DOI: 10.11992/tis.201706031
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20171021.1350.008.html

分层强化学习综述

周文吉,俞扬
(南京大学 软件新技术国家重点实验室,江苏 南京 210023)

摘 要: 强化学习(reinforcement learning)是机器学习和人工智能领域的重要分支,近年来受到社会各界和企业的广泛关注。强化学习算法要解决的主要问题是,智能体如何直接与环境进行交互来学习策略。但是当状态空间维度增加时,传统的强化学习方法往往面临着维度灾难,难以取得好的学习效果。分层强化学习(hierarchical reinforcement learning)致力于将一个复杂的强化学习问题分解成几个子问题并分别解决,可以取得比直接解决整个问题更好的效果。分层强化学习是解决大规模强化学习问题的潜在途径,然而其受到的关注不高。本文将介绍和回顾分层强化学习的几大类方法。

关键词: 人工智能; 机器学习; 强化学习; 分层强化学习; 深度强化学习; 马尔可夫决策过程; 半马尔可夫决策过程; 维度灾难
中图分类号: TP391 文献标志码: A 文章编号: 1673-4785(2017)05-0590-05

中文引用格式: 周文吉,俞扬.分层强化学习综述[J]. 智能系统学报, 2017, 12(5): 590-594.
英文引用格式: ZHOU Wenji, YU Yang. Summarize of hierarchical reinforcement learning [J]. CAAI transactions on intelligent systems, 2017, 12(5): 590-594.

Summarize of hierarchical reinforcement learning

ZHOU Wenji, YU Yang
(National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

Abstract: Reinforcement Learning (RL) is an important research area in the field of machine learning and artificial intelligence and has received increasing attentions in recent years. The goal in RL is to maximize long-term total reward by interacting with the environment. Traditional RL algorithms are limited due to the so-called curse of dimensionality, and their learning abilities degrade drastically with increases in the dimensionality of the state space. Hierarchical reinforcement learning (HRL) decomposes the RL problem into sub-problems and solves each of them to improve learning ability. HRL offers a potential way to solve large-scale RL, which has received insufficient attention to date. In this paper, we introduce and review several main HRL methods.

Keywords: artificial intelligence; machine learning; reinforcement learning; hierarchical reinforcement learning; deep reinforcement learning; Markov decision process; semi-Markov decision process; dimensional curse

强化学习要研究的问题是智能体(agents)如何在一个环境(environment)中学到一定的策略(policy),使得长期的奖赏(reward)最大。但是传统的强化学习方法面临着维度灾难,即当环境较为复杂或者任务较为困难时,agent 的状态(state)空间过大,会导致需要学习的参数以及所需的存储空间急速增长,强化学习难以取得理想的效果。为了解决维度灾难,研究者提出了分层强化学习(hierarchical reinforcement learning, HRL)。HRL 的主要目标是将复杂的问题分解成多个小问题,分别解决小问题从而达到解决原问题的目的[1]。近些年来,人们认为分层强化学习基本可以解决强化学习的维度灾难问题[2-3],转而将研究方向转向如何将复杂的问题抽象成不同的层级,从而更好地解决这些问题[4]。

现在已有的一些分层学习大致可以分为 4 大类,分别是基于选项(option)的强化学习、基于分层

收稿日期: 2017-06-09. 网络出版日期: 2017-10-21.
基金项目: 国家自然科学基金项目(61375061); 江苏省自然科学基金项目(BK20160066).
通信作者: 俞扬. E-mail: yuy@nju.edu.cn.

抽象机(hierarchical of abstract machines) 的分层强化学习、基于 MaxQ 值函数分解 (MaxQ value function decomposition) 的分层强化学习，以及端到端的(end to end) 分层强化学习。

1　背景知识

在本节中，我们主要介绍强化学习、马尔可夫决策过程(Markov decision process) 和半马尔克夫决策过程(Semi-Markov decision process) 的定义以及相关的背景知识。

1.1　强化学习与马尔可夫决策过程

强化学习是机器学习和人工智能中一个重要的领域，主要研究的问题是 agent 如何通过直接与环境交互来学习策略，使得长期的奖赏最大。强化学习有一些特点，比如无监督学习，奖赏的反馈有延迟，agent 选择的动作会影响之后接收的数据等。

大多数关于强化学习的研究都是建立在马尔可夫决策过程(MDP) 的基础上，MDP 可以表示为一个五元组 $\langle S,A,P,R,\gamma\rangle$。其中，$S$ 为状态(state) 的有限集合，集合中某个状态表示为 $s\in S$; A 为动作(action) 的有限集合，集合中某个动作表示为 $a\in A$，A 为状态 s 下可执行的动作集合; P 为状态转移方程，$P(s'|s,a)$ 表示在状态 s 执行动作 a 后将以 $P(s'|s,a)$ 的概率跳转到状态 s'; R 为奖赏函数(reward function); γ 为折损系数(discount factor)，$0\leqslant\gamma\leqslant 1$。假设一个 agent 观察到自己的状态 s，此时它选择一个动作 a，它会得到一个即时的奖赏 $r_s^a=R(s,a)$，然后以 $P(s'|s,a)$ 的概率达到下一个状态 s'。马尔可夫决策过程有马尔可夫性，即系统的下个状态只与当前状态有关，与之前的状态无关。当马尔可夫决策过程中作出决策时，只需要考虑当前的状态，而不需要历史数据，这样大大降低了问题的复杂度。

强化学习需要 agent 学习到一个策略 $\pi: S\times A\rightarrow[0,1]$，通过 $\pi(s,a)$ 的值来指导 agent 进行动作的选择。给定一个策略 π 和一个状态 s，V_s^π 表示从 s 开始按照策略 π 进行选择可以得到的期望累积奖赏。我们将 V 称作值函数(value function)，其具体的数学定义为 $V^\pi(s)=E\{r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots+r^n r_{t+n}\,|\,s=s_t,\pi\}$。强化学习的目标是学到一个最优的策略 π^*，最大化每一个状态下的 V 值，此时的最优值函数记作 V^*。

除了值函数，动作-值函数(action-value function) 也在强化学习中扮演着重要的角色，记作 $Q^\pi(s,a)$，表示给定一个策略 π，在状态 s 上执行动作 a 可以得到的期望累积奖赏。我们也将 $Q^\pi(s,a)$ 叫作 Q 函数，其具体的数学定义表示为

$$Q^\pi(s,a)=E\{r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots|\,s_t=s,a_t=a,\pi\}$$

同样的，我们也希望通过学习到一个最优的 Q 函数 Q^*，使 agent 可以直接通过 Q 函数来选择当前状态下应该执行的动作。

经过多年的研究，已经出现一些算法，致力于解决传统的强化学习问题，比如 Q-Learning、蒙特卡洛方法(Monte-Carlo learning)、时序差分方法(temporal-difference learning) 等。其中 Q-Learning 方法常常在分层强化学习中被使用。Q-Learning 通过不断迭代更新 Q 函数的值来逼近最优的 Q^*。其迭代式如下

$$Q_{k+1}(s,a)=(1-\alpha_k)Q_k(s,a)+\alpha_k(r+\gamma\max_{a'\in A_{s'}}Q_k(s',a'))$$

其中 s' 表示下一个状态。

1.2　半马尔可夫决策过程

马尔可夫决策过程中，选择一个动作后，agent 会立刻根据状态转移方程 P 跳转到下一个状态，而在半马尔可夫决策过程(SMDP) 中，当前状态到下一个状态的步数是个随机变量 τ，即在某个状态 s 下选择一个动作 a 后，经过 τ 步才会以一个概率转移到下一个状态 s'。此时的状态转移概率是 s 和 τ 的联合概率 $P(s',\tau|s,a)$。根据 τ 的定义域不同，SMDP 所定义的系统也有所不同。当 τ 的取值为实数值，则 SMDP 构建了一个连续时间-离散事件系统(continuous-time discrete-event system) [5]; 而当 τ 的取值为正整数，则是一个离散时间 SMDP (discrete-time SMDP) [6]。出于简单考虑，绝大部分分层强化学习都是在离散时间 SMDP 上进行讨论。

2　分层强化学习方法

分层强化学习是将复杂的强化学习问题分解成一些容易解决的子问题(sub-problem)，通过分别解决这些子问题，最终解决原本的强化学习问题[7-9]。常见的分层强化学习方法可以大致分为四大类，分别为基于选项(option) 的强化学习、基于分层抽象机(hierarchical of abstract machines) 的分层强化学习、基于 MaxQ 函数分解 (MaxQ value function decomposition) 的分层强化学习，以及端到端的(end to end) 的分层强化学习。本节将对它们逐一进行探讨。

2.1　基于选项的分层强化学习

“选项”(option) 的概念在 1999 年由 Sutton 等人提出[10]，是一种对动作的抽象。一般的，option

可表示为一个三元组 $\langle I,\pi,\beta\rangle$。其中，$\pi: S\times A\to[0,1]$ 表示此option中的策略；$\beta: S\to[0,1]$ 表示终止条件，$\beta(s)$ 表示状态 s 有 $\beta(s)$ 的概率终止并退出此 option；$I\subseteq S$ 表示 option 的初始状态集合。option $\langle I,\pi,\beta\rangle$ 在状态 s 上可用，当且仅当 $s\in I$。当 option 开始执行时，agent 通过该 option 的 π 进行动作选择直到终止。值得注意的是，一个单独的动作 a 也可以是一个 option，通常被称作 one-step option，$I=\{s: a\in A_s\}$，并且对任意的状态 s 都有 $\beta(s)=1$。

基于 option 的分层强化学习的过程如下：假设 agent 当前在某个状态，选择一个 option，通过这个 option 的策略，agent 选择了一个动作或者另一个 option。若选择了一个动作，则直接执行转移到下一个状态；若选择了另一个 option，则用选择的新 option 继续选择，直到最后得出一个动作。

为了使用 option 来解决分层强化学习问题，我们还需要定义一个更高层级的策略 $\mu: S\times O_s\to[0,1]$。其中，O 表示所有 option 的集合，而 O_s 表示状态 s 下可用的 option 的集合；$\mu(s,o)$ 表示在状态 s 时以 $\mu(s,o)$ 的概率选择 o 作为当前的 option。此时的 Q 函数定义为

$$Q^{\mu}(s,o)=E\{r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots\mid o_r=o,s_t=s\}$$

此时的 Q-Learning 的更新公式为

$$Q_{k+1}(s_t,o_t)=(1-\alpha_k)Q_k(s_t,o_t)+\alpha_k(r_t+\gamma r_{t+1}+\cdots+\gamma^{\tau-1}r_{t+\tau}+\gamma^{\tau}\max_{o'\in O_{s'}}Q_k(s_{t+\tau},o'))$$

其中，α_k 为第 k 轮迭代时的学习率，τ 表示 option o 在执行 τ 步之后在状态 $s_{t+\tau}$ 停止，而 o' 为在 o 执行结束后的下一个 option。可以注意到，当所有的 option 均为 one-step option 时，这个 Q-Learning 就退化为普通的 Q-Learning 过程。

2.2 基于分层抽象机的分层强化

分层抽象机(hierarchies of abstract machines，HAMs，)是 Parr 和 Russell 提出的方法[11]。和 option 的方法类似，HAMs 的方法也是建立在 SMDP 的理论基础之上的。HAMs 的主要思想是将当前所在状态以及有限状态机的状态结合考虑，从而选择不同的策略。

令 M 为一个有限 MDP，S 为状态集合，A 为动作集合。$\{H_i\}$ 为一个有限状态机的集合，其中每个有限状态机 H_i 都有一个状态集合 S_i、一个概率转移方程 δ_i 以及一个随机函数 $f_i: S\to S_i$。每个状态机都有 4 种类型的状态，即动作(action)、调用(call)、选择(choice)以及停止(stop)。action 类型的状态会根据状态机的具体状态执行一个 MDP 中的动作。在 call 类型的状态时，当前状态机 H_i 将被挂起，开始初始化下一个状态机 H_j，即把 H_j 的状态设置为 $f_j(s_t)$，其中 j 的值根据 m_t^i 得出，m_t^i 表示第 i 个状态机在时刻 t 时的状态。choice 类型的状态则是非确定性地选择当前状态机的下一个状态。stop 状态则是停止当前状态机的活动，恢复调用它的状态机的活动，同时 agent 根据之前选择的动作进行状态转移并得到奖赏。如果在这个过程中没有选择出动作，例如某个状态机 H_i 刚被调用就被随机函数 f_i 初始化到了一个 stop 状态以至于返回时并没有选出要执行的动作，则 M 保持当前的状态。

HAMs 也可以通过改进 Q-Learning 算法进行学习。对于一个马尔可夫决策过程 M 和任意一个状态机 H，$H\circ M$ 是一个 MDP[11]，其中状态集合为 $S\times S_H$，动作集合为 $A\times A_H$。只有当 H 的状态是 choice 类型的状态时，$H\circ M$ 才需要进行决策，其他状态下都可以根据状态机的状态自动进行状态转移，所以实际上 $H\circ M$ 是个 SMDP。因此我们需要维护的 Q 函数为 $Q([s_c,m_c],a_c)$，其中，c 表示 $H\circ M$ 中需要作出选择的状态的下标，$[s_c,m_c]$ 被称作选择点(choice point)。此时 Q-Learning 的更新公式为

$$Q_{k+1}([s_c,m_c],a_c)=(1-\alpha_k)Q_k([s_c,m_c],a_c)+\alpha[r_t+\gamma r_{t+1}+\cdots+\gamma^{\tau-1}r_{t+\tau-1}+\gamma^{\tau}\max_{a'}Q_k([s'_c,m'_c],a')]$$

其中，τ 为由 s_c 到 s'_c 所经过的步数。

2.3 基于 MaxQ 值函数分解的分层强化学习

MaxQ 值函数分解(MaxQ value function decomposition)，是由 Dietterich 提出的另外一种分层强化学习的方法[12]。首先将一个马尔可夫决策过程 M 分解成多个子任务$\{M_0,M_1,\cdots,M_n\}$，M_0 为根子任务，解决了 M_0 就意味着解决了原问题 M。对于每一个子任务 M_i，都有一个终止断言(termination predicate) T_i 和一个动作集合 A_i。这个动作集合中的元素既可以是其他的子任务，也可以是一个 MDP 中的 action。一个子任务的目标是转移到一个状态，可以满足终止断言，使得此子任务完成并终止。我们需要学到一个高层次的策略 $\pi=\{\pi_0,\cdots,\pi_n\}$，其中 π_i 为子任务 M_i 的策略。

令 $V(i,s)$ 表示子任务 i 在状态 s 的值函数，即该子问题从状态 s 开始一直按照某个策略执行最终达到终止状态的期望累计奖赏。类似的，令 $Q(i,s,j)$ 为子任务 i 在状态 s 执行动作 j 之后按照某个策略执行直到达到终止状态的期望累计奖赏，可以表示为

$$Q(i,s,j)=E(r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots\mid s_t=s,\pi)$$

假设选择的动作 j 一共执行了 τ 步才返回，那么我们可以把 Q 函数写成

$$Q(i,s,j) = E\{\sum_{u=0}^{\tau-1}\gamma^u r_{t+u} + \sum_{u=\tau}^{\infty}\gamma^u r_{t+u} \mid s_{t=s},\pi\}$$

其中右边的第 1 项实际上是 $V(j,s)$,第 2 项叫作完成函数(completion function),记作 $C(i,s,j)$。则 Q 函数的贝尔曼方程可以写为

$$Q(i,s,j) = \sum_{s',\tau} P(s',\tau \mid s,j)\,(V(j,s) + \gamma^{\tau}\max_{j'}Q(i,s',j')) = V(j,s) + C(i,s,j)$$

当选择的动作 j 完成后,得到下一个状态 s' 以及做完这个动作经过的时间 τ,则可更新完成函数

$$C_{t+1}(i,s,j) = (1-\alpha_t)\,C_t(i,s,j) + \alpha_t\gamma^{\tau}\,[\max_{a'}V(a',s') + C_t(i,s',a')]$$

这样也就更新了 Q 函数。

图 1 为利用 MaxQ 方法解决 taxi problem 的任务划分示意图[12]。

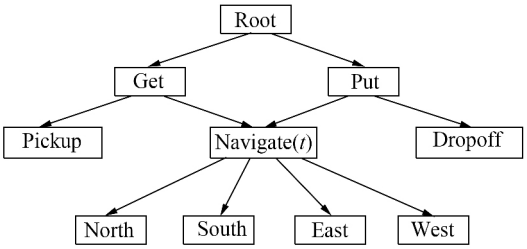

图 1　出租车问题的任务图

Fig.1　Task graph for the taxi problem

出租车问题是指一个出租车 agent 需要到特定位置接一位乘客并且把他送到特定的位置让其下车。一共有 6 个动作,分别是上车(pick up)、下车(drop off),以及向东南西北四个方向开车的动作。这里使用 MaxQ 方法,将原问题分解成了 get 和 put 两个子任务,这两个子任务又进行分解,get 分解成一个基本动作 pick up 和一个子任务 navigate,而 put 也分解成了一个基本动作 drop off 和一个子任务 navigate。子任务 navigate(t) 表示 t 时刻应该开车的方向。对于这个强化学习问题,agent 首先选择 get,然后 get 子问题 navigate,直到到达乘客所在地,然后 get 选择 pick up 动作,乘客上车。之后 agent 选择 put 子任务,put 子任务选择 navigate,直到到达乘客目的地,之后 put 子任务选择 drop off 动作,乘客下车,任务完成。

2.4　端到端的的分层强化学习

上述的几种方法,都需要人工来做很多工作,比如人工进行 option 的选取,人工进行 HAMs 的构建,人工划分子任务等。人工设计不仅耗时耗力,并且会直接影响最终强化学习结果的好坏。近些年来人们关注如何让 agent 自己学到合理的分层抽象,而非人为进行划分和指定。

有人提出利用蚁群算法启发式地寻找合理的划分点[13]。作者利用蚁群算法根据信息素的变化程度寻找“瓶颈”(bottle neck),瓶颈像一座桥梁一样连接着问题空间中不同的连续区域。图 2 为一个 Grid Word 问题,agent 需要从状态 s 出发到达状态 g。通过蚁群算法分析信息素的变化程度找出瓶颈在两个房间的窄门处,即图 2 中的状态 v 附近。

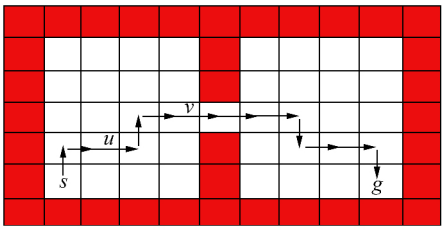

图 2　通过蚁群算法找到从 s 到 g 的最短路径

Fig.2　Shortest path between s and g found by ant system

通过多次探索留下的信息素密集程度来找到瓶颈即可将问题空间划分,再使用基于 option 的分层强化学习即可解决。

除了启发式的抽象方法,有人还提出使用神经网络结构来自动进行问题的分层抽象和学习[14-20]。近些年来神经网络快速发展,尤其是在图像识别领域,更是取得了很多成果。因此有人尝试通过结合神经网络和强化学习来设计电子游戏的 AI,输入为游戏画面,通过神经网络和强化学习学习到游戏策略。有些游戏复杂度较高,需要使用分层强化学习。文献[14]中提出了 Option-Critic 架构,旨在通过神经网络强大的学习能力,模糊发现 option 和学习 option 之间的界限,直接通过神经网络一起训练。在一些游戏上取得了比不使用分层强化学习的 Deep Q Network 更好的结果。文献[15]中提出了 Manager-Worker 架构,Manager 负责给 Worker 一个子目标,而 Worker 根据子目标和当前所处的状态给出具体执行的动作。在这个方法中,Manager 和 Worker 分别是两个不同的神经网络,并且用各自的梯度分别进行优化,在实验中也取得了很好的效果。

人工进行分层和抽象,不仅费时费力,而且容易忽视问题中不易发现的内在联系。因此使用端到端的分层强化学习,从分层抽象到训练学习,都通过机器学习的方法自动进行必然是今后人们不断研究的方向。

3　总束语

本文对于分层强化学习进行了回顾。首先介绍了强化学习、马尔科夫决策过程以及半马尔科夫决策过程的定义和基本概念,规定了本文的符号使

用。然后,在第2节分4个方面,阐述了Sutton等提出的option方法,Parr和Russell提出的HAMs方法以及Dierrerich等提出的MaxQ方法,阐述了这些方法具体的计算方法。分析了近两年来的研究方向,介绍了一些端到端的、自动抽象的分层强化学习。分层强化学习是解决大规模强化学习的潜在途径,然而其受到的关注不足。希望本篇综述能够引起更多人关注分层强化学习。

参考文献:

[1] BARTO A G, MAHADEVAN S. Recent advances in hierarchical reinforcement learning [J]. Discrete event dynamic systems, 2013,13(4): 341-379.

[2] YAN Q, LIU Q, HU D. A hierarchical reinforcement learning algorithm based on heuristic reward function [C]//In Proceedings of 2nd International Conference on Advanced Computer Control. Shenyang, China, 2010, 3: 371-376.

[3] DETHLEFS N, CUAYÁHUITL H. Combining hierarchical reinforcement learning and Bayesian networks for natural language generation in situated dialogue [C]// European Workshop on Natural Language Generation. Nancy, France, 2011: 110-120.

[4] AL-EMRAN M. Hierarchical reinforcement learning: a survey [J]. International journal of computing and digital systems, 2015, 4(2): 137-143.

[5] MAHADEVAN S, MARCHALLECK N. Self-improving factory simulation using continuous-time average-reward reinforcement learning [C]. In Proceedings of the Machine Learning International Workshop. Nashville, USA, 1997: 202-210.

[6] HOWARD R A. Semi-Markov and decision processes [M]. New York: DOVER Publications, 2007.

[7] GIL P, NUNES L. Hierarchical reinforcement learning using path clustering [C]//In Proceedings of 8th Iberian Conference on Information Systems and Technologies. Lisboa, Portugal, 2013: 1-6.

[8] STULP F, SCHAAL S. Hierarchical reinforcement learning with movement primitives [C]//In Proceedings of 11th IEEE-RAS International Conference on Humanoid Robots. Bled, Slovenia, 2011: 231-238.

[9] DU X, LI Q, HAN J. Applying hierarchical reinforcement learning to computer games [C]//In Proceedings of IEEE International Conference on Automation and Logistics. Xi'an, China, 2009: 929-932.

[10] SUTTON R S, PRECUP D, SINGH S. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning [J]. Artificial intelligence, 1999, 112(1/2): 181-211.

[11] PARR R, RUSSELL S. Reinforcement learning with hierarchies of machines [C]//Advances in Neural Information Processing Systems. Colorado, USA, 1998: 1043-1049.

[12] DIETTERICH T G. Hierarchical reinforcement learning with the MAXQ value function decomposition [J]. Journal of artificial intelligence research, 2000, 13: 227-303.

[13] MOHSEN G, TAGHIZADEH N, et al. Automatic abstraction in reinforcement learning using ant system algorithm [C]//In Proceedings of AAAI Spring Symposium: Lifelong Machine Learning. Stanford, USA, 2013: 114-122.

[14] PIERRE-LUC BACON, JEAN HARB. The option-critic architecture [C]//In Proceeding of 31th AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017: 1726-1734.

[15] VEZHNEVETS A S, OSINDERO S, SCHAUL T, et al. FeUdal networks for hierarchical reinforcement learning [C]//In Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia, 2017: 3540-3549.

[16] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(2): 529-533.

[17] TEJAS D. K, KARTHNIK N, ARDAVAN S, et al. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation [C]//Annual Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 3675-3683.

[18] CARLOS FLORENSA, YAN D, PIETER A. Stochastic neural networks for hierarchical reinforcement learning [EB/OL]. Berkeley, USA, arXiv. 2017, https://arxiv.org/pdf/1704.03012.pdf.

[19] LAKSHMINARAYANAN A S, KRISHNAMURTHY R, KUMAR P, et al. Option discovery in hierarchical reinforcement learning using spatio-temporal clustering [EB/OL]. Madras, India, arXiv, 2016, https://arxiv.org/pdf/1605.05359.pdf.

[20] XUE B , GLEN B. DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning [J]. ACM transactions on graphics,2017, 36(4): 1-13.

作者简介:

周文吉,男,1995**年生,硕士研究生,主要研究方向为强化学习和数据挖掘。**

俞扬,男,1982**年生,副教授,博士生导师,主要研究方向为人工智能、机器学习、演化计算、数据挖掘。曾获**2013**年全国优秀博士学位论文奖,**2011**年中国计算机学会优秀博士学位论文奖。发表论文**40**余篇。**