doi:10.3969/j.issn.1000-436x.2013.10.011

基于两层模糊划分的时间差分算法

穆翔[1]，刘全[1,2]，傅启明[1]，孙洪坤[1]，周鑫[1]
（1. 苏州大学 计算机科学与技术学院，江苏 苏州 215006；2. 吉林大学 符号计算与知识工程教育部重点实验室，吉林 长春 130012）

摘 要：针对传统的基于查询表或函数逼近的 Q 值迭代算法在处理连续空间问题时收敛速度慢、且不易求解连续行为策略的问题，提出了一种基于两层模糊划分的在策略时间差分算法——DFP-OPTD，并从理论上分析其收敛性。算法中第一层模糊划分作用于状态空间，第二层模糊划分作用于动作空间，并结合两层模糊划分计算出 Q 值函数。根据所得的 Q 值函数，使用梯度下降方法更新模糊规则中的后件参数。将 DFP-OPTD 应用于经典强化学习问题中，实验结果表明，该算法有较好的收敛性能，且可以求解连续行为策略。

关键词：强化学习；在策略；梯度下降；两层模糊划分；连续行为策略
中图分类号：TP181 **文献标识码**：A **文章编号**：1000-436X(2013)10-0092-08

TD algorithm based on double-layer fuzzy partitioning

MU Xiang[1], LIU Quan[1, 2], FU Qi-ming[1], SUN Hong-kun[1], ZHOU Xin[1]
(1. Institute of Computer Science and Technology, Soochow University, Suzhou 215006, China;
2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract: When dealing with the continuous space problems, the traditional Q-iteration algorithms based on lookup-table or function approximation converge slowly and are diff lt to get a continuous policy. To overcome the above weaknesses, an on-policy TD algorithm named DFP-OPTD was proposed based on double-layer fuzzy partitioning and its convergence was proved. The first layer of fuzzy partitioning was applied for state space, the second layer of fuzzy partitioning was applied for action space, and Q-value functions were computed by the combination of the two layer fuzzy partitioning. Based on the Q-value function, the consequent parameters of fuzzy rules were updated by gradient descent method. Applying DFP-OPTD on two classical reinforcement learning problems, experimental results show that the algorithm not only can be used to get a continuous action policy, but also has a better convergence performance.

Key words: reinforcement learning; on-policy; gradient descent; double layer fuzzy partitioning; continuous action policy

1 引言

强化学习(RL, reinforcement learning)是一种通过 agent 与环境进行交互学习，以获得最大累计奖赏值的机器学习方法[1,2]。通常基于马尔科夫决策过程(MDP, Markov decision process)来定义强化学习问题的一般框架。当强化学习问题满足 MDP 框架时，可以采用诸如动态规划(DP, dynamic programming)、蒙特卡罗(MC, Monte Carlo)和时间差分(TD, temporal difference)等类型的算法求解最优行为策略。

传统的强化学习方法一般用于求解小空间或离散空间的问题[1]。通过查询表(lookup-table)存储所有的状态或者状态动作对所对应的值函数，在学

收稿日期：2013-06-06；**修回日期**：2013-08-20
基金项目：国家自然科学基金资助项目(61070223, 61103045, 61070122, 61272005)；江苏省自然科学基金资助项目(BK2012616)；江苏省高校自然科学研究基金资助项目(09KJA520002, 09KJB520012)；吉林大学符号计算与知识工程教育部重点实验室基金资助项目(93K172012K04)
Foundation Items: The National Natural Science Foundation of China(61070223, 61103045, 61070122, 61272005); The Natural Science Foundation of Jiangsu Province(BK2012616); The High School Natural Foundation of Jiangsu Province(09KJA520002, 09KJB520012); The Foundation of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of Jilin University(93K172012K04)

习过程中不断地修改表项的值直至收敛，最终求得问题的最优行为策略。这类方法虽然能够有效地解决一些简单的任务，但不适用于求解大空间或连续空间的问题。目前解决此类问题最常用的方法是将函数逼近与强化学习算法相结合。通过采用带有一组参数的近似函数来描述强化学习中的值函数，使学习到的经验信息能够从状态空间子集泛化至整个状态空间。Agent 根据此近似函数选择最优动作序列[2~4]。当前已有多种函数逼近方法应用于强化学习问题。SUTTON 等人于 2009 年提出了梯度 TD(GTD, gradient TD)学习算法，该算法将 TD 学习算法与线性函数逼近相结合，同时引入一个与 Bellman 误差相关的新的目标函数[5]。SHERSTOV 等人于 2005 年提出一种基于在线自适应 Tile-Coding 编码的线性函数逼近算法，通过实验验证了算法的有效性[6]。HEINEN 等人于 2010 年提出利用增量式概率神经网络来逼近强化学习问题的值函数，可以较好地求解连续状态空间的问题[7]。

上文所述及目前常见的基于函数逼近的强化学习算法通常收敛速度较慢，而且一般只能用于求解离散行为策略[5~8]。基于模糊推理系统(FIS, fuzzy inference system)的强化学习算法通过引入先验知识，不仅可以有效地加快求解连续空间问题时的收敛速度，还能获得连续行为策略[9,10]。TADASHI 等人提出了模糊插值 Q 学习算法，可以用于求解连续空间问题，但算法的性能较依赖于先验知识[11]。GLORENNEC 和 JOUFFE 将 FIS 与 Q 学习算法相结合，利用先验知识并构造全局近似器，有效地加快了收敛速度，但该算法不能用于求解连续行为策略[12]。TOKARCHUK 等人提出的模糊 Sarsa 算法，在不影响算法性能的情况下可以有效地减小状态空间的规模，进而加快收敛速度，但该算法应用于多维状态空间问题时，更容易出现“维数灾”问题[13]。HSU 等人提出的基于二型模糊逻辑的自组织 Q 学习算法，对于噪声干扰有很强的顽健性，但时间复杂度较高，且不能保证收敛[10]。

虽然基于模糊推理系统的强化学习算法已经可以有效地加快收敛速度，但传统的基于一个模糊规则库的、并可用于求解关于状态的连续行为策略的 Q 值迭代算法，依旧存在由于某些原因而导致收敛速度慢的问题：算法的某一轮迭代会出现状态动作对所对应的 Q 值不唯一的情况。若算法进入下一轮迭代时，需要用到的状态动作对的 Q 值恰好是上述 Q 值不唯一的情况。已有的此类算法会简单地随机选择一个状态动作对所对应的 Q 值，而并没有固定的选择策略，或者固定选择策略也不一定有效。由于算法在整个的迭代过程中会多次出现这种情况，这会较大地减缓该类型算法的收敛速度。

针对传统的基于查询表和一个规则库的 Q 值迭代算法收敛速度慢的问题，本文提出一种基于两层模糊划分的在策略时间差分算法——DFP-OPTD (on-policy TD based on double-layer fuzzy partitioning)，并在理论上证明其收敛。算法在进行 2 次模糊划分时，首先在第一层将连续状态空间进行模糊划分，同时求得连续动作；其次，在第二层将第一层求得的连续动作进行模糊划分，同时求得 Q 值函数；最后，使用梯度下降方法，更新两层模糊划分共同的规则后件参数。将 DFP-OPTD 算法应用于倒立摆问题中，实验结果表明，DFP-OPTD 可以获得连续行为策略，且具有较好的收敛性能。

2　相关理论

2.1　马尔科夫决策过程

在强化学习框架下，agent 与环境交互构成一个有限的 MDP[13]，该 MDP 可描述为一个四元组形式 $M=<X,U,r,f>$，其中：

1) X 为所有状态的集合，且 $x_t \in X$ 为 agent 在 t 时刻所处的状态；

2) U 为所有动作的集合，且 $u_t \in U$ 为 agent 在 t 时刻所采取的动作；

3) $r: X \times U \to ¡^n$ 为奖赏值函数，表示 t 时刻的状态 x_t，在采取动作 u_t 并转移到状态 x_{t+1} 时，agent 所获得的立即奖赏 $r(x_t,u_t)$，此外，用 r_t 表示以 $r(x_t,u_t)$ 为均值的分布所产生的随机奖赏；

4) $f: X \times U \times X \to [0,1]$ 为状态转移函数，其中 $f(x,u,x')$ 表示状态 x 在采取动作 u 时转移到 x' 的概率。

强化学习中的策略 $h(x,u)$ 是从状态空间 X 到动作空间 U 的映射，$h: X \to U$。它表示在状态 x 处选择动作 u 的概率。利用策略 $h(x,u)$ 可以求解出状态值函数(V 值函数)或动作值函数(Q 值函数)。

强化学习的目标是求解最优行为策略 h^*，它是最优值函数的贪心策略，且在所有的策略中满足 $\forall x \in X: V^{h^*}(x) \geq V^h(x)$。在最优策略 h^* 下，最优 V 值函数满足式(1)，最优 Q 值函数满足式(2)，为

$$\forall x\in X: V^*(x)=\max_{u\in U}\left(r(x,u)+g\sum_{x'\in X}f(x,u,x')V^*(x')\right) \quad (1)$$

$$\forall x\in X, u\in h(x): Q^*(x,u)=r(x,u)+g\sum_{x'\in X}f(x,u,x')\max_{u'\in U}Q^*(x',u') \quad (2)$$

当 f 和 r 已知时，可以采用动态规划算法求解最优行为策略；当 f 和 r 未知时，则可以采用 TD 类型的算法求解最优行为策略，例如离策略的 Q 学习算法和在策略(on-policy)的 Sarsa 算法。

定义 1 是一个有界的 MDP 约束(主要是对状态空间、动作空间、奖赏值以及值函数空间的界定)，本文所有的算法都满足该定义。

定义 1　有界的 MDP 问题　已知 X 和 U 都是有限集合，令 Z 表示状态动作集合，即 $Z: X\times U$，则 Z 也为有限集合；奖赏值函数 r 满足 $0\le r(x,u)\le C$；MDP 的边界因子 $b=1/(1-g)$，其中，g 为折扣因子，且对于 $\forall x\in X$ 及 $\forall(x,u)\in Z$，$0\le V(x)\le bC$ 和 $0\le Q(x,u)\le bC$ 成立。

2.2　作为逼近器的模糊规则库

由文献[14]可得，模糊规则库的输出可以用作 Q 值函数的逼近器。当前有多种类型的模糊规则[15]，其中，TSK 形式的规则如式(3)所示，描述了规则的输出和输入部分的关系为

$$\begin{aligned}&Rule\ R_r: \text{if } x_1 \text{ is } c_1^r \text{ AND } \cdots \text{ AND } x_n \text{ is } c_n^r\\ &\text{then } \boldsymbol{y}=g_r(\boldsymbol{x})\end{aligned} \quad (3)$$

其中，$r\in 1,\cdots,N_R$ 是规则的下标，R_r 表示规则库中的第 r 条规则，$\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$ 表示 N 维输入参数。c_i^r 是第 r 条模糊规则中对应于第 i 维输入变量的模糊集，每一个模糊集 c_i^r 都由一个隶属度函数 $m_{c_{r,i}}(x_i): X\to[0,1]$ 定义。y 是输出变量，且 $g_1(x),\cdots,g_{N_R}(x): X\to Y$ 是以 x 为自变量的多项式函数。

当系统输入精确值 $\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$ 时，可以计算它在第 r 条规则下的激活强度 $f_r(x)$ (运算规则为 T-norm 积运算)为

$$f_r(\boldsymbol{x})=m_{C_{r,1}}(x_1)m_{C_{r,2}}(x_2)\cdots m_{C_{r,N}}(x_N), r=1,\cdots,N_R \quad (4)$$

将 $f_r(x)$ 用于计算模糊规则的输出值，以激活强度 $f_r(x)$ 为权重，与其对应的后件值 y_r 相乘并求和，可以得到最终的输出值为

$$Y(\boldsymbol{x})=\frac{\sum\left(f_1(\boldsymbol{x})y_1+f_2(\boldsymbol{x})y_2+\cdots+f_{N_R}(\boldsymbol{x})y_{N_R}\right)}{\sum\left(f_1(\boldsymbol{x})+f_2(\boldsymbol{x})+\cdots+f_{N_R}(\boldsymbol{x})\right)} \quad (5)$$

通常采用 MSE(mean square error)作为模糊规则库用于逼近目标函数时的逼近误差。当规则集合达到最优逼近效果时，其所有模糊规则后件值所构成的向量值 $\boldsymbol{q}$ 为

$$\boldsymbol{q}=\arg\min_{\boldsymbol{q}}\sum_{i=1}^{N}\left(Y_i(\boldsymbol{x})-\hat{Y}_i(\boldsymbol{x})\right)^2 \quad (6)$$

其中，$Y_i(x)$ 为目标函数，$\hat{Y}_i(x)$ 为逼近函数。

3　基于双层模糊划分的在策略 TD 算法

3.1　Q 值函数的计算和参数更新

在 MDP 框架下，使用两层模糊划分相对应的两层模糊规则库以计算 Q 值函数。

使用两层糊规则库逼近 Q 值函数的框架如图 1 所示，其中左框内的模糊规则库 1(FRB1, fuzzy rule base 1)以状态为输入，通过 FRB1 获得的连续动作为输出；右框内的模糊规则库 2 (FRB2, fuzzy rule base 2)以从 FRB1 中获得的连续动作为输入，通过 FRB2 获得的连续动作的 Q 值分量作为输出；最后，通过将两层模糊规则库输出部分相结合，逼近在状态 x 时采取连续动作 $C(x)$ 的 Q 值函数。

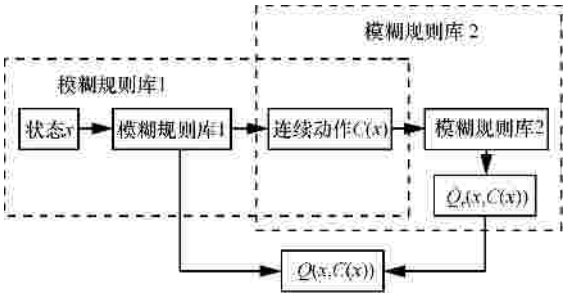

图 1　使用两层模糊规则库逼近 Q 值函数的框架

两层模糊划分的主要内容如下所述。

1) 模糊规则库 1 中的模糊规则如下

$$\begin{aligned}&\text{Rule } R_r: \text{if } x_1 \text{ is } c_1^r \text{ AND } \cdots \text{ AND } x_n \text{ is } c_n^r\\ &\quad\text{then } y=u_{r,1} \text{ with } q_{r,1}=q_{r,1}\\ &\quad\text{or } y=u_{r,2} \text{ with } q_{r,2}=q_{r,2}\\ &\quad\cdots\\ &\quad\text{or } y=u_{r,M} \text{ with } q_{r,M}=q_{r,M};\end{aligned}$$

其中，$\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$ 为状态，$u_{r,j}$ 为第 r 条模糊规则中的第 j 个离散动作。M 个离散动作由动作空间划分而成，$q_{r,j}$ 为第 r 条模糊规则中对应于第 j 个离散动作的 Q 值分量。当输入状态为 x 时，第 r 条规则的激活强度为

$$j_r(\boldsymbol{x}) = m_{c_{r,1}}(x_1)m_{c_{r,2}}(x_2)\cdots m_{c_{r,N}}(x_N) \tag{7}$$

在被状态 x 激活的规则 R_r 中，根据 $q_{r,j}$ 的大小，用 e-greedy 动作选择策略从 M 个离散动作中选出一个动作，该动作称为激活动作，用 $\hat{u}_r$ 表示。因而，结合式(5)，可以得到状态为 x 时的连续动作 $C(x)$ 为

$$C(\boldsymbol{x}) = \frac{\sum_{r=1}^{N_R} j_r(\boldsymbol{x})\hat{u}_r}{\sum_{r=1}^{N_R} j_r(\boldsymbol{x})} \tag{8}$$

把 $C(x)$ 称为连续动作的原因是 $C(x)$ 的变化是关于状态 x 连续的，它并非指的是状态 x 可以选择到连续动作空间中的任意动作。为简化式(8)，正则化激活强度 $j_r(x)$，可得

$$f_r(\boldsymbol{x}) = \frac{j_r(\boldsymbol{x})}{\sum_{r=1}^{N_R} j_r(\boldsymbol{x})} \tag{9}$$

则式(8)可写为

$$C(\boldsymbol{x}) = \sum_{r=1}^{N_R} f_r(\boldsymbol{x})\hat{u}_r \tag{10}$$

2) 模糊规则库 2 中的模糊规则如下

$$\hat{R}_{r,1}: \text{if } u \text{ is } n_{r,1} \text{ then } q_{r,1} = q_{r,1}$$
$$\hat{R}_{r,2}: \text{if } u \text{ is } n_{r,2} \text{ then } q_{r,2} = q_{r,2}$$
$$\cdots$$
$$\hat{R}_{r,M}: \text{if } u \text{ is } n_{r,M} \text{ then } q_{r,M} = q_{r,M}$$

FRB2 中规则的构建依赖于 FRB1，其 M 条规则中的规则 $\hat{R}_{r,j}$ 以 FRB1 中的第 r 条规则为基础：前件部分的 $n_{r,j}$ 为模糊集，它以 FRB1 中第 r 条规则的第 j 个动作为模糊中心，并用隶属度函数 $s_{n_{i,j}}(u)$ 描述；后件部分的 $q_{r,j}$ 与 FRB1 中规则后件的 $q_{r,j}$ 一一对应。

将从 FRB1 中得到的连续动作 $C(x)$ 作为 FRB2 中规则的输入，可以激活 N_R 条 FRB2 中的规则。通过 FRB2 的规则的输出，可以得到 FRB1 中第 r 条规则所对应的 Q 值分量 $\hat{Q}_r(\boldsymbol{x}, C(\boldsymbol{x}))$ 为

$$\hat{Q}_r(\boldsymbol{x}, C(\boldsymbol{x})) = \frac{\sum_{j=1}^{M} s_{n_{r,j}}(C(\boldsymbol{x}))q_{r,j}}{\sum_{j=1}^{M} s_{n_{r,j}}(C(\boldsymbol{x}))} \tag{11}$$

与推导公式(9)的方法相同，正则化式(11)中的隶属度函数 $s_{n_{r,j}}(C(x))$，得到 $m_{n_{r,j}}(C(x))$ 为

$$m_{n_{r,j}}(C(\boldsymbol{x})) = \frac{s_{n_{r,j}}(C(\boldsymbol{x}))}{\sum_{j=1}^{M} s_{n_{r,j}}(C(\boldsymbol{x}))} \tag{12}$$

则式(11)可写为

$$\hat{Q}_r(\boldsymbol{x}, C(\boldsymbol{x})) = \sum_{j=1}^{M} m_{n_{r,j}}(C(\boldsymbol{x}))q_{r,j} \tag{13}$$

由式(13)可得，FRB1 的激活规则 R_r 所求得的 Q 值分量为 $\hat{Q}_r(\boldsymbol{x}, C(\boldsymbol{x}))$，则对 FRB1 中所有的激活规则，可以得到在状态 x 下执行连续动作 $C(x)$ 时的 Q 值为

$$\begin{aligned} Q(\boldsymbol{x}, C(\boldsymbol{x})) &= \sum_{r=1}^{N_R} f_r(\boldsymbol{x})\hat{Q}_r(\boldsymbol{x}, C(\boldsymbol{x})) \\ &= \sum_{r=1}^{N_R}\sum_{j=1}^{M} f_r(\boldsymbol{x})m_{n_{r,j}}(C(\boldsymbol{x}))q_{r,j} \end{aligned} \tag{14}$$

由式(14)可以看出，Q 值的大小取决于两层 FRB 中的模糊集和共同的后件变量 $q_{r,j}$。由于模糊集是作为先验知识提前设定的，且在算法中不做改变，因而要得到收敛的 Q 值，需要在算法执行过程中更新 $q_{r,j}$，直到收敛。

为使 FRB 逼近 Q 值函数时的逼近误差最小，即参数向量 $\boldsymbol{q}$ 满足式(6)，DFP-OPTD 利用梯度下降(GD, gradient descent)方法，结合计算 Q 值函数的 Bellman 方程，更新两层 FRB 的共同后件参数向量 $\boldsymbol{q}$ 为

$$\begin{aligned} \boldsymbol{q}_{t+1} &= \boldsymbol{q}_t - \frac{1}{2}a\nabla_{\boldsymbol{q}_t}[r_{t+1} + gQ_t(\boldsymbol{x}_{t+1}, u_{t+1}) - Q_t(\boldsymbol{x}_t, u_t)]^2 \\ &= \boldsymbol{q}_t + a[r_{t+1} + gQ_t(\boldsymbol{x}_{t+1}, u_{t+1}) - \\ &\quad Q_t(\boldsymbol{x}_t, u_t)]\nabla_{\boldsymbol{q}_t}Q_t(\boldsymbol{x}_t, u_t) \end{aligned} \tag{15}$$

其中，$r_{t+1} + gQ_t(\boldsymbol{x}_{t+1}, u_{t+1}) - Q_t(\boldsymbol{x}_t, u_t)$ 是 TD 误差。令 $d = r_{t+1} + gQ_t(\boldsymbol{x}_{t+1}, u_{t+1}) - Q_t(\boldsymbol{x}_t, u_t)$，结合后向 TD 算法[1]，可以得到参数更新公式为

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + a\,d\nabla_{\boldsymbol{q}_t}Q_t(\boldsymbol{x}_t, u_t) \tag{16}$$

其中，a 是步长参数，$\nabla_{q_{r,j}^t}Q_t(x,u)$ 表示 t 时刻 Q 值函数对参数 $q_{r,j}^t$ 求偏导数之后得到的梯度值[1]，根据式(14)可以求得 $\boldsymbol{q}_t$ 中每一维在 t 时刻的梯度值为

$$\begin{aligned} \nabla_{q_{r,j}^t}Q_t(\boldsymbol{x}, u) &= \nabla_{q_{r,j}^t}\sum_{i=1}^{N_R}\sum_{j=1}^{M} f_r(\boldsymbol{x})m_{n_{r,j}}(u)q_{r,j} \\ &= f_r(\boldsymbol{x})m_{n_{r,j}}(u) \end{aligned} \tag{17}$$

其中，$r = 1, \cdots, N_R$, $j = 1, \cdots, M$。

则式(16)可进一步表示为

$$\boldsymbol{q}_{t+1}=\boldsymbol{q}_t+a\,df_r(\boldsymbol{x})m_{h_{r,j}}(u) \tag{18}$$

3.2 DFP-OPTD 算法的学习过程

基于文献[1]中的在策略 TD 算法，结合本文 3.1 节描述的内容，得到算法 DFP-OPTD。该算法不仅可以解决强化学习中连续状态、离散动作空间的问题，还可以解决连续状态、连续动作空间的问题。算法 1 为 DFP-OPTD 的学习流程。

算法 1 基于双层模糊划分的 DFP-OPTD 算法

1) 初始化参数向量 $\boldsymbol{?}=0$，步长参数 a

2) Repeat(对每一个情节)：

3) $\boldsymbol{x}\leftarrow$ 初始化状态

4) 根据式(7)计算 $f_r(\boldsymbol{x})$

5) 根据 e-greedy 策略选择激活动作 h_r

6) 根据式(10)选择状态为 $\boldsymbol{x}$ 时的执行动作 u

7) 根据式(12)计算 $m_{h_{r,j}}(u)$

8) 根据式(14)计算值函数 Q_u

9) Repeat(对情节中的每一步)

10) 执行动作 u，获得下一状态 x' 和立即奖赏 r

11) $d\leftarrow r-Q_u$

12) 根据 e-greedy 策略选择激活动作 h_r

13) 根据式(10)选择状态为 x' 时的执行动作 u'

14) 根据式(12)计算 $m_{h_{r,j}}(u')$

15) 根据式(7)计算 $f_r(x')$

16) 根据式(14)计算值函数 $Q_{u'}$

17) $d\leftarrow d+gQ_{u'}$

18) $\boldsymbol{q}=\boldsymbol{q}+a\,df_r(\boldsymbol{x})m_{h_{r,j}}(u)$

19) $u\leftarrow u'$

20) Until x' 为终止状态

21) Until 运行完设定情节数目或满足其他终止条件

3.3 算法收敛性分析

在文献[16]和文献[17]中，针对在策略(on-policy)TD 算法在使用线性函数逼近时的收敛性做了详细的分析，当该类型的算法满足一定的假设和引理时，可以以 1 的概率收敛。DFP-OPTD 正是一种使用线性函数逼近的在策略 TD 算法，当该算法满足文献[16]中定义的证明算法收敛所需的假设和引理时，即可说明其收敛。本文不再赘述对其收敛性的详细证明。

假设 1 MDP 中的状态转移函数和奖赏函数都服从稳定的分布。

引理 1 DFP-OPTD 依赖的马尔科夫链具有不可约性和非周期性，且算法的立即奖赏和值函数有界。

证明 首先证明其不可约性。根据马尔科夫过程的性质，如果一个马尔科夫过程的任意 2 个状态可以相互转移，则它具有不可约性[18]。DFP-OPTD 用于解决满足 MDP 框架的强化学习问题，且该 MDP 满足定义 1。因而对于该 MDP 中的任意状态 x，必定存在一个 f 满足 $f(x,u,x')\geq 0$，这表明状态 x 可以被无限次访问。因而可得每一个状态都可转移到任意的其他状态。因此，DFP-OPTD 依赖的马尔科夫链具有不可约性。

其次证明其非周期性。对于不可约的马尔科夫链，仅需证明某一个状态具有非周期性，即可证明整个马尔科夫链具有非周期性。而证明一个状态具有非周期性，只需证明该状态具有自回归性[18]。在 DFP-OPTD 依赖的 MDP 中，对于状态 x，必定存在一个 f 满足 $f(x,u,x)>0$，它表明了状态 x 具有自回归性，由此可得该 MDP 具有非周期性。因此，DFP-OPTD 依赖的马尔科夫链的非周期性得证。

最后证明其立即奖赏和值函数有界。由文献[1]可知，值函数是折扣的累计回报函数，即满足 $Q(x,u)=\sum_{i=0}^{\infty}g^i r(x,u), g\in(0,1)$。又由定义 1 可得，奖赏值函数 r 有界，且 $0\leq r(x,u)\leq C$，C 为一个非负数。因而有

$$Q(x,u)=\sum_{i=0}^{\infty}g^i r(x,u)<\sum_{i=0}^{\infty}g^i C=\lim_{i\to\infty}\frac{(1-g^i)}{1-g}C=\frac{C}{1-g} \tag{19}$$

由不等式(19)可以得出，值函数 $Q(x,u)$ 有界。

综上所述，引理 1 得证。

条件 1 对每一个隶属度函数 i 都存在唯一的状态 x_i，使 $m_i(x_i)>m_i(x),\forall x\neq x_i$，而其他的隶属度函数在状态 x_i 处的隶属度值都为 0，即有 $m_{i'}(x_i)=0,\ \forall i'\neq i$。

引理 2 DFP-OPTD 的基函数有界，并且基函数向量线性无关。

证明 首先证明其基函数有界。由 $f_r(\boldsymbol{x})\in[0,1]$ 和 $m_{h_{r,j}}(C(\boldsymbol{x}))\in[0,1]$ 可得

$$\|f_r(\boldsymbol{x})m_{h_{r,j}}(C(\boldsymbol{x}))\|_\infty\leq 1 \tag{20}$$

其中，$\|\ \|_\infty$ 为无穷范式。已知 DFP-OPTD 的基函数为 $f_r(\boldsymbol{x})m_{h_{r,j}}(C(\boldsymbol{x}))$，又由不等式(20)可得，DFP-

OPTD 的基函数有界。

其次证明基函数向量线性无关。为使 DFP-OPTD 的基函数向量线性无关，令算法所使用的基函数满足条件 1[14]，其函数形式如图 3 所示。由文献[14]可得，当满足条件 1 时，基函数向量线性无关。

可以将条件 1 的要求适当地放宽，使 $m_{i,}(x_i)$ 在状态 x_i 处的隶属度为一个较小的值，例如标准差较小的高斯隶属度函数。将该隶属度函数用于 DFP-OPTD 中，通过数次实验可得 DFP-OPTD 同样可以收敛，但目前还不能对该收敛性给出理论的证明。

综上所述，引理 2 得证。

引理 3　DFP-OPTD 的步长参数 a 满足

$$\sum_{t=0}^{\infty} a_t = \infty,\ \sum_{t=0}^{\infty} a_t^2 < \infty \tag{21}$$

证明　DFP-OPTD 所用的步长参数 $a = 1/(t+1)$，其中，t 为时间步。使用牛顿幂级数展开 $\sum_{t=0}^{\infty} a_t$ 可以得到

$$\sum_{t=0}^{\infty} a_t = \sum_{t=0}^{\infty}\left(1+1/2+\cdots+1/t\right) = \ln(t+1)+r \tag{22}$$

其中，$r \approx 0.577\,218$ 为欧拉常数。又因为 $\ln t$ 为递增函数，所以当 $t \to \infty$ 时，满足 $\sum_{t=0}^{\infty} a_t = \infty$。

$$\sum_{t=0}^{\infty} a_t^2 = \sum_{t=0}^{\infty}\left(1^2+(1/2)^2+\cdots+(1/t)^2\right) < (2t-1)/t = 2-1/t \tag{23}$$

不等式(23)中的不等式部分可通过归纳法证明，因而当 $t \to \infty$ 时，满足 $\sum_{t=0}^{\infty} a_t^2 < \infty$。

由式(22)和不等式(23)可以得出，DFP-OPTD 所用的步长参数满足式(21)，即引理 3 得证。

定理 1　在假设 1 的条件下，若 DFP-OPTD 满足引理 1~引理 3，则算法以 1 的概率收敛。

证明　由文献[16]可以得出，在假设 1 成立的条件下，在策略(on-policy)TD 算法在使用线性函数逼近时，如果满足引理 1~引理 3，该类型的算法收敛。满足假设 1 的算法 DFP-OPTD 是一种利用线性函数逼近的在策略 TD 算法，且该算法对引理 1~引理 3 成立。因而可以得出，DFP-OPTD 以 1 的概率收敛。

4　实验结果及分析

本文以强化学习中经典的情节式问题——倒立摆问题为例，验证 DFP-OPTD 的收敛性能和求得的连续行为策略的作用。

倒立摆问题的示意如图 2 所示，一个可以左右移动的小车位于水平面上，上面放置一根底端与小车相连且可以在一定角度范围内自由转动的硬质杆，其任务是通过小车的水平移动使硬质杆可以在一定的角度范围内($[-\pi/2, \pi/2]$)竖立于垂直方向。同样将该问题建立为一个 MDP 模型：系统的状态是 1 个二维变量，用硬质杆与垂直方向的夹角 q 和硬质杆的角速度 $\dot{q}$ 表示，即 $x = [q, \dot{q}]$，且有 $q \in [-\pi/2, \pi/2](\text{rad})$ 和 $\dot{q} \in [-16\pi, 16\pi](\text{rad/s})$；系统的动作为施加在小车上的力，其取值范围为 $[-50, 50](N)$。此外，施加的力上有外力的随机扰动，该外力服从 $[-10, 10](N)$ 的均匀分布。系统的动力学特性描述为

$$\ddot{q} = \frac{g\sin(q) - aml(\dot{q})^2\sin(2q)/2 - a\cos(q)u}{4l/3 - aml\cos^2(q)} \tag{24}$$

其中，$g = 9.8\ \text{m/s}^2$ 为重力加速度，$m = 2.0\ \text{kg}$ 为硬质杆的质量，$M = 8.0\ \text{kg}$ 为小车的质量，$l = 0.5\ \text{m}$ 为硬质杆的长度，常数 $a = 1/(m+M)$。系统的奖赏变化取决于状态的变化，在每一个时间步下，当硬质杆与垂直方向的角度不超过 $\pi/2$ 时，会收到大小为 0 的立即奖赏。而超过 $\pi/2$ 时收到的立即奖赏为 -1，同时该情节结束。

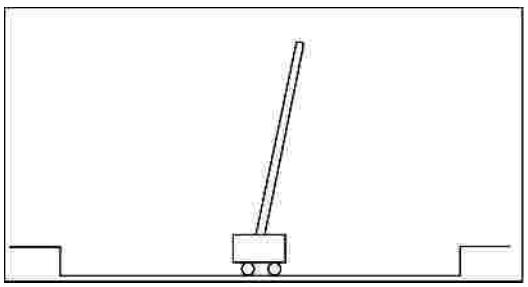

图 2　倒立摆

将 DFP-OPTD 算法与 SUTTON 等人提出的 GD-Sarsa(?)算法[3]进行比较。设置 DFP-OPTD 所需的参数，用三角隶属度函数作为 FRB1 和 FRB2 的模糊集的隶属度函数式(除了状态的定义域不同，夹角和角速度的模糊隶属度函数形式如图 3 所示)：分别采用 20 个模糊中心等距的模糊集对二维的连续状态空间的每一维进行三角模糊划分，模糊集的个数为 20×20=400；同理，用 12 个模糊中心等距的模糊集对连续动作空间进行三角模糊划分，模糊集的个数为 12。其他参数设置为 $e = 0.001$，$a = 0.9$，

$g=1.0$。GD-Sarsa(?)中采用 10 个 9×9 的 Tilings 来划分状态空间，参数设置依据文献[1]中给出的最优实验参数：$e=0.001$，$a=0.14$，$l=0.3$，$g=1.0$。

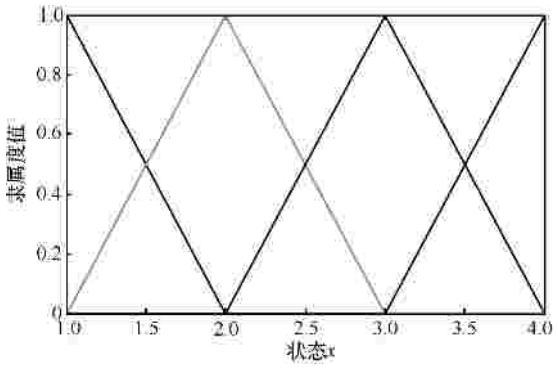

图 3 三角隶属度函数

DFP-OPTD，GD-Sarsa(?)针对倒立摆问题进行 30 次独立仿真实验的结果如图 4 所示，图中横坐标表示情节数，纵坐标表示硬质杆竖立于垂直方向及两侧的一定角度范围内所用的平均时间步。分析图 4 可得，DFP-OPTD 在收敛性能上明显优于 GD-Sarsa(?)。

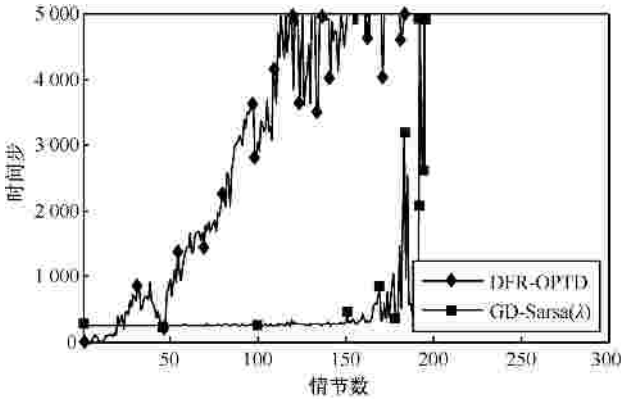

图 4 2 种算法收敛性能的比较

2 种算法的详细性能比较如表 1 所示，其中，以 DFP-OPTD 的一个平均迭代步所需的时间作为基准时间。

表 1 2 种算法在倒立摆问题中性能的比较

算法	算法收敛所需情节数		算法一个迭代步的平均时间
	最小情节数	平均情节数	
DFP-OPTD	142	155	100%
GD-Sarsa(?)	179	204	49%

图 5 描述的分别为 DFP-OPTD 和 GD-Sarsa(?)这 2 种算法在时间步增大的过程中，硬质杆与垂直方向的角度变化情况。其中，GD-Sarsa(?)基于离散动作，DFP-OPTD 基于连续动作。从图中可以清晰地看出，DFP-OPTD 所获得的连续行为策略可以使硬质杆摆动的角度只在较小的范围内变化，而 GD-Sarsa(?)所获得的离散行为策略会使硬质杆在较大的角度范围内摆动，这说明了 DFP-OPTD 求得的策略的稳定性优于 GD-Sarsa(?)。因而，DFP-OPTD 更适用于求解对策略稳定性要求较高的问题。

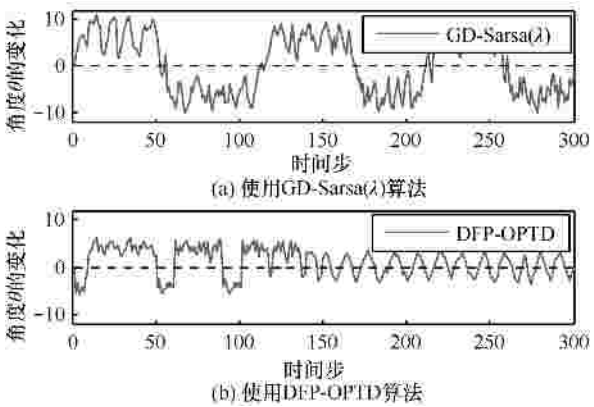

图 5 分别使用上述 2 种算法时，硬质杆的角度 ? 的变化情况

5 结束语

本文针对传统的强化学习算法中使用查询表或者函数逼近时收敛速度慢且不易获得连续行为策略的问题，提出一种基于两层模糊划分的强化学习算法——DFP-OPTD。该算法先将状态进行模糊划分，再将第一层模糊规则库所输出的连续动作，作为第二层模糊规则库的输入，同时对动作进行模糊划分。最后将这两层模糊规则库相结合以得到逼近的 Q 值函数。以该逼近的 Q 值函数与真实 Q 值函数的差值平方作为逼近误差，使用梯度下降方法更新 2 个模糊规则库中规则的共同后件值。将该算法与其他 3 种较新的相近算法应用于强化学习中经典的倒立摆问题中，通过实验数据分析可以得到，相比于已有的只使用一层模糊划分的强化学习算法，DFP-OPTD 虽然增加了时间复杂度，但需要较少的收敛步数。相比于基于查询表或者其他的函数逼近方法，DFP-OPTD 有更好的收敛性能，且可以获得连续行为策略。

DFP-OPTD 的性能主要依赖于两层模糊划分，而模糊规则库的逼近性能主要取决于模糊集的隶属度函数和模糊规则的个数。本文将隶属度函数和规则个数作为先验知识给出，且在算法执行过程中不做改变。为了获得更好的收敛性能，下一步将考虑使用合适的优化算法，使 DFP-OPTD 能在运行的过程中不断优化隶属度函数，并且能够自适应地调

整模糊规则的条数。

参考文献：

[1] SUTTON R S, BARTO A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 1998.

[2] **刘全, 闫其粹, 伏玉琛等. 一种基于启发式奖赏函数的分层强化学习方法**[J]. **计算机研究与发展**, 2011, 48(12): 2352-2358.
LIU Q, YAN Q C, FU Y C, *et al*. A hierarchical reinforcement learning method based on heuristic reward function[J]. Journal of Computer Research and Development, 2011, 48(12): 2352-2358.

[3] SUTTON R S, MCALLESTER D, SINGH S, *et al*. Policy gradient methods for reinforcement learning with function approximation[A]. Proc of the 16th Annual Conference on Neural Information Processing Systems[C]. Denver, 1999. 1057-1063.

[4] MAEI H R, SUTTON R S. GQ(*?*): a general gradient algorithm for temporal difference prediction learning with eligibili y traces[A]. International Conference on Artificial General Intelligence[C]. Lugano, 2010. 91-96.

[5] SUTTON R S, SZEPESV´ARI CS, MAEI H R. A convergent O(*n*) algorithm for off-policy temporal-difference learning with linear function approximation[A]. Proc of the 22nd Annual Confere n Neural Information Processing Systems[C]. Vancouver, 2009. 1609-1616.

[6] SHERSTOV A A, STONE P. Function approximation via tile coding: automating parameter choice[A]. Proc of the 5th Sympos um on Abstraction, Reformulation and Approximation[C]. New York, USA, 2005. 194-205.

[7] HEINEN M R, ENGEL P M. An incremental probabilistic neural network for regression and reinforcement learning tasks[A]. Proc of the 20th International Conference on Artificial Neural Networks[C]. Berlin, 2010. 170-179.

[8] PAZIS J, LAGOUDAKIS M G. Learning continuous-action control policies[A]. Proc of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning[C]. Washington, 2009. 169-176.

[9] BONARINI A, LAZARIC A, MONTRONE F, *et al*. Reinforcement distribution in fuzzy Q-learning[J]. Fuzzy Sets and Systems, 2009, 160(10):1420-1443.

[10] HSU C H, JUANG C F. Self-organizing interval type-2 fuzzy Q-learning for reinforcement fuzzy control[A]. Proc of the 2011 IEEE International Conference on Systems, Man, and Cybernetics[C]. New Jersey, 2011. 2033-2038.

[11] TADASHI H, AKINORI F, OSAMU, *et al*. Fuzzy interpolation-based Q-learning with continuous states and actions[A]. Proc of the Fifth IEEE International Conference on Fuzzy Systems[C]. New York, USA, 2011.594-600.

[12] GLORENNEC P Y, JOUFFE L. Fuzzy Q-learning[A]. Proc of the Sixth IEEE International Conference on Fuzzy Systems[C]. Cambridge, 1997.659-662.

[13] CHANG H S, FU M C, HU J, *et al*. Simulation-based Algorithms for Markov Decision Processes[M]. New York: Springer, 2007.

[14] LUCIAN B, ROBERT B, BART D S, *et al*. Reinforcement Learning and Dynamic Programming Using Function Approximation[M]. Florida: CRC Press, 2010.

[15] CASTILLO O, MELIN P. Type-2 Fuzzy Logic: Theory and Applications[M]. New York: Springer, 2008.

[16] TSITSIKLIS J N, ROY V B. An analysis of temporal-difference learning with function approximation[J]. IEEE Transactions Automatic Control, 1997, 42(5):674-690.

[17] DAYAN P D. The convergence of TD(*?*) for general *?*[J]. Machine Learning, 1992, 8(3-4):341-362.

[18] **刘次华. 随机过程**[M]. **武汉: 华中科技大学出版社**, 2008.
LIU C H. Stochastic Process[M]. Wuhan: Huazhong University o Science and Technology Press, 2008.

作者简介：

穆翔（1988-），男，江苏东海人，苏州大学硕士生，主要研究方向为强化学习。

刘全（1969-），男，内蒙古牙克石人，苏州大学教授、博士生导师，主要研究方向为强化学习、智能信息处理和自动推理。

傅启明（1985-），男，江苏淮安人，苏州大学博士生，主要研究方向为强化学习、贝叶斯推理和遗传算法。

孙洪坤（1988-），男，江苏淮安人，苏州大学硕士生，主要研究方向为强化学习。

周鑫（1989-），男，山西运城人，苏州大学硕士生，主要研究方向为强化学习。