第 26 卷 第 4 期
2000 年 7 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 26, No. 4
Jul., 2000

短文

平均准则问题的即时差分学习算法[1)]

胡光华 吴沧浦

(北京理工大学自动控制系 北京 100081)

(E-mail: wucpu@sun.ihep.ac.cn)

摘 要 考虑平均准则随机动态规划(SDP)问题的一族在线即时差分(TD)学习算法. 在学习中,平均问题的相对值函数是控制器所要学习的目标函数. 所提出的算法是已有的 TD(λ)算法及 R-学习算法的一种推广.

关键词 即时差分学习,强化学习,动态规划,Monte Carlo 方法.

TEMPORAL DIFFERENCE LEARNING ALGORITHMS FOR AVERAGE REWARD PROBLEM

HU Guanghua WU Cangpu

(*Department of Automatic Control, Beijing Institute of Technology, Beijing* 100081)

Abstract In this paper, some on-line TD(λ) learning algorithms for average reward stochastic dynamic programming problems are presented. During learning, the relative function is the object to be predicted by the agent. This work is an extension to and generalization of the work on previous TD(λ) methods and R-learning algorithms.

Key words Temporal-difference learning, reinforcement learning, dynamic programming, Monte Carlo method.

1 引言

即时差分(TD)学习算法首先由 Sutton[1]提出,用以逼近在给定策略下各状态的期望长期累积回报函数. 在此之后,有大量的文献对此进行了深入的研究[2~4]. 但其研究均集中在有限阶段或折扣 SDP 问题上. 这类问题自然具有很广的实际意义,同时理论上也较易于进行分析. 凭借动态规划算子的压缩性,其收敛性已得到了证明[2,3]. 然而有一部分实

1)国家自然科学基金资助项目.

收稿日期 1998-06-23 收修改稿日期 1999-06-23

际问题目标却是要优化每步平均报酬.若使用折扣的方法来求解常常会陷入次优解中,或者当折扣因子接近 1 时收敛速度十分慢[4].此时平均准则模型更为适合此类问题

设(S,P,r)为所考虑的 SDP 模型,其中 S 表示有限状态集,P 为转移概率矩阵,其元素 p_{ij}表示由状态 i 转入状态 j 的概率,$r(i,j)$为上述转移所获得的瞬时报酬.定义状态 i 的每步平均报酬 $\rho(i)$为

$$\rho(i) = \lim_{N\to\infty} E\left[\frac{\sum_{t=0}^{N-1} R_t(i)}{N}\right], \tag{1}$$

其中 $R_t(i)$为学习单元由状态 i 出发,到时刻 t 时所收到的瞬时报酬.

Bertsekas[5]证明了若 P 所对应的马氏链为单一遍历类,则 $\rho(i)$与 i 无关,即 $\rho(i)=\rho(j)=\rho \quad \forall\, i,j\in S$,同时还存在向量 $\boldsymbol{h}$ 使如下 Bellman 方程成立

$$\boldsymbol{h}(i) + \rho = \sum_{j\in S} p_{ij}(r(i,j) + \boldsymbol{h}(j)), \qquad \forall\, i \in S. \tag{2}$$

若令 $\boldsymbol{h}(s)=$常数(通常取为 0),其中 s 为一事先固定的状态,则向量 $\boldsymbol{h}$ 还是唯一的.

对于平均准则 SDP 问题,称

$$h(i) = \mathrm{E}[\sum_{t=0}^{\infty}(R_t(i) - \rho)], \quad \forall\, i \in S \tag{3}$$

为状态 i 的偏倚值,也称为相对值[4,5].

2 平均准则 TD(λ)学习算法

上节(2)式可重写为

$$\boldsymbol{h}(i) = \mathrm{E}[r(i,j) - \rho + \boldsymbol{h}(j)], \quad \forall\ i \in S, \tag{4}$$

由此便有相应的随机逼近算法

$$\boldsymbol{h}(i_t) := \boldsymbol{h}(i_t) + \gamma(r(i_t,i_{t+1}) - \rho + \boldsymbol{h}(i_{t+1})), \tag{5}$$

其中每一次更新在状态 $i_t(i_t\neq s)$被访问时进行,γ 为正的步长参数,可随时间而递减.在逼近 $\boldsymbol{h}$ 的同时必须对 ρ 进行估计.一种方法是用以前所有报酬的平均作为 ρ 的估值

$$\rho := \rho(1 - 1/t) + r(i_t,i_{t+1})/t.$$

另一种方法是把它与 $\boldsymbol{h}$ 的估值同时考虑

$$\rho := \rho(1 - \beta) + \beta(r(i_t,i_{t+1}) + \boldsymbol{h}(i_{t+1}) - \boldsymbol{h}(i_t)),$$

其中 β 为学习速率,t 为当前时刻.算法(5)仅考虑了当前的瞬时报酬,而(3)式却依赖于整个轨道的所有报酬.一种折衷的办法是固定一个非负整数 l,并考虑 $l+1$ 次转移所得的报酬.其相应的$(l+1)$步 Bellman 方程为

$$\boldsymbol{h}(i_t) = \mathrm{E}[\sum_{m=0}^{l} r(i_{t+m},i_{t+m+1}) - \rho(l+1) + \boldsymbol{h}(i_{t+l+1})].$$

由于没有任何选取某特殊 l 的理由,故考虑对所有多步 Bellman 方程的加权平均,即对某一 $\lambda\in(0,1)$,因$(1-\lambda)\sum_{l=0}^{\infty}\lambda^l=1$,故

$$\boldsymbol{h}(i_t) = (1-\lambda)\mathrm{E}[\sum_{l=0}^{\infty}\lambda^l(\sum_{m=0}^{l}(r(i_{t+m},i_{t+m+1}) - \rho) + \boldsymbol{h}(i_{t+l+1}))] =$$

$$\mathrm{E}[(1-\lambda)\sum_{m=0}^{\infty}(r(i_{t+m},i_{t+m+1})-\rho)\sum_{l=m}^{\infty}\lambda^l+\sum_{l=0}^{\infty}(\lambda^l-\lambda^{l+1})\boldsymbol{h}(i_{t+l+1})]=$$

$$\mathrm{E}[\sum_{m=0}^{\infty}\lambda^m(r(i_{t+m},i_{t+m+1})-\rho+\boldsymbol{h}(i_{t+m+1})-\boldsymbol{h}(i_{t+m})]+h(i_t). \tag{6}$$

记 $d_t=r(i_t,i_{t+1})-\rho+\boldsymbol{h}(i_{t+1})-\boldsymbol{h}(i_t)$为 t 时刻的即时差分,则(6)式可写为

$$\boldsymbol{h}(i_t)=\mathrm{E}(\sum_{m=t}^{\infty}\lambda^{m-t}d_m)+\boldsymbol{h}(i_t). \tag{7}$$

这并不使人觉得意外,因为由 Bellman 方程(4)可得出 $\mathrm{E}[d_m]=0$ 对任意 m 成立.

现记 $\hat{\boldsymbol{h}}$ 为 $\boldsymbol{h}$ 的估计,根据 Robbins-Monro 随机逼近的思想,由(7)式即得平均准则的即时差分(TD(λ))算法为

$$\hat{h}(i_t):=\hat{\boldsymbol{h}}(i_t)+\gamma\sum_{m=t}^{\infty}\lambda^{m-t}\hat{d}_m,\qquad \text{若 } i_t\neq s, \tag{8}$$

其中

$$\hat{d}_t=r(i_t,i_{t+1})-\rho^t+\hat{\boldsymbol{h}}(i_{t+1})-\hat{\boldsymbol{h}}(i_t) \tag{9}$$

为 t 时刻的即时差分,ρ^t 为 t 时刻的每步平均报酬的估计,γ 为正的步长参数,可随时间而递减.

下面给出另外一种形式的即时差分. 若在 Bellman 方程(4)中令 $h(s)=\rho$,则由第二节知存在唯一向量 $\boldsymbol{h}$ 满足

$$\boldsymbol{h}(i)=\mathrm{E}[r(i,j)-h(s)+\boldsymbol{h}(j)],\quad \forall\, i\in S.$$

类似于前面的讨论可得出另一形式的即时差分 $\hat{d}'_t$

$$\hat{d}'_t=r(i_t,i_{t+1})-\hat{h}(s)+\hat{h}(i_{t+1})-\hat{h}(i_t). \tag{10}$$

TD(λ)算法的一种更容易实现的递推形式可通过引入所谓适合度 $e_t\in\mathscr{R}^n$ 来实现,即定义

$$e_0=\begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix},\quad e_{t+1}(i)=\begin{cases}\lambda e_t(i) & \text{若 } i\neq i_t\\ \lambda e_t(i)+1 & \text{若 } i=i_t\end{cases}$$

则 TD(λ)算法(8)变为:在 t 时刻,对所有 $i\in S$,

$$\hat{\boldsymbol{h}}(i):=\hat{\boldsymbol{h}}(i)+\gamma\hat{d}_te_t. \tag{11}$$

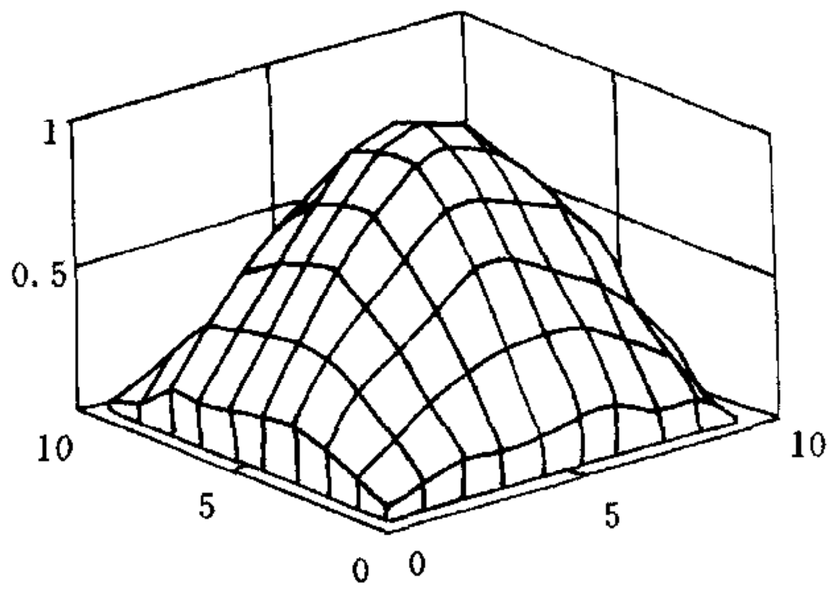
易见,若 $\lambda=1$,则(8)式正好是基于(3)式的 Monte Carlo 策略赋值方法. 而若 $\lambda=0$,则 TD(λ)算法(8)正好是在每一状态只有一个可用行动情形下的 R-学习算法.

3　实例

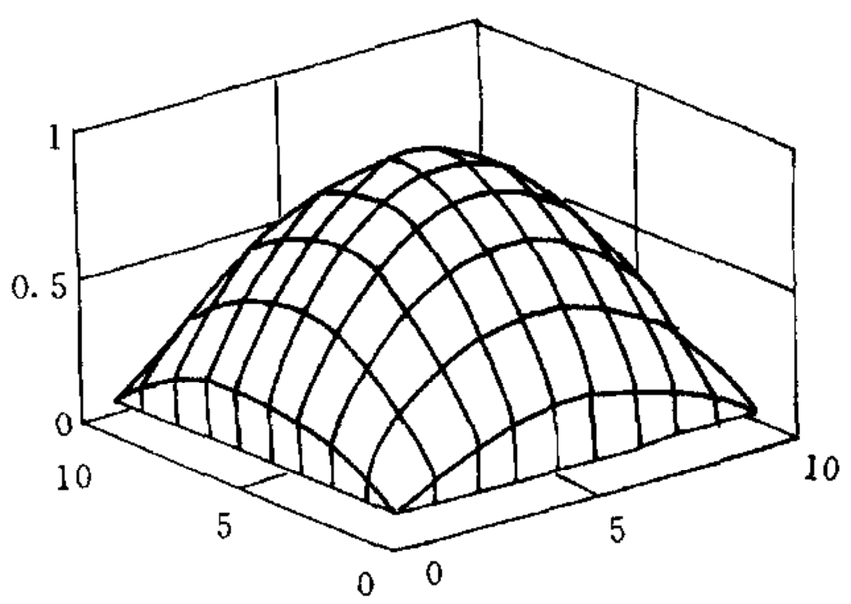
本节给出一个二维随机游动的例子. 一质点在 10×10 的二维格点上以相同的概率向上、下、左、右随机移动,每次一格. 质点所处的格点即为当前状态. 若游动出边界,则收到的增强信号为零;否则为 1. 中间四个格点之一为起始状态. 假设一旦质点游动出边界,则它立即以相同的概率回到四个起始状态中的一个. 用平均准则 TD(λ)学习算法来预报每步平均报酬及每一状态的相对值. 仿真结果见图 1. 其中 $\lambda=0.75$,相对值均初始化为 0,即时差分由(10)式所定义,固定状态 s 为起始状态之一.

(a)迭代次数＝1000

(b)迭代次数＝2000



(c)迭代次数＝10000



(d)真值

图 1 相对值函数曲面

4 结束语

本文给出了一族平均准则 SDP 的在线即时差分学习算法，在折扣及有限阶段随机动态规划问题的即时差分算法中，余留费用函数是学习单元所要学习的目标；而在本文算法中，平均问题的相对值函数是学习单元所要学习的目标函数. 所提出的算法是已有的 TD(λ)算法的一种推广，同时又是在每一状态仅有一个可用行动情形下 R-学习算法的推广. 仿真表明，所给算法能较好地预报其相对值函数，较为适中的 λ 逼近效果最佳.

参 考 文 献

1 Sutton R S. Learning to predict by the methods of temporal differences. *Machine Learning*, 1988, **3**: 9～44

2 Dayan P D. The convergence of TD(λ) for general λ. *Machine Learning*, 1992, **8**: 341～362

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

4 S. Mahadevan S. Average reward reinforcement learning: foundations, algorithms and empirical results *Machine Learning*, 1996, **22**: 159～195

5 Bertsekas D P. Dynamic Programming: Deterministic and Stochastic Methods. Englewood Cliffs, NJ: Prentice-Hall, 1987

胡光华 1962 年生. 1994 年在云南大学获硕士学位，现为北京理工大学自动控制系博士生. 研究领域为人工神经网络技术与智能学习控制.

吴沧浦 简介见本刊第 20 卷第 6 期.