概率从第二玩家在第一状态下的M1个可能动作中采样出可能动作，其中，第三采样概率大于第一采样概率。

[0241] 本文中描述的主题、动作和操作的实施例可以在数字电子电路、有形体现的计算

机软件或固件、计算机硬件中实现，包括本文中公开的结构及其结构等同物，或者它们中的一个或多个的组合。本文中描述的主题的实施可以实现为一个或多个计算机程序，例如，编码在计算机程序载体上的一个或多个计算机程序指令模块，用于由数据处理执行或控制数据处理装置的操作。例如，计算机程序载体可以包括一个或多个计算机可读存储介质，其具有编码或存储在其上的指令。载体可以是有形的非暂态计算机可读介质，例如磁盘、磁光盘或光盘、固态驱动器、随机存取存储器(RAM)、只读存储器(ROM)或其他介质类型。替代地或另外地，载体可以是人工生成的传播信号，例如，机器生成的电、光或电磁信号，其被生成以编码信息以便传输到合适的接收器装置以供数据处理装置执行。计算机存储介质可以是或部分是机器可读存储设备、机器可读存储基板、随机或串行存取存储器设备或它们中的一个或多个的组合。计算机存储介质不是传播信号。

[0242] 计算机程序，也可以被称为或描述为程序、软件、软件应用、app、模块、软件模块、引擎、脚本或代码，可以以任何形式的编程语言编写，包括编译或解释性语言、说明或程序性语言；它可以配置为任何形式，包括作为独立程序，或者作为模块、组件、引擎、子程序或适合在计算环境中执行的其他单元，该环境可包括由通信数据网络互联的在一个或多个位置上的一个或多个计算机。

[0243] 计算机程序可以但非必须对应于文件系统中的文件。计算机程序可以存储在：文件保存其他程序或数据的文件的一部分中，例如，存储在标记语言文档中的一个或多个脚本；专用于所讨论的程序的单个文件；或者多个协调文件，例如，存储一个或多个模块、子程序或代码部分的多个文件。

[0244] 举例来说，用于执行计算机程序的处理器包括通用和专用微处理器，以及任何类型的数字计算机的任何一个或多个处理器。通常，处理器将从耦接到处理器的非暂态计算机可读介质接收用于执行的计算机程序的指令以及数据。

[0245] 术语“数据处理装置”包括用于处理数据的所有类型的装置、设备和机器，包括例如可编程处理器、计算机或者多处理器或计算机。数据处理装置可以包括专用逻辑电路，例如FPGA(现场可编程门阵列)、ASIC(专用集成电路)或GPU(图形处理单元)。除了硬件，该装置还可以包括为计算机程序创建执行环境的代码，例如，构成处理器固件、协议栈、数据库管理系统、操作系统或者它们中的一个或多个的组合的代码。

[0246] 本文中描述的处理和逻辑流程可以由执行一个或多个计算机程序的一个或多个计算机或处理器执行，以通过对输入数据进行运算并生成输出来执行操作。处理和逻辑流程也可以由例如FPGA、ASIC或GPU的专用逻辑电路或专用逻辑电路与一个或多个编程计算机的组合来执行。

[0247] 适合于执行计算机程序的计算机可以基于通用和/或专用微处理器，或任何其他种类的中央处理单元。通常，中央处理单元将从只读存储器和/或随机存取存储器接收指令和数据。计算机的元件可包括用于执行指令的中央处理单元以及用于存储指令和数据的一个或多个存储器设备。中央处理单元和存储器可以补充有专用逻辑电路或集成在专用逻辑电路中。

[0248] 通常，计算机还将包括或可操作地耦接至一个或多个大容量存储设备，以从一个或多个存储设备接收数据或将数据传输到一个或多个大容量存储设备。存储设备可以是，例如，磁盘、磁光或光盘，固态驱动器或任何其他类型的非暂态计算机可读介质。但是，计算

机不需要具有这样的设备。因此，计算机可以耦接到例如本地和/或远程的一个或多个存储器的一个或多个存储器设备。例如，计算机可以包括作为计算机的整体部件的一个或多个本地存储器，或者计算机可以耦接到云网络中的一个或多个远程存储器。此外，计算机可以嵌入到另一个设备中，例如移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或例如通用串行总线(USB)闪存驱动器的便携式存储设备，仅举几例。

[0249] 组件可以通过例如直接地或经由一个或多个中间组件彼此电连接或光连接而可交换地彼此“耦接”。如果其中一个组件集成到另一个组件中，则组件也可以彼此“耦接”。例如，集成到处理器中的存储组件(例如，L2高速缓存组件)被“耦接到”处理器。

[0250] 为了提供与用户的交互，本文中描述的主题的实施例可以在计算机上实现或配置为与该计算机通信，该计算机具有：显示设备(例如，LCD(液晶显示器)监视器)，用于向用户显示信息；以及输入设备，用户可以通过该输入设备向该计算机提供输入，例如键盘和例如鼠标、轨迹球或触摸板等的指针设备。其他类型的设备也可用于提供与用户的交互；例如，提供给用户的反馈可以是任何形式的感觉反馈，例如视觉反馈、听觉反馈或触觉反馈；并且可以接收来自用户的任何形式的输入，包括声音、语音或触觉输入。另外，计算机可以通过向用户使用的设备发送文档和从用户使用的设备接收文档来与用户交互；例如，通过响应于从用户设备上的web浏览器接收的请求将网页发送到用户设备上的web浏览器，或者通过与在例如智能电话或电子平板电脑等的用户设备上运行的应用(app)交互。此外，计算机可以通过向个人设备(例如，运行消息应用的智能手机)轮流发送文本消息或其他形式的消息来并接收来自用户的响应消息来与用户交互。

[0251] 本文使用与系统、装置和计算机程序组件有关的术语“配置为”。对于被配置为执行特定操作或动作的一个或多个计算机的系统，意味着系统已经在其上安装了在运行中促使该系统执行所述操作或动作的软件、固件、硬件或它们的组合。对于被配置为执行特定操作或动作的一个或多个计算机程序，意味着一个或多个程序包括当被数据处理装置执行时促使该装置执行所述操作或动作的指令。对于被配置为执行特定操作或动作的专用逻辑电路，意味着该电路具有执行所述操作或动作的电子逻辑。

[0252] 尽管本文包含许多具体实施细节，但这些不应被解释为由权利要求本身限定的对要求保护的范围的限制，而是作为对特定实施例的具体特征的描述。在本文单独实施例的上下文中描述的某些特征也可以在单个实施例中组合实现。相反，在单个实施例的上下文中描述的各种特征也可以单独地或以任何合适的子组合在多个实施例中实现。此外，尽管上面的特征可以描述为以某些组合起作用并且甚至最初如此要求保护，但是在一些情况下，可以从要求保护的组合中删除来自该组合的一个或多个特征，并且可以要求保护指向子组合或子组合的变体。

[0253] 类似地，虽然以特定顺序在附图中描绘了操作并且在权利要求中叙述了操作，但是这不应该被理解为：为了达到期望的效果，要求以所示的特定顺序或依次执行这些操作，或者要求执行所有示出的操作。在某些情况下，多任务和并行处理可能是有利的。此外，上述实施例中的各种系统模块和组件的划分不应被理解为所有实施例中都要求如此划分，而应当理解，所描述的程序组件和系统通常可以一起集成在单个软件产品中或打包成多个软件产品。

[0254] 已经描述了主题的特定实施例。其他实施例在以下权利要求的范围内。例如，权利要求中记载的动作可以以不同的顺序执行并且仍然实现期望的结果。作为一个示例，附图中描绘的处理无需要求所示的特定顺序或次序来实现期望的结果。在某些情况下，多任务和并行处理可能是有利的。

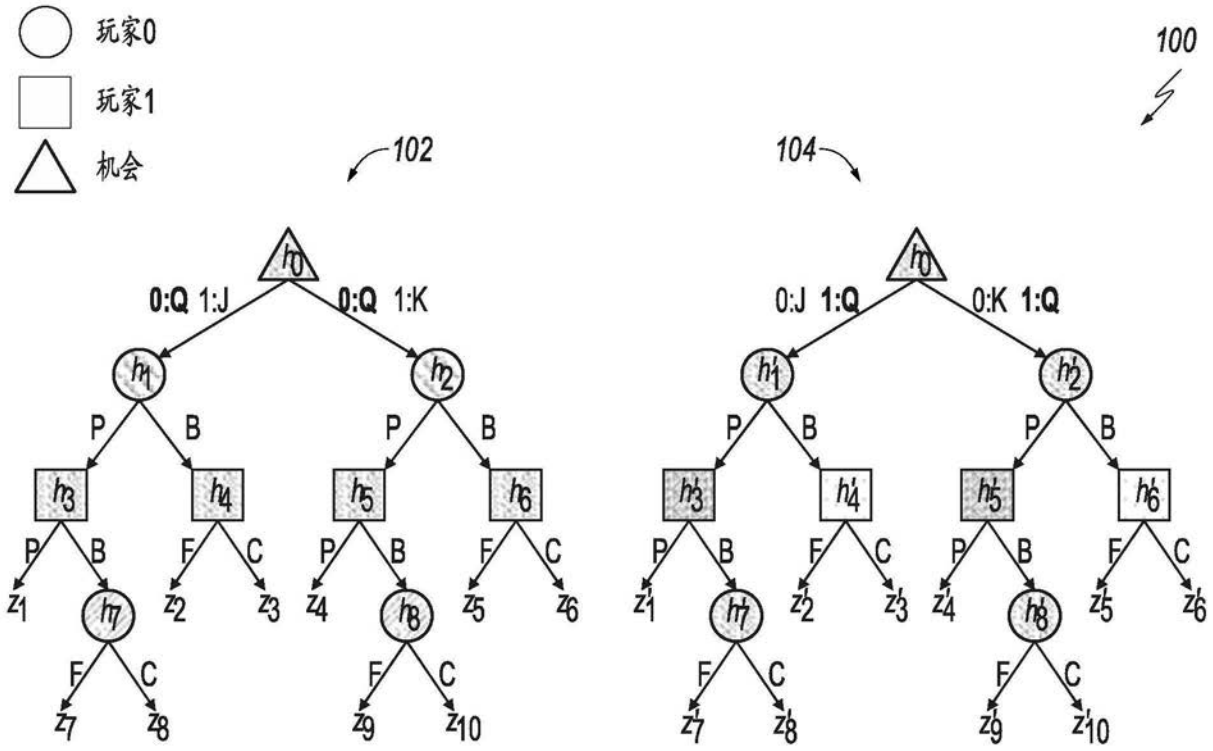

玩家0
玩家1
机会
100
102
104
0:Q 1:J
0:Q 1:K
0:J 1:Q
0:K 1:Q
P
B
F
C

图1

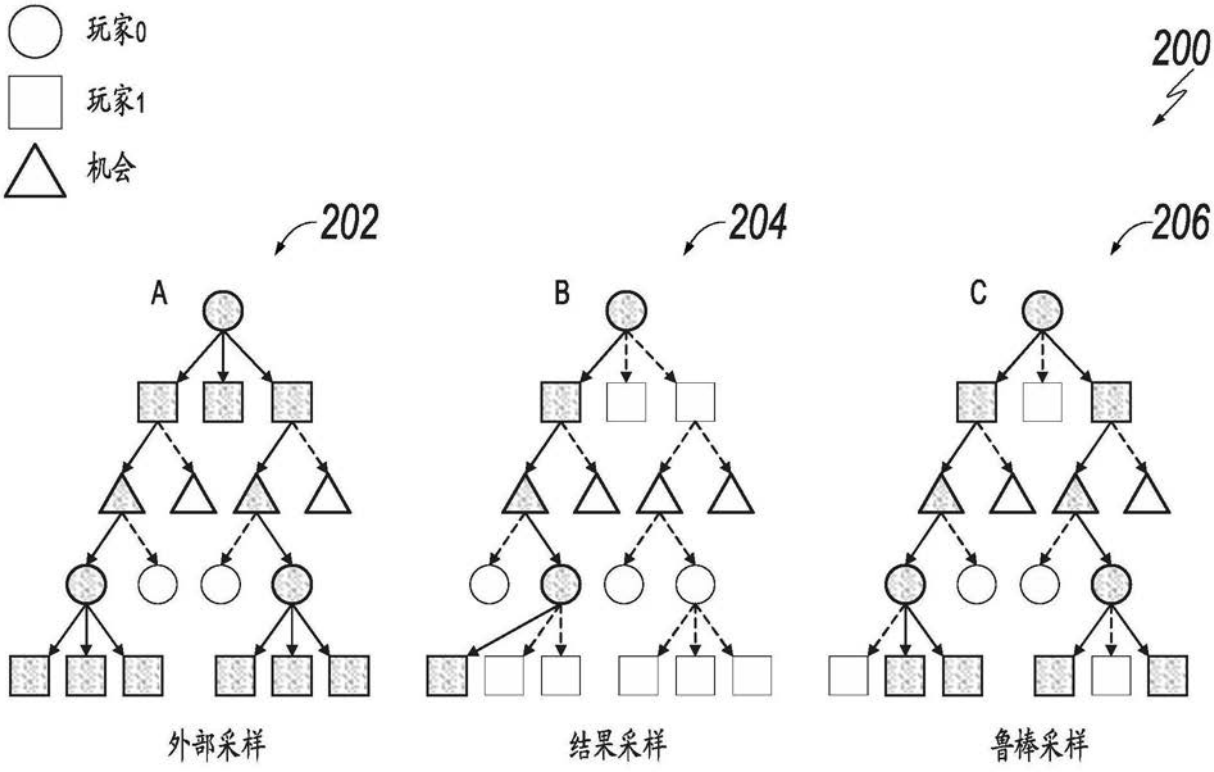

玩家0
玩家1
机会
200
202
204
206
A
B
C
外部采样
结果采样
鲁棒采样

图2

Algorithm: Robust Sampling Monte Carlo Counterfactual Regret Minimization

Function RS-MCCFR-ALG (t) :
  for $i = 1$ *to* t **do**
    RS-MCCFR ($t, \emptyset, 0, 1, 1$)
    RS-MCCFR ($t, \emptyset, 1, 1, 1$)

Function RS-MCCFR ($t, h, i, \pi_i, \pi_i^{rs(k)}$) :
  $I_i \leftarrow I_i(h)$ ▷ information set at sate h
  if $h \in Z$ **then**
    return $\frac{u_i(h)}{\pi_i^{rs(k)}}$ ▷ return game payoff
  else if $P(h) = -1$ **then**
    $a \sim \sigma_{-i}(I_i)$ ▷ sample an action from $\sigma_{-i}(h)$
    return RS-MCCFR($t, ha, i, \pi_i, \pi_i^{rs(k)}$)
  else if $P(h) = i$ **then**
    $\sigma_i(i_i) \leftarrow$ CalculateStrategy ($R_i^t(\cdot|I_i), I_i$) ▷ calculate current strategy
    $v_i(h) \leftarrow 0, r_i(\cdot|I_i) \leftarrow \vec{0}, s_i(\cdot|I_i) \leftarrow \vec{0}$ ▷ $r_i(\cdot|I_i)$ *and* $_i(\cdot|I_i)$ are two vectors over $A(I_i)$
    $A^{rs(k)}(I_i) \leftarrow$ sampling k different actions according to $\sigma_i^{rs(k)}$
    For $a \in A^{rs(k)}(I_i)$ **do**
      $v_i(a|h) \leftarrow$ RS-MCCFR($t, ha, i, \pi_i \sigma_i(a|I_i), \pi_i^{rs} \sigma_i^{rs(k)}(a|I_i)$)
      $v_i(h) \leftarrow v_i(h) + v_i(a|h)\sigma_i(a|I_i)$ ▷ update counterfactual value
    For $a \in A^{rs(k)}(I_i)$ **do**
      if $t == 0$ **then**
        $R_i^t(\cdot|I_i) \leftarrow R_i^{t-1}(\cdot|I_i) + (v_i(a|h) - v_i(h))$ ▷ update cumulative regret
        $S_i^t(a|I_i) \leftarrow S_i^{t-1}(a|I_i) + \pi_i^{\sigma}(I_i)\sigma_i(a|I_i)$ ▷ update average strategy numerator
      else
        $R_i^t(\cdot|I_i) \leftarrow (v_i(a|h) - v_i(h))$ ▷ update cumulative regret
        $S_i^t(a|I_i) \leftarrow \pi_i^{\sigma}(I_i)\sigma_i(a|I_i)$ ▷ update average strategy numerator
  else
    $\sigma_{-i}(I_i) \leftarrow$ CalculateStrategy ($R_{-i}^t(\cdot|I_i), I_i$) ▷ calculate current strategy
    $a \sim \sigma_{-i}(I_i)$ ▷ sample an action from $\sigma_{-i}(I_i)$
    return RS-MCCFR($t, ha, i, \pi_i, \pi_i^{rs(k)}$)

Function CalculateStrategy ($R_i(\cdot|I_i), I_i$) :
  $sum \leftarrow \Sigma_{a \in A(I_i)} \max(R_i(a|I_i), 0)$
  For $a \in A(I_i)$ **do**
    $\sigma_i(a|I_i) = \frac{\max(R_i(a|I_i), 0)}{sum}$ **if** sum > 0 **else** $\frac{1}{|A(I_i)|}$
  return $\sigma_i(I_i)$

300

图3

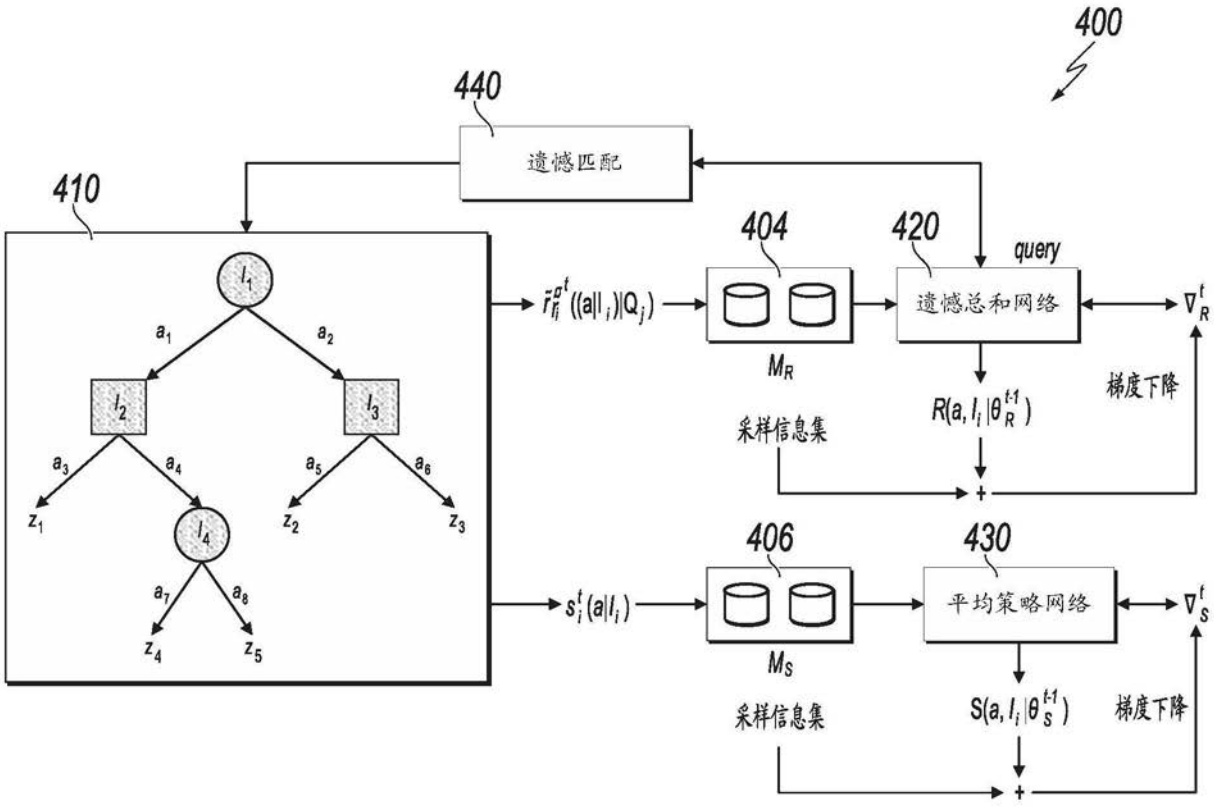

400
440
遗憾匹配
410
404
420
query
遗憾总和网络
梯度下降
采样信息集
406
430
平均策略网络
梯度下降
采样信息集

图4

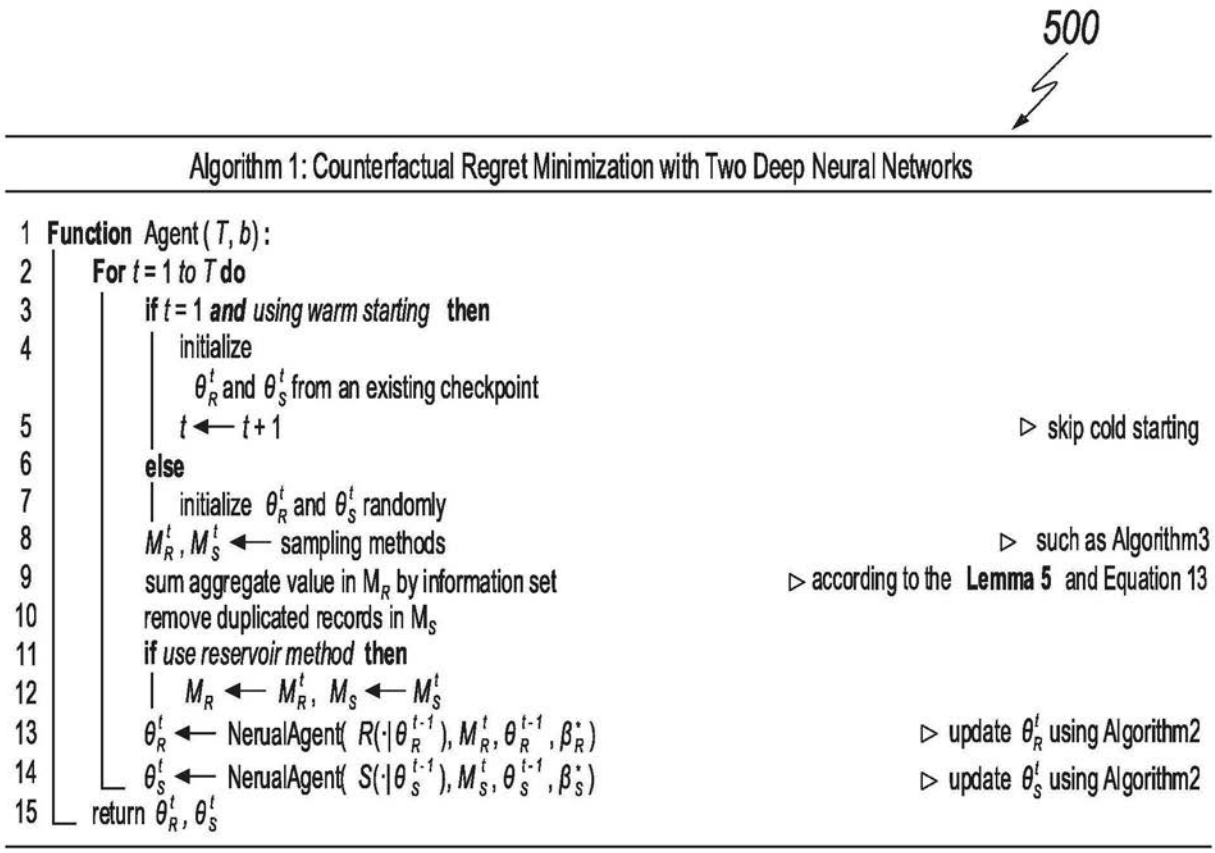

500
Algorithm 1: Counterfactual Regret Minimization with Two Deep Neural Networks
1 Function Agent (T, b):
2 For t = 1 to T do
3 if t = 1 and using warm starting then
4 initialize
θ_R^t and θ_S^t from an existing checkpoint
5 t ← t + 1 ▷ skip cold starting
6 else
7 initialize θ_R^t and θ_S^t randomly
8 M_R^t , M_S^t ← sampling methods ▷ such as Algorithm3
9 sum aggregate value in M_R by information set ▷ according to the Lemma 5 and Equation 13
10 remove duplicated records in M_S
11 if use reservoir method then
12 M_R ← M_R^t , M_S ← M_S^t
13 θ_R^t ← NeruralAgent(R(·|θ_R^{t-1}), M_R^t, θ_R^{t-1}, β_R^*) ▷ update θ_R^t using Algorithm2
14 θ_S^t ← NeruralAgent(S(·|θ_S^{t-1}), M_S^t, θ_S^{t-1}, β_S^*) ▷ update θ_S^t using Algorithm2
15 return θ_R^t , θ_S^t

图5



Algorithm 2: Optimization of Deep Neural Network

Function NeuralAgent $(f(\cdot|\theta^{t-1}), M, \theta^{t-1}, \beta^*)$
  initialize optimizer, scheduler ▷ gradient descent optimizer and learning rate scheduler
  $\theta^T \leftarrow \theta^{T-1}$, $l_{best} \leftarrow \infty$, $t_{best} \leftarrow 0$ ▷warm starting from the checkpoint of the last iteration
  For $t = 1$ to β_{epoch} **do**
    loss ← □ ▷ initialize *loss* as an empty list
    For each training *epoch* **do**
      $\{x^{(i)}, y^{(i)}\}_{i=1}^{m} \sim M$ ▷ sampling a mini-batch from M
      batch_loss $\leftarrow \frac{1}{m} \sum_{i=1}^{m} (f(x^{(i)}|\theta^{T-1}) + y^{(i)} - f(x^{(i)}|\theta^T))^2$
      back propagation batch_loss with learning rate *lr*
      clip gradient of θ^T to $[-\epsilon, \epsilon]^d$ ▷ d is the dimension of θ^T
      optimizer (*batch_loss*)
      loss.append (*batch_loss*)
    lr ← *sheduler* (*lr*) ▷ reduce learning rate adaptively when loss has stopped improving
    if *avg*(*loss*) < β_{loss} **then**
      $\theta_{best}^T \leftarrow \theta^T$, early stopping ▷ if loss is small enough, using early stopping mechanism
    else if *avg*(*loss*) < l_{best} **then**
      l_{best} = *avg*(*loss*), $t_{best} \leftarrow t$, $\theta_{best}^T \leftarrow \theta^T$
    if $t - t_{best} > \beta_{re}$ **then**
      lr ← β_{lr} ▷ reset learning rate to escape from potential saddle point or local minima
  return θ^T

图6

Algorithm 3: Mini-Batch RS-MCCFR with Double Neural Networks

Function Mini-Batch-MCCFR-NN(t):
  $M_R^t \leftarrow \emptyset$, $M_S^t \leftarrow \emptyset$
  For all $i = 1$ *to* b **do in parallel then**
    MCCFR-NN ($t, \emptyset, 0, 1, 1$)
    MCCFR-NN ($t, \emptyset, 1, 1, 1$)
  return M_R^t, M_S^t

Function MCCFR-NN($t, h, i, \pi_i, \pi_i^{rs(k)}$):
  $I_i \leftarrow I_i(h)$ ▷ information set at sate h
  if $h \in Z$ **then**
    return $\frac{u_i(h)}{\pi_i^{rs(k)}}$ ▷ return game payoff
  else if $P(h) = -1$ **then**
    $a \sim \sigma_{-i}(I_i)$ ▷ sample an action from $\sigma_{-i}(h)$
    return MCCFR-NN $(t, ha, i, \pi_i, \pi_i^{rs(k)})$
  else if $P(h) = i$ **then**
    $\hat{R}_i(\cdot|I_i) \leftarrow R(\cdot, I_i|\theta_R^t)$ **if** $t > 1$ **else** $\vec{0}$ ▷ inference the vector of cumulative regret $\mathrm{A}\, a \in A(I_i)$
    $\sigma_{-i}(I_i) \leftarrow$ CalculateStrategy ($\hat{R}_i(\cdot|I_i), I_i$) ▷ calculate current strategy
    $v_i(h) \leftarrow 0, r_i(\cdot|I_i) \leftarrow \vec{0}, s_i(\cdot|I_i) \leftarrow \vec{0}$ ▷ $r_i(\cdot|I_i)$ *and* $_i(\cdot|I_i)$ are two vectors over $A(I_i)$
    $A^{rs(k)}(I_i) \leftarrow$ sampling k different actions according to $\sigma_i^{rs(k)}$
    For $a \in A^{rs(k)}(I_i)$ **do**
      $v_i(a|h) \leftarrow$ MCCFR-NN($t, ha, i, \pi_i \sigma_i(a|I_i), \pi_i^{rs} \sigma_i^{rs(k)}(a|I_i)$)
      $v_i(h) \leftarrow v_i(h) + v_i(a|h)\sigma_i(a|I_i)$ ▷ update counterfactual value
    For $a \in A^{rs(k)}(I_i)$ **do**
      $r_i(a|I_i) \leftarrow v_i(a|h) - v_i(h)$ ▷ update cumulative regret
      $s_i(a|I_i) \leftarrow \pi_i^{\sigma}(I_i)\sigma_i(a|I_i)$ ▷ update average strategy numerator
    Store updated cumulative regret tuple ($I_i, r_i(\cdot|I_i)$) in M_R
    Store updated current strategy dictionary ($I_i, s_i(\cdot|I_i)$) in M_S
    return $v_i(h)$
  else
    $\hat{R}_{-i}(\cdot|I_i) \leftarrow R(\cdot, I_i|\theta_R^t)$ **if** $t > 1$ **else** $\vec{0}$ ▷ inference cumulative regret
    $\sigma_{-i}(I_i) \leftarrow$ CalculateStrategy ($\hat{R}_{-i}(\cdot|I_i), I_i$) ▷ calculate current strategy
    $a \sim \sigma_{-i}(I_i)$ ▷ sample an action from $\sigma_{-i}(I_i)$
    return MCCFR-NN $(t, ha, i, \pi_i, \pi_i^{rs(k)})$

Function CalculateStrategy ($R_i(\cdot|I_i), I_i$):
  $sum \leftarrow \Sigma_{a \in A(I_i)} \max(R_i(a|I_i), 0)$
  For $a \in A(I_i)$ **do**
    $\sigma_i(a|I_i) = \frac{\max(R_i(a|I_i), 0)}{sum}$ **if** sum > 0 **else** $\frac{1}{|A(I_i)|}$
  return $\sigma_i(I_i)$

700

图7

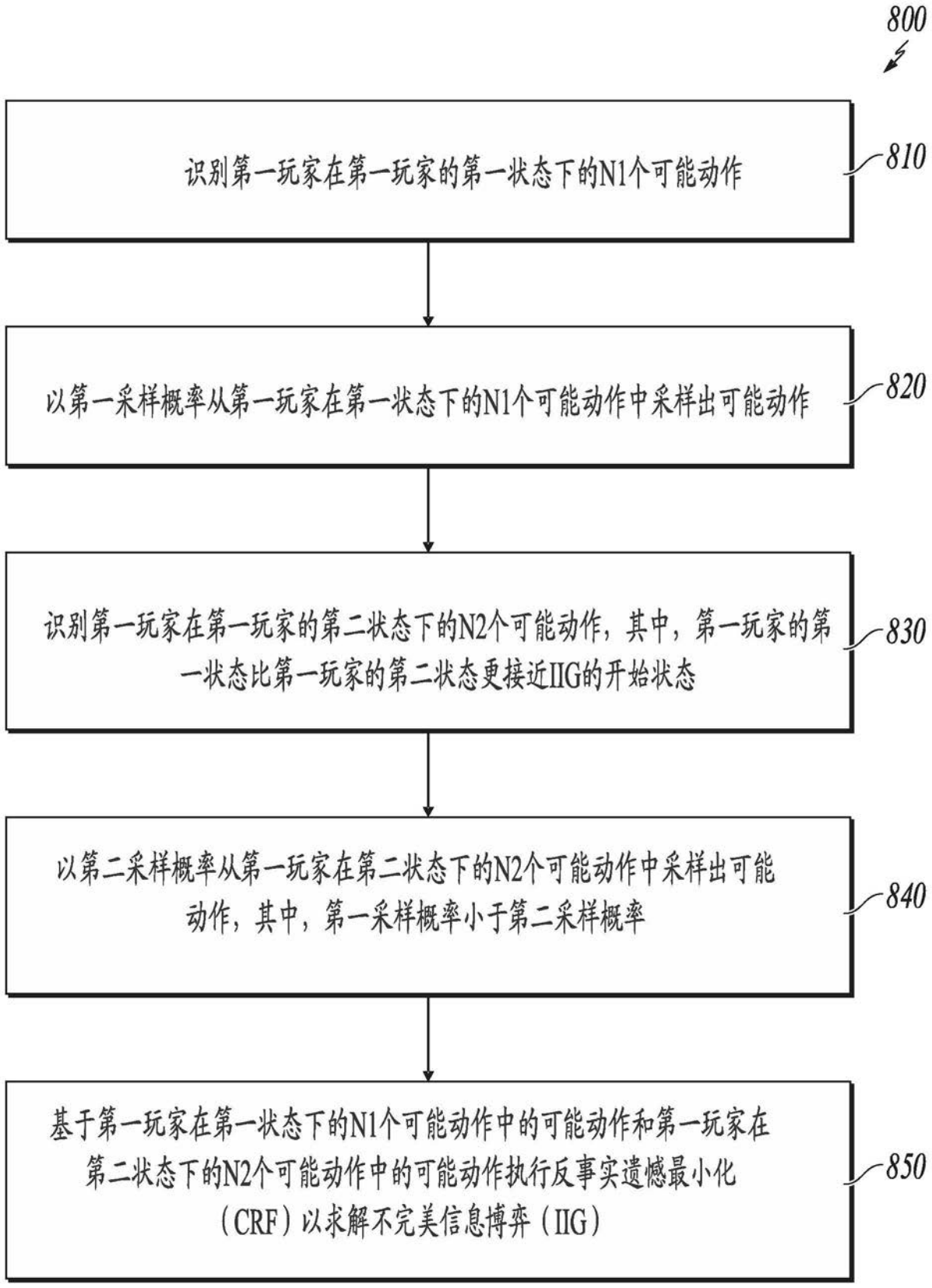

800
识别第一玩家在第一玩家的第一状态下的N1个可能动作
810
以第一采样概率从第一玩家在第一状态下的N1个可能动作中采样出可能动作
820
识别第一玩家在第一玩家的第二状态下的N2个可能动作，其中，第一玩家的第一状态比第一玩家的第二状态更接近IIG的开始状态
830
以第二采样概率从第一玩家在第二状态下的N2个可能动作中采样出可能动作，其中，第一采样概率小于第二采样概率
840
基于第一玩家在第一状态下的N1个可能动作中的可能动作和第一玩家在第二状态下的N2个可能动作中的可能动作执行反事实遗憾最小化（CRF）以求解不完美信息博弈（IIG）
850

图8

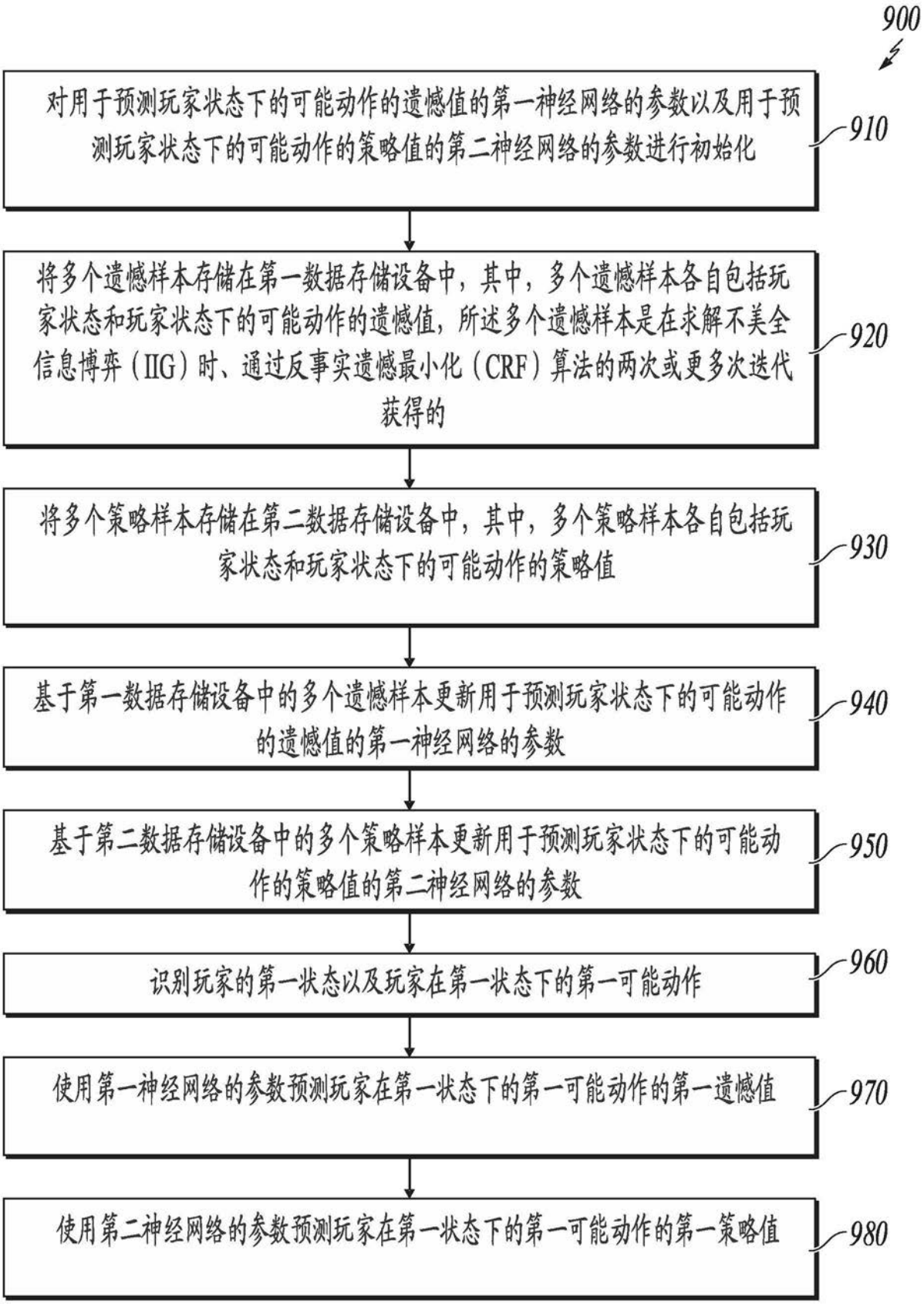

900
对用于预测玩家状态下的可能动作的遗憾值的第一神经网络的参数以及用于预测玩家状态下的可能动作的策略值的第二神经网络的参数进行初始化
910
将多个遗憾样本存储在第一数据存储设备中，其中，多个遗憾样本各自包括玩家状态和玩家状态下的可能动作的遗憾值，所述多个遗憾样本是在求解不美全信息博弈（IIG）时、通过反事实遗憾最小化（CRF）算法的两次或更多次迭代获得的
920
将多个策略样本存储在第二数据存储设备中，其中，多个策略样本各自包括玩家状态和玩家状态下的可能动作的策略值
930
基于第一数据存储设备中的多个遗憾样本更新用于预测玩家状态下的可能动作的遗憾值的第一神经网络的参数
940
基于第二数据存储设备中的多个策略样本更新用于预测玩家状态下的可能动作的策略值的第二神经网络的参数
950
识别玩家的第一状态以及玩家在第一状态下的第一可能动作
960
使用第一神经网络的参数预测玩家在第一状态下的第一可能动作的第一遗憾值
970
使用第二神经网络的参数预测玩家在第一状态下的第一可能动作的第一策略值
980

图9

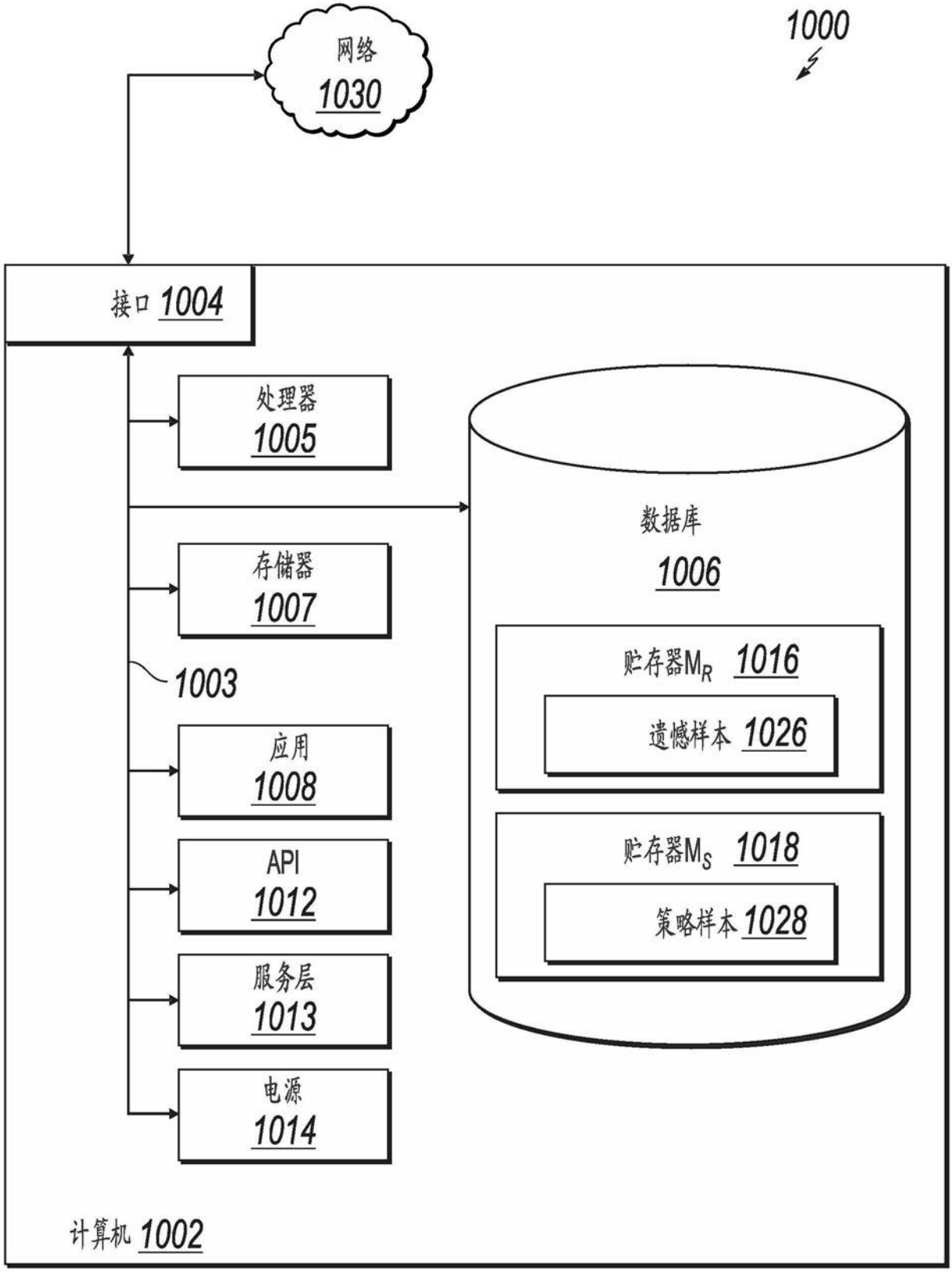

1000
网络
1030
接口 1004
处理器
1005
存储器
1007
1003
应用
1008
API
1012
服务层
1013
电源
1014
数据库
1006
贮存器M_R 1016
遗憾样本 1026
贮存器M_S 1018
策略样本 1028
计算机 1002

图10

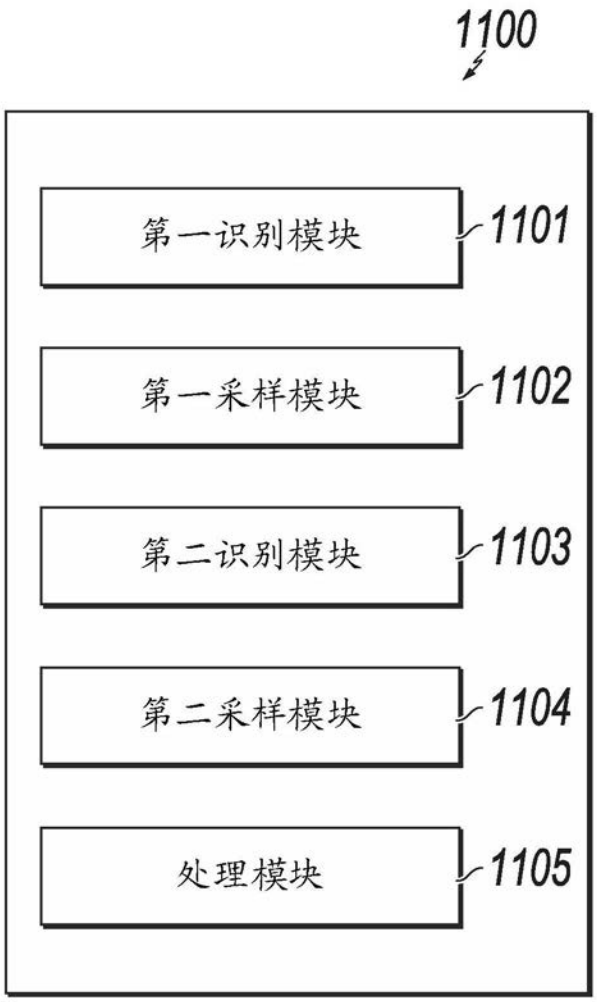

1100
第一识别模块
1101
第一采样模块
1102
第二识别模块
1103
第二采样模块
1104
处理模块
1105

图11

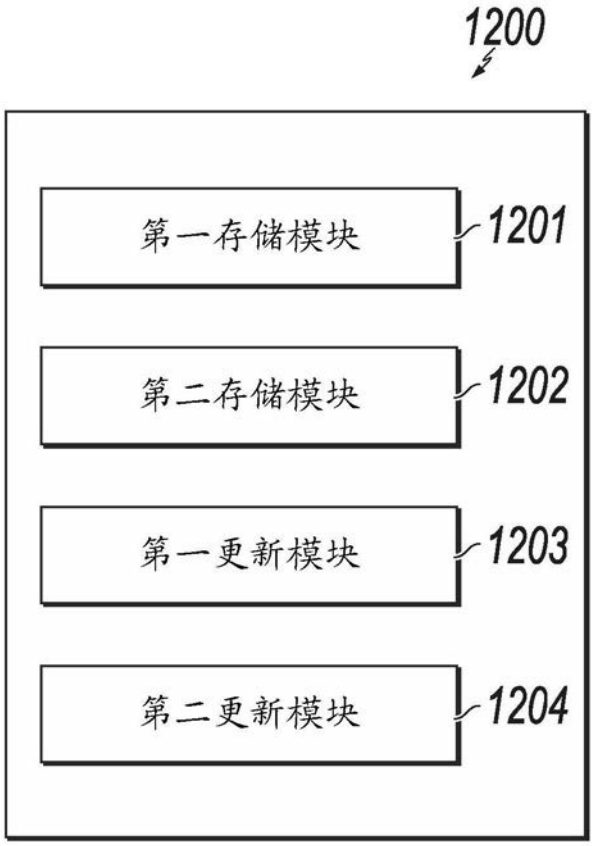


图12