第 25 卷第 1 期
2008 年 2 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 25 No. 1
Feb. 2008

文章编号: 1000−8152(2007)05−0071−04

动态环境中的分层强化学习

沈　晶，程晓北，刘海波，顾国昌，张国印

(哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

摘要: 现有的强化学习方法都不能很好地处理动态环境中的学习问题, 当环境变化时需要重新学习最优策略, 若环境变化的时间间隔小于策略收敛时间, 学习算法则不能收敛. 本文在Option分层强化学习方法的基础上提出一种适应动态环境的分层强化学习方法, 该方法利用学习的分层特性, 仅关注分层任务子目标状态及当前Option内部环境状态的变化, 将策略更新过程限制在规模较小的局部空间或维数较低的高层空间上, 从而加快学习速度.以二维动态栅格空间内两点间最短路径规划为背景进行了仿真实验, 实验结果表明, 该方法策略学习速度明显高于以往的方法, 且学习算法收敛性对环境变化频率的依赖性有所降低.

关键词: 分层强化学习; 动态环境; Option; 策略更新

中图分类号: TP18　　**文献标识码**: A

Hierarchical reinforcement learning in dynamic environment

SHEN Jing,　CHENG Xiao-bei,　LIU Hai-bo,　GU Guo-chang,　ZHANG Guo-yin

(School of Computer Science and Technology, Harbin Engineering University, Harbin Heilongjiang 150001, China)

Abstract: The existing reinforcement learning approaches cannot satisfactorily solve the learning problems in dynamic environment. The optimal strategy must be re-learned when environment changes. The learning algorithm cannot converge to optimal strategy if the interval between the changes is shorter than the duration of strategy converging. In this paper, a hierarchical reinforcement learning approach adapting to dynamic environments is presented based on the Option hierarchical reinforcement learning. According to the hierarchical characteristic of learning, the approach only takes into account the changes taking place in the sub-goal states of hierarchical tasks or the environment states of current Option. So the process of strategy update is limited in a small-scale local space or a low dimension high-level space. Consequently, the process of strategy update is accelerated. The experiments with shortest path planning in a two-dimensional dynamic grid space show that the presented approach is obviously faster than the existing approach in strategy update. Additionally the dependency of convergence of the learning algorithm on the frequency of environment change is reduced.

Key words: hierarchical reinforcement learning; dynamic environment; Option; strategy update

1　引言(Introduction)

强化学习(RL: reinforcement learning)[1]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. 但RL要求环境具有Markov特性, 否则学习算法难以收敛. 目前的相关研究工作[2]所考虑的动态环境主要是Markov意义下的动态, 这在实际应用中很难得到满足, 环境往往会受其它因素诱导而动态变化, 从而不具有Markov特性, 当环境状态发生频繁变化时, 学习系统需要不断重新学习最优策略, 效率极低. 特别地, 当环境变化的时间间隔小于策略的收敛时间时, 则学习算法根本无法收敛.

分层强化学习[3](HRL: hierarchical RL)是为克服RL的维数灾问题而提出的, 目前代表性的成果主要有Option[4]、HAM[5]和MAXQ[6]等方法, 这些方法同样不能直接处理这类动态环境中的学习问题. 但是, 强化学习任务经过分层之后, 学习任务或局限于Agent当前所处的规模较小的局部空间, 或局限于与底层细节无关的维数较低的高层空间, 这样, 在分层强化学习中, 如果Agent只关注当前局部空间内的环境变化和分层任务子目标状态的变化, 将策略更新过程限制在局部空间或高层空间上, 则可以加快学习速度, 使学习算法收敛性对环境变化频率的依赖性能够有所降低.本文从这一思想出发, 以Option(不难推广到HAM和MAXQ)方法为基础提出一种适应更广泛动态环境的HRL方法.

收稿日期: 2005−12−16; 收修改稿日期: 2007−01−05.
基金项目: 中国博士后基金资助项目(20060400809); 哈尔滨工程大学基础研究基金资助项目(HEUFT07022, HEUFT05068, HEUFT05021).

2 HRL原理(Principles of HRL)

RL是Agent从环境状态到动作映射的学习, 其目标是要获得一个最优行为策略$\pi^*: S \to A$, 使Agent选择的动作能够得到环境的最大奖赏, 其中: S为状态集, A为动作集. RL问题常采用MDP(Markov decision process)建模. RL的维数灾问题引发了对HRL方法的研究. HRL的核心思想是对学习任务进行分解, 形成可以完成子目标的若干个抽象(Abstraction), 抽象是由基本动作组成的动作序列, 每个抽象具有执行这些基本动作的内部策略. Agent无需在每个时间步都对动作做出决策, 而是在每个抽象按各自的内部策略执行完成之后才进行下一次决策, 因此, HRL问题常采用SMDP(Semi-MDP)建模. 不同的抽象形式产生了Option、HAM和MAXQ等不同的HRL方法, 作为本文的基础, 下面重点介绍Option方法.

Option方法中, 学习任务被抽象成若干Option, 每个Option可以理解为为完成某子目标而定义在某状态子空间上的按一定策略执行的动作或Option序列. 最简单的形式是定义在MDP上的Markov-Option, 用三元组$< \varphi, \pi, \beta >$表示, 其中, $\varphi \subseteq S$, 为入口状态集, 当且仅当$s \in \varphi$时, 该Option可依策略执行, 通常, φ包含且只包含该Option经历的所有可能状态, $\pi: \varphi \times A_\varphi \to [0,1]$为内部策略, A_φ为在状态集φ上可执行的动作集, $\beta: S \to [0,1]$为终止条件, Option在某一状态s'依概率$\beta(s')$终止, 通常, 将要达到的子目标状态s_G定义为$\beta(s_G) = 1$. 如果将策略定义在Option之上, 即$\mu: \varphi \times O_\varphi \to [0,1]$, O_φ为状态集φ上的可执行的Option集, φ和β定义不变, 则Option$< \varphi, \mu, \beta >$即为Semi-Markov-Option, 将其叠加在核心MDP之上, 便形成了SMDP. 在该方法中每个Option执行结束时进行一次学习, Q-学习更新公式如下:

$$Q_{k+1}(s,o) = (1-\alpha_k)Q_k(s,o) + \alpha_k[r + \gamma^\tau \max_{o' \in O_{s'}} Q_k(s',o')]. \quad (1)$$

其中: k为迭代次数, α_k为学习率, γ为折扣率, r是Agent从环境状态s经过τ个时间步完成o后到达s'所接受的奖赏值. Precup证明了在标准Q-学习收敛的条件下, Option方法以概率1 收敛到最优策略[7].

Option可以根据专家知识事先确定, 也可以自动生成. 目前解决自动分层问题的研究工作多集中在状态空间的子目标发现上, 根据子目标即可对状态和动作进行抽象以形成分层子任务[8~10].

3 动态环境中的HRL(HRL in dynamic environment)

任务分层的思想使得学习空间中的状态有了子目标状态和非子目标状态的区分, 环境的动态变化作用于不同的状态上对学习算法的影响亦有不同, 充分利用分层特性区别对待环境变化, 即可使得强化学习方法在动态环境中也能产生满意的学习效果. 本文因此构建动态环境中HRL框架, 其学习过程(简称DQ算法)如下:

Begin DQ()
1) 初始化空白状态转移结构图和随机Q表
2) Repeat
3) $s \leftarrow s_0$
4) Repeat
5) 观察当前状态s
6) 选择并执行一个动作a
7) 观察下一个状态s', 获得一个奖赏值r
8) 按式(1)调整Q值
9) 修改状态结构图
10) 记录学习经验(s, a, s', r)
11) Until(s'为目标状态)
12) Until(连续T个探测周期内没有探索到新状态)
13) 计算子目标, 生成Option入口状态集
14) 各Option内部策略学习
15) Repeat/*Semi-Markov-Option策略学习*/
16) 观察当前状态s
17) 选择一个Option o
18) 调用Execute-Option (o, s)
19) 按式(1)更新Q值
20) Until(满足学习结束条件)
End DQ

其中, 状态转移结构图定义为三元组$< V, U, E >$, 其中V为探明状态, 即第5)步中观察过的状态, U为探明的不可达状态或未探明状态, E为状态间的可达关系, 用0–1矩阵表示, 初值为全0, 状态转移结构图用于计算子目标和生成Option. 步骤9)根据探测结果修改E的元素值, 若第i点到第j点可达, 则置$E_{ij} = 1$. 步骤8)按式(1)调整Q值, 普通动作可以视为Option的一种特例, 则式(1)中取$\tau = 1$并用a替换o即成适用于步骤8)的标准Q-学习更新公式. 步骤10)记录的学习经验可在步骤14)(采用经验回放[11])学习Option内部策略时使用. 步骤3)到11)的一次循环称为一个探测周期, 参数T用于控制探测程度, 可根据经验确定, 一般设$T = 2$即可满足要求. 步骤13)中计算子目标可以采用强化梯度[8]、状态频率[9]、Q-Cut[10]等方法计算. 算法学习结束条件有多种设置方法, 如达到预定学习周期或误差要求等.

步骤18)中调用的Execute-Option是一个在动态

环境中执行Option的递归算法:

Begin Execute-Option(Option o, 入口状态s)

1) $s' \leftarrow s$

2) Repeat

3)　观察当前状态s_c

4)　If $(s_c \neq s')$ Then

5)　　If (子目标状态发生变化) Then

6)　　重新学习Semi-Markov-Option策略

7)　　Else

8)　　If (o的内部状态发生变化) Then

9)　　重新学习o的内部策略

10)　　End If

11)　End If

12)　Execute-Option(o, s)

13)　End If

14)　按o的内部策略选择并执行一个动作a

15)　观察下一个状态s'

16) Until (s'为o的终止状态)

17) 按式(1)调整Q值

End Execute-Option

算法的意图是明显的: 状态外因诱导下发生改变时, 仅重新学习当前Option或顶层Option, 而不再重新学习整个MDP.

4　仿真实验与分析(Simulation and analysis)

仿真以Agent在二维有移动障碍物的栅格空间中学习规划给定的起终点间的最短路径为任务背景(图1), 对强化学习框架下Q-学习(RQ)、HRL框架下的Q-学习[4](HQ)和本文的动态环境中HRL框架下的Q-学习(DQ) 3种算法的性能进行实验比较.

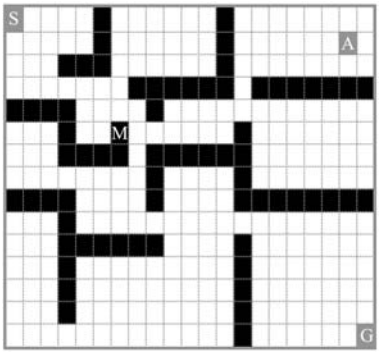

图 1　实验环境

Fig. 1　Experiment environment

图1中, 黑色栅格为障碍空间, 白色栅格为无障碍空间, 标有字母的4类栅格: S, G分别为起终点, A为学习Agent, M为移动障碍物, 移动障碍物在每个时间步内可随机地向上、下、左、右任一方向移动一个栅格. Agent在学习前对环境的静态信息(即静态障碍物分布情况)完全未知, Agent可以执行上、下、左、右4个基本动作和符合入口条件的Option, 以0.2的概率对环境进行探测(贪婪策略), 与障碍物碰撞时得到值为-20的惩罚信号, Agent到达目标点即完成一个学习周期, 并获得值为100的奖励信号, 然后重新回到起点开始下一个周期的学习, 设$T = 2$, 即连续两个周期中未发现新状态, 即开始计算子目标并生成Option, 计算子目标采用Q-Cut算法[10], Q-Cut算法在状态转移结构图的基础上, 利用最大流-最小割原理计算瓶颈状态作为子目标. 各Q-学习算法中, 学习率均设为固定的0.1, 折扣因子均设为0.9.

图1中起终点间最短路径为34步, 图2中以学习获得路径长度向最短路径收敛的趋势对比了3种算法的性能, 3种算法学习获得路径的平均长度依次为1313, 338和44步(10次实验的平均值).

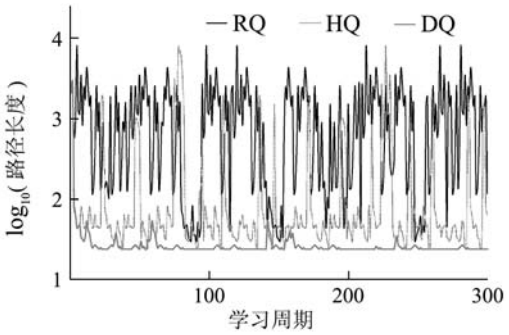

图 2　算法性能对比

Fig. 2　Performance comparison of algorithms

由图2可见, 3种算法都不能严格收敛, 但收敛趋势有所不同. RQ是完全发散的, 观察仿真程序执行过程可以发现, 障碍物的随机移动导致更新过的Q值不断失效, Q表无从真实地反映最优策略, 学习算法自然体现不出任何收敛趋势. 此外, 学习率设置为固定值对RQ的学习结果也有一定影响, 本文按$0.9 - 0.1$可变学习率设置做了实验, RQ获得的平均路径长度缩减到800步左右, 但是其发散特性并没有太大改善. HQ同样不收敛, 但是其习得的路径平均长度较小, 这是由于HQ在对环境进行2个学习周期探测之后即利用状态转移结构图离线生成Option, 该过程不受环境动态性干扰. 所以, 每次重新生成Option分层之后均能收敛到最短路径, 但是, 障碍物的移动会导致刚刚生成的Option分层失效, 于是频繁地重新分层, 尽管算法微观上有收敛趋势, 但是宏观上依然是发散的. DQ尽管也不收敛, 但是平均路径长度已接近最短路径, 这是因为DQ没有盲目地对环境的所有动态变化都做出响应, 只是关注影响子目标的状态和当前Option的内部状态而忽略了其它状态的变化, 并根据不同类型状态的变

化做出不同的反应, 将策略更新过程限制在局部空间(当前Option的内部状态时)或高层空间(子目标状态变化时)上, 从而加快更新速度, 在受到移动障碍物干扰后能够快速收敛到最短路径, 从而体现出一种“几乎处处”收敛的趋势.

改变环境动态变化频率, 保持其他参数不变, 观察3种算法性能所受的影响, 表1给出了3种变化频率下各算法习得最短路径平均值的统计结果.

表 1 算法对环境变化频率的依赖性

Table 1 The dependency of algorithms on the frequency of environment change

变化频率	RQ	HQ	DQ
1/100	52	45	42
1/10	845	255	42
1/1	1313	338	44

由表1可见, 环境变化频率较低(1/100)时, 3种算法学习效果均比较理想, 学习获得的最短路径没有特别明显区别, 都接近最短路径, 但路径长度依然体现出RQ>HQ>DQ的规律性, 而当环境变化频率升高(1/10和1/1)时, RQ和HQ性能急剧下降, 而DQ则所受影响极小. 这都是因为DQ忽略了与当前子任务无关的大量环境状态变化, 才有效降低了算法收敛性对环境变化频率的依赖性. 尽管在环境低频变化时RQ和HQ均有机会完成两个周期的学习, 但还不足以收敛, 包括DQ也是不收敛的, 就是因为它们总要或多或少地对环境的变化做出新的响应.

5 结论(Conclusions)

为了解决RL不能广泛适用于动态环境的问题, 本文以Option分层强化学习方法为基础提出一种适应更广泛动态环境的HRL方法, 该方法利用学习的分层特性, 仅关注分层任务子目标状态及当前Option内部环境状态的变化, 将策略更新过程限制在规模较小的局部空间或维数较低高层空间上, 从而加快更新速度. 以二维动态栅格空间内两点间最短路径规划为背景进行了仿真实验和对比分析, 结果表明, 算法在动态环境中能达到“几乎处处”收敛的效果, 且算法收敛性对环境变化频率的依赖性很低.

参考文献(References):

[1] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 30(1): 86 – 100.
(GAO Yang, CHEN Shifu, LU Xin. A survey on reinforcement learning[J]. *Acta Automatica Simica*, 2004, 30(1): 86 – 100.)

[2] EXCELENTE-TOLEDO C B, JENNINGS N R. Using reinforcement learning to coordinate better[J]. *Computational Intelligence*, 2005, 21(3): 217 – 245

[3] BARTO A G, MAHADEVAN S. Recent advances in hierarchical reinforcement learning[J]. *Discrete Event Dynamic Systems: Theory and Applications*, 2003, 13(4): 41 – 77.

[4] SUTTON R S, PRECUP D, SINGH S P. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning[J]. *Artificial Intelligence*, 1999, 112(1): 181 – 211.

[5] PARR R. *Hierarchical control and learning for markov decision processes*[D]. Berkeley: University of California, 1998.

[6] DIETTERICH T G. Hierarchical reinforcement learning with the MAXQ value function decomposition[J]. *J of Artificial Intelligence Research*, 2000, 13(1): 227 – 303.

[7] PRECUP D. *Temporal abstraction in reinforcement learning*[D]. Amherst: University of Massachusetts, 2000.

[8] DIGNEY B L. Learning hierarchical control structures for multiple tasks and changing environments[C] // *From Animals to Animats 5: Proc of the Fifth Int Conference on Simulation of Adaptive Behavior*. Cambridge: MIT Press, 1998: 321 – 330.

[9] MCGOVERN A, BARTO A. Autonomous discovery of subgoals in reinforcement learning using diverse density[C] // *Proceedings of the 8th Int Conf on Machine Learning*. San Fransisco: Morgan Kaufmann, 2001: 361 – 368.

[10] MENACHE I, MANNOR S, SHIMKIN N. Q-cut: dynamic discovery of sub-goals in reinforcement learning[C] // *Proc of the 13th European Conf on Machine Learning*. New York: ACM Press, 2002: 295 – 306.

[11] LIN L G. Self-improving reactive agents based on reinforcement learning, planning and teaching [J]. *Machine Learning*, 1992, 8(3): 293 – 321.)

作者简介:

沈 晶 (1969—), 女, 哈尔滨工程大学副教授, 博士, 主要研究方向为分层强化学习、人工免疫系统, E-mail: shenjing@hrbeu.edu.cn;

程晓北 (1962—), 男, 哈尔滨工程大学博士研究生, 主要研究方向为多智能体分层强化学习;

刘海波 (1976—), 男, 哈尔滨工程大学副教授, 博士, IEEE计算机学会专业会员, 主要研究方向为智能机器人体系结构、多智能体系统, E-mail: liuhaibo@hrbeu.edu.cn;

顾国昌 (1946—), 男, 哈尔滨工程大学教授, 博士生导师, 主要研究方向为嵌入式系统、智能机器人技术;

张国印 (1962—), 男, 哈尔滨工程大学教授, 博士生导师, 主要研究方向为嵌入式系统、智能机器人技术.