强化学习算法中启发式回报函数的设计及其收敛性分析*)

魏英姿[1,2,3] 赵明扬[1]

(中国科学院沈阳自动化所机器人学重点实验室 沈阳110016)[1] (沈阳理工大学 沈阳110168)[2]

(中国科学院研究生院 北京100039)[3]

摘 要 回报函数设计的好与坏对学习系统性能有着重要作用,按回报值在状态-动作空间中的分布情况,将回报函数的构建分为两种形式:密集函数和稀疏函数,分析了密集函数和稀疏函数的特点。提出启发式回报函数的基本设计思路,利用基于保守势函数差分形式的附加回报函数,给学习系统提供更多的启发式信息,并对算法的最优策略不变性和迭代收敛性进行了证明。启发式回报函数能够引导学习,加快学习进程,从而可以实现强化学习在实际大型复杂系统应用中的实时控制和调度。

关键词 强化学习,回报函数,马尔可夫决策过程,策略,收敛性

Design and Convergence Analysis of a Heuristic Reward Function for Reinforcement Learning Algorithms

WEI Ying-Zi[1,2,3] ZHAO Ming-Yang[1]

(Robotics Laboratory, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016)[1]

(Shenyang University of Technology, Shenyang 110168)[2] (Graduate School of the Chinese Academy of Sciences, Beijing 100039)[3]

Abstract The reward function has become the critical component for its effect of evaluating the action and guiding the reinforcement learning (RL) process. According to the distribution of rewards in the space of states, reward functions can have two basic forms, dense and sparse. Their effects act on RL algorithm performance differently. Sparse reward functions are more difficult to learn a value function for than dense ones. The idea of designing a heuristic reward function is proposed in this paper. The practice of the heuristic reward function in RL consists of supplying additional rewards to a learning system, beyond those supplied by the underlying Marko Decision Process (MDP). We can add a reward for transitions between states that is expressible as the difference in value of an arbitrary potential function applied to those states. The additional reward function F, based on a reward for transitions between states, is a difference of conservative potentials. The additional training reward F will provide more heuristic information and be used to guide the learning system to progress rapidly. The gradient inherent in heuristic reward functions tends to give more leverage when learning the value function. The proof of convergence of Q-value iteration is presented under a more general model of MDP, too. The heuristic reward function helps to implement an efficient reinforcement learning system on a real-time control or scheduling system.

Keywords Reinforcement learning, Reward function, Markov decision process, Policy, Convergence

1 引言

强化学习(Reinforcement Learning,RL)是近年来机器学习和人工智能领域内发展起来的一种重要的优化方法,在智能控制、机器人系统规划及分析预测等领域有许多应用。"强化"和"强化学习"一词出自行为心理学,由 Minsky 首次提出[1]。当时数学心理学家探索了各种计算模型,以解释动物和人类的学习行为。目前,在机器学习领域已把它与运筹学的相关理论相结合,形成一类算法,采用动态规划的基本思想和结构,汲取人工神经网络、计算机仿真、认知科学、人工智能等领域的成果和方法,通过对实际系统的仿真,对系统自身进行优化,提高系统的性能。Bertsekas 和 Tsitsiklis[2]提出的"神经元动态规划"则强调以神经网等为映射的函数拟合器,把神经网络与动态规划相结合作为研究基础。

RL 是一种在线动作-评价的机器学习方法。系统与外界环境之间的交互是学习系统获得智能的主要来源。Sutton[3]把它称为是一种直接的自适应优化控制。强化学习中的强化信号对产生动作的好坏作一种评价(通常为标量信号),而不是告诉强化学习系统(Reinforcement Learning System,RLS)如何去产生正确的动作。由于外部环境提供的信息很少,RLS 必须靠自身的经历进行学习。通过这种方式,RLS 在行动-评价的环境中获得知识,改进行动方案以适应环境。所以强化学习系统对环境具有自适应性。

Sutton 于1988年在《Machine Learning》上发表了题为"Learning to Predict by the Methods of Temporal Differences"的著名论文。奠定了 TD 算法的基础。Watkins 于1989年在其博士论文"learning from delayed rewards"提出了 Q-Learning 并对 Q-Learning 方法的收敛性进行了证明。为加快学习 Peng & williams 将 $TD(\lambda)$ 和 Q 学习相结合,提出了多步增量 $Q(\lambda)$ 学习。在上世纪六七十年代,强化学习研究进展

*)中国科学院先进制造基地创新基金项目(F010120),973计划课题(2002CB312200)。魏英姿 博士生,讲师,主要研究方向是机器学习、智能制造。赵明扬 研究员,博士导师,主要研究方向是机器人学、智能制造。

比较缓慢。进入80年代以后，随着人工神经网络的研究不断地取得进展，以及计算机技术的进步，人们对强化学习的研究又出现了高潮，逐渐成为机器学习研究中的活跃领域。

已有很多工作研究算法收敛性、泛化性能、搜索能力和记忆功能[2~5]，取得了一定成绩。目前，在强化学习实际应用中，常遇到一些实际问题：在大区域内学习时，RLS 的学习进程会很慢；机器人正在学习复杂技能时，RL 会因分散注意而迷失了自己真正想达到的目标。学习系统根据动作的作用效果——强化信号学会怎样在环境中进行操作。为加速学习进程，要求强化信号应能及时准确地描述系统的学习进程。因此，如何设计好的回报函数已成为需认真研究的关键部分[6~10]。国际上从20世纪90年代以来开始有了这方面报道[8]，而国内尚未见到这方面研究。本文提出启发式的回报函数的设计思想，以期利用回报函数的及时反馈来提高 RLS 系统的学习性能。

2 相关理论

2.1 学习系统与回报函数的关系

在强化学习系统中，系统的学习目标被形式化为一个特定信号——回报信号，它从环境传向系统。回报信号为一个值 r，随时间变化而变化。强化学习系统都采用回报函数来定义回报，学习系统的顶层用它来指定学习任务，并且回报函数可以用来定义将要学习的值函数，学习系统通过与外界环境的交互作用改变回报函数的大小，学习系统并不直接对回报函数起作用，回报函数值往往由环境状态和学习系统自身内在的状态决定的，回报函数根据学习系统执行任务的好坏客观地进行评价，分配回报。

研究认为，强化学习执行任务与接受回报是一致的，系统的目标就是使回报的总和最大。系统期望得到的是长期回报综合的最大值，而不仅仅是瞬时回报的最大值。

2.2 Q 函数及搜索策略

强化学习算法中，应用最广泛的有 Q 学习和 SARSA 方法等，它用 $Q(s,a)$ 函数来表达在每个状态之下的每种动作的效果。s 是状态，a 是可能的动作。在状态 s 执行 a 动作，$Q(s,a)$ 根据下式更新：

$$Q(s_t,a_t)\rightarrow(1-\alpha)Q(s_t,a_t)+\alpha(r+\gamma\max_{a'\in A}Q(s',a')) \quad (1)$$

其中，α 为学习率，γ 为折扣系数。

SARSA 与 Q 学习类似，不同的是，SARSA 采用下一状态-动作对的实际动作值作更新，而不是采用最大动作值，更新规则为：

$$Q(s_t,a_t)\leftarrow(1-\alpha)Q(s_t,a_t)+\alpha(r+\gamma Q(s',a')) \quad (2)$$

由学习系统在所有遍历状态时的 Q 值构成学习系统的 Q 值表，Q 值表的状态/动作对即为下一次行动策略的依据。一般依据 Q 值 boltsman 分布(3)式或采用贪心法(4)式进行动作选择：

$$\pi_t(s,a)=\frac{e^{Q(s,a)/\tau}}{\sum_{a'\in A}e^{Q(s,a')/\tau}} \quad (3)$$

$$\pi_t(s,a)=\arg\max_{a'\in A}Q(s,a') \quad (4)$$

3 回报函数分析

在学习过程中，系统能接受到的有效回报的多少往往取决于系统学习算法的好坏，也跟回报函数定义的形式密切相关。

根据回报在状态-回报空间中的分布特点，本文把回报函数的构建分为两种形式：密集函数和稀疏函数。

密集函数 r 采用如下列形式：

$$r_t=f(o_t,i_t),\forall\ o,i \quad (5)$$

其中 i_t、o_t 分别为 t 时步学习系统内在的状态和外界环境状态。这种定义的回报函数通常为学习系统状态的连续函数。与密集函数对比，稀疏回报函数一般采用下列形式：

$$r_t=\begin{cases}1 & \text{在“好”状态}\\ -1 & \text{在“坏”状态}\\ 0 & \text{其他}\end{cases} \quad (6)$$

稀疏回报函数只在少许几个“好”“坏”状态，例如，完成任务和任务失败等，才接受非零回报值，因此，按这种方式构建的回报函数的回报值在状态空间中的分布是稀疏的。由于稀疏回报函数设计起来比较简单，目前使用的回报函数大多为稀疏函数。对于密集函数，在设计时必须清楚状态-动作空间的全部情况，所以，密集函数设计起来相对较难。

稀疏回报函数比密集回报函数更难学会值函数。常用的强化学习系统获取信息的主要途径是通过采取行动并观测其作用效果获取信息。在早期学习阶段，系统对环境一无所知，也不知道如何进行行动，如果学习算法又采用稀疏回报函数形式，由式(1)、(2)可以看出，这时的系统 Q 值表均为空值或预设值，系统就需要花费大量的时间去寻找非空、有意义的 Q 值，而按照 Q 值分布进行动作选择的学习系统，此时执行的动作或多或少都属于随机性的任意动作。如果只有几个回报值，并且状态空间很大，那么，找到非零回报状态-动作对的几率就会微乎其微。只有当更多的信息以反馈的形式进入学习系统时，RLS 才能通过学习与迭代，使值函数逼近性能迅速得到改善。

回报函数应能提供学习进步的信息，嵌入隐含知识，并把它表达为回报函数中的形式，为此本文在建立回报函数时，对传统的稀疏回报函数附加对中间状态的评价，通过评价系统的进步与退步给学习系统提供更多的启发式信息。

4 启发式回报函数的设计

4.1 广义的 MDP 模型

对于有限状态的序列规划任务，学习系统与环境的交互作用通常可以用马尔可夫决策过程(Markov Decision Process，MDP)建模。有限状态的 MDP 过程，可以表示为一个五元数组，$M=(S,A,T,\gamma,R)$，S 为环境状态集；$A=(a_1,a_2,\cdots,a_k)$ 为学习系统的动作集合(其中 $k\geqslant 2$)；$T=\{P(s'|s,a)|s\in S,\ a\in A\}$ 为状态转移可能性集合，$P(s'|s,a)$ 给出在状态 s 下采取行动 a 后，学习系统转化为 s' 状态的转化可能性。$r\in(0,1]$ 为折扣因子，R 则用来指定回报的分配情况。$R^a_{ss'}$ 即为回报函数(reward function)，定义为期望回报值：

$$R^a_{ss'}=E(r_{t+1}|s_t=s,a_t=a,s_{t+1}=s') \quad (7)$$

为简化，通常都假定回报是确定性的、有界实数函数。常写为如下形式：$R:S\times A\mapsto R$，意义是：在状态 S 下，执行动作 a 所接受的回报为 $R(s,a)$。本文引入一个更广义的回报函数形式：$R:S\times A\times S\mapsto R$，于是有回报函数形式为：$R(s,a,s')$ 表示在状态 s 执行动作 a 转换为状态 s' 时接受的回报。本文运行的强化学习算法是基于变换后的 MDP 模型，广义模型 $M'=(S,A,T,\gamma,R')$。$R'=R+F$ 是启发式回报函数形式，R 为传统的回报函数，F 是 MDP 以外提供的附加回报。

人们已经证明，对任意的 MDP 都存在一个静态的、确定性的最优策略。MDP 求解的目标是寻找一个最优策略 π^*，以使每个状态 $s\in S$ 的 $V^{\pi^*}(s)$ 值同时达到最大。在强化学习中，

用于确定 MDP 的 T 和 R 并不预先知道。一个强化学习系统必须通过与 MDP 环境直接交互来寻找一个最优策略。

4.2 启发式的回报函数

启发式回报函数设计的基本思想是，通过对学习系统赋予启发式的回报，来引导学习算法更快地学会最优策略。

$F(s,a,s')$ 是 MDP 以外附加的状态转移回报，F 用来给学习系统提供更多的启发式信息。规定 $F:S\times A\times S\mapsto R$ 为有界、实值回报函数。在原 MDP 模型 M 中，从 s 状态转换为 s' 状态接受回报为 $R(s,a,s')$，则在广义的 MDP 模型 M' 中，接受回报为 $R(s,a,s')+F(s,a,s')$。为鼓励学习进程趋向目标，只要 s' 比 s 更接近目标，状态转移回报函数 $F(s,a,s')=r$ 则取正回报值，否则 r 为0。

采用启发式回报函数进行学习的强化学习系统框架如图1所示。

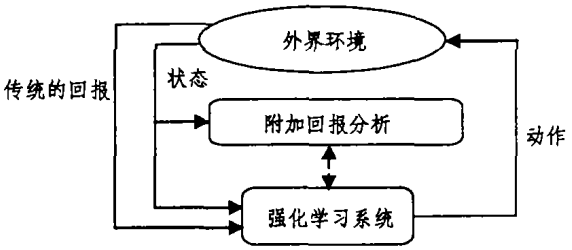

图1 学习系统框图

4.2.1 *嵌入先验知识* 为刺激某状态集 S_0 情况下执行动作 a_1，只要 $s\in S_0, a=a_1$，则取 $F(s,a,s')=r$ 为一非零值。这一特性对强化学习系统事先已知某些经验和知识时尤为重要，比如，存在有如下的产生式规则，

If $s\in S_0$, then $a=a_1, F(s,a,s')=r$

就可以通过附加额外回报的方式将先验知识和经验融入学习系统，这一做法对于机器人的学习控制问题、利用神经网络作函数逼近器、加入先验知识和经验等提供了一种新的途径。按照这种方式嵌入先验知识可以减少强化学习必须探索的有效状态-动作对数目，因此可以减少学习时间。

4.2.2 *值封闭性* 启发式回报函数应满足值封闭性：为保证系统经过 $(s_1\to s_2\to\cdots\to s_n\to s_1\to\cdots)$ 一系列状态循环之后，启发式回报函数不被人为地增加回报值，必须满足：

$$F(s_1,a_1,s_2)+\cdots+F(s_{n-1},a_{n-1},s_n)+F(s_n,a_n,s_1)\leqslant 0 \quad (8)$$

所以，公式 $R+F$ 中的 $F(s,a,s')$ 选为关于状态 s 的保守势函数 $\Phi(s)$ 的差分形式：

$$F(s,a,s')=\gamma\Phi(s')-\Phi(s) \quad (9)$$

其中，$\Phi:S\mapsto R, \forall s, |\Phi(s)|<c, c$ 为一个正的常数，把 Φ 限定为有界函数，因此，可以保证 $F(s,a,s')$ 和 $R'(s,a,s')$ 均为有界的实值函数。保守势函数 Φ 可以依据局域专家知识对状态的评价而确定，例如，$\Phi(s)$ 可以选为状态 s 与目标（或者子目标）状态之间的广义距离（如，人工势场力矢量的模等）的相反数。即，如果后一状态 s' 比前状态 s 更接近目标，则(9)式取非负值，$F(s,a;s')$ 就附加给 R' 一个非负值，如果远离目标，就施加一个非正的回报值。另外，如果系统有充分可用的局域专家知识，状态值函数 $V(s)$ 可确切表达出来，那么 $\Phi(s)$ 也可以选为 $V(s)$ 的线性函数。

传统的稀疏回报函数是状态空间内分布的离散点，附加回报使回报函数变换为可以提供梯度信息的连续函数，学习系统可以利用基于状态-回报的局部梯度信息选择搜索方向，迅速搜索趋向目标的状态，采用稀疏回报与附加回报相结合的启发式回报函数的仿真试验，可以验证启发式回报函数能够提高算法的搜索效率[11]。

4.2.3 *有约束条件问题的附加回报函数* 如果学习系统任务为有约束问题（比如，迷宫行走的避碰就是有约束问题），本文采用罚函数法将有约束问题转化为无约束问题。

罚函数法的基本思路是把约束问题中的目标函数进行修改，使问题中原有的约束条件集中反映到新的目标函数中去，并且要求这个转换过程不破坏原有的约束条件，再通过求解一系列的无约束极值问题来逐步逼近原问题的最优点。一般采用下面形式：

$$F\leftarrow F+f_{pen} \quad (10)$$

f_{pen} 即为惩罚项。对有约束的回报函数问题，惩罚函数应具有这样的性质：对非可行点产生一个负的惩罚，但对可行点则不进行惩罚。

4.2.4 *动态的评价准则* 由于学习系统在每步进行动作选择的过程中，（由式(3)、(4)可看出）都是依据当前 t 学习步计算状态-动作值表的评价进行动作选择，因此，在不同学习时间步，对于状态-动作值函数的评价可以选择不同的评价标准。本文采用动态的评价标准。

早期学习阶段，则不实施惩罚，若出现 $\gamma\Phi(s')-\Phi(s)<0$，取 $\gamma\Phi(s')-\Phi(s)=0$，而在后期则对学习系统的好坏实行奖惩分明的评价标准。这是由于在开始学习阶段应给予学习系统更多的鼓励，而在学习进行到后期则无论进步与退步，信息都应及时确切地反馈给学习系统。这种动态的评价准则符合循序渐进的学习规律。

5 最优策略不变性证明

有一固定动作集合 A，针对状态集 S 的策略为函数 $\pi:S\mapsto A$，可以注意到，策略是针对状态而定义的而不是面向 MDP 的，所以，只要两个 MDP 决策过程使用相同状态、动作集，同样的策略可以应用到两个不同的 MDP 中。

证明：如果 F 是基于状态的势差分形式的附加函数，那么，证明 M' 模型中的最优策略也是 M 模型中的最优策略。

对原模型 M，最优 Q 值函数 Q_M^* 满足文[3]中 Bellman 方程，

$$Q_M^*(s,a)=E[R(s,a,s')+\gamma\max_{a'\in A}Q_M^*(s',a')] \quad (11)$$

两边同时减去 $\Phi(s)$，并将右边作加减 $\gamma\Phi(s')$ 的恒等变换，

$$Q_M^*(s,a)-\Phi(s)=E[R(s,a,s')+\gamma\Phi(s')-\Phi(s)+\gamma\max_{a'\in A}(Q_M^*(s',a')-\Phi(s'))] \quad (12)$$

定义 $$\hat{Q}_{M'}(s,a)\triangleq Q_M^*(s,a)-\Phi(s) \quad (13)$$

把式(9)、(13)代入(12)式，有，

$$\hat{Q}_{M'}(s,a)=E[R(s,a,s')+F(s,a,s')+\gamma\max_{a'\in A}\hat{Q}_{M'}(s',a')]$$

$$\hat{Q}_{M'}(s,a)=E[R'(s,a,s')+\gamma\max_{a'\in A}\hat{Q}_{M'}(s',a')] \quad (14)$$

式(14)恰好是 M' 模型的 Bellman 方程。因此，

$$Q_{M'}^*(s,a)=\hat{Q}_{M'}(s,a)=Q_M^*(s,a)-\Phi(s) \quad (15)$$

M' 模型的最优策略必须满足下式，

$$\pi_{M'}^*(s)=\arg\max_{a\in A}Q_{M'}^*(s,a) \quad (16)$$

将(15)式代入，

$$=\arg\max_{a\in A}[Q_M^*(s,a)-\Phi(s)] \quad (17)$$

$$=\arg\max_{a\in A}Q_M^*(s,a) \quad (18)$$

由此可见，M' 的最优策略也是 M 模型的最优策略，因此，启发式回报函数形式并不改变原问题的最优策略，M 和 M' 的最优策略是一致的。

若 $\gamma=1$，则$\forall\ s\in S;\forall\ a\in A$

$$Q_{M'}^{*}(s,a)=Q_{M}^{*}(s,a)-\Phi(s) \tag{19}$$

由 $V^{*}(s)=\max_{a\in A}Q^{*}(s,a)$，有

$$V_{M'}^{*}(s)=V_{M}^{*}(s)-\Phi(s) \tag{20}$$

6 Q 值迭代收敛性证明

证明：对原马尔可夫决策过程模型 M，假设存在有全备份的 Q 值迭代算子 $B(Q)$，并假设原 M 模型的迭代算子 $B(Q)$为收敛算子，证明对广义的 M' 模型，Q 值迭代也是收敛的。

对于原模型 M，如果 $Q_{t+1}=B(Q_t)$成立，则，

$$\max_{s,a}|Q_{t+1}^{M}(s,a)-Q_{M}^{*}(s,a)|\leqslant\alpha\max_{s,a}|Q_{t}^{M}(s,a)-Q_{M}^{*}(s,a)|\quad(0<\alpha<1)$$

令 $L=\max_{s,a}|Q_{t}^{M}(s,a)-Q_{M}^{*}(s,a)|$，

则，$\forall\ (s,a)$，$Q_{M}^{*}(s,a)-L\leqslant Q_{t}^{M}(s,a)\leqslant Q_{M}^{*}(s,a)+L$

对于 M，$\forall\ (s,a)$，

$$Q_{t+1}^{M}(s,a)=R(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)V_{t}^{M}(s')$$

上式进行数学变换，

$$Q_{t+1}^{M}(s,a)-\Phi(s)=R(s,a,s')+\gamma\Phi(s')-\Phi(s)+\gamma\sum_{s'\in S}P(s'|s,a)V_{t}^{M}(s')-\gamma\Phi(s') \tag{21}$$

$$\leqslant R'(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)[V_{M}^{*}(s')+L]-\gamma\Phi(s') \tag{21-a}$$

$$\leqslant R'(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)[V_{M}^{*}(s')-\Phi(s')]+\gamma L \tag{21-b}$$

式(21)化为：

$$Q_{t+1}^{M}-\Phi(s)=R(s,a,s')+F(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)[V_{t}^{M}(s')-\Phi(s')]$$

$$Q_{t+1}^{M}-\Phi(s)=R'(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)[V_{t}^{M}(s')-\Phi(s')] \tag{22}$$

定义 $\hat{Q}_{t+1}^{M}(s,a)\triangleq Q_{t+1}^{M}(s,a)-\Phi(s)$ (23)

$$\hat{V}_{t}^{M}(s')\triangleq V_{t}^{M}(s')-\Phi(s') \tag{24}$$

(23)(24)式代入(22)式，有

$$\hat{Q}_{t+1}^{M}(s,a)=R'(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)\hat{V}_{t}^{M}(s') \tag{25}$$

(25)式恰好符合基于 M' 模型的 Bellman 方程的 Q 值更新规则，于是，

$$\hat{Q}_{t+1}^{M}(s,a)=Q_{t+1}^{M'}(s,a)=Q_{t+1}^{M}(s,a)-\Phi(s) \tag{26}$$

$$\hat{V}_{t}^{M}(s')=V_{t}^{M'}(s')=V_{t}^{M}(s')-\Phi(s') \tag{27}$$

将(20)(26)式代入(21-b)中，则，

$$Q_{t+1}^{M'}(s,a)\leqslant R'(s,a,s')+\gamma\sum_{s'\in S}P(s'|s,a)V_{M'}^{*}(s')+\gamma L$$

所以，$Q_{t+1}^{M'}(s,a)\leqslant Q_{M'}^{*}(s,a)+\gamma L$ 成立。

同理，可证明，$Q_{t+1}^{M'}(s,a)\geqslant Q_{M'}^{*}(s,a)-\gamma L$ 成立，因此，采用启发式回报函数的 Q 函数在广义 M' 模型中仍然满足 Bellman 方程 Q 值更新规则，并且保证了 Q 值迭代也是收敛的。

结论与展望 回报函数定义的好坏，对提高学习系统处理问题能力有着至关重要的作用。基于广义的 MDP 模型，本文提出能嵌入局部进步信息的启发式回报函数，采用基于保守势函数差分形式的附加回报，保证启发式回报函数的值封闭性。采用动态评价标准，形成循序渐进的学习方式，在广义 MDP 模型下，采用启发式回报函数并没有改变针对原状态集、动作集的最优策略，并证明了强化学习算法的迭代收敛性。

依据状态转移信息定义的附加回报，使回报函数成为可以提供局部梯度信息的连续函数，对于引导学习系统的搜索方向、加速学习进程、提高 RLS 的学习效率有重要作用，为强化学习进一步应用到实际复杂学习系统进行实时控制和调度拓宽了天地。

参 考 文 献

1 Minsky M L. Theory of Neural Analog Reinforcement Systems and its Application to the Brain Model Problem. New Jersey, USA, Princeton University, 1954

2 Bertsekas D P, Tsitsiklis J N. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA, 1996

3 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998

4 Boyan J A, Moore A W. Generalization in reinforcement learning: Safely approximating the value function. Advances in Neural Information Processing Systems, 7: 369～376

5 Watkins C J C H. Learning from Delayed Rewards: [Ph. D. Thesis]. King's College, University Library Cambridge, 1989

6 蒋国飞，等. 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

7 Sutton R S. Open Theoretical Questions in Reinforcement Learning. In: Proc. of the Fourth European Conf. on Computational Learning Theory. Fischer P, Simon H U, eds. Springer-Verlag, 1999. 11～17

8 Mataric M J. Reward Functions for Accelerated Learning. In: Machine Learning: Proc. of the Eleventh Intl. Conf. Morgan Kaufmann Publishers, San Francisco, CA, 1994. 181～189

9 Ng Y, Harada D, Russell S. Policy invariance under reward transformations: Theory and application to reward shaping. In: Proc. of the Sixteenth Intl. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1999. 278～287

10 Bonarini A, Bonacina C, Matteucci M. An Approach to the Design of Reinforcement Functions in Real World, Agent-Based Applications. Man and Cybernetics, Part B, IEEE Transactions on, 2001, 31(3): 288～301

11 WEI Ying-zi, ZHAO Ming-yang. Effective Strategies for Complex Skill Real-time Learning Using Reinforcement Learning. In: IEEE Intl. Conf. on Robotics, Intelligent Systems and Signal Processing, 2003. 388～392