DOI:10.16383/j.aas.2004.01.011

第30卷 第1期 自 动 化 学 报 Vol. 30, No. 1
2004年1月 ACTA AUTOMATICA SINICA Jan., 2004

强化学习研究综述[1)]

高 阳 陈世福 陆 鑫
(南京大学计算机软件新技术国家重点实验室 南京 210093)
(E-mail:{gaoy,chensf,lx}@nju.edu.cn)

摘 要 强化学习通过试错与环境交互获得策略的改进,其自学习和在线学习的特点使其成为机器学习研究的一个重要分支.该文首先介绍强化学习的原理和结构;其次构造一个二维分类图,分别在马尔可夫环境和非马尔可夫环境下讨论最优搜索型和经验强化型两类算法;然后结合近年来的研究综述了强化学习技术的核心问题,包括部分感知、函数估计、多agent强化学习,以及偏差技术;最后还简要介绍强化学习的应用情况和未来的发展方向.

关键词 强化学习,部分感知,函数估计,多agent强化学习
中图分类号 TP181

Research on Reinforcement Learning Technology: A Review

GAO Yang CHEN Shi-Fu LU Xin
(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093)
(E-mail:{gaoy,chensf,lx}@nju.edu.cn)

Abstract Reinforcement learning gets optimal policy through trial-and-error and interaction with dynamic environment. Its properties of self-improving and online learning make reinforcement learning become one of most important machine learning methods. In this paper, we firstly survey the foundation, structure and algorithms of reinforcement learning. We also discuss the exploration oriented algorithms and the exploitation oriented algorithms in Markov and non-Markov surroundings. Then we deeply discuss some key concepts of reinforcement learning, including partially observable environment, function approximation, multi-agent reinforcement learning and rule extraction from reinforcement learning. Finally, we briefly introduce some applications of reinforcement leaning and point out some directions of reinforcement learning.

Key words Reinforcement learning, partially observe, function approximation, multi-agent reinforcement learning

1) 国家自然科学基金(60103012,69905001)和国家"973"重点研究发展规划(2002CB312002)资助
Supported by National Natural Science Foundation of P. R. China(60103012 and 69905001) and the National Grand Fundamental Research "973" Program of China(2002CB312002)
收稿日期 2002-11-04 收修改稿日期 2003-03-14
Received November 4, 2002; in revised form March 14, 2003

1 引言

智能 agent 的一个主要特征是能够适应未知环境，其中学习能力是智能 agent 的关键属性之一. 在机器学习范畴，根据反馈的不同，学习技术可以分为监督学习(supervised learning)、非监督学习(unsupervised learning)和强化学习(reinforcement learning)三大类. 其中强化学习是一种以环境反馈作为输入的、特殊的、适应环境的机械学习方法. 从 20 世纪 80 年代末开始，随着对强化学习的数学基础研究取得突破性进展后，对强化学习的研究和应用日益开展起来，成为目前机器学习领域的研究热点之一[1,2].

本文对国内外强化学习的研究现状进行综述. 首先解释强化学习的原理、结构和主要算法；其次对强化学习的 4 个主要研究方向，如部分感知、函数估计、多 agent 强化学习和偏差，分别进行综述. 最后介绍强化学习的主要应用和未来研究方向.

2 强化学习

2.1 强化学习原理和结构

所谓强化学习是指从环境状态到动作映射的学习，以使动作从环境中获得的累积奖赏值最大. 该方法不同于监督学习技术那样通过正例、反例来告知采取何种行为，而是通过试错(trial-and-error)来发现最优行为策略. 它通常包括两个方面：一是将强化学习作为一类问题；二是指解决这类问题的一种技术. 如果将强化学习作为一类问题，目前的学习技术大致分成两类：一类是搜索 agent 的行为空间，以发现 agent 最优的行为，通常可以通过遗传算法等搜索技术实现；另一类是采用统计技术和动态规划方法来估计在某一环境状态下的动作的效用函数值. 研究人员将这种学习技术特指为强化学习技术[1]. 在本文中认为强化学习是一种学习技术. 它是从控制论、统计学、心理学等相关学科发展而来，最早可以追溯到巴普洛夫的条件反射实验. 但直到 20 世纪 80 年代末、90 年代初强化学习技术才在人工智能、机器学习和自动控制等领域中得到广泛研究和应用，并被认为是设计智能 agent 的核心技术之一[3].

标准的 agent 强化学习框架结构如图 1 所示. agent 由状态感知器 I、学习器 L 和动作选择器 P 三个模块组成. 状态感知器 I 把环境状态 s 映射成 agent 内部感知 i；动作选择器 P 根据当前策略选择动作 a 作用于环境 W；学习器 L 根据环境状态的奖赏值 r 以及内部感知 i，更新 agent 的策略知识. W 在动作 a 的作用下将导致环境状态的变迁 s'. 强化学习技术的基本原理是：如果 agent 的某个动作导致环境正的奖赏(强化信号)，那么 agent 以后产生这个动作的趋势便会加强；反之 agent 产生这个动作的趋势减弱.

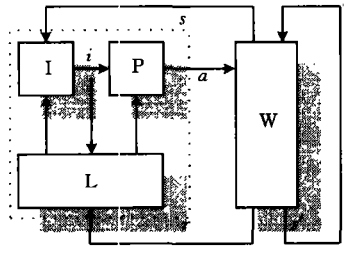

图 1　强化学习的框架结构
Fig. 1　The framework of reingorcement learning

既然强化学习的目标是学习一个行为策略 $\pi: S \rightarrow A$，使 agent 选择的动作能够获得环境最大的奖赏. 但在多数问题中，往往需要考虑 agent 行为的长期影响. 因此需要定义一个目标函数来表明从长期的观点确定什么是优的动作. 通常

以状态的值函数(value function)或状态-动作对的值函数表达此目标函数,函数形式有以下三种:

$$V^{\pi}(s_t)=\sum_{i=0}^{\infty}\gamma^i r_{t+i},\quad 0<\gamma\leqslant 1 \tag{1}$$

$$V^{\pi}(s_t)=\sum_{t=0}^{h}r_t \tag{2}$$

$$V^{\pi}(s_t)=\lim_{h\to\infty}\left(\frac{1}{h}\sum_{t=0}^{h}r_t\right) \tag{3}$$

其中 γ 为折扣因子,r_t 是 agent 从环境状态 s_t 到 s_{t+1} 转移后所接受到的奖赏值,其值可以为正、负或零. 式(1)为无限折扣模型,agent 考虑未来无限步的奖赏,并以某种形式的折扣累积在值函数中;式(2)为有限模型,agent 只考虑未来 h 步的奖赏和;式(3)为平均奖赏模型,agent 考虑其长期平均奖赏. 显然,如果能够确定目标函数,则根据下式即可以确定最优行为策略

$$\pi^{*}=\arg\max_{\pi}V^{\pi}(s),\quad \forall s\in S \tag{4}$$

只能利用不确定的环境奖赏值来发现最优行为策略是强化学习主要特征和难点. 不同于监督学习技术,强化学习不需要事先提供训练例,因此强化学习是一种在线学习技术;而在监督学习技术中实现在线学习是非常困难的[4].

强化学习也不同于规划技术. 主要区别在于规划需要构造复杂的状态图,而强化学习 agent 只需要记忆其所处的环境状态和当前策略知识;其次,规划技术总假定环境是稳定的,agent 和环境的交互作用可以通过某种搜索过程来预测. 由于规划技术并没有真正地考虑行为如何适应环境的问题,其只适用于系统完全了解并可控制的环境;相反强化学习强调系统与环境的交互作用. 因此,强化学习技术比规划技术适用面更广[4].

同样强化学习也不同于自适应控制技术. 虽然两者都有共同的奖赏函数形式,但在自适应控制中,尽管不能事先确定系统的动态模型,但系统模型必须可以从统计数据中估计,而且系统的动态模型必须是固定的. 因此,自适应控制本质上是一个参数估计问题,而且可以通过统计分析进行假设估计. 相反在强化学习技术中,并没有这些限制.

2.2 强化学习分类图

为了对目前强化学习技术进行分析、并探索和研究未来发展方向,我们建立一个二维分类图. 由于在强化学习中,agent 通过其选择的动作策略将影响训练样例的分布,这就导致一个问题:哪一种试验策略可以产生最有效的学习. 因此强化学习面临搜索(exploration)和利用(exploitation)的两难问题:是选择搜索未知的状态和动作(搜索新的知识),还是利用已获得的、可以产生高回报的状态和动作. 由于搜索新动作能够带来长期的性能改善,因此搜索可以帮助收敛到最优策略;而利用可以帮助系统短期性能改善,但可能收敛到次优解上. 因此我们把强调获得最优策略的强化学习算法称为最优搜索型(exploration oriented);而把强调获得策略性能改善的强化学习算法称为经验强化型(exploitation oriented). 在图 2 中,分类图的横轴分为最优搜索型和经验强化型两大类. 分类图的纵轴是强化学习所面临的环境类别,基本上可以分为马尔可夫型环境和非马尔可夫型环境. 图 2 给出一些代表性的算法在分类图中的表示.

通常强化学习面临两类任务:一类是非顺序型任务;另一类是顺序型任务. 在非顺序型

任务中，当 agent 学习环境状态空间到 agent 行为空间的映射时，agent 的动作会瞬时得到环境奖赏值，而不影响后继的状态和动作. 而在顺序型任务中，agent 采用的动作可能影响未来的状态和未来的奖赏报酬. 在这种情况下，agent 需要在更长的时间周期内与环境交互，估计当前动作对未来状态的影响. 因此 agent 的学习涉及到时间信度分配问题(temporary credit assignment problem)，即 agent 在采用一个动作后得到的奖赏值，如何分配到过去每个行为动作上. 当前的研究主要集中在顺序型任务，下面结合图 2 分别讨论具体的学习算法.

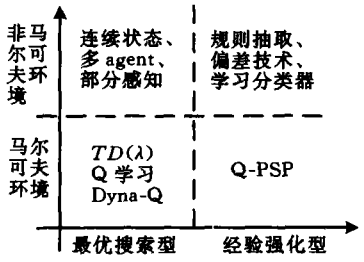

图 2　强化学习分类图

Fig. 2　The categories of reinforcement learning

3　最优搜索型强化学习算法

假设环境是马尔可夫型的，则顺序型强化学习问题可以通过马尔可夫决策过程(Markov decision process，MDP)建模. 首先给出马尔可夫决策过程的形式定义.

定义 1.　马尔可夫决策过程包含一个环境状态集 S，agent 行为集合 A，奖赏函数 R：$S\times A\to Real$ 和状态转移函数 T：$S\times A\to PD(S)$. 记 $R(s,a,s')$ 为 agent 在状态 s 采用 a 动作使环境状态转移到 s' 获得的瞬时奖赏值；记 $T(s,a,s')$ 为 agent 在状态 s 采用 a 动作使环境状态转移到 s' 的概率.

马尔可夫决策过程的本质是：当前状态向下一状态转移的概率和奖赏值只取决于当前状态和选择的动作，而与历史状态和历史动作无关. 因此在已知状态转移概率函数 T 和奖赏函数 R 的环境模型知识下，可以采用动态规划技术求解最优策略. 而强化学习着重研究在 T 函数和 R 函数未知的情况下，agent 如何获得最优行为策略. 由于顺序型强化学习问题面临的时间信度分配难题，目前通用的解决方法是采用迭代技术来调整当前状态和下一状态的值函数的估计值.

考虑式(1)目标函数，首先给出最优策略下值函数的定义，如式

$$V^*(s)=\max_a\Big(\gamma\sum_{s'\in S}T(s,a,s')(r(s,a,s')+V^*(s'))\Big),\quad \forall s\in S \tag{5}$$

而在式

$$V(s)=(1-\alpha)V(s)+\alpha(r(s,a,s')+\gamma V(s')) \tag{6}$$

中，通过 Bellman 迭代逼近最优策略下的值函数. 事实上，为满足学习速度和收敛性，各种强化学习算法都会对式(6)做相应地修改.

如果在学习过程中 agent 无需学习马尔可夫决策模型知识(即 T 函数和 R 函数)，而直接学习最优策略，将这类方法称为模型无关法(model-free)；而在学习过程中先学习模型知识，然后根据模型知识推导优化策略的方法，称为基于模型法(model-base). 由于不需要学习 T 函数和 R 函数，模型无关方法每次迭代计算量较小. 但由于没有充分利用每次学习中获取的经验知识，相比基于模型法收敛要慢得多. 常见的强化学习算法中 TD 算法和 Q-学习算法属于典型的模型无关法，而 Sarsa 和 Dyna-Q 算法属于基于模型法.

3.1 TD 算法

TD(temporal difference)学习是强化学习技术中最主要的学习技术之一. TD 学习是蒙特卡罗思想和动态规划思想的结合,即一方面 TD 算法在不需要系统模型情况下可以直接从 agent 经验中学习;另一方面 TD 算法和动态规划一样,利用估计的值函数进行迭代[5].

最简单的 TD 算法为一步 TD 算法,即 TD(0)算法,这是一种自适应的策略迭代算法. 所谓一步 TD 算法,是指 agent 获得的瞬时奖赏值仅向后回退一步,也就是只迭代修改了相邻状态的估计值. TD(0)算法的迭代公式为

$$V(s_t)=V(s_t)+\alpha(r_{t+1}+\gamma V(s_{t+1})-V(s_t)) \tag{7}$$

其中 α 为学习率,$V(s_t)$指 agent 在 t 时刻访问环境状态 s_t 时估计的状态值函数,$V(s_{t+1})$指 agent 在 $t+1$ 时刻访问环境状态 s_{t+1} 时估计的状态值函数,r_{t+1}指 agent 从状态 s_t 向状态 s_{t+1} 转移时获得的瞬时奖赏值. 学习开始时,首先初始化 V 值;然后 agent 在 s_t 状态,根据当前策略确定动作 a_t,得到经验知识和训练例$\langle s_t,a_t,s_{t+1},r_{t+1}\rangle$;其次根据此经验知识依据式(7)修改状态值函数. 当 agent 访问到目标状态,算法终止一次迭代循环. 算法继续从初始状态开始新的迭代循环,直至学习结束.

TD 算法是由 Sutton 于 1988 年提出,并证明当系统满足马尔可夫属性,α 绝对递减条件下,TD 算法必然收敛[5]. 但 TD(0)算法存在收敛慢的问题,其原因在于 TD(0)中 agent 获得的瞬时奖赏值只修改相邻状态的值函数估计值. 更有效的方法是 agent 获得的瞬时奖赏值可以向后回退任意步,称为 TD(λ)算法. TD(λ)算法的收敛速度有很大程度的提高,算法迭代公式可用下式

$$V(s)=V(s)+\alpha(r_{t+1}+\gamma V(s_{t+1})-V(s))e(s) \tag{8}$$

表示,其中 $e(s)$定义为状态 s 的选举度. 实际应用中 $e(s)$可以通过以下方法计算:

$$e(s)=\sum_{k=1}^{t}(\lambda\gamma)^{t-k}\delta_{s,s_k},\quad \delta_{s,s_k}=\begin{cases}1, & \text{if } s=s_k\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

$$e(s)=\begin{cases}\gamma\lambda e(s)+1, & \text{if } s \text{ 是当前状态}\\ \gamma\lambda e(s), & \text{otherwise}\end{cases} \tag{10}$$

式(9)中,奖赏值向后传播 t 步. 在当前状态历史的 t 步中,如果一个状态 s 被多次访问,其选举度 $e(s)$越大,表明其对当前奖赏值的贡献最大,然后其值函数通过式(8)迭代修改. 式(10)是式(9)的一种改进.

3.2 Q-学习

Q-学习是由 Watkins 提出的一种模型无关的强化学习算法[6,7],又称为离策略 TD 学习(off-policy TD). 不同于 TD 算法,Q-学习迭代时采用状态-动作对的奖赏和 $Q^*(s,a)$作为估计函数,而非 TD 算法中的状态奖赏和 $V(s)$,因此在 Agent 每一次学习迭代时都需要考察每一个行为,可确保学习过程收敛. Q-学习算法的基本形式如下

$$Q^*(s,a)=\gamma\sum_{s'\in S}T(s,a,s')(r(s,a,s')+\max_{a'}Q^*(s',a')) \tag{11}$$

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha(r_{t+1}+\gamma\max_{a}Q(s_{t+1},a)-Q(s_t,a_t)) \tag{12}$$

式(11)中 $Q^*(s,a)$表示 agent 在状态 s 下采用动作 a 所获得的最优奖赏折扣和. 由此可知,最优策略为在 s 状态下选用 Q 值最大的行为. 类似于 TD 学习,Q-学习首先初始化 Q 值;然后 agent 在 s_t 状态,根据 ε-贪心策略确定动作 a_t,得到经验知识和训练例 $\langle s_t,a_t,s_{t+1},r_{t+1}\rangle$;

其次根据此经验知识依据式(12)修改 Q 值. 当 agent 访问到目标状态，算法终止一次迭代循环. 算法继续从初始状态开始新的迭代循环，直至学习结束. 在这个过程中，Q-学习不同于 TD 算法有两点：1) Q-学习迭代的是状态-动作对的值函数；2)Q-学习中只需采用贪心策略选择动作，无须依赖模型的最优策略.

由于在一定条件下 Q-学习只需采用贪心策略即可保证收敛，因此 Q-学习是目前最有效的模型无关强化学习算法. Watkins 等人利用随机过程和不动点理论，证明当 α 满足一定条件时 MDP 模型 Q-学习过程的收敛性. 并给出更加详细的泛化证明[7, 8]. 同样，Q-学习也可根据 TD(λ)算法的方式扩充到 Q(λ)算法.

3.3 Sarsa

Sarsa 算法是 Rummery 和 Niranjan 于 1994 年提出的一种基于模型算法，最初被称为改进的 Q-学习算法[8]. 它仍然采用的是 Q 值迭代. Sarsa 是一种在策略 TD 学习(on-policy TD). 一步 Sarsa 算法可用下式表示：

$$Q(s_t,a_t)=Q(s_t,a_t)+\alpha(r_{t+1}+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)) \tag{13}$$

agent 在每个学习步，首先根据 ε-贪心策略确定动作 a_t，得到经验知识和训练例$\langle s_t,a_t,s_{t+1},r_{t+1}\rangle$；其次再根据 ε-贪心策略确定状态 s_{t+1}时的动作 a_{t+1}，并依据式(13)进行值函数修改；然后将确定的 a_{t+1}作为 agent 所采取的下一个动作. 显然，Sarsa 与 Q-学习的差别在于 Q-学习采用的是值函数的最大值进行迭代，而 Sarsa 则采用的是实际的 Q 值进行迭代. 除此之外，Sarsa 学习在每个学习步 agent 依据当前 Q 值确定下一状态时的动作；而 Q-学习中依据修改后的 Q 值确定动作. 因此称 Sarsa 是一种在策略 TD 学习.

4　经验强化型强化学习算法

在最优搜索型强化学习算法中，动作的选择总是基于当前值函数采用贪心策略. 而在经验强化型学习算法中，为充分利用已获得的经验知识，根据经验维持的动作规则进行动作的选择. 古典的强化学习都是经验强化型学习算法，如 Samuel 的西洋跳棋游戏中的动作选择[9]，以及 Holland 在分类器系统中的救火龙(bucket brigade)算法[10].

Horiuchi 等人提出一种学习方法叫 Q-PSP[11]，它将遗传算法中的利润共享方法(Profit sharing plan，PSP)结合进 Q-学习中. Q-PSP 类似于 Q(λ)算法，采用有限的状态回退. Q-PSP 基本思想是，当 agent 获得经验知识$\langle s_t,a_t,s_{t+1},r_{t+1}\rangle$后，构造规则集合 R；并在状态 s_{t+1}时，生成备选规则集合 R'，并基于 R'确定动作 a_{t+1}；当 agent 再次从环境获得奖赏时，依据一定的规则将奖赏分配在 R'上；然后进行下一状态学习.

显然，由于 Q-PSP 利用已得到的经验知识构造规则集合，因此其学习速度比单步 Q-学习更快；但是因为 Q-PSP 会导致强化无用规则，因此不能满足收敛要求. 实验表明当采用较大的状态步回退或备选规则集合时，Q-PSP 算法的性能也随之下降. 另外，由于 Q-PSP 属于经验强化型强化学习算法，对于动态环境则性能较差. 因此在经验强化学习算法中，如何设计有效的奖赏分配函数，将是研究的核心问题.

从目前研究分析，首先是进一步发展马尔可夫环境下的高效的学习算法，其次是对经验强化型强化学习算法进行理论分析.

5 部分感知

在实际的问题中，agent 往往无法完全感知环境信息. 即使环境属于马尔可夫型，但由于感知的不全面，对于状态间的差异也无法区别. 因此，部分感知问题属于非马尔可夫型环境[12]. 在部分感知问题中，如果不对强化学习算法进行任何处理就加以应用的话，学习算法将无法收敛. 目前最主要的研究方法是预测模型法. 这种方法是建立在部分可观察马尔可夫决策过程 POMDP(partially observable MDP)模型之上、基于最优搜索型强化学习算法. 下面首先给出 POMDP 模型的定义[13].

定义 2. 在马尔可夫决策模型 S,A,T 和 R 上，Ω 是 agent 可以感知的世界状态集合，观察函数 $O: S\times A\to PD(\Omega)$. agent 在采取动作 a 转移到状态 s' 时，观察函数 O 确定其在可能观察上的概率分布，记为 $O(s',a,o)$.

在 POMDP 模型中，不仅考虑动作的不确定性，同时也考虑状态的不确定性. 这种数学描述更接近现实世界，因此应用面比 MDP 模型更广. 解决 POMDP 问题的基本思路是将系统转换为 MDP 描述，即假设存在部分可观测的隐状态集 S 满足马尔可夫属性.

在预测模型方法中，将状态迁移的历史知识应用于预测模型或构建 agent 内部状态，同时引入对内部状态的置信度，将 POMDP 转化为统计上的 MDP 求解[13]. 状态的置信度又称为信用状态(belief state). 信用状态 b 定义为在隐状态集 S 上的概率分布，记 $b(s)$ 为对状态 s 的置信度. 在信用状态 b，Agent 执行动作 a，得到新的观察 o，此时根 据 Bayes 原理，新信用状态 b' 计算如下：

$$b'(s')=\Pr(s'\mid o,a,b)=\frac{O(s',a,b)\sum_{s\in S}T(s,a,s')b(s)}{\Pr(o\mid a,b)} \tag{14}$$

显然，当信用状态是可计算时，POMDP 问题最优策略学习转变为“信用状态 MDP”(belief MDP)最优策略的学习. 下面给出信用状态 MDP 模型的定义.

定义 3. B 是 agent 所有信用状态的集合，A 是动作集合，记状态转移函数为 $\tau(b,a,b')$，奖赏函数为 $\rho(b,a)$. 显然，

$$\tau(b,a,b')=\sum_{o\in O}\Pr(b'\mid a,b,o)\Pr(o\mid a,b) \tag{15}$$

$$\rho(b,a)=\sum_{s\in S}b(s)R(s,a) \tag{16}$$

对 POMDP 问题的学习，目前是强化学习中一个非常重要的研究方向. 根据以上定义，类似于 Q-学习，Kaelbling 等人给出相应的算法[13]. 但由于信用状态 MDP 模型是一个连续状态的模型，随着环境复杂程度增加($|S|>15$，$|O|>15$)，预测模型的大小呈爆炸性的增大，算法实际上不可行. 因此，如何结合第 6 节的函数估计方法有效地减少计算量、加快学习算法的收敛速度是待解决的研究课题之一.

6 函数估计

对于大规模 MDP 或连续空间 MDP 问题中，强化学习不可能遍历所有状态. 因此要求

强化学习具有泛化能力.强化学习中的映射关系包括 $S\to A$,$S\to R$,$S\times A\to R$,$S\times A\to S$ 等.函数估计本质就是用参数化的函数逼近这些映射.

用算子 Γ 来表示式(6).假设初始的值函数记为 V_0,则学习过程产生的值函数逼近序列为

$$V_0,\Gamma(V_0),\Gamma(\Gamma(V_0)),\Gamma(\Gamma(\Gamma(V_0))),\cdots.$$

在经典的强化学习算法中,值函数采用策略查找表(lookup-table)保存.在函数估计中,采用参数化的函数替代策略查找表.此时,强化学习基本结构如图 3 所示.记 V 为目标函数,$\hat{V}$ 为估计函数,则 $M:V\to\hat{V}$ 为函数估计算子.假设值函数初值为 V_0,则学习过程中产生的值函数序列为

$$V_0,M(V_0),\Gamma(M(V_0)),M(\Gamma(M(V_0))),$$
$$\Gamma(M(\Gamma(M(V_0)))),\cdots$$

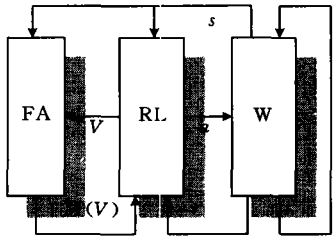

图 3　函数估计强化学习的框架结构
Fig. 3　The framework of function approximation of reinforcement learning

因此,类似于 Q-学习算法,函数估计强化学习算法迭代公式做以下修改

$$Q(s,a)=(1-\alpha)\hat{V}(s,a)+\alpha(r(s,a,s')+\max_{a'}\hat{V}(s',a')) \tag{17}$$

$$\hat{V}(s,a)=M(Q(s,a)) \tag{18}$$

在函数估计强化学习中,同时并行两个迭代过程:一是值函数迭代过程 Γ,另一是值函数逼近过程 M.因此,M 过程逼近的正确性和速度都将对强化学习产生根本的影响.目前函数估计的方法通常采用有导师监督学习方法,如状态聚类[14,15]、函数插值[16]、函数拟合、决策树[17]、人工神经网络[18]和 CMAC[19]等方法.

状态聚类将整个状态空间分成若干区域,在同一区域的状态认为其值函数相等.于是一个连续或较大规模的 MDP 问题被离散化为规模较小的 MDP 问题.状态聚类最简单的方法是区格法,它将状态空间的每一维等分为若干区间,而将整个状态空间划分为若干相同大小的区域,对二维来说就是区格划分.更复杂的划分方法是变步长划分和三角划分,采用状态聚类方法的函数估计强化学习已经被证明是收敛的[15].需要指明的是,尽管状态聚类强化学习是收敛的,但并不一定收敛到原问题的最优解上.要使收敛的值函数达到一定的精度,状态聚类的步长不能太大.因此对于大规模 MDP 问题,它仍然面临着"维数灾难"的困难.

线性插值和多线性插值是状态聚类的改进,它并不将一个区间(或区格)的值函数设为一个值,而是对顶点进行线性插值,从而可以取得更好的性能.Davies 等研究在一个二维的问题上,使用 $11\times11=121$ 的区格上的双线性插值便可以取得 $301\times301=90601$ 的区格法相当的性能[20].线性插值和多线性插值也已被证明是收敛的,但其仍然面临"维数灾难"的困难.而非线性插值则不能保证收敛性[21].

目前函数估计强化学习研究的热点是神经网络方法、线性拟合方法等.虽然这些可以大幅度提高强化学习的速度,但并不能够保证收敛性.因此,研究既能保证收敛性,又能提高收敛速度的新型函数估计方法,仍然是学者们研究的重点之一[22].

7　多 agent 强化学习

多 agent 系统是另一种形式的非马尔可夫环境.多 agent 强化学习机制被广泛应用到

各个领域，例如游戏[23, 24]、邮件路由选择[25]、口语对话系统[26]以及机器人足球[27]等等. Weiss 将多 agent 学习分成三类：乘积(multiplication)形式、分割(division)形式和交互(interaction)形式[28]. 这种分类方法基于要么将多 agent 系统作为一个可计算的学习 agent；要么是每个 agent 都有独立的强化学习机制，通过与其他 agent 适当交互加快学习过程. 每个 agent 拥有独立的学习机制，并不与其他 agent 交互的强化学习算法称之为 CIRL(Concurrent Isolated RL). CIRL 算法只能够应用在合作多 agent 系统，并只在某些环境中优于单 agent 强化学习. 而每个 agent 拥有独立的学习机制，并与其他 agent 交互的强化学习算法称之为交互强化学习(interactive RL). 交互式强化学习面临的主要问题是结构信用分配问题(structural credit assignment problem)，即系统获得的奖赏如何分配到每个agent的行为上. 交互式强化学习典型的算法是 ACE 和 AGE，但只能够应用在合作多agent系统. 另外一个主要的问题是 agent 之间为什么交互？

为了回答这个问题，并理论分析多 agent 系统中的交互作用，我们借助于对策论(game theory)数学工具对多 agent 强化学习进行进一步分析. 在对策模型中，每个 agent 获得的瞬时奖惩不仅仅取决于自身的动作，同时还依赖于其他 agent 的动作. 因此，可以将多 Agent 系统中每个离散状态 s 形式化为一个对策 g. 那么强化学习的马尔可夫决策模型扩展为多 Agent 系统的马尔可夫对策模型，其定义如下.

定义 4. 离散的状态集 S(即对策集 G)，agent 动作集 A_i 的集合 A，联合奖赏函数 $R_i: S \times A_1 \times \cdots \times A_n \rightarrow Real$ 和状态转移函数 $T: S \times A_1 \times \cdots \times A_n \rightarrow PD(S)$. 每个 Agent 目标都是最大化期望折扣奖赏和.

基于此，我们将多 agent 强化学习分成三种形式：合作型多 agent 强化学习、竞争型多 Agent 强化学习和半竞争型多 agent 强化学习. 下面分别分析各自的特点和主要算法.

7.1 合作型多 agent 强化学习

在合作型多 agent 强化学习中，由于在任意离散状态，马尔可夫对策的联合奖赏函数 R_i 对每个 agent 来说是一致的、相等的. 因此，每个 agent 最大化自身期望折扣奖赏和的目标与整个多 agent 系统的目标是一致的. 事实上，前述的并发独立强化学习和交互强化学习都属于合作型多 agent 强化学习. 在合作多 agent 系统中，合作进化学习(cooperative coevolution learning)可以达到问题的最优解[29].

7.2 竞争型多 agent 强化学习

在竞争型多 agent 强化学习中，任意离散状态下马尔可夫对策的联合奖赏函数 R_i 对每个 agent 来说是互为相反的. 因此每个 agent 自身目标与其他 agent 的目标是完全相反的. 为叙述方便，我们以两个 agent 为例，即系统中包含 agent A 和对手 agent B. 图 4 给出两个agent系统中某一状态下的对策模型. 显然，该模型满足零和对策的定义：在任何策略下所有 agent 的奖赏和为 0.

		agent B	
		b_1	b_2
agent A	a_1	(1, −1)	(4, −4)
	a_2	(2, −2)	(3, −3)

图 4 两个 agent 零和对策模型
Fig. 4 The zero-sum game model in two agent system

由于 agent A 的奖赏值取决于 agent B 的动作，因此传统单 agent 强化学习算法在竞争型多 agent 强化学习中不适用. 解决这一问题最简单的方法是采用极小极大 Q 算法：在每个状态 s，对于 agent A 其最优策略为 agent B 选择最坏动作情况下，agent A 选择奖赏最大的动作. 因此，定义竞争型多 agent 强化学习的值函数为

$$V(s)=\max_{a\in A}\min_{b\in B}Q(s,a,b) \tag{19}$$

显然，如果将马尔可夫对策中每个状态都形式化为如图 4 的零和对策模型，那么极小极大 Q 算法可以发现最优的策略. 然而在竞争多 agent 系统中，如果允许多个 agent 同时进化，将导致系统非常复杂. Sandholm 等通过对追杀问题的实验表明，竞争进化学习(Competitive Coevolution Learning)不能够得到稳定解. 但在竞争环境中，如果自身 agent 不采用进化学习，而对手 agent 采用进化学习，则任何稳定策略都会被击败. 因此，在竞争多agent 系统中，需要对是否存在进化稳定策略或者 agent 何时采用进化学习等问题做出明确解释[30].

7.3　半竞争型多 agent 强化学习

在许多实际多 agent 系统中，往往单个 agent 的所得奖赏并不是其他 agent 所得奖赏和的负值，所以多 agent 系统中离散状态 s 只能形式化为非零和对策. 一个典型事例是图 5 表示的囚犯两难问题. 如果采用极小极大算法求解，其最优解为(a_1,b_1)，奖赏为$(-9,-9)$；而显然囚犯两难问题的最优解为(a_2,b_2)，奖赏为$(-1,-1)$. 因此在非零和 Markov 对策模型中，用极小极大 Q 算法得不到最优解.

		agent B	
		b_1	b_2
agent A	a_1	$(-9,-9)$	$(0,-10)$
	a_2	$(-10,0)$	$(-1,-1)$

图 5　两个 agent 非零和对策模型
Fig. 5　The non-zero-sum game model in two agent system

本质上非零和对策模型更能反应多 agent 系统中个体理性(individual rationality)与集体理性(group rationality)冲突的本质，高阳等人从 Littman 的思想[31]出发，采用元对策(metagame)理论，提出解决多 agent 非零和 Markov 对策的强化学习模型和算法[32]. 其基本思想是，在已知非零和对策模型下，考虑 agent 自身的愿望和预测对手的策略来修正自己的策略，能有效地解决“组合谬误”问题，从而求得系统的最优策略.

除了以上介绍的一些典型方法外，某些研究者将对策模型扩展为随机对策模型，并将 agent 的确定性策略转为概率意义上的混合策略，也能够获得比较满意的优化策略.

在半竞争多 agent 系统中，结构信用分配问题同样也非常突出. 当 agent 性能改善时，它并不知道改善是由于自身行为引起，还是由于其他 agent 行为造成的. 因此如何设计信用分配函数是竞争和半竞争多 agent 系统的难点. 另外，由于多 agent 学习离不开 agent 之间的通信，特别是在实时系统中，这种通讯代价必须考虑. 因此多 agent 强化学习的通讯也是一个不容忽视的重要问题[33].

8　符号学习和强化学习偏差

8.1　Dyna-Q

正如第 2 节所讨论的一样，当系统模型已知时，强化学习转变为规划问题. 因此，当 Agent每次试错所获得的经验知识$\langle s_t,a_t,s_{t+1},r_{t+1}\rangle$，既可以直接被用来进行优化策略的学习，也可以被用来进行模型的估计，然后从估计的模型中规划动作.

Dyna-Q 学习算法是一个典型的基于模型的算法[34]. 它与 Sarsa 的不同在于：Sarsa 学习中模型是隐含在当前 Q 函数中；而 Dyna-Q 学习算法明确地学习系统的模型. 其主要目的在于充分利用每次学习经验中获取的知识，从而解决 TD 算法和 Q-学习迭代速度较慢的问

题. Dyna-Q 学习算法的框架结构如图 6 所示.

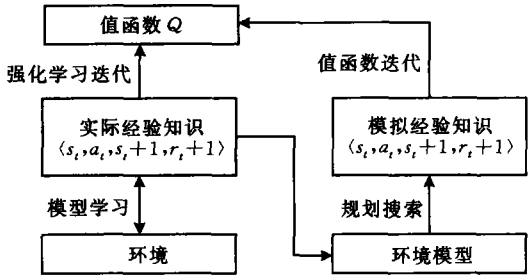

图 6 Dyna-Q 学习算法框架结构图

Fig. 6 The framework of Dyna-Q algorithm

在 Dyna-Q 学习算法中，agent 通过三步学习优化策略. 首先 agent 使用学习经验来建立环境模型，其次是用经验调节策略，最后使用模型来调节策略. 具体算法步骤如下：

1)根据当前 Q 值，在状态 s_t 选择动作 a_t；

2)agent 观察到转移状态 s_{t+1} 和奖赏值 r_t+1，得到经验知识 $\langle s_t, a_t, s_{t+1}, r_{t+1}\rangle$；

3)根据式(12)进行值函数迭代；

4)根据学习经验集采用概率统计技术建立模型，即 T 函数和 R 函数的估计；

5)利用新模型，随机模拟确定 k 个经验知识 $\langle s_t, a_t, s_{t+1}, r_{t+1}\rangle$，并进行值函数迭代；

6)agent 根据当前 Q 值，在状态 s_{t+1} 选择一个优化动作，转回 2).

8.2 强化学习中的规则抽取

单纯依靠强化学习技术构造学习 agent 存在的主要问题包括：强化学习中的值函数、最优策略的表达方式让人难以理解；当系统环境发生变化时，无法充分地利用 agent 已学习的经验；强化学习需要不断的迭代，计算复杂性和收敛速度往往都不能满足要求.

解决这些问题的一个方法是采用规划规则抽取技术. 其核心思想是：将 agent 通过强化学习技术所得到的策略，通过抽取规则，转化成其他学习技术所能够处理的表示形式，从而使 agent 可以利用其他技术进行更深层次的学习和推理；同时当环境发生改变时，可以将上次抽取出的规则用于强化学习中，以提高新一次学习的收敛速度.

Sun 等人提出了 Beam Search 算法[35]，从强化学习 agent 学习的值函数中抽取无条件规划和条件规划. 研究表明，这种技术能够得到概率规划规则. 但 Beam Search 算法存在一些重要缺陷，表现在：首先 Beam Search 算法仍然依赖强化学习过程中估计的概率迁移函数；其次 Beam Search 算法未考虑具体规划步数对规划的影响；另外算法没有考虑强化学习技术在线的奖赏函数和折扣奖赏目标函数. 为解决这些问题，高阳等人在 Sun 等人研究的基础上，进一步研究强化学习值函数和学习算法，提出一个不依赖估计概率函数的、确定规划步的、基于强化学习的规划规则抽取算法[36].

8.3 强化学习偏差

传统的强化学习研究中 agent 没有任何先验的启发知识. 但在实际应用中，无启发知识的强化学习算法收敛非常慢. 在现实中总存在各种形式的启发知识，因此研究人员研究强化学习的各种偏差(bias)技术，以提高强化学习的收敛速度. 常用的偏差技术包括整形(shaping)、局部强化(local reinforcement)、模仿(imitation)、任务分解(task decomposition)等

等[37]. 从目前国际研究看来，强化学习偏差研究内容主要包括两方面：一是先验知识以何种形式影响 agent 的强化学习过程；二是 agent 如何得到这个启发知识. 目前研究着重于第一方面，而对于启发知识通常总是由设计人员给出.

强化学习整形技术主要有两种方法：一是构造导师 agent；二是将先验知识直接综合到强化学习算法中. 早期的研究主要集中在第一种方法. 图 7 是具有导师 agent 的强化学习结构图.

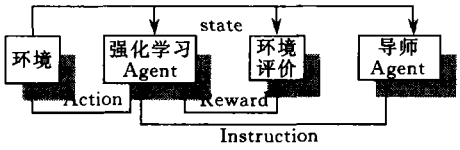

图 7　具有导师 agent 的强化学习整形结构

Fig. 7　The shapping architecture of reinforcement learning with advisor agent

图 7 中导师 agent 同样接受环境状态的输入，然后根据此输入提供一个行为指导给学习 agent. 学习 agent 同时接受环境的奖惩信号和导师 agent 的行为指导，并根据这些信息学习 agent 的行为策略. 在 Lin 等的工作中，学习 agent 在问题求解过程中完全接受导师 agent 的指导行为，并以此更新自己的行为策略[38, 39]. 在 Clouse 等研究的系统中，学习 agent 只是偶尔接受导师 agent 的指导行为[40].

由于构造导师 agent 将增加软件系统的系统复杂度，因而在目前强化学习整形研究中通常采用第二种方法. Maclin 等采用 if-then 规则，通过基于知识的神经网络技术将导师的指导行为直接编译到学习 agent 的策略中[41]；相似地，Gordon 等也采用 if-then 规则，然后转化成可操作性的规则插入到遗传算法的种群中[42]. 尽管研究者采用了不同的方法将先验知识综合到强化学习系统中，但每一种方法都能很好地显示收敛性.

9　强化学习应用

由于强化学习在大空间、复杂非线性系统中具有良好的学习性能，使其在实际中获得越来越广泛的应用. 强化学习的应用主要可以分为四类：制造过程控制、各种任务调度、机器人设计和游戏.

Moore 等研究如何将强化学习应用到实际制造过程控制中[43]. 一个具体的实验例是包装行业中生产线上如何确保包装容器符合特定的规格. Moore 等描述了一种综合式动态规划方法进行生产线控制. 蒋国飞等人在倒立摆控制中应用 Q-学习算法[44]，研究表明强化学习方法性能超过了工人手工操作和传统的控制器.

同样，强化学习也被应用到各种各样调度任务中，典型的应用包括电梯调度、车间作业调度[45]、交通信号控制[46]以及网络路由选择. Robert Crites 等研究了在高层建筑中利用强化学习的多个电梯的调度算法[17]. 这个算法综合了强化学习和前馈神经网络. 实验结果表明，这个学习算法比现有 8 种电梯调度算法性能更优. Thomas Dittarich 等在车间调度应用 TD(λ)算法，一系列应用表明强化学习可以成功地解决组合优化问题.

强化学习在机器人中的应用最为广泛. 除了可以应用强化学习技术控制机器人的手臂外，还可以用来学习多个机器人的协商行为. 典型的应用如 Christopher 提出的控制机器人

手臂运动的学习算法和 Peter Stone 等研究的机器人足球等学习算法[47].

强化学习还被广泛应用在一些游戏中[48],其中典型的应用是 Samuel 的西洋跳棋系统[9],通过设计目标函数和奖赏函数,经过上百万次的自我学习,计算机系统能够击败人类棋手.另外,强化学习在学习分类器中的应用也逐渐成为研究的热点[10].学习分类器一方面由遗传算法产生分类规则新的种群;另一方面由强化学习强化有用的分类规则,从而可以在递增的训练例中在线、增量学习分类规则.

10 结束语

强化学习是一种无导师的在线学习技术.在马尔可夫环境中,最优搜索型强化学习算法已经被证明收敛性.但对非马尔可夫环境可以进一步分为部分感知强化学习、函数估计、多 agent 强化学习以及强化学习偏差技术研究.目前对非马尔可夫环境下的强化学习研究正成为研究的热点.

References

1 Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 1996, **4**(2): 237～285

2 Li Ning, GaoYang, Lu Xin, Chen Shi-Fu. A learning agent based on reinforcement learning. *Computer Research and Development*, 2001, **38**(9):1051～1056 (in Chinese)

3 Singh S. Agents and reinforcement learning. San Mateo, CA, USA: Miller Freeman Publish Inc, 1997

4 Kaelbling L P. A situated automata approach to the design of embedded agents. *SIGART Bulletin*, 1991, **2**(4):85～88

5 Sutton R S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988, **3**: 9～44

6 Watkins P. Dayan. Q-learning. *Machine Learning*, 1992, **8**(3):279～292

7 Tsitsiklis, John N. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 1994, **16**(3):185～202

8 Rummery G, Niranjan M. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994

9 Samuel A L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 1959, **3**: 211～229

10 Holland J H. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In: Michalski R S, Carbonell J G, Mitchell T M, Machine learning: An artificial intelligence approach, Volumn 2, San Francisco: Morgan Kaufmann, 1986. 593～623

11 Horiuchi T, Katai O. Q-PSP learning: An exploitation-oriented Q-learning algorithm and its applications. *Transactions of the Society of Instrument and Control Engineers*, 1999, **39**(5):645～653

12 Lovejoy W S. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 1991, **28**:47～65

13 Leslie Pack Kaelbling, Michael L Littman, Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998, **101**: 99～134

14 Singh S, Jaakkola T, Jordan M I. Reinforcement learning with soft state aggregation. In: Tesauro G, Touretzky D, Advances in Neural Information Processing Systems, 7. Morgan Kaufmann: MIT Press, 1995. 361～368

15 Moore A W. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state spaces. In: Jack D Cowan, Gerald Tesauro, Joshua Alspector, Advances in Neural Information Processing Systems, 6. Morgan Kaufmann Publishers, 1994. 711～718

16 McCallum A K. Reinforcement learning with selective perception and hidden State [Ph. D. dissertation]. Department CS, University Rochester, 1996

17 Crites R H, Barto A G. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 1998, **33**(2): 235～262

18 Sutton R S. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In: Touretzky D, Mozer M, Hasselmo M, Advances in Neural Information Processing Systems, 8. NY: MIT Press, 1996. 1038～1044

19 Dayan P, Hinton G E. Feudal reinforcement learning. In: Hanson S J, Cowan J D, Giles C L, Advances in Neural Information Processing Systems 5. San Mateo, CA: Morgan Kaufmann, 1993. 1～8

20 Davies S. Multidimensional triangulation and interpolation for reinforcement learning. In: Michael C Mozer, Michael I Jordan, Thomas Petsche, Advances in Neural Information Processing Systems 9. NY: The MIT Press, 1997. 1005～1010

21 Gordon G J. Stable function approximation in dynamic programming. In: Armand Prieditis, Stuart Russell, Proceedings of the twelfth international conference on machine learning. San Francisco, CA: Morgan Kaufmann, 1995. 261～268

22 Sutton R S. Open theoretical questions in reinforcement learning. In: Fischer P, Simon H U, Computational Learning Theory. London: Springer, 1999. 11～17

23 Tan M. Multi-agent reinforcement learning: independent vs. cooperative agents. In: Michael N Huns, Munindar P Singh, Proceedings of Tenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1993. 487～494

24 Pan Gu. A framework for distributed reinforcement learning. In: Weiss G, Sandip Sen, Adaption and learning in multi-agent system. Lecture Notes in Computer Sciences, 1042. NY: Springer, 1996. 97～112

25 Littman M L. Markov games as a framework for multi-agent reinforcement learning. In: Cohen W W, Hirsh H, Proceedings of Eleventh International Conference on Machine Learning. New Brunswick, NJ: Morgan Kaufmann, 1994. 157～163

26 Roy N, Pineau J, Thrun S. Spoken dialogue management using probabilistic reasoning. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong: 2000

27 Peter Stone, Manuela Veloso. Team-partitioned, opaque-transition reinforcement learning. In: Oren Etizioni, Jorg P Muller, Jeffrey M Bradshaw, Proceedings of the 3rd International Conference on Autonomous Agents. Seattle: ACM Press, 1999. 206～212

28 Weiss G, Dillenbourg P. What is multi in multiagent learning? In: Dillenbourg P, Collaborative learning, cognitive and computational approaches. Amsterdam: Pergamon Press, 1998. 64～80

29 Narendra P, Sandip S, Maria Gordin. Shared memory based cooperative coevolution. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Alaska, USA: IEEE Press, 1998. 570～574

30 Tuomas W Sandholm, Robert H Crites. On multiagent Q-learning in a semi-competitive domain. In: Weiss G, Sen S, Adaption and learning in Multi-agent System. Lecture Notes in Artificial Intelligence, 1042. NY: Springer, 1996. 191～205

31 Littman M. Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on Machine Learning. New Brunswick: Morgan Kaufmann, 1994. 157～163

32 Gao Yang, Zhou Zhi-Hua, He Jia-Zhou, Chen Shi-Fu. Research on Markov game-based multiagent reinforcement learning model and algorithms. *Computer Research and Development*, 2000, **37**(3):257～263(in Chinese)

33 Takuya Ohko, Kazuo Hiraki, Yuichiro Anzai. Learning to reduce communication cost on task negotiation among multiple autonomous mobile robots. In: Weiss G, Sen S, Adaption and learning in multi-agent system. Lecture Notes in Artificial Intelligence, 1042. NY: Springer, 1996. 177～190

34 Sutton R S, Barto A G, Williams R. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 1991, **12**(2): 19～22

35 Sun R, Sessions C. Extracting plans from reinforcement learners. In: Xu L, Chan L, King I, Fu A, Proceedings of the 1998 International Symposium on Intelligent Data Engineering and Learning. New York: Springer-Verlag, 1998. 243～248

36 Gao Yang, Lu xin, Li Ning, Chen Shi-Fu. An adaptive rule extracting algorithm in probabilistic plan. *Journal of Nanjing University*, 2003, **39**(2):1～8 (in Chinese)

37 Tham C K, Prager R W. A modular Q-learning architecture for manipulator task decomposition. In: Cohen W, Hir-

sh H, Proceedings of the 11th International Conference on Machine Learning. New Brunswick, New Jersey: Morgan Kaufmann, 1994. 309～317

38 Lin Long-Ji. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 1992, **8**: 293～321

39 Lin Long-Ji. Scaling up reinforcement learning for robot control. In: Proceedings of the 10th International Conference on Machine Learning. Amherst, MA: Morgan Kaufmann, 1993. 182～189

40 Clouse J A. Learning from an automated training agent. In: Gerhard Weiss, Sandip Sen, Adaptation and Learning in Multiagent Systems. Berlin: Springer Verlag, 1996. 45～82

41 Maclin R, Shavlik J W. Incorporating advice into agents that learn from reinforcements. In: Proceedings of the 12th National Conference on Artificial Intelligence. Seattle, WA: MIT Press, 1994. 694～699

42 Gordon D, Subramanian. A multistrategy learning scheme for agent knowledge acquisition. *Informatica*, 1993, **17** (4): 331～346

43 Moore A W. Variable resolution dynamic programming: Efficiently learning action maps in multivariate real valued spaces. In: Proceedings of the 8th International Machine Learning: Workshop, San Maeto, CA: Morgan Kaufmann, 1991. 333～337

44 Jiang Guo-Fei, Wu Cang-Pu. Learning to control an inverted pendulum using Q-learning and neural networks. *Acta Automatica Sinca*, 1998, **24**(5):666～669(in Chinese)

45 Jiang Guo-Fei, Wu Cang-Pu. Inventory control using Q-learning and neural networks. *Acta Automatica Sinca*, 1999, **25**(2):236～241(in Chinese)

46 Yang Yu-Pu, Ou Hai-Tao. Self-organized control of traffic signals based on reinforcement learning and genetic algorithm. *Acta Automatica Sinca*, 2002, **28**(4):564～568(in Chinese)

47 Peter Stone. Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. Cambridge, MA: MIT Press, 2000

48 Thrun S. Learning to play the game of chess. In: Tesauro G, Touretzky D, Leen T, Advances in Neural Information Proceesing System, 7. San Fransisco: Morgan Kaufmann, 1995. 1069～1076

高 阳 南京大学计算机系副教授，1993年获大连理工大学学士学位，1996年获南京理工大学硕士学位，2000年获南京大学博士学位. 目前主要研究领域为分布式人工智能，机器学习.

(**GAO Yang** Associate professor in the Computer Science Department at Nanjing University. Recieved his bachelor degree from Dalian University of Technology in 1993, master degree from the Nanjing University of Science and Technology in 1996, and Ph. D. degree from the Nanjing University in 2000. His research interests include distributed artificial intelligence and machine learning.)

陈世福 教授，博士生导师，主要研究领域为人工智能.

(**CHEN Shi-Fu** Professor in the Computer Science Department at Nanjing University. His research interests include artificial intelligence.)

陆 鑫 硕士研究生，主要研究领域为强化学习.

(**LU Xin** Master student at Nanjing University. His research interests include reinforcement learning technology.)