第 19 卷 第 11 期　　控 制 与 决 策　　2004 年 11 月
Vol. 19 No. 11　　Control and Decision　　Nov. 2004

文章编号: 1001-0920(2004)11-1263-04

求解部分可观测马氏决策过程的强化学习算法

王学宁[1], 贺汉根[1], 徐 昕[1,2]

(1. 国防科技大学 自动化研究所, 湖南 长沙 410073; 2. 国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 针对部分可观测马氏决策过程(POMDP)中, 由于感知混淆现象的存在, 利用 Sarsa 等算法得到的无记忆策略可能发生振荡的现象, 研究了一种基于记忆的强化学习算法——CP*n*Sarsa(λ)学习算法来解决该问题. 它通过重新定义状态, Agent 结合观测历史来识别混淆状态. 将 CP*n*Sarsa(λ)算法应用到一些典型的 POMDP, 最后得到的是最优或近似最优策略. 与以往算法相比, 该算法的收敛速度有了很大提高.

关键词: 强化学习; 部分可观测 Markov 决策过程; Sarsa 学习; 无记忆策略

中图分类号: TP319　　**文献标识码**: A

Reinforcement learning algorithm for partially observable Markov decision processes

WANG Xue-ning, HE Han-gen, XU Xin

(Institute of Automation, National University of Defence Technology, Changsha 410073, China. Correspondent: WANG Xue-ning, E-mail: wxn9576@163.com)

Abstract: In partially observable markov decision processes (POMDP), due to perceptual aliasing, the memoryless policies obtained by Sarsa-learning may oscillate. A memory-basednew reinforcement learning algorithm—CpnSarsa(λ) is studied to solve this problem. With new definitions of states, the agent combines current observation with preobservations to distinguish aliasing states. With application of the algorithm to some typical POMDP, the optimal or almost-optimal policies are obtained. Comparing with previous algorithms, this algorithm greatly improves the convergence rate.

Key words: reinforcement learning; POMDP; Sarsa-learning; memoryless policy

1 引　言

在部分可观测的 Markov 决策过程(简称 POMDP)中, 如果两个不同的实际状态呈现出相同的观测值, 智能体(Agent)将无法区分它们, 这种现象称为感知混淆现象. 正是由于感知混淆现象的存在, 使得强化学习的一些常用算法(如 *Q*-学习或 Sarsa 学习)不能找到 POMDP 的最优策略[1,2], 并且 Littman 等在理论上证明了 *Q*-学习算法求解 POMDP 问题时并不收敛[3,4]. 但是, Loch 等提出结合资格迹的 Sarsa(λ)算法在解决 POMDP 问题时却有很强的适应性, 并指出 Sarsa(λ)算法可找到 POMDP 的最优无记忆策略[5].

在有些 POMDP 中, 即使是最优的无记忆策略, 也会发生振荡, 使得策略的性能降低. 为解决这一问题, Masayuki 等提出在回溯过程中跳过混淆状态[6]. 但如何将混淆状态与非混淆状态区分开则是一个难题.

如果结合前 *n* 步的观测状态, 那么就有可能识

收稿日期: 2003-11-20; **修回日期**: 2004-02-10.

基金项目: 国家自然科学基金重点项目(60234030); 青年科学基金资助项目(60303012).

作者简介: 王学宁(1976—), 男, 山东阳谷人, 博士生, 从事机器学习、智能控制等研究; 贺汉根(1943—), 男, 浙江杭州人, 教授, 博士生导师, 从事智能控制、机器学习等研究.

别混淆状态. 本文提出的新算法CPnSarsa(λ)正是基于这一思想,消除了因感知混淆现象引起的策略振荡,最后可找到最优或近似最优策略. 实际上,CPnSarsa(λ)算法的实质是利用Sarsa(λ)求解由POMDP转换成的MDP,因此与其他解决POMDP的算法相比,收敛速度要快得多.

2 POMDP及其无记忆策略的振荡

一个POMDP可利用6元组表示为$\langle S, A, T, R, Z, O\rangle$[7]. 其中:$S$为状态空间,$A$为行为空间,$T: S \times A \to \Pi(S)$为状态转移概率,$R: S \times A \to R$为回报函数,$Z$为观测状态空间,$O: S \times A \to \Pi(Z)$为观测概率. 在给定$t$时刻$S$和$A$的情况下,决定$t+1$时刻的观测状态在观测状态空间的概率分布.

学习的目标是选择一个策略,使得折扣型总回报函数

$$J = E\left[\sum_{t=0}^{\infty} \gamma r_t\right] \tag{1}$$

达到最大值. 其中:$0 < \gamma < 1$为折扣因子,r_t为时刻t的回报. 在强化学习问题中,假定Agent不知道状态转移概率和观测概率等环境模型信息.

在POMDP中,要保证找到最优策略,必须记住整个过程的历史[9],即策略是全部历史到行为空间A上概率分布的一个映射:$H \to A$. 但记住全部历史显然是不可行的,解决方案之一是将策略定义成从最后一个观测状态到行为空间的映射:$\pi\ Z \to A$,这就是无记忆策略. 但即使在简单的4×3方格问题中,利用Sarsa(λ)算法找到的最优无记忆策略也会发生振荡,如图1所示. 图中,黑色表示障碍,+1表示目标状态,-1表示惩罚状态,目标状态和惩罚状态都是吸收状态. 其他格内左上方的数字表示观测状态,右上方的箭头表示利用Sarsa(λ)算法得到的最优无记忆策略,左下方数字表示环境的实际状态.

0 ↑ 0	2 → 1	2 → 2	+1 3
3 ↑ 4		0 ↑ 5	−1 6
0 ↑ 7	2 → 8	2 → 9	1 ← 10

图1 4×3方格问题及其最优无记忆策略

在4×3方格问题中,共有11个状态,4个行分别为上、下、左、右. 状态的转移有一定的随机性,有80%的可能性到达目的地,有10%的可能性滑向侧面的任何一方,Agent仅能观测到自己的左侧和右侧有无障碍,因此有4个观测状态. 另外,Agent还能识别目标状态和惩罚状态,使Agent到达目标状态、惩罚状态及其他行为的回报分别为1,-1和-0.04. 由于不完全可观性,状态0,5,7具有相同的观测状态0,则Agent认为这3个状态是同一状态,而导致最优无记忆策略在状态0时形成振荡. 同样,右下角的状态9与10之间也出现振荡. 这两处振荡使无记忆策略的性能很低,平均每步得到的回报只有0.02.

3 求解POMDP问题的CPnSarsa(λ)算法

求解POMDP问题时,Sarsa(λ)学习算法是将观测状态作为实际状态,也就是说,只要两个状态的观测状态相同,Agent便认为这两个状态是同一个状态. 因此,在感知混淆现象严重时,无法找到最优策略. 在本文的CPnSarsa(λ)算法中,认为只有在前n步的观测状态都分别相同的情况下,两个状态才相同,这样便可有效地解决感知混淆的现象. 此时,需要定义状态为观测状态的组合,而不是环境的实际状态.

定义1 在CPnSarsa(λ)算法中,当前状态为$\hat{s} = (z^n, z^{n-1}, \cdots, z^1, z)$,如果下一步的观测状态为$\hat{z}$,则下一步的状态为

$$\hat{s} = (z^{n-1}, z^{n-2}, \cdots, z^1, z, \hat{z}). \tag{2}$$

其中:z^i为前面第i步的观测状态,z为当前的观测状态.

定义2 策略π的行为值函数

$$Q^{\pi}(s, a) = Q^{\pi}(z^n, z^{n-1}, \cdots, z^1, z, a) = E_{\pi}\left(\sum_{k=0}^{\infty} \gamma r_k \,\middle|\, z^n, z^{n-1}, \cdots, z^1, z, a\right) \tag{3}$$

表示在前n步观测状态为$(z^n, z^{n-1}, \cdots, z^1, z)$时,采取行为$a$,然后一直采取策略$\pi$得到的回报的期望值. 其中:$0 < \gamma < 1$为折扣因子,$r_t$为时刻$t$的回报.

定义3 最优行为值函数

$$Q^{*}(s, a) = \max_{\pi} Q^{\pi}(s, a). \tag{4}$$

如果能得到最优行为值函数,就很容易得到最优策略[9]. 根据上述3个定义,可利用与Sarsa(λ)算法中相同的迭代方法逼近最优行为值函数. 值得一提的是,资格迹包含了多步预测的思想,因而在解决POMDP时,利用资格迹使得算法具有良好的适应性[5],并可加快收敛速度. 因此本文算法中采用了包含资格迹的Sarsa(λ)迭代方法.

对于所有的$(z^n, z^{n-1}, \cdots, z^1, z, a)$,值函数迭代公式为

$$Q_{t+1}(z^n, z^{n-1}, \cdots, z^1, z, a) =$$

$$Q_t(z^n, z^{n-1}, \cdots, z^1, z, a) + \alpha\delta_t e_t(z^n, z^{n-1}, \cdots, z^1, z, a). \quad (5)$$

其中

$$\delta_t = r_{t+1} + \gamma Q_t(z_t^{n-1}, \cdots, z_t, \hat{z}_t, a_{t+1}) - Q_t(z_t^n, \cdots, z_t, a_t),$$

α 为学习因子, r_t 为时刻 t 的回报, e_t 为资格迹

对于所有的 $(z^n, z^{n-1}, \cdots, z^1, z, a)$, 资格迹的迭代公式为

$$e_t(z^n, \cdots, z, a) = \begin{cases} \gamma\lambda e_{t-1}(z^n, \cdots, z, a) + 1, z^i = z_t^i, a = a_t; \\ \gamma\lambda e_{t-1}(z^n, \cdots, z, a), \text{otherwise} \end{cases} \quad (6)$$

其中: $0 < \gamma < 1$ 为折扣因子, $0 \leq \lambda \leq 1$ 为常数

下面给出 CPnSarsa(λ) 算法的完整描述:

给定如下条件: 有限离散观测状态空间和行为空间 POMDP 的观测状态集 Z 和行为集 A; 折扣总回报目标函数, 其中折扣因子为 $0 < \gamma < 1$; 以表格形式存储的行为值函数估计 $Q(s, a)$, 即

$$Q(z^n, z^{n-1}, \cdots, z^1, z, a);$$

行为探索策略 π_0.

(1) 初始化行为值函数估计 Q 和学习因子 α

(2) 对所有的 $(z^n, z^{n-1}, \cdots, z^1, z, a)$, 令资格迹 $e(z^n, z^{n-1}, \cdots, z^1, z, a) = 0$

(3) 循环, 直到满足停止条件:

1) 初始化 $(z_0^n, z_0^{n-1}, \cdots, z_0^1, z_0, a_0)$, 令 $t = 0$;

2) 循环, 直到目标状态(或吸收状态) 出现:

① 执行行为 a_t, 观测下一时刻的观测状态 $\hat{z}$ 以及回报 r;

② 根据策略 π_0 和行为值函数 Q, 选择状态为 $(z_t^{n-1}, \cdots, z_t^1, z_t, \hat{z})$ 时的行为 a_{t+1};

③ 根据式(5) 更新当前行为值函数;

④ 根据式(6) 更新资格迹;

⑤ 令

$$(z_{t+1}^n, z_{t+1}^{n-1}, \cdots, z_{t+1}^1, z_{t+1}) = (z_t^{n-1}, \cdots, z_t^1, z_t, \hat{z}),$$

更新学习因子 α

⑥ 令 $t = t + 1$, 并返回 ①.

4　算法的收敛性

CPnSarsa(λ) 算法的实质是利用 Sarsa(λ) 求解 MDP 问题, 所以其收敛条件与 Sarsa(λ) 算法的收敛条件相同. 根据文献[10] 给出的表格型 Sarsa(λ) ($\lambda = 0$) 算法的收敛性定理, 可得到如下定理:

定理 1　对于有限观测状态和行为空间的 POMDP, 设 $Q(z^n, z^{n-1}, \cdots, z^1, z, a)$ 为 CPnSarsa(λ) ($\lambda = 0$) 算法迭代计算得到的行为值函数估计, 当满足以下条件:

1) 行为值函数以表格形式存储;

2) 学习因子满足

$$0 < \alpha_t < 1, \sum_0^\infty \alpha_t = \infty, \sum_0^\infty \alpha_t^2 < \infty;$$

3) $\mathrm{Var}[r] < \infty$;

4) 行为选择策略保证算法对状态和行为空间进行无限遍历

则 $Q(z^n, z^{n-1}, \cdots, z^1, z, a)$ 以概率 1 收敛到最优行为值函数

5　实验与讨论

多步预测可改善 POMDP 中行为值函数的收敛性[5], 因而 λ 越趋近于 1, 算法的收敛性越好. 综合考虑收敛速度等性能指标, 本文在各例中选择 $\lambda = 0.9$. 在选择行为时, 采用 ϵ- 贪心策略, 算法开始时为 $\epsilon = 0.2$, 随着步数的增多而线性递减, 到 2×10^5 步时, ϵ 衰减到 0.

通过实验发现, 对于很多 POMDP 问题, 只需要结合前一步或两步观测状态, 就能找到近似最优策略. 例如: 将 CPnSarsa(λ) 算法应用于 4×3 方格问题时, 只需要结合前一步的观测状态便可得到近似最优策略, 其平均每步的回报接近于最优策略的 0.12, 远大于 Sarsa(λ) 算法的 0.02, 如图 2 所示. 尽管平均每步的回报的性能指标稍不如利用 SPOVA-RL 方法得到的策略[11] 及 WITNESS 算法, 但在收敛速度上却有很大提高, 如表 1 所示

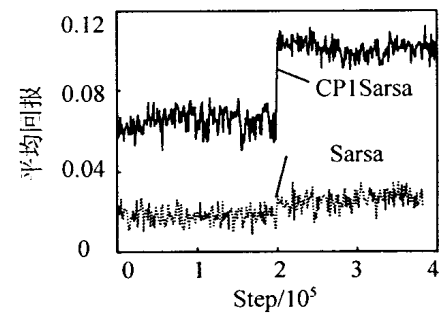

图 2　4×3 方格问题中 CP1Sarsa(λ) 算法与 Sarsa(λ) 算法性能比较

将 CPnSarsa(λ) 算法应用于 Shuttle 问题时, 需要利用前两步的观测. Shuttle 问题的任务是在两个码头之间运送货物, 详见文献[8]. 它共有 8 个状态: 5 个观测状态, 由于干扰的原因, 可能会得到错误的观测状态; 3 个行为分别是调头、前进和后退, 其中后退时到达预期目的的可信度较低. 图 3 所示的 3 条曲线分别是无记忆策略平均每步的回报

表 1 各种算法解决 4 × 3 问题的性能比较

算 法	所用时间	平均每步回报
CP1Sarsa(λ)	60 s	0.11
SPOVA-RL	42 min	0.12
WITNESS	> 1 h	0.12

(Sarsa 算法), 结合前一步观测状态得到的策略平均每步的回报(CP1Sarsa 算法), 结合前两步观测状态得到的策略平均每步的回报(CP2Sarsa). 可以看出, 结合前两步观测状态得到的策略平均每步的回报约为 1.8, 与最优策略相近

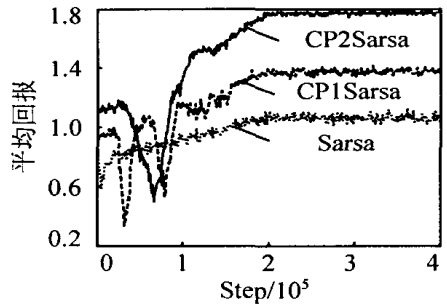

图 3 Shuttle 问题中各种算法性能比较

本文算法解决 Shuttle 问题比其他算法(如 IState-GPOMDP)[12] 的收敛速度要快得多, 如表 2 所示

表 2 各种算法解决 Shuttle 问题所用时间

算 法	所用时间
CP2Sarsa(λ)	70 s
Istate-GPOMDP	5 min

6 结 语

由实验结果可以看出, 尽管 Sarsa(λ) 可解决行为值函数不收敛的问题, 但由于无法识别混淆状态, 在求解一些 POMDP 问题时, 得到的无记忆策略会发生振荡. 通过重新定义状态, CP*n*Sarsa(λ) 算法结合前 *n* 步观测状态, 可较好地识别隐藏状态, 因而可消除振荡, 并最后得到的策略是最优或近似最优的. 与以往的算法相比, 收敛速度有了明显提高.

参考文献(References):

[1] Tsitsiklis J N, Roy B V. An analysis of temporal difference learning with function approximation [J]. *IEEE Trans on Automatic Control*, 1997, 42(5): 674-690.

[2] Chrisman L. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach [A]. *Proc of the Tenth National Conf on Artificial Intelligence*[C]. California, 1992. 183-188.

[3] Littman M. Memoryless policies: Theoretical limitations and practical results[A]. *Proc of the Third Int Conf on Simulation of Adaptive Behavior*[C]. Combridge, 1994. 238-245.

[4] Singh S, Jaakkola T, Jordan M. Learning without state-estimation in partially observable Markov decision processes[A]. *Proc of the Eleventh Int Conf on Machine Learning*[C]. New Brunswick, 1994. 284-292.

[5] Loch L, Singh S. Using eligibility traces to find the best memoryless policy in partially observable Markov decision processes[A]. *Proc of the Fifteenth Int Conf on Machine Learning*[C]. Madison, 1998. 323-331.

[6] Masayuki Ohta, Itsuky Noda. Adjusting backup-length automaticlly in reinforcement learning[A]. *Proc of the Second Int Conf on Machine Learning and Cybernetics*[C]. Xi'an, 2003. 1624-1629.

[7] Kaelbling L, Littman M, Cassandra A. Planning and acting in partially observable stochastic domains [J]. *Artificial Intelligence*, 1998, 101(1): 99-134.

[8] Cassandra A. Exact and approximate algorithms for partially observable Markov decision processes [D]. Brown University, 1998.

[9] Sutton R, Barto A. *Reinforcement Learning: An Introduction*[M]. MIT Press, 1998.

[10] Singh, Jaakkola, Littman, et al. Convergence results for single-step on-policy reinforcement learning algorithms[J]. *Littman, Machine Learning*, 2000, 38(3): 287-308.

[11] Parr R, Russell S. Approximating optimal policies for partially observable stochastic domains[A]. *Proc of the Int Joint Conf on Artificial Intelligence*[C]. San Francisco, 1995. 1088-1094.

[12] Douglas Alexander Aberdeen. Policy-gradient algorithms for partially observable Markov decision processes[D]. Australian National University, 2003.