深度逆向强化学习研究综述

陈希亮,曹 雷,何 明,李晨溪,徐志雄
CHEN Xiliang, CAO Lei, HE Ming, LI Chenxi, XU Zhixiong

陆军工程大学 指挥信息系统学院,南京 210007
College of Command Information System, Army Engineering University, Nanjing 210007, China

CHEN Xiliang, CAO Lei, HE Ming, et al. Overview of deep inverse reinforcement learning. Computer Engineering and Applications, 2018, 54(5):24-35.

Abstract: Deep inverse reinforcement learning is a new research hotspot in the field of machine learning. It aims at recovering the reward function of deep reinforcement learning by the experts' example trajectories. This paper systematically introduces three kinds of classic deep reinforcement learning methods. Then inverse reinforcement learning algorithms including apprenticeship learning, max margin plan, structured classification and probability models are described; then, some frontier researches of deep inverse reinforcement learning are reviewed, including the deep max margin plan inverse reinforcement learning, deep inverse reinforcement learning based on DQN and deep maximum entropy inverse reinforcement learning and recovering reward functions from non-expert trajectories etc. Finally, the existing issues and development direction are summarized.

Key words: deep learning; reinforcement learning; deep inverse reinforcement learning

摘 要: 深度逆向强化学习是机器学习领域的一个新的研究热点,它针对深度强化学习的回报函数难以获取问题,提出了通过专家示例轨迹重构回报函数的方法。首先介绍了3类深度强化学习方法的经典算法;接着阐述了经典的逆向强化学习算法,包括基于学徒学习、最大边际规划、结构化分类和概率模型形式化的方法;然后对深度逆向强化学习的一些前沿方向进行了综述,包括基于最大边际法的深度逆向强化学习、基于深度Q网络的深度逆向强化学习和基于最大熵模型的深度逆向强化学习和示例轨迹非专家情况下的逆向强化学习方法等。最后总结了深度逆向强化学习在算法、理论和应用方面存在的问题和发展方向。

关键词: 深度学习; 强化学习; 深度逆向强化学习
文献标志码: A **中图分类号:** TP181 doi:10.3778/j.issn.1002-8331.1711-0289

1 引言

传统强化学习方法在解决状态和动作空间有限的任务上都表现得不错,但在求解状态和动作空间维度很高的问题时,就显得无能为力。其局限性在于,有限样本和计算单元条件下对复杂函数的表示能力有限。解决上述问题的一个有效途径,就是将强化学习中的策略或值函数用线性函数、核函数、神经网络等[1]显性表达。其中,深度神经网络不仅具有强大的函数逼近能力,而且可以实现端到端学习,能够直接从原始输入映射到分类或回归结果,从而避免了由于特征提取等工作引入的人为因素。深度学习就是通过构建包含多隐层的深度神经网络模型,并基于大量的数据样本集进行网络参数学习,以实现对非线性复杂函数的逼近,最终达到提升分类或预测准确性的目的。

基金项目: 国家重点研发计划(No.2016YFC0800606);中国工程院重点咨询课题(No.2017-XZ-05);总装备部预研基金(No.9140A06020315JB25081);江苏省自然科学基金(No.BK20161469,No.BK20150721);中国博士后基金(No.2015M582786,No.2016T91017);江苏省重点研发计划(No.BE2015728,No.BE2016904)。

作者简介: 陈希亮(1985—),男,博士研究生,研究领域为深度强化学习,指挥信息系统工程,E-mail:383618393@qq.com;曹雷(1965—),男,教授,研究领域为指挥信息系统工程,决策理论与方法;何明(1978—),男,博士,教授,研究领域为机器学习,无线传感网;李晨溪(1989—),男,博士研究生,研究领域为深度强化学习,决策理论与方法;徐志雄(1994—),男,硕士研究生,研究领域为机器学习。

收稿日期: 2017-11-21 **修回日期:** 2018-01-18 **文章编号:** 1002-8331(2018)05-0024-12

但是，深度学习的局限性显而易见，其需要大量的样本数据，并且在很多任务上表现不佳[2]。在训练数据有限的条件下，通过直接模仿学习学得的策略适用性不强。其原因是，样本轨迹因不可能包括所有状态空间，监督学习学得的策略函数泛化能力有限。虽然增加训练时间和计算能力能在一定程度上弥补这个不足，但预测和泛化能力弱的问题并不能从根本上得到改善。深度强化学习算法由于能够基于深度神经网络实现从感知到决策控制的端到端自学习，与监督学习的方法相比，具有更强的预测和泛化能力。但是，在实际的多步强化学习中，设计回报函数是相当困难的。

基于以上探讨，构建报酬函数的困难是更广泛地应用强化学习的显著障碍，因而如何有效构造报酬函数对强化学习具有重要意义。如果能够基于专家演示数据让学习者从示例中学习，通常比直接指定回报函数要更加符合实际，这种从专家处学习（如观察学习、模仿学习、从演示中学习）构建回报函数的任务就是逆向强化学习（Inverse Reinforcement Learning，IRL）的思想[3]。

2　MDP模型与逆向强化学习

2.1　MDP模型

标准的强化学习设置为一个agent在离散时间步长的方式与环境进行交互，可以用马尔科夫决策过程（Markov Decision Processes，MDP）进行描述[4-5]，MDP可以定义为一个五元组 (S, A, π, R, γ)。在每一个时间步长 t，agent接收状态 s_t，并且依据策略 π 在可能的动作集合 A 中选择一个动作，π 是 s_t 到 a_t 的映射。作为回报，agent接收到下一个状态的 s_{t+1} 和接收一个奖励 R_t，这个过程持续到agent达到最终状态并重新启动的过程。返回 $R_t=\sum_{k=0}^{\infty}\gamma^k r^{t+k}$ 是在时间步长 t 采用折扣因子 $\gamma\in(0,1)$ 计算的累计收益。agent的目标是在每个状态 s_t 最大限度地提高预期收益[4-5]。

2.2　强化学习

强化学习的目的是要学习一个策略（Policy）。这个策略可以看作是一个函数，输入当前的状态 s，输出采取动作 a 的概率 $\pi(s,a)$。强化学习的策略就是选择相应的动作，能够使未来的奖励最大化。基于MDP的强化学习基本模型如图1所示。

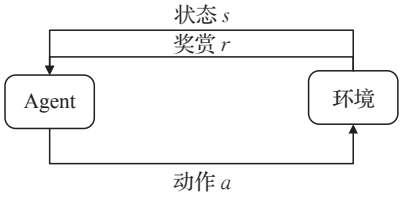

图1　基于MDP的强化学习基本模型

状态-动作值函数 $Q^{\pi}(s,a)=E\{R_t|s_t=s,a\}$ 表示在策略 π 时，从状态 s 出发，选取动作 a 后使用策略 π 的累计期望回报。最优值函数是在采用任意策略在状态 s 下，动作为 a 时的最大动作值回报。

$$Q^{*}(s,a)=\max_{\pi}Q^{\pi}(s,a)=\max_{\pi}E[R_t|s_t=s,a_t=a,\pi] \tag{1}$$

最优状态-动作值函数服从贝尔曼方程。这基于以下假设：如果在下一时间步长的序列 s' 的最优值 $Q^{*}(s',a')$ 对于所有可能的动作 a' 是已知的，则最佳策略是选择动作 a' 使得期望值 $r+\gamma Q^{*}(s',a')$ 最大：

$$Q^{*}(s,a)=E_{s'\sim\varepsilon}[r+\gamma\max_{a'}Q^{*}(s',a')|s,a] \tag{2}$$

同样的，状态值函数 $V^{\pi}(s)=E\{R_t|s_t=s\}$ 是在策略 π 时，状态 s 的值函数。是在策略 π 时，状态 s 的期望回报。因此状态值函数的 γ 折扣累计回报为：

$$V^{\pi}(s)=E_{\pi}\{\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0=s\} \tag{3}$$

在策略 π 时，动作值函数 $Q^{\pi}(s,a)$：$S\times A\rightarrow R$ 为：

$$Q^{\pi}(s,\ a)=E_{\pi}\{\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|s_0=s,\ a_0=a\} \tag{4}$$

2.3　逆向强化学习

强化学习是求累积回报期望最大时的最优策略，即时回报是人为或环境给定的。但是，在很多复杂任务中，环境在没有最终结果前回报稀疏。除外，人为设计即时回报函数非常困难，并且带有很大的主观性和经验性。回报函数的不同将导致最优策略的不同。如果没有合适的即时回报，强化学习算法将很难收敛。

在很多实际任务中存在一些专家完成任务的序列被认为获取了比较高的累积回报。人类专家在完成复杂任务时，可能未考虑回报函数。但是，这并不是说人类专家在完成任务时就没有回报函数。从某种程度上来讲，人类专家在完成具体任务时有潜在的回报函数。Ng等人提出[6]，专家在完成某项任务时，其决策往往是最优或接近最优的，可以假设，当所有的策略所产生的累积回报期望都不比专家策略所产生的累积回报期望大时，所对应的回报函数就是根据示例学到的回报函数。

因此，IRL可以定义为从专家示例中学到回报函数，即IRL考虑的情况是在MDP中，回报函数未知。相应的，有一个由专家演示轨迹组成的集合 $D=\{\xi_1,\xi_2,\cdots,\xi_n\}$，每一个演示轨迹都包括了一个状态动作对的集合 $\xi_i=\{(s_0,a_0),(s_1,a_1),\cdots,(s_k,a_k)\}$。由此，定义了一个没有回报函数的MDP\R过程，定义为元组 $\{S,A,T,\gamma,D\}$ [7]。如图2所示，在已知一系列专家示例策略 π^{*} 的情况下，是否能够还原回报函数 R，即IRL的目标是从专家演示中发现专家演示背后的回报函数的结构。

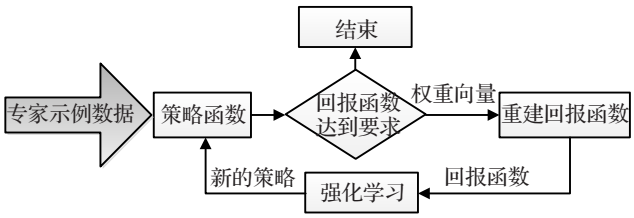

图2 逆向强化学习流程图

3 深度强化学习研究进展

很多实际情况中,强化学习的状态维度和动作维度过高,使Agent在巨大的状态或动作空间下,很难或无法遍历所有情况,导致算法收敛慢或无法学到合理的策略。解决上述问题的一个有效途径就是使用函数近似的方法,将值函数或者策略用一个函数显性表示。常用的有线性函数、核函数、神经网络等[8-9]。近年来最成功的就是将深度神经网络作为近似函数引入到强化学习中。本章从值函数近似、策略搜索和基于模型的强化学习三种分类介绍DRL方法。

3.1 值函数近似

深度学习与强化学习最早由Lange等人[10]将Auto-Encoder应用于强化学习中,解决了路径规划寻优的问题。2013年Mnih V等人在NIPS上提出的DQN算法[11],利用深度神经网络进行动作值函数的近似;采用经验回放机制,将探索环境得到的数据以记忆单元$(s_t, a_t, r_{t+1}, s_{t+1})$的形式储存起来,然后采取从记忆回放缓存中随机选取样本的方式来训练神经网络的参数,称为Nature DQN。DQN采用参数为θ的深层神经网络值函数进行近似,$Q(s,a;\theta)\approx Q^*(s,a;\theta)$,这种无模型的强化学习算法解决了"模型灾难问题",而采用值函数的泛化逼近方法解决了强化学习的"维数灾难问题"[12]。

然而研究者在实验中发现[13-15],采用DQN训练的强化学习在对Q函数进行逼近时存在不稳定现象,主要原因如下:(1)观察序列的数据具有较大的相关性,导致基于梯度下降的优化算法失效;(2)训练的Q函数的微小变化会导致策略的巨大改变,导致算法不易收敛。其中,原始DQN的经验回放机制解除了观察序列的相关性,经验回放机制先将探索环境中的数据存储起来,之后从存储的数据中随机采样以更新深度神经网络的参数;第二个问题由Mnih V等人[13]在2015年提出Target DQN迭代式更新的方式,进一步减小了数据的关联性。

在标准的Q-learning算法中,采用ε贪心算法求取最大值时,在动作的选择与评估上采用了相同的网络,这种方法很容易过估计。为预防这种情况,可以对评估采用不同的网络,这就是Van等人[16]在2010年提出的Double Q-learning算法,2015年Van等人[17]在double Q-learning算法的基础上,训练两个值函数。这样的话就有了两个权重集,θ^-和θ'^-,在每次更新的时候,一个用于贪心算法策略的选择,一个用于值函数的评估。

强化学习通常采用玻尔兹曼机进行状态动作对的迭代,每一步都要对动作值进行评估。但是,对于很多状态来说,不需要对所有动作的值进行评估。对于某些状态,选择哪个动作是极为重要的,但是对于一些状态,不管发生什么情况都对动作的选择没有影响。基于这个认识,Deepmind团队的Ziyu Wang等人[18]设计了竞争网络架构,包括两个共享一个共同的卷积特征学习模块的流表示的值函数和优势函数使得值函数的估计更加精确。在频繁出现agent采取不同动作但对应值函数相等的情形下,竞争架构的DQN模型性能提升最为明显[18]。由于竞争网络框架的DQN输出是一个Q函数,它可以与许多现有的算法训练,如DDQN和SARSA等。此外,它可以利用这些算法的任何改进,包括更好的重播记忆,更好的探索策略和内在动机等。

为解决观察序列相关性和算法难收敛问题,Schaul T等人[14]提出了Prioritized Replay的采样策略改进DQN,实现了对重要记忆单元的优先回放,取得了更好的效果。对于部分可观察的MDP问题,Graves A等人[19]将RNN与强化学习结合,提出了DRQN算法,也取得了很好的效果。Osband等[20]提出一种引导深度Q网络,通过使用随机值函数让探索的效率和速率得到了显著的提升。

3.2 策略近似

以DQN算法为基础的值函数近似方法在实践中取得了突破性的进展,但值函数近似仍然存在求取随机性策略困难,难以解决连续动作问题等。因此,David Silver等人提出了DPG[21]算法。DPG算法包含一个参数化的Actor策略函数$a=\pi_\theta(s)$,代表当前的策略能够确定性地映射状态到一个动作。Critic值函数$Q(s,a;\theta)$基于Q-Learning来学习。David Silver证明了等式:

$$\nabla_\theta J(\theta)=E_{\pi_\theta}[\nabla_\theta\pi_\theta(s)\nabla_a Q^{\pi_\theta}(s,a)|a=\pi_\theta(s)] \quad (5)$$

就是目标函数的梯度。式(5)与值函数近似的目标函数梯度最大的区别在于其策略函数变成了确定性策略函数。这样在进行估计时,不再考虑动作空间,极大提高了连续空间中梯度的估计效率。

DDPG[22]在DPG的基础上,将Actor-Critic算法与深度神经网络结合。除使用DQN的经验回放和目标网络分离之外,还使用了batch normalization以提升深度学习的性能。

Mnih等人[15]提出了异步梯度下降的轻量级框架训练深度神经网络,与RL方法结合取得了较好且稳定的效果,尤其是A3C算法(Asynchronous Advantage Actor-Critic)大幅提升了算法效率,并提出DRL的一个新范式:并行地异步执行多agent运算,在每个时间步长,并行agent会经历多个不同的状态,使数据多样化,去掉了

关联性,因而不再需要记忆回放机制。除外,策略梯度方法还有GPS[23]、TRPO[24]、SVG[25]、GAE[26]等。

3.3　间接强化学习

除了值函数近似和策略近似,间接深度强化学习方法也取得了一定的进展。Gu等人[27]提出了NAF算法,基于Dyna[28]或LQG[29]框架,使用简单的局部线性模型来预测行动后的下一个状态,同时使用卡尔曼滤波根据想要达成的目标状态采取相应的控制策略。NAF利用优势函数,将 $Q(s,a)$ 分解成 $V(s)$ 和 $A(s,a)$,然后将 $A(s,a)$ 建模成一个关于 a 的二次函数,而建模这个二次函数的方法是通过建模条件均值和方差进行的,这样可以用解析方法直接得到给定 s 后 a 的最优解。这样做的好处是可以用一个模型同时对 $Q(s,a)$ 和 $V(s)$ 建模。

此外,知识驱动的强化学习方法也取得了一定的进展[5,30]。

4　逆向强化学习基础算法

RL方法寻求策略使得在给定回报函数的情况下累计回报期望最大。相比之下,IRL寻求回报函数使得专家策略轨迹的累计回报最大。IRL在专家不能给出任务的回报函数时是非常有用的。Agent可以从专家的行为中学习回报函数,并模仿它。

如2.3节所述,IRL的输入是在MDP模型中,根据专家示例轨迹还原回报函数R的过程,使得最优策略 π 与示例轨迹一致的过程。假设最优策略为 π,根据贝尔曼最优方程,可知 $\pi(s)\equiv a_1$ 的充分必要条件为:

$$a_1\equiv\pi(s)\in\arg\max_{a\in A}\sum_{s'}P_{sa}(s')V^{\pi}(s')\ \forall s\in S \tag{6}$$

$$\Leftrightarrow\sum_{s'}P_{sa_1}(s')V^{\pi}(s')\geqslant\sum_{s'}P_{sa}(s')V^{\pi}(s')$$
$$\forall s\in S,\ a\in A \tag{7}$$

写成向量形式,上式为:

$$P_{a_1}V^{\pi}\geqslant P_aV^{\pi}\quad\forall a\in A\backslash a_1 \tag{8}$$

由于 $\pi(s)\equiv a_1$,等式(1)贝尔曼方程可以写为:

$$V^{\pi}=(I-\gamma P_{a_1})^{-1}R \tag{9}$$

将贝尔曼最优方程(9)带入得:

$$P_{a_1}(I-\gamma P_{a_1})^{-1}R\geqslant P_a(I-\gamma P_{a_1})^{-1}R\quad\forall a\in A\backslash a_1 \tag{10}$$

因此:

$$a_1\equiv\pi(s)\Leftrightarrow(P_{a_1}-P_a)(1-\gamma P_{a_1})^{-1}R\geqslant 0 \tag{11}$$

根据以上证明,Ng等人给出了以下定理[6]。

定理1　定义一个有限状态集 S,动作 $A=\{a_1,a_2,\cdots,a_k\}$,转移概率矩阵 P_a,折扣因子 $\gamma\in(0,1]$,则最优策略 π 中状态 s 上的动作 $\pi(s)\equiv a_1$ 的充分必要条件是,对于所有的动作 $a=a_2,a_3,\cdots,a_k$,回报函数满足 $(P_{a_1}-P_a)(1-\gamma P_{a_1})^{-1}R\geqslant 0$。

定理1给出了求解回报函数的理论基础,但是由于(1)退化解的存在:无论采取什么行动,$R=0$ 总是一个解决方案,那么包括 $\pi(s)\equiv a_1$ 在内的任何策略都是最优的;(2)回报函数的歧义性:可能存在多个回报函数 R 满足约束条件,如何在多个满足约束的回报函数中选择一个最优的。IRL算法在此基础上寻求解决退化解和歧义性的问题。

4.1　学徒学习方法

为解决退化解和回报函数歧义性问题,Abbeel等人[7]提出了学徒学习(Apprenticeship Learning,AL)方法,在值函数空间内对IRL进行扩展。它将回报函数表示为一系列特征值的线性组合,未知的回报函数一般都是状态的函数,因此将回报函数定义为 $R(s)$,参数化为 K 个特征函数 $\phi_k(s,a)$ 的和:

$$R(s)=\sum_{k=1}^{K}\theta_k\phi_k(s) \tag{12}$$

智能体从专家示例中学到回报函数,使得在该回报函数下所得到的最优策略在专家示例策略附近。回报函数 $R(s)$ 一般是状态到奖励的映射,因此可以用函数逼近方法对其进行参数逼近,逼近形式定义为:$R(s)=\theta^{\mathrm{T}}\cdot\phi(s)$,其中 $\phi(s)$ 为基函数。IRL求的是回报函数中的参数 θ。根据值函数的定义,策略 π 的值函数为:

$$E_{s_0\sim D}\left[V^{\pi}(s_0)\right]=E\left[\sum_{t=0}^{\infty}\gamma^tR(s_t)|\pi\right]=$$
$$E\left[\sum_{t=0}^{\infty}\gamma^t\theta\cdot\phi(s_t)|\pi\right]=\theta\cdot E\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t)|\pi\right] \tag{13}$$

定义特征期望为:$\mu^{\pi}=E\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t)|\pi\right]$。需要注意的是,特征期望跟策略 π 有关,策略不同时,策略期望也不相同。

定义了特征期望之后,值函数可以写为:

$$E_{s_0\sim D}\left[V^{\pi}(s_0)\right]=\theta\cdot\mu^{\pi}(s_0) \tag{14}$$

当给定 m 条专家示例轨迹后,根据定义可以估计专家策略的特征期望为:

$$\tilde{\mu}_E=\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{\infty}\gamma^t\phi(s_t^{(i)}) \tag{15}$$

基于学徒学习的IRL可以归结为:找到一个策略,使该策略的表现与专家策略相近。可以利用特征期望来表示一个策略的好坏,就是找到一个策略 $\tilde{\pi}$ 的特征期望与专家策略的特征期望相近,如下不等式成立:

$$\left\|\mu(\tilde{\pi})-\mu_E\right\|_2\leqslant\epsilon \tag{16}$$

当该不等式成立时,对于任意的权重:$\|\theta\|_1\leqslant 1$。

值函数满足如下不等式:

$$\left|E\left[\sum_{t=0}^{\infty}\gamma^tR(s_t)|\pi_E\right]-E\left[\sum_{t=0}^{\infty}\gamma^tR(s_t)|\tilde{\pi}\right]\right|=$$
$$\left|\theta^{\mathrm{T}}\mu(\tilde{\pi})-\theta^{\mathrm{T}}\mu_E\right|\leqslant\|\theta\|_2\left\|\mu(\tilde{\pi})-\mu_E\right\|_2\leqslant 1\cdot\epsilon=\epsilon \tag{17}$$

学徒学习算法求解策略 $\tilde{\pi}$ 的方法如算法1所示。

算法1 基于学徒学习的IRL算法

步骤1 随机选取策略 $\pi^{(0)}$，计算(或者采用蒙特卡洛算法近似) $\mu^{(0)}=\mu(\pi^{(0)})$，设置 $i=1$；

步骤2 求 $\boldsymbol{w}$：计算 $t^{(i)}=\max_{\theta:\|\theta\|_2\leqslant 1}\min_{j\in(0,1,\cdots,i-1)}\boldsymbol{\theta}^{\mathrm{T}}(\mu_E-\mu^{(j)})$，取 $\boldsymbol{w}^{(i)}=\max_{t^{(i)}}\boldsymbol{w}$；

步骤3 如果 $t^{(i)}<\varepsilon$，终止；

步骤4 求最优策略：在回报函数为 $R=(\boldsymbol{w}^{(i)})^{\mathrm{T}}\phi$ 时，采用强化学习算法，求得最优策略 $\pi^{(i)}$；

步骤5 计算或估计 $\mu^{(i)}=\mu(\pi^{(i)})$；

步骤6 设置 $i=i+1$，回到步骤2。

其中，步骤2的目标函数为：

$$t^{(i)}=\max_{\boldsymbol{w}:\|\boldsymbol{w}\|_2\leqslant 1}\min_{j\in(0,1,\cdots,i-1)}\boldsymbol{w}^{\mathrm{T}}(\mu_E-\mu^{(j)}) \quad (18)$$

可以看作是根据专家示例轨迹猜测回报函数的过程。写成标准的优化形式为：

$$\begin{aligned}&\max_{t,\boldsymbol{w}}\ t\\&\text{s.t. }\boldsymbol{w}^{\mathrm{T}}\mu_E\geqslant\boldsymbol{w}^{\mathrm{T}}\mu^{(j)}+t,\ j=0,1,\cdots,i-1\\&\|\boldsymbol{w}\|_2\leqslant 1\end{aligned} \quad (19)$$

公式(19)表明算法试图找到一个回报函数 $R(s)$ 使得 $E_{s_0\sim D}\left[V^{\pi_E}(s_0)\right]\geqslant E_{s_0\sim D}\left[V^{\pi^{(i)}}(s_0)\right]+t$，即：使得专家示例轨迹的累计回报比其他任何算法找到的策略的累计回报值大，边际为 t。

步骤2求解时，$\mu^{(j)}$ 中的 $j\in\{0,1,\cdots,i-1\}$ 是前 $i-1$ 次迭代得到的最优策略。即第 i 次求解参数时，$i-1$ 次迭代的策略已知。此时最优函数值 t 相当于专家策略 μ_E 与 $i-1$ 个迭代策略之间的最大边际。

如图3所示，专家策略为一类，其他策略为另一类，参数的求解等价于找一条超曲面将专家策略和其他策略分开，即使得两类之间的边际最大。步骤4是在第二行求出参数后，便有了回报函数 $R=(\theta^{(i)})^{\mathrm{T}}\phi$，利用该回报函数进行强化学习，从而得到该回报函数下的最优策略 $\pi^{(i)}$。

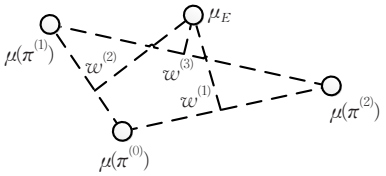

图3 最大边际方法的逆向强化学习

4.2 最大边际规划算法

最大边际规划算法(Max Margin Plan，MMP)将IRL问题建模为：$D=\{(S_i,\mathcal{A}_i,p_i,F_i,y_i,\mathcal{L}_i)\}_{i=1}^{n}$，其中，$S_i$ 为状态空间，$\mathcal{A}_i$ 为动作空间，p_i 为状态转移概率，f_i 为回报函数的特征向量，y_i 为专家示例轨迹，$\mathcal{L}_i$ 为策略损失函数：

$$\begin{aligned}&\min_{w,\varsigma_i}\frac{1}{2}\|\boldsymbol{\theta}\|^2+\frac{\gamma}{n}\sum_i\beta_i\varsigma_i^q\\&\text{s.t. }\boldsymbol{\theta}^{\mathrm{T}}f_i(y_i)+\varsigma_i\geqslant\max_{y\in Y}\boldsymbol{\theta}^{\mathrm{T}}f_i(y_i)+L_i(y)\end{aligned} \quad (20)$$

其中 ς_i 为松弛变量，允许违反这些约束对于以超参数 $\gamma\geqslant 0$ 进行缩放惩罚。$\beta^i>0$ 是数据相关的标量，当示例具有不同的长度时可用于标准化。这里的策略损失函数 $L_i(y)$ 是指策略 y 与第 i 条专家轨迹 y_i 之间的差，该差可以利用轨迹中两种策略选择不同动作的综合来衡量。$q\in\{1,2\}$ 用于区分是采用L1或L2的松弛处罚[31]。

这个框架描述了影响状态动作对的损失函数，在确定性非循环的路径规划问题中，损失函数是规划者访问示例数据没有访问的状态的次数，通过平滑这个损失函数能够得到更好的性能，这样的话临近的路径也是可以接受的。在一个通用的MDP模型中，在任何一个状态下，惩罚与专家示例数据不同的动作，或者惩罚专家示例轨迹选择不采取的动作[32]。假设 $\mathcal{L}(y,y_i)\geqslant 0$。

MMP需要找到使得专家示例策略具有比其他策略更大累计回报的状态到回报的映射，映射的参数为向量 $\boldsymbol{\theta}$，在这个映射下，最优策略能够逼近专家示例策略[32]。其中约束包括：

(1)约束只允许专家示例得到最好的回报的那些权值存在。

(2)回报的边际差，即专家示例轨迹的值函数与其他策略的值函数的差值与策略损失函数成正比。

μ_i 表示第 i 个专家策略，μ 表示任意的策略。回报函数可利用特征的线性组合表示，式(20)中回报函数 $f_i(y_i)=F_i\mu_i$，其中 F_i 为特征基底。问题(20)可形式化：

$$\begin{aligned}&\min_{\boldsymbol{w},\varsigma_i}\frac{1}{2}\|\boldsymbol{\theta}\|^2+\frac{\gamma}{n}\sum_i\beta_i\varsigma_i^q\\&\text{s.t. }\boldsymbol{\theta}^{\mathrm{T}}F_i\mu_i+\varsigma_i\geqslant\max_{\mu\in G_i}\boldsymbol{\theta}^{\mathrm{T}}F_i\mu+l_i\mu\end{aligned} \quad (21)$$

其中 μ 是指每个状态被访问的频次，状态 s' 处的频次应满足流入流出关系：

$$\sum_{x,a}\mu^{x,a}p_i(x'|x,a)+s_i^{x'}=\sum_a\mu^{x',a} \quad (22)$$

其中的 $s_i^{x'}$ 表示初始位置。接下来处理不等式约束右侧的最大值，右侧的最大值等价于下面问题：

$$\begin{aligned}&\max_{\mu\in G_i}\boldsymbol{w}^{\mathrm{T}}F_i\mu+l_i\mu\\&\text{s.t. }\sum_{x,a}\mu^{x,a}p_i(x'|x,a)+s_i^{x'}=\sum_a\mu^{x',a}\end{aligned} \quad (23)$$

根据拉格朗日对偶原理，其对偶问题为：

$$\begin{aligned}&\min_{v\in V_i}\boldsymbol{s}_i^{\mathrm{T}}v\\&\text{s.t. }\forall x,a,v^x\geqslant\left(\boldsymbol{w}^{\mathrm{T}}F_i+l_i\right)^{x,a}+\sum_{x'}p_i(x'|x,a)v^{x'}\end{aligned} \quad (24)$$

这样MMP将IRL问题转化为一个二次规划问题：

$$\min_{\boldsymbol{w},\varsigma_i,\upsilon_i}\frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\gamma}{n}\sum_i\beta_i\varsigma_i^q$$

$$\text{s.t. }\forall i\ \boldsymbol{w}^{\mathrm{T}}F_i\mu_i+\varsigma_i\geqslant\boldsymbol{s}_i^{\mathrm{T}}\upsilon_i$$

$$\forall i,x,a,\upsilon_i^x\geqslant\left(\boldsymbol{w}^{\mathrm{T}}F_i+l_i\right)^{x,a}+\sum_{x'}p_i(x'|x,a)\upsilon_i^{x'}\quad(25)$$

4.3　结构化分类方法

AL和MMP方法能够通过迭代更新的方式实现对回报函数的优化,每次迭代更新都需要依据回报函数进行强化学习,这是一种及其低效的方法。因此,Klein等人[33]提出结构化分类(Structured Classification,SC)方法,该方法通过线性化参数方式从训练集中生成一个多分类器,设置线性参数化的回报函数。给定状态的决策规则是使得该状态的累计回报最大。SC的基本思想是对专家的期望特征进行估计,作为参数化的回报函数。这样计算出来的参数定义的回报函数使得专家策略接近最优。

这种方法的明显优势是它不需要解决多次直接强化学习。它需要通过策略评估的方法估计专家特征期望。此外,使用一些启发式的方法,SC可以从专家轨迹样本中采样,而不需要对整条轨迹样本进行采样。在学徒学习算法中,回报函数可表示为:

$$R_\theta(s)=\boldsymbol{\theta}^{\mathrm{T}}\phi(s)\quad(26)$$

动作值函数表示为:

$$Q_\theta^\pi(s,a)=\boldsymbol{\theta}^{\mathrm{T}}\mu^\pi(s,a)\quad(27)$$

其中 $\mu^\pi(s,a)=E[\sum_{t>0}\gamma^t\phi(s_t)|s_0=s,\ a_0=a,\pi]$ 称为特征函数。关于特征函数,第 i 个元素 $\mu_i^\pi(s,a)=Q_{\phi_i}^\pi(s,a)$,可以理解为立即回报函数为 $\phi_i(s)$ 时对应的值函数。最后得到的行为值函数其实是不同立即回报函数所对应的值函数的线性组合。

为避免迭代求解MDP,可以这样考虑问题:对于一个行为空间很小的问题,最终的策略其实是找到每个状态所对应的最优动作。每个动作可以看作一个类标签,那么策略就是把所有的状态分成四类,分类的标准是值函数,正确的分类对应最大的值函数。利用这个思想,对于专家示例轨迹 ξ_i,IRL可以形式化为:

$$\min_{\theta,\varsigma}\frac{1}{2}\|\boldsymbol{\theta}\|^2+\frac{\eta}{N}\sum_{i=1}^N\varsigma_i$$

$$\text{s.t. }\forall i,\ \boldsymbol{\theta}^{\mathrm{T}}\hat{\mu}^{\pi_E}(s_i,a_i)+\varsigma_i\geqslant\max_a\boldsymbol{\theta}^{\mathrm{T}}\hat{\mu}^{\pi_E}(s_i,a)+L(s_i,a)\quad(28)$$

约束中的 $\{s_i,a_i\}$ 为专家轨迹元组,$\hat{\mu}^{\pi_E}(s_i,a_i)$ 可以利用蒙特卡罗的方法求解。而对于 $\hat{\mu}^{\pi_E}(s_i,a\neq a_i)$,则可以利用启发的方法来得到。

结构化分类方法和最大边际规划方法有很多相似的地方。但两者的本质不同体现在,结构化分类方法对每个状态处的每个动作进行约束,而最大边际规划方法是对一个MDP解进行约束。从计算量来看,结构化分类方法要小很多。

4.4　基于概率模型形式化方法

MMP和SC方法往往会产生歧义,比如或许存在很多不同的回报函数导致相同的专家策略。在此情况下,学到的回报函数可能具有随机的偏好。为克服这个缺点,Ziebart等人[34-35]利用概率模型,提出基于最大熵的和基于交叉熵的IRL方法。

4.4.1　基于最大熵的逆向强化学习

最大熵原理指出:对一个随机事件的概率分布进行预测时,预测应当满足全部已知条件,并且不能对未知情况做任何主观假设。在此情况下,概率分布最均匀,预测风险最小,因为这时概率的信息熵最大,所以被称为最大熵模型[35]。在IRL中,特征期望可定义为:

$$\mu(\pi)=E[\sum_{t=0}^{\infty}\gamma^t\phi(s_t)|\pi]\quad(29)$$

给定 m 条专家示例轨迹时,专家示例轨迹的特征期望为:

$$\hat{\mu}_E=\frac{1}{m}\sum_{i=1}^m\sum_{t=0}^{\infty}\gamma^t\phi(s_t^{(i)})\quad(30)$$

从概率模型的角度建模IRL,可定义为:存在一个潜在的概率分布,在该概率分布下,产生了专家轨迹。这是典型的已知数据求模型的问题。即:知道专家轨迹,求解产生该轨迹分布的概率模型。此时,已知条件为:

$$\sum_{Path\,\xi_i}P(\xi_i)f_{\xi_i}=\tilde{f}\quad(31)$$

这里用 f 表示特征期望,$\tilde{f}$ 表示专家特征期望。满足公式(31)约束条件的所有概率分布中,熵最大的概率分布是除了约束外对其他任何未知信息没有进行任何假设。因此,最大熵方法可以避免歧义性问题。IRL的目标函数可定义为熵最大的最优问题。形式化为:

$$\max -p\lg p$$

$$\text{s.t. }\sum_{Path\varsigma_i}P(\varsigma_i)f_{\varsigma_i}=\tilde{f}$$

$$\sum P=1\quad(32)$$

利用拉格朗日乘子法,该优化问题可转化为:

$$\min L=\sum_{\varsigma_i}p\lg p-\sum_{j=1}^n\lambda_j\left(pf_j-\tilde{f}_j\right)-\lambda_0\left(\sum p-1\right)\quad(33)$$

对概率 p 进行微分,并令其导数为0,可以得到:

$$\frac{\partial L}{\partial p}=\sum_{\varsigma_i}\lg p+1-\sum_{j=1}^n\lambda_jf_j-\lambda_0=0\quad(34)$$

最后得到拥有最大熵的概率为:

$$p=\frac{\exp(\sum_{j=1}^n\lambda_jf_j)}{\exp(1-\lambda_0)}=\frac{1}{Z}\exp(\sum_{j=1}^n\lambda_jf_j)\quad(35)$$

对于确定性MDP问题而言,这个函数是凸的,它的最优解可以使用基于梯度的优化方法得到。

其中参数 θ_j 对应着回报函数中的参数。可以利用最大似然的方法进行求解。一般而言,利用最大似然的方法对式(35)中的参数进行求解时,往往会遇到未知的配分函数项 Z,因此不能直接求解。一种可行的方法是利用次梯度的方法。

$$\nabla L(\theta)=\tilde{f}-\sum_{\xi}P(\xi|\theta,T)f_{\xi}=\tilde{f}-\sum_{s_i}D_{s_i}f_{s_i} \tag{36}$$

4.4.2 基于交叉熵的逆向强化学习

IRL的目标是求得回报函数使得专家策略最优,即求出一个分布,使得依该分布生成的轨迹数据与专家数据一致,即专家轨迹策略的分布与生成的最优轨迹数据的分布差异最小化。即使得两个分布的交叉熵最小。

在基于最大熵的IRL中,最后求解参数时,需要利用如公式(36)所示的次梯度,次梯度计算时需要利用轨迹的概率 $P(\varsigma)$。该轨迹的概率可表示为[35]:

$$P_{\xi}(\tau|\theta,T)\propto d_0(s_1)\exp(\sum_{i=1}^{k}\theta_i f_i^{\xi})\prod_{t=1}^{H}T(s_{t+1}|s_t,a_t) \tag{37}$$

其中 $\xi=s_1a_1s_2a_2\cdots s_Ha_H$。求解该式的前提是系统的状态转移概率 $T(s_{t+1}|s_t,a_t)$ 是已知的。然而,在无模型的强化学习中,该模型是未知的。为了解决这个问题,Boularias等人[36]受交叉熵的启发将该问题建模为求解交叉熵最大。

设 D 为专家示例轨迹的集合,示例轨迹的长度为 H,P 为集合 D 上的轨迹的概率分布。设 Q 为利用基准策略和转移矩阵 T^a 产生的轨迹分布,要求解的问题可形式化为求解 P 和 Q 交叉熵的最小值:

$$\begin{aligned}&\min_{P}\sum_{\xi\in D}P(\xi)\ln\frac{P(\xi)}{Q(\xi)}\\&\text{s.t. }\forall i\in\{1,2,\cdots,k\}:\left|\sum_{\xi\in D}P(\tau)f_i^{\xi}-\hat{f}_i\right|\leqslant\epsilon_i\\&\sum_{\xi\in D}P(\xi)=1\\&\forall\xi\in D:P(\xi)\geqslant 0\end{aligned} \tag{38}$$

其中 ϵ_i 为阈值,可以使用Hoeffding上限确定[36]。同样,利用拉格朗日乘子法和KKT条件,可以得到相对熵最大的解:

$$P(\xi|\theta)=\frac{1}{Z(\theta)}Q(\xi)\exp(\sum_{i=1}^{k}\theta_i f_i^{\xi}) \tag{39}$$

跟最大熵IRL方法相同,参数的求解过程利用次梯度的方法:

$$\nabla L(\theta)=\tilde{f}_i-\sum_{\xi\in D}P(\xi|\theta)f_i^{\tau}-\alpha_i\epsilon_i \tag{40}$$

在利用次梯度的方法进行参数求解时,最关键的问题是估计式(40)中的概率 $P(\xi|\theta)$。由最大相对熵的求解可以得到该概率的计算公式,如式(39)。将 Q 显示表述出来,由定义知道,它是在策略为基准策略时得到的轨迹分布,因此可将其分解为:

$$Q(\xi)=D(\xi)U(\xi) \tag{41}$$

$$D(\xi)=d_0(s_1)\prod_{t=1}^{H}T(s_{t+1}|s_t,a_t) \tag{42}$$

$$U(\xi)=\frac{1}{|A|^H} \tag{43}$$

将式(41)带入到式(39),可以得到最大相对熵解为:

$$P(\tau|\theta)=\frac{D(\tau)U(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^{\tau})}{\sum_{\tau\in T}D(\tau)U(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^{\tau})} \tag{44}$$

这时,再利用重要性采样对式(44)进行估计,得到次梯度为:

$$\begin{aligned}\nabla L(\theta)=&\tilde{f}_i-\frac{1}{N}\sum_{\xi\in D}\frac{P(\xi|\theta)}{D(\xi)\pi(\xi)}f_i^{\xi}-\alpha_i\epsilon_i=\\&\tilde{f}_i-\frac{\sum_{\xi\in D}\frac{U(\xi)}{\pi(\xi)}\exp\left(\sum_{j=1}^{k}\theta_j f_j^{\xi}\right)f_j^{\xi}}{\sum_{\xi\in D}\frac{U(\xi)}{\pi(\xi)}\exp\left(\sum_{j=1}^{k}\theta_j f_j^{\xi}\right)}-\alpha_i\epsilon_i\end{aligned} \tag{45}$$

其中 $\pi(\xi)=\prod_{t=1}^{H}P_{\xi}(a_t|s_t)$。计算过程见文献[36]。

5 深度逆向强化学习研究进展

IRL的目标是基于观察和对环境模型推断生成潜在的回报函数的结构,引导agent的行为。这种对回报函数建模的方式提供了一种让agent模仿演示者的具体行为的方法[32]。目前的方法大多基于预先确定的回报函数的参数化特征。为了实现特征函数更好的泛化性能,Abbeel、Ratliff、Ziebart等人[7,32,34]采用加权线性组合回报函数的特征。

为了克服线性模型固有的局限性,Choi等人[37]通过学习一套对原子特征进行逻辑连接的非线性回报函数,采用非参数方法,如高斯过程(Gaussian Processes,GPs)来满足潜在的复杂的、非线性的回报函数[38]。虽然这种方法将非线性近似方法扩展到了回报函数的逼近上,增强了IRL的灵活性。但是,使用这种方法倾向于需要大量的奖励样本,以逼近复杂的回报函数[39-41]。甚至如文献[38]所述的稀疏的高斯过程也会使得算法的时间复杂性依赖于动作集或经验状态奖励对的数量。并且回报函数的复杂性使得对于诱导点数量的需求急剧增加,使得这种非参数化的方法在计算上变得不可行。而采用端到端的学习方式从原始输入直接映射到回报值,而无需对输入表示进行压缩或预处理。

但是学徒学习、最大边际法、最大熵、交叉熵等传统方法不能很好地扩展到具有大量状态的系统[6-7,33,37]。因此,将这些算法与深度学习相结合,在神经网络中学习状态动作对的回报,系统可以根据需要复杂或者很大。

将深度学习与IRL相结合,开辟了一种新的利用环境和状态特征的复杂相关性来学习复杂的任务方法。

5.1　基于最大边际法的深度逆向强化学习

IRL要学习的是回报函数,以便避免人为设定回报函数。但是,在进行学习回报函数的时候又引入了需要人为指定的特征函数,即之前已经假设了回报函数的形式为[40]:

$$R_\theta(s)=\theta^{\mathrm{T}}\phi(s) \tag{46}$$

其中 $\phi(s)$ 是人为指定的特征函数。对于大规模问题,人为指定的特征函数表示能力不足,只能覆盖到部分回报函数形式,也难以泛化到其他状态空间。其中一种解决方法是利用神经网络表示回报函数的基底。这时,回报函数可表示为: $r(s)=\theta^{\mathrm{T}}f(s)$,其中 $f(s)$ 为神经网络,如图4所示。

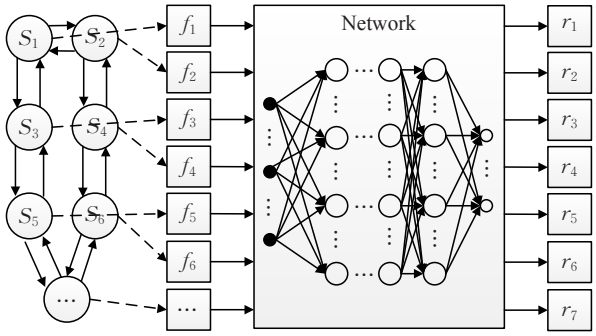

图4　基于深度神经网络的回报函数状态特征表示结构

神经逆向强化学习的整个框架仍然是最大边际法的框架,因此问题形式化为:

$$\begin{aligned}&\min_\theta\frac{1}{2}\|\theta\|_2^2+C\sum_{i=1}^{N_\tau}\xi^i\\&\text{s.t. }Q_\theta^{\pi_E}\left(s_t^{(i)},a_t^{(i)}\right)+\xi^{(i)}\geqslant\\&\qquad\max_\pi Q^\pi\left(s_t^{(i)},a_t^{(i)}\right)+l(s_t^{(i)},a_t^{(i)})\end{aligned} \tag{47}$$

其中 $Q^\pi\left(s_t^{(i)},a_t^{(i)}\right)$ 是智能体在状态 $s_t^{(i)}$ 时的 Q 值, $Q_\theta^{\pi_E}\left(s_t^{(i)},a_t^{(i)}\right)$ 是专家策略的 Q 值。如果学习到的状态动作对与专家策略一致,那么损失函数 $l(s,a)=0$,否则 $l(s,a)=1$ 。

给每个专家示例轨迹设置一个松弛变量 ξ ,以便约束违规行为的惩罚。因此,通过最小化目标函数来简化优化问题:

$$\begin{aligned}J(\theta)=&\sum_{i=1}^{N_\tau}\sum_{t=1}^{L_i}\max_\pi\left(Q^\pi\left(s_t^{(i)},a_t^{(i)}\right)+l\left(s_t^{(i)},a_t^{(i)}\right)\right)-\\&Q_\theta^{\pi_E}\left(s_t^{(i)},a_t^{(i)}\right)+\frac{\lambda_1}{2}\|\theta\|_2^2\end{aligned} \tag{48}$$

$\lambda_1\geqslant0$ 是一个用于平衡惩罚和期望的经验常数。 $J(\theta)$ 可以通过梯度下降法优化:

$$\theta\leftarrow\theta-\sigma_1\frac{\partial J(\theta)}{\partial\theta} \tag{49}$$

其中 $\sigma_1\in[0,1]$ 为步长。在计算出 θ 后,就可以使用公式 $R_\theta(s)=\theta^{\mathrm{T}}\phi(s)$ 计算回报函数了。

5.2　基于深度Q网络的深度学徒学习

基于深度Q网络的学徒学习架构由两部分构成:深度学徒Q网络(DAQN)和深度学徒回报网络(DARN)[42-43]。

5.2.1　深度学徒Q网络

DAQN用来学习回报函数,估计在某个状态下动作的回报值。该结构输出每个可能的动作的softmax预测。因此,对于每个状态,网络预测要采取的下一个动作。这个网络的损失函数为:

$$J(w)=\sum_a[q_w(s,a)-\hat{q}(s,a)]^2 \tag{50}$$

w 为学到的权重, $Q_w(s,a)$ 是DAQN的输出, $\hat{q}(s,a)$ 是专家实际采取的动作,使用一个one-hot数组表示。因此,对于输入 $(s=s_t,a=a_t\in D_E)$,如果 $a=a_t$,则数组 $\hat{q}$ 等于1。

5.2.2　深度学徒回报网络

一旦DAQN训练好了,DARN用来从DAQN学到的专家策略中解析回报函数。DARN与DAQN具有相同的网络结构。然而,它在损失函数中使用的是从DAQN中的soft-max函数输出。DARN的输入是 (s,a,s') 对,其损失函数L2范数形式是:

$$J(w)=\left\|r_w(s,a)-\hat{r}(s,a)\right\|_2 \tag{51}$$

w 是学习到的权重, $r_w(s,a)$ 是DARN的输出, $\hat{r}(s,a)$ 是从DAQN学习到的 (s,a) 的值函数。与贝尔曼方程相比, $r_w(s,a)$ 是状态动作对 (s,a) 的目标值和 $\hat{r}(s,a)$ 从专家策略学到的 (s,a) 的值:

$$\hat{r}(s,a)=DAQN^{\mathrm{PS}}(s,a)-\gamma\max_{a'}DAQN^{\mathrm{PS}}(s',a') \tag{52}$$

$DAQN^{\mathrm{PS}}(s,a)$ 是输入为 (s,a) 时DAQN的presoft值, $\max_{a'}DAQN^{\mathrm{PS}}(s',a')$ 是DAQN在输入为 (s',a') 时的最大的presoft值, γ 为折扣因子。因此,DAQN扩展的损失函数为:

$$J_r(w)=\left\|r_w(s,a)-(DAQN^{\mathrm{PS}}(s',a'))\right\|_2 \tag{53}$$

DARN使用从专家数据中提取的独立的状态到状态的transitions数据集。神经网络的泛化能力使得即使某个状态专家数据中没有,也依然可以通过该网络产生输出。

5.3　基于最大熵模型的深度逆向强化学习

传统的最大熵的IRL由于表征能力的局限性,只能够用在规模小、离散的任务上[32],采用深层神经网络架构近似回报函数的IRL方法通过分层结构中的许多非线性结果的组合和重用,使得其具备对高度非线性函数的表征能力[39]。此外,DNNs提供良好的相对于示例演示的计算复杂度 $(O(1))$, 使得它很容易扩展到大状态空间和复杂的回报函数。

解决IRL问题可以限定在贝叶斯推理的最大后验

概率,在给定回报函数结构和参数 θ 时,最大限度地提高观察专家示范 D 的联合后验分布[40]:

$$L(\theta)=\lg P(D,\theta|r)=\lg P(D|r)+\lg P(\theta) \tag{54}$$

定义:$L_D=\lg P(D|r), L_\theta=\lg P(\theta)$。

此联合对数似然相对于网络参数 θ 可微,因此,可以使用梯度下降方法进行优化。式(54)的数据项 L_D 所给出的基于最大熵的目标函数相对于奖赏 r 是可微的,因此可以使目标梯度的反向传播到网络的权重。最后的梯度是由相对于 θ 的数据项和 L_D 和模型项 L_θ 梯度的总和。

$$\frac{\partial L}{\partial\theta}=\frac{\partial L_D}{\partial\theta}+\frac{\partial L_\theta}{\partial\theta} \tag{55}$$

数据项的梯度可以用专家演示对奖励的导数,以及这些奖励对网络权值 θ 的导数来表示,如下式:

$$\frac{\partial L_D}{\partial\theta}=\frac{\partial L_D}{\partial r}\cdot\frac{\partial r}{\partial\theta}=(\mu_D-E[\mu])\cdot\frac{\partial}{\partial\theta}g(f,\theta) \tag{56}$$

在上式中 $r=g(f,\theta)$,专家演示轨迹的梯度 L_D 对状态的奖励 r 的导数,等于专家演示轨迹的状态访问次数和学习系统轨迹分布的期望访问数的差值,依赖于由相应的最优策略给出的回报函数的近似。

$$E[\mu]=\sum_{\xi:(s,a)\in\xi}P(\xi|r) \tag{57}$$

$E[\mu]$ 通常涉及了许多可能的轨迹指数求和。Ziebart等人[34]提出了一种基于动态规划的有效算法,能够在多项式时间内计算这个量。$\frac{\partial L_D}{\partial\theta}$ 梯度首先使用这个算法计算访问次数的差异,然后将这个值作为误差信号使用反向传播方式传递给网络。该算法用线性最大熵公式给出的式(54)(55)的损失和梯度求导。用来计算loss的专家的状态动作频率 μ_D^a,将动作的频率进行加和就得到了专家状态的频率 $\mu_D=\sum_{a=1}^{A}\mu_D^a$。

带有网络参数的模型项 L_θ 的导数表示为一个正则化项,L1、L2正则化和dropout方法都可以用来防止深层神经网络的过拟合。这可以看作是一个子模型平均法或打破原始训练数据的方法,例如,通过添加噪声训练模型的不变性。

5.4 从非专家示例轨迹中进行逆向强化学习

现有的IRL算法只有从成功的示例轨迹数据中学习,即在执行任务的过程中收集专家数据。然而在许多实际任务中,相比成功的专家演示轨迹,失败的示例轨迹更容易获取。在专家示例稀缺的情况下,如果任务可以在模拟的环境下进行交互,非专家往往可以轻松地演示多种故障模式,产生的数据以补充稀缺的成功示例。文献[44]探索了这种方法,在那里机器人学会避免人们使用失败逃避的模拟。最后,失败的示例轨迹可能被用来探索状态空间,这也是一个从演示中学习回报函数和策略的思路。

IRLF(Inverse Reinforcement Learning from Failure)适用于学徒可以访问失败的示例数据集 F 和专家数据集 E[45]。虽然回报函数最初未知,但回报函数能够被参数化为 K 个特征函数 $\phi_k(s,a)$ 的和:

$$R(s,a)=\sum_{k=1}^{K}\theta_k\phi_k(s,a) \tag{58}$$

由于 θ 独立于状态和动作,因此值函数定义为:

$$V^\pi(s)=E\{\sum_{k=1}^{k}\theta_k\sum_{t=1}^{h}\phi_k(s_t,a_t)|s_1=s\}=\sum_{k=1}^{k}\theta_k\mu_k^{\pi,1:h}|s_1=s \tag{59}$$

智能体从专家示例中学到回报函数,使得在该回报函数下所得到的最优策略在专家示例策略附近。未知的回报函数 $R(s)$ 一般都是状态的函数,因为它是未知的,所以可以利用函数逼近的方法对其进行参数逼近,其逼近形式可设为:$R(s,a)=\theta^{\mathrm{T}}\cdot\phi(s,a)$,其中 $\phi(s,a)$ 为基函数,可以为多项式基底,也可以为傅里叶基底。逆向强化学习求的是回报函数中的系数 θ。根据值函数的定义,策略 π 的值函数为:

$$E_{s_0\sim D}\left[V^\pi(s_0)\right]=E\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)|\pi\right]=E\left[\sum_{t=0}^{\infty}\gamma^t\theta\cdot\phi(s_t,a_t)|\pi\right]=\theta\cdot E\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t,a_t)|\pi\right] \tag{60}$$

定义特征期望为:$\mu^\pi=E\left[\sum_{t=0}^{\infty}\gamma^t\phi(s_t)|\pi\right]$。需要注意的是,特征期望跟策略 π 有关,策略不同时,策略期望也不相同。之后,值函数可以写为:

$$E_{s_0\sim D}\left[V^\pi(s_0)\right]=\theta^{\mathrm{T}}\cdot\mu^\pi(s_0) \tag{61}$$

学习者从 m 个轨迹的数据集学习,$D=\{\tau_1,\tau_2,\cdots,\tau_n\}$。$\tau_i=\left\{\left(s_1^{\tau_1},a_1^{\tau_1}\right),\left(s_2^{\tau_1},a_2^{\tau_1}\right),\cdots,\right\}$ 是由专家生成的长度 h 的状态动作序列。当给定 m 条专家示例轨迹后,根据定义可以估计专家策略的特征期望为:

$$\tilde{\mu}_E=\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{\infty}\gamma^t\phi(s_t^{(i)}) \tag{62}$$

在策略 π 时,$s_1=s$ 的情况下,在步骤1和 h 之间的特征 ϕ_k 的累计。只考虑步长 h 内的累计回报,h 之后的省略。

$$\tilde{\mu}_k^D=\frac{1}{N}\sum_{\tau\in D}\sum_{t=1}^{h}\phi(s_t^\tau,a_t^\tau) \tag{63}$$

$\tilde{\mu}_k^D$ 是状态独立的,隐含地估计了专家初始状态的期望值。为比较策略 π 与 $\tilde{\mu}_k^D$ 的特征期望,可以通过边缘化 s_1 获得 π 的特征期望的类似状态独立度量:

$$\mu_k^\pi|_D=\sum_{s\in S}P_D(s_1=s)\mu_k^\pi|_{s_1=s} \tag{64}$$

$P_D(s_1=s)=N_1(s)/N$ 是专家初始状态分布的最大似然估计,$N_1(s)$ 是专家轨迹中 $s_1=s$ 的轨迹的数量。学习者的目标是找到 θ,使 $\tilde{\boldsymbol{\mu}}^D=[\tilde{\mu}_1^D,\tilde{\mu}_2^D,\cdots,\tilde{\mu}_k^D]$ 和向量

$\mu^{\pi}|D=[\mu_1^{\pi}|_D,\mu_2^{\pi}|_D,\cdots,\mu_k^{\pi}|_D]$ 根据某种度量的距离最小，同时也泛化到不可预见的初始条件。使得集合 F 的特征期望与最优策略尽可能远的直接方法是在优化问题中加入不等式约束：

$$|\tilde{\mu}_k^F-\tilde{\mu}_k^{\pi}|_F>a_k \quad \forall k \tag{65}$$

$\tilde{\mu}_k^F$ 是根据 F 得到的特征 K 的经验期望，计算类似于式(64)，a_k 是一个将会被最大化的变量，它是优化目标的一部分，增加参数向量 $\theta=[\theta_1,\theta_2,\cdots,\theta_K]^{\mathrm{T}}$ 和特征期望的差异到目标函数中，得到目标函数：

$$\max_{\pi,a}H(A^h\|S^h)+\sum_{k=1}^{K}\theta_k(\mu_k^{\pi}|_F-\tilde{\mu}_k^F) \tag{66}$$

新的等式约束形式 $\mu_k^{\pi}|_F-\tilde{\mu}_k^F=z_k$ 与 $z_k\in R$，通过引入变量 z_k 作为优化目标中的一项，最大化最优策略生成的轨迹的特征期望与失败的轨迹的特征期望差值。因此，完全约束优化转化为：

$$\begin{aligned}&\max_{\pi,\theta,z}H(A^h\|S^h)+\sum_{k=1}^{K}\theta_k z_k-\frac{\lambda}{2}\|\theta\|^2\\&\text{subject to: }\mu_k^{\pi}|_D=\tilde{\mu}_k^{\pi}\ \forall k\\&\text{and: }\mu_k^{\pi}|_F-\tilde{\mu}_k^F=z_k\ \forall k\\&\text{and: }\sum_{a\in A}\pi(s,a)=1\ \forall s\in S\\&\text{and: }\pi(s,a)\geqslant 0\ \forall s\in S,a\in A\end{aligned} \tag{67}$$

除此之外Finn C等人[46]提出了引导损失函数的IRL方法，将损失函数作为优化目标，生成最接近专家示例轨迹数据的损失函数。Jonathan H，Finn C等人[47-48]提出，在生成器的概率密度能够估计的前提下，GAN中生成器的结果等价于一个基于采样的最大熵的逆向强化学习，并以此为基础构建了基于能量函数的模型，通过GAN训练得到损失函数。

6　算法分析与总结

人类的智能更多地体现在依靠较少的数据完成复杂的任务。因此，人工智能的发展过程中，如何通过更少的数据进行学习是至关重要的。而IRL方法提供了一种小数据驱动的问题求解方法。为人工智能从“大数据，小任务”到“小数据，大任务”的模式转变提供了思路。然而，尽管IRL理论、算法和应用研究在国内外已普遍开展，并且也已经取得了较多的研究成果，但仍然有许多问题还亟待解决。

6.1　深度逆向强化学习算法理论分析与对比

在算法和理论方面，传统的IRL方法已经建立了较为完善的收敛理论，但是面对连续、高维的马氏决策问题面临的维数灾难和组合爆炸问题无能为力，因此在实际问题的解决上进展不大。得益于深度神经网络强大的端到端学习能力，使用深度神经网络表示回报函数，避免了特征提取工作，在游戏、机器人控制等领域取得了比较好的效果。但是，目前已经提出的IRL方法仍然存在学习效率不高、回报函数质量难以评价等缺点，有待进行更加深入的研究，以扩大深度IRL在实际问题中的应用。

IRL发展的初级阶段，其算法实验场景应用于Mountain Car、Grid World、Objectworld等经典控制问题的解决[1]，随着其相关理论和技术的发展，IRL在自动驾驶[7,38]、路径导航[8,35,39-40]、机器人控制[8,47]等领域都取得了一定的成功。

从深度IRL的发展来看，求解算法向复杂度优化和实用化方向发展。具体体现在以下几方面：一是消除了算法求解目标的歧义性；基于最大边际法的深度IRL和基于深度Q网络的IRL无法保证求解出的回报函数唯一性，并且由于需要求解MDP模型，导致算法的复杂性高；后续的基于最大熵的IRL算法和IRLF通过最大熵和交叉熵模型，引导损失函数生成的IRL和基于GAN的IRL通过基于能量的模型的方法解决了回报函数歧义性的问题；二是无须通过求解MDP的方式获取回报函数，提升了算法的运行效率。而IRLF算法以失败的轨迹数据尽可能远离最优策略为约束条件达到了利用失败的轨迹数据探索状态空间的目的。提升了IRL算法的适用范围和效果。深度逆向强化学习算法性能对比如表1所示。

6.2　深度逆向强化学习的应用展望

在应用方面，实现逆向强化学习在复杂、不确定系统中的优化控制问题，对于推动机器人控制、无人飞行器控制、军事决策等各领域的发展有重要的意义，特别是对于多智能体对抗系统来说，逆向强化学习是实现对

表1　深度逆向强化学习算法性能对比

算法	数据应用	计算复杂度	歧义性	算法目标
Deep MMP	专家轨迹数据	需求解MDP，$O((m\cdot n^p)^k)$	有	回报函数 R
Deep AL	专家轨迹数据	需求解MDP，$O((m+1)\cdot n^p)$	有	回报函数 R
Deep ME IRL	专家轨迹数据	不求解MDP，已知状态转移概率 $O(n^p)$	无	回报函数 R
IRLF	专家和非专家轨迹数据	不求解MDP，无须知道状态转移概率 $O(n^p)$	无	回报函数 R 和价值函数 Q
引导损失函数生成方法	专家轨迹数据	不求解MDP，无须知道状态转移概率 $O(n^p)$	无	损失函数 C
基于生成式对抗网络的逆向强化学习	专家轨迹数据	不求解MDP，无须知道状态转移概率 $O(n^p)$	无	生成式网络 G 和判别网络 D

注：计算复杂度时，m 为产生的训练样本数量，n 为神经元个数，p 为神经网络层数，k 为迭代次数。

专家示例数据高效利用的有效手段,为解决智能agent的知识获取这个瓶颈问题提供了一个可行之法。

参考文献:

[1] Sutton R,Barto A.Reinforcement learning:An introduction[M].[S.l.]:MIT Press,1998.

[2] Deng L,Yu D.Deep learning:Methods and applications[J].Foundations & Trends in Signal Processing,2013,7(3):197-387.

[3] 周志华.机器学习[M].北京:清华大学出版社,2015:390-392.

[4] Silver D.Tutorial:Deep reinforcement learning[R].2016.

[5] 李晨溪,曹雷,张永亮,等.基于知识的深度强化学习研究综述[J].系统工程与电子技术,2017,39(11):2603-2613.

[6] Ng A Y,Russell S J.Algorithms for inverse reinforcement learning[C]//Seventeenth International Conference on Machine Learning,2000:663-670.

[7] Abbeel P,Ng A Y.Apprenticeship learning via inverse reinforcement learning[C]//International Conference on Machine Learning,2004:1-8.

[8] Whiteson S,Stone P.Evolutionary function approximation for reinforcement learning[J].Journal of Machine Learning Research,2006(7):877-917.

[9] Preux P,Girgin S,Loth M.Feature discovery in approximate dynamic programing[C]//Proceedings IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning,2009:109-116.

[10] Lange S,Riedmiller M.Deep auto-encoder neural networks in reinforcement learning[C]//Proceedings of the 2010 International Joint Conference on Neural Networks(IJCNN),2010:1-8.

[11] Mnih V,Kavukcuoglu K,Silver D,et al.Playing atari with deep reinforcement learning[C]//Proceedings of Workshops at the 26th Neural Information Processing Systems 2013,Lake Tahoe,USA,2013:201-220.

[12] 刘全,翟建伟,章宗长,等.深度强化学习综述[J].计算机学报,2017,40(1):1-28.

[13] Mnih V,Kavukcuoglu K,Silver D,et al.Human-level control through deep reinforcement learning[J].Nature,2015,518(7540):29-33.

[14] Schaul T,Quan J,Antonoglou I,et al.Prioritized experience replay[EB/OL].[2017-11-16].https://arxiv.org/abs/1511.05952.

[15] Mnih V,Badia A P,Mirza M,et al.Asynchronous methods for deep reinforcement learning[EB/OL].[2017-11-16].https://arxiv.org/abs/1602.01783.

[16] Van H.Double Q-learning[C]//Advances in Neural Information Processing Systems,2010:2613-2621.

[17] Van H V,Guez A,Silver D.Deep reinforcement learning with double q-learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence,Phoenix,USA,2016:2094-2100.

[18] Wang Z,Freitas N D,Lanctot M.Dueling network architectures for deep reinforcement learning[C]//Proceedings of the International Conference on Machine Learning,New York,USA,2016:1995-2003.

[19] Graves A,Mohamed A,Hinton G.Speech recognition with deep recurrent neural networks[C]//Proceedings of the IEEE Conference on Acoustics,Speech and Signal Processing,Vancouver,Canada,2013:6645-6649.

[20] Osband I,Russo D,Wen Z,et al.Deep exploration via randomized value functions[EB/OL].[2017-11-16].https://arxiv.org/abs/1703.07608.

[21] Silver D,Lever G,Heess N,et al.Deterministic policy gradient algorithms[C]//Proceedings of the International Conference on Machine Learning,Beijing,China,2014:387-395.

[22] Lillicrap T P,Hunt J J,Pritzel A,et al.Continuous control with deep reinforcement learning[EB/OL].[2017-11-16].https://arxiv.org/abs/1509.02971.

[23] Levine S,Koltun V.Guided policy search[C]//International Conference on Machine Learning,2013:1-9.

[24] Schulman J,Levine S,Abbeel P,et al.Trust region policy optimization[C]//ICML 2015,2015:1889-1897.

[25] Heess N,Wayne G,Silver D,et al.Learning continuous control policies by stochastic value gradients[C]//NIPS 2015,2015:2926-2934.

[26] Schulman J,Moritz P,Levine S,et al.High dimensional continuous control using generalized advantage estimation[J].arXiv preprint arXiv:1506.02438,2015.

[27] Gu Shixiang,Lillicrap T,Sutskever I,et al.Continuous deep q-learning with model-based acceleration[J].arXiv preprint arXiv:1603.00748,2016.

[28] Sutton R S.Dyna,an integrated architecture for learning,planning,and reacting[J].ACM Sigart Bulletin,1991,2(4):160-163.

[29] Li Weiwei,Todorov E.Iterative linear quadratic regulator design for nonlinear biological movement systems[C]//ICINCO 2004,2004:222-229.

[30] Li Chenxi,Cao L,Liu X,et al.A study of qualitative knowledge-based exploration for continuous deep reinforcement learning[J].IEICE Transactions on Information & Systems,2017,E100.D(11):2721-2724.

[31] Tsochantaridis I,Joachims T,Hofmann T,et al.Large margin methods for structured and interdependent output variables[J].Journal of Machine Learning Research,2005,9(6):1453-1484.

[32] Ratliff N D,Bagnell J A,Zinkevich M A.Maximum margin planning[C]//Proceedings of the 23rd International Conference on Machine Learning,2006:729-736.

[33] Klein E,Geist M,Piot B,et al.Inverse reinforcement learning through structured classification[C]//Advances in Neural Information Processing Systems,2012:1007-1015.

[34] Ziebart B,Maas A,Bagnell A,et al.Maximum entropy inverse reinforcement learning[C]//Proceedings of the Twenty-third AAAI Conference on Articial Intelligence,2008:1433-1438.

[35] Aghasadeghi N,Bretl T.Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals[C]//International Conference on Intelligent Robots and Systems,2011:1561-1566.

[36] Boularias A,Kober J,Peters J.Relative entropy inverse reinforcement learning[C]//Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics,2011:182-189.

[37] Choi J,Kim K E.Bayesian nonparametric feature construction for inverse reinforcement learning[C]//Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence,2013:1287-1293.

[38] Levine S,Popovic Z,Koltun V.Nonlinear inverse reinforcement learning with gaussian processes[C]//Advances in Neural Information Processing Systems,2011:19-27.

[39] Chen X,Kamel A E.Neural inverse reinforcement learning in autonomous navigation[J].Robotics & Autonomous Systems,2016,84:1-14.

[40] Wulfmeier M,Ondruska P,Posner I.Maximum entropy deep inverse reinforcement learning[EB/OL].[2017-11-16].https://arxiv.org/abs/1507.04888.

[41] Bengio Y,LeCun Y.Scaling learning algorithms towards AI[J].Large-Scale Kernel Machines,2007,34(5).

[42] Bogdanovic M,Markovikj D,Denil M,et al.Deep apprenticeship learning for playing video games[J].European Judaism,2015,39(1):44-48.

[43] Todd H,Matej V,Olivier P,et al.Deep Q-learning from demonstrations[EB/OL].[2017-11-16].https://arxiv.org/abs/1704.03732.

[44] Choi S,Kim E,Lee K,et al.Leveraged non-stationary Gaussian process regression for autonomous robot navigation[C]//2015 IEEE International Conference on Robotics and Automation(ICRA),2015:473-478.

[45] Whiteson S,Whiteson S,Whiteson S.Inverse reinforcement learning from failure[C]//International Conference on Autonomous Agents & Multiagent Systems,2016:1060-1068.

[46] Finn C,Levine S,Abbeel P.Guided cost learning:deep inverse optimal control via policy optimization[C]//International Conference on Machine Learning,2016:49-58.

[47] Jonathan H,Stenfano E.Generative adversarial imitation learning[EB/OL].[2017-11-16].https://arxiv.org/abs/1606.03476.

[48] Finn C,Christiano P,Abbeel P,et al.A connection between generative adversarial networks,inverse reinforcement learning,and energy-based models[EB/OL].[2017-11-16].https://arxiv.org/abs/1611.03852.

(上接23页)

[36] Gheyas I A,Smith L S.Feature subset selection in large dimensionality domains[J].Pattern Recognition,2010.

[37] 徐仙,卢先领,王洪斌.行为识别中基于GA优化的加速度特征选择方法[J].计算机工程与应用,2016,52(6):139-143.

[38] Cheng M,Zhang G,Mitra N,et al.Global contrast based salient region detection[C]//Proceedings of IEEE Conference on Computer Visual and Pattern Recognition,2011:409-416.

[39] Liu Z,Xue Y,Yan H,et al.Efficient saliency detection based on Gaussian models[J].IET Image Processing,2013,5(2):122-131.

[40] Shi Q,Cheng L,Wang L,et al.Human action segmentation and recognition using discriminative semi-markov models[J].International Journal of Computer Vision,2011,93(1):22-32.

[41] 何文锋.基于WIFI的手势识别研究[D].广东深圳:深圳大学,2015.

[42] Krizhevsky A,Sutskever I,Hinton G E.ImageNet classification with deep convolutional neural networks[J].Communications of the ACM,2012,60(2).

[43] Wang X,Gao L,Mao S,et al.CSI-based fingerprinting for indoor localization:A deep learning approach[J].IEEE Transactions on Vehicular Technology,2017,66(1):763-776.

[44] Wang X,Gao L,Mao S,et al.DeepFi:Deep learning for indoor fingerprinting using channel state information[C]//Wireless Communications and Networking Conference,New Orleans,LA,USA,2015:1666-1671.

[45] Zhu D,Pang N,Li G,et al.WiseFi:Activity localization and recognition on commodity off-the-shelf WiFi devices[C]//IEEE 18th International Conference on High Performance Computing and Communications;IEEE 14th International Conference on Smart City;IEEE 2nd International Conference on Data Science and Systems,2016:562-569.

[46] Gao Q,Wang J,Ma X,et al.CSI-based device-free wireless localization and activity recognition using radio image features[J].IEEE Transactions on Vehicular Technology,2017,66(11):10346-10356.

[47] Ji S,Yang M,Yu K.3D convolutional neural networks for human action recognition[J].IEEE Transactions on Pattern Analysis & Machine Intelligence,2013,35(1):221-231.

[48] Ali K H,Wang T.Learning features for action recognition and identity with deep belief networks[C]//International Conference on Audio,Language and Image Processing,Shanghai,China,2015:129-132.