基于MAXQ方法的分层强化学习

庞士焕[1],朱相冰[2],张 琦[3],汤萍萍[4]
(1.安徽师范大学 教育科学学院,安徽 芜湖 241000;
2.安徽师范大学 物理与电子信息工程学院,安徽 芜湖 241000;
3.西北大学 软件学院,陕西 西安 710069;
4.东南大学 计算机学院,江苏 南京 211189)

摘 要:强化学习是机器学习领域的一个重要分支,但在强化学习系统中,学习的数量会随着状态变量的个数成指数级增长,从而形成"维数灾"。为此提出了一种基于MAXQ的分层强化学习方法,通过引入抽象机制将强化学习任务分解到不同层次上来分别实现,使得每层上的学习任务仅需在较小的空间中进行,从而大大减少了学习的数量和规模。并给出具体算法——MAXQ-RLA。

关键词:分层强化学习;MAXQ;MDP
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)04-0154-03

Hierarchical Reinforcement Learning with MAXQ Method

PANG Shi-huan[1], ZHU Xiang-bing[2], ZHANG Qi[3], TANG Ping-ping[4]
(1. Department of Educational Science, Anhui Normal University, Wuhu 241000, China;
2. Department of Physics and Electronic Information, Anhui Normal University, Wuhu 241000, China;
3. Department of Software Engineering, Northwest University, Xi'an 710069, China;
4. Department of Computer Science, Southeast University, Nanjing 211189, China)

Abstract: Reinforcement learning is an important branch of machine learning. In the system of reinforcement learning, the learning stategies increase exponentially along with the number of state variables, which is called "dimensions disaster". Here a hierarchical reinforcement learning based on the MAXQ is proposed to solve this problem, which is realized by decomposing the task to different level, thus sub-tasks in every level can be solved in relatively smaller scale. This method turns out to be effective to decrease the stategies. Finally, offer the concerned algorithm—MAXQ-RLA.

Key words: hierarchical reinforcement learning; MAXQ; MDP

0 引 言

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、非监督学习和强化学习三大类。其中强化学习是一种以环境反馈作为输入的机器学习方法,其最大的特点就是很强的自动适应环境的能力[1],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。具体来说,强化学习是指从环境状态到行为映射的学习,以使系统行为从环境中获得的累积奖赏值最大。该方法主要是通过试错的方法来发现最优策略[2]。但是,在强化学习系统中,学习的数量会随着状态变量的个数成指数级增长,从而形成所谓的"维数灾"[3]。为了克服这种维数灾,通过引入抽象机制将强化学习任务分解到不同层次上来分别实现,这样每层上的学习任务仅需在较小的空间中进行,从而大大减少了学习的数量和规模,这种方法称之为分层强化学习。

在分层强化学习中,由于每个子任务的参数比较少,所以学习中需要的试错也比较少,再加上子任务可以重用,因此学习新问题的速度会更快。当前有三种方法来实现这个目标,第一种是把分层作为一个固定的模块,这样可以大大加快最优化策略的计算;第二种是通过一个执行引擎来设计一个抽象的分层结构,它可以限制各种可能的学习策略,该方法被称为分层最优化;第三种也同样依靠设计好的执行引擎进行分层,

收稿日期:2008-07-01
基金项目:安徽省教育重点项目(KJ2008A142C);安徽省自然科学基金项目(KJ2007B061)
作者简介:庞士焕(1955-),女,安徽芜湖人,实验师,研究方向为计算机科学。

在这种分层中每个子任务对应着自己的马尔可夫决策过程(Markov Decision Process,MDP),最终是寻求一个对每个子任务都局部最优的策略,这种方法被称为递归最优化[4]。

MAXQ方法是利用先验的知识将任务按照一定的层次进行组织并只在调用子任务时才进行决策的方法,具有良好的在线学习能力[5,6]。文中将介绍用 MAXQ 的方法来实现分层强化学习,它是对递归最优化方法的扩展。在这种分层结构的每一层,每个子任务都符合马尔可夫过程,从而使得分解后的所有子任务能够被标准的强化学习方法所解决。

1　用 MAXQ 来实现分层强化学习

首先通过一个出租车的例子感性地认识一下 MAXQ 方法。

如图 1 所示,一个 5 * 5 的正方形,一共有 25 个方格,其中 R、B、G、Y 代表四个固定的位置,乘客可以选择四个中的任意一个位置上车,然后在任意一个位置下车,出租车必须先到达乘客等候的位置再接送到指定的位置,其中每走一格都要消耗一定的油量,F 代表着加油站。因此,在这个问题中有七个基本动作:四个表示方向的动作 North、South、East 和 West,一个上车的动作 Pickup ,一个下车的动作 Putdown,一个加油的动作 Fillup。每一个动作得到 - 1 分的奖赏,如果成功地接送乘客到指定的位置得到 + 20 分的奖赏,非法的上车和下车得到 - 10 分的奖赏,最后如果没油了就得到 - 20 分的奖赏。希望找到一个策略使得每一步的平均奖赏最大,也就是使得完成一次接送任务的总奖赏值最大。

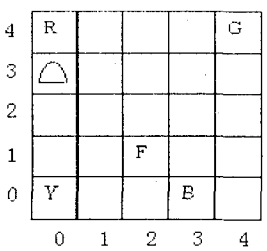

图 1　出租车行驶的范围

如图 2 所示,就是上述出租车问题的一张 MAXQ 图,这张图有两种节点:MAX 节点(三角形表示)和 Q 节点(矩形表示)。图中没有子节点的 MAX 节点是七个基本动作,有子节点的 MAX 节点代表着子任务:Navigate(t)表示指定方向,Get 表示到指定位置接乘客上车,Put 是到目的位置后乘客下车,Refuel 表示到加油站加油,Root 表示根节点。与 MAX 节点直接关联的子节点是 Q 节点,Q 节点表示的是完成父节点任务所需完成的动作。区分 MAX 节点和 Q 节点是为了使子任务能被共享和重用。

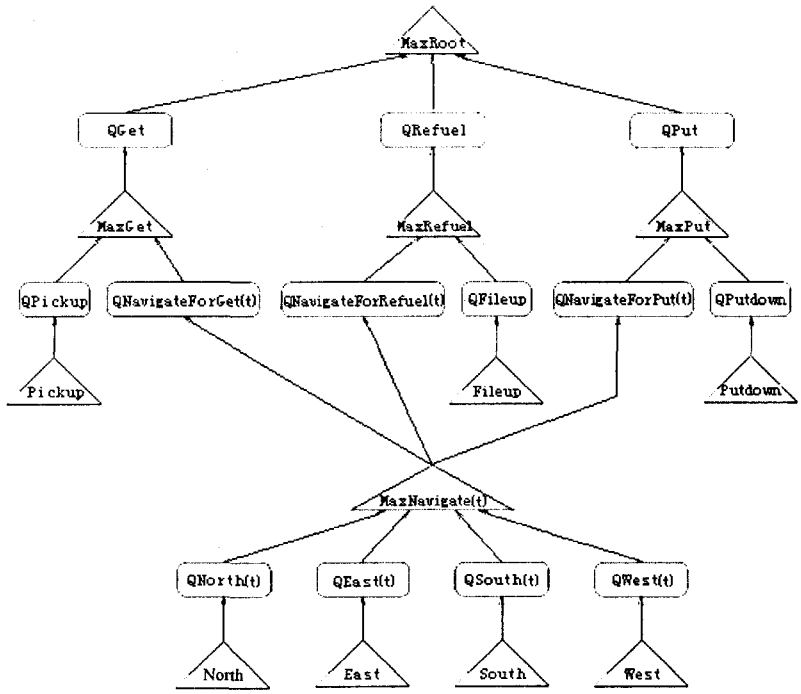

图 2　出租车问题的 MAXQ 图

下面给出 MAXQ 方法的形式化定义,在这之前首先给出马尔可夫决策过程(MDP)的形式化 定义。马尔可夫决策过程是由四元组 $< S, A, R, P >$ 定义的,它包含一个环境状态集合 S,系统行为集合 A,奖赏函数 R 和状态转移函数 P。

记 $R(s' \mid s, a)$ 为系统在状态 s 采用 a 动作使环境状态转移到 s' 获得的瞬时奖赏值:

$$R_{ss'}^{a} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\} \text{for all} s, s' \in S, a \in A(s)$$

记 $P(s' \mid s, a)$ 为系统在状态 s 采用 a 动作使环境状态转移到 s' 的概率:

$$P_{ss'}^{a} = Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \text{for all} s, s' \in S, a \in A(s)$$

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。

在 MAXQ 分层方法中,整个任务被看作是一个马尔可夫决策过程 M, 每一个 MAX 节点 i 对应着一个独立的子任务 M_i,MAX 节点 i 的子节点是 M_i 的动作。M_i 将状态集合 S 分成两个不相交的子集 S_i 和 T_i,其中

T_i 表示终止状态。T_i 的子集 G_i 表示目标状态，π_i 表示子任务 i 的策略。MAXQ 的分层策略就是一组策略的集合 $\pi = \{\pi_0, \cdots, \pi_n\}$，每个策略指导每个 MAX 节点如何选择行动。

下面给出 MAX 节点 i 的值函数 $V_i^\pi(s)$，它表示从初始状态 s 按照策略 π 直到进入终止状态所得的累积奖赏值。对于一个固定的策略 π 子任务都有一个状态转移概率函数 $P_i^\pi(s' \mid s, a)$，它表示环境在状态 s 下 M_i 执行 a 动作后转移到 s' 状态的概率。这样，就得到了一个值函数递归分解公式：

$$V_i^\pi(s) = V_a^\pi(s) + \sum_{s'} P_i(s' \mid s, a) V_i^\pi(s') \quad (1)$$

其中，$a = \pi_i(s)$，即 a 是子任务 M_i 的子节点。

再引入一个完成函数 $C_i^\pi(s, a)$ 来代替公式(1) 中的 $\sum_{s'} P_i(s' \mid s, a) V_i^\pi(s')$，它表示在状态 s 下按照策略 π 执行动作 a 后完成子任务 M_i 的累积奖赏值。这样，公式(1) 可以改为：

$$V_i^\pi(s) = V_a^\pi(s) + C_i^\pi(s, a) \quad (2)$$

其中，$a = \pi_i(s)$，$C_i^\pi(s, a) = \sum_{s'} P_i(s' \mid s, a) V_i^\pi(s')$

因此，根据上面的公式，在状态 s 下按照策略 π，假设顶层子任务 M_0 的策略选择了子任务 Ma1，Ma1 的策略选择了 Ma2，如此依次选择下去，直到子任务 Man - 1 的策略选择了基本动作 a_n，则根节点子任务的值函数 $V_0^\pi(s)$ 分解为：

$$V_0^\pi(s) = V_{a_n}^\pi(s) + C_{a_{n-1}}^\pi(s, a_n) + \cdots + C_{a_1}^\pi(s, a_2) + C_0^\pi(s, a_1) \quad (3)$$

其中，假设 $(0, a_1, a_2, \cdots, a_n)$ 为按照策略 π 得到的一条自上而下的节点路径。图 3 显示了 $V_0^\pi(s)$ 的计算方法。

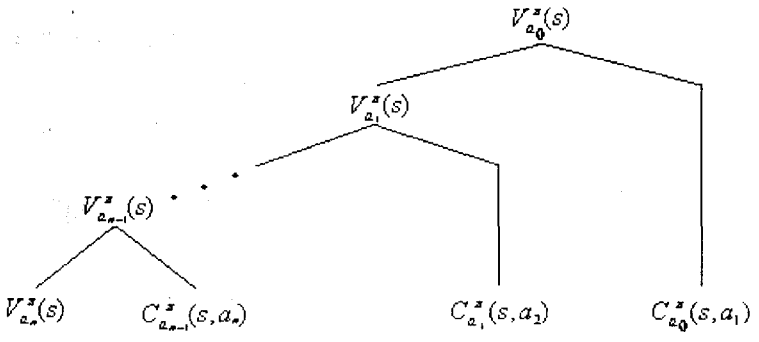

图 3 出租车实例根节点值函数的计算方法

在 MAXQ 图中，MAX 节点和 Q 节点被看作是值函数计算的执行部分，每个 MAX 节点 i 用来计算子任务的值函数 $V_i^\pi(s)$，Q 节点保存完成函数 $C_i^\pi(s, a)$ 的值，非基本子任务的值函数要调用子 Q 节点来计算 $V_a^\pi(s)$，再加上 $C_i^\pi(s, a)$。以下将给出具体的实现算法——MAXQ - RLA。

2 MAXQ - RLA 学习算法

这里将具体给出基于 MAXQ 方法的分层强化学习算法——MAXQ - RLA。在该学习算法中，对于任意的马尔可夫决策过程 M，都将找到一个最优的策略。它引入两个完成函数 $C_i(s, a)$ 和 $\tilde{C}_i(s, a)$，并且使用不同的更新规则来学习这两个函数。MAXQ - RLA 算法采用递归实现，下面给出算法描述(见表 1)：

表 1 MAXQ - RLA 学习算法

```
function MAXQ - RLA(MaxNodei, State s)
初始化 seq = ()
if i 是一个元 MaxNode;
执行 i，接受奖赏值 r，观测结果状态 s';
V_{t+1}(i,s) = (1 - α_t(i)) · V_t(i,s) + α_t(i) · r_t
将 s 加入 seq 之首
else
初始化 count = 0
while 没有进入终态 do
根据当前的策略 π 选择一个动作 a
初始化 childSeq = MAXQ - RLA(a,s)
观测结果状态 s'
初始化 a* = arg max_a' [C̃_t(i,s',a') + V_t(a',s')]
初始化 N = 1
for childSeq 中的每一个 s do
C̃_{t+1}(i,s,a) = (1 - α_t(i)) · C̃_{t+1}(i,s,a) + α_t(i)
· γ^N [R̃_i(s') + C̃_t(i,s',a*) + V_t(a*,s)]
C_{t+1}(i,s,a) = (1 - α_t(i)) · C_{t+1}(i,s,a) + α_t(i)
· γ^N [C_t(i,s',a*) + V_t(a*,s')]
N = N + 1
end
将 childSeq 加到 seq 之首
s = s'
end
end
return seq
end MAXQ - RLA
```

在此算法中，$a*$ 是指根据在状态 s' 下的完成函数 $\tilde{C}$ 和值函数 V 而得到的最优动作。$\alpha_t(i)$ 是指在时间 t 下的学习率。因此，当 MAXQ - RLA 算法收敛时，递归最优策略可以通过计算每个节点的最大 $\tilde{Q}(i, s, a) = \tilde{C}(i, s, a) + V(a, s)$ 来得到。

3 结束语

文中定义了基于 MAXQ 的分层强化学习方法，并显示了 MAXQ 图的分层结构原理，它可以精确地表示与子任务一致的值函数，最后还给出了具体的学习算

(下转第 169 页)

式目标的端口扫描行为。通过实验分析,可以看出 APIDS 体系结构可以有效地检测分布式攻击。

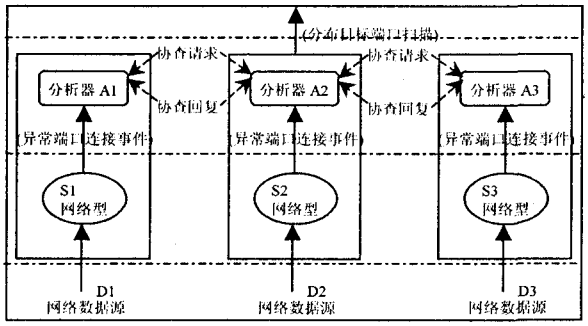

图 3　APIDS 模型

5　结束语

笔者创新点在于将移动 Agent 技术和 P2P 理论结合起来,构造出一个新型的入侵检测模型。移动 Agent 技术和 P2P 理论都是近年来的新兴技术,鉴于它们都具有自适应、自学习、分布式的特点,文中试图利用它们在技术上的优势来构建分布式入侵检测模型。它融合了两者的优势,是一个智能的、鲁棒的入侵检测系统,能同时进行多层次的监测和不同级别的响应。基于 P2P 的移动 Agent 入侵检测系统模型可以实时地检测基于网络和基于主机的入侵行为。该模型是个开放的系统,易于加入新的 Agent,扩充新的检测模式。模型充分利用了 P2P 分布式无中心和移动 Agent 的特点,各 Agent 间既分工又联合的协同作战,合作完成检测任务。另外模型采用一定状态的检查和验证策略,保证了系统自身的安全系统是完全分布式的,监视 Agent 生成后在网络上漫游,因此一个节点被攻破不会导致整个系统丧失检测功能。

下一步的研究工作主要集中在如何实现一个高效的算法,来保障 Agent 之间的通信以及它们在网络中移动的安全问题。

参考文献:

[1] Allen J, Christie A. State of the Practice of Intrusion Detection Technologies[R]. Technical Report, Networked Systems Survivability Program. [s.l.]:[s.n.],2000:47-83.

[2] 徐　峰,宋如顺,赵　洁,等. 基于 P2P 多 Agent 数据融合入侵检测模型研究[J]. 计算机工程与应用, 2004(17):159-161.

[3] The Intrusion Detection Message Exchange Format . draft-i-etf-idwg-id-mef-xml-12[S/OL]. 2005-04. http://www.ietf.org.

[4] 李　洛,李拥军. 基于 Agent 多媒体数据库模型的研究[J]. 计算机应用研究,2002(10):191-194.

[5] 黄道颖,黄建华,庄　雷,等. 基于主动网络的分布式 P2P 网络模型[J]. 软件学报, 2004(7):1081-1089.

[6] Zhu Y, Hu Y M. Efficient Proximity-Aware Load Balancing For DHT-Based P2P Systems[J]. IEEE Tran. Parallel and Distributed Systems,2005,16(4):349-361.

[7] 张云勇. 刘锦德. 移动 Agent 技术[M]. 北京:清华大学出版社,2003.

[8] Asaka M, Okazawa S. The Implementation of IDA: An Intrusion Detection Agent System[C]//Proceedings. North Falmouth:[s.n.],2001:81-92.

[9] Jansen W, Mell P, Karygiannis T, et al. Applying Mobile Agents to Intrusion Detection and Response[R]. National Institute of Standards and Technology Computer Security Division, NIST Interim Report (IR)-6461.[s.l.]:[s.n.],1999.

[10] 李　兵. 一种基于对等模型的网络入侵检测系统模型[J]. 计算机技术与发展,2008,18(3):172-176.

(上接第 156 页)

法——MAXQ-RLA,该算法可以轻便地找到最优的策略。但是,MAXQ 方法是利用先验知识对任务进行人工分层,自动分层的能力较弱,且分层粒度不够精细,如在出租车问题中,它难于进一步对导航子任务进行抽象,这些问题还有待于进一步的研究。

参考文献:

[1] Iima H, Kuroe Y. Swarm reinforcement learning algorithms - exchange of information among multiple agents[C]//SICE, 2007. Annual Conference. JAPAN: SICE, 2007:2779-2784.

[2] Erfu Y, Yang E. A Multiagent Fuzzy Policy Reinforcement Learning Algorithm with Application to Leader-Follower Robotic Systems[C]//Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference. New YorK: IEEE, 2006:3197-3202.

[3] Handa H. Evolutionary Computation on Multitask Reinforcement Learning Problems[C]//Networking, Sensing and Control, 2007 IEEE International Conference. New York: IEEE, 2007:685-688.

[4] Watanabe T, Takahashi Y. Hierarchical reinforcement learning using a modular fuzzy model for multi-agent problem[J]. Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference. New York: IEEE, 2007:1681-1686.

[5] Dietterich T G. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition[J]. Journal of Artificial Intelligence Research, 2000, 13:227-303.

[6] Dietterich T G. The MAXQ method for hierarchical reinforcement learning[C]//Proc of the 15th ICML. San Francisco: Morgan Kaufmann, 1998:118-126.