强化学习及其在电脑围棋中的应用

陈兴国 [1, 2] 俞 扬 [2]

摘 要 强化学习是一类特殊的机器学习, 通过与所在环境的自主交互来学习决策策略, 使得策略收到的长期累积奖赏最大. 最近, 在围棋和电子游戏等领域, 强化学习被成功用于取得人类水平的操作能力, 受到了广泛关注. 本文将对强化学习进行简要介绍, 重点介绍基于函数近似的强化学习方法, 以及在围棋等领域中的应用.

关键词 强化学习, 函数近似, 核方法, 神经网络, 加性模型, 深度强化学习

引用格式 陈兴国, 俞扬. 强化学习及其在电脑围棋中的应用. 自动化学报, 2016, **42**(5): 685−695

DOI 10.16383/j.aas.2016.y000003

Reinforcement Learning and Its Application to the Game of Go

CHEN Xing-Guo[1, 2] YU Yang[2]

Abstract Reinforcement learning is a particular type of machine learning that autonomously learns from interactions with the environment, so that its long-term reward is maximized. It has recently been successfully applied to playing the game of Go and video games, and human expert level is demonstrated. Since these results are receiving increasing attentions, this paper briefly introduces reinforcement learning, focusing on the methods with function approximation, and its applications in the game of Go.

Key words Reinforcement learning, linear function approximation, kernel methods, neural networks, additive model, deep reinforcement learning

Citation Chen Xing-Guo, Yu Yang. Reinforcement learning and its application to the game of Go. *Acta Automatica Sinica*, 2016, **42**(5): 685−695

强化学习 (Reinforcement learning, RL) 是机器学习的子领域[1]. 在强化学习中, 机器 (常被称为智能体/Agent) 被放置在一个环境中, 如图 1 所示, 需要通过与环境的交互, 即观察环境状态、在环境中执行动作、并接收环境的奖赏反馈, 从而自主地了解环境并完成任务. 通常强化学习的结果是得到一个策略, 对于任意的状态, 该策略可给出相应的动作, 学习的目标是获得最优的策略, 使得机器收到的累积奖赏最大[2−3]. 强化学习与其他机器学习任务 (例如监督学习) 的显著区别在于, 首先没有预先给出训练数据, 而是要通过与环境的交互来产生, 其次在环境中执行一个动作后, 没有关于这个动作好坏的标记, 而只有在交互一段时间后, 才能得知累积奖赏, 从而推断之前动作的好坏. 例如, 在下棋时, 机器没有被告知每一步落棋的决策是好是坏, 直到许多次决策分出胜负后, 才收到了总体的反馈, 并从最终的胜负来学习, 以提升自己的胜率. 可见强化学习需要探索环境、并从滞后的反馈中学习, 也正因此, 强化学习被用于许多自主学习问题中, 例如自动驾驶、机器人操控、推荐系统等等.

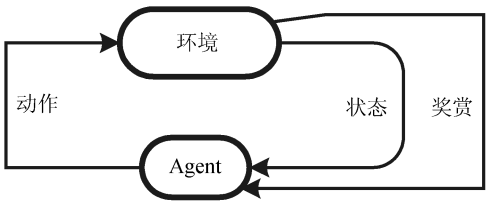

图 1 强化学习

Fig. 1 Illustration of reinforcement learning

经典强化学习方法针对离散状态和动作空间, 然而在许多应用中状态和动作空间往往是连续的, 因此基于函数近似的强化学习近年来得到了关注. 多种表示方法的发展使得强化学习能够有效处理连续状态空间的学习任务, 尤其是基于深度神经网络的函数近似强化学习方法在围棋和电子游戏领域中取得了显著的进步. 本文对强化学习的基本内容进

收稿日期 2016-04-28 录用日期 2016-05-10
Manuscript received April 28, 2016; accepted May 10, 2016
国家自然科学基金 (61403208, 61375061), 南京邮电大学引进人才科研启动基金 (NY214014) 资助
Supported by National Natural Science Foundation of China (61403208, 61375061), and Science Foundation of Nanjing University of Posts and Telecommunications (NY214014)
本文责任编委 周志华
Recommended by Associate Editor ZHOU Zhi-Hua
1. 南京邮电大学计算机学院/软件学院 南京 210046 2. 南京大学计算机软件新技术国家重点实验室 南京 210023
1. Nanjing University of Posts and Telecommunications, School of Computer Science & Technology, School of Software, Nanjing 210046 2. National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023

行介绍: 第 1 节介绍经典强化学习方法, 第 2 节介绍函数近似的强化学习, 第 3 节介绍强化学习在游戏和围棋中的应用, 第 4 节总结本文.

1 经典强化学习

强化学习的经典研究基于马尔科夫决策过程 (Markov decision process, MDP), 算法的主要思想是在 MDP 上进行动态规划, 寻找最大化累积奖赏的策略.

1.1 马尔科夫决策过程

通常假设强化学习任务满足马尔科夫性, 形式化为 MDP: 每个 MDP 由一个四元组 $\langle S, A, T, R\rangle$ 组成, 其中 S 表示状态空间, A 表示动作空间, $T: S \times A \times S \to [0,1]$ 是状态转移函数, 表示在一个状态下执行一个动作后转到另一个状态的概率, $R: S \times A \to \mathbf{R}$ 是奖赏函数, 表示发生状态转移时环境给出的立即奖赏. 通常, 在强化学习任务中, T 和 R 是未知的, 需要机器去探索, 有的问题上 S 也是未知的, 只有访问到的状态才能知道这个状态的存在.

机器从状态 s 出发, 采取动作 $a \in A(s)$, 收到环境会反馈的奖赏 $R(s,a)$, 并且以 $T(s,a,s')$ 的概率转移到一个新状态 $s' \in S$, 其中 $A(s)$ 表示在状态 s 可采取动作的集合. 此过程可无限进行下去, 也可以到终止状态处结束.

策略 π 为状态到动作的映射: $S \times A \to [0,1]$. 强化学习的目标是找到一个最优策略 π^* 以最大化累计奖赏 R_π:

$$\pi^* = \arg\max_\pi R_\pi$$

累积奖赏可以有多种计算方式, 对于机械控制等任务, 常用的是基于折扣系数的累积奖赏: $R_\pi = \mathrm{E}_\pi[\sum_{t=0}^{\infty}\gamma^t r_t]$, 其中, $\gamma \in [0,1]$ 是折扣因子, r_t 是在时间步 t 的奖赏, $\mathrm{E}_\pi[\cdot]$ 是策略 π 下的期望; 而在下棋游戏等任务中, 常用 T 步累积奖赏: $R_\pi = \mathrm{E}_\pi[\sum_{t=0}^{T} r_t]$.

1.2 值函数

对于一个策略, 如果我们可以在一个状态上就看到这个策略未来将会取得的累积奖赏, 这将为强化学习带来很大的方便, 提供这种功能的函数在强化学习中成为值函数 (Value function). 值函数可以分为两类: 状态值函数 $V(s)$ 和状态 – 动作对值函数 $Q(s,a)$. 其中, 给定一个稳定策略 π, 以折扣累积奖赏为例, 状态值函数定义如下:

$$V^\pi(s) = \mathrm{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s\right]$$

状态 – 动作对值函数定义如下:

$$Q^\pi(s,a) = \mathrm{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s, a_0 = a\right]$$

基于 T 步累积奖赏的值函数也相似定义. 从以上定义, 我们可以看到它们分别是某个状态、某个状态 – 动作对下的累计奖赏的期望值. 因此, 只要获取最优值函数就可以最大化累计奖赏, 这与强化学习的目标是一致的.

基于最优策略 π^*, 可以定义最优状态值函数:

$$V^*(s) = \mathrm{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s\right]$$

以及最优状态 – 动作对值函数:

$$Q^*(s,a) = \mathrm{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma^t r_t | s_0 = s, a_0 = a\right]$$

如果我们在时间上将值函数展开一步, 根据 Bellman 最优等式, 可有:

$$V^*(s) = \max_a \mathrm{E}[r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] = \max_a \sum\nolimits_{s'} T(s,a,s')[R(s,a) + \gamma V^*(s')] \tag{1}$$

以及

$$Q^*(s,a) = \mathrm{E}[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a] = \sum\nolimits_{s'} T(s,a,s')[R(s,a) + \gamma \max_a Q^*(s', a)] \tag{2}$$

由上述两式可知两种值函数存在以下关系:

$$V^*(s) = \max_a Q^*(s,a) \tag{3}$$

1.3 策略求解

在经典强化学习中, 求解策略是通过求解值函数来完成, 而求解值函数, 则是使用动态规划的思想, 根据式 (1) 和式 (2) 的展开形式来求解. 求解最优策略的过程可以统一在广义策略迭代 (Generalized policy iteration, GPI) 下.

如图 2 所示, 广义策略迭代包含两个交互过程: 策略评估 (Policy evaluation) 和策略改进 (Policy improvement). 其中, 策略评估指的是根据当前的策略评估值函数, 而策略改进指的是对当前值函数取最优以获得新的策略. 该过程亦可描述如下:

$$\pi_0 \xrightarrow{\mathrm{E}} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{\mathrm{E}} V^{\pi_1} \xrightarrow{I} \cdots \xrightarrow{I} \pi^* \xrightarrow{\mathrm{E}} V^*$$

其中, V^{π_i} 表示第 i 次交互中策略 π_i 对应的值函数.

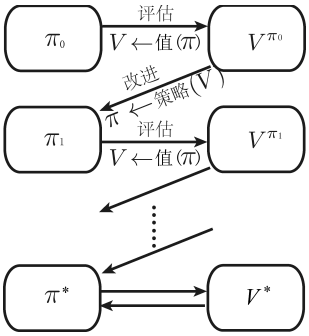

图 2　广义策略迭代: 值函数与策略交互直到最优

Fig. 2　General value iteration: iterative update between the value function and the policy until convergence

1.4　基于表格值的经典算法

由于经典强化学习中状态空间和动作都是离散有限的, 可以使用表格来记录值函数, 即表格值: 值函数为每个状态或状态–动作对分配一个存储空间, 记录其对应的函数值. 基于表格值的经典强化学习算法包括时序差分 (Temporal difference, TD) 学习算法、Sarsa 学习算法以及 Q 学习 (Q-learning) 算法等. 其中, TD 算法为预测型算法, 用于策略评估, 即其学习对象为 V 函数; Sarsa 学习和 Q 学习算法为控制型算法, 用于求解最优策略, 即学习对象为 Q 函数. 这三种经典算法的值函数更新公式可以统一为

$$f(\omega_t) \leftarrow f(\omega_t) + \alpha\delta \tag{4}$$

其中

$$\delta \leftarrow (r_t + \gamma f(\omega'_t)) - f(\omega_t) \tag{5}$$

$\alpha \in (0,1)$ 为学习率, $f : \Omega \rightarrow \mathbf{R}$ 为值函数, $\omega \in \Omega$ 为更新点, $\omega' \in \Omega$ 为后继. 若值函数 f 为状态值函数 V, 则 $\Omega = S$; 若值函数 f 为状态动作对值函数 Q, 则 $\Omega = S \times A$. 给 f, ω, ω' 赋上具体内容时, 如表 1 所列, 以上更新公式就对应了具体的算法. 以 Q 学习为例, 当 f 为 Q 函数, ω 为状态–动作, ω' 为贪心策略时, 就得到了如下所示的 Q 学习方法.

表 1　经典算法中值函数更新公式的区别与联系

Table 1　Updating formulas in classical reinforcement learning

算法	f	ω_t	ω'
TD	V	s_t	s_{t+1}
Sarsa	Q	(s_t, a_t)	(s_{t+1}, a_{t+1})
Q 学习	Q	(s_t, a_t)	$(s_{t+1}, \arg\max_{a' \in A} Q(s_{t+1}, a'))$

Q 学习算法伪码.

1. 初始化 $Q(s,a)$ 为任意值
2. **Repeat** (每个 episode)
3.　$s \leftarrow$ episode 的初始状态
4.　**Repeat** (episode 的每步):
5.　　$a \leftarrow$ 根据状态 s 及策略 π 选择的动作
6.　　采取动作 a, 观察奖赏 r 和后继状态 s'
7.　　$\delta \leftarrow r + \gamma \max_{a'} Q(s', a') - Q(s, a)$
8.　　$Q(s,a) \leftarrow Q(s,a) + \alpha\delta$
9.　　$s \leftarrow s'$
10.　**Until** s 是结束状态
10. **Until** episode 用完

经过理论分析, 上述算法的收敛性得到了严格证明, 如 TD 学习[4−5]、Sarsa 学习[6]、Q 学习[7−9]. 并且, 这些算法在小规模离散空间的强化学习任务中有着非常出色的表现.

2　函数近似强化学习

虽然基于表格的经典算法在小规模离散空间的强化学习任务上表现不错, 但更多的实际问题中状态数量很多, 甚至是连续状态空间, 这时经典强化学习方法难以有效学习. 特别对于连续状态空间离散化的方法, 当空间维度增加时, 离散化得到的状态数量指数增加, 可见基于表格值的强化学习算法不适用于大规模离散状态或者连续状态的强化学习任务[2], Bellman 用 “维度灾难” 来描述这一困难.

研究人员从不同的角度试图克服维度灾难. 例如, 分层强化学习 (Hierarchical reinforcement learning) 采用了 “分而治之” 的思想, 把一个强化学习问题分解成一组具有层次的子强化学习问题, 降低了强化学习问题的复杂度[10−20]; 迁移强化学习 (Transfer learning in reinforcement learning) 侧重于如何利用一个已学习过的强化学习问题的经验提高另一个相似但不同的强化学习问题的学习性能[17, 21−30]; 函数近似 (Function approximation) 则将策略或者值函数用一个函数显示描述[4, 31−39] 等. 这三类方法分别从不同角度求解维度灾难问题, 其中函数近似方法是最直接的求解方法, 受到了很多关注.

根据函数近似的对象不同划分, 可以分为策略搜索 (Policy search) 和值函数近似 (Value function approximation).

1) 策略搜索指直接在策略空间进行搜索, 又分为基于梯度 (Gradient-based) 的方法和免梯度 (Gradient-free) 方法.

基于策略梯度的方法: 从一个随机策略开始, 通过策略梯度上升的优化方法不断地改进策略, 例如通用策略梯度方法 (Policy gradient method)[40]、自然策略梯度方法 (Natural policy gradient)[41]、自然演员–评论员方法 (Natural actor-critic)[42];

免梯度方法: 从一组随机策略开始, 根据优胜劣汰的原则, 通过选择、删除和生成规则产生新的一组策略, 不断迭代这个过程以获取最优策略, 例如遗传算法 (Genetic algorithm)[43]、交叉熵方法 (Cross entropy methods)[44−45]、蚁群优化算法 (Ant colony optimizaition) 等.

2) 值函数近似指在值函数空间进行搜索, 又分为策略迭代 (Policy iteration) 和值迭代 (Value iteration).

策略迭代: 从一个随机策略开始, 通过策略评估 (Policy evaluation) 和策略改进 (Policy improvement) 两个步骤不断迭代完成, 例如最小二乘策略迭代 (Least squares policy iteration)[35, 46]、λ 最小二乘策略迭代 (Least squares λ policy iteration)[47−48]、改进的策略迭代 (Modified policy iteration)[49−50];

值迭代: 从一个随机值函数开始, 每步迭代更新改进值函数. 由于天然的在线学习 (Online learning) 特性, 值迭代是强化学习研究中的最重要的研究话题.

而根据函数模型的不同划分, 函数近似包括基于线性值函数近似的强化学习、基于核方法的强化学习、基于加性模型的强化学习和基于神经网络的强化学习. 下面我们从函数模型的角度介绍相关工作.

2.1 基于线性值函数近似的强化学习

线性值函数近似通过一组特征 ϕ 和对应权重 θ 的线性乘积来估计某个状态 s 的值:

$$V_\theta(s) = \sum_{i=1}^{n} \theta_i \phi_i(s) \tag{6}$$

在强化学习函数近似中, 线性函数近似因其简便实现以及简易分析的特点, 引起了强化学习研究者的广泛关注, 并得到了深入研究. 其相关工作主要从两个方面展开: 梯度法和最小二乘法.

梯度法: 1988 年, Sutton 首次提出了线性时间差分学习 (Linear temporal difference, linear TD) 以及 TD(λ) 算法[4], 并证明了线性 TD(0) 在最小化均方差 (Mean square error, MSE) 意义下的收敛性[4]. 1997 年, Tsitsiklis 等证明了线性 TD(λ) 的收敛性, 并给出了误差界[51]. 然而当学习率 α 或者资格跟踪参数 e 选得不合适时, 线性 TD(λ) 甚至会发散[32]. Bradtke 等在 1995 年提出的 "归一化线性 TD(λ)" (Normalized TD(λ))[32], 它不修改调整的方向, 而是限定调整值的大小, 以此减少 TD 不稳定行为的几率. 1995 年, Baird 等提出了一种的残差法 (Residual algorithms), 在保证残差梯度法的收敛性的同时, 提高了收敛速度[52]. 2008 年, Sutton 等提出了梯度时间差分学习 (Gradient temporal difference, GTD) 算法, 该算法的复杂度为 O(n), 适用于离策略的学习, 并且被证明可以收敛到最小二乘解[53]. 不过, GTD 算法与传统的线性 TD 方法比, 收敛速度要慢很多. 上述几种算法的目标是最小化 Bellman 均方差 (Mean square Bellman error, MSBE), 与此不同, 2009 年, Sutton 等提出了两种具有里程碑意义的新型算法, GTD 二代 (GTD2) 以及 TDC, 这两种算法的目标是最小化投影 Bellman 均方差 (Mean square projected Bellman error, MSPBE)[39]. Sutton 等人揭示了这两种算法才是真正的梯度下降方法 (换序二次偏导值相等), 它们的计算复杂度为 O(n), 并且收敛的速度比 GTD 要快很多, 但比直接梯度法和残差梯度法慢. 2010 年, Maei 通过最小化 λ-权重的 MSPBE, 得到了一个学习预测算法 GQ(λ)[54], 并且将之扩展为一个学习控制算法 Greedy-GQ[55].

最小二乘法: 1996 年, Bradtke 等提出了最小二乘时间差分算法 (Least square temporal difference, LSTD)[33]. Boyan 于 2002 年将 LSTD 扩展到 LSTD(λ)[34]. 2003 年 Lagoudakis 提出了最小二乘策略迭代算法 (Least square policy iteration, LSPI), 以获得更好的稳定性[35]. Bradtke 等于 1996 年根据增量求逆技巧, 提出了在线的递归 LSTD 算法 (Recursive LSTD, RLSTD)[33]. RLSTD 每步的计算复杂度依然是 O(n^2), 这对于很多具有大量特征的应用 (例如围棋的特征有 100 多万) 而言是不现实的[36−37]. 2006 年, Geramifard 提出了增量最小二乘时间差分学习 (iLSTD) 算法, 其计算复杂度为 O(n), 空间复杂度为 O(n^2), 相比于传统的 TD 方法, 具有更好的数据有效性, 相比于 LSTD 方法具有更好的计算有效性[56]. 同年, Geramifard 又提出了带资格跟踪的增量最小二乘时间差分学习 (iLSTD with eligibility traces), 并给出了收敛性证明[38]. 此外, 还有众多研究者提出了基于正则化方法的最小二乘迭代算法[57−61]. 2010 年, 为了解决高维度特征的强化学习问题, 尤其当特征数目超过样本数目时, Ghavamzadeh 提出了基于随机投影的 LSTD 方法 (LSTD with random projections, LSTD-RP)[62]; 2011 年, 为了解决在关联矩阵求逆中出现的近乎奇异问题, Bertsekas 提出了新的时间差分算法[63].

2.2 基于核方法的强化学习

1998 年, Sutton 等人提出了一类基于径向基函数网络 (Radial basis function, RBF) 的强化学习方法[2]. 2002 年, Ormoneit 等人明确提出了基于核方法的强化学习 (Kernel-based reinforcement

learning, KBRL), 确立了 KBRL 的研究方向[64]. 近年, KBRL 吸引了众多的国内外强化学习研究者. 在 2006 年的 International Conference on Machine Learning (ICML) 会议上, 专门设立了一个 Workshop on Kernel Machines and Reinforcement Learning 来讨论 KBRL 的问题. 国际著名期刊 IEEE Transactions on Neural Networks and Learning Systems 于 2012 年专门组织了 Special Issue on Online Learning in Kernel Methods.

根据表示定理, 基于核方法的强化学习的值函数通过一组核函数 $k(\cdot,\cdot)$ 和对应权重 θ 的线性乘积来估计某个状态 s 的值:

$$V_\theta(s)=\sum_{i=1}^{n}\theta_i k(s,s_i) \tag{7}$$

其中, 集合 $\{s_i\}$ 称为字典 (Dictionary/D). 基于核方法的强化学习需要考虑以下三个问题: 核函数 $k(\cdot,\cdot)$ 如何选择、字典 D 能否稀疏化构造、以及值函数的参数怎样估计.

核函数 $k(\cdot,\cdot)$ 的选择: 不同的核函数适用于不同的强化学习问题. 对于复杂或困难的强化学习问题, 单个核函数并不有效、甚至无法求解. 此外, 在目前关于 KBRL 的研究中, 核函数都是根据经验或者实验人员的试错决定的. 因此, 针对任意的强化学习问题, 如何找到一个普适的方法或原则来自动选择核函数, 将成为 KBRL 研究的热点.

字典 D 的稀疏化构造: 基于核方法的值函数有一对矛盾: 1) 字典 D 中元素 (s_i) 个数越多, 值函数的表达能力越强; 2) 字典 D 中元素 (s_i) 个数越多, 则值函数的复杂度越大, 越不利于参数学习.

在 KBRL 的研究初期, 字典 D 是预先设定好的, 如文献 [2, 64−73]. 此后, 字典的自动稀疏化构造方法得到了关注: 1) 近似线性依赖 (Approximate linear dependence, ALD) 方法[74−77], 该方法的单步计算复杂度为 $O(n^2)$; 2) 核界定感知方法 (Bounded kernel-based perceptron)[78−80]; 3) 基于选择性集成学习 (Selective ensemble learning) 的字典稀疏化, 如基于核距离的在线稀疏化方法[81].

值函数的参数估计: 基于单个核函数的值函数可以通过多种方法来进行参数估计, 如高斯过程时间差分学习 (Gaussian process temporal difference, GPTD)[74−75]、基于核方法的奖赏回归 (Kernel rewards regression, KRR)[65]、基于核方法的优先排序遍历方法 (Kernel-based prioritized sweeping, KBPS)[66]、基于核方法的稀疏最小二乘时间差分学习方法 (Kernel-based LSTD, KLSTD)[76]、基于核方法的最小二乘策略迭代方法 (Kenel-based least-squares policy iteration, KLSPI)[77]、Bellman 残差最小化 (Bellman residual minimization)[69]、Bellman 残差消除算法 BRE(SV)[70]、基于核密度估计的无参动态规划 (Non-parametric dynamic programming, NPDP)[73]、以及基于核方法的在线选择时间差分学习 (Online selective kernel-based temporal difference learning, OSKTD)[81].

多核学习方法是当前机器学习领域的一个新的热点[82]. 在监督学习中, 多核学习方法是将多个核函数进行组合, 可以用于求解数据异构、数据不规则、样本规模巨大、样本不均匀分布等问题[83−84]. 因此, 将多核学习引入 KBRL 有利于求解复杂或困难的强化学习问题.

2.3 基于加性模型的强化学习

加性模型在监督学习中的使用较为常见, 例如 Boosting 方法得到的模型都是加性模型, 这一类模型将多个现有模型结合起来, 有很强的表示能力, 而在强化学习中基于加性模型的方法还很少. 目前已有的基于加性模型的强化学习方法都是策略梯度方法, 因此这些方法直接用加性模型来表示策略, 而不表示值函数:

$$\pi(s)=\sum_{i=1}^{k}\theta_i f_i(s)$$

其中, f_i 可以是任意现有监督学习回归模型, 例如决策树或者神经网络. 加性模型的学习, 不涉及每个基模型 f_i 的学习, 通常假设 f_i 可由现有学习方法来解决, 例如线性回归算法、随机森林等经典机器学习算法. 加性模型的学习, 是基于损失函数 (长期累积奖赏) 的泛函梯度, 对 f_i 一个一个的顺序学习, 已减少损失函数残差.

NPPG 方法是第一个基于加性模型的策略梯度算法[85]. 然而后来发现, NPPG 仅在简单问题上有效, 问题复杂时会出现过拟合问题, 影响了对策略的搜索[86], 由此提出了 PolicyBoost 方法[86]. 需要注意的是, 加性模型的训练每一轮会增加一个基模型, 当迭代次数很多时, 加性模型自身的计算开销就会很大, 由此提出了 Napping 方法来将线性增加的模型数量降到了常数大小[87].

2.4 基于神经网络的强化学习

基于神经网络的强化学习顾名思义采用神经网络作为函数近似的模型.

多层神经网络: 克服了单层感知器不能进行非线性分类问题, 多层神经网络于上世纪 80 年代强势回归. 神经网络用于求解强化学习问题稍晚一些, 早期的代表作是 1995 年前后轰动一时的 TD-Gammon. 它采用了三层神经网络模型 (即输入层、

隐含层、输出层), 结合 TD(λ) 学习、自我博弈、梯度下降的误差反向传播法则以及多步约简搜索. 其中, TD-Gammon 1.0 版本并未采用任何西洋双陆棋戏 (Backgammon) 的领域知识, 达到了当时电脑程序的最佳水平; TD-Gammon 2.0 版本添加了基于领域知识的特征作为神经网络的输入, 并结合两步搜索, 达到了人类顶级专家的水平[88−89].

基于三层神经网络模型的强化学习算法, 其性能 (不考虑参数更新方式) 依赖于首层的网络输入, 一旦使用了专家级的领域知识, 问题将变得容易求解, 效果也会很好. 然而在实际应用中, 专家级的领域知识并不容易获取, 如何在没有领域知识的情况下获得高性能成为了近年来强化学习的研究热潮.

深度神经网络: 深度学习模型是深层的神经网络, 即有多个 (三个以上) 隐含层. 自 2006 年开始, 深度学习在语音识别、手写数字识别等图像、视频、语音和音频处理取得了突破性进展[90−91]. 深度学习的成功在于, 它把原始数据通过一些简单的但是非线性的模型转变成为更高层次的, 更加抽象的表达. 这个过程不需要利用人工工程来设计的, 而是使用一种通用的学习过程从数据中学习的. 因此, 深度学习实际上是一种特征学习方法. Conference on Neural Information Processing Systems (NIPS) 从 2015 年起开始举办深度强化学习研讨会, 预示着深度强化学习正在迅速发展.

深度学习用于强化学习的研究同样滞后于监督学习. 2010 年, Deep fitted Q-iteration (DFQ) 采用基于深度自动编码器的神经网络学习图像的特征, 结合批量 Q 学习, 获取了路径寻优策略[92]. 其中, 与常见深度学习模型多层受限玻尔兹曼机 (Restricted Boltzmann machine, RBM) 不同, DFQ 采用了深层感知器模型, 使用 RProp 规则[93] 进行逐层预训练和微调获得编码器, 并使用该编码器构造特征. 在同样的图像数据集上的对比实验, 结果表明深度学习只需要两个维度就能重现出比主成分分析方法 (Principal component analysis, PCA) 好的图像. 由于深度神经网络在状态表示上显示出的优势, 基于深度神经网络的 “深度强化学习” 研究呈现井喷涌现的势态, 研究论文主要出现在 ICML、NIPS 等会议上.

3 强化学习在游戏上的应用

3.1 单人游戏

单人游戏即一个玩家在预设的游戏场景中按照游戏规则进行行动, 目标通常是最大化得分 (例如俄罗斯方块中方块会不断掉落, 目标是取得的分越高越好) 或到达目标状态 (例如吃豆人游戏中, 目标是吃掉所有的豆并且不被幽灵吃掉).

单人游戏定义了机器所在的环境, 机器在游戏中自动学习如何取得目标, 这一问题恰好符合强化学习的框架, 因此可以使用强化学习来解决: 通过游戏的局面定义机器可观察的状态, 游戏中通常已经定义了可以使用的动作, 单步游戏的得分作为立即奖赏, 或立即奖赏设置为 0, 达到目标获得奖赏 1, 最大步长未达目标获得奖赏 −1.

在强化学习用于单人游戏方面近年最受关注的工作, 是 2013 年 Google DeepMind 团队在 NIPS 的深度学习 Workshop 上提出的 DQN (Deep Q-networks) 算法[94], 通过直接输入游戏的原始视频作为状态进行强化学习, 在 “雅达利” (Atari) 游戏平台中 7 款游戏的 6 款上都超过了以往的算法, 并在其中 3 款游戏超过了人类水平. 该工作 2015 年扩展发表在 Nature 上[95], 在 49 款游戏上达到了人类水平.

DQN 是一种基于函数近似的 Q 学习算法, 在 Q 学习的基础上主要做了两项改进: 采用深度卷积神经网络模型, 以直接从视频输入作为状态, 省去了状态特征的人工设计; 维护了一个 “重放” 集合, 记录许多以往策略执行产生的历史数据, 在更新神经网络时, 与 Q 学习中只用一步数据进行更新不同, DQN 在大量的 “重放” 历史数据上更新.

基于 Atari 游戏平台, 深度强化学习最近得到了迅速发展. 例如, Oh 等[96] 基于动作的条件转化深度网络被用于改进深度强化学习在 Atari 上的策略表示. 其中值得一提的是强化学习与搜索的结合: Guo 等[97] 利用离线蒙特卡洛树搜索方法产生大量的游戏操作数据, 并基于 “模仿学习” 的方法训练基于深度学习的策略, 在 Atari 游戏平台上取得了超过 DQN 的游戏水平. 蒙特卡洛树搜索 (Monte Carlo tree search)[98] 是一种最好优先的树搜索方法 (Best-first tree search), 与经典 A* 算法类似, 在树搜索过程中, 每次选择从预估最好的树节点状态往下进行. 与 A* 算法不同的是, 蒙特卡洛树搜索不需要设计启发式函数来对节点的好坏进行评估, 而是通过蒙特卡洛采样来进行评估. 如图 3 所示, 蒙特卡洛树搜索在现有的搜索树上选择当前最优叶节点, 将叶节点展开出子节点, 并且进入其中一个子节点; 对于该子节点状态的评估, 蒙特卡洛树搜索采用 “蒙特卡洛” 采样的方法: 随机行走下去, 直到游戏终止, 并且使用游戏终止时的得分作为该次蒙特卡洛采样的结果; 该结果用以更新叶节点上的估值, 以及回溯到根节点路基上每一个节点的估值. 一种简单的估值的方法是对所有的采样求平均值, 然而在采样有限的情况下, 平均值可能存在较大偏差, 因此更常用的估值方法是采用平均值加上与采样数量有

关的置信度, 即上置信度界方法 (Upper confidence bound, UCB). 蒙特卡洛树搜索方法是一种状态搜索方法, 在游戏中可以取得很好的效果, 然而其应用的不足在于每做一次决策需要进行大量搜索, 时间开销很大, 而基于学习的方法在做决策时只需要根据策略模型计算出决策动作即可, 效率很高. 因此 Guo 等[97] 提出的方法是在训练阶段, 使用蒙特卡洛树搜索方法产生接近最优的决策序列, 并采用模仿学习方式, 直接从决策序列中学习策略模型.

3.2　对弈游戏

对弈游戏有两个或以上的玩家参与, 因此与经典的强化学习环境有所不同. 考虑两个玩家的情况, 常用的训练强化学习的方式是将对手固定下来, 这样对手属于环境的一部分, 从而转化为单人游戏. 在训练阶段, 对手的选择可以使用随机策略, 现有的优秀策略, 或者当前训练得到策略的拷贝. 使用当前训练策略的拷贝时, 在更新策略后, 也重新拷贝策略更新对手, 因而形成 “自我对下” 的情形. 值得注意的是, 对手的选择对强化学习的影响很大, 即使在完全信息游戏 (例如围棋、象棋、黑白棋等) 上, 除非已逼近最优策略, 否则学习到的策略仅仅是针对对手策略更优的, 而对于其他策略则可能更差, 例如在黑白棋上[99], 基于 Q 学习对 BENCH 方法训练的策略与 BENCH 对弈胜率可超过 80 %, 而与 HEUR 方法对弈胜率不到 40 %, 同时对 HEUR 方法训练的策略对弈 HEUR 时胜率超过 80 %, 而与 BENCH 对弈胜率只有 58 %.

在双人对弈的完全信息游戏中, 围棋被认为是最为复杂的项目. 由于围棋的动作候选可达 300, 通常需对战 100 步以上, 巨大的状态和动作空间使得以往的强化学习方法只能在缩小的棋盘上取得较好的效果[100−101].

2015 年 10 月, DeepMind 的 AlphaGo[102] 战胜围棋职业二段樊辉, 2016 年 3 月, 以 4∶1 的成绩战胜职业九段韩国围棋选手李世石, 受到了全球关注. AlphaGo 是蒙特卡洛树搜索和强化学习结合的系统, 以蒙特卡洛树搜索为主干. AlphaGo 利用强化学习对蒙特卡洛树搜索进行三处主要改进. 首先, 通过模仿学习, 从人类专家的对弈数据中学出模仿人类的策略, 用在蒙特卡洛树搜索的采样阶段, 与模式匹配一起用来替代随机的采样, 可极大地提高少量样本时的采样估计; 其次, 在模仿学习的基础上, 使用策略梯度学习方法, 通过拷贝训练策略产生对手, 并使用深度卷积网络作为策略的表示, 来进行强化学习, 学习得到的策略 (被称为策略网络), 被用来在蒙特卡洛树搜索展开节点时, 为子节点的选择产生顺序; 在强化学习进行的同时, 从训练产生的数据中学习回归模型, 得到估值网络, 用来直接给出一个节点的估值, 该估值与蒙特卡洛树搜索采样得到的估值加权求和, 作为最终蒙特卡洛树搜索的估值.

基于深度神经网络的强化学习已成为当前的研究热点, 尽管已取得上述突破, 深度强化学习还需要在更多的理论和应用中进行深入研究. 例如, 当前的深度强化学习研究只采用了三种深度神经网络模型: 自动编码器、感知器模型、卷积神经网络, 其他常见的深度神经网络如深度信念网络、深度稀疏编码等尚未应用于强化学习问题等等. 强化学习用于游戏的详细回顾, 可参考文献 [103].

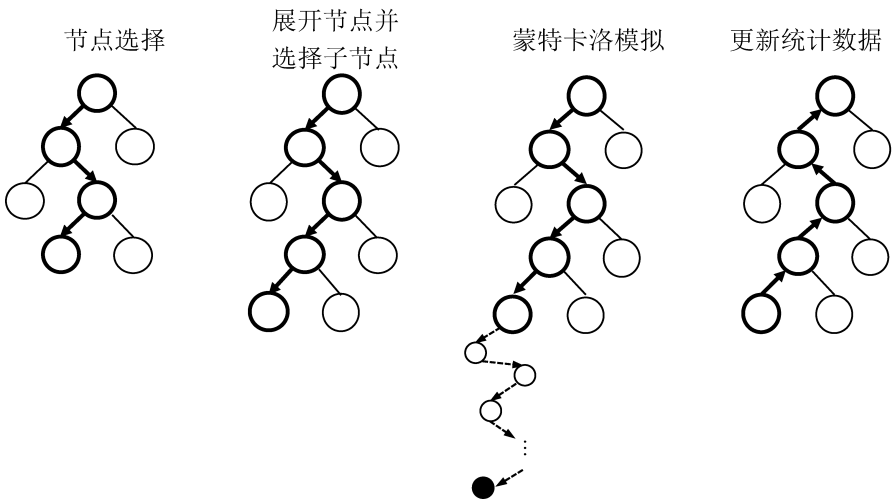

图 3　蒙特卡洛树搜索

Fig. 3　Monte-Carlo tree search

4 总结

本文简要介绍了强化学习, 重点集中在基于函数近似的强化学习方法, 这些方法通过将值函数或策略函数用模型来表示, 可直接在巨大离散状态空间或连续状态空间中进行强化学习, 比经典强化学习的方法更加贴近实际问题. 基于深度神经网络在特征抽取上的进展, 强化学习在围棋等问题上发挥了重要作用, 取得了令人瞩目的突破. 围棋等游戏是现实问题的抽象和精简, 成功解决游戏中问题, 为解决实际问题奠定基础。

然而强化学习还有许多有待解决的问题, 例如, 策略梯度方法的优化目标 (累积奖赏) 是关于策略参数的高度非凸函数, 梯度方法只能帮助改善策略, 而无法取得最优策略, 是否能够有效解决强化学习面临的非凸问题优化; 在 AlphaGo 训练时, 使用了大量的人类对弈数据, 并且自我对弈了千万局, 而在更多的任务中, 难以获得如此巨大的数据或大量的模拟, 如何在少量数据和模拟次数的条件下取得良好性能等等.

致谢

感谢南京大学高阳教授领导的学习和推理研究组, 本文部分内容得益于该研究组的帮助.

References

1 Zhou Zhi-Hua. *Machine Learning*. Beijing: Tsinghua University Press, 2016.
(周志华. 机器学习. 北京: 清华大学出版社, 2016.)

2 Sutton R S, Barto A G. *Reinforcement Learning: an Introduction*. Cambridge, MA: 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
(高阳, 陈世福, 陆鑫. 强化学习研究综述. 自动化学报, 2004, **30**(1): 86−100)

4 Sutton R S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988, **3**(1): 9−44

5 Dayan P. The convergence of TD (λ) for general λ. *Machine Learning*, 1992, **8**(3−4): 341−362

6 Singh S, Jaakkola T, Littman M L, Szepesvári C. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 2000, **38**(3): 287−308

7 Watkins C J C H, Dayan P. Q-learning. *Machine Learning*, 1992, **8**(3−4): 279−292

8 Jaakkola T, Jordan M I, Singh S P. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 1994, **6**(6): 1185−1201

9 Tsitsiklis J N. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 1994, **16**(3): 185−202

10 Barto A G, Mahadevan S. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 2003, **13**(4): 341−379

11 Dietterich T G. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 2000, **13**: 227−303

12 Dietterich T G. The MAXQ method for hierarchical reinforcement learning. In: Proceedings of the 15th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1998. 118−126

13 Morimoto J, Doya K. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. *Robotics and Autonomous Systems*, 2001, **36**(1): 37−51

14 Dietterich T G. An overview of MAXQ hierarchical reinforcement learning. In: Proceedings of the 4th International Symposium on Abstraction, Reformulation, and Approximation. Horseshoe Bay, USA: Springer, 2000. 26−44

15 Marthi B, Russell S, Latham D, Guestrin C. Concurrent hierarchical reinforcement learning. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. 779−785

16 Hengst B. Discovering hierarchy in reinforcement learning with HEXQ. In: Proceedings of the 19th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 2002. 243−250

17 Mehta N, Natarajan S, Tadepalli P, Fern A. Transfer in variable-reward hierarchical reinforcement learning. *Machine Learning*, 2008, **73**(3): 289−312

18 Ravindran B, Barto A G. Model minimization in hierarchical reinforcement learning. In: Proceedings of the 5th International Symposium on Abstraction, Reformulation, and Approximation. Kananaskis, Alberta, Canada: Springer, 2002. 196−211

19 Ghavamzadeh M, Mahadevan S. Continuous-time hierarchical reinforcement learning. In: Proceedings of the 18th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2001. 186−193

20 Goel S, Huber M. Subgoal discovery for hierarchical reinforcement learning using learned policies. In: Proceedings of the 16th International FLAIRS Conference. St. Augustine, Florida, USA: AAAI, 2003. 346−350

21 Taylor M E, Whiteson S, Stone P. Transfer via intertask mappings in policy search reinforcement learning. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-agent Systems. New York: ACM, 2007. Article No. 37

22 Taylor M, Stone P. Transfer learning for reinforcement learning domains: a survey. *The Journal of Machine Learning Research*, 2009, **10**: 1633−1685

23 Konidaris G, Barto A G. Building portable options: skill transfer in reinforcement learning. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. Hyderabad, India: AAAI, 2007. 895−900

24 Taylor M E, Stone P. Behavior transfer for value-function-based reinforcement learning. In: Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems. Utrecht, The Netherlands: ACM Press, 2005. 53−59

25 Taylor M E, Stone P. Cross-domain transfer for reinforcement learning. In: Proceedings of the 24th International Conference on Machine Learning. New York, NY, USA: ACM, 2007. 879−886

26 Liu Y X, Stone P. Value-function-based transfer for reinforcement learning using structure mapping. In: Proceedings of the 21st National Conference on Artificial Intelligence. Boston, MA: AAAI, 2006. 415−420

27 Torrey L, Walker T, Shavlik J, Maclin R. Using advice to transfer knowledge acquired in one reinforcement learning task to another. In: Proceedings of the 16th European Conference on Machine Learning on Machine Learning: ECML 2005. Porto, Portugal: Springer, 2005. 412−424

28 Mahadevan S. Enhancing transfer in reinforcement learning by building stochastic models of robot actions. In: Proceedings of the 9th International Workshop on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. 290−299

29 Torrey L, Shavlik J, Walker T, Maclin R. Relational macros for transfer in reinforcement learning. In: Proceedings of the 17th International Conference on Inductive Logic Programming. Corvallis, OR, USA: Springer, 2008. 254−268

30 Wang Hao, Gao Yang, Chen Xing-Guo. Transfer of reinforcement learning: the state of the art. *Acta Electronica Sinica*, 2008, **36**(12A): 39−43
(王皓, 高阳, 陈兴国. 强化学习中的迁移: 方法和进展. 电子学报, 2008, **36**(12A): 39−43)

31 Bertsekas D P, Tsitsiklis J N, Tsitsiklis J. *Neuro-dynamic Programming*. Athens, Greece: Athena Scientific Press, 1996.

32 Bradtke S J. Incremental Dynamic Programming for On-line Adaptive Optimal Control [Ph. D. dissertation], University of Massachusetts, USA, 1995.

33 Bradtke S J, Barto A G. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 1996, **22**(1): 33−57

34 Boyan J A. Technical update: least-squares temporal difference learning. *Machine Learning*, 2002, **49**(2): 233−246

35 Lagoudakis M G, Parr R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 2003, **4**: 1107−1149

36 Silver D, Sutton R S, Müller M. Reinforcement learning of local shape in the game of Go. In: Proceedings of the 20th International Joint Conferences on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007. 1053−1058

37 Silver D, Sutton R S, Müller M. Temporal-difference search in computer Go. *Machine Learning*, 2012, **87**(2): 183−219

38 Geramifard A, Bowling M H, Zinkevich M, Sutton R S. iLSTD: eligibility traces and convergence analysis. In: Advances in Neural Information Processing Systems 19. Cambridge, MA: MIT Press, 2006. 441−448

39 Sutton R S, Maei H R, Precup D, Bhatnagar S, Silver D, Szepesvári C, Wiewiora E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proceedings of the 26th Annual International Conference on Machine Learning. New York: ACM, 2009. 993−1000

40 Sutton R S, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems 12. Cambridge, MA: MIT Press, 1999. 1057−1063

41 Kakade S M. A natural policy gradient. In: Advances in Neural Information Processing Systems 14. Cambridge, MA: MIT Press 2001. 1531−1538

42 Peters J, Schaal S. Natural actor-critic. *Neurocomputing*, 2008, **71**(7−9): 1180−1190

43 Böhm N, Kókai G, Mandl S. An evolutionary approach to Tetris. In: Proceedings of the 6th Metaheuristics International Conference (MIC2005). Vienna, Austria, 2005.

44 Szita I, Lörincz A. Learning Tetris using the noisy cross-entropy method. *Neural Computation*, 2006, **18**(12): 2936−2941

45 Thiery C, Scherrer B. Improvements on learning Tetris with cross entropy. *International Computer Games Association Journal*, 2009, **32**(1): 23−33

46 Lagoudakis M G, Parr R, Littman M L. Least-squares methods in reinforcement learning for control. In: Proceedings of the 2nd Hellenic Conference on AI, SETN, Methods and Applications of Artificial Intelligence. Thessaloniki, Greece, 2002. 249−260

47 Thiery C, Scherrer B. Least-squares λ policy iteration: bias-variance trade-off in control problems. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). Haifa, Israel, 2010. 1071−1078

48 Scherrer B. Performance bounds for λ policy iteration and application to the game of Tetris. *The Journal of Machine Learning Research*, 2013, **14**: 1181−1227

49 Gabillon V, Ghavamzadeh M, Scherrer B. Approximate dynamic programming finally performs well in the game of Tetris. In: Advances in Neural Information Processing Systems 26. Lake Tahoe, Nevada, USA: Curran Associates, Inc., 2013. 1754−1762

50 Scherrer B, Ghavamzadeh M, Gabillon V, Lesner B, Geist M. Approximate modified policy iteration and its application to the game of Tetris. *The Journal of Machine Learning Research*, 2015, **16**(1): 1629−1676

51 Tsitsiklis J N, Van Roy B. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 1997, **42**(5): 674−690

52 Baird L. Residual algorithms: reinforcement learning with function approximation. In: Proceedings of the 12th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1995. 30−37

53 Sutton R S, Szepesvári C, Maei H R. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In: Proceedings of the 21st Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2008. 1609−1616

54 Maei H R, Sutton R S. GQ(λ): a general gradient algorithm for temporal-difference prediction learning with eligibility traces. In: Proceedings of the 3rd Conference on Artificial General Intelligence. Paris: Atlantis Press, 2010. 91−96

55 Maei H R, Szepesvái C, Bhatnagar S, Sutton R S. Toward off-policy learning control with function approximation. In: Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010. 719−726

56 Geramifard A, Bowling M, Sutton R S. Incremental least-squares temporal difference learning. In: Proceedings of the 21st AAAI Conference on Artificial Intelligence. Boston, MA: AAAI, 2006. 356−361

57 Loth M, Davy M, Preux P. Sparse temporal difference learning using LASSO. In: Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning. Honolulu, HI: IEEE, 2007. 352−359

58 Kolter J Z, Ng A Y. Regularization and feature selection in least-squares temporal difference learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: ACM 2009. 521−528

59 Hoffman M W, Lazaric A, Ghavamzadeh M, Munos R. Regularized least squares temporal difference learning with nested l_2 and l_1 penalization. In: Proceedings of the 9th European Workshop on Recent Advances in Reinforcement Learning. Athens, Greece: Springer, 2012. 102−114

60 Ghavamzadeh M, Lazaric A, Munos R, Hoffman M. Finite-sample analysis of lasso-TD. In: Proceedings of the 28th International Conference on Machine Learning. Bellevue, Washington, USA, 2011. 1177−1184

61 Geist M, Scherrer B, Lazaric A, Ghavamzadeh M. A Dantzig selector approach to temporal difference learning. In: Proceedings of the 29th International Conference on Machine Learning. Edinburgh, Scotland, UK, 2012.

62 Ghavamzadeh M, Lazaric A, Maillard O A, Munos R. LSTD with random projections. In: Advances in Neural Information Processing Systems 23. Vancouver, British Columbia, Canada: Curran Associates, Inc., 2010. 721−729

63 Bertsekas D P. Temporal difference methods for general projected equations. *IEEE Transactions on Automatic Control*, 2011, **56**(9): 2128−2139

64 Ormoneit D, Sen Ś. Kernel-based reinforcement learning. *Machine Learning*, 2002, **49**(2−3): 161−178

65 Schneegaβ D, Udluft S, Martinetz T. Kernel rewards regression: an information efficient batch policy iteration approach. In: Proceedings of the 24th IASTED International Multi-Conference on Artificial Intelligence and Applications. Innsbruck, Austria, 2006. 428−433

66 Jong N K, Stone P. Kernel-based models for reinforcement learning. In: Proceedings of the ICML-06 Workshop on Kernel Machines for Reinforcement Learning. Pittsburgh, PA, 2006.

67 Deisenroth M P, Peters J, Rasmussen C E. Approximate dynamic programming with Gaussian processes. In: Proceedings of the 2008 American Control Conference. Seattle, WA: IEEE, 2008. 4480−4485

68 Reisinger J, Stone P, Miikkulainen R. Online kernel selection for Bayesian reinforcement learning. In: Proceedings of the 25th International Conference on Machine Learning. Helsinki, Finland, 2008. 816−823

69 Antos A, Szepesvári C, Munos R. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008, **71**(1): 89−129

70 Bethke B, How J P, Ozdaglar A. Kernel-based reinforcement learning using Bellman residual elimination. *Journal of Machine Learning Research*, to be published

71 Bethke B, How J P. Approximate dynamic programming using Bellman residual elimination and Gaussian process regression. In: Proceedings of the 2009 American Control Conference. St. Louis, MO: IEEE, 2009. 745−750

72 Taylor G, Parr R. Kernelized value function approximation for reinforcement learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Quebec, Canada, 2009. 1017−1024

73 Kroemer O B, Peters J. A non-parametric approach to dynamic programming. In: Proceedings of Advances in Neural Information Processing Systems 24. Granada, Spain: Curran Associates, Inc., 2011. 1719−1727

74 Engel Y, Mannor S, Meir R. Bayes meets Bellman: the Gaussian process approach to temporal difference learning. In: Proceedings of the 20th International Conference on Machine Learning. Washington DC: AAAI, 2003. 154−161

75 Engel Y, Mannor S, Meir R. Reinforcement learning with Gaussian processes. In: Proceedings of the 22nd International Conference on Machine Learning. Bonn, Germany, 2005. 201−208

76 Xu X. A sparse kernel-based least-squares temporal difference algorithm for reinforcement learning. In: Proceedings of the 2nd International Conference on Advances in Natural Computation. Xi' an, China: Springer, 2006. 47−56

77 Xu X, Hu D W, Lu X C. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 2007, **18**(4): 973−992

78 Orabona F, Keshet J, Caputo B. The projectron: a bounded kernel-based perceptron. In: Proceedings of the 25th International Conference on Machine Learning. Helsinki, Finland: ACM, 2008. 816−823

79 Orabona F, Keshet J, Caputo B. Bounded kernel-based online learning. *The Journal of Machine Learning Research*, 2009, **10**: 2643−2666

80 Robards M, Sunehag P, Sanner S, Marthi B. Sparse Kernel-SARSA (λ) with an eligibility trace. In: Proceedings of the 22nd European Conference on Machine Learning and Knowledge Discovery in Databases. Athens, Greece: Springer, 2011. 1−17

81 Chen X G, Gao Y, Wang R L. Online selective kernel-based temporal difference learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2013, **24**(12): 1944−1956

82 Wang Hong-Qiao, Sun Fu-Chun, Cai Yan-Ning, Chen Ning, Ding Lin-Ge. On multiple kernel learning methods. *Acta Automatica Sinica*, 2010, **36**(8): 1037−1050
(汪洪桥, 孙富春, 蔡艳宁, 陈宁, 丁林阁. 多核学习方法. 自动化学报, 2010, **36**(8): 1037−1050)

83 Orabona F, Jie L, Caputo B. Multi kernel learning with online-batch optimization. *The Journal of Machine Learning Research*, 2012, **13**: 227−253

84 Tobar F A, Kung S-Y, Mandic D P. Multikernel least mean square algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 2014, **25**(2): 265−277

85 Kersting K, Driessens K. Non-parametric policy gradients: a unified treatment of propositional and relational domains. In: Proceedings of the 25th International Conference on Machine Learning (ICML′08). Helsinki, Finland: ACM, 2008. 456−463

86 Yu Y, Hou P-F, Da Q, Qian Y. Boosting nonparametric policies. In: Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS′16). Singapore, 2016.

87 Da Q, Yu Y, Zhou Z-H. Napping for functional representation of policy. In: Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS′14). Paris, France, 2014. 189−196

88 Tesauro G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 1994, **6**(2): 215−219

89 Tesauro G. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 1995, **38**(3): 58−68

90 Hinton G E, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006, **18**(7): 1527−1554

91 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, **521**(7553): 436−444

92 Lange S, Riedmiller M. Deep auto-encoder neural networks in reinforcement learning. In: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN). Barcelona: IEEE, 2010. 1−8

93 Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the IEEE International Conference on Neural Networks. San Francisco, CA: IEEE, 1993. 586−591

94 Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing Atari with deep reinforcement learning. In: Proceedings of Deep Learning, Neural Information Processing Systems Workshop. Harrahs and Harveys, Lake Tahoe, USA, 2013.

95 Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. *Nature*, 2015, **518**(7540): 529−533

96 Oh J, Guo X X, Lee H, Lewis R, Singh S. Action-conditional video prediction using deep networks in Atari games. In: Proceedings of Advances in Neural Information Processing Systems 28. Montreal, Quebec, Canada, 2015. 2845−2853

97 Guo X X, Singh S, Lee H, Lewis R L, Wang X S. Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In: Proceedings of Advances in Neural Information Processing Systems 27. Montreal, Quebec, Canada, 2014. 3338−3346

98 Kocsis L, Szepesvári C. Bandit based Monte-Carlo planning. In: Proceedings of the 17th European Conference on Machine Learning. Berlin, Heidelberg: Springer-Verlag, 2006. 282−293

99 Van Der Ree M, Wiering M. Reinforcement learning in the game of Othello: learning against a fixed opponent and learning from self-play. In: Proceedings of the 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). Singapore: IEEE, 2013. 108−115

100 Gelly S, Silver D. Achieving master-level play in 9×9 computer Go. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence. Chicago, Illinois: AAAI, 2008. 1537−1540

101 Silver D, Sutton R S, Müller M. Sample-based learning and search with permanent and transient memories. Proceedings of the 25th International Conference on Machine Learning. New York: ACM, 2008. 968−975

102 Silver D, Huang A, Maddison C J, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search . *Nature*, 2016, **529**(7587): 484−489

103 Szita I. Reinforcement learning in games. *Reinforcement Learning*. Berlin Heidelberg: Springer-Verlag, 2012. 539−577

陈兴国 南京邮电大学计算机学院/软件学院讲师. 2014 年获得南京大学计算机系博士学位. 主要研究方向为机器学习, 强化学习.
E-mail: chenxg@njupt.edu.cn
(**CHEN Xing-Guo** Lecturer at the School of Computer Science & Technology and the School of Software, Nanjing University of Posts and Telecommunications. He received his Ph. D. degree from Nanjing University. His research interest covers machine learning and reinforcement learning.)

俞 扬 南京大学计算机系副教授, 2011 年获得南京大学计算机系博士学位. 主要研究方向为机器学习, 演化学习, 强化学习. 本文通信作者.
E-mail: yuy@nju.edu.cn
(**YU Yang** Associate professor in the Department of Computer Science and Technology, Nanjing University. He received his Ph. D. degree from the Department of Computer Science and Technology, Nanjing University in 2011. His research interest covers machine learning, evolutionary learning, reinforcement learning. Corresponding author of this paper.)