计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20148160
52(12):2764-2775,2015

一种近似模型表示的启发式 Dyna 优化算法

钟 珊[1,2] 刘 全[1,3,5] 傅启明[1,4] 章宗长[1] 朱 斐[1] 龚声蓉[1,2]
[1](苏州大学计算机科学与技术学院 江苏苏州 215006)
[2](常熟理工学院计算机科学与工程学院 江苏常熟 215500)
[3](软件新技术与产业化协同创新中心 南京 210000)
[4](苏州科技学院电子与信息工程学院 江苏苏州 215006)
[5](符号计算与知识工程教育部重点实验室(吉林大学) 长春 130012)
(sunshine-620@163.com)

A Heuristic Dyna Optimizing Algorithm Using Approximate Model Representation

Zhong Shan[1,2], Liu Quan[1,3,5], Fu Qiming[1,4], Zhang Zongzhang[1], Zhu Fei[1], and Gong Shengrong[1,2]
[1](*School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu* 215006)
[2](*School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, Jiangsu* 215500)
[3](*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing* 210000)
[4](*College of Electronic & Information Engineering, Suzhou University of Science and Technology, Suzhou, Jiangsu* 215006)
[5](*Key Laboratory of Symbol Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun* 130012)

Abstract In allusion to the problems of reinforcement learning with Dyna-framework, such as slow convergence and inappropriate representation of the environment model, delayed learning of the changed environment and so on, this paper proposes a novel heuristic Dyna optimization algorithm based on approximate model—HDyna-AMR, which approximates Q value function via linear function, and solves the optimal value function by using gradient descent method. HDyna-AMR can be divided into two phases, such as the learning phase and the planning phase. In the former one, the algorithm approximately models the environment by interacting with the environment and records the feature appearing frequency, while in the latter one, the approximated environment model can be used to do the planning with some extra rewards according to the feature appearing frequency. Additionally, the paper proves the convergence of the proposed algorithm theoretically. Experimentally, we apply HDyna-AMR to the extended Boyan Chain problem and Mountain Car problem, and the results show that HDyna-AMR can get the approximately optimal policy in both discrete and continuous state space. Furthermore, compared with Dyna-LAPS (Dyna-style planning with linear approximation and prioritized sweeping) and Sarsa(λ), HDyna-AMR outperforms Dyna-LAPS and Sarsa(λ) in terms of convergence rate, and the robustness to the changed environment.

Key words reinforcement learning (RL); model learning; planning; function approximation; machine learning

收稿日期:2014-10-30;**修回日期**:2015-03-19

基金项目:国家自然科学基金项目(61272005,61303108,61373094,61472262,61502323,61502329);江苏省自然科学基金项目(BK2012616);江苏省高校自然科学研究项目(13KJB520020);吉林大学符号计算与知识工程教育部重点实验室基金项目(93K172014K04);苏州市应用基础研究计划项目(SYG201422)

通信作者:刘 全(quanliu@suda.edu.cn)

摘　要　针对基于查询表的 Dyna 优化算法在大规模状态空间中收敛速度慢、环境模型难以表征以及对变化环境的学习滞后性等问题,提出一种新的基于近似模型表示的启发式 Dyna 优化算法(a heuristic Dyna optimization algorithm using approximate model representation, HDyna-AMR),其利用线性函数近似逼近 Q 值函数,采用梯度下降方法求解最优值函数. HDyna-AMR 算法可以分为学习阶段和规划阶段. 在学习阶段,利用 agent 与环境的交互样本近似表示环境模型并记录特征出现频率;在规划阶段,基于近似环境模型进行值函数的规划学习,并根据模型逼近过程中记录的特征出现频率设定额外奖赏. 从理论的角度证明了 HDyna-AMR 的收敛性. 将算法用于扩展的 Boyan chain 问题和 Mountain car 问题. 实验结果表明,HDyna-AMR 在离散状态空间和连续状态空间问题中能学习到最优策略,同时与 Dyna-LAPS(Dyna-style planning with linear approximation and prioritized sweeping)和 Sarsa(λ)相比,HDyna-AMR 具有收敛速度快以及对变化环境的近似模型修正及时的优点.

关键词　强化学习;模型学习;规划;函数逼近;机器学习

中图法分类号　TP181

强化学习(reinforcement learning, RL)是一种源于动物心理学[1]并在控制理论、统计学和心理学等相关科学的相互发展中形成的一种介于监督学习和非监督学习之间的连接主义机器学习方法[2]. Agent 在与环境交互的过程中,通过不断"试错"学习,寻求期望累积奖赏最大的动作[3],从而实现环境状态到动作的最优映射. 目前 RL 已在电子电路、图像处理、生物医学和网络应用等方面获得了广泛的应用[4-8].

强化学习可以分为直接学习和间接学习[9]. 直接学习是指 agent 利用与环境交互所得到的真实经验实现值函数的增量学习,从而获取最优策略;间接学习是指 agent 利用环境模型给出的模拟经验,实现值函数的学习和最优策略的获取. 直接学习和间接学习也可以分别简称为学习和规划,学习和规划的本质区别在于是否采用真实经验,因此,通过将学习和规划相结合对同一个值函数更新,可以加快值函数的收敛[10-11].

Sutton[12]首次提出了一种将学习、规划和反馈组合在一起的结构,并将该结构命名为 Dyna. 该算法在学习阶段和规划阶段采用一步表格式 Q 学习方法对值函数更新,并在规划过程中采用随机采样方式获取模拟样本以加快算法收敛速度. Moore 等人[13]和 Peng 等人[14]在 Sutton 提出的 Dyna 的基础上,采用优先级扫描采样方式替代随机采样方式,能有效地改善 Dyna 的收敛性能. 此后,Sutton 等人[15]在 Dyna 基础上,引入 V 值函数的线性逼近和模型近似,得到改进的 Dyna 优化算法 Dyna-LAPS (Dyna-style planning with linear approximation and prioritized sweeping),Dyna-LAPS 在规划中加入了 Moore 和 Peng 的优先级扫描方式,能收敛至最小二乘法相同的最优解. Santos 等人[16]提出了将启发式搜索算法 A* 与 Dyna 优化算法相结合的 Dyna-H (a heuristic Dyna),将其应用于角色扮演游戏中,通过实验证明了 Dyna-H 方法能产生比一步 Q 学习 Dyna 算法更好的效果,但启发式函数的构造需要领域知识,使得 Dyna-H 方法的应用具有一定的局限性.

孙洪坤等人[17]在经典的 Dyna 优化算法的基础上,设计了一种采用优先级扫描方法并适合随机环境的 Dyna-PS(Dyna with prioritized sweeping)算法,以加快算法收敛速度. 于俊等人[18]针对贝叶斯 Q 学习方法存在的收敛速度慢和收敛精度低的问题,设计了一种基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法(Bayesian Q learning with Dyna architecture and prioritized sweeping, Dyna-PS-BayesQL),学习过程采用贝叶斯 Q 学习更新动作值函数参数,规划过程采用优先级扫描和动态规划法对 Q 值函数进行更新.

以上工作均着力于改善 Dyna 优化算法,能在实现模型学习的同时加快算法的收敛速度,具有重要的意义,但仍存在一定的不足. 如文献[15]对一个状态的多个动作建立同一个期望特征迁移概率矩阵和立即奖赏向量,因此,学习的模型不够精确,同时无法及时感知环境动态性变化. 文献[12-14,16-18]均采用表格式方式存储动作值函数,因此,在大规模或连续状态空间问题中会面临"维数灾"的问题[1],同时无法估计 agent 未访问过的状态动作对的 Q 值,从而不具备泛化能力[19].

为了解决这些问题,本文设计了一种近似模型表示的启发式 Dyna 优化算法(a heuristic Dyna

optimization algorithm using approximate model representation, HDyna-AMR),该算法采用线性函数逼近器逼近 Q 值函数,对模型进行近似学习,在学习过程中记录特征的出现频率,将特征更新的时间差分(temporal difference, TD)误差作为优先级;在规划过程中根据优先级大小选择特征进行重点采样,对出现频率较小的特征设定额外奖赏,从而提高算法整体的收敛速度和学习模型的精确性,能有效地解决大规模连续或离散状态空间问题.

1 Markov 决策过程和最优策略

在强化学习框架中,满足 Markov 性质的学习任务可称为 Markov 决策过程(Markov decision processes, MDPs)[20]. 一个 MDP 可以表示为一个四元组 $M=\langle X,U,\rho,f\rangle$,其中:

1) X 为状态空间,$x_t\in X$ 为 agent 在时刻 t 所处的状态;

2) U 为动作空间,$u_t\in U$ 为 agent 在时刻 t 采取的动作;

3) $\rho:X\times U\to\mathbb{R}$ 为立即奖赏函数,表示 agent 在时刻 t 位于状态 x_t 执行动作 u_t 并转移到状态 x_{t+1} 获得的立即奖赏值,可以表示为 $r_t=\rho(x_t,u_t)$;

4) $f:X\times U\times X\to[0,1]$为状态转移函数,表示 agent 在时刻 t 位于状态 x_t 执行动作 u_t 所能转移到的下一状态 x_{t+1},可以表示为 $f(x_t,u_t,x_{t+1})$.

在 RL 中,策略 $h:X\to U$ 是状态空间 X 到动作空间 U 的映射,可以表示为 $u_t=h(x_t)$,表示 agent 在时刻 t 位于状态 x_t 采取动作 u_t. 根据当前的策略可以求出状态动作对的值函数,称为 Q 值函数.

Q 值函数可以表示为

$$\forall x\in X,u=h(x):Q^h(x,u)=\rho(x,u)+\gamma\sum_{x'\in X}f(x,u,x')V^h(x'). \tag{1}$$

RL 的目标是寻求最优策略 h^*,即最优值函数的贪心策略. 对于任意策略的 Q 值均满足 $\forall x\in X$: $Q(x,h^*(x))\geqslant Q(x,h(x))$,且最优 Q 值函数 $Q^*(x,u)$可表示为

$$\forall x\in X,u=h(x):Q^*(x,u)=\rho(x,u)+\gamma\sum_{x'\in X}f(x,u,x')\max_{u'\in U}Q^*(x',u'). \tag{2}$$

当环境模型已知,即 ρ 和 f 已知时,可以采用动态规划或启发式方法求解最优策略;当环境模型未知时,即 ρ 和 f 未知时,可以采用蒙特卡洛(Monte Carlo, MC)方法和 TD 方法求最优策略.

2 HDyna-AMR 算法

2.1 线性函数逼近和 TD(λ)

在目前已有的 Dyna 优化算法中,学习和规划都采用 1 步 TD 更新方式. 为了进一步加快收敛速度,可以在学习和规划阶段引入资格迹,同时为了克服传统二维表格表示 Q 值函数的缺点,采用函数逼近器来表示 Q 值函数,使得子状态空间中的经验信息能通过泛化扩展到整个状态空间,满足大规模状态空间或连续状态空间的需求.

在策略 $h:X\to U$ 下,agent 在时刻 t 位于状态 $\boldsymbol{x}_t\in X$ 选择动作 $u_t\in U$. $\boldsymbol{x}=(x_1,x_2,\cdots,x_N)$表示状态的 N 维输入参数. 状态动作对$(\boldsymbol{x}_t,u_t)$可以通过特征向量 $\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)$表示,即 $\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)=\{\phi_t(1),\phi_t(2),\cdots,\phi_t(n-1),\phi_t(n)\}$,$\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)\in\mathbb{R}$. 参数向量 $\boldsymbol{\theta}_t$ 可以表示为 $\boldsymbol{\theta}_t=(\theta_t(1),\theta_t(2),\cdots,\theta_t(n))^{\mathrm{T}}$,则原始的样本序列可以转换为特征样本序列:$(\boldsymbol{\phi}_1,r_2,\boldsymbol{\phi}_2)$, $(\boldsymbol{\phi}_2,r_3,\boldsymbol{\phi}_3)$,$(\boldsymbol{\phi}_3,r_4,\boldsymbol{\phi}_4)$,…. 此时,采用线性函数逼近的近似状态动作值函数 $Q(\boldsymbol{x}_t,u_t)$,可以表示为 $Q_t(\boldsymbol{\theta}_t,\boldsymbol{\phi}_t)$,如式(3)所示:

$$Q_t(\boldsymbol{\theta}_t,\boldsymbol{\phi}_t)=\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}_t\approx E\{\sum_{n=0}^{\infty}\gamma^n r_{t+n+1}\mid \boldsymbol{x}=\boldsymbol{x}_t,u=u_t\}, \tag{3}$$

其中,$\gamma\in[0,1]$为折扣因子. $Q^h(\boldsymbol{x},u)$表示在策略 h 下状态动作对$(\boldsymbol{x},u)$的真实 Q 值. 当目标函数为最小化所有状态动作分布的均方误差(mean-square error, MSE)时,可表示为

$$e_t^{\mathrm{MSE}}=\sum_t p(\boldsymbol{x}_t,u_t)(Q^h(\boldsymbol{x}_t,u_t)-Q_t(\boldsymbol{x}_t,u_t))^2, \tag{4}$$

其中,$p(\boldsymbol{x},u)$表示不同状态动作值 $Q_t(\boldsymbol{x},u)$的误差分布权重,用于平衡状态动作的重要性对 MSE 的影响. 通常假设 $p(\boldsymbol{x},u)$与状态动作对样本分布相同,因此,在计算时可以忽略. 此时,根据式(3)和式(4)以及梯度下降方法,可以得到参数向量 $\boldsymbol{\theta}_t$ 的更新 $\boldsymbol{\theta}_{t+1}$为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha(r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi})\boldsymbol{\phi}, \tag{5}$$

其中,α 为步长因子,$r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}$ 为当前特征的 TD 误差.

Sutton 等人[15]采用 1 步 TD 进行参数更新,收敛性能不佳. 为使得算法更快收敛,借鉴 Sutton 等人[1]提出的前向更新和后向更新的等价观点,引入资格迹实现前向更新. 在每个时间步,对所有特征的资格迹进行衰减:

$$\boldsymbol{e}(u,d)=\lambda\gamma\boldsymbol{e}(u,d), \tag{6}$$

其中,$u=1,2,\cdots,|U|$;$d=1,2,\cdots,D$,D 表示特征向量的维数.

假设 agent 当前所处的状态为 $\boldsymbol{x}_t$,选择的动作为 u_t,其对应的特征为 $\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)$. 更新资格迹时,对当前状态动作对$(\boldsymbol{x}_t,u_t)$的特征 $\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)$的资格迹加 1,而 $\boldsymbol{x}_t$ 对应的其他动作的特征资格迹赋值为 0,资格迹的更新为

$$\begin{cases}\boldsymbol{e}(u,d)=\boldsymbol{e}(u,d)+\boldsymbol{\phi}, & u=u_t,\\ \boldsymbol{e}(u,d)=0, & u\neq u_t,\end{cases} \tag{7}$$

其中, $d=1,2,\cdots,D$.

采用 Sarsa(λ)算法对 Q 值进行学习,实现对参数向量 $\boldsymbol{\theta}_t$ 的更新 $\boldsymbol{\theta}_{t+1}$ 为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha\boldsymbol{e}(u,d)(r+\gamma\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}), \tag{8}$$

由式(8)可以看出,资格迹的引入能将当前状态动作对的 Q 值误差传播到较早时刻的状态动作对,使它们受到当前状态动作对 Q 值更新的影响,因此,能有效地加快学习的收敛速度.

2.2　探索机制和启发式额外奖赏

在学习阶段,采用软最大化方法,即 Softmax 动作选择策略,对于 M 个可选的离散动作,动作 u 被选择的概率为

$$p(\boldsymbol{\phi}(\boldsymbol{x},u))=\frac{\exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x},u)/\tau)}{\sum_{i=1}^{M}\exp(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x},u_i)/\tau)}, \tag{9}$$

其中,$\tau(\tau>0)$是温度参数,用于控制动作被选择的概率. 初始时 τ 可以设定为较大的值,以使得 agent 能在状态空间中进行全面地探索,避免过早陷入局部最优. τ 值的更新可以表示为

$$\tau=T_0/(t+1), \tag{10}$$

其中,$T_0>0$ 为初始温度,t 为算法的当前迭代次数. τ 的减少使得贪心动作(Q 值最大的动作)被选择的概率逐渐增加. 当 τ 的值降低到接近 0 时,agent 在所有状态处均趋向于选择贪心动作. 此时若环境的动态性发生变化,由式(9)计算得到的探索概率已趋向于一个很小的值,仅在学习阶段进行探索不能发现环境的变化,从而无法及时对模型进行修正;而在规划阶段,由于模型已经错误,规划的次数越多,反而越难发现环境发生的变化.

为了能在算法运行的后期仍具备一定的探索能力,从而及时发现环境的变化和模型的错误. 本文设计了一种启发式的立即奖赏补偿方式. 在学习阶段记录特征被选择的次数作为特征的出现频率;而在规划过程中,在对特征的 Q 值进行更新时,对频率小的特征,启发式的为立即奖赏增加一个额外奖赏,以增大出现频率小的特征的 Q 值,从而提高该特征对应的动作在学习阶段被选择的概率. 额外奖赏的计算可表示为

$$r'=\eta\sqrt{\frac{1}{\chi}}, \tag{11}$$

其中,η 为足够小的正数,χ 为特征的出现频率.

2.3　近似模型学习

在 Dyna-LAPS 模型[15]中,虽然对模型进行了近似地学习,但在求取特征迁移概率矩阵和立即奖赏向量时,均对所有动作求期望,因此,学习的近似模型不够精确.

为了解决该问题,可以对每个动作 u 都维护一个特征迁移概率矩阵 $\boldsymbol{F}_u\in\mathbb{R}^D\times\mathbb{R}^D$和立即奖赏向量 $\boldsymbol{b}_u\in\mathbb{R}^D$,$\boldsymbol{F}_u$ 表示特征在动作 u 作用下迁移到其他特征的概率期望值,$\boldsymbol{b}_u$ 表示特征在动作 u 作用下可以获得的奖赏的期望值. 假设 agent 在当前时刻位于状态 $\boldsymbol{x}_t$,采取动作 u_t 时对应的特征为 $\boldsymbol{\phi}=\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)$,获得的立即奖赏为 r_t,下一个状态为 $\boldsymbol{x}_{t+1}$. 根据当前的策略 h,可以得到下一个时刻采取的动作 u_{t+1},下一个状态动作对的特征为 $\boldsymbol{\phi}'=\boldsymbol{\phi}(\boldsymbol{x}_{t+1},u_{t+1})$. 在时刻 t 时,特征迁移概率矩阵 $\boldsymbol{F}_u$ 和立即奖赏向量 $\boldsymbol{b}_u$ 的更新为

$$\begin{cases}\boldsymbol{F}_u^t\leftarrow\boldsymbol{F}_u^{t-1}+\alpha(\boldsymbol{\phi}'-\boldsymbol{F}_u^{t-1}\boldsymbol{\phi})\boldsymbol{\phi}^{\mathrm{T}},\\ \boldsymbol{b}_u^t\leftarrow\boldsymbol{b}_u^{t-1}+\alpha(r-(\boldsymbol{b}_u^{t-1})^{\mathrm{T}}\boldsymbol{\phi})\boldsymbol{\phi}.\end{cases} \tag{12}$$

2.4　基于优先级的重点采样

在学习阶段,为了寻求最优状态动作值函数 Q^*,当 $Q\neq Q^*$,在每个动作对处产生 TD 误差,即 $\delta=Q^*-Q$,将$|\delta|$作为优先级,则:

$$pri(\boldsymbol{\phi})=|\delta|=|r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}|. \tag{13}$$

假设优先级阈值为 pri,此时如果特征 $\boldsymbol{\phi}$ 的优先级 $pri(\boldsymbol{\phi})>pri$ 时,将特征 $\boldsymbol{\phi}$ 加入到优先级队列中,并按优先级大小对队列中的特征进行重新排序.

在规划阶段,为了克服随机采样造成的一些无意义的更新,对优先级队列中的特征进行重点采样,每次从优先级队列中选择队首元素,即优先级最大的特征 $\boldsymbol{\phi}=\boldsymbol{\phi}(\boldsymbol{x}_t,u_t)$,采用回溯机制扫描其前驱特征,前驱特征 $\boldsymbol{\phi}'=\boldsymbol{\phi}(\boldsymbol{x}_{t-1},u_{t-1})$和对应的立即奖赏 r 的计算为

$$\begin{cases}\boldsymbol{\phi}\leftarrow\boldsymbol{F}_u\boldsymbol{\phi}',\\ r\leftarrow\boldsymbol{b}_u^{\mathrm{T}}\boldsymbol{\phi}'.\end{cases} \tag{14}$$

计算前驱特征 $\boldsymbol{\phi}'=\boldsymbol{\phi}(\boldsymbol{x}_{t+1},u_{t+1})$的优先级为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha\boldsymbol{e}(u,d)(r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\phi-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'), \tag{15}$$

其中,$\boldsymbol{F}_{u_{ij}}$ 中 i 和 j 分别表示 $\boldsymbol{\phi}$ 和 $\boldsymbol{\phi}'$ 在 $\boldsymbol{F}_u$ 中的行和列的序号. 当前驱特征 $\boldsymbol{\phi}'=\boldsymbol{\phi}(\boldsymbol{x}_{t+1},u_{t+1})$ 的优先级 $pri(\boldsymbol{\phi}')>pri$ 时,将 $\boldsymbol{\phi}'$ 插入优先级队列并根据优先级大小重新排序,此时参数向量的更新为

$$pri(\boldsymbol{\phi}')=|\delta|=\boldsymbol{F}_{u_{ij}}|r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'|. \quad (16)$$

2.5 算法描述

HDyna-AMR 是一个改进的 Dyna 优化算法,能在处理更复杂的强化学习任务的同时实现模型的学习,同时尽可能地提高算法的收敛速度. HDyna-AMR 主要包括 2 部分,即算法 1 和算法 2.

算法 1. HDyna-AMR 学习-规划算法.

输入:小正数 η、折扣因子 γ、步长因子 α、优先级阈值 pri、衰减因子 λ、初始温度 T_0、目标位置 $goal$,最大时间步 I_{step}、最大情节数目 I_{episode}、动作个数 M;

输出:参数向量 $\boldsymbol{\theta}$、特征迁移概率矩阵 $\boldsymbol{F}$、立即奖赏向量 $\boldsymbol{b}$.

初始化:参数向量 $\boldsymbol{\theta}=\boldsymbol{0}$,资格迹 $\boldsymbol{e}=\boldsymbol{0}$,特征迁移概率矩阵 $\boldsymbol{F}=\eta\boldsymbol{I}$($\boldsymbol{I}$ 为单位矩阵),立即奖赏向量 $\boldsymbol{b}=\boldsymbol{0}$,特征出现频率 $\chi(\boldsymbol{x},u)=\eta$,时间步 $t=0$,情节数 $s=0$,优先级队列 $que=\varnothing$.

① Repeat (对每个情节):
② $\boldsymbol{x}\leftarrow$初始状态,$t=0$;
③ Repeat (对每个时间步)
④ 计算当前状态 $\boldsymbol{x}$ 对应各动作的特征:$\boldsymbol{\phi}(\boldsymbol{x},u_i)$,$1\leqslant i\leqslant M$;
⑤ 根据式(9)选择动作 u;
⑥ 根据式(3)计算特征 $\boldsymbol{\phi}(\boldsymbol{x},u)$的 Q 值;
⑦ 计算特征 $\boldsymbol{\phi}(\boldsymbol{x},u)$的优先级:$pri(\boldsymbol{\phi})\leftarrow|r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}|$;
⑧ If $pri(\boldsymbol{\phi})\geqslant pri$:
将$\boldsymbol{\phi}(\boldsymbol{x},u)$加入 que,并根据 $pri(\boldsymbol{\phi})$对 que 排序;
End If
⑨ 更新特征出现频率:$\chi(\boldsymbol{x},u)\leftarrow\chi(\boldsymbol{x},u)+1$;
⑩ 根据式(6)(7)更新资格迹;
⑪ 执行动作 u,获得下一个状态 $\boldsymbol{x}'$和立即奖赏 r;
⑫ 计算状态 $\boldsymbol{x}'$对应各动作的特征:$\boldsymbol{\phi}(\boldsymbol{x}',u_i)$,$1\leqslant i\leqslant M$;
⑬ 根据式(9)选择下一个动作 u';
⑭ 更新参数向量:$\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}+\alpha\boldsymbol{e}(u,d)(r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi})$;
⑮ 更新特征迁移概率矩阵和立即奖赏向量:
$$\begin{cases}\boldsymbol{F}_{u_i}\leftarrow\boldsymbol{F}_{u_i}+\alpha(\boldsymbol{\phi}'-\boldsymbol{F}_{u_i}\boldsymbol{\phi})\boldsymbol{\phi}^{\mathrm{T}},\\ \boldsymbol{b}_{u_i}\leftarrow\boldsymbol{b}_{u_i}+\alpha(r-(\boldsymbol{b}_{u_i})^{\mathrm{T}}\boldsymbol{\phi})\boldsymbol{\phi},\end{cases}\quad 1\leqslant i\leqslant M;$$
⑯ 执行算法 2 进行模型规划;
⑰ $\tau\leftarrow T_0/(t+1)$;
$a\leftarrow 1/(t+1)$;
$\boldsymbol{x}\leftarrow\boldsymbol{x}'$;
$u\leftarrow u'$;
$t\leftarrow t+1$;
⑱ Until {$\boldsymbol{x}'$为目标状态或 $t\geqslant I_{\mathrm{step}}$};
⑲ $s\leftarrow s+1$;
⑳ Until $s\geqslant I_{\mathrm{episode}}$或满足其他终止条件.

算法 2. HDyna-AMR 规划算法.

初始化:规划次数 N.

① Loop (N 次或 $que\neq\varnothing$):
② $\boldsymbol{\phi}\leftarrow que$ 中优先级最大的特征;
③ 根据式(14)计算其对应的各前驱特征 $\boldsymbol{\phi}_i'$和立即奖赏 r_i,$1\leqslant i\leqslant M$;
④ 计算前驱特征的优先级:$pri(\boldsymbol{\phi}_i')\leftarrow\boldsymbol{F}_{u_i}|r+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_i'|$;
⑤ If $pri(\boldsymbol{\phi}_i')\geqslant pri$, $1\leqslant i\leqslant M$
将$\boldsymbol{\phi}_i'$ 加入优先级队列 que 并根据 $pri(\boldsymbol{\phi}_i')$对 que 排序;
End If
⑥ Loop (所有前驱特征 $\boldsymbol{\phi}_i'(\boldsymbol{x},u_i)$,$1\leqslant i\leqslant M$):
⑦ 根据式(6)(7)更新资格迹;
⑧ 计算额外奖赏:$r'\leftarrow\eta\sqrt{\dfrac{1}{\chi(\boldsymbol{x},u_i)}}$;
⑨ 更新立即奖赏:$r_i\leftarrow r'+r_i$;
⑩ 更新参数向量:$\boldsymbol{\theta}_{t+1}\leftarrow\boldsymbol{\theta}_t+\alpha\boldsymbol{e}(u,d)(r_i+\gamma\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}_i'-\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi})$;
⑪ End Loop.
⑫ End Loop.

算法 1 作为主算法主要实现值函数学习和近似模型学习,并调用算法 2 作为其规划部分实现同一值函数的学习,最终输出最优值函数参数向量 $\boldsymbol{\theta}$、特征迁移概率矩阵 $\boldsymbol{F}$ 和立即奖赏向量 $\boldsymbol{b}$,此时对于任意状态动作对$(\boldsymbol{x},u_i)$,均可以计算其特征 $\boldsymbol{\phi}(\boldsymbol{x},u_i)$,再根据式(3)计算其 Q 值 $Q^*(\boldsymbol{x},u_i)$,agent 在任意位置 $\boldsymbol{x}$ 处最优策略 $h^*(\boldsymbol{x})$如式(17)所示:

$$h^*(\boldsymbol{x})=\arg\max_{u_i}Q^*(\boldsymbol{x},u_i),\ 1\leqslant i\leqslant M. \quad (17)$$

3　算法收敛性分析

文献[21]针对随机梯度下降算法的收敛性作了详尽的分析,并得出结论:当随机梯度下降算法满足一定的引理时,能以概率 1 收敛,并满足$\lim_{t\to\infty}\nabla J(\boldsymbol{\theta}_t)=0$. HDyna-AMR 正是一个在学习和规划阶段都采用随机梯度下降方法的算法,因此,当 HDyna-AMR 满足文献[21]中定义的收敛所需的引理时,即可说明 HDyna-AMR 的收敛性.

为了表示方便,在证明前首先对 $\boldsymbol{\theta}_t$ 进行变换:

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha_t\boldsymbol{e}_t(u,d)(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t+\boldsymbol{\theta}_t^{\mathrm{T}}(\gamma\boldsymbol{F}-\boldsymbol{I})\boldsymbol{\phi}_t). \tag{18}$$

令 $\gamma\boldsymbol{F}-\boldsymbol{I}=-\boldsymbol{D}$,可以将式(18)变换为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha_t\boldsymbol{e}_t(u,d)(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\phi}_t). \tag{19}$$

再令 $\boldsymbol{v}_t=(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\phi}_t)$,此时式(19)可变换为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha_t\boldsymbol{e}(u,d)\boldsymbol{v}_t. \tag{20}$$

假设 1. $a_t=\dfrac{1}{t+1}$为步长参数,t 是时间步.

假设 2. 势函数定义为 $J(\boldsymbol{\theta}_t)=\dfrac{1}{2}E[(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\phi}_t+\gamma\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{F}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}_t)^2]$.

假设 3. 特征迁移概率矩阵 $\boldsymbol{F}$ 满足 $r(\boldsymbol{F})=\max_{\|\boldsymbol{x}\|_2=1}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{x}\leqslant 1$.

假设 4. $\boldsymbol{\phi}_t$ 是独立同分布的且具有上界的随机变量.

假设 5. $\boldsymbol{A}=E[\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}}]$是非奇异的.

引理 1. HDyna-AMR 步长序列(a_t)满足 $\sum_{t=0}^{\infty}\alpha_t=\infty$, $\sum_{t=0}^{\infty}\alpha_t^2<\infty$.

证明. 根据假设 1 中 $a_t=\dfrac{1}{t+1}$,采用牛顿幂级数对 $\sum_{t=0}^{\infty}\alpha_t$ 展开,则:

$$\sum_{t=0}^{\infty}\alpha_t=\sum_{t=0}^{\infty}(1+1/2+\cdots+1/t)=\ln(t+1)+o, \tag{21}$$

其中,$o\doteq 0.577218$ 为欧拉常数. 由于 ln 为单调递增函数,所以当 $t\to\infty$时,满足 $\sum_{t=0}^{\infty}\alpha_t=\infty$,而 $\sum_{t=0}^{\infty}\alpha_t^2$ 如式(22)所示:

$$\sum_{t=0}^{\infty}\alpha_t^2=\sum_{t=0}^{\infty}(1^2+(1/2)^2+\cdots+(1/t)^2)<(2t-1)/t=2-1/t, \tag{22}$$

其中,不等式部分可通过归纳法证明,因而当 $t\to\infty$时,满足 $\sum_{t=0}^{\infty}\alpha_t^2<\infty$,因此,引理 1 得证. 证毕.

引理 2. 势函数 $J(\boldsymbol{\theta})$是非负的.

证明. 根据假设 2 中给出的势函数:$J(\boldsymbol{\theta}_t)=\dfrac{1}{2}E[(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\phi}_t+\gamma\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{F}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}_t)^2]$,可知 $J(\boldsymbol{\theta}_t)\geqslant 0$,因此,引理 2 得证. 证毕.

引理 3. $\nabla J(\boldsymbol{\theta}_t)$的 Lipschitz 连续性.

证明. $\nabla J(\boldsymbol{\theta}_t)$的 Lipschitz 连续性即证明式(23)成立:

$$\frac{\|\nabla J(\boldsymbol{\theta}_1)-\nabla J(\boldsymbol{\theta}_2)\|^2}{\|(\boldsymbol{\theta}_1-\boldsymbol{\theta}_2)\|^2}\leqslant\omega, \tag{23}$$

其中,ω 为某一确定常数.

将 $J(\boldsymbol{\theta}_t)=\dfrac{1}{2}E[(\boldsymbol{b}^{\mathrm{T}}\boldsymbol{\phi}_t+\gamma\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{F}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{\phi}_t)^2]$对参数 $\boldsymbol{\theta}_t$ 求导,得到的$\nabla J(\boldsymbol{\theta}_t)$:

$$\nabla J(\boldsymbol{\theta}_t)=(\gamma\boldsymbol{F}-\boldsymbol{I})(E(\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}})\boldsymbol{b}-E(\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}})(\boldsymbol{I}-\gamma\boldsymbol{F})^{\mathrm{T}}\boldsymbol{\theta}_t). \tag{24}$$

将 $\boldsymbol{A}=E(\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}})$和 $\boldsymbol{D}=-(\gamma\boldsymbol{F}-\boldsymbol{I})$代入式(24),得到:

$$\nabla J(\boldsymbol{\theta}_t)=-\boldsymbol{D}(\boldsymbol{A}\boldsymbol{b}-\boldsymbol{A}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{\theta}_t). \tag{25}$$

此时,$\|\nabla J(\boldsymbol{\theta}_1)-\nabla J(\boldsymbol{\theta}_2)\|^2$ 可以表示为

$$\|\nabla J(\boldsymbol{\theta}_1)-\nabla J(\boldsymbol{\theta}_2)\|^2=\|\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{\mathrm{T}}(\boldsymbol{\theta}_1-\boldsymbol{\theta}_2)\|^2\leqslant\|\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{\mathrm{T}}\|\|\boldsymbol{\theta}_1-\boldsymbol{\theta}_2\|^2. \tag{26}$$

假设 3 成立时,特征迁移概率矩阵 $\boldsymbol{F}$ 满足,$r(\boldsymbol{F})=\max_{\|x\|_2=1}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{x}\leqslant 1$,$\boldsymbol{D}=-(\gamma\boldsymbol{F}-\boldsymbol{I})$中元素均小于 1,而 $\boldsymbol{A}=E(\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}})$,且假设 4 成立时,即$(\boldsymbol{\phi}_t)$是独立同分布的且具有上界的随机变量,此时 $\boldsymbol{A}=E(\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}})$有界,因此,$\|\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{\mathrm{T}}\|$为确定有界的常数,引理 3 得证. 证毕.

引理 4. 参数向量 $\boldsymbol{\theta}_t$ 期望更新方向的伪随机性.

证明. 参数向量 $\boldsymbol{\theta}_t$ 期望更新方向的伪随机性为

$$c\|\nabla J(\boldsymbol{\theta}_t)\|_2^2\leqslant-\nabla J(\boldsymbol{\theta}_t)^{\mathrm{T}}E[\boldsymbol{v}_t|\boldsymbol{\theta}_t], \tag{27}$$

其中,c 为正常数,定义 $\boldsymbol{v}_t'=E[\boldsymbol{v}_t|\boldsymbol{\theta}_t]=\boldsymbol{A}\boldsymbol{b}-\boldsymbol{A}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{\theta}_t$,此时根据前面的证明可知$\nabla J(\boldsymbol{\theta}_t)=-\boldsymbol{D}\boldsymbol{v}_t'$,因此,$\|\nabla J(\boldsymbol{\theta}_t)\|_2^2=(\boldsymbol{v}_t')^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{v}_t')$,此时伪随机性的证明可以转换为证明 $c(\boldsymbol{v}_t')^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{v}_t')\leqslant(\boldsymbol{v}_t')^{\mathrm{T}}\boldsymbol{D}\boldsymbol{v}_t'$ 成立,其要求对于任意非零向量 $\boldsymbol{v}$,均有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{v}>0$,这就等价于假设 3,即 $r(\boldsymbol{F})=\max_{\|x\|_2=1}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{x}\leqslant 1$ 时,由此,引理 4 得证. 证毕.

引理 5. 更新期望值有界,即 $E[\|(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t-\boldsymbol{\theta}_t^{\mathrm{T}}(\boldsymbol{I}-\gamma\boldsymbol{F})\boldsymbol{\phi}_t)\boldsymbol{\phi}_t\|_2^2|\boldsymbol{\theta}_t]\leqslant O(\|\nabla J(\boldsymbol{\theta}_t)\|_2^2)$.

证明. 令 $\boldsymbol{v}_t' = E[\boldsymbol{v}_t \mid \boldsymbol{\theta}_t] = \boldsymbol{Ab} - \boldsymbol{AD}^{\mathrm{T}}\boldsymbol{\theta}_t$, 由于 $E[\|(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t - \boldsymbol{\theta}_t^{\mathrm{T}}(\boldsymbol{I}-\gamma\boldsymbol{F})\boldsymbol{\phi}_t)\boldsymbol{\phi}_t\|_2^2 \mid \boldsymbol{\theta}_t] \leqslant O(\|\nabla J(\boldsymbol{\theta}_t)\|_2^2)$等价于 $E[\|\boldsymbol{v}_t\|_2^2 \mid \boldsymbol{\theta}_t] \leqslant O(\|\nabla J(\boldsymbol{\theta}_t)\|_2^2)$, 而 $E[\|\boldsymbol{v}_t\|_2^2 \mid \boldsymbol{\theta}_t] = (\boldsymbol{v}_t')^{\mathrm{T}}(\boldsymbol{v}_t')$, $\|\nabla J(\boldsymbol{\theta}_t)\|_2^2 = (\boldsymbol{v}_t')^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{v}_t')$, 而 $\boldsymbol{D} = -(\gamma\boldsymbol{F}-\boldsymbol{I})$中元素均小于 1, 因此, $E[\|(\boldsymbol{b}_t^{\mathrm{T}}\boldsymbol{\phi}_t - \boldsymbol{\theta}_t^{\mathrm{T}}(\boldsymbol{I}-\gamma\boldsymbol{F})\boldsymbol{\phi}_t)\boldsymbol{\phi}_t\|_2^2 \mid \boldsymbol{\theta}_t] \leqslant O(\|\nabla J(\boldsymbol{\theta}_t)\|_2^2)$, 引理 5 得证. 证毕.

定理 1. 在满足假设 1～5 条件下, HDyna-AMR 能以概率 1 收敛, 且收敛的解为 $\lim\limits_{t\to\infty}\boldsymbol{\theta}_t = (\boldsymbol{D}^{\mathrm{T}})^{-1}\boldsymbol{b} = (\boldsymbol{I}-\gamma\boldsymbol{F}^{\mathrm{T}})^{-1}\boldsymbol{b}$.

证明. 由文献[21]可知, 梯度下降算法如果满足引理 1～5, 则这类算法能以概率 1 收敛, 并满足 $\lim\limits_{t\to\infty}\nabla J(\boldsymbol{\theta}_t) = 0$.

由于 HDyna-AMR 在假设 1～4 满足的条件下, 能使得引理 1～5 均成立, 因此, HDyna-AMR 能以概率 1 收敛. 此时将 $\nabla J(\boldsymbol{\theta}_t) = -\boldsymbol{D}(\boldsymbol{Ab}-\boldsymbol{AD}^{\mathrm{T}}\boldsymbol{\theta}_t)$代入 $\lim\limits_{t\to\infty}\nabla J(\boldsymbol{\theta}_t) = 0$ 中. 由于假设 5 中说明 $\boldsymbol{A} = E[\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}}]$ 是非奇异的, 即 $\boldsymbol{A}$ 是满秩的, $r(\boldsymbol{D}) \leqslant 1$, $\boldsymbol{A}$ 和 $\boldsymbol{D}$ 均为可逆矩阵, 因此, $\lim\limits_{t\to\infty}\boldsymbol{\theta}_t = (\boldsymbol{D}^{\mathrm{T}})^{-1}\boldsymbol{b} = (\boldsymbol{I}-\gamma\boldsymbol{F}^{\mathrm{T}})^{-1}\boldsymbol{b}$. 定理 1 得证. 证毕.

定理 2. HDyna-AMR 算法中由特征、奖赏和下一个特征组成的 Markov 链为 $l_{\mathrm{MDP}} = \boldsymbol{\phi}_1, r_2, \boldsymbol{\phi}_2, r_3, \boldsymbol{\phi}_3, r_4, \cdots, \boldsymbol{\phi}_n$, 其最小二乘时间差分(least square temporal difference, LSTD)的特征迁移概率函数和期望奖赏分别为 $\boldsymbol{F}$ 和 $\boldsymbol{b}$. 假设 $\boldsymbol{A} = \sum\limits_{t=1}^{n}\boldsymbol{\phi}_t\boldsymbol{\phi}_t^{\mathrm{T}}$ 是满秩的, 则 HDyna-AMR 的唯一解 $(\boldsymbol{I}-\gamma\boldsymbol{F}^{\mathrm{T}})^{-1}\boldsymbol{b}$ 与 LSTD 在该 MDP 模型上的解为相同解.

证明. $l_{\mathrm{MDP}} = \boldsymbol{\phi}_1, r_2, \boldsymbol{\phi}_2, r_3, \boldsymbol{\phi}_3, r_4, \cdots, \boldsymbol{\phi}_n$ 在 LSTD 上的解为

$$\sum_{t=1}^{n}\boldsymbol{\phi}_t(r_t + \gamma(\boldsymbol{\phi}_t')^{\mathrm{T}}\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^{\mathrm{T}}\boldsymbol{\theta}_t) = 0. \tag{28}$$

令 $\boldsymbol{B} = \sum\limits_{t=1}^{n}\boldsymbol{\phi}_t(\boldsymbol{\phi}_{t+1})^{\mathrm{T}}$, $\bar{\boldsymbol{r}} = \sum\limits_{t=1}^{n}\boldsymbol{\phi}_t r_t$, 并代入式(28), 得到:

$$\bar{\boldsymbol{r}} + \gamma\boldsymbol{B}\boldsymbol{\theta} + \boldsymbol{A}\boldsymbol{\theta} = 0. \tag{29}$$

将式(29)两边同时乘以 $\boldsymbol{A}^{-1}$, 得到:

$$\boldsymbol{A}^{-1}(\bar{\boldsymbol{r}} + \gamma\boldsymbol{B}\boldsymbol{\theta} - \boldsymbol{A}\boldsymbol{\theta}) = 0. \tag{30}$$

由于 $\boldsymbol{F}^{\mathrm{T}} = \boldsymbol{A}^{-1}\boldsymbol{B}$, $\boldsymbol{b} = \boldsymbol{A}^{-1}\bar{\boldsymbol{r}}$, 因此, 式(30)可以表示为

$$\boldsymbol{b} + \gamma\boldsymbol{F}^{\mathrm{T}}\boldsymbol{\theta} - \boldsymbol{\theta} = 0. \tag{31}$$

求解式(31), 可以得到 $\boldsymbol{\theta} = (\boldsymbol{I}-\gamma\boldsymbol{F}^{\mathrm{T}})^{-1}\boldsymbol{b}$, 由此, 定理 2 得证. 证毕.

4 实验结果与分析

4.1 实验描述

为了对文中方法进行验证, 采用强化学习中的 2 个经典的情节式任务问题 Boyan chain 和 Mountain car 进行实验. 本文的 Boyan chain 问题是对经典 Boyan chain[22-23]离散型任务问题的扩展, 将状态个数扩展到 150 个状态, 每个情节从状态 150 开始运行并在状态 0 处终止. 所有的状态 $x>2$ 均有相等的转移概率转移到 $x-2$ 和 $x-1$, 获得的立即奖赏均为 -3; 状态 2 和状态 1 均以概率 1 转移到状态 0, 并分别获得 -2 和 0 的立即奖赏, 其示意图可以表示为如图 1 所示:

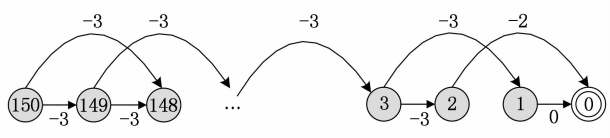


Fig. 1 Boyan chain.

图 1 链问题实验

Mountain car 是强化学习中一个经典的连续状态空间的情节式任务, 每个情节开始于坡底位置, 终止于坡顶目标位置, 如图 2 中右边顶端的五角星所示:

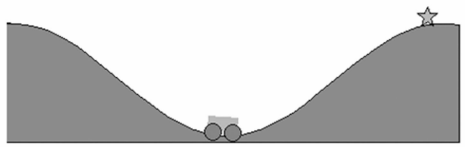


Fig. 2 Mountain car.

图 2 Mountain car 问题实验

在图 2 中, 一辆动力不足的小车试图爬到坡顶目标位置. 小车初始时位于坡底, 由于动力不足, 小车无法直接通过油门加速到达坡顶. 小车的状态采用二维向量来表示, 包含水平方向的位置和速度, 分别采用变量 p 和 v 表示. 小车可能发生的动作有 3 个: 向右加速、向左加速和不加速, 分别采用常量 $+1$, -1, 0 表示. 在时刻 t 时, 小车的状态为 $\boldsymbol{x}_t = [p, v]^{\mathrm{T}}$, 时刻 $t+1$ 时小车的状态如式(32)所示:

$$\begin{cases} v_{t+1} = bound[v_t + 0.001u_t + g\cos(3p_t)], \\ p_{t+1} = bound[p_t + v_{t+1}], \end{cases} \tag{32}$$

其中, $bound$ 表示限界函数, 小车水平位置的界限表示为 $bound(p_t) \in [-1.2, +0.5]$, 小车速度的界限

可以表示为 $bound(v_t)\in[-0.07,+0.07]$. g 为重力加速度,$g=-0.0025$. 在时刻 $t+1$ 时,当小车到达目标位置时,环境反馈的立即奖赏为 1,否则为 0. 奖赏函数为

$$\rho_t(\boldsymbol{x}_t,u_t,\boldsymbol{x}_{t+1})=\begin{cases}0, & p_{t+1}=0.5,\\ -1, & -1.2<p_{t+1}<0.5.\end{cases}\tag{33}$$

4.2　Boyan chain 实验结果分析

采用一维的高斯径向基函数对状态空间进行编码. 高斯径向基函数的个数为 40,函数形式为

$$\phi(\boldsymbol{x},\bar{\boldsymbol{x}}_i)=\exp\left(-\frac{\|\boldsymbol{x}-\bar{\boldsymbol{x}}_i\|^2}{2\sigma}\right),\tag{34}$$

其中,$\bar{\boldsymbol{x}}_i$ 是第 i 个基函数的中心,并满足 $1\leqslant i\leqslant 40$. 为了对文中方法进行验证,将文中算法 HDyna-AMR 与 Dyna-LAPS[15]和 Sarsa(λ)[24]进行比较. 这 3 种算法均采用式(34)所示的径向基函数进行编码,方差 $\sigma=0.9$. 3 种算法共同设置的参数为 $a_t=\frac{1}{t+1}$, $I_{\text{episode}}=200$, $I_{\text{step}}=100$, $goal=0$, $\lambda=0.9$ 和 $\gamma=0.8$,其他参数设置如表 1 所示:

Table 1　Parameters Setting in Different Algorithms

表 1　算法参数设置

Parameters	HDyna-AMR	Dyna-LAPS	Sarsa(λ)
η	0.4	0.4	
pri	−0.5	−0.5	
T_0	200		
K	50	50	
ε		0.1	0.1

参数设置的具体依据如下:小正数的作用是初始化特征迁移概率矩阵以及通过式(11)计算额外奖赏,通常取为(0,1)之间的正数. 为了比较的公平性,文中算法和 Dyna-LAPS 算法中的小正数均初始化为 $\eta=0.4$.

折扣因子和衰减因子是强化学习中的常用因子,两者的经验取值范围为(0.8,0.95). 为了比较的公平性,实验中 3 种算法的折扣因子 $\gamma=0.8$,衰减因子 $\lambda=0.9$. 步长因子取值如引理 1 中所示为 $a_t=\frac{1}{t+1}$.

优先级阈值的设置需要根据具体的问题选定一个合适的值,如在 Boyan chain 实验中最大的立即奖赏为 0,最小的立即奖赏为−3,因此,Boyan chain 中的优先级阈值大致位于[−3,0]之间. 下面在其他参数都赋值的情况下,对优先级取−3,−2,−1,−0.5,0 时的算法收敛性能进行比较,得到的收敛性能结果如图 3 所示:

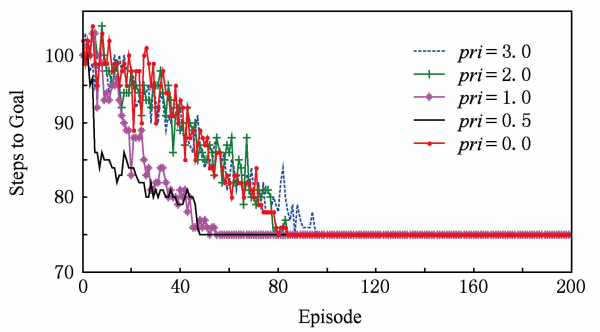


Fig. 3　Comparison of convergence performance for different priority thresholds in Boyan chain.

图 3　Boyan chain 中不同优先级阈值的收敛性能比较

从图 3 可以看出,当优先级阈值取−1,−0.5 时的收敛效果较好,其中,优先级阈值取−0.5 时具有最好的收敛效果,因此,在 Boyan chain 实验中优先级阈值为 $pri=-0.5$.

初始温度的设置. 本文算法采用软最大化方法进行动作选择概率分配,温度与动作选择概率之间的关系如式(9)所示,温度随着迭代次数更新如式(10)所示. 初始温度越大,算法的探索能力越强,能避免算法陷入局部最优. 然而,初始温度设置过大会带来算法时间复杂度的增加并降低算法收敛速度. Boyan chain 实验中初始温度取不同值对应的算法收敛性能比较如图 4 所示:

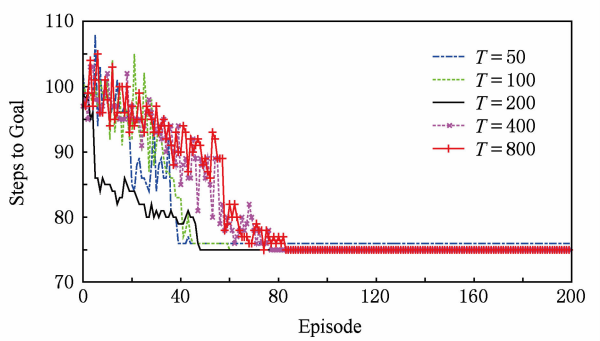


Fig. 4　Comparison of convergence performance for different initial temperatures in Boyan chain.

图 4　Boyan chain 中不同初始温度的收敛性能比较

从图 4 可以看出,当初始温度 $T_0=200$ 时,本文算法在情节 48 时收敛到 75 个时间步,具有最好的收敛性能,因此,本文算法中的初始温度为 $T_0=200$.

在采用表 3 所示的参数设置的基础上,将 3 种算法运行 50 次并对达到目标所需要的时间步作平均,结果如图 5 所示:

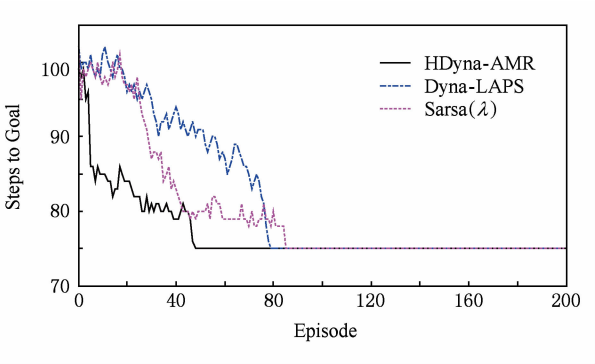


Fig. 5 Comparison of convergence performance in Boyan chain.

图 5 Boyan chain 收敛性能比

从图 5 可以看出,本文提出的算法 HDyna-AMR 在情节(episode)48 时就已收敛到 75 个时间步. Dyna-LAPS 在情节 79 时趋于收敛;Sarsa(λ)在情节 85 处收敛. 由此可知,HDyna-AMR 收敛性能最优,Dyna-LAPS 次之,而 Sarsa(λ)收敛性能最差. 这是因为 HDyna-AMR 和 Dyna-LAPS 均对模型进行了近似学习,并在规划阶段充分利用模型产生的模拟经验进行规划,因此,收敛速度较 Sarsa(λ)有显著改善.

在前 76 个情节中,Dyna-LAPS 的收敛效果并不优于 Sarsa(λ),这是因为 Dyna-LAPS 在更新中采用 1 步 TD 更新方式,因此,学习速度较慢. 同时在模型学习时对所有动作均学习同一个特征迁移概率矩阵和立即奖赏向量,使得近似模型的学习不够精确,因此,前期收敛性能不如 Sarsa(λ). HDyna-AMR 较 Dyna-LAPS 的收敛性能更优,主要是因为 HDyna-AMR 在学习和规划阶段都引入了资格迹,实现当前的 TD 误差的前向传递和更新;同时 HDyna-AMR 对每个动作都近似表示对应的特征迁移概率矩阵和立即奖赏向量,因此,模型表达更为精确. 此外 HDyna-AMR 在近似值函数时采用 Q 值,避免了 V 值到 Q 值的进一步映射,因此,具有较快的收敛速度.

为了验证环境动态性发生变化对 3 种方法收敛性的影响以及比较 3 种算法对模型及时修正的能力,假设 Boyan chain 在情节 200 时环境动态性发生变化,即状态 100 以概率 1 转移到状态 0,此时 agent 达到目标状态并获得值为 10 的立即奖赏. 前 400 个情节中 3 种方法得到的累积奖赏随着情节变化的情况如图 6 所示:

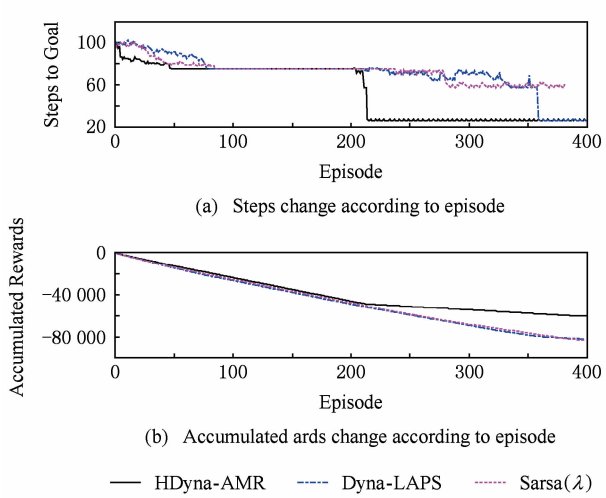


Fig. 6 Comparison of time step and accumulated reward in Boyan chain.

图 6 Boyan chain 时间步和累积奖赏比较图

从图 6 可以看出,HDyna-AMR 在 400 个情节内累积奖赏都高于 Dyna-LAPS 和 Sarsa(λ). 在前 200 个情节中,3 种算法的累积奖赏变化率相差不大,但在情节 213 后,HDyna-AMR 累积奖赏的减少较 Dyna-LAPS 和 Sarsa(λ)要慢很多(由于大部分情况下,立即奖赏为负值). 这表明 HDyna-AMR 已经寻求到环境动态性变化后的近似最优解为 26 个时间步,并重新学习到了一个较为精确的近似模型.

Dyna-LAPS 和 Sarsa(λ)分别在情节 359 和情节 386 后才重新寻求到近似最优解为 26 个时间步,此时两者的累积奖赏变化斜率与 HDyna-AMR 方法相同,表明了 HDyna-AMR 对系统动态性变化具有较大的鲁棒性. 这是由于 HDyna-AMR 对出现频率较少的特征引入了额外奖赏,从而提高这些特征在学习阶段被选择的概率,使得 HDyna-AMR 更容易探测环境发生的变化,从而修正已有的模型,最终加快算法的收敛速度. Dyna-LAPS 和 Sarsa(λ)均采用 ε-greedy 策略,分别在情节 79 和情节 85 处收敛后,以较低的概率去执行非最优动作,因此,直到情节 359 和情节 386 后才寻求到环境动态性发生的变化.

4.3 Mountain car 实验结果分析

由于时刻 t 小车的状态为 $\boldsymbol{x}_t=[p_t,v_t]^{\mathrm{T}}$,因此,采用二维的高斯径向基函数对小车状态进行编码,将状态的 2 个维度分别分成 6 份,一共包含 36 个径向基函数. 高斯径向基函数为

$$\phi(\boldsymbol{x},\bar{\boldsymbol{x}}_i)=\exp\left(-\frac{(p-\bar{p}_i)(v-\bar{v}_i)}{2\sigma^2}\right), \quad (35)$$

其中,$\bar{\boldsymbol{x}}_i$ 是第 i 个基函数的中心,$\bar{\boldsymbol{x}}_i=[\bar{p}_i,\bar{v}_i]^{\mathrm{T}}$.

将 HDyna-AMR,Dyna-LAPS 和 Sarsa(λ)进行比较,3 种算法均采用式(35)所示高斯径向基函数进行编码,方差 $\sigma=1$. 这 3 种方法的参数设置与 Boyan chain 实验基本相同,其不同的参数为:优先级阈值 $pri=0.1$,目标 $goal=0.5$,最大时间步 $I_{\text{step}}=800$.

为了确定规划次数对实验结果的影响,从而选择较优的规划次数,分别设置 $K=5,10,50,100$ 并运行 HDyna-AMR,达到目标所需要的平均时间步随情节变化的性能比较如图 7 所示:

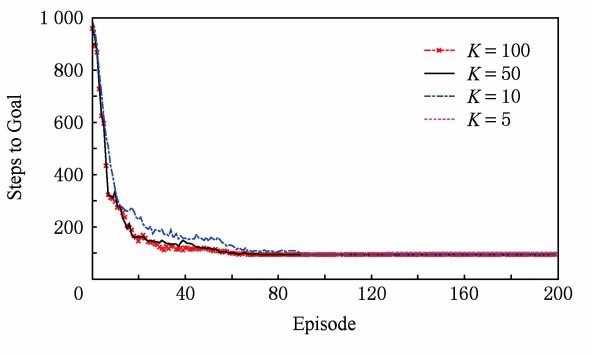


Fig. 7　Effect of K value to the convergence performance.

图 7　K 值对收敛性能的影响

从图 7 可以看出,HDyna-AMR 算法在不同的 K 值处具有不同的收敛效果. 当 $K=5$ 时,在情节 101 时收敛到 102 个时间步;当 $K=10$ 时,在情节 91 时收敛到 92 个时间步;当 $K=50$ 时,在情节 72 时收敛到 92 个时间步;当 $K=100$ 时,在情节 67 时收敛到 92 个时间步. 由此可见,当 $K=5$ 时,由于规划的次数过少,规划过程对最优策略的获取的影响有限,因此,收敛速度慢,同时得到的是一个局部最优解为 102 个时间步;当 K 值增大到 10 时,收敛速度加快,同时得到一个较 102 个时间步更好的解 92 个时间步;当 K 值增大到 50 时,收敛所需要的情节数由 91 个减少到 72 个,收敛速度显著改善;当 $K=100$ 时,收敛所需要的情节数为 67 个,收敛的时间步与 $K=10$ 和 $K=50$ 相同,收敛速度较 $K=50$ 得到轻微改善,但是由于每个时间步都要进行 100 次规划,因此计算开销大大增加,综合上述因素,文中选择规划次数 $K=50$.

为了进一步验证 HDyna-AMR 的优越性,在大规模连续状态空间实验 Mountain car 中,将 HDyna-AMR,Dyna-LAPS,Sarsa(λ)再次进行比较,达到目标状态所需时间步随情节变化的收敛情况如图 8 所示:

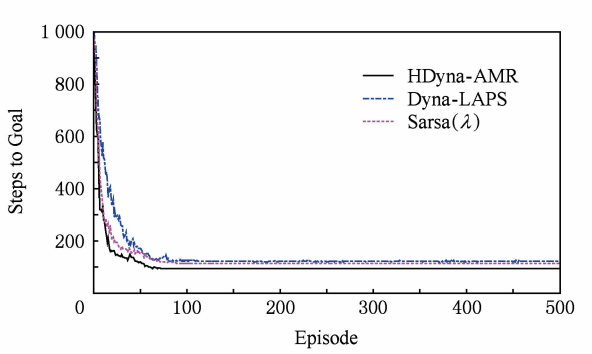


Fig. 8　Comparison of convergence performance in Mountain car.

图 8　Mountain car 收敛性能比较图

从图 8 可以看出,HDyna-AMR 方法在情节 72 时收敛,并收敛到最优解为 92 个时间步;Dyna-LAPS 在情节 86 时收敛到 121 个时间步;Sarsa(λ)在情节 89 时收敛到 113 个时间步. 显然,HDyna-AMR 无论在收敛速度还是在收敛效果上,均优于 Dyna-LAPS 和 Sarsa(λ). 这是由于在连续状态空间中,Dyna-LAPS 对所有动作学习相同的特征概率迁移矩阵和立即奖赏向量,导致学习的模型不够精确,从而收敛到一个局部最优值. Sarsa(λ)由于没有引进规划机制,因此收敛速度较 HDyna-AMR 和 Dyna-LAPS 慢,同时又由于其动作选择机制采用 ε-greedy,不能及时感知环境动态性发生的变化,因此陷入了局部最优.

5　结束语

经典的 Dyna 优化算法采用查询表存储 Q 值或 V 值,难以适用于大规模状态空间的中最优策略求解需求. 为此,本文将近似模型和启发式方法引入 Dyna 优化算法,提出 HDyna-AMR 算法,并对 HDyna-AMR 的收敛性作出理论证明,证明了 HDyna-AMR 能与 LSTD 收敛到唯一的相同解. 将 HDyna-AMR 算法应用于扩展的 Boyan Chain 和 Mountain car 经典的强化学习问题. 在实验中对 HDyna-AMR,Dyna-LAPS,Sarsa(λ)从多个角度进行了比较和分析. 实验结果表明了 HDyna-AMR 相对 Dyna-LAPS 和 Sarsa(λ),具有更快的收敛速度,尤其是在环境发生变化时能较快发现模型的错误,对模型进行及时修正,从而更新最优策略并最大化长期的累积奖赏. 通过对算法中规划次数 K 的调整发现,当 K 值较小时,值函数的收敛性能随着 K 值的增加改善得很快;当 K 增加到一定值时,值函数

的收敛性能随着 K 值的增加不再明显提高,此时若再增加 K 值反而会带来较大的额外计算开销,因此,规划次数 K 的设定需要权衡收敛速度和计算开销.

本文在对学习模型时采用线性近似方式,而在实际的问题中,大部分均为非线性模型.非线性近似模型的建立和训练需要较大的时空开销,从而影响了算法的收敛速度.如何寻找一个更为通用的模型近似方式,能有效实现各类模型的精确学习以及提高值函数的收敛性能是下一步要解决的工作.

参 考 文 献

[1] Sutton R S, Barto A G. Reinforcement Learning: An Introduction [M]. Cambridge, MA: MIT Press, 1998

[2] Lee D, Seo H, Jung M W. Neural basis of reinforcement learning and decision making [J]. Annual Review Neuroscience, 2012, 35: 287-308

[3] Fu Qiming, Liu Quan, Wang Hui, et al. A novel off policy $Q(\lambda)$ algorithm based on linear function approximation [J]. Computer Journal of Computers, 2014, 37(3): 677-686 (in Chinese)
(傅启明, 刘全, 王辉等. 一种基于线性函数逼近的离策略 $Q(\lambda)$ 算法[J]. 计算机学报, 2014, 37(3): 677-686)

[4] Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: A survey [J]. International Journal of Robotics Research, 2013, 32(11): 1238-1274

[5] Xu X, Zuo L, Huang Z H. Reinforcement learning algorithms with function approximation: Recent advances and applications [J]. Information Sciences, 2014, 261(10): 1-31

[6] Shiner T, Seymour B, Wunderlich K, et al. Dopamine and performance in a reinforcement learning task: Evidence from parkinson's disease [J]. Brain, 2012, 135(6): 1871-1883

[7] Silver D, Sutton R S, Muller M. Temporal-difference search in computer Go [J]. Machine Learning, 2012, 87(2): 183-219

[8] Badre D, Frank M J. Mechanism of hierarchical reinforcement learning in corticostriatal circuits 2: Evidence from FMRI [J]. Cerebral Cortex March, 2012, 3(22): 527-536

[9] Szepesvari C. Algorithms for Reinforcement Learning [M]. San Rafael, CA: Morgan & Claypool, 2010

[10] Sutton R S. Planning by incremental dynamic programming [C] //Proc of the 8th Int Workshop on Machine Learning. San Francisco, CA: Morgan Kaufman, 1991: 353-357

[11] Sutton R S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming [C] //Proc of the 7th Int Conf on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1990: 216-224

[12] Sutton R S. Dyna, an integrated architecture for learning, planning, and reacting [J]. ACM SIGART Bulletin, 1991, 2(4): 160-163

[13] Moore A W, Atkeson C G. Prioritized sweeping: Reinforcement learning with less data and less real time [J]. Machine Learning, 1993, 1(13): 103-130

[14] Peng J, Williams R J. Efficient learning and planning within the Dyna framework [J]. Adaptive Behavior, 1993, 1(4): 437-454

[15] Sutton R S, Szepesvar C, Geramifard A et al. Dyna-style planning with linear function approximation and prioritized sweeping [C] //Proc of the 24th Conf on Uncertainty in Artificial Intelligence. Helsinki, Finland: AUAI, 2008: 528-536

[16] Santos M, Martin J A, Lopez V, et al. Dyna-H: A heuristic planning reinforcement learning algorithm applied to role-playing game strategy decision systems [J]. Knowledge Based Systems, 2012, 32(1): 28-36

[17] Sun Hongkun, Liu Quan, Fu Qiming, et al. An optimized Dyna architecture algorithm with prioritized sweeping [J]. Journal of Computer Research and Development, 2013, 50(10): 2176-2184 (in Chinese)
(孙洪坤, 刘全, 傅启明, 等. 一种优先级扫描的 Dyna 结构优化算法[J]. 计算机研究与发展, 2013, 50(10): 2176-2184)

[18] Yu Jun, Liu Quan, Fu Qiming, et al. Bayesian Q learning method with Dyna architecture and prioritized sweeping [J]. Journal on Communications, 2013, 11(34): 129-139 (in Chinese)
(于俊, 刘全, 傅启明, 等. 基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法[J]. 通信学报, 2013, 11(34): 129-139)

[19] Sutton R S, Mcallester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation [C] //Proc of the 13th Annual Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 1999: 1057-1063

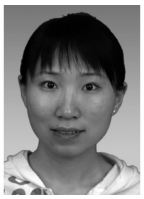
[20] Chang H S, Fu M C, Hu J, et al. Simulation-Based Algorithms for Markov Decision Processes [M]. Berlin: Springer, 2007

[21] Bertsekas D P, Tsitsiklis J. Neuro-Dynamic Programming [M]. Belmont, MA: Athena Scientific, 1996

[22] Boyan J A. Least-square temporal difference learning [C] // Proc of the 16th Int Conf on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1999: 49-56

[23] Boyan J A. Technical update: Least-squares temporal difference learning [J]. Machine Learning, 2002, 49(23): 233-246

[24] Sutton R S, Hamid R M, Precup D, et al. Fast gradient-descent methods for temporal-difference learning with linear function approximation [C] //Proc of the 26th Int Conf on Machine Learning. New York: ACM, 2009: 993-1000



Zhong Shan, born in 1983. PhD candidate, lecturer. Her research interests include machine learning and deep learning (sunshine-620@163.com).

Liu Quan, born in 1969. Post doctor, professor and PhD supervisor. Senior member of China Computer Federation. His main research interests include intelligence information processing, automated reasoning and machine learning.

Fu Qiming, born in 1985. PhD. His main research interests include reinforcement learning, Bayesian method and genetic algorithm (fqm_1@126.com).

Zhang Zongzhang, born in 1985. PhD, associate professor. His main research interests include partially observable Markov decision processes, reinforcement learning, and multi-agent systems (zzzhang@suda.edu.cn).

Zhu Fei, born in 1978. PhD, associate professor. Member of China Computer Federation. His main research interests include reinforcement learning, text mining, and pattern recognition (zhufei@suda.edu.cn).

Gong Shengrong, born in 1966. PhD, professor and PhD supervisor. His main research interests include reinforcement learning and image processing (shrgong@suda.edu.cn).