

基于凸多面体抽象域的自适应强化学习技术研究

陈冬火¹⁾ 刘 全^{1),2)} 朱 斐¹⁾ 金海东¹⁾

¹⁾(苏州大学计算机科学与技术学院 江苏 苏州 215006)

²⁾(符号计算与知识工程教育部重点实验室(吉林大学) 长春 130012)

摘 要 表格驱动算法是解决强化学习问题的一类重要方法,但由于“维数灾”现象的存在,这种方法不能直接应用于解决具有连续状态空间的强化学习问题.解决维数灾问题的方法主要包括两种:状态空间的离散化和函数近似方法.相比函数近似,基于连续状态空间离散化的表格驱动方法具有原理直观、程序结构简单和计算轻量化的特点.基于连续状态空间离散化方法的关键是发现合适的状态空间离散化机制,平衡计算量及准确性,并且确保基于离散抽象状态空间的数值性度量,例如 V 值函数和 Q 值函数,可以较为准确地对原始强化学习问题进行策略评估和最优策略 π^* 计算.文中提出一种基于凸多面体抽象域的自适应状态空间离散化方法,实现自适应的基于凸多面体抽象域的 $Q(\lambda)$ 强化学习算法(Adaptive Polyhedra Domain based $Q(\lambda)$, APDQ(λ)).凸多面体是一种抽象状态的表达方法,广泛应用于各种随机系统性能评估和程序数值性属性的验证.这种方法通过抽象函数,建立具体状态空间至多面体域的抽象状态空间的映射,把连续状态空间最优策略的计算问题转化为有限大小的和易于处理的抽象状态空间最优策略的计算问题.根据与抽象状态相关的样本集信息,设计了包括 *BoxRefinement*、*LFRefinement* 和 *MVLFRefinement* 多种自适应精化机制.依据这些精化机制,对抽象状态空间持续进行适应性精化,从而优化具体状态空间的离散化机制,产生符合在线抽样样本空间所蕴涵的统计奖赏模型.基于多面体专业计算库 PPL(Parma Polyhedra Library)和高精度数值计算库 GMP(GNU Multiple Precision)实现了算法 APDQ(λ),并实施了实例研究.选择典型的连续状态空间强化学习问题山地车(Mountain Car, MC)和杂技机器人(Acrobatic robot, Acrobot)作为实验对象,详细评估了各种强化学习参数和自适应精化相关的阈值参数对 APDQ(λ)性能的影响,探究了抽象状态空间动态变化情况下各种参数在策略优化过程中的作用机理.实验结果显示当折扣率 γ 大于 0.7 时,算法展现出较好的综合性能,在初期,策略都快速地改进,后面的阶段平缓地趋向收敛(如图 6~图 13 所示),并且对学习率 α 和各种抽象状态空间精化参数都具有较好的适应性;当折扣率 γ 小于 0.6 时,算法的性能衰退较快.抽象解释技术用于统计学习过程是一种较好的解决连续强化学习问题的思想,有许多问题值得进一步研究和探讨,例如基于近似模型的采样和值函数更新等问题.

关键词 强化学习;凸多面体抽象域;连续状态空间; $Q(\lambda)$; 自适应精化

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2018.00112

The Research on Adaptive Reinforcement Learning Technique Based on Convex Polyhedra Abstraction Domain

CHEN Dong-Huo¹⁾ LIU Quan^{1),2)} ZHU Fei¹⁾ JIN Hai-Dong¹⁾

¹⁾(Institute of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006)

²⁾(Key Laboratory of Symbolic Computation and Knowledge Engineering (Jinlin University), Ministry of Education, Changchun 130012)

Abstract The table-driven based algorithm is an important method for solving the reinforcement learning problems, but for the real world problems with continuous state spaces, the method is

收稿日期:2016-04-17;在线出版日期:2017-03-20. 本课题得到国家自然科学基金项目(61272005, 61303108, 61373094, 61472262, 61502323, 61502329)、江苏省自然科学基金项目(BK2012616)、江苏省高校自然科学研究项目(13KJB520020)、吉林大学符号计算与知识工程教育部重点实验室项目(93K172014K04)、苏州市应用基础研究计划项目(SYG201422)、苏州大学高校省级重点实验室基金项目(KJS1524)、中国国家留学基金项目(201606920013)、浙江省自然科学基金(LY16F010019)资助. 陈冬火,男,1974年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为程序验证、模型检验、机器学习. E-mail: dhchen@suda.edu.cn. 刘 全,男,1969年生,博士,教授,博士生导师,中国计算机(CCF)学会会员,主要研究领域为智能信息处理、自动推理、机器学习. 朱 斐,男,1978年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为机器学习、文本挖掘、模式识别. 金海东,男,1973年生,硕士,讲师,主要研究方向为强化学习和深度学习.

challenged by the curse of dimensionality, also named as the state explosion problem. Two methods have been presented for attacking the curse of dimensionality, including discretization of continuous state space and function approximation. For the usage of the discretization of continuous state space, the table-driven based algorithm is of some advantages than the function approximation based algorithms, namely straightforward principle, the implementation with concise data structure and the lightweight computation. Note that the core of algorithm is to discover a qualified discretization mechanism with which the computation cost and the accuracy of the abstract model are well balanced, and the optimal policy of an original reinforcement learning problem can be approximately derived according to its abstract state space and quantitative reward metrics. This paper presents an adaptive discretization technique based on the convex polyhedra abstraction domain, and designs an adaptive polyhedra domain based $Q(\lambda)$ algorithm (APDQ(λ)) on the basis of $Q(\lambda)$, an important algorithm in reinforcement learning. Convex polyhedron is a qualified representation of abstract state, which is widely in performance evaluation of complex stochastic systems and verification of numerical properties of programs. The method abstracts a continuous (infinite) concrete state space into a discrete and manageable set of abstract states by defining an abstract function, such that the control problem of the original system can be resolved directly by the corresponding abstract system. Especially, some adaptive refinement operators, such as *BoxRefinement*, *LFRefinement* and *MVLFRefinement*, are studied, which are dependent on the online samples information for a refined abstract polyhedron state. The abstract state space is dynamically adjusted, such that a finite and discrete model is statistically derived according to online samples, which approximates the dynamic and reward model of continuous Markov system. Finally, APDQ(λ) is implemented, in which, the involved algebraic and geometrical computations of polyhedra with the requirement of high precision are programmed by calling the APIs of Parma polyhedra library (PPL) and GNU multiple precision (GMP), and some case studies are conducted for showing the performance of APDQ(λ). In the experiments, using mountain car (MC) and acrobatic robot (Acrobot) with the continuous state spaces as the experimental subjects, the ability and the limitation of APDQ(λ) under different combinations of parameters values are probed in detail. The experimental results demonstrate that (1) APDQ(λ) behave well when $\gamma \geq 0.7$, just as shown in figures 6—13; The policy is rapidly improved in the initial phase and inclines to converge later on, moreover, it has fine adaptability to learning rate α and all sorts of parameters related to the refinement of abstract state space; (2) the performance of the algorithm degrades severely when $\gamma \leq 0.6$. Summarily, it is a novel idea on solving reinforcement learning problems with continuous state space that abstraction interpretation technique is applied to statistical learning process, and many topics deserve more attention, such as the sampling policy and the update of value functions in the context of abstract approximate model.

Keywords reinforcement learning; convex polyhedra abstraction domain; continuous states pace; $Q(\lambda)$; adaptive refinement

1 引言

强化学习是一类重要的机器学习方法,在智能控制机器人及分析预测等领域有着广泛的应用^[1-3]. 强化学习的主要任务是通过对学习主体与环境之间

交互样本的分析,利用数值型的环境回馈(奖赏),对学习主体的行为策略进行评估控制,从而得到使得累积奖赏(回报)最大化的最优控制策略.然而,强化学习方法应用于解决实际问题时,面临着“维数灾”问题,即状态爆炸问题.解决或缓解维数灾问题的基本方法主要有两种:大状态空间的离散化和函数近

似,此处的大状态空间包括连续无限状态空间.具体而言,这些方法包括状态聚集方法^[4-5]、策略空间搜索^[6]、策略函数近似、值函数近似^[7-8]、关系强化学习^[9-11]和分层强化学习^[12-14]等,它们都在状态/动作空间离散化或函数近似的基础上,结合各种不同的策略,提高了强化学习算法的可扩展性、适用性和效率.

函数近似方法通过一组基函数、核函数或神经网络来构造各种线性或非线性的值函数近似器,从而把策略评估和控制问题转化为一组参数的求解问题^[7-8,15-20].函数近似方法具有一般性,理论上能处理任意规模的状态/动作空间.但实际上,基函数或核函数的合理选择需要对环境具备足够的先验知识.如果基函数选择不合适,将导致算法收敛速度慢,甚至不收敛,或者出现泛化能力不足的情况.许多研究工作^[15-16,21-23]表明,在一些约束条件下,基于线性函数近似的时间差分(Temporal Difference, TD)类型的算法用于策略评估具有稳定性和收敛性,例如 $TD(\lambda)$;但一般情况下,基于线性函数近似的异策略类型 TD 算法不可靠,不具备收敛性,例如 Q 学习算法 $Q(\lambda)$,相关文献[18]给出了反例,显示参数的发散性.这些基于函数近似的 TD 类型算法在实现时,一般都涉及各种复杂的矩阵和函数计算.相比函数近似方法,基于状态/动作空间的离散化方法具有原理简单,实现的算法结构简单,计算类型轻量化等鲜明的特点.这种方法的关键是自动发现合适的离散化机制,能准确地用于学习各种具有连续状态空间的强化学习问题最优或近优策略.离散化方法需要平衡计算代价和模型准确性两种因素.一般而言,离散化的粒度越细,泛化能力越弱,并且计算代价越大;反之,离散化粒度越粗,泛化能力越强,计算代价越小.如果离散化机制设计不合理,使得基于抽象离散状态空间的近似动力学模型和价值模型极大地偏离环境真实的状态/动作的迁移模型和奖惩信息模型,则难以得到有效的最优策略或近优策略.

基于连续状态空间离散化强化学习技术的工作已经取得了较多的研究成果,相关实验表明该技术对于低维度的强化学习问题具有较好的学习效果^[24].粗糙编码是较早提出的一种离散化状态空间的技术^[25-26],它通过特征集合 $\{F_1, F_2, \dots, F_n\}$ 把状态空间离散化为 n 个抽象状态,原环境系统任意一个状态只满足一个特征,即只属于一个抽象状态.该方法最大的局限性在于如何提取及表示有意义的,满足强化学习需要的特征集合.与粗糙编码相比,另

外一种方法:“瓦片编码”(tile coding)在强化学习领域得到了更为广泛的应用^[27-28].在瓦片编码里,环境系统特征的提取与输入域无关,采用预先定义好的特征划分方法对原状态空间进行离散化,例如:对于二维连续状态空间可采用正方体方格对其进行状态特征的划分.此外,利用小脑神经网络(CMAC)的思想,瓦片编码技术可以灵活地结合函数近似的方法,构造出有效的强化学习算法^[29].文献[30]提出基于相邻向量量化的方法处理连续状态空间的强化学习问题,该方法具有一定的自适应性.算法对状态空间的初始离散化机制是不敏感的,可以利用在线评估所得状态动作对 Q 值,对离散化机制进行持续精化,得到更加准确的离散化机制.文献[31]提出了基于判定树的连续状态空间的离散化方法.文献[30,31]所提出的方法都具有一定的自适应性,但这两种方法描述离散状态空间的表达方式和离散化自适应调整机制具有较大的局限性,只在奖赏模型为离散型,并且对此模型具有较多先验的情况下具有较好的效果.

基于函数近似求解连续状态空间强化学习问题一直是研究重点,它的优点和不足在上文已经有所论述.针对基函数或核函数的构造和数量选择方面的问题,文献[32,33]提出了非参函数近似的方法.非参函数近似采用核函数的方法,并且核函数的数量根据样本空间的数量确定.这种方法有两个缺陷:首先,不能保证核函数的质量,例如核函数应尽量稀疏;再者单个样本所确定的核函数,难以有效地确定与核函数相关的特征参数.文献[34]提出了线性贝叶斯强化学习的方法,并且作用于 MC 问题取得了良好的学习效果,1000 个情节内获得了稳定的 100 以内的近优策略.但这种方法只能有效地应用于动力学模型为线性的强化学习问题,例如 MC 问题;另外贝叶斯网也是一种具有高计算代价的计算模型.

函数的凸性特征对最优控制求解具有重要的意义,是确保获得最优或近优策略的重要条件.以往的大部分工作集中于利用凸性函数表示 Q 和 V 值函数,而本文工作把凸性多面体应用于状态的抽象表示,这与这些工作有着根本的差别.另外,文献[35]通过凸优化提高参数稀疏度,从而提高函数近似器的质量;文献[36]把不具有凸性特征最优贝尔曼残差计算问题转化为凸性函数的计算问题,这些工作都是凸性优化在强化学习领域有意的应用.

连续状态空间的离散化是软件形式化方法领域

的一种核心技术,该领域具有大量的与状态空间离散化及离散化精化机制相关的理论、技术和算法,用于分析各类计算机系统的功能属性和性能属性(数值属性)^[37-39].这些理论、技术和算法及相关的软件工具可用于连续状态空间的马尔科夫决策系统数值属性——状态值函数或状态动作对值函数的求解,进而求解最优策略.本文的主要贡献包括提出基于凸多面体抽象域和抽象解释的连续状态空间近似表达方法,并在此基础上建立形式化的抽象和抽象精化理论;其次,结合强化学习问题状态和状态动作对相关的数值属性,即 V 值函数和 Q 值函数的特点,提出可行的离散化机制精化方法,设计具体的精化操作,使得相关的算法对初始离散机制不敏感,具有较好的可扩展性和适应性;设计并实现了基于凸多面体抽象域自适应离散化 $Q(\lambda)$ 算法 (Adaptive Polyhedra Domain based $Q(\lambda)$, APDQ(λ));最后,通过求解山地车 (Mountain Car, MC) 和杂技机器人 (Acrobatic robot, Acrobot) 等经典强化学习实例,对算法进行了详细的评估,显示了所提方法用于求解连续状态空间强化学习问题的可行性和有效性.

本文第 2 节介绍强化学习的基础知识,包括本文所涉及的各种基本概念及相关的符号定义;第 3 节是本文的重点,详细讨论凸多面体抽象域的形式化系统,以及基于多面体抽象域的离散化机制和精化技术,并给出详细的 APDQ(λ);第 4 节进行 MC 等强化学习问题的实例研究,通过各种实验数据从各个视角对算法的性能进行评估;最后总结本文,提出进一步的工作方向.

2 强化学习技术

2.1 强化学习基础框架

所谓的强化学习是指学习主体通过与环境进行交互——感知环境的状态,选择可行的动作,获得可度量的价值奖赏,计算如何进行动作的选择,使得目标状态获得最大的长期累积奖赏,这种状态到动作的映射也被称作行为策略.与监督学习不同,强化学习是通过试错来发现最优行为策略,这种试错机制基于数值化奖赏值的评估实现,而不是像监督学习那样,通过对学习主体行为作出正确或错误的二元判断来指导学习主体获得最优行为策略.下面将基于马尔科夫决策过程 (Markov Decision Process, MDP) 框架建立强化学习问题的形式化定义.

定义 1. 马尔科夫决策过程 MDP 是一四元组

$\langle S, A, P, R \rangle$, 其中 S, A, P 和 R 分别解释如下:

(1) S 是一状态集合;

(2) A 是一动作集合;

(3) $P: S \times A \times S \rightarrow [0, 1]$ 称为迁移概率函数, $P(s, a, s')$ 表示在状态 s (t 时刻) 采取动作 a 迁移到状态 s' ($t+1$ 时刻) 的概率. $P(s, a, s')$ 可简记为 $P_a(s, s')$, 显然 $\sum_{s' \in S} P(s, a, s') = 1$;

(4) $R: S \times A \rightarrow \mathbb{R}$ 称为奖赏函数, 其中 $\mathbb{R}$ 是实数集合, $R(s, a)$ 表示在状态 s , 采取动作 a 所能获得的奖赏.

MDP 结构具有描述随机性及交互性的能力, 是自动机的一种扩充, 用于大多数强化学习问题环境系统建模, 是讨论强化学习问题及各种强化学习算法的基础^[40-41]. 学习主体通过策略和环境进行交互, 实施控制. 具体的策略可用函数 $\pi: S \times A \rightarrow [0, 1]$ 进行形式化定义, $\pi(s, a)$ 表示在状态 s , 选择动作 a 的概率, 对于任意状态 s , $\sum_{a \in A} \pi(s, a) = 1$. 当对于任意状态动作对 (s, a) , $\pi(s, a) = 1$ 或 $\pi(s, a) = 0$, 策略 π 为确定性策略, $\pi(s, a) = 1$ 可简记为 $\pi(s) = a$. 按照惯例, 最优策略表示为 π^* . 各种强化学习算法普遍采用状态值函数 $V: S \rightarrow \mathbb{R}$ 和状态动作对值函数 $Q: S \times A \rightarrow \mathbb{R}$ 对策略的优劣进行评估. 给定策略 π , 状态或状态动作对值函数与环境动态性之间的关系可用贝尔曼方程进行描述^[42-43], 具体如式(1)和(2)所示.

$$\begin{aligned} V^\pi(s) &= E_\pi \{ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s_t = s \} \\ &= E_\pi \{ r_t + \gamma V^\pi(s_{t+1}) | s_t = s \} \\ &= \sum_{a \in A} \pi(s, a) (R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V^\pi(s')) \quad (1) \end{aligned}$$

$$\begin{aligned} Q^\pi(s, a) &= R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V^\pi(s') \\ &= R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') \sum_{a' \in A} \pi(s', a') Q^\pi(s', a') \quad (2) \end{aligned}$$

式(1)和(2)中的 V^π 、 Q^π 和 E_π 分别为与策略 π 相关的状态值函数、状态动作对值函数和期望, γ 为折扣因子, $0 \leq \gamma \leq 1$. 当 π 为最优策略时, 贝尔曼方程可用一种更特殊的方式——贝尔曼最优方程表示, 如式(3)和(4)所示.

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s, a, s') (R(s, a) + \gamma V^*(s')) \quad (3)$$

$$\begin{aligned} Q^*(s, a) &= R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') V^*(s') \\ &= R(s, a) + \gamma \sum_{s' \in S} P(s, a, s') \max_{a' \in A} Q^*(s', a') \quad (4) \end{aligned}$$

贝尔曼方程和贝尔曼最优方程是求解各类强化

学习问题方法的基础,主要的强化学习方法包括动态规划法、蒙特卡洛方法和时间差分方法.相比动态规划法,蒙特卡洛方法和时间差分方法不需要完备的环境动态性模型,它们通过在线或者离线样本统计的方法对策略的值函数进行评估,对策略进行改进,从而获得最优或近优策略.

2.2 强化学习算法

强化学习算法的基本任务是通过评估和改进状态值函数和状态动作对值函数,进而得到最优策略或近优策略.在模型未知的情况下,状态动作对值函数 Q 比状态值函数 V 应用更为广泛.如果已知最优的状态动作对值函数 Q^* ,可以采用贪婪的方法,根据式(5)得到最优策略,如果仅有最优状态值函数 V^* ,还需要环境的动态性知识方能获得最优策略.因此,下文的阐述和讨论围绕状态动作对值函数评估和控制的算法展开,若没有特别说明,值函数专指状态动作对值函数.当然,本文所提出的原理和方法也适用于状态值函数的评估和控制.

$$\pi^*(s) = \arg \max_{a \in A(s)} Q^*(s, a) \quad (5)$$

各种算法的具体设计和实现都采用迭代的方法,通过统计样本所获得的信息,逐次对值函数进行更新修改,以获得最优策略值函数.更新方法的基本原理来自贝尔曼方程或贝尔曼最优方程,实际更新过程取决于算法所涉及的各种因素.时间差分方法结合了蒙特卡罗方法和动态规划方法的思想,是最为重要的强化学习技术之一,在强化学习领域受到广泛的关注,各种基于 TD 方法的强化学习算法在强化学习理论及应用领域也起着基础性的作用,例如 SARSA(λ)算法和 $Q(\lambda)$ 算法.

算法 1. TD 算法.

输入: $Q_0, \alpha, \gamma, \pi_b, \pi_u$

输出: $\tilde{Q}^*$

1. WHILE (for each episode) DO
2. Initialize(s)
3. WHILE (for each step of episode) DO
4. $a \leftarrow \text{ChooseAct}(s, \pi_b)$
5. $r, s' \leftarrow \text{Executing}(a)$
6. $Q(s, a) \leftarrow Q(s, a) + \alpha [Q^{\pi_u}(s, a) - Q(s, a)]$
7. $s \leftarrow s'$
8. ENDWHILE
9. ENDWHILE

算法 1 给出了基本 TD 算法的框架,算法的输入参数 π_b 和 π_u 分别是行为策略和值函数更新策略,当 $\pi_b = \pi_u$ 时,算法称为同策略算法,否则称为异策略算法.时间差分误差(TD 误差)是 Q 值函数更新的

基本依据,算法 1 中第 6 行中的 $Q^{\pi_u}(s, a) - Q(s, a)$ 是 TD 误差计算式,当更新策略为贪婪策略时,时间差分误差为 $r + \gamma \max_{a'} Q(s', a') - Q(s, a)$.

$Q(\lambda)$ 是一种重要的基于模型未知的异策略型算法,它的行为策略和值函数更新策略分别为 ϵ -贪婪策略和纯贪婪策略,算法 2 中包含了该算法的基本逻辑框架.原理上,通过参数 λ , $Q(\lambda)$ 综合了复杂的时间信用分配模式,使得算法具有良好的适应性和扩展性;另外,具体的算法设计和实现采用了后向传播的方法,通过被称为资格迹的因子,以一种自然的,易于实现的方式实现 TD 误差的信用分配^[44-46].

根据后向观点的解释,所谓的资格迹是指当前的奖赏或时间差分误差对所访问过状态的信用分配. $Q(\lambda)$ 通过增量迭代的方式计算资格迹,进而更新值函数.状态动作对 (s, a) 的资格迹记为 $\xi(s, a)$,算法 $Q(\lambda)$ 按照下式对状态动作对的资格迹进行更新:

$$\begin{aligned} \xi_t(s, a) &= I_{ss_t} I_{aa_t} + \\ &\begin{cases} \gamma \lambda \xi_{t-1}(s, a), & Q_{t-1}(s_t, a_t) = \max_a Q_{t-1}(s_t, a) \\ 0, & Q_{t-1}(s_t, a_t) \neq \max_a Q_{t-1}(s_t, a) \end{cases} \end{aligned} \quad (6)$$

上式中 I_{ss_t} 和 I_{aa_t} 分别称为状态指示函数和动作指示函数.令 x 和 y 为状态或动作,假如 $x = y$, $I_{xy} = 1$, 否则 $I_{xy} = 0$. 值函数按照式(7)进行迭代更新.式(7)中 $\delta_t = r_{t+1} + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)$ 为 TD 误差.

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \delta_t \xi_t(s, a) \quad (7)$$

$Q(\lambda)$ 是一种强大的解决强化学习问题的算法,但基本的 $Q(\lambda)$ 算法是基于表格驱动的算法,只能处理离散的状态及动作空间.面对连续状态空间的强化学习问题,本文提出了 APDQ(λ)算法.该算法具有 $Q(\lambda)$ 的特点,包括异策略机制和基于资格迹的后向时序差分分配机制等,并通过集成一种“抽象解释-抽象精化”循环迭代机制,在表格驱动算法结构框架下处理连续状态空间的强化学习问题.下一节将详细介绍多面体域抽象及精化的基本原理及融合凸多边形域抽象和精化技术的 $Q(\lambda)$ 算法 APDQ(λ).

3 连续状态空间自适应离散化

3.1 抽象解释

在计算机科学领域,抽象解释(简称抽象)是关于计算机程序的近似模型构建的理论,广泛应用于程序静态分析及验证领域.对于形式化方法而言,抽

象技术更是该领域的基础理论及技术基石,是各种形式化软件平台能应用于大规模复杂系统分析与验证的关键.通过抽象解释技术,可以把具有连续状态空间的系统 $\mathcal{M}$,抽象成具有有限状态空间的近似系统 $\mathcal{M}_a$.这样,对系统 $\mathcal{M}$ 属性 p 的分析问题可以转化为对抽象系统 $\mathcal{M}_a$ 的属性 p' 分析,并且这种分析是可靠的,即 $\mathcal{M}_a \models p' \Rightarrow \mathcal{M} \models p$.

抽象解释是以代数格为基础的关于程序语义理论,下面对它的形式化基本知识进行介绍,关于抽象解释及抽象解释在程序分析和验证领域等领域具体应用的详细内容,请参阅相关的文献[37-39].令 $\mathcal{L} = \langle L, \sqsubseteq \rangle$ 和 $\bar{\mathcal{L}} = \langle \bar{L}, \bar{\sqsubseteq} \rangle$ 为偏序格, $\alpha: L \rightarrow \bar{L}, \gamma: \bar{L} \rightarrow L$ 为单调函数,此处函数的单调性建立在偏序关系 $\sqsubseteq$ 和 $\bar{\sqsubseteq}$ 上: $a, b \in L, a \sqsubseteq b, \alpha(a) \bar{\sqsubseteq} \alpha(b), a, b \in \bar{L}, a \bar{\sqsubseteq} b, \gamma(a) \sqsubseteq \gamma(b)$.如果函数 α 和 γ 满足条件: $\forall x \in L \cdot \forall \bar{y} \in \bar{L} \cdot \alpha(x) \bar{\sqsubseteq} \bar{y} \Leftrightarrow x \sqsubseteq \gamma(\bar{y})$,则四元组 $(\mathcal{L}, \bar{\mathcal{L}}, \alpha, \gamma)$ 称为抽象系统,简记为 $\mathcal{L}$ 和 $\bar{\mathcal{L}}$.函数对 (α, γ) 称作格 $\mathcal{L}$ 和 $\bar{\mathcal{L}}$ 之间的 Galois 连接^[47],其中 α 被称为抽象函数, γ 为具体函数, L 和 $\bar{L}$ 分别为具体状态空间和抽象状态空间, $\mathcal{L}$ 和 $\bar{\mathcal{L}}$ 分别为具体格和抽象格.在实际应用中,抽象函数 α 是关注的重点.通过设计适当的 α ,可以使得 $|\bar{L}| \ll |L|$,至少 $|\alpha(L)| \ll |L|$,其中 $\ll$ 表示“远远小于”的概念, $\alpha(L)$ 表示抽象函数 α 的值域,由具体模型所产生的抽象模型具有较小的规模和复杂性,容易处理和分析.图1给出了一个抽象系统的示例.图中 $L = \mathbb{P}(\mathbb{R})$, $\bar{L} = \{\perp, \top, 0^-, 0^+, 0, +, -\}$,抽象格的偏序关系 $\bar{\sqsubseteq}$ 由图中的盖斯图所定义.抽象状态空间 $\bar{L}$ 中的元素解释了具体格中元素所含的“正负”信息熵特征,例如 $\alpha(x) = \top$ 意味着集合 x 既包含正数,又包含负数.假如需要判断集合 x

是个正整数集合,不需要对 x 中的元素进行逐一检查,只要计算 $\alpha(x)$ 函数值即可,当然,图1所示的抽象系统过于简单,不具备实际应用价值.

通过上文的论述及图1的示例可以看出,抽象状态空间是具体状态空间的一种近似描述,在损失准确性的前提下,其规模比具体状态空间小,容易分析和处理.

3.2 凸多面体抽象域

在形式化分析与验证领域,已经提出了各种各样的抽象系统,用于各种类型系统的分析和验证,例如谓词抽象、图抽象、笛卡尔抽象、区间抽象、有界差分矩阵抽象和多面体抽象等,其中区间抽象、有界差分抽象和多面体抽象经常用于程序数值属性分析和验证.强化学习中最优策略求解问题可以归结为状态值函数和动作值函数的计算问题,这两种值函数是系统典型的数值属性.基于形式化抽象解释理论设计强化学习技术框架,具有以下几方面的显著特点:

(1) 抽象和抽象精化广泛应用于复杂计算机系统的分析和验证,具有成熟的理论基础和技术框架,这为设计具有自适应性强化学习算法奠定良好的基础,并为改进算法的性能提供指导.

(2) 形式化方法领域提供了丰富的语言机制,它们以不同的抽象粒度表达系统的状态抽象,表达先验知识.针对特定的问题,可以选择不同的抽象表达,以平衡计算成本和系统表达的准确性.

(3) 形式化方法领域具有各种高效的软件工具包,为强化学习算法中与状态抽象表达和抽象精化相关的计算提供有效的支撑,例如各种命题逻辑可满足性判定器、模理论可满足性判定器和线性规划求解器,多面体计算库^[48]和高精度数值计算库,与APDQ(λ)算法相关的软件工具为多面体计算库和高精度数值计算库.

鉴于强化学习框架中环境模型的复杂性和待分析属性的特点,在各种抽象解释中,基于多面体域的抽象系统符合学习 V 值函数和 Q 值函数对状态空间表达的需求.凸多面体是一类重要的多面体,具有许多有效的算法和软件工具用于凸多面体的代数及几何运算,因此可以广泛应用于软硬件系统的优化、综合、分析和验证众多领域.下面对凸多面体域进行详细的解释.

令 $X = \{x_1, x_2, \dots, x_n\}$ 是实型变量的集合, $a = \langle a_1, a_2, \dots, a_n \rangle \in \mathbb{R}^n$ 为 n 维行向量,则 a^T 为相应的 n 维列向量.由 m 个形如 $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{in}x_n \diamond \beta_i$

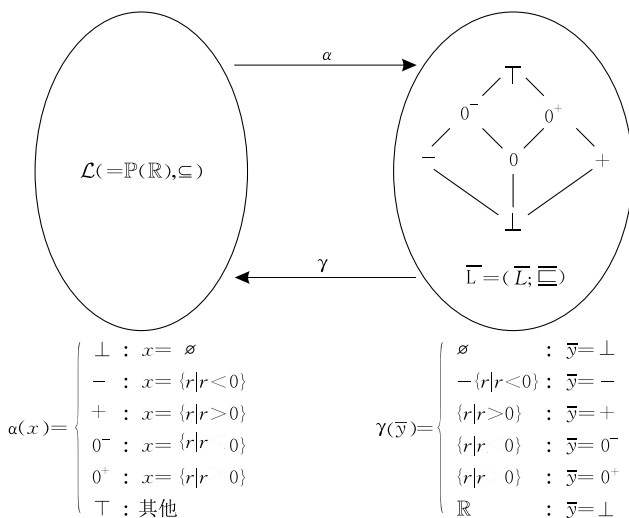


图1 $(\mathcal{L}, \bar{\mathcal{L}}, \alpha, \gamma)$ 抽象系统

的线性不等式方程构成的方程组称为凸多面体, $m \geq 1$, 式中 $1 \leq i \leq m, \alpha_{ij}, \beta_i \in \mathbb{R}, 1 \leq j \leq n$, 符号 $\diamond$ 代表 $\{\leq, \geq, <, >, =\}$ 中任意操作符. 当 $\diamond$ 仅限于 $\leq, \geq$ 和 $=$ 时, 相应的凸多面体称为闭凸多面体. 对于闭凸多面体, 可以通过对线性不等式方程进行等价变换, 得到一种统一的规范化表示, 许多多面体计算包都采用了这种处理方法, 例如 PPL^[48]. $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{in}x_n \leq \beta_i$ 通过线性变换可得等价的不等式方程 $-\alpha_{i1}x_1 + (-\alpha_{i2}x_2) + \dots + (-\alpha_{in}x_n) \geq -\beta_i$; 同理 $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{in}x_n = \beta_i$ 可用以下不等式方程组等价表示:

$$\begin{cases} \alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{in}x_n \geq \beta_i \\ -\alpha_{i1}x_1 + (-\alpha_{i2}x_2) + \dots + (-\alpha_{in}x_n) \geq -\beta_i \end{cases}$$

令 $1 \leq i \leq m, \alpha_i = \langle \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in} \rangle, A = \langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$ 为 $m \times n$ 矩阵, $B = \langle \beta_1, \beta_2, \dots, \beta_m \rangle$, 凸多面体可记为 $\mathcal{P}^X = (A, B)^X$, 表示 m 个形如 $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \dots + \alpha_{in}x_n \geq \beta_i$ 的不等式方程组所定义的闭凸多面体. α 和 α^T 分别可视为 1 行 n 列和 n 行 1 列矩阵, 直接参与矩阵的各种运算. 在不引起歧义的情况下, $\mathcal{P}^X$ 中的 X 可省略. $\mathbb{CP}^X = \{ \mathcal{P}^X = (A, B) \mid A = (a_{ij})_{n \times |X|} \in \mathbb{R}^{n \times |X|}, B = (b_{ij})_{n \times 1} \in \mathbb{R}^{n \times 1} \}$ 称为 n 维闭凸多面体抽象域. 机器浮点数为有理数, 系数为浮点数的不等式方程可以用只含有整数系数的不等式方程等价表达. 因此, 在许多情况下, 令式中 a_{ij} 和 b_{ij} 为整数, 相应的多面体抽象域具备足够的表达能力. 按照惯例, $\mathbb{CP}^X$ 简记为 $\mathbb{CP}$. 图 2 是 2 维多面体示例, 该闭凸多面体可表示为 (A, B) , 其中:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}, B = [1 \quad -1 \quad 3 \quad -8].$$

基于 n 维状态空间 $\mathbb{R}^n$ 和闭多面体抽象域 $\mathbb{CP}$, 可以定义抽象系统 $(\mathbb{P}(\mathbb{R}^n), \subseteq) \xrightarrow[\gamma]{\alpha} (\mathbb{CP}, \bar{\subseteq})$, 其中 $\mathbb{P}(\mathbb{R}^n)$ 表示 $\mathbb{R}^n$ 的幂集. 偏序关系 $\bar{\subseteq}$ 定义如下:

$$\mathcal{P}_1 \bar{\subseteq} \mathcal{P}_2 \triangleq \text{con}(\mathcal{P}_1) \subseteq \text{con}(\mathcal{P}_2),$$

其中 $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}$, 函数 $\text{con}: \mathbb{CP} \rightarrow \mathbb{P}(\mathbb{R}^n)$ 为 γ 同类型的函数, 用于表示多面体所包含点的集合, 即表示多面体的不等式约束方程组解的集合. 对于 $\mathcal{P} = (A, B) \in \mathbb{CP}$, $\text{con}(\mathcal{P}) = \{x \in \mathbb{R}^n \mid A \times x^T \geq B^T\}$, 式中 $\geq$ 表示向量之间的比较关系, $\langle b_1, b_2, \dots, b_n \rangle \diamond \langle b'_1, b'_2, \dots, b'_n \rangle$ 表示向量的相应分量满足 $b_i \diamond b'_i, 1 \leq i \leq n$. 在实际应用中常用的抽象系统为 $(\mathbb{P}(\mathbb{R}^n), \subseteq) \xrightarrow[\gamma]{\alpha} (\mathbb{CP}, \bar{\subseteq})$, 其中抽象格 $(\mathbb{CP}_n, \bar{\subseteq})$ 为 $(\mathbb{CP}, \bar{\subseteq})$ 的子

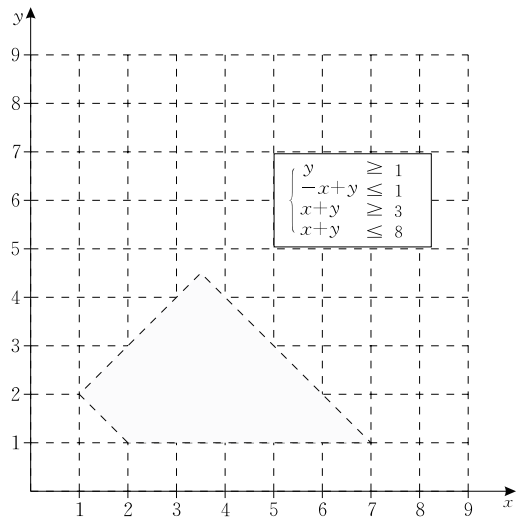


图 2 2 维凸多面体示例

格, 满足以下性质:

- (1) $\mathbb{CP}_n \subseteq \mathbb{CP}$;
- (2) 对于任意闭凸多面体 $\mathcal{P}_1, \mathcal{P}_2 \in \mathbb{CP}_n$, 如果 $\text{Con}(\mathcal{P}_1) \cap \text{Con}(\mathcal{P}_2) \neq \emptyset$, 则 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 满足以下两条条件之一:
 - ① $\text{Con}(\mathcal{P}_1) \subseteq \text{Con}(\mathcal{P}_2) \vee \text{Con}(\mathcal{P}_2) \subseteq \text{Con}(\mathcal{P}_1)$;
 - ② 对于任意 $v = \langle a_1, a_2, \dots, a_n \rangle \in \text{Con}(\mathcal{P}_1) \cap \text{Con}(\mathcal{P}_2)$, $\sigma_1, \sigma_2, \dots, \sigma_n$ 为任意小的非负实数, 且存在 $i, 1 \leq i \leq n, \sigma_i \neq 0$, 则 $(\langle a_1 \pm \sigma_1, a_2 \pm \sigma_2, \dots, a_n \pm \sigma_n \rangle \in \text{Con}(\mathcal{P}_1) \rightarrow \langle a_1 \pm \sigma_1, a_2 \pm \sigma_2, \dots, a_n \pm \sigma_n \rangle \notin \text{Con}(\mathcal{P}_2)) \wedge (\langle a_1 \pm \sigma_1, a_2 \pm \sigma_2, \dots, a_n \pm \sigma_n \rangle \in \text{Con}(\mathcal{P}_2) \rightarrow \langle a_1 \pm \sigma_1, a_2 \pm \sigma_2, \dots, a_n \pm \sigma_n \rangle \notin \text{Con}(\mathcal{P}_1))$.

满足以上两条性质的闭凸多面体集合 $\mathbb{CP}_n$ 称为规范凸多面体集合. 对于该集合中没有包含关系的两个凸多面体 $\mathcal{P}_1$ 和 $\mathcal{P}_2$: 要么没有相交的部分, 即 $\text{Con}(\mathcal{P}_1) \cap \text{Con}(\mathcal{P}_2) = \emptyset$; 要么它们的相交部分必位于边界, 如图 3 所示. 给定马尔科夫决策过程 $\mathcal{M} = \langle S, A, P, R \rangle$, 令 $\mathcal{S}_{fin} \subseteq \mathbb{CP}_n$ 是一有限的凸多面体集合, $\bigcup_{\mathcal{P} \in \mathcal{S}_{fin}} \text{Con}(\mathcal{P}) \supseteq S$, 结构 $\langle \mathcal{S}_{fin}, A_{ab}, P_{ab}, R_{ab} \rangle$ 为马尔科夫决策过程 $\mathcal{M}$ 的一种抽象, 其中 $A_{ab} = A, P_{ab}$ 的类型为 $\mathcal{S}_{fin} \times A \times \mathcal{S}_{fin} \rightarrow [0, 1]$, R_{ab} 的类型为 $\mathcal{S}_{fin} \times A \rightarrow \mathbb{R}$. 由于本文基于模型未知的方法进行最优策略求解, 因此不对抽象系统的动态性函数 P_{ab} 和 R_{ab} 进行详细的定义, 关于这方面的知识可参考相关的文献^[49]. MC 是强化学习研究领域用于显示算法特性和性能经常采用的例子, 它的状态空间包含位移和速度两个连续维度, 为无限状态空间, 图 3 显示了 MC 的状态空间的一种抽象 $\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\} \in \mathbb{P}(\mathbb{CP}_n)$. 值得注意的是, 抽象往往蕴含着先验知识,

例如图 3 显示的是一种“直觉的经验”：速率和位移的方向预示着不同的决策方案。“好”的抽象能改善算法的性能，加速算法的收敛速度。在不考虑系统动态性 P_{ab} 和 R_{ab} 的情况下，很容易建立系统不同抽象之间的精化关系，精化关系记为 $\sqsubseteq$ 。对于 $S_1, S_2 \in \mathcal{P}(\mathbb{CP}_n)$ ：

$$S_1 \sqsubseteq S_2 \text{ iff } \forall P \in S_2 \cdot \exists P' \in S_1 \cdot \text{Con}(P) \subseteq \text{Con}(P').$$

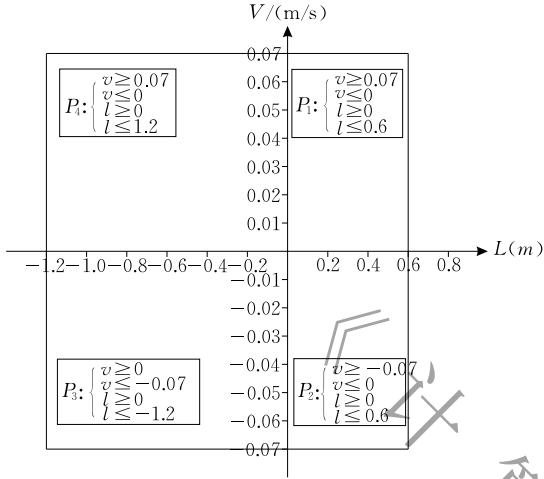


图 3 MC 状态空间抽象初始化

如果 S_1 和 S_2 满足精化关系 $\sqsubseteq$ ，抽象系统 $\langle S_2, A, P_{ab}^2 \rangle$ 称为 $\langle S_1, A, P_{ab}^1 \rangle$ 的精化。在不考虑系统动态性的情形下，抽象系统可以用抽象状态空间作为简称，例如 S_1 和 S_2 。显然，结构 $\langle \mathcal{P}(\mathbb{CP}_n), \sqsubseteq \rangle$ 为偏序结构。另外，如果 S 是 S 的抽象系统， $s \in S, P \in S, s \in \text{Con}(P)$ ，那么 $A(P) \subseteq A(s)$ 。

3.3 APDQ(λ) 算法

APDQ(λ) 算法的核心是在给定待求解问题初始抽象 S_0 的情形下，采用逐步求精的方法，获得一抽象系列 $S_1, S_2, \dots, S_n$ ，其中 $S_i \sqsubseteq S_{i+1}, 0 \leq i < n$ 。 S_n 是满足算法收敛性要求所获得的原马尔科夫决策过程的抽象系统。通过 S_n 可以获得近似最优策略。对于任意状态 $s \in S, P \in S_n, s \in \text{Con}(P)$ ：

$$\tilde{\pi}^* = \arg \max_{a \in A(s)} \tilde{Q}^*(P, a) \quad (8)$$

算法 2 显示了 APDQ(λ) 的细节，该算法继承了 $Q(\lambda)$ 的基本框架，但集成了基于凸多面体抽象域的自适应抽象精化机制，通过这种机制，APDQ(λ) 可以处理具有连续状态空间的强化学习问题。该算法应用具有较强表达能力，而又较易处理的凸多面体描述抽象状态；在相关阈值参数的控制下，根据在线样本空间和由在线样本空间所诱导的抽象状态样本空间的数值性属性，自适应地对离散化机制进行精化和计算近优策略。

算法 2. APDQ(λ) 算法.

输入： $\gamma, \lambda, \alpha, \epsilon, \delta$ /* ϵ -贪婪策略； γ, λ, α 强化学习相关参数； ϵ, δ 精化操作相关参数， ϵ 为 Q 值方差阈值， δ 为分辨率阈值 */
输出： $\tilde{Q}^*$

1. $S_0 \leftarrow \text{InitializingAbstraction}(S)$ /* S 为环境具体状态空间 */
2. $\text{Initialize}(\tilde{Q})$
3. $i \leftarrow 0$
4. WHILE (for each episode) DO
5. $\text{Reset}(e)$ /* 重置资格迹 */
6. $s \leftarrow s_0$
7. $P \leftarrow \text{RegisterConcreteState}(s)$ /* $P \in S_i, s \in \text{Con}(P)$ */
8. $a \leftarrow \text{ChooseAct}(s)$ /* 随机选择动作 $a \in A(s)$ */
9. WHILE (for each step of episode) DO
10. $(r, s') \leftarrow \text{Executing}(a)$
11. $P' \leftarrow \text{RegisterConcreteState}(s')$
12. IF ($\text{TestResolution}(\delta)$) THEN
13. $S_{i+1} \leftarrow \text{Refinement}(S_i)$
14. $i \leftarrow i + 1$
15. ENDIF
16. $a^* \leftarrow \arg \max_{a \in A(s')} Q(P', a)$
17. $a' \leftarrow \text{ChooseActWithProbability}(\epsilon)$
18. $\Delta \leftarrow r + \gamma Q(P', a^*) - Q(P, a)$
19. $e(P, a) \leftarrow e(P, a) + 1$
20. WHILE (for all (P, a) pairs) DO
21. IF ($a' \neq a^*$) THEN
22. $e(P, a) \leftarrow \gamma \lambda e(P, a)$
23. ELSE
24. $e(P, a) \leftarrow 0$
25. ENDIF
26. ENDWHILE
27. WHILE (for all $P \in S_i$) DO
28. IF ($\text{TestQMVB}(\epsilon)$) THEN
29. $S_{i+1} \leftarrow \text{Refinement}(S_i)$
30. $i \leftarrow i + 1$
31. ENDIF
32. ENDWHILE
33. $\text{Widen}(\zeta)$
34. $a \leftarrow a'$
35. $P \leftarrow P'$
36. ENDWHILE
37. ENDWHILE

算法 2 实线框中的伪代码与抽象精化机制相

关,重点包括精化操作触发评估机制、精化操作实施方法和抽象宽化操作. 精化操作触发评估机制决定精化操作实施的条件,精化操作实施的条件对 APDQ(λ)算法的性能有重要影响. 算法 2 采用两种类型的评估方法:一种方法基于抽象状态的分辨率,另一种方法基于抽象状态相关样本集合的 Q 值方差. 抽象状态的分辨率反映抽象状态的聚集程度,本文采用累积奖赏对抽象状态的分辨率进行度量. 累积奖赏记为 $accReward$,它的计算是与样本相关的. 对于给定样本 $sam: s_0 \xrightarrow{a_1, r_1} s_1 \xrightarrow{a_2, r_2} s_2 \xrightarrow{a_3, r_3} \dots \xrightarrow{a_n, r_n} s_n$ $accReward_0 = 0$, $accReward_{i+1} =$:

① $accReward_i + r_{i+1}$ if $\mathcal{P}_i = \mathcal{P}_{i+1}$, $a_i = a_{i+1}$, $s_i \in Con(\mathcal{P}_i)$, $s_{i+1} \in Con(\mathcal{P}_{i+1})$;

② r_{i+1} if $\mathcal{P}_i \neq \mathcal{P}_{i+1} \vee a_i \neq a_{i+1}$, $s_i \in Con(\mathcal{P}_i)$, $s_{i+1} \in Con(\mathcal{P}_{i+1})$.

其中 $\mathcal{P}_i, \mathcal{P}_{i+1} \in \mathcal{S}$, $\mathcal{S}$ 为执行样本 sam 的动作 a_{i+1} 时马尔科夫决策系统的抽象状态空间. 当 $accReward$ 超过分辨率阈值时,实施精化操作. 显然,给定的分辨率阈值 δ 越大,则抽象系统具有更大的近似性,抽象状态空间更小,问题较易处理,但不利之处是算法的稳定性相对较差,较难甚至不能收敛到最优策略. 基于分辨率确定精化操作实施的方法对具体状态空间采用统一的评估标准,但实际上系统不同的子状态空间具有不同的数值属性特征,例如 Q 值,与之相适应,不同的子状态空间应采用不同的抽象机制. 另外,基于抽象分辨率的精化方法需要对环境的奖惩模型具有一定的先验知识. 基于抽象状态相关样本 Q 值方差的方法则体现了具体状态空间的局部性特征,可以得到与状态空间数值属性相关的抽象精化机制,并且这种方法不需要环境奖惩模型的先验知识. 给定抽象状态 $\mathcal{P} \in \mathcal{S}$,令 $CS(\mathcal{P})$ 为在线所获取样本中满足 $s \in Con(\mathcal{P})$ 的具体状态 s 的集合, $Var(\mathcal{P}, a)$ 为抽象状态的 Q 值方差,该方差通过样本状态集合的 Q 值计算而得

$$Var(\mathcal{P}, a) =$$

$$\sqrt{1/(n-1) * \sum_{s \in CS(\mathcal{P})} (Q(s, a) - E(\mathcal{P}, a))^2} \quad (9)$$

式(9)中 n 为有限集合 $CS(\mathcal{P})$ 的大小, $E(\mathcal{P}, a)$ 使用 $1/n * \sum_{s \in CS(\mathcal{P})} Q(s, a)$ 作为近似值. 当 $\max_{a \in A(\mathcal{P})} Var(\mathcal{P}, a) \geq \epsilon$ 时,对抽象状态 $\mathcal{P}$ 实施精化操作, ϵ 为方差阈值. 具体实现时,也可以选择更为简单的判断标准 $\max_{s \in CS(\mathcal{P}), a \in A(\mathcal{P})} |Q(s, a) - E(\mathcal{P}, a)| \geq \epsilon$. $A(\mathcal{P})$ 为在抽象状态所能选择动作的集合,严格地说, $A(\mathcal{P}) =$

$\bigcap_{s \in CS(\mathcal{P})} A(s)$, 但一般情况下,对于任意 $s \in CS(\mathcal{P})$, $A(\mathcal{P}) = A(s)$. 这种评估标准把抽象状态的抽样 Q 值特征作为是否实施抽象精化的依据,体现了抽象状态数值属性的抽象本质,避免了基于分辨率方法的不足. 当然,基于样本状态 Q 值方差的方法会增加算法的计算成本,例如为了得到样本状态的统计 Q 值,算法中需要采用时序差分的方法对样本状态的 Q 值进行更新,当样本状态空间较大时,这部分计算成本会影响算法的时空复杂性. 并且这种方法由于涉及抽象系统和具体系统资格迹和 Q 值函数的同步更新,因此有着更复杂的参数调控机制.

具体的抽象精化操作有三种类型,分别称为盒式精化操作 ($BoxRefinement$)、线性拟合精化操作 ($LFRefinement$) 和最小方差线性拟合精化操作 ($MVLFRefinement$),这三种精化操作的形式化类型为 $\mathbb{CP}_n \rightarrow \mathbb{P}(\mathbb{CP}_n)$. 令 $s \in Con(\mathcal{P})$, $s[i]$ 为状态 s 第 i 维(分量)的数值, $Max(\mathcal{P}, i)$ 为抽象状态 $\mathcal{P}$ 第 i 维的最大取值,即存在一状态 s ,对于任意状态 $s' \in Con(\mathcal{P})$, $Max(\mathcal{P}, i) = s[i]$, $s[i] \geq s'[i]$;类似的方法可以定义 $Min(\mathcal{P}, i)$. 对于 n 维闭凸多面体 $\mathcal{P}$,可计算出 n 组 $Max(\mathcal{P}, i)$ 和 $Min(\mathcal{P}, i)$,通过关系运算符 $\leq$ 和 $\geq$,可以构造出 n 组形如 $x_i \leq (Max(\mathcal{P}, i) - Min(\mathcal{P}, i))/2$ 和 $x_i \geq (Max(\mathcal{P}, i) - Min(\mathcal{P}, i))/2$ 的不等式. 每组相关不等式选择其中之一可构成 2^n 个不等式方程组,即 2^n 个闭凸多面体,为了简单,把这 2^n 个由 $\mathcal{P}$ 诱导产生的多面体用 $[0, 2^n - 1]$ 之间的整数表示,记为 $\mathcal{P}[0], \mathcal{P}[1], \dots, \mathcal{P}[2^n - 1]$. 这样, $BoxRefinement$ 可定义如下:

$$BoxRefinement(\mathcal{P}) = \{\mathcal{P}' \in \mathbb{CP}_n \mid \exists 0 \leq i \leq 2^n - 1. \\ Con(\mathcal{P}') = Con(\mathcal{P}) \cap Con(\mathcal{P}[i])\}.$$

显然,当 n 较小时, $BoxRefinement$ 是一种成本较小的精化操作,算法 2 中, $BoxRefinement$ 一般结合基于分辨率的评估方法一起使用. 相较而言, $LFRefinement$ 和 $MVLFRefinement$ 精化操作更加复杂,与抽象状态的样本状态集合 $CS(\mathcal{P})$ 密切相关. 以所有的 $s \in CS(\mathcal{P})$ 作为样本点,选择某一状态维作为因变量,采用最小二乘方法进行线性拟合,获得线性方程 $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n + b = 0$. 如果设置第 i 维为因变量,则 $\theta_i \neq 0$,最小二乘公式如下所示:

$$\sum_{s = \langle a_1, a_2, \dots, a_n \rangle \in CS(\mathcal{P})} (a_i - (\theta_1 * a_1 + \dots + \theta_{i-1} * a_{i-1} + \theta_{i+1} * a_{i+1} + \dots + \theta_n * a_n + b) / (-\theta_i))^2.$$

令多面体 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 满足:

$$\textcircled{1} \text{Con}(\mathcal{P}_1) = \{s = \langle a_1, a_2, \dots, a_n \rangle \mid s \in \text{Con}(\mathcal{P}) \wedge \theta_1 * a_1 + \dots + \theta_n * a_n + b \geq 0\};$$

$$\textcircled{2} \text{Con}(\mathcal{P}_2) = \{s = \langle a_1, a_2, \dots, a_n \rangle \mid s \in \text{Con}(\mathcal{P}) \wedge \theta_1 * a_1 + \dots + \theta_n * a_n + b \leq 0\}.$$

显然 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 是闭凸多面体. 根据 $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 可以给出 *LFRRefinement* 的定义:

$$\text{LFRRefinement}(\mathcal{P}) = \{\mathcal{P}_1, \mathcal{P}_2\}.$$

MVLFRefinement 的定义方式与 *LFRRefinement* 类似, $\text{MVLFRefinement}(\mathcal{P}) = \{\mathcal{P}_1, \mathcal{P}_2\}$, $\mathcal{P}_1$ 和 $\mathcal{P}_2$ 由拟合直线 $\theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n + b = 0$ 产生, 拟合条件为

$$\textcircled{1} \min\left(\sum_{a \in A(\mathcal{P}_1)} \text{Var}(\mathcal{P}_1, a) + \sum_{a \in A(\mathcal{P}_2)} \text{Var}(\mathcal{P}_2, a)\right);$$

$$\textcircled{2} \max_{a \in A(\mathcal{P}_1)} \text{Var}(\mathcal{P}_1, a) \leq \epsilon;$$

$$\textcircled{3} \max_{a \in A(\mathcal{P}_2)} \text{Var}(\mathcal{P}_2, a) \leq \epsilon.$$

LFRRefinement 和 *MVLFRefinement* 都与抽象状态相关的样本状态分布特征相关, 但 *LFRRefinement* 拟合的直线只与状态维的值特征相关, 而 *MVLFRefinement* 拟合的直线则与样本状态 Q 值特征相关. 一般而言, *MVLFRefinement* 精化操作更符合问题和抽象的语义内涵, 但 *MVLFRefinement* 操作作为含有不等式约束条件下的优化计算, 具有较高的时空复杂性, 这会极大地增加频繁调用这类操作的算法的时空开销, 进而影响样本空间的大小和学习效果. *LFRRefinement* 是介于 *BoxRefinement* 和 *MVLFRefinement* 之间的一种精化操作. 由于 *MVLFRefinement* 具有较大的计算代价, 因此在实例研究中只采用了 *BoxRefinement* 和 *LFRRefinement*.

算法 2 中第 1 和 7 行的 *InitializingAbstraction()* 和 *RegisterConcreteState()* 分别用于计算环境的初始抽象状态集合 $\mathcal{S}_0$ 和判定样本状态 s 所属的抽象状态 $\mathcal{P}$. 第 12 和 26 代码块中的 *TestResolution()* 和 *TestQMV()* 分别用于在给定相关阈值的情况下, 评估是否实施精化操作, 评价方法为上文所讨论的两种方法. *Refinement()* 采用 *BoxRefinement*、*LFRRefinement* 或 *MVLFRefinement* 的方法实现. 第 28 行的 *Widen()* 的类型为 $\mathbb{P}(\mathbb{CP}_n) \rightarrow \mathbb{CP}$, 该操作用于把抽象状态空间中的多个闭凸多面体聚集成为 1 个多面体, 以便得到更小的抽象状态空间, 加速收敛过程. 多个凸多面体的并操作结果 $\bigcup_{i=1}^n \mathcal{P}_i$ 并不一定是凸多面体, 这为设计合适的 *Widen()* 带来很大的难度, 如果按照文献[50]的方法设计操作 $\sqcup$ 代替 $\cup$, 这样虽然能保证 $\bigcup_{i=1}^n \mathcal{P}_i$ 为凸多面体, 并且 $\text{Con}(\bigcup_{i=1}^n \mathcal{P}_i) \supseteq$

$\text{Con}(\bigcup_{i=1}^n \mathcal{P}_i)$, 但不能保证 $\{\bigcup_{i=1}^n \mathcal{P}_i\} \cup \mathcal{S} \subseteq \mathbb{CP}_n$. 基于上述的原因, 算法 2 在实现时并没有引入程序数值属性分析中常用的 *Widen* 技术, 用于加速算法的收敛, 重点通过调整各种学习参数和与抽象精化相关的阈值提高算法的性能.

下面分两种情况对算法 *APDQ*(λ) 的最坏计算代价进行讨论: (1) 分辨率结合 *BoxRefinement*; (2) Q 值方差结合 *LFRRefinement*. 算法 2 所涉及的计算代价主要包括 4 方面: 抽象状态 Q 值和资格迹的计算、具体状态 Q 值和资格迹的计算、具体状态至抽象状态映射的计算和抽象精化操作. 为了简化讨论, 把单个具体状态至抽象状态映射的计算、 Q 值计算和资格迹计算视作代价相同的操作; 另外, 由于 $\mathcal{S}_0$ 和 $\mathcal{S}_n$ 的大小是确定的, 抽样学习过程中, 精化操作调用的次数也是确定的, 因此可以统一用 cost_{ref} 表示这方面的计算代价. 令单个样本长度的上限为 l , 样本数量为 n , 由于抽象状态空间大小是动态变化的, 因此单次样本所涉及的抽象状态数目平均为 $1/2|\mathcal{S}_n|$. 对于单个样本而言, 第 1 种情况涉及 l 次状态映射计算, 计算 TD 误差所需的 $|A|$ 次状态 Q 值枚举比较计算, 抽象状态 Q 值和资格迹计算次数皆为 $1/2|\mathcal{S}_n|$, 因此第 1 种情况总的计算代价可表示为 $n(|\mathcal{S}_n| + |A| + l) + \text{cost}_{ref}$. 第 2 种情况则在第 1 种情况的基础上多了 l 次具体状态 Q 值和 l 次资格迹的计算, 因此总的计算代价为 $n(|\mathcal{S}_n| + |A| + 3l) + \text{cost}_{ref}$.

决定 *APDQ*(λ) 算法性能的主要因素包括基于凸多面体计算的抽象评估和精化机制. 凸多面体的各种计算是一类具有较高时空复杂性的代数和几何计算, 为了尽可能地降低这类计算对算法时空性能的负面影响, *APDQ*(λ) 算法的实现采用了多面体计算的专用库 PPL^[48], 并且算法只涉及闭凸多面体对象. PPL 是由意大利 Parma 大学数学系一个研究小组所开发的开源数学工具包, 该工具包从语法和语义层次对凸多面体的计算提供高效的支撑, PPL 还提供了各种高层数值属性抽象方法, 专门用于程序数值属性的验证和分析. 另外, *APDQ*(λ) 所涉及的各种具有高精度要求的基础数值计算通过调用 GMP(GNU Multiple Precision)库^①所提供相关函数实现. GMP 库提供了整数、有理数和浮点数任意精度计算的功能.

① GMP Man 6. 1. 2 (CAIDA). <http://gmplib.org/gmp-man-6.1.2.pdf>, 2017, 3, 10

3.4 APDQ(λ)算法的收敛性

文献[44]给出了对于有限状态空间和动作空间情况下 $TD(\lambda)$ 算法的收敛性证明, 文献[45-46]给出了有限状态空间和动作空间情况下 $Q(\lambda)$ 算法收敛性的证明. 对于无限随机马尔科夫决策问题, 文献[51]给出了满足条件(1)、(2)和(3)下基于网格离散化的 Q 学习算法收敛性的证明:

(1) 状态集合 S 和动作集合 A 为紧集, 并且对于任意 $s, s' \in S, a, a' \in A, a \in A(s), a' \in A(s')$, 满足 $\|a - a'\| \leq F\|s - s'\|$, 其中 F 为一正的常数.

(2) 对于任意 $s, s' \in S, a, a' \in A, w, w' \in W$, 状态迁移函数 f 和奖赏函数 R 满足 Lipschitz 连续性条件:

$$\textcircled{1} \|f(s, a, w) - f(s', a', w')\| \leq L(\|s - s'\| + \|a - a'\| + \|w - w'\|);$$

$$\textcircled{2} \|R(s, a, w) - R(s', a', w')\| \leq M(\|s - s'\| + \|a - a'\| + \|w - w'\|).$$

式中 L 和 M 为正的常数, $\|\cdot\|$ 为范数, W 是随机噪声集合, W 用于为系统的随机性建模.

(3) 噪声信息 w 的概率分布依赖于状态动作对 (s, a) , 且概率分布函数 $p(s, a)$ 满足 Lipschitz 连续性条件, 即 $\|p(s, a) - p(s', a')\| \leq O(\|s - s'\| + \|a - a'\|)$.

当条件(3)成立时, 状态迁移函数 f 和奖赏函数 R 的 Lipschitz 连续性条件可简化为

$$\textcircled{1} \|f(s, a, w) - f(s', a', w')\| \leq L'(\|s - s'\| + \|a - a'\|);$$

$$\textcircled{2} \|R(s, a, w) - R(s', a', w')\| \leq M'(\|s - s'\| + \|a - a'\|).$$

令 $S = \{S_1, \dots, S_n\}$ 和 $A = \{A_1, \dots, A_m\}$. 分别是无限状态空间 S 和动作空间 A 经过网格化处理后所得的有限抽象状态空间, 例如算法 2 持续精化所得的多面体状态空间 $S_1, S_2, \dots, S_n$ 和 d_A 分别表示离散化状态空间 S 和动作空间 A 的离散化尺度^[39]. $\bar{Q}^*$ 依概率 1 收敛于 Q^* 的条件为 $d_S + d_A \rightarrow 0$, 即 $d_S \rightarrow 0, d_A \rightarrow 0$. 算法 2 仅涉及状态空间的离散化, 因此, 收敛条件可简化为 $d_S \rightarrow 0$. 当样本的数目趋于无穷时, 算法 2 能否依概率 1 收敛取决于算法中所采取的精化操作机制. 下面分别对采取精化操作 $BoxRefinement$ 和 $LFRRefinement$ 的收敛性特性进行讨论.

令 $S = \{P_1, P_2, \dots, P_n\}$ 为基于多面体抽象机制的离散化状态空间, 对于 $P_i, 1 \leq i \leq n, d_{P_i}$ 为多面

体 P_i 的直径, $d_{P_i} = \max_{s_1, s_2 \in \text{Com}(P_i)} \|s_1 - s_2\|$, 则 $d_S = \max_{1 \leq i \leq n} d_{P_i}$. 显然对于连续状态空间而言, 当 $d_S \rightarrow 0, n \rightarrow \infty$. 在算法 2 中, 当初始抽象状态空间 S_0 为有限集合, 当对 S_i 实施 $BoxRefinement$ 精化操作时, 根据 $BoxRefinement$ 的定义, $|S_i| < |S_{i+1}| \leq |S_i| + 4$, 因此 $n \rightarrow \infty$ 意味着算法 2 中实施 $BoxRefinement$ 精化操作的次数不受限制. 在保证获取样本公平性的基础上, 能否持续进行 $BoxRefinement$ 精化取决于累积奖赏 $accReward$ 阈值 δ 的设置. 一般情况下, 由于环境的模型因素 P 和 R 是未知的, 无法确保设计正确的阈值. 当奖赏模型类似于 MC 问题时, 基于分辨率的方法最容易调控, 且能保证收敛; 但一般缺乏环境的动力学属性的情况下, 不能保证收敛. 对于 $LFRRefinement$ 操作, 当 $\text{Var}(\mathcal{P}, a) \rightarrow 0$, 根据 $\text{Var}(\mathcal{P}, a)$ 的定义, 意味着抽象状态相关的抽样样本具体状态 Q 值保持一致, 这意味着采用 $LFRRefinement$ 精化操作具有收敛性.

4 实 验

4.1 实验设计

本节通过设计一系列针对算法 APDQ(λ) 的实验, 评估各类参数和精化方法对其性能的影响. 选取 MC 和 Acrobot 强化学习问题作为实验对象, MC 问题具有 2 维状态空间, 而 Acrobot 问题更复杂, 其状态空间具有 4 个连续维度, 它们的环境系统模型示意图如图 4 和图 5 所示. 由于 APDQ(λ) 算法在执行过程中, 抽象状态空间是持续动态变化的, 因此传统的强化学习参数, 例如折扣率 γ 和学习率 α , 对算法的学习效果具有更加复杂的影响机制; 另外, 与精化操作相关的各种阈值参数直接决定着动态状态空间大小的变化率、绝对大小和局部状态空间的局部特征, 这些状态空间度量特征对抽象状态和具体状态 Q 值函数的计算、策略评估和控制的准确性和效率等有着重要的影响.

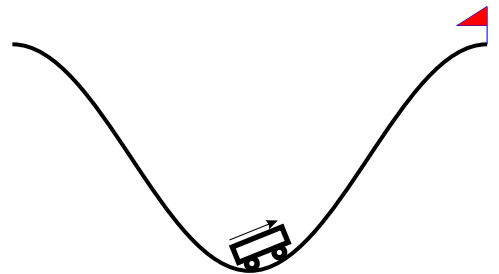


图 4 MC 问题模型

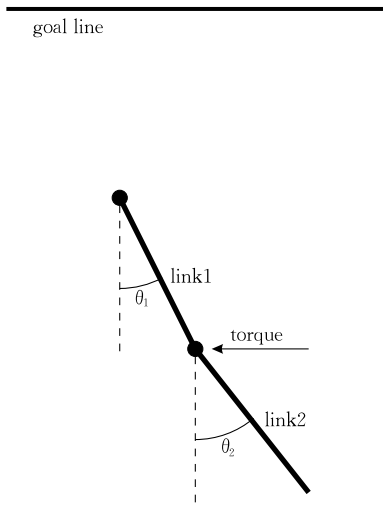


图 5 Acrobot 问题模型

MC 问题的状态空间 $S = L \times V$, 其中 $L = [-1.2, 0.6]$ 称为位移维, $V = [-0.07, 0.07]$, 称为速度维, 离散动作空间 $A = \{BRA, ZERO_ACC, ACC\}$, BRA 、 $ZERO_ACC$ 和 ACC 分别表示动作“制动”、“0 加速”和“加速”. 动作 BRA 、 $ZERO_ACC$ 和 ACC 分别用整数 -1 、 0 和 1 进行编码, MC 的奖惩模型用下式表示:

$$R_{MC}(s, a, s') = \begin{cases} 0, & s'[1] \geq 0.5 \\ -1, & s'[1] < 0.5 \end{cases}$$

令 $s' = f(s, a)$ 为 MC 的状态迁移模型, s, a, s' 满足: (1) $s'[1] = s[1] + s[2]$; (2) $s'[2] = s[2] + a * 0.001 + \cos(3 * s[1]) * (-0.0025)$. 对于 $s \in S, 1 \leq i \leq 2, s[i]$ 表示 2 维状态第 i 分量的值. 显然, 此处的 MC 环境模型是一种简化的, 不具有随机性的模型. Acrobot 问题状态空间具有较高的维度, 包括 2 个角度维和 2 个角速度维, 属比 MC 问题更难求解的问题. 2 个角度维用 θ_1, θ_2 表示, 2 个角速度维用 $\dot{\theta}_1, \dot{\theta}_2$ 表示, Acrobot 问题的状态空间 $S = \theta_1 \times \theta_2 \times \dot{\theta}_1 \times \dot{\theta}_2$, $\theta_1 = \theta_2 = [-\pi, \pi]$, $\dot{\theta}_1 = [-4\pi, 4\pi]$, $\dot{\theta}_2 = [-9\pi, 9\pi]$. 在 Acrobot 问题中, 通过在 link1 和 link2 的连接处施加扭力矩 (torque), 对 link1 和 link2 的状态进行控制, 使 link2 的末端达到或超过目标线 (goal line). 动作空间 $A = \{NEG_TOR, ZERO_TOR, POS_TOR\}$, 3 个动作分别表示施加反向扭力矩, 不施加任何扭力矩和施加正向扭力矩. Acrobot 问题的动力学模型比 MC 问题更复杂, 给定状态 $\langle \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2 \rangle \in S$, 后继状态角度和角速度分量的值通过计算角加速度积分获得, link1 和 link2 的角加速度 $\ddot{\theta}_1, \ddot{\theta}_2$ 的计算如下式所示:

$$\textcircled{1} \ddot{\theta}_1 = d_1^{-1} (d_2 \ddot{\theta}_2 + \phi_1);$$

$$\textcircled{2} \ddot{\theta}_2 = (m_2 l_{c_2}^2 + I_2 - d_2^2/d_1)^{-1} (\tau + d_2/d_1 \phi_1 - m_2 l_1 l_{c_2} \ddot{\theta}_1^2 \sin \theta_2 - \phi_2).$$

$\phi_2 = m_2 l_{c_2} g \cos(\theta_1 + \theta_2 - \pi/2)$, 其中 d_1, d_2, ϕ_1, ϕ_2 计算如下:

$$\textcircled{1} d_1 = m_1 l_{c_1}^2 + m_2 (l_1^2 + l_{c_2}^2 + 2l_1 l_2 \cos \theta_2) + I_1 + I_2;$$

$$\textcircled{2} d_2 = m_2 (l_{c_2} + l_1 l_{c_2} \cos \theta_2) + I_2;$$

$$\textcircled{3} \phi_1 = -m_2 l_1 l_{c_2} \ddot{\theta}_2^2 \sin \theta_2 - 2m_2 l_1 l_{c_2} \dot{\theta}_2 \dot{\theta}_1 \sin \theta_2 + (m_1 l_{c_2} + m_2 l_1) g \cos(\theta_1 - \pi/2) + \phi_2;$$

$$\textcircled{4} \phi_2 = m_2 l_{c_2} g \cos(\theta_1 + \theta_2 - \pi/2).$$

上式中, $l_1 = l_2 = 1 \text{ m}$, $m_1 = m_2 = 1 \text{ kg}$, $l_{c_1} = l_{c_2} = 0.5 \text{ m}$ 分别为 link1 和 link2 的长度, 质量和重心位置, $g = 9.8$ 为重力加速度. $\tau \in \{-1, 0, 1\}$ 为动作编码, 每 0.2 s 进行 1 次动作决策. $I_1 = I_2 = 1$ 为采用近似积分所引入的与惯性相关的常数. Acrobot 问题的奖惩模型如下:

$$R_{Ac}(s, a, s') = \begin{cases} 0, & -(\cos s'[1] + \cos(s'[1] + s'[2])) > 1 \\ -1, & -(\cos s'[1] + \cos(s'[1] + s'[2])) \leq 1 \end{cases}$$

根据算法 APDQ(λ) 的主要特点, 本文设计了 8 个系列的实验: 第 I 系列实验探究不同学习参数作用下算法的性能; 第 II 系列实验探究不同精化操作参数及不同精化操作作用下算法的性能; 最后, 设计相关的对比实验, 比较算法 APDQ(λ) 与基于统一离散化算法的性能. 为了方便, 把第 I 系列和第 II 系列实验称为验证实验, 第 III 系列称为对比实验.

4.2 实验结果及分析

4.2.1 验证实验

4.2.1.1 第 I 系列实验

第 I 系列实验主要研究了行为策略为 ϵ -Greedy ($\epsilon = 0.01$), 参数 α 和 γ 取不同值对算法性能的影响. 对于 MC 和 Acrobot 问题, 算法的性能主要以在线情节的数量与相应情节到达目标位置的步数之间的关系进行衡量, 每个情节步数的上限为 1000. 表 1 给出了该系列实验详细的参数设置.

表 1 第 I 系列实验参数设置

固定参数	测试参数	BoxRefinement($\square$)	LFRRefinement($\square$)
$\delta = -3\square$			
$\alpha = 0.9\square$	$\gamma = 0.3, 0.7, 0.9$	MC(图 6)	MC(图 8)
$\epsilon = 0.5\square$			Acrobot(图 10)
$\delta = -3\square$			
$\gamma = 0.9$	$\alpha = 0.3, 0.7, 0.9$	MC(图 7)	MC(图 9)
$\epsilon = 0.5\square$			

在表 1 中的固定参数中, 具有上标 $\square$ 的参数

用于与 *BoxRefinement* 精化操作相关的实验;具有上标 $\square$ 的参数用于与 *LRefinement* 精化操作相关的实验;没有上标的参数则可用于两种精化操作相关的实验。*BoxRefinement* 只适用于具有低维度状态空间强化学习问题的求解,因此,表 1 中 *BoxRefinement* 所在的行仅有状态空间为 2 维度的 MC 问题。

MC 问题

图 6 显示 $\alpha=0.9$, 折扣率值分别为 0.9、0.7 和 0.3, 采用 *BoxRefinement* 精化操作情况下情节数目与步数之间的关系图。该图显示随着折扣率的降低, 学习效果逐渐变差。当 $\gamma=0.9$ 时, 算法的收敛性较好, 且当步数小于 400 的情况下, 到达目的位置所需要的步数出现快速的下降, 后面的变化趋势较为平缓。当 $\gamma=0.7, 0.3$ 时, 图形显示算法在情节数为 2000 以内时并没有出现收敛的趋势, 且性能出现下降。但以下两方面的特点较为显著: (1) 随着 γ 值的降低, 步数达到情节步数上界, 即未达到目的位置的情节数目越来越多; (2) 对于达到目的位置的情节, 随着 γ 值的降低, 分布在步数较多范围的情节数目增加, 分布在步数较低范围的情节数目降低。

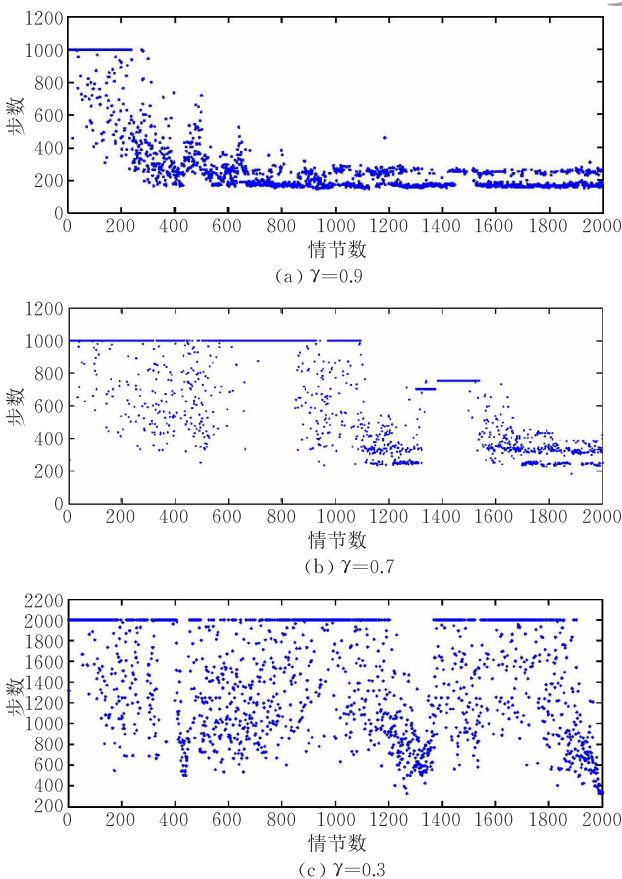
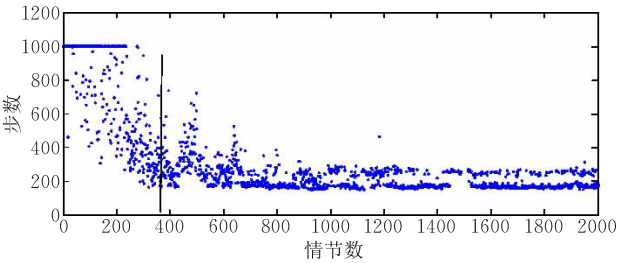


图 6 折扣率 γ 取不同值时情节数目与步数之间的关系图

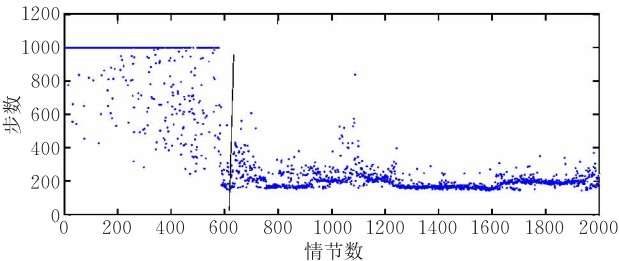
折扣率 γ 的大小直接决定后继状态动作对的价值奖赏对前驱状态动作对 Q 值的信用更新, 随着回溯时间步的增加, 更新值以 γ 的指数级衰减。导致算法 2 出现震荡的因素包括: (1) 抽象状态空间不准确, 即根据多面体抽象状态动作对 $(\mathcal{P}, a)$ 的 Q 值所做的决策依据不能准确体现 $\mathcal{P}$ 所包含的具体状态的真实情况; (2) 样本空间过小, Q 值函数不能充分收敛, 或样本信息没有被充分利用。根据算法 2 第 12 行代码块可以看出, 当分辨率参数 *accReward* 超过阈值时, 启动精化操作 *Refinement()*。当分辨率 *accReward* 超出阈值时, 此时, 为了避免不正确的信用后向传播, 在算法实现时仅进行了抽象的精化, 并没有更新任何资格迹和 Q 值, 这降低了样本的利用率。在学习的开始阶段, 由于初始抽象状态空间较小, 一般会出现频繁的精化操作, 此时样本的利用率更低。当折扣率 γ 值较小时, 后向信用分配衰减得更快, 在线产生的样本的利用率进一步降低。*Refinement()* 通过增大抽象状态空间, 提高了抽象的准确性, 但抽象空间不能过大。抽象空间增大, 也会导致正确的信用分配速度降低及大状态空间难以进行有效抽样。从提高样本利用率的角度而言, γ 值应设置得大些, 例如本实验中的 0.9, 但面对不同的学习问题, 还应考虑学习问题的环境模型的影响, 本次实验对 MC 问题的环境进行了简化, 没有考虑随机性, 因此抽样较准确地反映了环境的动力学性质, γ 值可以设置得尽量大, 而当环境模型具有很强的随机性时, γ 值的设置应偏小些。

图 7 显示了 $\gamma=0.9, \alpha$ 分别为 0.9、0.7 和 0.3, 采用 *BoxRefinement* 精化操作情况下情节数目与步数之间的关系图。该图显示 α 取不同的值对算法 2 性能的影响不像 γ 那么显著。当 $\gamma=0.9$ 时, α 取各种不同的值, 算法 2 的学习性能较好, 能快速地接近收敛, 而且震荡较小。这与 Q 值的更新规则相关, 即 Q 值的更新与 α 线性相关的。不过图 7 有一明显的特征与 α 相关: 随着 α 的减小, 初始震荡区域, 即垂直坐标轴与虚线之间的距离增大, 即当 α 较大时, 算法能较快地获得一个较好的策略, 且保持在较稳定的状态。

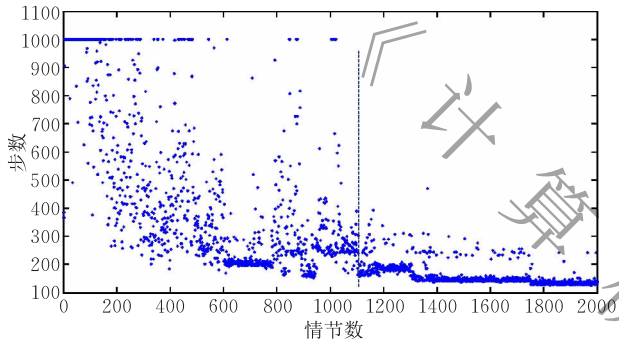
根据分辨率来确定是否实施精化操作需要具有环境奖惩模型的知识, 且与环境的动力学模型具有较大的关联性, 而 *LRefinement* 则没有这种局限性。图 8 和图 9 分别显示了学习参数 α 和 γ 取不同的值, 采用 *LRefinement* 精化操作的情况下算法 APDQ(λ) 的性能。图 8 和图 9 所显示的结果说



(a) $\alpha=0.9$

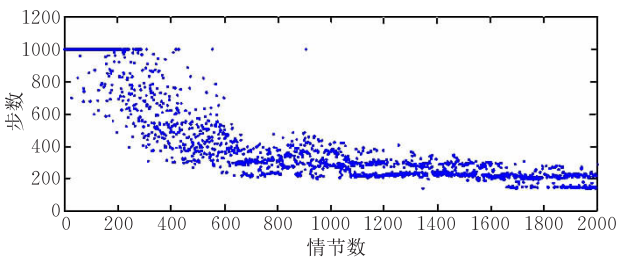


(b) $\alpha=0.7$

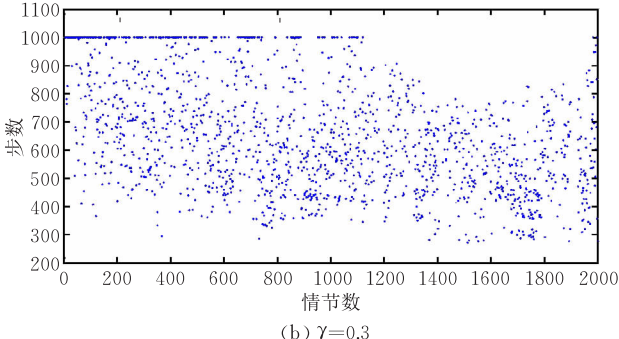


(c) $\alpha=0.3$

图 7 学习率 α 取不同值时情节数目与步数之间的关系图



(a) $\gamma=0.9$

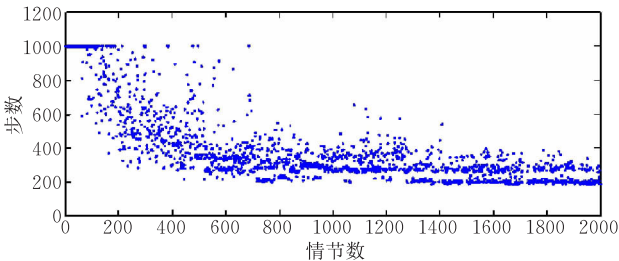


(b) $\gamma=0.3$

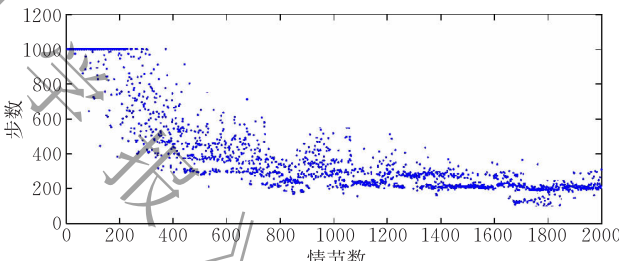
图 8 γ 取 0.9 和 0.3 时情节数目与步数之间的关系图

明在 *BoxRefinement* 和 *LRefinement* 的作用下, APDQ(λ) 算法体现出较一致的规律性, 例如 γ 取不同的值算法性能变化较大, 相对而言 α 对算法性能的

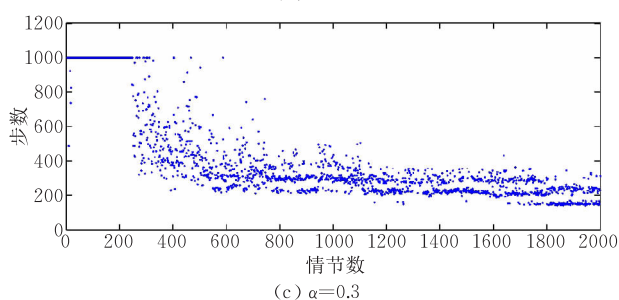
影响则没有那么显著. 但比较图 6(图 7) 和图 8(图 9) 可以看出, 在学习的初期, *LRefinement* 并没有显示出优势, 许多情况下, 性能不如 *BoxRefinement*, 这一点图 9(c) 表现尤其突出. 这种现象是由于抽象模型的不准确性造成的, 在学习的初始阶段, 抽象模型与原始模型差距较大, 当学习率过小时, 抽象状态空间精化操作频率过小, 这会导致某些抽象状态会反复出现在样本所诱导的抽象状态路径上, 由于 APDQ(λ) 采用的是累加资格迹的实现方式, 这样计算所得的抽象状态动作对的 Q 值会产生很大的误差, 从而导致学习性能恶化. 采用分辨率作为评估标准时, 由于当抽象状态路径中相邻的抽象状态相同时, 为了避免不正确的信用传播, 对资格迹和 Q 不进行更新, 从而可以减轻或避免这种现象的出现.



(a) $\alpha=0.9$



(b) $\alpha=0.7$



(c) $\alpha=0.3$

图 9 α 取 0.9, 0.7 和 0.3 时情节数目与步数之间的关系图

Acrobot 问题

为了更进一步显示 APDQ(λ) 的性能、可扩展性和适用性, 选取更复杂的具有 4 维度连续状态空间的 Acrobot 问题作为实验对象, 4.1 节显示单个维度值空间比 MC 大得多. 该问题的动力学模型是非线性的, 计算过程涉及积分计算, 本实验采用 Runge-Kutta 积分方法, 而不是 Euler 公式积分方

法,这使得该问题具有较大的难度. Acrobot 问题不适合采用基于分辨率的方法,经过对实验过程的观察发现,即使使分辨率阈值为 -2 ,在采用类似于 MC 问题类似的初始离散化方案的情况下,进行 500 次情节在线抽样,所得到的抽象状态数目小于 1000,难以得到用于近优策略计算所需的近似抽象模型. 这说明基于分辨率的精细化机制方法具有很大的局限性,受奖惩模型和动力学模型双重影响.

图 10 显示了 $\alpha=0.5, \epsilon=0.5$,采用 *LFR refinement* 精化操作情况下 γ 取值对算法 2 性能,由于 Acrobot 问题较难,因此情节数目的上限设为 5000. 关于 MC 和 Acrobot 问题的实验显示, α 和 γ 对 $APDQ(\lambda)$ 的作用规律具有一致性,因此由于篇幅的问题,图 10 和图 8 中删除了 $\gamma=0.5$ 的情形, α 为可变参数的情形则被整体删除. 图 10 所显示的结果说明 $APDQ(\lambda)$ 相关的原理和技术可有效地作用于更难更复杂的强化学习问题. 由于 Acrobot 问题的复杂性,出现明显收敛趋势的图 10(a)相比 MC 问题,具有更大的震荡;要得到更好的收敛结果,需要抽取更多的在线样本. 需要说明的是通过进一步降低方差阈值,虽然会使得抽象模型更准确,但会降低离散机制的泛化能力,增加样本需求. 图 10(b)虽然没有出现收敛的趋势,但从图中也可看出,实验样本所诱导的离散机制具有较好的泛化能力,大部分决策动作都能达到目标,且所需的步数都在 500 以下.

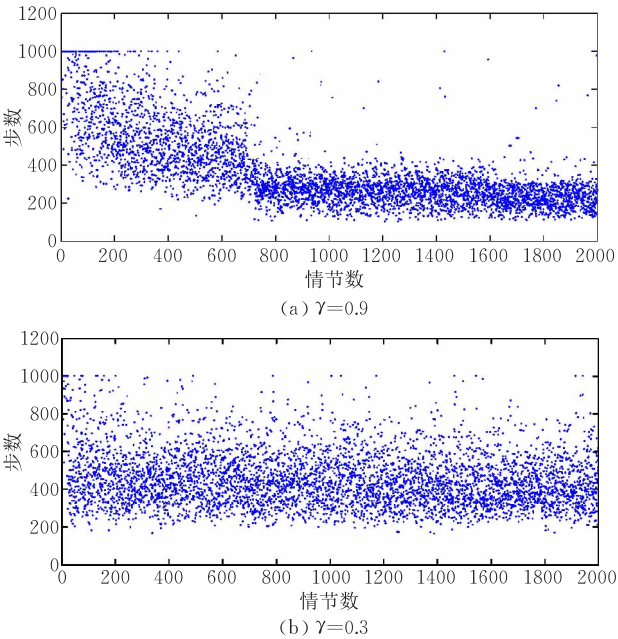


图 10 γ 取 0.9,0.3 时情节数目与步数之间的关系图

4.2.1.2 第 II 系列实验

第 II 系列实验说明与精化操作相关的参数对算

法 $APDQ(\lambda)$ 性能的影响. 表 2 详细给出了该系列实验参数配置和实验安排.

表 2 第 II 系列实验参数设置

固定参数	测试参数	<i>BoxRefinement</i> ($\square$)	<i>LFR refinement</i> ($\square$)
$\gamma=0.9$ $\alpha=0.9$	$\delta=-2, -4, -6$	MC(图 11)	
$\gamma=0.9$ $\alpha=0.5$	$\epsilon=0.8, 1, 1.2$	Acrobot(图 12)	

图 11 显示了采用不同的分辨率情况下采用 *BoxRefinement* 精化操作的学习效果. 实验结果表明,当分辨率为 $-2, -4, -6$ 时,算法性能较稳定,在情节数大于 400 时,处于较稳定的状态;分辨率越小,学习结果越不稳定,出现越多的震荡. 当分辨率为 -6 时,算法 2 可以得到稳定的近优策略,这说明 MC 问题 Q 值函数局部性较好.

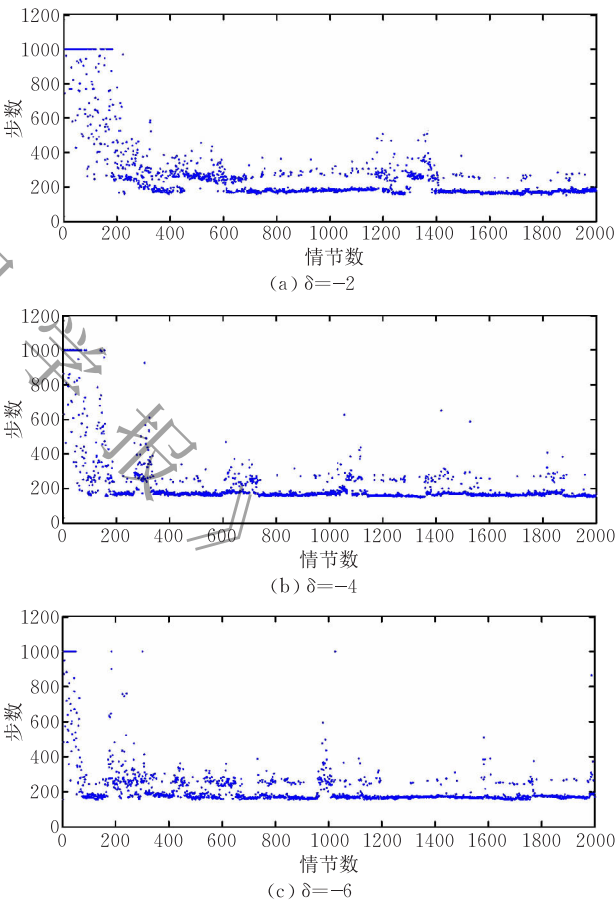


图 11 不同分辨率的学习效果图(MC 问题)

不同的 Q 值样本方差阈值对 *LFR refinement* 的影响,在学习的初始阶段并没有明显的规律性,这从图 12 可以看出;不过在中后期阶段,阈值为 0.8 的图 12(a)震荡幅度更小. 从图 12 和图 11 的比较可以看出,在一定范围内,算法 2 对分辨率和 Q 值方差域值具有较好的实验性.

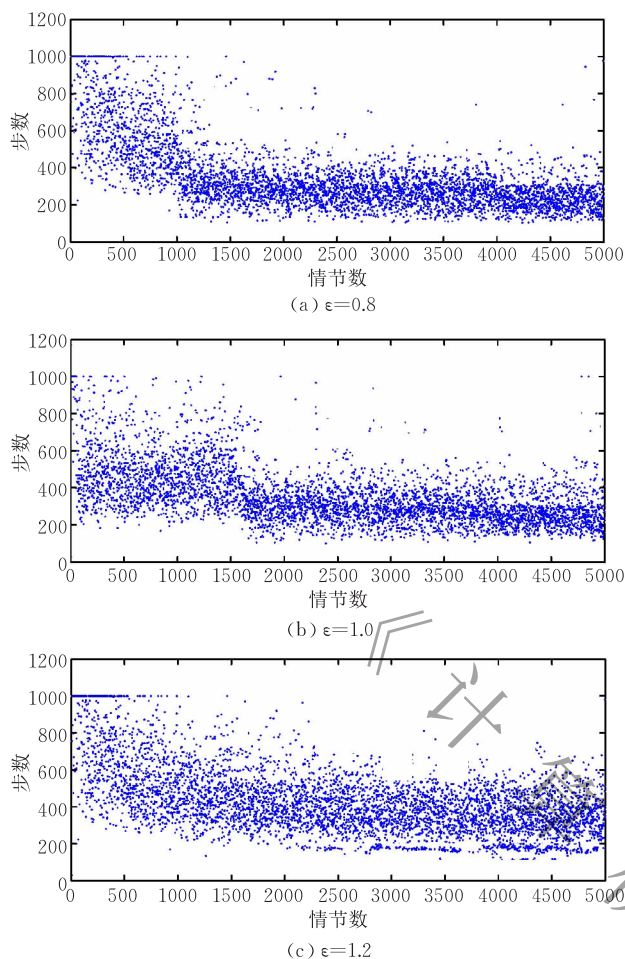


图 12 不同 Q 值方差阈值的学习效果图 (Acrobot 问题)

上面的几组实验说明,采用自适应离散化方法处理强化学习问题,算法性能与各种学习参数有着复杂的关系.由于抽象系统是原系统的一种近似,在一定量,甚至较大规模的抽样样本的情况下,震荡不可避免会在一定范围内存在.当采样的数目较多时,这种规律性的震荡现象很容易观察出来.在验证实验中固定参数设置为实验观测到的最优值或较优值,例如 $\gamma=0.9, \alpha=0.5$ 时,并没有针对 $\alpha-\gamma-r_p$ (r_p 为 δ 或 ϵ) 3 维和 $\alpha-\gamma$ 2 维笛卡尔乘积参数值空间测试 α, γ 和 r_p 对 APDQ(λ) 算法学习性能的影响.采用这种实验方案受已有的相关研究工作和 APDQ(λ) 算法测试过程的观测两方面因素的影响.

已有的相关研究工作表明针对不同的强化学习问题, $\alpha-\gamma$ 对许多典型的强化学习算法性能的影响是难以得到一致的,具有通用价值的定性和定量描述,甚至对于同一强化学习问题也是如此.针对 α 和 γ 的作用效果,往往从探索和利用,信用分配模式和算法收敛速度等方面分别进行讨论和说明.对于 APDQ(λ) 算法,它的行为决策模型具有抽象性,近

似性和动态变化的性质,并且在有限样本空间内 α 和 γ 的作用对算法的性能展现出较好的规律性,这从关于 MC 和 Acrobot 问题的验证实验中得到了较好的体现.文中对造成这一现象的原因进行了详细的学理上的讨论.尽管 MC 和 Acrobot 两种强化学习问题具有一定的代表性,但这种规律性是否具有严格形式上的描述和证明,是否能推广到别的强化学习问题还有待于进一步的实验和研究.在有限的样本空间及一定的精化阈值参数的条件下,为了提高样本的利用效率和抽象系统的泛化能力,一般令 $\alpha \geq 0.5, \gamma \geq 0.7$. 但值得强调的是提高样本的利用效率和提高抽象系统的泛化能力有时候是相向的目标,例如,当采用 *LFRRefinement* 精化操作时, α 和 γ 设置得越大,那么与抽象状态相关的 Q 值方差会被放大,从而更加频繁地调用 *LFRRefinement* 操作,导致泛化能力降低.图 9 显示当 γ 取 0.9 时, α 对算法性能影响的显著度明显小于 γ ,但是相关的实验显示,当 α 的值小于 0.15 时,算法性能出现较显著的下降.

4.2.2 对比实验

本文所提的方法具有一般性,可用于各种强化学习问题的求解.为了进一步说明它的可行性和性能,设计了相关的对比实验.对比实验采取两种典型的处理连续状态空间问题的技术,即瓦片编码和径向基函数近似(RFB).这两种方法的实现采用 Python 系统的开源强化学习包 rlpy 实现.两种实验中与强化学习有关的参数分别为 $\alpha=0.5, \gamma=\lambda=0.9$, 算法 $Q(\lambda)$ 的行为策略为 ϵ -Greedy ($\epsilon=0.01$).图 13(a) 情节数为 2000, 每情节的步数上限为 10000, 2 维瓦片特征为 8×8 , 共有 10 个瓦片;图 13(b) 采用高斯径向基函数, 100×100 的状态维分辨率,高斯径向基函数所涉及的均值和方差都由状态维分辨率确定的特征块的每个维度的宽度所产生.

图 13 的实验结果显示,采用瓦片编码方法的实验效果较差,而基于 RFB 方法则得到了稳定的近优策略.导致图 13(a) 实验结果的基本原因为: (1) 瓦片的特征和瓦片数量的设置不合理,瓦片之间的关系简单,仅通过平移来确定每个瓦片的位置; (2) 特征的表达方式简单,且缺乏自动调整的手段.图 13(b) 所得的近优策略与 APDQ(λ) 算法在实例研究中所得结果较为接近,但图 13(b) 收敛时震荡明显比采用 APDQ(λ) 方法的实验结果小.这是因为 APDQ(λ) 在执行过程中,抽象状态空间动态调整的

必然结果,但采用 RBF 的方法需要预设状态维的分辨率和径向基函数及有关的参数.实验中当改变状态维分辨率的值后,例如把 100×100 改为 20×20 或 1000×1000 ,实验效果急剧下降.采用 SARSA(λ) 算法的实验结果也类似.另外,在第 1 节的相关工作中,具有对近年一些工作的讨论和比较.

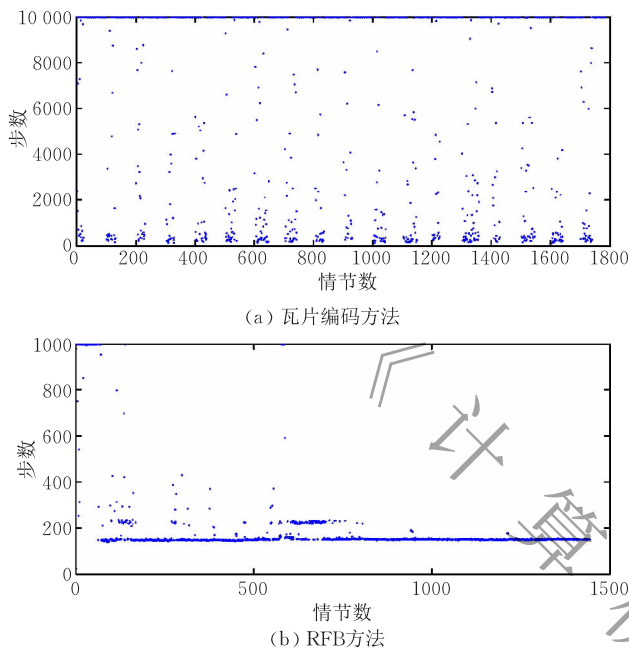


图 13 关于 MC 问题的对比实验

5 结 论

本文提出了基于凸多面体抽象域的自适应离散化强化学习技术,用于对具有连续状态空间和连续动作空间的强化学习问题进行求解.基于这种思想,实现了 APDQ(λ)学习器.通过以 MC 和 Acrobot 两个强化学习问题的实例研究,初步说明了各种参数对算法性能的作用规律;说明了作为一种具有一般性意义的强化学习技术的可行性和有效性.下一步的工作将重点围绕两方面展开:一是进一步优化算法,提供效率,例如如何在抽象状态空间上利用哈希技术,提高计算具体状态和抽象状态的执行效率,以及进行更多具有高维度连续状态空间的实例研究;二是结合基于多面体状态表示和基于多面体抽象状态值函数表示的方法,以便更加有效地处理具有更高连续维度的强化学习问题.

致 谢 匿名评审专家为本文工作的改进提出了许多宝贵的意见,在此致以诚挚的感谢!

参 考 文 献

- [1] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 1996, 4(5): 247-331
- [2] Sutton R S, Barto A G. *Reinforcement Learning: An Introduction*. Cambridge, USA: MIT Press, 1998
- [3] Gao Yang, Chen Shi-Fu, Lu Xin. Research on reinforcement learning technology: A review. *Acta Automatica Sinica*, 2004, 30(1): 86-100(in Chinese)
(高阳, 陈世福, 陆鑫. 强化学习综述. *自动化学报*, 2004, 30(1): 86-100)
- [4] Singh S P, Jaakola T, Jordan M I. Reinforcement learning with soft state aggregation//*Proceedings of the 7th International Conference on Neural Information Processing Systems*. Denver, USA, 1995: 361-368
- [5] Chen Zong-Hai, Wen Feng, Nie Jian-Bin, Wu Xiao-Shu. A reinforcement learning method based on node-growing k -means clustering algorithm. *Journal of Computer Research and Development*, 2006, 43(4): 661-666(in Chinese)
(陈宗海, 文锋, 聂建斌, 吴晓曙. 基于节点生长 k -均值聚类算法的强化学习. *计算机研究与发展*, 2006, 43(4): 661-666)
- [6] Latorre A, Pena J M, Muelas S. Learning hybridization strategies in evolutionary algorithms. *Intelligent Data Analysis*, 2010, 14(3): 333-386
- [7] Akiyama T, Hachiya H, Sugiyama M. Efficient exploration through active learning for value function approximation in reinforcement learning. *Neural Networks*, 2010, 23(5): 639-686
- [8] Langlois M, Sloan R H. Reinforcement learning via approximation of the Q -function. *Journal of Experimental and Theoretical Artificial Intelligence*, 2010, 22(3): 219-253
- [9] Zaragoza J H, Morales E F. Relational reinforcement learning with continuous actions by combining behavioural cloning and locally weighted regression. *Journal of Intelligent Learning Systems and Applications*, 2010, 2(2): 69-79
- [10] Mohan S, Laird J E. Relational reinforcement learning in infinite mario//*Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Atlan, USA, 2010: 1953-1956
- [11] Liu Quan, Gao Yang, Chen Dao-Xu, et al. A logic reinforcement learning method based on heuristic list. *Journal of Computer Research and Development*, 2008, 45(11): 1824-1830(in Chinese)
(刘全, 高阳, 陈道蓄等. 一种基于启发式轮廓表的逻辑强化学习. *计算机研究与发展*, 2008, 45(11): 1824-1830)
- [12] Wang Wen-Xi, Xiao Shi-De, Meng Xiang-Yin, et al. Model and architecture of hierarchical reinforcement learning based on Agent. *Journal of Mechanical Engineering*, 2010, 46(2): 76-82(in Chinese)

- (王文玺, 肖世德, 孟祥印等. 基于 Agent 的递阶强化学习模型与体系结构. 机械工程学报, 2010, 46(2): 76-82)
- [13] Shen Jing, Gu Guo-Chang, Liu Hai-Bo. A survey of hierarchical reinforcement learning. *Pattern Recognition & Artificial Intelligence*, 2005, 18(5): 574-581(in Chinese)
(沈晶, 顾国昌, 刘海波. 分层强化学习研究综述. 模式识别与人工智能, 2005, 18(5): 574-581)
- [14] Kozlova O, Sigaud O, Meyer C. TexDYNA: Hierarchical reinforcement learning in factored MDPs//*Proceedings of the 11th International Conference Simulation of Adaptive Behavior from Animals to Animates*. Paris, France, 2010: 489-500
- [15] Tsitsilis J N, Van R B. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 1997, 42(5): 674-690
- [16] Maei H R, Szepesvari C, Bhatnagar S. Convergent temporal-difference learning with arbitrary smooth function approximation //*Proceedings of the 22nd International Conference on Neural Information Processing Systems*. Vancouver, Canada, 2009: 1204-1212
- [17] Wang Xue-Song, Zhang Yi-Yang, Cheng Yu-Hu. Reinforcement learning for continuous spaces based on Gaussian process classifier. *Acta Electronica Sinica*, 2009, 37(6): 1153-1158 (in Chinese)
(王雪松, 张依阳, 程玉虎. 基于高斯过程分类器的连续空间强化学习. 电子学报, 2009, 37(6): 1153-1158)
- [18] Baird L C. Residual algorithm: Reinforcement learning with function approximation//*Proceedings of the 12th International Conference on Machine Learning*. Tahoe, USA, 1995: 30-37
- [19] Lu Xin, Gao Yang, Li Ning, Chen Shi-Fu. Research on a reinforcement learning algorithm based on neural network. *Journal of Computer Research and Development*, 2002, 39(8): 981-985(in Chinese)
(陆鑫, 高阳, 李宁, 陈世福. 基于神经网络的强化学习算法研究. 计算机研究与发展, 2002, 39(8): 981-985)
- [20] Wang Xue-Ning, Xu Xin, Wu Tao, He Han-Gen. The optimal reward baseline for policy-gradient reinforcement learning. *Chinese Journal of Computers*, 2005, 28(6): 1021-1027 (in Chinese)
(王学宁, 徐昕, 吴涛, 贺汉根. 策略梯度强化学习中的最优回报基线. 计算机学报, 2005, 28(6): 1021-1027)
- [21] Sutton R S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988, 3(1): 9-44
- [22] Jaakkola T, Jordan M I, Singh S P. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computing*, 1994, 6(6): 1185-1201
- [23] Dayan P D. The convergence of TD(λ) for general λ . *Machine Learning*, 1992, 8(3): 341-362
- [24] Munos R. A study of reinforcement learning in the continuous case by the means of viscosity solutions. *Machine Learning*, 2000, 40(3): 265-363
- [25] Waltz M D, Fu K S. A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control*, 1965, 10: 390-397
- [26] Hinton G E. Distributed representations. *Carnegi Mellon University, Pittsburgh, USA: Technique Report CMU-CS-84-157*, 1984
- [27] Shewchun J, Dean T. Towards learning time-varying functions with high input dimensionality//*Proceedings of the 5th International Symposium on Intelligent Control*. Philadelphia, USA, 1990: 383-390
- [28] Sutton R S. Generation in reinforcement learning: Successful examples using sparse coarse coding//*Proceedings of the 9th International Conference on Neural Information Processing Systems*. Denver, USA, 1996: 1038-1081
- [29] Gao Yang, Hu Jing-Kai, Wang Ben-Nian, Wang Dong-Li. Elevator group control using reinforcement learning with CMAC. *Acta Electronica Sinica*, 2007, 35(2): 362-365 (in Chinese)
(高阳, 胡景凯, 王本年, 王东黎. 基于 CMAC 网络强化学习的电梯群控调度. 电子学报, 2007, 35(2): 362-365)
- [30] Lee I S K, Lau H Y K. Adaptive state space partitioning for reinforcement learning. *Artificial Intelligence*, 2004, 17(6): 577-588
- [31] Pyeatt L D, Howe A E. Decision tree function approximation in reinforcement learning. *Colorado State University, Fort Collins, USA: Technical Report CS-98-112*, 1998
- [32] Shawe-Taylor J, Cristianini N. *Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge University Press, 2004
- [33] Deisenroth M P, Rasmussen C E, Peters J. Gaussian process dynamic programming. *Neurocomputing*, 2009, 72(7): 1508-1524
- [34] Nikolaos T, Christos D, Konstantinos B. Linear Bayesian reinforcement learning//*Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. Beijing, China, 2013: 1721-1725
- [35] Qin Zhi-Wei, Li Wei-Chang, Janoos F. Sparse reinforcement learning via convex optimization//*Proceedings of the 31st International Conference on Machine Learning*. Beijing, China, 2014: 424-432
- [36] Bilal P, Matthieu G, Olivier P. Difference of convex functions programming for reinforcement learning//*Proceedings of the 27th International Conference on Neural Information Processing Systems*. Montreal, Canada, 2014: 2519-2527
- [37] Cousot P, Cousot R. Abstract interpretation: A unified lattice model for the static analysis of programs by construction or approximation of fixpoints//*Proceedings of the 4th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. New York, USA, 1977: 238-252
- [38] Cousot P. Verification by abstract interpretation//*Proceedings of the Verification: Theory and Practice*. Sicily, Italy, 2003: 3-7
- [39] Dams D, Gerth R, Grumberg O. Abstract interpretation of reactive systems. *ACM Transactions on Program Language System*, 1997, 19(2): 253-291

- [40] Mundhenk M, Goldsmith J, Lusena C, Allender E. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 2000, 12(3): 633-644
- [41] Panangaden P. *Labeled Markov Processes*. London, UK: Imperial College Press, 2009
- [42] Dolcetta C, Ishii H. Approximate solutions of the Bellman equations deterministic control theory. *Applied Mathematics and Optimization*, 1984, 11(1): 161-181
- [43] Aguilar C O, Krener A J. Numerical solutions to the Bellman equations of optimal control. *Journal of Optimization Theory & Applications*, 2014, 160(2): 527-552
- [44] Jaakkola T, Singh S P, Jordan M I. Reinforcement learning algorithm for partially observable Markov decision problems // *Proceedings of the 7th International Conference on Neural Information Processing Systems*. Denver, USA, 1995: 345-352
- [45] Watkins C, Dayan P. *Q learning*. *Machine Learning*, 1992, 8(3): 279-292
- [46] Szepesvari C. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation // *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. Vancouver, Canada, 2009: 1204-1212
- [47] Cousot P, Cousot R. A Galois connection calculus for abstract interpretation. *ACM SIGPLAN Notices*, 2014, 49(1): 3-4
- [48] Bagnara R, Hill P M, Zaffanella E. The Parma polyhedra library: Toward a complete set of numerical abstraction for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 2008, 72(6): 3-21
- [49] Cerny P, Henzinger T H, Radhakrishna A. Quantitative abstraction refinement. *ACM SIGPLAN Notices*, 2013, 48(1): 115-128
- [50] Bagnara R, Hill P M, Zaffanella E. Exact join detection for convex polyhedra and other numerical abstractions. *Computational Geometry*, 2009, 43(5): 453-473
- [51] Jiang Guo-Fei, Gao Hui-Qi, Wu Cang-Pu. Convergence of discretization procedure in Q-learning. *Control Theory and Application*, 1999, 16(2): 194-198(in Chinese)
(蒋国飞, 高慧琪, 吴沧浦. Q 学习算法中网格离散化方法的收敛性分析. *控制理论与应用*, 1999, 16(2): 194-198)



CHEN Dong-Huo, born in 1974, Ph.D., lecturer. His main research interests include program verification, model checking, and automated reasoning.

LIU Quan, born in 1969, Ph.D., professor, Ph.D. supervisor. His main research interests include intelligence

information processing, automated reasoning, and machine learning.

ZHU Fei, born in 1978, Ph.D., associate professor. His main research interests include reinforcement learning, text mining, and pattern recognition.

JIN Hai-Dong, born in 1973, M.S., lecturer. His main research interests include reinforcement learning and deep learning.

Background

Reinforcement learning is a branch of machine learning, which is concentrated on how software agents ought to take actions according to the state of an environment so as to maximize some quantitative metric, named the long-term cumulative reward. Thus, reinforcement learning is particularly well suited to problems which include a long-term versus short-term reward trade-off, such as robot control, elevator scheduling, path planning, web information search and portfolio management. But in practice, faced on the reinforcement learning problems with continuous state space, all kinds of reinforcement learning technology is challenged by the curse of dimensionality, also named the state explosion problem.

Two existing methods attacking the curse of dimensionality are the discretization of continuous state space and function approximation. Theoretically, function approximation can deal with reinforcement learning problems with any size of

state space under condition that a set of basis functions or kernel functions encoding the features of state space is reasonably set. But function approximation does not scale well due to the requirement of deep insight into domain knowledge when setting the basis or kernel functions set and the high computation cost. The algorithm based on the discretization of continuous state space performs on a lookup table, which has the advantages of straightforward principle, the implementation with concise data structure and the lightweight computation. But the existing research need further be advanced in some respects, typically including the formal specification of abstract state with rigid syntax and the automated tool and programming support of complex reasoning and computation, and adaptive refinement of discretization mechanism. Considering the above-mentioned research backgrounds, the paper presents a presentation of abstract state based on

convex polyhedra domain, and the convex polyhedra domain has enough expressiveness power for solving reinforcement learning problems with continuous space. Moreover, in terms of abstract interpretation in formal method, a solid framework of discretization mechanism and viable refinement functions is formally built. A variant $APDQ(\lambda)$ of $Q(\lambda)$ integrating adaptive refinement mechanism is designed and implemented. Summarily, $APDQ(\lambda)$ embodies some attributes of off-policy, eligibility trace based backward temporal difference assignment and the convergence guarantee besides being easily scaled to the most reinforcement learning problems. The case studies about mountain car (MC) and acrobatic robot (Acrobot) with the continuous state spaces showed that the performance of $APDQ(\lambda)$ meets expectations and the ideas in the paper would be a promising research

aspect for challenging the continuous state space, even continuous action space in reinforcement learning.

The research work in the paper is supported by the National Natural Science Foundation of China (61272005, 61303108, 61373094, 61472262, 61502323, 61502329), the Natural Science Foundation of Jiangsu (BK2012616), the High School Natural Foundation of Jiangsu (13KJB520020), the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172014K04), the Suzhou Industrial Application of Basic Research Program Part (SYG201422), the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1524) and the China Scholarship Council (201606920013).

