文章编号: 1001-0920 (2014) 12-2121-08 DOI: 10.13195/j.kzyjc.2013.1467

连续空间增量最近邻时域差分学习

张春元[1,2], 朱清新[1], 钟 声[2]

(1. 电子科技大学 计算机科学与工程学院，成都 611731；2. 海南大学 信息科学技术学院，海口 570228)

摘 要: 针对连续空间强化学习问题, 提出一种基于局部加权学习的增量最近邻时域差分 (TD) 学习框架. 通过增量方式在线选取部分已观测状态构建实例词典, 采用新观测状态的范围最近邻实例逼近其值函数与策略, 并结合 TD 算法对词典中各实例的值函数和资格迹迭代更新. 就框架各主要组成部分给出多种设计方案, 并对其收敛性进行理论分析. 对 24 种方案组合进行仿真验证的实验结果表明, SNDN 组合具有较好的学习性能和计算效率.

关键词: 时域差分学习；值函数逼近；策略逼近；局部加权学习

中图分类号: TP18 **文献标志码:** A

Temporal difference learning with incremental nearest neighbors in continuous spaces

ZHANG Chun-yuan[1,2], *ZHU Qing-xin*[1], *ZHONG Sheng*[2]

(1. School of Computer Science and Engineering，University of Electronic Science and Technology of China，Chengdu 611731，China；2. College of Information Science and Technology，Hainan University，Haikou 570228，China. Correspondent：ZHANG Chun-yuan，E-mail：zhangcy@hainu.edu.cn)

Abstract: Based on locally weighted learning, a temporal difference(TD) learning framework with incremental nearest neighbors is proposed for reinforcement learning problems in continuous spaces. The framework selects some observed states to construct an instance dictionary in increments, uses the range nearest neighbor instances of the new observed state to approximate its value function and policy, and combines with a TD algorithm to update the value function and eligibility trace of each instance in the dictionary iteratively. Some schemes are designed for each key component of the framework, and theoretical analyses are given for its convergence. Finally, twenty-four scheme combinations are verified by simulations, which show that the combination SNDN has better learning performance and computational efficiency.

Key words: temporal difference learning；value function approximation；policy approximation；locally weighted learning

0 引 言

在实际工程应用中, 许多强化学习问题具有大规模离散或连续的状态与动作空间, 传统强化学习算法在求解此类问题时往往存在维数灾难, 即难以采用表格形式存储和迭代所有状态值函数 $V(s)$ 或动作值函数 $Q(s,a)$. 因此, 必须对传统强化学习算法加以改造, 以提升其泛化能力, 即利用有限的学习经验和记忆实现对一个大范围空间知识的有效获取和表示[1]. 值函数逼近作为解决这一问题的有效手段之一, 近年来一直是强化学习领域的研究热点.

根据逼近器的结构特性, 值函数逼近可分为线性和非线性逼近两类. 虽然前者的逼近精度从理论上讲不如后者, 但是更易理解且收敛性具有更好的保证, 因此应用更为广泛[2]. 根据逼近器的空间特性, 值函数逼近又可分为全局和局部逼近两类, 前者 (如 LSTD、GPTD 等算法的逼近) 建立在整个输入空间上, 后者通常建立在当前状态的近邻输入空间上, 相较前者, 后者更加稳定且更服从形式化分析[3].

近年来, 国内外针对值函数局部线性逼近的研究已取得一些成果, 同时也存在一些问题. Sutton[4] 采用 CMAC 网络对连续状态空间均匀离散化, 以实现 Sarsa(λ) 算法的值函数逼近. Martin 等[5]提出了 kNN-TD、kNN-TD(λ) 和 Ex$\langle a\rangle$(λ) 算法, 预先将连续状态与动作空间均匀离散化成若干实例, 然后基于 k 个

收稿日期: 2013-10-23；**修回日期:** 2014-02-23.

基金项目: 国家自然科学基金项目(61100118, 60671033)；海南省自然科学基金项目(613153).

作者简介: 张春元(1973−), 男, 副教授, 博士生, 从事强化学习、信息检索的研究；朱清新(1954−), 男, 教授, 博士生导师, 从事计算运算学、生物信息学等研究.

最近邻实例采用距离加权平均逼近值函数和策略. 这两类算法均需手工调优均匀离散化的颗粒度, 对于高维空间仍面临维数灾难问题. Smart 等[6]采用范围最近邻和局部加权回归实现值函数逼近, 提出了 HEDGER 算法. Ormoneit 等[7]基于近邻核平均进行值函数逼近, 并给出了算法的一致收敛条件. HEDGER 算法需不断在线收集当前观测样本作为学习实例, 近邻核平均算法则需要预先离线收集一批样本实例, 两种算法均没有考虑样本实例的在线稀疏化问题. Ratitcht 等[8]采用稀疏分布记忆模型 (SDM) 在线存储部分已观测样本, 提出了一种值函数逼近框架. Lee 等[9]采用增量接收域加权逼近作为 Actor-critic 的值函数评价器. 陈兴国等[10]提出了一种基于在线稀疏化词典和选择性核方法的值函数逼近算法 OSKTD. 这 3 类算法考虑了样本实例的在线稀疏化问题, 但 SDM 和接收域算法学习效果并不理想, OSKTD 的应用仅限于离散动作空间.

鉴于此, 本文提出一种基于增量最近邻实例加权学习实现值函数与策略逼近的 TD 学习框架, 用于求取 Agent 在连续状态与连续动作空间或连续状态与离散动作空间中的最优策略. 分别就实例词典的在线稀疏化构建、当前观测状态的值函数和策略逼近、实例的值函数与资格迹的迭代更新等框架的主要组成部分给出多种设计方案, 并从理论上分析框架的时间复杂度和收敛性. 最后, 通过两种小车爬山问题, 对比 kNN-TD(λ) 和 Ex$\langle a\rangle$(λ) 算法[6], 对 Sarsa(λ) 类型框架的 24 种方案组合进行有效性验证.

1 线性 TD 学习

TD 算法是一类基于 TD 误差的 model-free 强化学习方法, 主要包括 TD(0)、Q-learning、Sarsa 和 Actor-critic 等类型[11]. 在实际应用中, TD 算法常与资格迹相结合以进一步提高算法的学习效率. 为了后续行文的需要, 仅对线性 Sarsa(λ) 和 Actor-critic 加以简述.

1.1 线性 Sarsa(λ) 算法

记状态和动作空间为 S 和 A, 在线性 Sarsa(λ) 中, 对于 $\forall s\in S$, $\forall a\in A$, 其值函数逼近形式为

$$\widetilde{Q}_t(s,a)=\boldsymbol{w}_t\boldsymbol{\phi}^{\mathrm{T}}(s,a)=\sum_{i=1}^{n}w_i\phi_i(s,a). \tag{1}$$

其中: $\boldsymbol{\phi}(s,a)=[\phi_1(s,a),\phi_2(s,a),\cdots,\phi_n(s,a)]$ 为基向量, 对于局部线性 Sarsa(λ), 大部分基函数取值为 0; $\boldsymbol{w}_t=[w_1,w_2,\cdots,w_n]$ 为参数向量, 更新方式为

$$\boldsymbol{w}_{t+1}=\boldsymbol{w}_t+\alpha_t\delta_t\boldsymbol{e}_t, \tag{2}$$

α_t 为学习率, δ_t 为 TD 误差, 计算公式为

$$\delta_t=r_t+\gamma\widetilde{Q}_t(s_{t+1},a_{t+1})-\widetilde{Q}_t(s_t,a_t), \tag{3}$$

r_t 为 Agent 在观测状态 s_t 下执行动作 a_t 转入观测状态 s_{t+1} 获得的奖励值, γ 为折扣因子.

式 (2) 中, $\boldsymbol{e}_t=[e_1,e_2,\cdots,e_n]$ 为资格迹向量, 有累加迹和替代迹两种更新方式[11], 累加迹更新方式为

$$\boldsymbol{e}_{t+1}=\gamma\lambda\boldsymbol{e}_t+\boldsymbol{\phi}(s_{t+1},a_{t+1}), \tag{4}$$

替代迹更新方式[12]为

$$\boldsymbol{e}_{t+1}=\max(\gamma\lambda\boldsymbol{e}_t,\ \boldsymbol{\phi}(s_{t+1},a_{t+1})). \tag{5}$$

1.2 Actor-critic 算法

Actor-critic 由 Actor 和 Critic 两部分组成: 前者负责策略改进, 通常基于标准或自然策略梯度下降实现; 后者负责策略评估, 可以通过各种 TD 算法完成[11,13]. 这种 Actor 和 Critic 相分离的结构, 使得其适合处理连续动作空间强化学习问题.

对于大规模离散或连续动作空间, Actor 的策略函数也可以通过参数化形式定义. 随机策略定义为 π: $S\times A\times\Psi\to[0,1]$, 即 $\pi_t(s_t,a_t,\boldsymbol{\psi}_t)$ 为 Actor 在 s_t 和给定参数向量 $\boldsymbol{\psi}_t$ 下 a_t 的选取概率. 确定性策略定义为 $\pi: S\times\Psi\to A$, 即 $a_t=\pi_t(s_t,\boldsymbol{\psi}_t)$. 类似式 (2), 确定性策略的线性逼近形式[13]可以写为

$$\widetilde{\pi}_t(s)=\boldsymbol{\psi}_t\boldsymbol{\phi}^{\mathrm{T}}(s)=\sum_{i=1}^{n}\psi_i\phi_i(s). \tag{6}$$

$\boldsymbol{\psi}_t$ 基于标准策略梯度下降的更新方式[2]为

$$\boldsymbol{\psi}_{t+1}=\boldsymbol{\psi}_t+\beta_t\nabla_{\boldsymbol{\psi}_t}V_t^{\pi_t}(s_t). \tag{7}$$

当 Actor 采用 ε-greedy 或 Softmax 策略时, 策略由 Critic 间接确定, Actor-critic 退化为 Critic-only 方法.

2 局部加权学习

局部加权学习 (LWL) 是一类基于查询点 q 的近邻样本预测其输出值 $f(q)$ 的消极学习方法, 主要包括最近邻、加权平均和局部加权回归 (LWR) 等类型[14]. 其中加权平均算法通过最小化 $f(q)$ 的预测值与 q 的各近邻样本输出值之间的加权平方误差求取 q 的最优预测值 $\widetilde{f}(q)$. 记 q 的近邻样本为 $x_1,x_2,\cdots,x_m$, 对应的输出值为 $f(x_1),f(x_2),\cdots,f(x_m)$, 有

$$\widetilde{f}(q)=\sum_{i=1}^{m}f(x_i)\varphi(q,x_i), \tag{8}$$

其中 $\varphi(q,x_i)$ 为加权函数, 通常采用归一化距离加权函数形式进行定义. 此时, 加权平均算法称为距离加权平均 (DWA) 算法[14], 即

$$\varphi(q,x_i)=k(d(q,x_i))\Big/\sum_{j=1}^{m}k(d(q,x_j)). \tag{9}$$

其中: $d(\cdot,\cdot)$ 为距离函数, 常用的度量有 Euclidean、Minkowski 等; $k(\cdot)$ 为距离加权函数, 常用形式有逆距离、高斯核、指数核、二次方核、均匀加权核等[14].

3 增量最近邻 TD 学习

值函数局部线性逼近算法大多具有模型简单、计算效率高和易于实现等优点, 但仍存在一些不足. 为此, 针对连续状态与离散动作空间、连续状态与

连续动作空间两类强化学习问题, 采用Critic-only结构, 提出一种基于LWL的增量最近邻TD学习框架INNTD, 并就框架各主要组成部分给出多种设计方案.

3.1 增量最近邻 TD 学习框架

连续空间的有效表示是强化学习领域的一个开放问题, 主要有离散化、聚类、在线构建实例词典等方式[2]. 为了使框架能同时用于离散和连续动作空间, 将连续动作空间均匀离散化成若干个点值表示, 两种空间统一记为 $A=\{a_1,a_2,\cdots,a_m\}$; 连续状态空间通过在线选取部分观测状态构建实例词典来表示, 记为 $D_t=\{x_1,x_2,\cdots,x_{|D_t|}\}$. 为了防止因实例间距离过小导致框架泛化能力下降[10]和词典规模过大, 定义在线稀疏化条件 SIF(s) 以判断当前观测状态 s 是否可以加入词典. 若 SIF(s) 为真, 则令 $ch(\cdot)=\cdot$, 否则 $ch(\cdot)$ 为空. 词典的在线稀疏化构建过程可表示为

$$D_{t+1}=D_t\bigcup\{ch(s)\}. \tag{10}$$

基于上述空间表示, 结合词典中的实例定义基向量 $\boldsymbol{\phi}_t(s)=[\phi(s,x_1),\phi(s,x_2),\cdots,\phi(s,x_{|D_t|})]$, 对于 $\forall a_j\in A$, 式 (1) 可重新定义为

$$\widetilde{Q}_t(s,a_j)=\boldsymbol{w}_t^{a_j}\boldsymbol{\phi}_t^{\mathrm{T}}(s)=\sum_{i=1}^{|D_t|}w_{ji}\phi(s,x_i), \tag{11}$$

其中 $\boldsymbol{w}_t^{a_j}$ 为参数矩阵 $\boldsymbol{W}_t$ 的第 j 行向量, 表示 a_j 下各实例的参数. 参照式 (2), $\boldsymbol{W}_t$ 迭代更新方式为

$$\boldsymbol{W}_{t+1}=\boldsymbol{W}_t+\alpha_t\delta_t\boldsymbol{E}_t, \tag{12}$$

其中 $\boldsymbol{E}_t$ 为资格迹矩阵, 其迭代更新方式将在第 3.4 节给出. 类似地, 对于连续动作空间, 式 (6) 可以改写为

$$\widetilde{\pi}_t(s)=\boldsymbol{\psi}_t\boldsymbol{\phi}^{\mathrm{T}}(s)=\sum_{i=1}^{|D_t|}\psi_i\phi(s,x_i). \tag{13}$$

在强化学习中, 相邻状态的值函数和策略通常具有较高的相似性, 因此可以基于LWL通过 s 的近邻实例实现其值函数和策略逼近. 记 s 在 D_t 的 θ 范围最近邻集为 D_θ^s, 加权向量 $\boldsymbol{\varphi}(s)=[\varphi(s,s_1),\varphi(s,s_2),\cdots,\varphi(s,x_{|D_\theta^s|})]$, 基于加权平均算法, $\widetilde{Q}_t(s,a_j)$ 和 $\widetilde{\pi}_t(s)$ 可以表示为

$$\widetilde{Q}_t(s,a_j)=\sum_{i=1}^{|D_\theta^s|}Q_t(x_i,a_j)\varphi(s,x_i), \tag{14}$$

$$\widetilde{\pi}_t(s)=\sum_{i=1}^{|D_\theta^s|}\pi_t(x_i)\varphi(s,x_i). \tag{15}$$

如果式 (11) 和 (13) 中的基函数按如下方式取值:

$$\phi(s,x_i)=\begin{cases}\varphi(s,x_i),\ \forall x_i\in D_\theta^s;\\ 0,\ \forall x_i\in\overline{D_\theta^s}.\end{cases} \tag{16}$$

则对比式 (11) 与 (14)、(13) 与 (15) 不难发现, 词典中各实例对应的值函数和策略可看作是式 (11) 和 (13) 中的参数, 因此可以分别运用式 (12) 和 (7) 进行迭代更新. 为了保证框架对离散与连续动作空间具有较好的兼容性, 统一选用 ε-greedy 或 Softmax 策略, 这样, 对连续动作空间问题无需采用式 (7) 进行迭代更新.

对于连续动作空间, 由于 $\widetilde{\pi}_t(s)$ 由式 (15) 逼近获得, 一般而言, $\widetilde{\pi}_t(s)\notin A$, $\widetilde{Q}_t(s,a)$ 不能直接采用式 (14) 进行逼近. 由式 (15) 可知, $\widetilde{\pi}_t(s)$ 建立在 $A_\theta^s=\{\pi_t(x_i)|\ x_i\in D_\theta^s\}$ 上, 因此 $\widetilde{Q}_t(s,a)$ 可以定义为

$$\widetilde{Q}_t(s,a)=\sum_{i=1}^{|D_\theta^s|}Q_t(x_i,\pi_t(x_i))\varphi(s,x_i). \tag{17}$$

由于词典采用增量方式构建, $\boldsymbol{W}_t$ 和 $\boldsymbol{E}_t$ 除迭代更新外, 还需增量更新, 二者的增量更新过程为

$$\boldsymbol{W}_{t+1}=[\boldsymbol{W}_{t+1}\ \ ch(\widetilde{\boldsymbol{Q}}_t(s,A))], \tag{18}$$

$$\boldsymbol{E}_{t+1}=[\boldsymbol{E}_{t+1}\ \ ch(\boldsymbol{0}_{m\times1}))]. \tag{19}$$

其中: $\widetilde{\boldsymbol{Q}}_t(s,A)=[\widetilde{Q}_t(s,a_1),\widetilde{Q}_t(s,a_2),\cdots,\widetilde{Q}_t(s,a_m)]^{\mathrm{T}}$, 各元素由式 (14) 求取; $\boldsymbol{0}_{m\times1}$ 为 m 行 1 列零向量.

根据以上分析, 结合线性 Sarsa(λ), 给出增量最近邻TD学习框架INNTD.

Step 1: 令 $L=1$, $t=0$, $A=\{a_1,a_2,\cdots,a_m\}$, 初始化各参数和 s_t, 置 $D_t=\{s_t\}$, $\boldsymbol{W}_t=\boldsymbol{E}_t=[\boldsymbol{0}_{m\times1}]$.

Step 2: 确定 $D_\theta^{s_t}$, 按第 3.3 节方法确定 $\boldsymbol{\varphi}_t(s_t)$.

Step 3: 对于离散动作空间, $\forall a_j\in A$, 由式 (14) 逼近 $\widetilde{Q}_t(s_t,a_j)$, 按给定策略确定 a_t 和 $\widetilde{Q}_t(s_t,a_t)$; 对于连续动作空间, $\forall x_i\in D_\theta^{s_t}$, 按给定策略确定 $\pi_t(x_i)$ 和 $Q_t(x_i,\pi_t(x_i))$, 由式 (15) 和 (17) 分别逼近 a_t 和 $\widetilde{Q}_t(s_t,a_t)$.

Step 4: 按式 (10)、(18) 和 (19) 增量更新 D_t、$\boldsymbol{W}_t$ 和 $\boldsymbol{E}_t$.

Step 5: 执行 a_t, 获取 s_{t+1} 和 r_t, 由 Step 2 和 Step 3 最终确定 a_{t+1} 和 $\widetilde{Q}_t(s_{t+1},a_{t+1})$.

Step 6: 计算 δ_t, 按式 (12) 和第 3.4 节所述方法迭代更新 $\boldsymbol{W}_{t+1}$ 和 $\boldsymbol{E}_{t+1}$, 再按式 (10)、(18) 和 (19) 增量更新 D_{t+1}、$\boldsymbol{W}_{t+1}$ 和 $\boldsymbol{E}_{t+1}$.

Step 7: 若 $D_{t+1}=D_t\bigcup\{s_{t+1}\}$, 则计算 $\boldsymbol{\varphi}_{t+1}(s_{t+1})$. 对于连续空间, 令 $Q_{t+1}(s_{t+1},\pi_t(s_{t+1}))=\widetilde{Q}_t(s_{t+1},a_{t+1})$; 否则 $\boldsymbol{\varphi}_{t+1}(s_{t+1})=\boldsymbol{\varphi}_t(s_{t+1})$. 对于离散与连续动作空间, 分别按式 (14) 和 (17) 计算 $\widetilde{Q}_{t+1}(s_{t+1},a_{t+1})$.

Step 8: 令 $t=t+1$. 如果 $t\bmod t_{\max}>0$ 且 s_t 未达终态, 则返回 Step 5; 否则, 令 $L=L+1$. 如果 $L>L_{\max}$, 则算法终止; 否则, 置 $\boldsymbol{E}_t$ 各元素为 0, 初始化 s_t, 返回 Step 2.

为了提高算法的运行效率, 可以略去 Step 7, 直接采用 $\widetilde{Q}_{t-1}(s_t,a_t)$ 替代 $\widetilde{Q}_t(s_t,a_t)$ 进行 δ_t 计算, 即

$$\delta_t=r_t+\gamma\widetilde{Q}_t(s_{t+1},a_{t+1})-\widetilde{Q}_{t-1}(s_t,a_t). \tag{20}$$

为了对式(3)和(20)两种δ_t计算方案加以区别, 将前者和后者分别称为同步和异步δ_t方案.

3.2 词典在线稀疏化构建

实例词典是状态空间的一种近似表示, 词典中实例的分布直接影响强化学习的算法性能[8]. 由于式(9)中$k(\cdot)$常采用核函数进行定义, LWL与核学习存在一定的等价性[14], 可以借鉴核稀疏化方法构建词典. 常用的核稀疏化方法主要有近似线性相关(ALD)、核主分量分析(KPCA)和新颖度标准(NC), 三者的时间复杂度分别为$O(n^2)$、$O(n^3)$、$O(n)$[10]. NC方法基于距离和预测误差进行稀疏化[15], 其低时间复杂度使其适合实例词典的在线构建, 但是由于强化学习存在探索与开采平衡问题, 在稀疏化过程中过早整合TD误差反而可能导致算法早熟[8].

本文给出3种基于距离度量的词典在线稀疏化构建方案, 如图1所示. 图1中, 小三角形代表观测状态s, 实心圆点代表词典D_t中的实例, 虚线圆圈代表s的近邻范围θ, 空心圆圈$\bar{x}_t$代表s的近邻实例平均值, $\bar{x}'_t$代表s和其近邻实例的总平均值, 即

$$\bar{x}_t = \sum_{i=1}^{|D_\theta^s|} x_i / |D_\theta^s|, \tag{21}$$

$$\bar{x}'_t = \Big(\sum_{i=1}^{|D_\theta^s|} x_i + s\Big) \Big/ (|D_\theta^s| + 1). \tag{22}$$

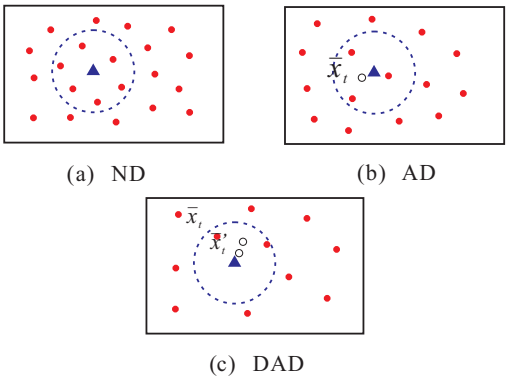
3种方案的在线稀疏化条件SIF(s)定义分别为

$$\min(d(s, D_t)) > \varsigma; \tag{23}$$

$$d(s, \bar{x}_t) > \varsigma,\ \min(d(s, D_t)) > \varsigma; \tag{24}$$

$$|d(s, \bar{x}_t) - d(s, \bar{x}'_t)| > \varsigma. \tag{25}$$

其中: $d(\cdot,\cdot)$为距离函数, ς为距离阈值.



(a) ND (b) AD

(c) DAD

图1 实例词典在线稀疏化构建方案

若要保证新增实例与词典中现有实例对所有观测状态形成的基向量线性无关, 记$D'_t = D_t \bigcup \{s\}$, $\boldsymbol{\phi}_t(x_i) = [\phi(x_i, x_1), \phi(x_i, x_2), \cdots, \phi(x_i, x_{|D'_t|})]$, $\boldsymbol{\Phi}_{D'_t} = [\boldsymbol{\phi}_t^{\mathrm{T}}(x_1), \boldsymbol{\phi}_t^{\mathrm{T}}(x_2), \cdots, \boldsymbol{\phi}_t^{\mathrm{T}}(x_{|D'_t|})]$, 则SIF$(s)$还需满足

$$|\boldsymbol{\Phi}_{D'_t}| \neq 0. \tag{26}$$

为了便于讨论, 将式(23)、(24)和(25)三种方案分别称为最近邻距离(ND)方案、平均近邻距离(AD)方案和双平均距离(DAD)方案. 其中: ND方案近似于均匀稀疏化, 当三者的θ和ς相同时, 稀疏化程度最低; DAD方案与ND方案相反; AD方案稀疏化程度介于二者之间. 3种方案可以看作是NC方法和文献[10]所提稀疏化方法的改进版本.

3.3 值函数与策略加权逼近

由式(14)、(15)和(17)可知, 值函数与策略的逼近实际是建立在D_θ^s和$\boldsymbol{\varphi}(s)$之上, 对给定近邻范围θ, D_θ^s直观上可通过范围最近邻法确定. 但是, 该办法存在一定问题, 由于词典D_t基于在线稀疏化方法构建, 特别是在框架运行初期, D_θ^s有可能为$\varnothing$, 使得框架无法运行. 此时, 可选用s在$\overline{D_\theta^s}$中的最近邻实例进行补救. 记

$$D_\theta^s(1) = \{x_i | x_i \in D_t, d(s, x_i) \leqslant \theta\}, \tag{27}$$

$$D_\theta^s(2) = \{\underset{x_i \in \overline{D_\theta^s(1)}}{\operatorname{argmin}}\ d(s, x_i)\}. \tag{28}$$

则s的θ范围近邻集D_θ^s可定义为

$$D_\theta^s = \begin{cases} D_\theta^s(1), & |D_\theta^s(1)| > 0; \\ D_\theta^s(2), & \text{otherwise}. \end{cases} \tag{29}$$

考虑到D_t和D_θ^s均由距离函数确定, $\boldsymbol{\varphi}(s)$直接建立在各近邻实例之上, 因此$\boldsymbol{\varphi}(s)$各权值$\varphi(s, x_i)$直观上可采用DWA算法进行定义, 由此给出值函数和策略的DWA逼近方案. 由式(9), 对于$\forall x_i \in D_\theta^s$, 有

$$\varphi(s, x_i) = k(d(s, x_i)) \Big/ \sum_{j=1}^{|D_\theta^s|} k(d(s, x_j)). \tag{30}$$

然而, DWA方案对框架存在一定问题. 由于D_t采用增量稀疏化方式构建, 在框架运行初期或者D_t稀疏度较大时, 有些观测状态将置于其范围最近邻集构成的凸包之外[6], 采用DWA逼近会带来较大误差. 特别地, 当$|D_\theta^s| = 1$时, s的值函数和策略将与其最近邻实例完全相同. 理论上讲, LWR可以较好地解决这一问题, 然而要想获得较高的逼近精度, 要求D_θ^s必须具备合适的规模, 否则容易造成欠拟合或过拟合问题. 由于D_t进行了稀疏化处理, D_θ^s规模通常较小, 采用LWR逼近效果并不理想. 考虑到值函数估计的极限分布是一个高斯过程[7], 可以选用以$\bar{x}_t$为中心的高斯分布近似描述值函数曲线, 即

$$\widetilde{Q}_t(s, a_j) = Q_t(\bar{x}_t, a_j) k_g(d(s, \bar{x}_t)). \tag{31}$$

其中: $Q_t(\bar{x}_t, a_j) = \sum_{i=1}^{|D_\theta^s|} Q_t(x_i, a_j)/|D_\theta^s|$, $k_g(\cdot)$为高斯核. 对比式(14), 对于$\forall x_i \in D_\theta^s$, 有

$$\varphi(s, x_i) = k_g(d(s, \bar{x}_t))/|D_\theta^s|. \tag{32}$$

将上述方案称为近邻均值高斯加权 (NAG) 方案. 由式 (32) 可知

$$\sum_{i=1}^{|D_\theta^s|}\varphi(s,x_i)=k_g(d(s,\bar{x}_t))\leqslant 1,$$

因此NAG方案并不严格满足加权平均算法的权值归一化条件. 因为 θ 通常取值较小, 所以 $k_g(d(s,\bar{x}_t))\approx 1$, 由NAG逼近得到的值函数和策略的绝对值通常只略小于DWA方案. 考虑到值函数逼近存在过估计倾向[3,6], NAG方案反而更适于值函数逼近. 当 $|D_\theta^s|=1$ 时, NAG较DWA方案解释也更合理一些.

3.4 值函数与资格迹迭代更新

词典中各实例的值函数迭代更新除了可以采用线性Sarsa(λ)外, 还可以采用GTD、GTD2和TDC等[2,10], 从OSKTD的实验结果看, 采用线性Sarsa(λ)的学习效果要优于这些方法[10]. 由于框架与OSKTD具有一定的等价性, 建议采用线性Sarsa(λ)作为值函数迭代更新的优选方案.

第3.1节并没有给出资格迹矩阵 $\boldsymbol{E}_t$ 的迭代更新方式, 主要原因是难以直接利用式 (4) 和 (5) 进行定义. 对于 $\forall a_j\in A$, 由式 (11) 有 $\nabla_{\boldsymbol{w}_t^{a_j}}\widetilde{Q}_t(s,a_j)=\boldsymbol{\phi}_t(s)$, $\boldsymbol{\phi}_t(s)$ 不能体现相同状态下不同动作间的值函数差异. 在框架中, 由于 $w_{ji}=Q(x_i,a_j)$, 可以尝试对传统表格型资格迹[11]进行改造以构建适用于框架的资格迹.

为了便于讨论, 对于离散动作空间, 当前观测状态 s 下选用的动作 a 也统一用集合形式表示为 $A_\theta^s=\{a\}$. 由于框架中 s 和 a 在值函数和策略逼近过程中是由 D_θ^s 和 A_θ^s 替代表示的, 直观上可以将传统表格型资格迹中的当前观测状态和所用动作替换为 D_θ^s 和 A_θ^s. 由 $\nabla_{\boldsymbol{w}_t^{a_j}}\widetilde{Q}_t(s,a_j)=\boldsymbol{\phi}_t(s)$ 可知, 框架中的资格迹更新还应考虑不同实例状态之间的差异, 直观上可以采用 $\phi(s,x_i)$ 进一步替换传统表格型资格迹中的1. 但是, 框架中 $\phi(s,x_i)$ 由式 (16) 确定, 由于 $\boldsymbol{\varphi}(s)$ 进行了 (近似) 归一化处理, 对于 $\forall x_i\in D_\theta^s$, $\phi(s,x_i)\approx 1/|D_\theta^s|$, 当 $|D_\theta^s|$ 较大时, $\phi(s,x_i)$ 将远小于1. 此外, 在框架运行过程中, $|D_\theta^s|$ 的大小并不固定, 如果直接采用 $\phi(s,x_i)$ 作为 (x_i,a) 的资格迹, 则会导致资格迹更新波动较大. 鉴于此, 采用 $\phi(s,x_i)|D_\theta^s|$ 替换传统表格型资格迹中的1来解决上述问题. 基于上述分析, $\boldsymbol{E}_t$ 的累加迹迭代更新方式定义为

$$\boldsymbol{e}_{t+1}^{a_j}=\gamma\lambda\boldsymbol{e}_t^{a_j}+\boldsymbol{\phi}_t(s)|D_\theta^s|\mathbf{1}_{A_\theta^s}(a_j).\tag{33}$$

其中: $\boldsymbol{e}_t^{a_j}$ 为 $\boldsymbol{E}_t$ 的第 j 行向量, $\mathbf{1}_{A_\theta^s}(a_j)$ 为指示函数.

$\boldsymbol{E}_t$ 各元素的替代迹迭代更新方式为

$$e_{t+1}(x_i,a_j)=\begin{cases}\phi(s,x_i)|D_\theta^s|,\ x_i\in D_\theta^s,\ a_j\in A_\theta^s;\\ 0,\ x_i\in D_\theta^s,\ a_j\notin A_\theta^s;\\ \gamma\lambda e_t(x_i,a_j),\ \text{otherwise.}\end{cases}\tag{34}$$

4 INNTD框架分析

框架具有模型简单、计算效率高和易于理解等优点. 由式 (14)、(15) 和 (17) 可知, 值函数和策略逼近为线性模型. 框架中, D_θ^s 确定、值函数和策略逼近、$\boldsymbol{W}_t$ 和 $\boldsymbol{E}_t$ 迭代更新的时间复杂度分别为 $O(|D_t|)$、$O(|D_\theta^s|)$ 和 $O(|D_t||A|)$, 词典、$\boldsymbol{W}_t$ 和 $\boldsymbol{E}_t$ 的空间复杂度分别为 $O(|D_t|)$、$O(|D_t||A|)$. 当不考虑SIF(s)是否满足式 (26) 时, 词典构建的时间复杂度为 $O(|D_t|)$. 即使考虑式 (26), 由于SIF(s)需要在满足式 (23)、(24) 或 (25) 后才对该式进行检验, 该式运行的次数并不是太多, 当 $|D_t|$ 较小时, 式 (26) 的 $O(|D_t|^3)$ 时间复杂度并不会对词典构建的运行效率造成太大影响. 框架将参数向量直接解释成实例状态的值函数和策略, 通过对新观测状态的近邻实例状态的值函数和策略进行加权, 以实现其值函数和策略逼近, 整个逼近看上去更加直观. 此外, 框架还具有良好的开放性和可扩展性. 框架采用开放式架构设计, 用户可以在框架不同部分灵活选用第3.2～3.4节给出的多种方案, 也可以自行定义相应的方案. 在第3节中, 仅给出了Sarsa(λ)类型框架, 参照该节内容用户容易结合TD(λ)、Q(λ)、GTD、GTD2、TDC等算法将该框架扩展为新的类型. 当加权函数采用核函数定义时, 框架也可看作是一类核强化学习算法, 采用现有的核强学习方法和理论解释和扩展. 从这一层面上看, 框架与近邻核平均[7]和OSKTD[10]算法存在一定的等价性.

同步 δ_t 框架秉承了线性逼近器和内插器 (平均器)[2,16-17]拥有的良好收敛特性, 当其满足一定假设条件时, 将以概率1收敛. 由式 (16)、(30)、(32) 可知, 基函数具有如下性质: 1) $0\leqslant\phi(s,x_i)\leqslant 1$; 2) $\sum_{i=1}^{|D_t|}\phi(s,x_i)\leqslant 1$. 性质1) 表明基函数有界, 性质2) 表明框架的值函数逼近器为内插器, 具有非扩张性. 对于式 (11), 有

$$\|\boldsymbol{w}_{t1}^{a_j}\boldsymbol{\phi}_t^{\mathrm{T}}(s)-\boldsymbol{w}_{t2}^{a_j}\boldsymbol{\phi}_t^{\mathrm{T}}(s)\|_\infty\leqslant\|\boldsymbol{w}_{t1}^{a_j}-\boldsymbol{w}_{t2}^{a_j}\|_\infty,$$

式 (17) 也具有类似性质. 由于词典采用在线稀疏化方法构建, 对有限连续状态空间而言, 随着学习的进行, 词典大小必定有界, 符合文献[16]的FT假设. 设 $\lim\limits_{t\to\infty}|D_t|=n$, 随着实例增量的添加, 记式 (11) 和 (17) 形成的逼近器序列为 $\boldsymbol{F}^{(1)},\boldsymbol{F}^{(2)},\cdots,\boldsymbol{F}^{(n)}$. 若 $\boldsymbol{F}^{(i)}$ 与 $Q(s,a)$ 有界、$Q(s,a)$ 适度光滑, 且 $\|\boldsymbol{F}^{(i)}-Q(s,a)\|\leqslant C\mathrm{dens}(D^{(i)})$. 其中: C 为常数, $\mathrm{dens}(D^{(i)})$ 为实例数量为 i 的词典密度. 则Q-learning类型框架收敛[16]. Q-learning算法采用off-policy策略, 其收敛条件较采用on-policy策略的TD(0)和Sarsa算法要求更加严格, 因此TD(0)和Sarsa类型框架在满足上述条件下也将收敛[16]. 文献[17]基于ODE就内插Q-learning算法的收敛性进行了证明, 当满足所列假设条件时,

Q-learning类型框架也收敛, 但其假设条件较文献[16]更为严格, 在此不作过多的讨论. 由于资格迹能提供更严格的误差边界[17], 文献[16-17]所列的收敛条件稍作修改即可用于TD(λ)、Q(λ)和Sarsa(λ)类型框架. 框架同时也是一类线性逼近算法, 且其词典最终将保持稳定, 因此也可直接采用文献[18]所列收敛条件对TD(λ)和Sarsa(λ)类型框架的收敛性进行分析. 如果词典最终能对状态空间形成有效表示, 当SIF(s)满足式(26)时, 基函数矩阵列满秩, 又由基函数性质1), 这两类框架只要满足文献[18]假设1和假设4即可收敛. 异步δ_t框架的收敛性尚无法给出理论证明, 但实验表明, 在学习率α较小时同样可以收敛.

5 仿真分析

针对框架同时面向连续状态与离散动作、连续状态与连续动作空间强化学习问题设计这一特点, 选用离散与连续动作空间两种小车爬山问题[11,19]对第3节所述Sarsa(λ)框架的24种方案组合的有效性进行验证, 并与kNN-TD(λ)和Ex$\langle a\rangle$(λ)算法[6]进行实验结果对比.

5.1 仿真问题和方案

小车爬山问题的任务是要以尽量短的时间使小车从山谷最低点到达右侧山顶, 其状态空间为二维连续空间, 由两个连续变量x_t和$\dot{x}_t$分别表示t时刻小车的位置(m)和速度(m/s), $x_t \in [-1.5, 0.5]$, $\dot{x}_t \in [-0.07, 0.07]$. 对于离散动作空间小车爬山问题(简称离散小车爬山), $a_t \in \{-1, 0, 1\}$表示对小车实施满油门后退、零油门或满油门前进操作; 对于连续动作空间小车爬山问题(简称连续小车爬山), $a_t \in [-1, 1]$. 两种小车爬山的动力学方程统一描述为

$$\begin{cases} x_{t+1} = \text{bound}[x_t + \dot{x}_{t+1}], \\ \dot{x}_{t+1} = \text{bound}[\dot{x}_t + 0.001\, a_t - 0.002\,5\, \cos(3x_t)]. \end{cases} \tag{35}$$

当x_{t+1}到达左边界, 置$\dot{x}_{t+1} = 0$. 当x_{t+1}到达右边界, 表明小车已到达山顶, $r_t = 100$, 否则$r_t = -1$. 当小车到达山顶或运行步数达到episode的最大时间步$t_{\max}$时, 当前episode学习结束.

第3节给出的Sarsa(λ)框架方案可形成48种组合形式, 考虑到替代迹的学习效果通常要优于累加迹[4,11,12], 本文只对24种采用替代迹的方案组合进行验证. 为了便于讨论, 每一替代迹方案组合采用一个4字母单词加以简记, 其中首字母S、A表示该组合采用同步或异步δ_t方案, 次字母Y、N表示该组合在词典构建过程中考虑或不考虑式(26), 第3个和第4个字母分别采用第3.2节和第3.3节各方案简写的首字母, 表示该组合具体采用何种稀疏化和加权逼近方案. 例如, SYND表示该组合采用同步δ_t方案、词典构建过程中考虑式(26)、词典稀疏化选用ND方案、值函数和策略加权逼近选用DWA方案.

在两种小车爬山仿真中, 各组合均采用greedy策略, $t_{\max}$为1 000, 总episodes数$L_{\max}$为500, 各episode初始状态均为$(x, \dot{x}) = (-0.5, 0)$, λ为0.95, ς为0.03. DWA方案的$k(d(s, x_i))$采用逆距离进行定义, $k(d(s, x_i)) = 1/(1+d^2(s, x_i))$, NAG方案的$k_g(d(s, \bar{x}_t))$为$\mathrm{e}^{(-d^2(s, \bar{x}_t))}$, 两种方案距离函数均采用Euclidean距离进行度量. 考虑到状态空间在不同维度上的值域范围差异较大, 在计算状态与实例间的距离时, 先将各分量均统一规范化到$[-1, 1]$. 仿真实验着重考察α、θ对各组合学习性能的影响, $\alpha \in \{0.05, 0.10, \cdots, 1\}$, $\theta \in \{0.08, 0.10\}$, 其他参数配置分别为: 离散小车爬山$\gamma = 0.95$, 连续小车爬山$\gamma = 1$, 连续动作空间均匀离散为$A = \{-1, -0.8, \cdots, 0.8, 1\}$.

为进一步验证框架的有效性, 选用kNN-TD(λ)[6]与Ex$\langle a\rangle$(λ)算法进行对比仿真, 两种算法可近似看作是ANND组合的离线版本. 为了使对比结果更加客观, 两种算法的k值和状态空间离散颗粒度尽量沿用kNN-TD(λ)算法在离散小车爬山仿真中所用配置: 1) 当$\theta = 0.08$时, $k = 4$, 状态空间离散成21×21个实例; 2) 当$\theta = 0.10$时, $k = 5$, 状态空间离散成26×26个实例. 两种算法学习策略和其他参数配置与各对比组合完全相同.

5.2 仿真结果和分析

α对各组合的学习性能影响曲线如图2和图3所示. 图中: $*$为ND, 方形为NN, 菱形为AD, 三角为AN, 五星为DD, 圆形为DN. 图2(a)中: ND曲线为小车采用SYND组合在不同α下第50～500 episode到达山顶的平均时间步数, 其余各图含义依此类推. 由于kNN-TD(λ)和Ex$\langle a\rangle$(λ)算法与ANND组合比较接近, 两种算法的实验结果分别置于图2(g)、图2(h)、图3(g)和图3(h)中, 采用六星符号标记. 各组合在20种α下学习到的最小平均时间步数和该次学习结束时词典的大小如表1和表2所示. 表1第1行中的103.9表示SYND组合学习到的最小平均时间数, 由图2(a)可知, 此时$\alpha = 0.65$. 表1和表2中各数据的含义同样依此类推, 限于篇幅, 略去α对各组合的词典大小、每一时间步逼近时使用的平均近邻数的影响等图表, 相关结论直接给出.

由仿真结果可以得出如下结论: 1) 当$\alpha > 0.5$时, 采用异步δ_t方案的组合收敛性能急剧下降, 主要是因为其δ_t值一般要大于同步δ_t值, 相当于增大了同步δ_t方案的α值; 2) 当学习收敛时, α对各组合的学习性能(平均时间步数)、词典大小、平均近邻数影响不大;

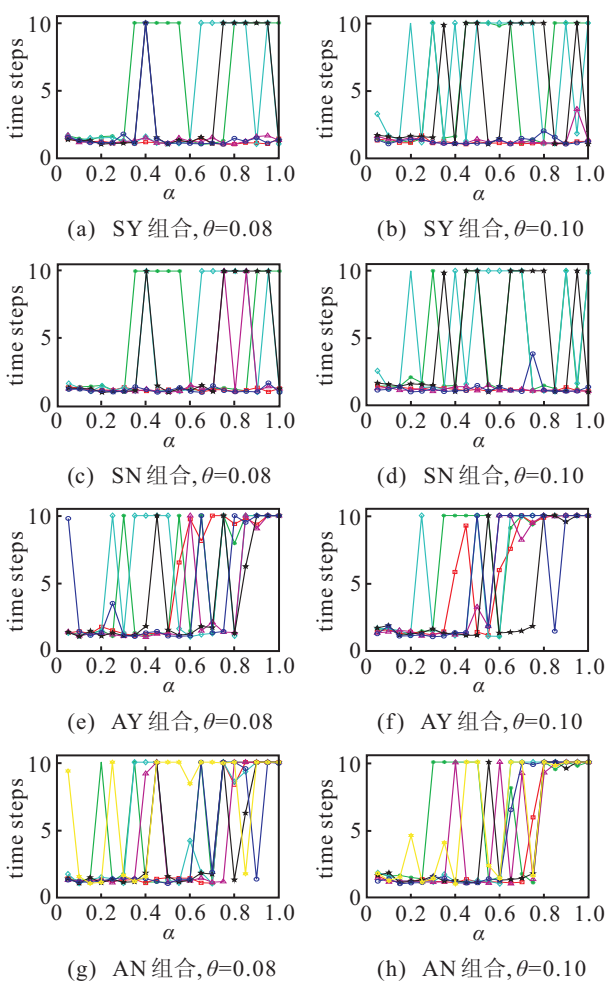


图 2　α 对各组合离散小车爬山学习性能影响曲线

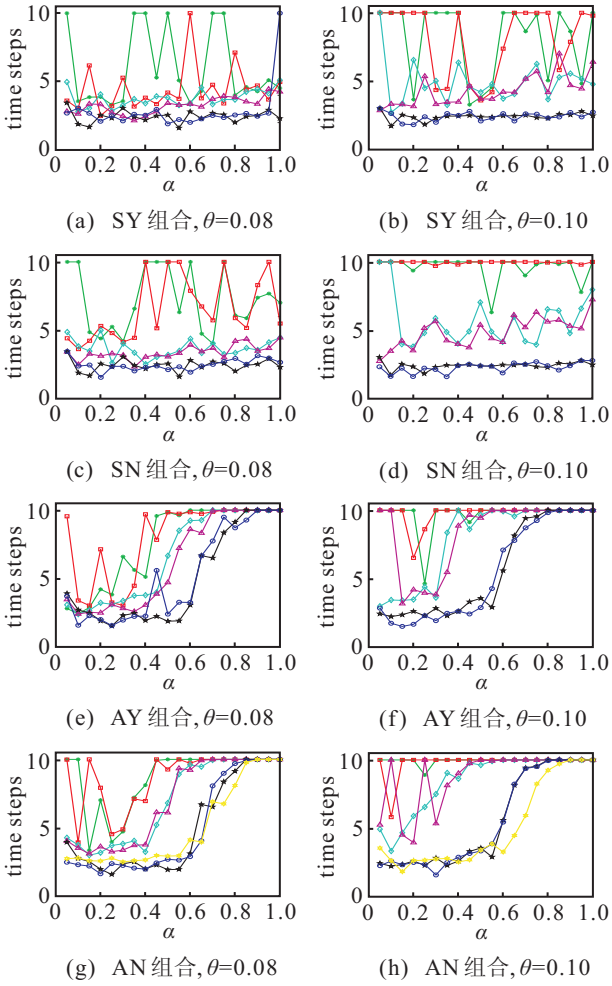


图 3　α 对各组合连续小车爬山学习性能影响曲线

表 1　离散小车爬山各组合时间步数与词典大小

组合	θ	最小平均时间步数				对应实例词典大小			
		SY	SN	AY	AN	SY	SN	AY	AN
ND	0.08	103.9	106.8	109.4	106.0	522	1073	565	1 035
NN	0.08	104.0	106.0	113.2	107.0	537	1050	495	1 269
AD	0.08	106.0	107.1	106.1	104.3	516	836	553	674
AN	0.08	104.9	106.0	102.0	110.2	547	831	511	734
DD	0.08	104.0	104.0	106.7	106.7	297	297	361	361
DN	0.08	104.0	103.9	107.1	103.0	348	280	309	300
ND	0.10	111.9	121.9	104.2	115.4	459	1347	434	1 172
NN	0.10	104.3	111.0	116.4	113.3	465	795	443	1 022
AD	0.10	110.3	108.9	108.0	108.0	442	746	413	488
AN	0.10	109.0	110.0	111.0	108.0	467	841	460	767
DD	0.10	105.0	105.0	115.0	115.0	166	166	276	276
DN	0.10	104.8	105.4	105.0	107.3	297	320	230	334

表 2　连续小车爬山各组合时间步数与词典大小

组合	θ	最小平均时间步数				对应实例词典大小			
		SY	SN	AY	AN	SY	SN	AY	AN
ND	0.08	330.8	401.6	246.9	335.6	521	1 317	560	1 675
NN	0.08	249.8	362.3	306.9	393.1	519	1 781	577	1 529
AD	0.08	279.2	251.9	245.2	299.3	592	1 035	564	941
AN	0.08	215.3	233.7	247.0	310.8	482	820	537	996
DD	0.08	160.8	160.8	158.9	158.9	415	415	344	344
DN	0.08	190.8	155.2	162.0	163.3	335	370	370	383
ND	0.10	327.7	631.6	463.2	889.3	444	1 975	445	1 444
NN	0.10	359.6	970.6	652.3	581.8	462	1 816	487	1 913
AD	0.10	271.7	380.9	298.3	333.7	444	1 048	455	961
AN	0.10	286.7	274.5	317.7	395.5	476	842	467	1 017
DD	0.10	172.3	172.3	222.7	222.7	312	312	281	281
DN	0.10	181.8	160.6	151.0	159.2	335	314	226	302

3) 考虑式 (26) 即基向量的线性无关性, 可明显增大 ND 和 AD 两种词典构建方案的稀疏化程度, 缩小词典规模, 但对各组合的学习性能影响不大; 4) 对于离散小车爬山, 各组合的收敛性能主要受加权逼近方案影响, NAG 方案明显优于 DWA 方案, 多数组合的学习性能相当优秀, 其收敛性能也要优于 kNN-TD(λ) 算法; 5) 对于连续小车爬山, 只有采用了 DAD 方案的组合取得了较好的学习结果, 但仍远差于离散小车爬车仿真结果, 主要是因为加权逼近方案具有平滑作用, 特别是 ND 和 AD 两种词典构建方案的稀疏化程度较小, 导致策略逼近时过于平滑, 较大程度地偏离了 ± 1 动作值, SNDN 和 ANDN 等组合的学习性能较好, 要优于 Ex$\langle a\rangle$(λ) 算法, 同时也优于文献 [19] 170 ~ 180 步的实验结果; 6) ς 和 θ 对各组合的平均近邻数影响较大, ς 过小和 θ 过大将导致各组合的泛化能力下降, 甚至学习不能收敛; 7) 当方案组合在某些 α 上不收敛时, 其词典规模较其在收敛 α 下得到的词典小, 平均近邻数则正好相反, 表明此时方案组合探索程度

不够且存在过平滑现象, 可选用ε-greedy 策略增加探索能力并适当增大ς值; 8) 综合考虑计算效率和学习性能, 建议采用SNDN 组合, α取值建议在0.2 ∼ 0.6之间.

最后综合讨论λ、γ、ς和θ的取值问题. 由于框架中值函数逼近建立在局部学习基础之上, 有必要扩大δ_t对非近邻实例的影响[4]. 同时, 由于后续状态s_{t+1}多为s_t的近邻, 也需重视s_{t+1}的值函数带来的影响, 因此λ和γ均建议采用较大的取值. ς和θ原则上需手工确定, 此处仅给出其一般调优原则: ς不宜过小, θ不宜过大, 否则将产生过度平滑和加权逼近误差过大等问题, 特别是对于连续动作空间强化学习问题. 从小车爬山仿真结果看, SNDN 组合的平均近邻数保持在3 左右较好.

6 结 论

本文针对连续状态与离散动作空间、连续状态与连续动作空间两类强化学习问题, 提出了一种基于LWL 的增量最近邻TD 学习框架. 框架建立在强化学习中相邻状态的值函数和策略通常具有较高的相似性这一规律之上, 通过增量方式在线选取部分观测状态作为学习实例表示状态空间, 基于新观测状态的近邻实例加权逼近其值函数和策略. 就框架各主要组成部分给出了多种方案, 并通过仿真实验对这些方案组合的有效性进行了验证. 结果表明, SNDN 方案组合具有较好的学习性能和计算效率, 框架具有简单、高效、开放、易于理解等优点, 在满足一定假设条件下, δ_t同步框架将以概率1 收敛. 下一步的研究方向是从理论上分析θ和ς对框架收敛性能带来的影响, 以及如何自适应地确定θ和ς的取值.

参考文献(References)

[1] 徐昕. 增强学习与近似动态规划[M]. 北京: 清华大学出版社, 2010: 10.
(Xu X. Reinforcement and approximate dynamic programming[M]. Beijing: Tsinghua University Press, 2010: 10.)

[2] Wiering M, Van Otterlo M. Reinforcement learning: State-of-the-art[M]. Berlin: Springer, 2012: 207-251.

[3] Vamplew P, Ollington R. Global versus local constructive function approximation for on-line reinforcement learning[C]. Advances in Artificial Intelligence. Berlin: Springer, 2005: 113-122.

[4] Sutton R S. Generalization in reinforcement learning: successful examples using sparse coarse coding[C]. NIPS 1996. Cambridge: MIT Press, 1996: 1038-1044.

[5] Martin H J A, de Lope J, Maravall D. Robust high performance reinforcement learning through weighted k-nearest neighbors[J]. Neurocomputing, 2011, 74(8): 1251-1259.

[6] Smart W D, Kaelbling L P. Practical reinforcement learning in continuous spaces[C]. ICML 2000. Burlington: Morgan Kaufmann Publishers, 2000: 903-910.

[7] Ormoneit D, Sen S. Kernel-based reinforcement learning [J]. Machine Learning, 2002, 49(2/3): 161-178.

[8] Ratitch B, Precup D. Sparse distributed memories for on-line value-based reinforcement learning[C]. Machine Learning: ECML 2004. Berlin: Springer, 2004: 347-358.

[9] Lee D, Lee J. Incremental receptive field weighted actor-critic[J]. IEEE Trans on Industrial Informatics, 2013, 9(1): 62-71.

[10] Chen X, Gao Y, Wang R. Online selective kernel-based temporal difference learning[J]. IEEE Trans on Neural Networks and Learning Systems, 2013, 24(12): 1944-1956.

[11] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT Press, 1998: 133-226.

[12] Framling K. Replacing eligibility trace for action-value learning with function approximation[C]. The 15th European Symposium on Artificial Neural Networks. Bruxelles: D-side Publishing, 2007: 313-318.

[13] Grondman I, Busoniu L, Lopes G A D, et al. A Survey of actor-critic reinforcement learning: standard and natural policy gradients[J]. IEEE Trans on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2012, 42(6): 1291-1307.

[14] Atkeson C G, Moore A W, Schaal S. Local weighted learning[J]. Artificial Intelligence Review, 1997, 11(1): 11-73.

[15] Platt J. A resource-allocating network for function interpolation[J]. Neural Computation, 1991, 3(2): 213-225.

[16] Szepesvari C, Smart W D. Interpolation-based Q-learning[C]. ICML 2004. New York: ACM Press, 2004: 100-107.

[17] Melo F S, Ribeiro M I. Convergence of Q-learning with linear function approximation[C]. ECC 2007. KOS: Greece, 2007: 2671-2678.

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

[19] Melo F S, Lopes M. Fitted natural actor-critic: A new algorithm for continuous state-action MDPs[M]. Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, 2008: 66-81.

(责任编辑: 郑晓蕾)