基于多智能体的 Option 自动生成算法

沈 晶, 顾国昌, 刘海波
(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:目前分层强化学习中的任务自动分层都是采用基于单智能体的串行学习算法,为解决串行算法学习速度较慢的问题,以 Sutton 的 Option 分层强化学习方法为基础框架,提出了一种基于多智能体的 Option 自动生成算法,该算法由多智能体合作对状态空间进行并行探测并集中应用 aiNet 实现免疫聚类产生状态子空间,然后并行学习生成各子空间上的内部策略,最终生成 Option. 以二维有障碍栅格空间内 2 点间最短路径规划为任务背景给出了算法并进行了仿真实验和分析. 结果表明,基于多智能体的 Option 自动生成算法速度明显快于基于单智能体的算法.

关键词:分层强化学习;自动分层;多智能体系统;Option;aiNet
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-4785(2006)01-0084-04

Algorithm for automatic constructing Option based on multi-agent

SHEN Jing, GU Guo-chang, LIU Hai-bo
(School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: In current hierarchical reinforcement learning, the automatic task hierarchies are constructed by low speed serial learning algorithm based on single-agent. A multi-agent based algorithm for constructing Options automatically was presented for speeding up the learning algorithm. The algorithm was developed on the basis of the Option HRL framework proposed by Sutton. Firstly, multiple agents cooperated in parallel exploring the state space. Then the state space was partitioned into several sub-spaces via immune clustering based on aiNet. Next, the agents learned the local strategies of the different sub-space concurrently. Consequently, the Options were constructed. The theoretical analyses and experiments with shortest path planning in a two-dimensional grid space with obstacles show that the speed of multi-agent based algorithm for automatically constructing Options was obviously faster than that of single-agent based algorithms.

Keywords: hierarchical reinforcement learning; automatic hierarchy; multi-agent system; Option; aiNet

分层强化学习(HRL)是克服强化学习(RL)维数灾难的有效方法,其代表性的研究成果主要有 Option[2], HAM[3] 和 MAXQ[4] 方法. 其中的层次结构可以由设计者根据专家知识事先确定,也可以自动生成. 由于在复杂环境或者未知环境内学习时,任务层次结构很难事先确定,因此自动分层方法一直是最近几年的研究热点. 现有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. 典型的研究成果有:Digney[5] 将强化信号梯度高的状态作为子目标,McGovern 和 Barto[6] 根据状态出现的频率选择子目标,Menache 等[7] 通过最大流—最小割方法确定状态转移图中的瓶颈状态,并将其定义为子目标状态,而 Mannor[8] 则是通过状态聚类将学习分解成若干阶段. 上述分层方法中,自动分层过程均串行完成. 为加快自动分层速度,该文提出了一种基于多智能体的 Option 自动分层算法,速度上的优越性在仿真实验中得到了验证.

收稿日期:2005-12-28.
基金项目:哈尔滨工程大学基础研究基金资助项目(HEUFT05021, HEUFT05068).

1 Option 分层强化学习基本原理

Option 分层强化学习框架中,学习任务被抽象成若干 Option,并将这些 Option 作为一种特殊的"动作"加入到原来的动作集中. 一个 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,普通动作可以视为 Option 的一种特例. 设 S 和 A 分别为智能体的状态集和动作集,最简单的一种 Option 是直接定义在马氏过程(MDP)上的,用 3 元组 $<\phi,\pi,\beta>$ 表示,其中,$\phi\subseteq S$,为入口状态集,当且仅当 $s\in\phi$ 时,Option $<\phi,\pi,\beta>$ 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,$\pi:\phi\times A_\phi\rightarrow[0,1]$ 为 Option 内部策略,A_ϕ 为在状态集 ϕ 上可执行的动作集;$\beta:S\rightarrow[0,1]$ 为 Option 终止条件,Option 在某一状态 s 依概率 $\beta(s)$ 终止. 通常,将 Option 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 $\mu:\phi\times O_\phi\rightarrow[0,1]$,$O_\phi$ 为状态集 ϕ 上的可执行的 Option 集,ϕ 和 β 定义不变,则 $<\phi,\mu,\beta>$ 即形成分层 Option. Option $<\phi,\pi,\beta>$ 称为 Markov-Option,Option $<\phi,\mu,\beta>$ 称为 Semi-Markov-Option. 将 Semi-Markov-Option 叠加在核心 MDP 上,便形成了 Semi-MDP(SMDP).

需要说明的是,在多智能体参与学习的情况下,状态转移不能由某一智能体的单个动作决定,原有的 MDP 及 SMDP 模型不能简单适用于多智能体学习. 但该文的算法仅涉及状态空间探测和 Option 内部策略学习等局部求解问题,而不涉及多智能体同时学习问题,因而无需考虑决策模型到多智能体领域的拓展问题.

2　基于多智能体的 Option 自动生成算法

以下算法假定以多智能体在二维有障碍环境空间内合作学习规划任意起点到给定终点间最短路径为任务背景,环境空间用矩形栅格地图表示. 设地图最大边长为 E_{max},每个栅格用“0”标记为空白,用“1”标记为障碍,每个智能体能够执行上、下、左、右 4 个基本动作及符合条件的 Option,从而在环境空间中移动,智能体能够探测并避免与环境空间中的障碍相撞. 定义状态结构图为表示智能体对环境空间视图的黑板结构,也采用最大边长为 E_{max} 的矩形栅格表示,栅格状态除上述“0”、“1”,增加“2”表示未探测.

文中的自动分层方法借鉴了 Mannor[8] 的思想,即不去发现子目标,而是通过状态聚类将学习分解成若干阶段,每个阶段构成一个 Option,每个 Option 学习一组内部策略,这组策略确定了从 Option 内部任意一状态到达任意出口状态的最短路径.

2.1　基于多智能体的状态空间探测算法

1) 系统对状态结构图进行初始化,将所有栅格的状态标记为“2”;

2) 系统对 X 个(可根据问题规模确定)多智能体进行初始化,随机生成各智能体在地图中的初始位置,每个智能体的探测计数器 C_e 清零;

3) 多智能体并行执行:

①随机选择一个基本动作;②计算动作后的位置(下一位置);③观测下一位置的状态(下一状态);④如果下一状态为“0”,则将状态结构图中对应位置状态置“0”,智能体移到下一位置;⑤如果下一状态为“1”,则将状态结构图中对应位置状态置“1”,智能体不移动;⑥$C_e=C_e+1$;⑦如果 $C_e\leq E_{max}^2$ 则转①,否则,停止探测;

4) 待智能体都停止探测,启动状态子空间聚类算法.

2.2　状态子空间聚类算法

状态子空间聚类算法基于 Castro 和 Zuben 提出人工免疫网络 aiNet[9] 构建,用到如下主要参数:Ag:抗原集合,将 Agent 探测过的每一状态定义为一个抗原,抗原 i 的表示形式为

$$\mathrm{Ag}_i=(x_{i_1},x_{i_2},\cdots,x_{i_n}).$$

式中:$(x_{i_1},x_{i_2},\cdots,x_{i_n})$ 为与抗原对应的 n 维状态的空间坐标,该文讨论的算法中 n 取 2,$\mathrm{Ag}=\{\mathrm{Ag}_i\mid i=1,\cdots,m\}$,$m$ 为正整数;Ab:抗体集合,表示形式同抗原,$\mathrm{Ab}=\{\mathrm{Ab}_i\mid \mathrm{Ab}_i=(x_{i_1},x_{i_2},\cdots,x_{i_n}),i=1,\cdots,N\}$,$N$ 为正整数,随机初始化后,在算法中自动调节,算法停止后,抗体数即为最终聚类数;Ab_{sub} 为从 Ab 中选出的具有高亲和力的待克隆抗体子集;C 为克隆抗体集合,对抗体子集 Ab_{sub} 中的每个抗体进行克隆产生的抗体集合;C 为变异抗体集合,对 C 中的每个抗体进行变异后产生的集合;M_i 为抗原 Ag_i 的记忆抗体集合;F 为亲和力矩阵,$f_{ij}=1/g(\mathrm{Ag}_i,\mathrm{Ab}_j)$,表示抗原 Ag_i 与抗体 Ab_j 之间的亲和力,其中 $g(\mathrm{Ag}_i,\mathrm{Ab}_j)$ 表示抗原 Ag_i 与抗体 Ab_j 之间的欧氏距离;D 为亲和力矩阵,$d_{ij}=1/g(\mathrm{Ag}_i,\mathrm{Ab}_j)$,表示抗原 Ag_i 与 C 中抗体 Ab_j 之间的亲和力;L 为相似性矩阵,$l_{ij}=g(\mathrm{Ab}_i,\mathrm{Ab}_j)$,表示抗体之间的相似性;$\xi$ 为成熟抗体被选择的比例;σ_d 为自然死亡阈值;σ_s 为抑制阈值;$|X|$ 为集合 X 中元素个数.

算法流程如下:

1) 系统对 aiNet 进行初始化:根据状态结构图中状态节点数设置集合 Ag,随机生成初始集合 Ab.

2) 对每个抗原 Ag_i,执行①~⑨的操作:

①确定 Ag_i 与 Ab 中每个抗体 Ab_j 的亲和力 $f_{ij}(j=1,\cdots,|\mathrm{Ab}|)$;②选择 q 个高亲和力抗体产生

一个抗体子集 Ab_{sub},其中 q 可根据预期聚类数估定; ③对 Ab_{sub} 中的每个抗体 Ab_j 进行克隆,克隆后的全部抗体构成抗体集合 $C = \{C_j \mid j = 1, \cdots, q\}$,抗体 Ab_j 与抗原 Ag_i 的亲和力越高,克隆数 C_j 越大,$C_j = |C| \times f_{ij} / \sum f_{ij}$,式中:$|C|$ 为抗体克隆规模,可根据抗原数依经验确定; ④对抗体集合 C 进行变异处理产生集合 C',$C'_j = C_j + 1 / f_{ij} \times (Ag_i - C_j)$,抗体向靠近抗原方向变异,亲和力越小,变异越大; ⑤确定 Ag_i 与 C 中抗体 Ab_j 的亲和力 d_{ij} ($j = 1, \cdots, |C'|$); ⑥在 C 中重新选择 ξ% 个高亲和力抗体,加入记忆抗体集合 M_i 中; ⑦消除记忆抗体集合 M_i 中所有 $d_{ij} > \sigma_d$ 的克隆抗体; ⑧计算记忆抗体集合内抗体间的亲和力 l_{tk} ($t, k = 1, \cdots, |M_i|$); ⑨消除所有 $l_{tk} < \sigma_s$ 的记忆抗体 ($t, k = 1, \cdots, |M_i|$).

3) 将所有抗原的记忆抗体集合 M_i ($i = 1, \cdots, |Ag|$) 加入到抗体集合 Ab 中.

4) 计算抗体集合 Ab 内抗体间的亲和力 l_{tk} ($t, k = 1, \cdots, |Ab|$).

5) 消除 $l_{tk} < \sigma_s$ 的抗体 ($t, k = 1, \cdots, |Ab|$).

6) 聚类结束条件为真时转向 2),否则转 7).

7) 执行 ⑩~⑬生成 Option:

⑩以每一个聚类中的抗原所对应的状态子空间作为一个 Option 入口状态集; ⑪对于各聚类的每个边界状态 s,将 Option 的终止条件 $\beta(s)$ 置为 1,这里将 s 称为出口状态; ⑫随机设置各 Option 内部初始策略; ⑬将 Option 加入到动作集 A 中;

8) 启动 Option 内部策略并行学习算法.

算法中 1)~6) 是根据状态结构图采用 aiNet 对已探明状态空间进行聚类,聚类结束条件可以是:达到预定义的步数,达到预定义的聚类数、抗体总平均亲和力增量达到预定义的值. 预定义步数往往不能准确控制聚类误差,聚类数通常很难合理预估. 根据对实验结果的观察和比较,用抗体总平均亲和力增量控制算法结束比较理想. 步骤 7) 生成了内部策略尚未确定的 Option.

2.3 Option 内部策略并行学习算法

1) 系统随机初始化 Y 个(由 Option 个数确定)智能体的策略集(策略用 Q 表表示,策略数等于对应 Option 的出口状态数).

2) 多智能体并行执行:

①选择一个未学习过的出口状态作为终点; ②选择一个未学习过的入口状态作为起点; ③执行标准 Q - 学习,修改终点对应的 Q 表,学习起—终点间的最短路径; ④如果还有未学习过的入口状态则转 ②; ⑤如果还有未学习过的出口状态则转 ①.

3) 算法结束.

至此,Option 已由多智能体并行生成,建立的 HRL 方法与文献[8]中的无异,不再赘述.

3 仿真实验与分析

仿真实验以图 1 所示二维有障碍的栅格环境空间中学习规划任意起点到给定终点间的最短路径为任务背景,算法参数中的抗体数目(小于抗原数)和位置均随机生成,高亲和力抗体选择数 q 设为当前抗体数的 1/10,克隆规模 $|C|$ 设为 $3q$,成熟抗体被选择的比例 ξ 设为 60,自然死亡阈值 σ_d 设为 $7/E_{max}$,抑制阈值 σ_s 设为 3,结束条件为抗体总平均亲和力增量 ≤ 0,上述参数,均依据多次实验测试结果确定. 智能体 Q - 学习时,均以 0.2 的概率对环境进行探测,到达目标点的奖励信号设为 100,到达出口状态的奖励信号设为10,其他为 0.

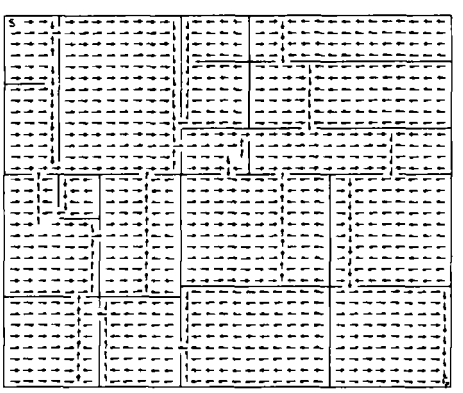
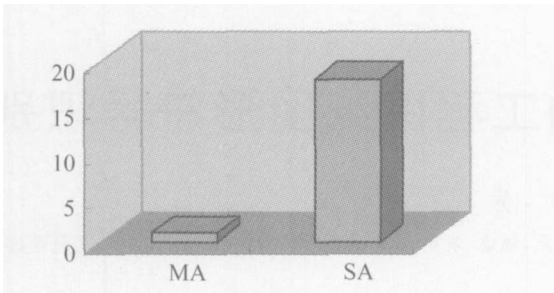


图 1 环境空间

Fig. 1 Environment space

实验对文中的基于多智能体的 Option 自动生成算法(MA)与 Mannor 的基于单智能体的 Option 自动生成算法[8](SA)的 Option 生成速度进行了比较,该实验条件下,两算法耗时比例关系如图 2 所示. MA 以空间消耗换取分层速度,这一点是容易理解的. 设生成 Option 的数量为 n,生成 Markov-Option 分层结构时,SA 的时间复杂度为 $O(n)$,MA 的时间复杂度为 $O(1)$,尽管只是从多项式级降为常数级,从理论角度讲对算法时间复杂度没有实质性的改进,但是从工程应用角度而言,这种改进是有实用价值的. 这是因为,强化学习算法本身(如 Q - 学习算法)学习时间与状态数呈指数级关系,学习速度很慢,SA 使得这种指数级耗时加倍,在大规模学习空间中,这种加倍效果会导致系统性能的急剧下降,而 MA 则有效地遏制住了这种指数级耗时的扩增,使系统性能保持在一个基本稳定的水平. 对于生成

Semi-Markov-Option 分层结构,效果尤为显著.



SA —基于单智能体算法;MA —基于多智能体算法.

图 2　耗时比例关系

Fig. 2　The scale of time-consuming

图 3 给出了基于不同生成算法的 HRL 收敛速度的实验对比结果(为了便于观察曲线变化细节,对纵坐标进行了取对数处理,数据为 100 次运行的平均结果). 由于 HRL 算法收敛速度一般都比 RL 算法快,Option 生成算法在整个 HRL 算法中所占用的时间比例明显增加,因而 Option 生成算法的速度对 HRL 算法收敛性的影响已经不可忽视. 图 3 中,基于 MA 的 HRL 算法明显比基于 SA 的算法收敛快.

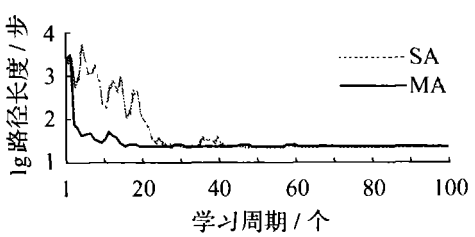


图 3　收敛速度对比

Fig. 3　The comparison of convergence speed

4　结束语

文中提出的基于多智能体系统的 Option 自动生成算法以 Option 分层强化学习方法为理论框架,主要探讨了生成 Markov-Option 分层结构的算法及其时间性能,不难拓展到 Semi-Markov-Option. 仿真实验结果和分析均表明了该算法在速度上的优越性及其在 HRL 中的实用性. 尽管时间复杂度从多项式级降为常数级在理论上对算法不构成实质性改进,但这种改进却有很好的工程应用价值.

参考文献:

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

[2] 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.

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

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

[5] DIGNEY B L. Learning hierarchical control structures for multiple tasks and changing environments[A]. Proc of the 5th International Conference on Simulation of Adaptive Behavior[C]. Zurich, Switzerland, 1998.

[6] MCGOVERN A, BARTO A. Autonomous discovery of subgoals in reinforcement learning using diverse density [A]. Proc of the 8th International Conference on Machine Learning[C]. San Fransisco: Morgan Kaufmann, 2001.

[7] MENACHE I, MANNOR S, SHIMKIN N. Q-cut: dynamic discovery of sub-goals in reinforcement learning [A]. Proc the 13th European Conference on Machine Learning[C]. Helsinki, Finland, 2002.

[8] MANNOR S, MENACHE I, HOZE A, et al. Dynamic abstraction in reinforcement learning via clustering[A]. Proc of the 21th International Conference on Machine Learning[C]. Banff, Canada, 2004.

[9] DE CASTRO L N, VON ZUBEN F N. An evolutionary immune network for data clustering[A]. Proc of the IEEE Brazilian Symposium on Artificial Neural Networks[C]. Rio de Janeiro, Brazil, 2000.

作者简介:

沈　晶,女,1969 年生,哈尔滨工程大学在读博士生. 主要从事分层强化学习、人工免疫理论的研究. 在国内外会议、期刊发表学术论文 30 余篇,参加翻译出版译著 1 部.

E-mail: shenjing@hrbeu.edu.cn

顾国昌,男,1946 年生,教授,博士生导师. 主要从事智能控制、智能机器人技术以及嵌入式系统研究,发表论文 100 余篇,并有多篇被 EI、ISTP 等收录. 任中国人工智能学会智能机器人学会理事、黑龙江省计算机学会副理事长.

刘海波,男,1976 年生,博士,IEEE 专业会员,IAIA 会员,中国计算机学会会员. 主要从事神经心理学理论、多智能体技术与智能机器人体系结构相融合的研究,发表学术论文 50 余篇,出版编著 3 部、译著 1 部.