基于自适应归一化 RBF 网络的 Q-V 值函数协同逼近模型

刘 全[1),2),3)] 肖 飞[1)] 傅启明[1)] 伏玉琛[1)] 周小科[1)] 朱 斐[1),2)]

[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 长春 130012)

[3)](江苏省软件新技术与产业化协同创新中心 南京 210046)

摘 要 径向基函数网络逼近模型可以有效地解决连续状态空间强化学习问题.然而,强化学习的在线特性决定了 RBF 网络逼近模型会面临“灾难性扰动”,即新样本作用于学习模型后非常容易对先前学习到的输入输出映射关系产生破坏.针对 RBF 网络逼近模型的“灾难性扰动”问题,文中提出了一种基于自适应归一化 RBF(ANRBF)网络的 Q-V 值函数协同逼近模型及对应的协同逼近算法——$QV(\lambda)$.该算法对由 RBFs 提取得到的特征向量进行归一化处理,并在线自适应地调整 ANRBF 网络隐藏层节点的个数、中心及宽度,可以有效地提高逼近模型的抗干扰性和灵活性.协同逼近模型中利用 Q 和 V 值函数协同塑造 TD 误差,在一定程度上利用了环境模型的先验知识,因此可以有效地提高算法的收敛速度和初始性能.从理论上分析了 $QV(\lambda)$算法的收敛性,并对比其他的函数逼近算法,通过实验验证了 $QV(\lambda)$算法具有较优的性能.

关键词 强化学习;函数逼近;径向基函数;灾难性扰动;协同逼近

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2015.01386

Collaborative Q-V Value Function Approximation Model Based on Adaptive Normalized Radial Basis Function Network

LIU Quan[1),2),3)] XIAO Fei[1)] FU Qi-Ming[1)] FU Yu-Chen[1)] ZHOU Xiao-Ke[1)] ZHU Fei[1),2)]

[1)](*Institute of Computer Science and Technology, Soochow University, Suzhou, Jiangsu* 215006)

[2)](*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun* 130012)

[3)](*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing* 210046)

Abstract The radial basis function (RBF) network approximation models can effectively solve the reinforcement learning problems with continuous state space. However, the online characteristic of reinforcement learning determines that the RBF network approximation models are facing the “catastrophic interference” problem, namely the input-output mapping learned in the past is easily collapsed by the learning of new training data. In order to solve the “catastrophic interference” problem of the RBF approximation models, we proposed a collaborative Q-V value function approximation model and a corresponding collaborative algorithm named $QV(\lambda)$ based on the adaptive normalized RBF (ANRBF) network. The algorithm normalizes the feature vector generated by RBFs, and adjusts the number of the ANRBF network's hidden layer nodes, the center and width of each node online and adaptively, which can effectively improve the anti-interference capacity and flexibility of the approximation model. The collaborative approximation model uses

收稿日期:2013-12-26;最终修改稿收到日期:2014-12-28.本课题得到国家自然科学基金(61272005,61070223,61103045,61070122,61472262)、江苏省自然科学基金(BK2012616)、吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172012K04)资助.**刘 全**,男,1969 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为强化学习、智能信息处理、自动推理.E-mail:quanliu@suda.edu.cn.**肖 飞**,男,1988 年生,硕士,主要研究方向为强化学习、支持向量机.**傅启明**,男,1985 年生,博士研究生,主要研究方向为强化学习、贝叶斯推理.**伏玉琛**,男,1968 年生,博士,副教授,主要研究方向为强化学习.**周小科**,男,1976 年生,博士研究生,讲师,主要研究方向为强化学习.**朱 斐**,男,1978 年生,博士研究生,讲师,主要研究方向为强化学习、贝叶斯推理.

the Q and V value functions to shape the TD error collaboratively, which can obtain some prior knowledge of the environment model. So we can improve the convergence speed and the initial performance effectively. The convergence of $QV(\lambda)$ algorithm was analyzed theoretically. Extensive experiments were conducted to show that $QV(\lambda)$ algorithm has better performance than the other function approximation methods.

Keywords　reinforcement learning; function approximation; radial basis function; catastrophic interference; collaborative approximation

1　引　言

强化学习(Reinforcement Learning, RL)是一类由学习环境状态到动作的映射方法. 在与环境交互中,Agent 选择动作,环境作出反应,到达新的状态,并对每个状态或状态动作对通过值函数评价其好坏,最终通过值函数确定到达目标的最优策略. 目前强化学习方法被广泛地应用于工业控制、仿真、博弈等领域[1-4].

在离散状态空间强化学习系统中,值函数通常利用查询表(Lookup-Table)的方式进行存储,状态和动作是表的两个维度,其估计值与表格中的表项相对应. 这种方法非常适合于离散的小状态空间任务,而对于大的或连续空间任务,会面临"维数灾",从而导致收敛速度慢甚至无法收敛.

目前解决"维数灾"问题主要采取 3 种方法:(1)采取聚类等编码方式将任务转化成查询表强化学习可以解决的问题[5];(2)对任务进行分解,采用分层、并行等技术来提高强化学习方法的执行效率[6-8];(3)采取函数逼近方法对强化学习中的值函数或策略进行建模,利用样本来不断调整参数,逐渐逼近真实的问题模型[9].

函数逼近通常包括带参函数逼近和无参函数逼近两种[10-11]. 带参数函数逼近其函数形式和参数个数事先预定,初始模型限制了学习效果,这种逼近方法非常容易陷入局部极小值. 相比而言,无参函数逼近方法其模型根据样本的个数和形式不断调整,因此灵活性和逼近精度都大大提高.

由于带参的函数逼近模型方法简单,容易在理论上保证其收敛性,使得其应用广泛. Tadic[12] 提出了一种与线性函数逼近器相结合的时间差分(Temporal Difference, TD)学习方法. Sherstov 等人[13] 提出了一种基于粗糙编码的在线自适应线性函数逼近方法. Sutton 等人[14] 将梯度下降线性函数逼近器结合于时间差分算法中,提出了一种梯度时间差分学习方法. 而后,Sutton 等人[15] 对梯度时间差分算法做进一步改进,提出了 GTD2 及带有梯度校正的线性时间差分方法(linear TD with gradient Correction, TDC). Sutton 等人提出的系列时间差分方法使得离策略时间差分学习算法的不稳定问题,在一定程度上得到解决. 另外,Maei 等人提出了 GQ(λ)(General Q(λ))算法[16] 和 Greedy-GQ 算法[17],并于 2009 年将文献[14-15]中的线性函数逼近模型扩展到非线性函数逼近模型[18]. Bonarini 等人[19] 提出了一种基于模糊逻辑的 Q 学习算法,并验证了算法的有效性. Heinen 等人[20] 提出了一种基于概率神经网络的强化学习方法,该方法增量式地逼近问题的值函数.

目前无参函数逼近模型主要包括高斯过程函数逼近模型和核函数逼近模型两种. Engel 等人[21] 在无参函数逼近器的基础上,提出了高斯过程时间差分算法,利用高斯过程来逼近强化学习中的值函数. Ormoneit 等人[22] 提出了一种基于核的强化学习方法. Xu 等人[23] 提出了基于核的最小二乘时间差分方法(Kernel-based Least Squares TD, KLSTD),将基于核的逼近器与最小二乘相结合,取得了一定的效果. 在 KLSTD 基础上,Xu 等人[24] 提出了 KLSPI 及 KLSTD-Q 算法,并证明了算法的有效性. 而后 Taylor 等人[25] 证明了 KLSPI 及 KLSTD-Q 算法的等价性. 对于无参函数逼近模型来说,由于强化学习的样本是在线获得的,因此难以保证其收敛.

径向基函数 RBF(Radial Basis Function)网络逼近模型是一种局部逼近神经网络,可以用来解决连续空间强化学习问题. 该模型既利用了核函数等机制来提高模型的表达能力,又具有线性逼近模型的简单性. 针对 RBF 网络逼近模型应用于强化学习会出现"灾难性扰动"等问题,Barreto 等人[26] 提出了基于 RBF 网络的 RGD-TD(0)算法,经扩展得到 RGD-Sarsa(λ)等系列算法,并从理论和实验两方面验证了算法的有效性. 这些算法一定程度上解决了 RBF 网络逼近模型中的"灾难性扰动"问题,但是算法的收敛速度却不理想.

针对 RBF 网络逼近模型的有效性及存在的问题，本文提出了一种基于自适应归一化径向基函数 RBF (Adaptive Normalized RBF，ANRBF)网络的 Q-V 值函数协同逼近模型及对应的协同逼近算法——$QV(\lambda)$. 对于连续状态，利用 ANRBF 网络来进行编码和特征提取，得到特征向量. 进一步经过归一化处理，能够得到平滑的逼近模型，在一定程度上可提高模型的抗干扰性. 为了有效地提高逼近模型的灵活性，该网络逼近模型可以自适应地调整隐藏层节点的个数、宽度和中心. 另外，Q-V 值函数协同逼近模型利用状态值函数来塑造奖赏，将环境模型知识以奖赏的形式传递给学习器，从而有效地提高了算法的收敛速度和初始性能. 将 $QV(\lambda)$算法应用于连续状态空间强化学习标准验证仿真平台 Mountain Car，实验表明，$QV(\lambda)$算法在时间、空间及收敛性上都具有较优的性能.

2 相关理论

2.1 问题描述

强化学习问题通常可以用马尔可夫决策过程来建模.

定义 1. 在 t 时刻，系统的 n 个状态可以用实数向量表示为 $\boldsymbol{x}_t=[x_1(t),x_2(t),\cdots,x_n(t)]^{\mathrm{T}}\in\mathbb{R}^n$. 不考虑时刻 t，则可以表示为 $\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$.

定义 2. 设 P 为强化学习任务，P 的状态空间是由 n 维状态向量 $\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$ 各分量而构成的子状态空间 $X(P)=\{\boldsymbol{x}\mid\boldsymbol{x}=[x_1,\cdots,x_n]^{\mathrm{T}},x_i\in D_i,i=1,\cdots,n\}$，其中 D_i 是状态向量第 i 个分量的值域.

定义 3. 一个四元组 $M=\{X,U,R,T\}$ 为马尔可夫决策过程，其中 $X=\{\boldsymbol{x}_t\mid t\in\mathbb{R}\}$ 是任意状态空间(可以为连续的)，$U=\{u_i\}_{i=1}^k$，$k\geq 2$ 是离散动作空间，$R:X\times U\times X\mapsto\mathbb{R}$ 是立即奖赏函数，$T:X\times U\times X\mapsto\mathbb{R}$ 是状态迁移函数.

2.2 GD-Sarsa(λ)

线性带参 Q 值函数逼近器是通过 n 维基函数向量 $\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{u})$ 与 n 维参数向量 $\boldsymbol{w}$ 建模得到，其计算公式为

$$Q(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{u}) \tag{1}$$

其中 $\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{u})$ 为状态动作对 $\langle\boldsymbol{x},\boldsymbol{u}\rangle$ 的特征向量，$\boldsymbol{w}$ 为参数向量.

Sutton 等人在式(1)的基础上，提出了 GD-Sarsa(λ)算法[1]，该方法利用梯度下降方法来迭代更新值函数的参数向量，其迭代公式如式(2)：

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

其中 $\boldsymbol{w}_t$ 和 $\boldsymbol{w}_{t+1}$ 为参数向量. $\alpha\in[0,1]$ 为学习率. 时间差分 TD 误差为 $\delta_t=\boldsymbol{r}_{t+1}+\gamma Q_t(\boldsymbol{x}_{t+1},\boldsymbol{u}_{t+1})-Q_t(\boldsymbol{x}_t,\boldsymbol{u}_t)$，$\boldsymbol{r}_{t+1}$ 为立即奖赏，$\gamma\in[0,1]$ 为折扣率，$Q_t(\boldsymbol{x}_t,\boldsymbol{u}_t)$ 和 $Q_t(\boldsymbol{x}_{t+1},\boldsymbol{u}_{t+1})$ 分别为在 t 时刻状态动作对 $\langle\boldsymbol{x}_t,\boldsymbol{u}_t\rangle$ 及 $\langle\boldsymbol{x}_{t+1},\boldsymbol{u}_{t+1}\rangle$ 的估计值. 向量 $\boldsymbol{e}_t$ 为 t 时刻的资格迹，具体更新如式(3)：

$$\boldsymbol{e}_t=\gamma\lambda\boldsymbol{e}_{t-1}+\nabla_{\boldsymbol{w}_t}Q_t(\boldsymbol{x}_t,\boldsymbol{u}_t) \tag{3}$$

式(3)采用累加迹(accumulating trace)的更新方式. $\lambda\in[0,1]$ 为资格迹衰减因子. 通过对函数 f 的参数向量 $\boldsymbol{w}_t$ 的每一个分量求偏导，得到梯度向量，如式(4)：

$$\nabla_{\boldsymbol{w}_t}f(\boldsymbol{w}_t)=\left(\frac{\partial f(\boldsymbol{w}_t)}{\partial\boldsymbol{w}_t(1)},\frac{\partial f(\boldsymbol{w}_t)}{\partial\boldsymbol{w}_t(2)},\cdots,\frac{\partial f(\boldsymbol{w}_t)}{\partial\boldsymbol{w}_t(n)}\right)^{\mathrm{T}} \tag{4}$$

3 Q-V 值函数协同逼近模型

3.1 Q-V 值函数协同机制

在强化学习中，通常利用值函数来评估策略，具体可以分为动作值函数和状态值函数两种. 根据两类值函数的不同特性，提出一种协同机制来解决连续状态空间的强化学习问题.

针对连续状态空间的强化学习任务，其中 $X=\{\boldsymbol{x}\mid\boldsymbol{x}=[x_1,\cdots,x_n]^{\mathrm{T}},x_i\in D_i,i=1,\cdots,n\}$ 为连续状态空间，$U=\{u_i\}_{i=1}^k$，$k\geq 2$ 为离散动作空间，构建 Q-V 值函数协同模型，如式(5)：

$$\begin{cases}Q_i(\boldsymbol{x})=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})=\sum_{j=1}^m w_{ij}\phi_j(\boldsymbol{x}),\ \ i=1,2,\cdots,k\\ V(\boldsymbol{x})=\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})=\sum_{j=1}^m w_{lj}\phi_j(\boldsymbol{x}),\ \ l=k+1\end{cases} \tag{5}$$

其中包含 k 个 Q 值函数和 1 个 V 值函数，$\boldsymbol{\phi}(\boldsymbol{x})=[\phi_1(\boldsymbol{x}),\cdots,\phi_m(\boldsymbol{x})]^{\mathrm{T}}\in\mathbb{R}^m$ 是状态 $\boldsymbol{x}$ 的 m 维基函数特征向量，$\boldsymbol{w}_i=[w_{i1},\cdots,w_{im}]^{\mathrm{T}}\in\mathbb{R}^m$ 是第 i 个动作的 Q 值函数的权值向量，$\boldsymbol{w}_l=[w_{l1},\cdots,w_{lm}]^{\mathrm{T}}\in\mathbb{R}^m$ 是 V 值函数的权值向量.

在强化学习算法中，通常利用 TD 误差来修正值函数的参数向量，动态有效地降低函数逼近模型的“灾难性扰动”. 为了使得 TD 误差相对稳定，Ng 等人[27]利用塑造奖赏机制来重塑 TD 误差. 通过将模型知识以奖赏的形式调整 TD 误差，使得 Agent 根据新 TD 误差更新值函数，最终能够减少次优动

作的选择次数，从而加快算法的收敛速度. 另外，Randløv 等人[28]同时也指出，如果塑造奖赏机制使用不当，将会减缓算法收敛速度. 本文在文献[27-28]的基础上，提出了一种利用状态值函数对奖赏进行塑造的方法. 基于状态值函数的塑造奖赏定义如下.

定义 4.　基于状态值函数 $V:X\mapsto\mathbb{R}$ 定义一个有界实值映射 $F:X\times U\times X\mapsto\mathbb{R}$，其中 X 为任意状态空间，U 为离散动作空间，状态值函数塑造奖赏(Shaping Reward based on the State Value Function，SR-SVF)定义如下：

$$F(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{x}_{t+1})=\gamma V(\boldsymbol{x}_{t+1})-V(\boldsymbol{x}_t) \tag{6}$$

其中，$\gamma\in[0,1]$为折扣率，$V(\boldsymbol{x}_t)$和 $V(\boldsymbol{x}_{t+1})$分别为状态 $\boldsymbol{x}_t$和 $\boldsymbol{x}_{t+1}$的状态值函数.

定义 5.　考虑 t 时刻的状态 $\boldsymbol{x}_t$，Agent 根据当前行为策略选择一个动作 $\boldsymbol{u}_t=\boldsymbol{u}_i\in U(1\leq i\leq k)$作用于环境，状态转移至 $\boldsymbol{x}_{t+1}$，立即奖赏值 $\boldsymbol{r}_{t+1}$，并根据行为策略选择状态 $\boldsymbol{x}_{t+1}$下的动作 $\boldsymbol{u}_{t+1}=\boldsymbol{u}_j\in U(1\leq j\leq k)$. 在 t 时刻，利用塑造奖赏机制的协同 TD 误差(Collaborative Temporal Difference Error，C-TDE)计算方法如式(7)：

$$\begin{aligned}\delta_t&=\boldsymbol{r}_{t+1}+\gamma Q_j(\boldsymbol{x}_{t+1})-Q_i(\boldsymbol{x}_t)+F(\boldsymbol{x}_t,\boldsymbol{u}_t,\boldsymbol{x}_{t+1})\\&=\boldsymbol{r}_{t+1}+\gamma Q_j(\boldsymbol{x}_{t+1})-Q_i(\boldsymbol{x}_t)+\gamma V(\boldsymbol{x}_{t+1})-V(\boldsymbol{x}_t)\\&=\boldsymbol{r}_{t+1}+\gamma(Q_j(\boldsymbol{x}_{t+1})+V(\boldsymbol{x}_{t+1}))-(Q_i(\boldsymbol{x}_t)+V(\boldsymbol{x}_t))\end{aligned} \tag{7}$$

为了简化表示，式(7)中值函数 Q_i、Q_j以及 V 都省略时间步下标 t.

由于协同逼近模型的特殊性，本文对资格迹重新定义，如式(8)：

$$\boldsymbol{e}_j(t)=\begin{cases}\max(\gamma\lambda\boldsymbol{e}_j(t-1),\nabla_{w_{ij}}Q_i(\boldsymbol{x}_t)), & j=i\text{ 或 }j=l\\ \gamma\lambda\boldsymbol{e}_j(t-1), & \text{其他}\end{cases} \tag{8}$$

其中 $j=1,2,\cdots,l$，$\boldsymbol{e}_j(t)$是在 t 时刻第 j 个子模型的权值向量的资格迹.

3.2　基于 ANRBF 网络的 Q-V 值函数协同逼近模型

在基于函数近似的强化学习算法中，状态特征的提取直接影响着所逼近的值函数模型的质量. 本文利用高斯径向基函数构建 RBF 网络，用于对状态特征进行提取，并对基函数进行归一化处理，通过引入自适应机制，最终构建一个 ANRBF 网络. 下面给出基于 ANRBF 网络的 Q-V 值函数协同逼近模型的结构图及描述，如图 1 所示.

图 1 给出的 Q-V 值函数协同逼近模型是一个 $n\times m\times(k+1)$的三层归一化带反馈机制的神经网络，该网络可以在线自适应地调整隐藏层各节点的

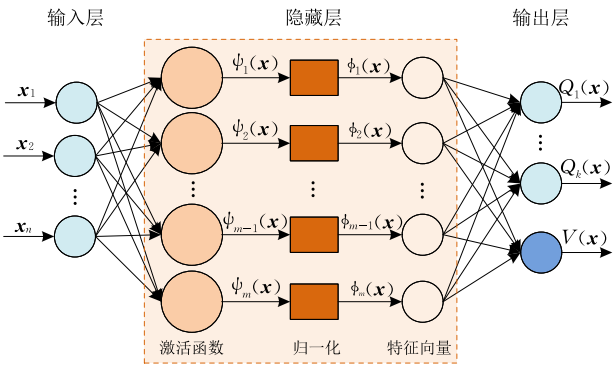


图 1　基于 ANRBF 网络的 Q-V 值函数协同逼近模型结构图

权值. 描述如下：

(1) 左边层为输入层，输入一个 n 维状态向量 $\boldsymbol{x}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]^{\mathrm{T}}\in\mathbb{R}^n$.

(2) 中间层为隐藏层，包括激活函数、归一化、特征向量 3 个部分. 激活函数共有 m 个隐节点，采用高斯径向基函数为激活函数. 激活函数定义如式(9)：

$$\psi_i(\boldsymbol{x})=\exp\left(-\sum_{j=1}^{n}\frac{(\boldsymbol{x}_j-\boldsymbol{c}_{ij})^2}{2\boldsymbol{\sigma}_{ij}^2}\right),\ i=1,2,\cdots,m \tag{9}$$

其中 n 维向量 $\boldsymbol{c}_i=[\boldsymbol{c}_{i1},\cdots,\boldsymbol{c}_{in}]^{\mathrm{T}}$、$\boldsymbol{\sigma}_i=[\boldsymbol{\sigma}_{i1},\cdots,\boldsymbol{\sigma}_{in}]^{\mathrm{T}}$分别表示第 i 个高斯径向基函数的中心及宽度. 通过归一化方法对激活函数的输出进行归一化处理得到状态的特征向量，其中特征向量的第 i 个分量如式(10)：

$$\phi_i(\boldsymbol{x})=\psi_i(\boldsymbol{x})\left[\sum_{j=1}^{m}\psi_j(\boldsymbol{x})\right]^{-1},\ i=1,2,\cdots,m \tag{10}$$

(3) 右边层为输出层，包含 $l=k+1$ 个输出，分别是 k 个动作值函数和 1 个状态值函数，值函数模型构建如式(5)所示，其中式(5)中的 $\boldsymbol{w}_{ij}$ $(i=1,2,\cdots,l,j=1,2,\cdots,m)$是输出层的权值.

基于 ANRBF 网络的 Q-V 值函数协同逼近模型的描述如下：

(1) 将状态向量 $\boldsymbol{x}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n]^{\mathrm{T}}\in\mathbb{R}^n$ 作为输入，经过式(9)所示的激活函数计算得到特征向量 $\boldsymbol{\psi}(\boldsymbol{x})\in\mathbb{R}^m$.

(2) 将 $\boldsymbol{\psi}(\boldsymbol{x})$代入式(10)所示的归一化公式进行处理，输出特征向量 $\boldsymbol{\phi}(\boldsymbol{x})\in\mathbb{R}^m$.

(3) 将 $\boldsymbol{\phi}(\boldsymbol{x})$代入式(5)所示的值函数模型，得到 Q 和 V 值函数的近似值.

(4) 根据 Q 值函数的估计值，结合动作选择策略选择一个动作，获得后续状态并立即奖赏. 利用式(6)、(7)来计算协同 TD 误差 δ. 然后根据式(8)所定义的资格迹将当前得到的协同 TD 误差反向传播至整个状态及状态动作空间，最后根据式(2)自适应地调整输出层权值及隐藏层各激活函数的中心和宽度.

协同逼近模型的动态结构主要包括 4 个方面的调整：

(1) 调整输出层权值向量 $\boldsymbol{w}_i=[\boldsymbol{w}_{i1},\boldsymbol{w}_{i2},\cdots,\boldsymbol{w}_{im}]^{\mathrm{T}}$，$i=1,2,\cdots,l$. 方法为 $\boldsymbol{w}_i=\boldsymbol{w}_i+\alpha\delta\boldsymbol{e}_i$，其中 $\boldsymbol{e}_i=[\boldsymbol{e}_{i1},\boldsymbol{e}_{i2},\cdots,\boldsymbol{e}_{im}]^{\mathrm{T}}$，资格迹的更新策略为 $\forall i=1,2,\cdots,l$，如果 $i=ID(\boldsymbol{u}_t)$ 或者 $i=l$，那么 $\boldsymbol{e}_i=\max(\gamma\lambda\boldsymbol{e}_i,\boldsymbol{\phi}(\boldsymbol{x}))$，否则 $\boldsymbol{e}_i=\gamma\lambda\boldsymbol{e}_i$. 其中 $ID(\boldsymbol{u}_t)$ 表示当前动作 $\boldsymbol{u}_t$ 在动作集合中的编号.

(2) 如果 $|\delta|>\eta_1$ 且 $\forall j=1,2,\cdots,m,\max(\phi_j(\boldsymbol{x}))<\eta_2$，则增加一个新的隐节点，$m=m+1$. 初始化该隐节点所含的 RBF 激活函数的中心向量、宽度向量以及对应的输出层权值向量：$\forall\boldsymbol{u}=1,\cdots,n,\boldsymbol{c}_{mu}=\boldsymbol{x}_u,\boldsymbol{\sigma}_{mu}=0.1,\forall i=1,\cdots,l,\boldsymbol{w}_{im}=0,\boldsymbol{e}_{im}=0$. 其中 η_1 和 η_2 为预先设定的阈值.

(3) 调整中心 $\boldsymbol{c}_j=[\boldsymbol{c}_{j1},\cdots,\boldsymbol{c}_{jn}]^{\mathrm{T}}$，$j=1,2,\cdots,m$. 假设 $\forall j=1,2,\cdots,m,\boldsymbol{c}_z=\arg\max_{\boldsymbol{c}}(\phi_j(\boldsymbol{x}))$，则 $\forall\boldsymbol{u}=1,\cdots,n$，

$$\boldsymbol{c}_{zu}=\boldsymbol{c}_{zu}-\frac{1}{2}\beta_1\frac{\partial\delta^2}{\partial\boldsymbol{c}_{zu}}=\boldsymbol{c}_{zu}+\beta_1\delta\boldsymbol{w}_{iz}(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})\frac{(\boldsymbol{x}_u-\boldsymbol{c}_{zu})}{\boldsymbol{\sigma}_{zu}^2},$$

其中 β_1 为预先设定的参数.

(4) 调整宽度 $\boldsymbol{\sigma}_j=[\boldsymbol{\sigma}_{j1},\cdots,\boldsymbol{\sigma}_{jn}]^{\mathrm{T}}$，$j=1,2,\cdots,m$. 假设 $\forall j=1,2,\cdots,m,\boldsymbol{\sigma}_z=\arg\max_{\boldsymbol{\sigma}}(\phi_j(\boldsymbol{x}))$，则 $\forall u=1,\cdots,n$，

$$\boldsymbol{\sigma}_{zu}=\boldsymbol{\sigma}_{zu}-\frac{1}{2}\beta_2\frac{\partial\delta^2}{\partial\boldsymbol{\sigma}_{zu}}=\boldsymbol{\sigma}_{zu}+\beta_2\delta\boldsymbol{w}_{iz}(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})\frac{(\boldsymbol{x}_u-\boldsymbol{c}_{zu})^2}{\boldsymbol{\sigma}_{zu}^3},$$

其中 β_2 为预先设定的参数.

上述 ANRBF 协同逼近模型的动态结构调整中，第(1)、(3)和(4)步都是采用梯度下降的方法来进行调整的，第(1)步的调整策略如式(4)和(8)所示，第(3)和(4)步调整策略的详细推导如下：

$$\boldsymbol{c}_{zu}=\boldsymbol{c}_{zu}-\frac{1}{2}\beta_1\frac{\partial\delta^2}{\partial\boldsymbol{c}_{zu}}=\boldsymbol{c}_{zu}+\beta_1\delta\frac{\partial(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}))}{\partial\boldsymbol{c}_{zu}}=\boldsymbol{c}_{zu}+\beta_1\delta\boldsymbol{w}_{iz}\frac{\partial\phi_z(\boldsymbol{x})}{\partial\boldsymbol{c}_{zu}}=\boldsymbol{c}_{zu}+\beta_1\delta\boldsymbol{w}_{iz}\frac{(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})}{\boldsymbol{\sigma}_{zu}^2}(\boldsymbol{x}_u-\boldsymbol{c}_{zu}),$$

同理，

$$\boldsymbol{\sigma}_{zu}=\boldsymbol{\sigma}_{zu}-\frac{1}{2}\beta_2\frac{\partial\delta^2}{\partial\boldsymbol{\sigma}_{zu}}=\boldsymbol{c}_{zu}+\beta_2\delta\boldsymbol{w}_{iz}\frac{\partial\phi_z(\boldsymbol{x})}{\partial\boldsymbol{\sigma}_{zu}}=\boldsymbol{\sigma}_{zu}+\beta_2\delta\boldsymbol{w}_{iz}(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})\frac{(\boldsymbol{x}_u-\boldsymbol{c}_{zu})^2}{\boldsymbol{\sigma}_{zu}^3},$$

其中 $\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$ 为当前选择的第 i 个动作的 Q 值函数，$\phi_z(\boldsymbol{x})$ 形式如式(10)所示.

4 Q-V 值函数协同逼近算法

4.1 $QV(\lambda)$

根据 Q-V 值函数协同逼近模型，提出了一种基于 ANRBF 网络的协同逼近算法 $QV(\lambda)$，假设 $A=|\delta_{\mathrm{new}}|>\eta_1\wedge\forall j=1,2,\cdots,m,\max(\phi_j(\boldsymbol{x}))<\eta_2$，$B=\delta\boldsymbol{w}_{iz}(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})$，算法的详细描述如下.

算法 1. Q-V 值函数协同逼近算法 $QV(\lambda)$.

输入：任务环境模型

输出：各状态对应的策略

1. 初始化 ANRBF 网络隐藏层各节点激活函数；
2. 构建如式(5)所示的 Q-V 值函数协同逼近模型；
3. $\forall i=1,2,\cdots,l,\boldsymbol{w}_i=\boldsymbol{0}\in\mathbb{R}^m,\boldsymbol{e}_i=\boldsymbol{0}\in\mathbb{R}^m$；
4. repeat(对每一个情节)
5. 当前状态 $\boldsymbol{x}$ 及动作 $\boldsymbol{u}$；
6. repeat(对该情节中的每一步)
7. 执行动作 $\boldsymbol{u}$，观察 $\boldsymbol{r},\boldsymbol{x}'$；
8. 将数据$\langle\boldsymbol{x}',U(\boldsymbol{x}')\rangle$输入当前值函数模型；
9. $\boldsymbol{u}'\leftarrow$通过 ε-greedy 等策略选择 $\boldsymbol{x}'$下的动作；
10. 收集数据$\langle\boldsymbol{x},\boldsymbol{u},\boldsymbol{r},\boldsymbol{x}',\boldsymbol{u}'\rangle$；
11. 计算 TDE，$\delta_{\mathrm{old}}=\boldsymbol{r}+\gamma Q_{ID(\boldsymbol{u}')}(\boldsymbol{x}')-Q_{ID(\boldsymbol{u})}(\boldsymbol{x})$；
12. 计算 SR-SVF，$F(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')=\gamma V(\boldsymbol{x}')-V(\boldsymbol{x})$；
13. 计算 C-TDE，$\delta_{\mathrm{new}}=\delta_{\mathrm{old}}+F(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$；
14. 更新资格迹，$\forall i=1,2,\cdots,l$，
15. 若 $i=ID(\boldsymbol{u})\vee i=l$，则 $\boldsymbol{e}_i\leftarrow\max(\gamma\lambda\boldsymbol{e}_i,\boldsymbol{\phi}(\boldsymbol{x}))$；
16. 否则，$\boldsymbol{e}_i\leftarrow\gamma\lambda\boldsymbol{e}_i$；
17. 将 δ_{new}反馈给 Q-V 值函数协同逼近模型；
18. $\forall i=1,2,\cdots,l,\boldsymbol{w}_i=\boldsymbol{w}_i+\alpha\delta_{\mathrm{new}}\boldsymbol{e}_i$；
19. 调整协同逼近模型的隐藏层结构，
20. 若 A 成立，
21. 新增一个隐藏节点，$m\leftarrow m+1$；
22. 初始化，$\forall u=1,\cdots,n,\boldsymbol{c}_{mu}=\boldsymbol{x}_u,\boldsymbol{\sigma}_{mu}=0.1$，$\forall i=1,\cdots,l,\boldsymbol{w}_{im}=0,\boldsymbol{e}_{im}=0$；
23. 否则，
24. $\forall j=1,2,\cdots,m,\boldsymbol{c}_z=\arg\max_{\boldsymbol{c}}(\phi_j(\boldsymbol{x}))$，$\boldsymbol{\sigma}_z=\arg\max_{\boldsymbol{\sigma}}(\phi_j(\boldsymbol{x}))$；
25. 如果 $\delta\boldsymbol{w}_{iz}>0,\forall\boldsymbol{u}=1,\cdots,n$，$\boldsymbol{c}_{zu}\leftarrow\boldsymbol{c}_{zu}+\beta_1B(\boldsymbol{x}_u-\boldsymbol{c}_{zu})\boldsymbol{\sigma}_{zu}^{-2}$；
26. 否则，$\forall\boldsymbol{u}=1,\cdots,n$，$\boldsymbol{\sigma}_{zu}\leftarrow\max(\boldsymbol{\sigma}_{zu}+\beta_2B(\boldsymbol{x}_u-\boldsymbol{c}_{zu})^2\boldsymbol{\sigma}_{zu}^{-3},0.001)$；
27. $\boldsymbol{x}=\boldsymbol{x}',\boldsymbol{u}=\boldsymbol{u}'$；
28. 直到 $\boldsymbol{x}$ 是终止状态；
29. 直到运行完设定情节数或满足其他终止条件.

算法对 Q-V 值函数协同逼近模型的调整主要

包含两个方面：(1) 对输出层权重的调整；(2) 对隐藏层结构的调整. 其中，调整策略是文献[26]中所给出的 RGD 策略.

假设 1.　C-TDE 满足 $|\delta_{\text{new}}|\leq\eta_1$ 或者 $\forall j=1,2,\cdots,m$, $\max(\phi_j(\boldsymbol{x}))\geq\eta_2$，即 C-TDE 不大于阈值 η_1，或者逼近模型隐藏层输出的最大分量不小于阈值 η_2，其中 η_1 和 η_2 为事先设定的阈值.

假设 2.　令 $\boldsymbol{c}_z=\arg\max_{\boldsymbol{c}}(\phi_j(\boldsymbol{x}))$，$\boldsymbol{\sigma}_z=\arg\max_{\boldsymbol{\sigma}}(\phi_j(\boldsymbol{x}))$，即 $\boldsymbol{c}_z$ 和 $\boldsymbol{\sigma}_z$ 分别是隐藏层各激活函数中离 $\boldsymbol{x}$ 最近的一个基函数的中心向量和宽度向量.

定理 1.　在假设 1 和假设 2 成立的前提下，RGD 调整策略有效，即逼近模型关于状态向量 $\boldsymbol{x}$ 收缩.

证明.　下面从中心向量和宽度向量的更新过程来分别证明定理 1 的正确性：

① 由于 $\boldsymbol{\phi}(\boldsymbol{x})$ 经过归一化处理，$\forall z=1,2,\cdots,n$, $\phi_z(\boldsymbol{x})\in[0,1]$，有 $(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})\boldsymbol{\sigma}_{zu}^{-2}\geq 0$ 成立；

② 根据 RGD 调整策略，在状态 $\boldsymbol{x}$ 下选择的动作为 $\boldsymbol{u}$（在动作集合中的编号记为 $i=ID(\boldsymbol{u})$），只有当 $\delta\boldsymbol{w}_{iz}>0$ 时，才调整 $\boldsymbol{c}_{zu}$ 的大小. 又因为 $\Delta\boldsymbol{c}_{zu}=\beta_1\delta\boldsymbol{w}_{iz}(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})(\boldsymbol{x}_u-\boldsymbol{c}_{zu})\boldsymbol{\sigma}_{zu}^{-2}$，因此 $\Delta\boldsymbol{c}_{zu}$ 的正负取决于 $(\boldsymbol{x}_u-\boldsymbol{c}_{zu})$ 的正负；

③ 根据①和②可知：$\boldsymbol{c}_z$ 的 $\boldsymbol{u}$ 分量 $\boldsymbol{c}_{zu}$ 向状态向量 $\boldsymbol{x}$ 的 $\boldsymbol{u}$ 分量 $\boldsymbol{x}_u$ 方向调整；

④ 在③的基础上，考虑中心向量的每一维，可得中心向量 $\boldsymbol{c}_z$ 始终向状态向量 $\boldsymbol{x}$ 的方向调整；

⑤ 因为 $\phi_z(\boldsymbol{x})\in[0,1]$ 且 $\boldsymbol{\sigma}_{zu}>0$，则 $(1-\phi_z(\boldsymbol{x}))\phi_z(\boldsymbol{x})(\boldsymbol{x}_u-\boldsymbol{c}_{zu})^2\boldsymbol{\sigma}_{zu}^{-3}\geq 0$，因此，当 $\delta\boldsymbol{w}_{iz}<0$ 时，第 z 个 RBF 的宽度向量的 $\boldsymbol{u}$ 分量 $\boldsymbol{\sigma}_{zu}$ 将被缩小.

⑥ 由③可知，激活函数的中心向量调整的方向与状态向量 $\boldsymbol{x}$ 保持一致；由⑤可知激活函数的宽度向量调整是收缩的. 综上所述，RGD 调整策略有效，且逼近模型关于状态向量 $\boldsymbol{x}$ 收缩.　　证毕.

4.2　算法收敛性分析

下面从 Q-V 值函数逼近模型的结构和权值调整两个方面来分析 Q-V 逼近模型的收敛性. 参考文献[23,29]给出了引理 1，该引理主要用来辅助选择 Q-V 逼近模型隐藏层激活函数的中心向量.

引理 1[23,29].　假设 Q-V 逼近模型隐藏层激活函数的候选中心向量集为 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$，基于 ALD 方法得到 $t-1$ 时刻的中心向量集为 $\boldsymbol{X}_{t-1}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，$1<m\leq n$. 考虑一个新样本 $\boldsymbol{x}_t$，采用文献[23]中所给出的 ALD 特征选择方法，当 $\min_{\boldsymbol{c}}\left\|\sum_j\boldsymbol{c}_j\boldsymbol{\phi}(\boldsymbol{x}_j)-\boldsymbol{\phi}(\boldsymbol{x}_t)\right\|^2\leq\boldsymbol{\mu}$ 成立时丢弃样本，否则 $\boldsymbol{X}_t=\boldsymbol{X}_{t-1}\bigcup\{\boldsymbol{x}_t\}$，其中 $\boldsymbol{c}=[\boldsymbol{c}_1,\cdots,\boldsymbol{c}_j]$，$\boldsymbol{\mu}$ 为预先设定的阈值，则 Q-V 逼近模型隐藏层的中心向量能够稳定.

文献[23,29]给出了引理 1 的证明. 根据引理 1 所给出的策略，Q-V 逼近模型选择隐藏层激活函数的中心向量，则可以保证模型的隐藏层结点数及中心向量能够稳定.

基于引理 1 及定理 1，本文给出了定理 2，其证明在 Q-V 逼近模型的隐藏层结构趋于稳定的情况下，$QV(\lambda)$ 算法能够收敛.

定理 2.　假设 Q-V 逼近模型的隐藏层趋于稳定，在确定性 MDP 中，若算法具有相同的样本序列，则根据 C-TDE 更新值函数的 $QV(\lambda)$ 算法与利用 TDE 结合 V 值函数进行更新的 GD-Sarsa(λ) 算法具有相同的收敛性.

证明.　考虑具有连续状态空间 X 和离散动作空间 U 的强化学习问题，分别用 L 和 L' 表示 GD-Sarsa(λ) 算法和 $QV(\lambda)$ 算法的学习器. $\forall\boldsymbol{x}\in X$, $\boldsymbol{u}\in U$, $Q(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{w}_{ID(\boldsymbol{u})}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$ 为 L 中值函数，$Q(\boldsymbol{x},\boldsymbol{u})=Q_0(\boldsymbol{x},\boldsymbol{u})+V(\boldsymbol{x})=\boldsymbol{\omega}_0^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})+\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$ 是初始值函数；$Q'(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{\omega}_{ID(\boldsymbol{u})}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$ 为 L' 的值函数，其初始值为 $Q'(\boldsymbol{x},\boldsymbol{u})=Q_0(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{\omega}_0^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$. 为了书写方便，用 $Q'(\boldsymbol{x},\boldsymbol{u})$ 表示 $Q'_{ID(\boldsymbol{u})}(\boldsymbol{x})$. 假设在状态 $\boldsymbol{x}$ 下采取动作 $\boldsymbol{u}$ 转移到状态 $\boldsymbol{x}'$，立即奖赏为 $\boldsymbol{r}$，在 $\boldsymbol{x}'$ 下根据行为策略选择动作 $\boldsymbol{u}'$，则该样本数据可以用五元组 $\langle\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}',\boldsymbol{r},\boldsymbol{u}'\rangle$ 表示. 基于此五元组样本数据，L 和 L' 分别更新权值，更新公式如式(11)和式(12)：

$$\boldsymbol{w}_{ID(\boldsymbol{u})}\leftarrow\boldsymbol{w}_{ID(\boldsymbol{u})}+\alpha\delta Q(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}_{ID(\boldsymbol{u})}\tag{11}$$

$$\boldsymbol{\omega}_{ID(\boldsymbol{u})}\leftarrow\boldsymbol{\omega}_{ID(\boldsymbol{u})}+\alpha\delta Q'(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}'_{ID(\boldsymbol{u})}\tag{12}$$

其中 $\delta Q(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{r}+\gamma Q(\boldsymbol{x}',\boldsymbol{u}')-Q(\boldsymbol{x},\boldsymbol{u})$ 为 TDE，$\delta Q'(\boldsymbol{x},\boldsymbol{u})=\boldsymbol{r}+\gamma Q'(\boldsymbol{x}',\boldsymbol{u}')-Q'(\boldsymbol{x},\boldsymbol{u})+F(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ 为 C-TDE. $\boldsymbol{e}_{ID(\boldsymbol{u})}$ 和 $\boldsymbol{e}'_{ID(\boldsymbol{u})}$ 分别是值函数模型中的第 $ID(\boldsymbol{u})$ 个子模型的资格迹. 状态动作对 $\langle\boldsymbol{x},\boldsymbol{u}\rangle$ 先后两次值函数的差分别记为 $\Delta Q(\boldsymbol{x},\boldsymbol{u})$，$\Delta Q'(\boldsymbol{x},\boldsymbol{u})$. 根据式(11)和式(12)的权值更新调整 Q 值函数的更新，Q 值函数的更新量计算如式(13)和式(14)：

$$\begin{aligned}\Delta Q(\boldsymbol{x},\boldsymbol{u})&=Q(\boldsymbol{x},\boldsymbol{u})-Q_0(\boldsymbol{x},\boldsymbol{u})-V(\boldsymbol{x})\\&=(\alpha\delta Q(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}_{ID(\boldsymbol{u})})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})\end{aligned}\tag{13}$$

$$\begin{aligned}\Delta Q'(\boldsymbol{x},\boldsymbol{u})&=Q'(\boldsymbol{x},\boldsymbol{u})-Q_0(\boldsymbol{x},\boldsymbol{u})\\&=(\alpha\delta Q'(\boldsymbol{x},\boldsymbol{u})\boldsymbol{e}'_{ID(\boldsymbol{u})})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})\end{aligned}\tag{14}$$

给定相同的学习样本序列，接下来利用归纳法证明 L 和 L' 的等价性，即证明 $\forall\boldsymbol{x}\in X,\boldsymbol{u}\in U$, $\Delta Q(\boldsymbol{x},\boldsymbol{u})=\Delta Q'(\boldsymbol{x},\boldsymbol{u})$. 证明过程如下：

① 当 $Q(\boldsymbol{x},\boldsymbol{u})$ 和 $Q'(\boldsymbol{x},\boldsymbol{u})$ 都是初始值时，则

$\Delta Q(\boldsymbol{x},\boldsymbol{u})=\Delta Q'(\boldsymbol{x},\boldsymbol{u})=0$;

② 假设当前存在 $\forall \boldsymbol{x}\in X,\boldsymbol{u}\in U$, $\Delta Q(\boldsymbol{x},\boldsymbol{u})=\Delta Q'(\boldsymbol{x},\boldsymbol{u})$,则在此基础上,根据新的样本数据$\langle \boldsymbol{x},\boldsymbol{u},\boldsymbol{x}',r,\boldsymbol{u}'\rangle$,利用 TDE 及 C-TDE 可得

$$\begin{aligned}\delta Q(\boldsymbol{x},\boldsymbol{u})&=r+\gamma Q(\boldsymbol{x}',\boldsymbol{u}')-Q(\boldsymbol{x},\boldsymbol{u})\\&=r+\gamma(Q_0(\boldsymbol{x}',\boldsymbol{u}')+V(\boldsymbol{x}')+\Delta Q(\boldsymbol{x}',\boldsymbol{u}'))-\\&\quad Q_0(\boldsymbol{x},\boldsymbol{u})-V(\boldsymbol{x})-\Delta Q(\boldsymbol{x},\boldsymbol{u})\\&=r+\gamma(Q_0(\boldsymbol{x}',\boldsymbol{u}')+\Delta Q(\boldsymbol{x}',\boldsymbol{u}'))-\\&\quad Q_0(\boldsymbol{x},\boldsymbol{u})-\Delta Q(\boldsymbol{x},\boldsymbol{u})+\gamma V(\boldsymbol{x}')-V(\boldsymbol{x}),\\\delta Q'(\boldsymbol{x},\boldsymbol{u})&=r+\gamma Q'(\boldsymbol{x}',\boldsymbol{u}')-Q'(\boldsymbol{x},\boldsymbol{u})+F(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\\&=r+\gamma(Q_0(\boldsymbol{x}',\boldsymbol{u}')+\Delta Q'(\boldsymbol{x}',\boldsymbol{u}'))-\\&\quad Q_0(\boldsymbol{x},\boldsymbol{u})-\Delta Q'(\boldsymbol{x},\boldsymbol{u})+\gamma V(\boldsymbol{x}')-V(\boldsymbol{x})\\&=r+\gamma(Q_0(\boldsymbol{x}',\boldsymbol{u}')+\Delta Q(\boldsymbol{x}',\boldsymbol{u}'))-\\&\quad Q_0(\boldsymbol{x},\boldsymbol{u})-\Delta Q(\boldsymbol{x},\boldsymbol{u})+\gamma V(\boldsymbol{x}')-V(\boldsymbol{x}).\end{aligned}$$

根据上述计算,可得 $\delta Q'(\boldsymbol{x},\boldsymbol{u})=\delta Q(\boldsymbol{x},\boldsymbol{u})$,即在新样本数据下,$L$ 和 L'所计算的 TDE 和 C-TDE 相同.此外,由于具有相同的样本序列,L 和 L'中资格迹更新也是相同的,又根据每个时间步 L 和 L'获得相同的 TDE 和 C-TDE,因此,L 和 L'资格迹在每个时间步具有相同的更新,即对于任一时间步,$\boldsymbol{e}_{ID(\boldsymbol{u})}=\boldsymbol{e}'_{ID(\boldsymbol{u})}$.再根据式(13)和式(14)可得,$\forall \boldsymbol{x}\in X,\boldsymbol{u}\in U$,$\Delta Q(\boldsymbol{x},\boldsymbol{u})=\Delta Q'(\boldsymbol{x},\boldsymbol{u})$.综上所述,在相同的样本序列下,$L$ 和 L'等价,即 $QV(\lambda)$算法与 GD-Sarsa(λ)算法具有一致的收敛性. 证毕.

定理 3. 在式(15)所给的条件下,如果学习率 α 随时间衰减,则利用梯度下降方法更新参数的 $QV(\lambda)$算法能够收敛至一个局部最优解.

$$\sum_{t=0}^{\infty}\alpha_t=\infty,\ \sum_{t=0}^{\infty}\alpha_t^2<\infty \tag{15}$$

证明. 根据文献[10]得知,采用梯度下降方法的线性学习算法,例如 Sutton 等人提出的 GD-Sarsa(λ)算法在满足式(15)的条件下,如果学习率 α 随时间衰减,则能够保证该算法能收敛至一个局部最优解.又根据定理 2,在具有相同样本序列的情况下,$QV(\lambda)$算法与 GD-Sarsa(λ)算法具有一致的收敛性.而在强化学习算法中,任意初始化值函数都不会影响算法的收敛结果,因此,在学习率 α 满足式(15)且随时间衰减的情况下,$QV(\lambda)$算法一定能够收敛至一个局部最优解. 证毕.

5 实验及结果分析

5.1 实验描述及设置

Mountain Car 问题是一个经典的具有连续状态空间、离散动作空间的强化学习问题,如图 2 所示.

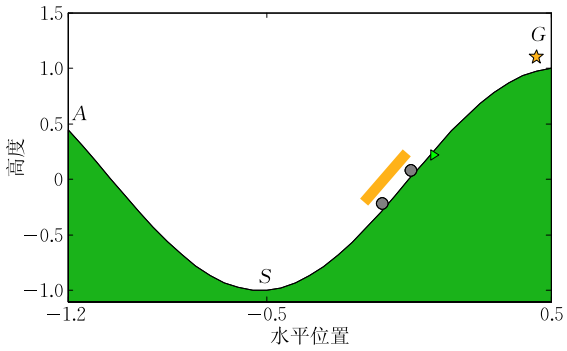


图 2 Mountain Car 问题示意图

图 2 描述的是一个动力不足的小车,在山谷中的任何一点通过前后摆动的方式最快到达山顶的 G 点的问题.其中状态定义如式(16):

$$\boldsymbol{X}=\{[y,v]^{\mathrm{T}}\in\mathbb{R}^2\} \tag{16}$$

其中 y 为小车的水平位移($-1.2\leq y\leq 0.5$),v 为小车的水平速度($-0.07\leq v\leq 0.07$).系统的动力学特性如式(17):

$$\begin{cases}\dot{v}=\mathrm{bound}[v+0.001\boldsymbol{u}-g\cos(3y)]\\\dot{y}=\mathrm{bound}[y+\dot{v}]\end{cases} \tag{17}$$

其中 $g=0.0025$ 为与重力有关的常数,$\boldsymbol{u}$ 为控制动作,取 3 个离散值,+1,0 和−1,分别代表全油门向前、零油门和全油门向后.另外,小车运动过程中的奖赏函数如式(18):

$$\boldsymbol{r}_t=\begin{cases}-1, & y<0.5\\0, & y\geq 0.5\end{cases} \tag{18}$$

其中下标 t 表示时间步.

实验采用 ε-greedy 作为行为策略[1],每次实验的情节数为 1000,每个情节的最大时间步数为 1000.每次实验随机生成小车的初始状态,当小车到达 G 点或者时间步数超过 1000 时,情节结束,并开始下一个情节的学习.

算法的性能指标包括 3 个方面:(1)收敛速度,即算法能够在多少个情节内收敛;(2)收敛结果,即算法收敛后,小车从初始点到达目标点 G 所用的平均时间步;(3)初始性能,即每次实验的前 20 个情节中,小车从初始点到达目标点 G 所用的平均时间步.将本文所提的 $QV(\lambda)$算法与 GD-Sarsa(λ)算法[1],Greedy-GQ 算法[17]以及 RGD-Sarsa(λ)算法[26]在 Mountain Car 问题中进行对比分析.并着重分析资格迹、学习率、SR-SVF 及自适应机制对算法性能的影响.

5.2　实验分析

在 $QV(\lambda)$算法及 RGD-Sarsa(λ)算法中，分别采用 8 个等距的高斯 RBFs 来对二维连续状态空间的每一维进行划分，RBFs 的数量为 $8\times8=64$，宽度向量为 $\boldsymbol{\sigma}=[0.1,0.1]^{\mathrm{T}}$. 另外 $\varepsilon=0.0$，$\alpha=0.9$，$\lambda=0.9$，$\gamma=1.0$. 在 GD-Sarsa(λ)中，采用 10 个 9×9 的 Tilings 来划分状态空间，并利用文献[1]中给出的最优参数：另外，$\varepsilon=0.0$，$\alpha=0.14$，$\gamma=1.0$，在采用替代迹的情况下，$\lambda=0.9$，在采用累加迹的情况下，$\lambda=0.3$. 在 Greedy-GQ 算法中，$\varepsilon=0.0$，$\alpha=0.1$，$\gamma=1.0$. 从图 3 和图 4 中可以看出，$QV(\lambda)$算法在收敛速度和初始性能上明显优于其他 3 个算法.

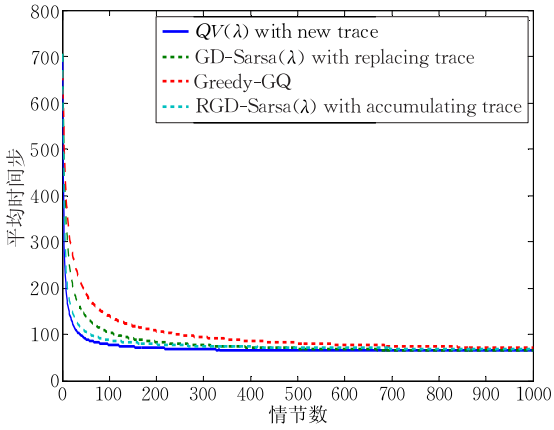
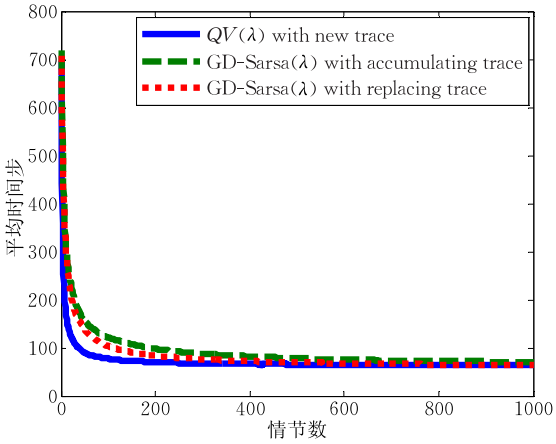


图 3　算法性能比较



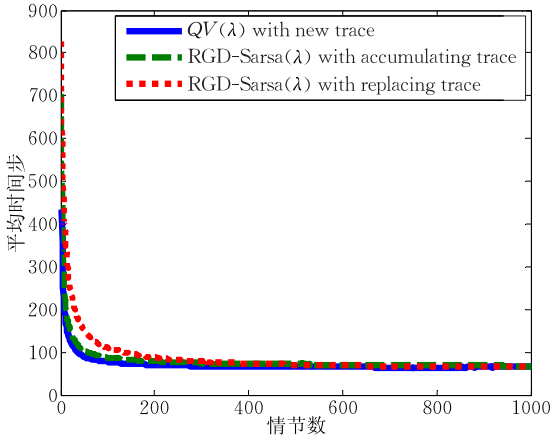
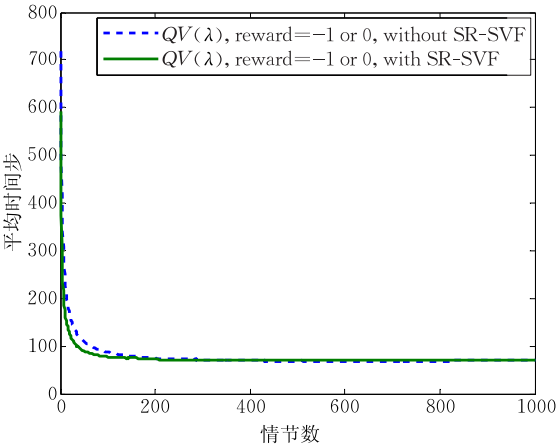


图 4　算法性能比较(续)

采用上述给出的 RBFs 配置，$\varepsilon=0.0$，$\alpha=0.9$，$\lambda=0.9$，$\gamma=0.99$. 分别考虑两个不同的立即奖赏模型：(1) 根据式(18)给出的奖赏模型；(2) 奖赏模型为：当 $y<0.5$ 时，$\boldsymbol{r}_t=0$；当 $y\geq0.5$ 时，$\boldsymbol{r}_t=1$. 实验结果如图 5 所示，左图为第 1 种奖赏设置下的算法执行结果，右图为第 2 种奖赏设置下的算法执行结果. 从图 5 可以看出，带有 SR-SVF 机制的 $QV(\lambda)$算法在收敛速度和初始性能方面明显优于不带 SR-SVF 机制的 $QV(\lambda)$算法. 在采用第 1 种奖赏模型的情况下，由于奖赏是 -1 和 0，是一种惩罚型奖赏，SR-SVF 机制对算法收敛速度和初始性能影响较小；当采用第 2 种奖赏模型时，由于奖赏是 0 和 1，是一种鼓励型奖赏，只有到达目标状态时，才能有益于算法的学习，因此 SR-SVF 机制可以在学习初期通过环境模型知识构造塑造奖赏，并传递给学习器，从而能够有效地提高算法的收敛速度和初始性能.



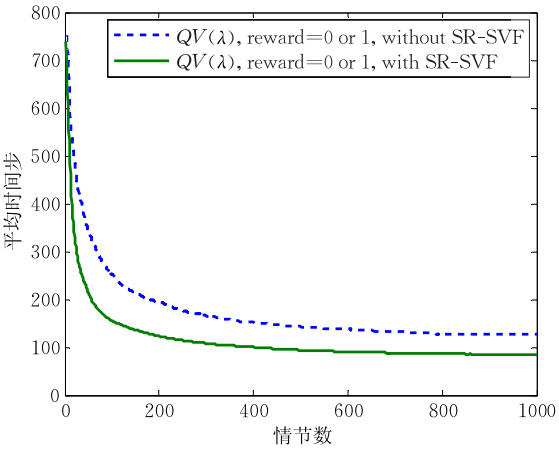


图 5　SR-SVF 机制对 $QV(\lambda)$算法的性能影响分析

下面重点分析学习率 α 对算法性能的影响. 在 $QV(\lambda)$算法中,本文定义了新的资格迹,$\lambda=0.9$,α 分别取值 0.1,0.5,0.9,1.0;在 GD-Sarsa(λ)算法中,采用替代迹,$\lambda=0.9$,α 分别取值 0.01,0.05,0.14,0.18,其余参数设置不变;在 Greedy-GQ 算法中,α 的取值分别为 0.005,0.015,0.05,0.1,其余参数不变;在 RGD-Sarsa(λ)算法中,采用替代迹,$\lambda=0.9$,α 的取值分别为 0.1,0.5,0.9,1.0,其余参数不变. 实验结果如图 6 所示,与基于线性函数近似的 GD-Sarsa(λ)、Greedy-GQ 算法以及基于非参函数近似的 RGD-Sarsa(λ)算法相比,本文所给出的 $QV(\lambda)$ 算法对于不同学习率具有较强的鲁棒性.

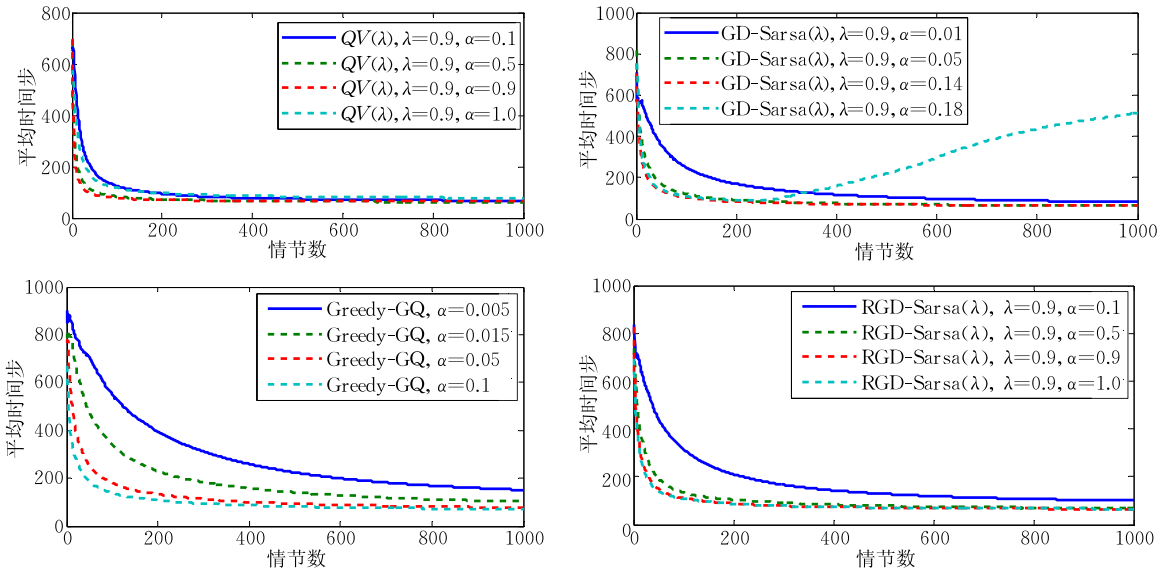


图 6 算法关于学习率的敏感度分析

表 1 给出了 $QV(\lambda)$算法的平均性能. 当 RBFs 设置为 5×5=25 时,带有自适应机制的 $QV(\lambda)$算法大约需要 229 个情节收敛,且收敛后大约需要 73 个时间步到达目标点,而不带自适应机制的 $QV(\lambda)$算法收敛后大约需要 275 个情节收敛,且收敛后大约需要 142 个时间步到达目标点;当 RBFs 设置为 3×3=9 时,带有自适应机制的 $QV(\lambda)$算法大约需要 252 个情节收敛,且收敛后大约需要 75 个时间步到达目标点,而不带自适应机制的 $QV(\lambda)$算法大约需要 269 个情节收敛,且收敛后大约需要 276 个时间步到达目标点.

表 1 自适应特性对 $QV(\lambda)$算法的性能影响分析

	Init RBFs=5×5		Init RBFs=3×3	
	$QV(\lambda)$ with adaptive	$QV(\lambda)$ without adaptive	$QV(\lambda)$ with adaptive	$QV(\lambda)$ without adaptive
收敛情节数	**229**±15	**275**±15	**252**±15	**269**±15
平均时间步	**73**±15	**142**±15	**75**±15	**276**±15

6 结束语

本文利用 ANRBF 网络将状态空间映射到高维特征空间,通过一组高斯基函数的线性组合构建 Q-V 值函数协同逼近模型. 该模型既具有非线性逼近模型的强表达能力,又具有线性逼近模型的简单性及良好的收敛性. ANRBF 网络可以有效地提高算法的鲁棒性和灵活性,从而在一定程度上解决了强化学习逼近模型所面临的"灾难性扰动"问题. 通过 Q 值函数和 V 值函数构造协同 TD 误差,从而有效地提高算法的收敛速度和初始性能. 此外,本文定义了一种新的资格迹更新方法,并用来处理基于连续编码的 Q-V 值函数协同逼近模型的时间信度分配问题,实验结果表明具有较优的效果.

考虑到强化学习的自学习和在线学习特性,如何设计有效的逼近模型来更好的适应和处理在线强化学习问题,有待进一步研究.

参 考 文 献

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

[2] Sutton R S, Modayil J, Delp M, et al. Horde: A scalable

real-time architecture for learning knowledge from unsupervised sensorimotor interaction//Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems. Taipei, China, 2011: 761-768

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

[4] Liu Quan, Fu Qi-Ming, Gong Sheng-Rong, Fu Yu-Chen, Cui Zhi-Ming. A reinforcement learning algorithm based on minimum state method and average reward. Journal on Communications, 2011, 32(1): 66-71(in Chinese)
(刘全，傅启明，龚声蓉，伏玉琛，崔志明. 最小状态变元平均奖赏的强化学习方法. 通信学报，2011，32(1)：66-71)

[5] Chen Zong-Hai, Wen Feng, Nie Jian-Bin, Wu Xiao-Shu. A reinforcement learning method based on node-growing k-means clustering algorithm. Journal of Computer Research and Development, 2006, 34(4): 661-666(in Chinese)
(陈宗海，文锋，聂建斌，吴晓曙. 基于节点生长 k-均值聚类算法的强化学习方法. 计算机研究与发展，2006，34(4)：661-666)

[6] Liu Quan, Yan Qi-Cui, Fu Yu-Chen, Hu Dao-Jing, Gong Sheng-Rong. A hierarchical reinforcement learning method based on heuristic reward function. Journal of Computer Research and Development, 2011, 48(12): 2352-2358(in Chinese)
(刘全，闫其粹，伏玉琛，胡道京，龚声蓉. 一种基于启发式奖赏函数的分层强化学习方法. 计算机研究与发展，2011，48(12)：2352-2358)

[7] Kretchmar R M. Parallel reinforcement learning//Proceedings of the 6th World Conference on Systemics, Cybernetics, and Informatics. New York, USA, 2002: 114-118

[8] Meng Wei, Han Xue-Dong. Parallel reinforcement learning algorithm and its application. Chinese Computer Engineering and Applications, 2009, 45(34): 25-28(in Chinese)
(孟伟，韩学东. 并行强化学习算法及其应用研究. 计算机工程与应用，2009，45(34)：25-28)

[9] Geist M, Pietquin O. Parametric value function approximation: A unified view//Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning. Paris, France, 2011: 9-16

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

[11] Busoniu L, Babuska R, De Schutter B, Erns D. Reinforcement Learning and Dynamic Programming Using Function Approximators. New York: CRC Press, 2010

[12] Tadic V. On the convergence of temporal-difference learning with linear function approximation. Machine Learning, 2001, 42(3): 241-267

[13] Sherstov A A, Stone P. Function approximation via tile coding: Automating parameter choice//Proceedings of the 5th Symposium on Abstraction, Reformulation and Approximation. Airth Castle, UK, 2005: 194-205

[14] Sutton R S, Szepesvári Cs, Maei H R. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation//Proceedings of the 22nd Annual Conference on Neural Information Processing Systems. Vancouver, Canada, 2009: 1609-1616

[15] Sutton R S, Hamid R M, Precup D, et al. Fast gradient-descent methods for temporal-difference learning with linear function approximation//Proceedings of the 26th International Conference on Machine Learning. Montreal, Canada, 2009: 993-1000

[16] Maei H R, Sutton R S. GQ(λ): A general gradient algorithm for temporal difference prediction learning with eligibility traces//Proceedings of the 3rd Conference on Artificial General Intelligence. Atlanta, USA, 2010: 91-96

[17] Maei H R, Szepesvári Cs, Bhatnagar S, Sutton R S. Toward off-policy learning control with function approximation//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 719-726

[18] Maei H R, Szepesvári Cs, Bhatnagar S, et al. Convergent temporal-difference learning with arbitrary smooth function approximation//Proceedings of the 23rd Annual Conference on Neural Information Processing Systems. Vancouver, Canada, 2009: 1204-1212

[19] Bonarini A, Lazaric A, Montrone F, Restelli M. Reinforcement distribution in fuzzy Q-learning. Fuzzy Sets and Systems, 2009, 160(10): 1420-1443

[20] Heinen M R, Engel P M. An incremental probabilistic neural network for regression and reinforcement learning tasks//Proceedings of the 20th International Conference on Artificial Neural Networks. Thessaloniki, Greece, 2010: 170-179

[21] Engel Y, Mannor S, Meir R. Bayes meets Bellman: the Gaussian process approach to temporal difference learning//Proceedings of the 20th International Conference on Machine Learning. Washington, USA, 2003: 154-161

[22] Ormoneit D, Sen Ś. Kernel-based reinforcement learning. Machine Learning, 2002, 49(2-3): 161-178

[23] Xu X, Xie T, Hu D, Lu X. Kernel least-squares temporal difference learning. International Journal of Information Technology, 2005, 11(9): 54-63

[24] Xu X, Hu D, Lu X. Kernel-based least squares policy iteration for reinforcement learning. IEEE Transactions on Neural Networks, 2007, 18(4): 973-992

[25] Taylor G, Parr R. Kernelized value function approximation for reinforcement learning//Proceedings of the 26th International Conference on Machine Learning. Montreal, Canada, 2009: 1017-1024

[26] Barreto A D M S, Anderson C W. Restricted gradient-descent algorithm for value-function approximation in reinforcement learning. Artificial Intelligence, 2008, 172(4-5): 454-482

[27] Ng A Y, Harada D, Russell S. Policy invariance under reward transformations: Theory and application to reward

shaping//Proceedings of the 16th International Conference on Machine Learning. San Francisco, USA, 1999: 278-287

[28] Randløv J, Alstrøm P. Learning to drive a bicycle using reinforcement learning and shaping//Proceedings of the 15th International Conference on Machine Learning. San Francisco, USA, 1998: 463-471

[29] Engel Y, Mannor S, Meir R. The kernel recursive least-squares algorithm. IEEE Transactions on Signal Processing, 2004, 52(8): 2275-2285

LIU Quan, born in 1969, Ph. D., professor, Ph. D. supervisor. His main research interests include reinforcement learning, intelligence information processing and automated reasoning.

XIAO Fei, born in 1988, M. S. His main research interests include reinforcement learning and support vector machine.

FU Qi-Ming, born in 1985, Ph. D. candidate. His main research interests include reinforcement learning and Bayesian method.

FU Yu-Chen, born in 1968, Ph. D., associate professor. His main research interest is reinforcement learning.

ZHOU Xiao-Ke, born in 1976, Ph. D. candidate, lecturer. His main research interest is reinforcement learning.

ZHU Fei, born in 1978, Ph. D. candidate, lecturer. His main research interests include reinforcement learning and Bayesian method.

Background

This paper is supported by the National Natural Science Foundation of China (61272005, 61070223, 61103045, 61070122, 61472262), the Natural Science Foundation of Jiangsu (BK2012616), the High School Natural Foundation of Jiangsu (09KJA520002, 09KJB520012), the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172012K04).

Reinforcement learning is a significant technology to optimize the target behavior in many fields, such as process controlling, robot controlling, path planning, web information search, portfolio management, option pricing and so on. It is well known that it is difficult to deal with the reinforcement learning problems with continuous state space. As a kind of useful function approximation model, RBF network approximation models are used widely to solve the reinforcement learning problems with continuous state space. However, RBF network approximation models are facing the "catastrophic interference" problem.

In this paper, we introduced a new function approximation model named collaborative Q-V value function approximation model to solve "catastrophic interference" problem of RBF approximation models. A function approximation algorithm named $QV(\lambda)$ was proposed based on the collaborative Q-V value function approximation model. The convergence of $QV(\lambda)$ algorithm was analyzed theoretically. We verified the performance of $QV(\lambda)$ on Mountain Car benchmark problem and compared with GD-Sarsa (λ) Greedy-GQ and RGD-Sarsa(λ). The results show that $QV(\lambda)$ has better performance than the other three.