软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(11):2676−2686 [doi: 10.3724/SP.J.1001.2013.04466]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种高斯过程的带参近似策略迭代算法*

傅启明[1], 刘 全[1,2], 伏玉琛[1], 周谊成[1], 于 俊[1]

[1](苏州大学 计算机科学与技术学院,江苏 苏州 215006)

[2](符号计算与知识工程教育部重点实验室(吉林大学),吉林 长春 130012)

通讯作者: 刘全, E-mail: quanliu@suda.edu.cn

摘 要: 在大规模状态空间或者连续状态空间中,将函数近似与强化学习相结合是当前机器学习领域的一个研究热点;同时,在学习过程中如何平衡探索和利用的问题更是强化学习领域的一个研究难点.针对大规模状态空间或者连续状态空间、确定环境问题中的探索和利用的平衡问题,提出了一种基于高斯过程的近似策略迭代算法.该算法利用高斯过程对带参值函数进行建模,结合生成模型,根据贝叶斯推理,求解值函数的后验分布.在学习过程中,根据值函数的概率分布,求解动作的信息价值增益,结合值函数的期望值,选择相应的动作.在一定程度上,该算法可以解决探索和利用的平衡问题,加快算法收敛.将该算法用于经典的 Mountain Car 问题,实验结果表明,该算法收敛速度较快,收敛精度较好.

关键词: 强化学习;策略迭代;高斯过程;贝叶斯推理;函数近似

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 傅启明,刘全,伏玉琛,周谊成,于俊.一种高斯过程的带参近似策略迭代算法.软件学报,2013,24(11):2676−2686. http://www.jos.org.cn/1000-9825/4466.htm

英文引用格式: Fu QM, Liu Q, Fu YC, Zhou YC, Yu J. Parametric approximation policy iteration algorithm based on Gaussian process. Ruan Jian Xue Bao/Journal of Software, 2013,24(11):2676−2686 (in Chinese). http://www.jos.org.cn/1000-9825/4466.htm

Parametric Approximation Policy Iteration Algorithm Based on Gaussian Process

FU Qi-Ming[1], LIU Quan[1,2], FU Yu-Chen[1], ZHOU Yi-Cheng[1], YU Jun[1]

[1](School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

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

Corresponding author: LIU Quan, E-mail: quanliu@suda.edu.cn

Abstract: In machine learning with large or continuous state space, it is a hot topic to combine the function approximation and reinforcement learning. The study also faces a very difficult problem of how to balance the exploration and exploitation in reinforcement learning. In allusion to the exploration and exploitation dilemma in the large or continuous state space, this paper presents a novel policy iteration algorithm based on Gaussian process in deterministic environment. The algorithm uses Gaussian process to model the action-value function, and in conjunction with generative model, obtains the posteriori distribution of the parameter vector of the action-value function by Bayesian inference. During the learning process, it computes the value of perfect information according to the posteriori distribution, and then selects the appropriate action with respect to the expected value of the action-value function. The algorithm achieves the balance between exploration and exploitation to certain extent, and therefore accelerates the convergence. The experimental results on the Mountain Car problem show that the algorithm has faster convergence rate and better convergence performance.

Key words: reinforcement learning; policy iteration; Gaussian process; Bayesian inference; function approximation

* 基金项目: 国家自然科学基金(61070223, 61103045, 61170020, 61272005, 61272244); 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基金(93K172012K04)

收稿时间: 2013-01-29; 修改时间: 2013-07-16; 定稿时间: 2013-08-27

强化学习(reinforcement learning,简称 RL)是在未知的、动态的环境中在线求解最优策略,以获取最大化期望回报的一类算法.根据算法的执行流程,强化学习算法可以被分为两大类:策略迭代算法(policy iteration,简称 PI)和值迭代算法(value iteration,简称 VI)[1].这两类算法最早可以追溯到动态规划(dynamic programming,简称 DP)中的策略迭代算法和值迭代算法.然而,与动态规划算法相比,强化学习算法具有两个重要的特征:(1) 动态规划一般是在模型已知的情况进行规划学习,而强化学习却不存在这个限制,它可以用来求解模型未知的最优化问题;(2) 强化学习算法可以在线求解最优策略,也就是说,它可以在 Agent 与环境的交互过程中进行学习,而传统动态规划算法却不能在线求解策略[2].

维数灾问题一直是制约强化学习,甚至机器学习发展和应用的一个重要问题,主要是指解决问题的复杂度将随着状态空间维数的增长而呈几何级数增长的现象[3].因此,在大规模状态空间或者连续状态空间问题中,传统的基于查询表的强化学习算法难以收敛.从 20 世纪 90 年代末开始,将函数近似引入强化学习,利用近似函数表示值函数,极大地提升了强化学习算法解决问题的能力.近 10 年来,基于函数近似的强化学习算法是当前强化学习领域的一个研究热点.Tadic 将线性函数逼近器和 TD 学习相结合,用于有限维度状态空间问题,并证明了算法有效性,但实验结果表明,与传统的 TD 学习相比,该算法的执行效率不高[4].Shah 等人将决策树与强化学习相结合,利用一种基于决策树的函数逼近器对值函数进行估计,以提高值函数的泛化性能[5].Precup 等人提出了将函数逼近用于离策略的 TD(λ)算法,并证明了算法收敛性,但该算法对于初始行为策略有一定的要求,算法在执行过程中,可能因为初始行为策略不满足要求而出现无法收敛的情况[6].Lagoudakis 等人利用最小二乘法求解近似策略迭代问题,但该算法只能用于环境模型已知的情况[7].Engel 等人结合高斯过程,利用贝叶斯推理方法改进策略评估方法,降低环境随机性对算法的影响,提高算法的鲁棒性[8].Sutton 等人利用线性函数将 TD 学习与函数逼近相结合,提出了 GTD,GTD2 和 TDC 算法,并在实验和理论上均证明了这几种算法的收敛性[9,10].

强化学习的一个重要特征就是可以通过 Agent 与环境的交互在线求解最优策略,因此在 Agent 与环境的交互过程中,如何平衡探索(学习新的策略)和利用(利用已有的策略)问题就显得极为重要.该问题也是当前强化学习领域的一个研究难点[2].传统的强化学习算法通常利用ε-greedy 策略或者模拟退火方法选择动作,在选择最优策略的基础上加入一定的探索信息,但是这类方法却无法合理地平衡探索和利用,在一定程度地保证算法收敛至最优策略的基础上,却极大地降低了算法的收敛速度.Dearden 在 1998 年首次将贝叶斯推理与强化学习相结合,提出一种贝叶斯 Q 学习算法.该算法利用概率模型对值函数进行建模,利用贝叶斯推理获取值函数的后验,确定值函数的置信度,以解决探索和利用的平衡问题,但该方法过于依赖参数的初始值,且只能用于小规模状态空间问题.近年来,贝叶斯强化学习得到了许多研究者的关注[11].Ghavamzadeh 等人利用高斯过程对策略梯度进行建模,利用贝叶斯推理求解策略梯度的后验,提高预测的精度[12].Dimitrakakis 等人将多任务强化学习(multitask reinforcement learning)与反转强化学习(inverse reinforcement learning)相结合,利用贝叶斯推理求解参数后验,提出一种贝叶斯多任务反转强化学习算法[13].Xu 等人将核方法与贝叶斯强化学习方法相结合,提出一种基于核方法的最小二乘贝叶斯强化学习方法,降低了算法对模型的依赖性[14].Wingat 等人对策略参数建模,结合贝叶斯推理,提出一种结合策略先验的贝叶斯策略搜索方法[15].Ross 等人提出将贝叶斯强化学习用于部分感知的 MDP 问题中,并在学习过程中利用采样信息建模,结合规划方法加快算法的收敛速度[16].

本文主要针对大规模或者连续状态空间、确定环境问题中的平衡和探索问题,提出一种基于高斯过程的带参近似策略迭代算法,考虑利用高斯过程对带参的值函数进行建模,结合概率生成模型,通过贝叶斯推理,求解值函数参数的后验分布,提高预测的精确性.在学习过程中,通过求解动作的信息价值增益,结合动作值函数的期望值选择相应的动作,以解决学习过程中探索和利用的平衡问题.将基于高斯过程的带参近似策略迭代算法用于 Mountain Car 问题,实验结果表明,该算法与传统的近似策略迭代算法相比,算法的收敛精度和收敛速度都有较大的提高.

1 马尔可夫决策过程

马尔可夫决策过程(Markov decision process,简称 MDP)一般可用于对顺序决策过程问题建模.MDP 问题可

用一个四元组$\langle X,U,f,\rho\rangle$表示,其中,$X$是状态集合;$U$是动作集合;$f$是状态转移函数,$f:X\times U\times X\to[0,1]$;$\rho$是奖赏函数,$\rho:X\times U\times\mathbb{R}\to[0,1]$.为了简化问题,$Z$表示状态动作集合,即$Z:X\times U$.根据$f$和$\rho$的输出是否为定值,可将MDP问题分成确定MDP问题和随机MDP问题.其中,确定MDP问题是指状态的转移及相应的奖赏值是定值;随机MDP问题是指状态的转移及相应的奖赏值不是一个定值,而是满足某一个分布.为不失一般性,考虑随机MDP的情况,假设在时刻k,状态为x_k,选择动作u_k,策略为h,状态转移至x_{k+1},奖赏值为r_k,为了后续表示方便,记$r(x_k)=r_k$或者$r(z_k)=r_k$,其中,状态转移函数及奖赏值函数如公式(1)、公式(2)所示.

$$P(x_{k+1}=x'|x_k,u_k)=f(x_k,u_k,x') \tag{1}$$

$$P(r_{k+1}=r|x_k,u_k)=\rho(x_k,u_k,r) \tag{2}$$

其中,$P(x_{k+1}=x'|x_k,u_k)$表示在状态为x_k,选择动作u_k,转移到状态x'的概率;$P(r_{k+1}=r|x_k,u_k)$表示在状态为x_k,选择动作u_k,获取奖赏值r的概率.

根据不同时刻,选择动作的策略是否一致,策略可以分为稳定策略和不稳定策略.在强化学习中,通常选择稳定策略为研究对象,本文中出现的策略都属于稳定策略.策略是指在给定状态情况下,动作选择的概率分布,即$h:X\times U\to[0,1]$.对于一个给定的策略h,可以给出在当前策略下状态转移的概率$P^h(x'|x)$,如公式(3)所示:

$$P^h(x'\mid x)=\int_{u\in U}h(u\mid x)P(x'\mid x,u)\mathrm{d}u \tag{3}$$

其中,$P(x'|x,u)$是指在当前状态x下选择动作u并转移至x'的概率,这是由环境本身决定的.对于一个给定的策略h以及一组状态序列$\Omega=\{x_0,x_1,x_2,\ldots,x_t\}$,状态序列的转移概率如公式(4)所示:

$$P^h(\Omega)=\prod_{i=1}^{t}p^h(x_i\mid x_{i+1}) \tag{4}$$

强化学习的目标是在未知环境中求解一个最优策略,以获取最大化的期望回报.公式(5)给出累积折扣回报R的定义:

$$R(x)=\left\{\sum_{i=0}^{\infty}\gamma^i r(x_i)\mid x_0=x\right\} \tag{5}$$

其中,$x_{i+1}\sim P(\cdot\mid x_i,\cdot)$,$\gamma$是折扣因子.在随机MDP中,$R(x_0)$的随机性主要来源于环境中状态转移的随机性以及立即奖赏的相关的噪声,且这两部分都是由MDP本身所决定的.通过对公式(5)变形,可得公式(6):

$$R(x)=r(x)+\gamma R(x') \tag{6}$$

在强化学习中,状态值函数$V(x)$(或者动作值函数$Q(x,u)$)是指当前状态x(或者状态-动作对(x,u))下回报值的期望值,用$E_h\{\cdot\}$表示在当前策略h下取期望操作.因此在强化学习中,通过值函数对当前策略h进行评估,其中,状态值函数$V^h(x)$的定义如公式(7)所示:

$$V^h(x)=E_h\{R(x)\} \tag{7}$$

将公式(6)带入公式(7),得到公式(8):

$$V^h(x)=\bar{r}(x)+\gamma E_{x'}\{V^h(x')\} \tag{8}$$

其中,$\bar{r}(x)=\int_{\mathbb{R}}p(r\mid x)r\mathrm{d}r$,$E_{x'}\{V^h(x')\}=\int_X p^h(x'\mid x)V^h(x')\mathrm{d}x'$.同理,给出$Q$值的Bellman公式,如公式(9)所示:

$$Q^h(x,u)=\hat{r}(x,u)+\gamma E_{x'}\{E_{u'}\{Q^h(x',u')\}\} \tag{9}$$

其中,$\hat{r}(x,u)=\int_{\mathbb{R}}p(r\mid x,u)r\mathrm{d}r$,$E_{u'}\{Q^h(x',u')\}=\int_U p(u'\mid x')Q^h(x',u')\mathrm{d}u'$.

在强化学习中,能够获得最大化期望回报的策略称为最优策略,用h^*表示,与之相对应的最优状态值函数以及最优动作值函数分别为$V^*(x)$和$Q^*(x,u)$.因此,对于任意策略h及任意状态x或者状态动作对(x,u),都存在$V^*(x)\geqslant V^h(x)$或者$Q^*(x,u)\geqslant Q^h(x,u)$.对于一个强化学习问题,可能存在多个最优策略,但最优状态值函数或者动作值函数却是唯一的,其更新公式如公式(10)和公式(11)所示:

$$V^*(x)=\bar{r}(x)+\gamma\max_u\int_X p^h(x'\mid x,u)V^*(x')\mathrm{d}x' \tag{10}$$

$$Q^*(x,u)=\hat{r}(x)+\gamma\int_X p(x'\mid x,u)\max_{u'}Q^*(x',u')\mathrm{d}x' \tag{11}$$

为了便于描述问题,给出有界MDP的定义(主要是对状态空间、动作空间、奖赏值以及值函数边界的界定).

定义 1(有界 MDP 问题). 假设 Z,X 和 U 都是一个有限集合,其中,$|Z|$,$|X|$,$|U|$分别是集合中元素的个数;立即奖赏值有界;设$\beta=1/(1-\gamma)$,其中,γ为折扣因子,则对于$\forall x\in X$ 及$\forall(x,u)\in Z$,$0\leqslant V(x)\leqslant\beta C$ 和 $0\leqslant Q(x,u)\leqslant\beta C$ 成立,其中, C 是常数.

在有界 MDP 问题中,当求得最优状态值函数或者动作值函数时,即 V^*或者 Q^*,可以根据相应的值函数求得最优策略 h^*.对于任意状态 x,最优动作 $h^*(x)$如公式(12)或者公式(13)所示:

$$h^*(x)=\arg\max_u\{\hat{r}(x,u)+\int_X p(x'\mid x,u)V^*(x')\mathrm{d}x\} \tag{12}$$

$$h^*(x)=\arg\max_u Q^*(x,u) \tag{13}$$

2 高斯过程

高斯过程(Gaussian process,简称 GP)是一组随机变量的集合,每一个随机变量都包含一个输入变量 $x(x\in X)$,其中,任意有限个随机变量都服从联合高斯分布[17].用 F 表示一个高斯过程,对于任意给定的 $x\in X$,$F(x)$就是其中一个随机变量,且与其他随机变量服从联合高斯分布.公式(14)给出一个高斯过程的形式化表示:

$$F\sim GP(m,k) \tag{14}$$

其中,m 是均值函数,k 是协方差函数,$F(x)\sim N(m(x),k(x,x))$.

在利用高斯过程求解问题的过程中,需要构造合适的概率生成模型,通常需要包含以下 3 个要素:

(1) 构造一个连接可观测随机过程与不可观测随机过程的等式,且等式中通常还包含一个噪声项.

(2) 指定噪声项服从某一概率分布.等式中的噪声项也可以理解为一个随机过程.

(3) 指定不可观测随机过程服从某一先验概率分布,这也是进行贝叶斯推理的前提.

根据以上分析,给出一个线性概率生成模型的形式化表示,如公式(15)所示:

$$Y=HF+N \tag{15}$$

其中,H 是线性操作,F 是不可观测的随机过程,Y 是可观测的随机过程,N 是噪声项.F 和 N 都服从某一高斯分布,且两者之间相互独立.

在学习过程中,对于一组给定的样本数据 $\{(x_i,y_i)\}_{i=1}^t$, 其中,$x_i\in X$ 是输入变量,$y_i\in Y$ 是可观测变量,可以得到 t 组等式,如公式(16)所示:

$$\boldsymbol{Y}_t=\boldsymbol{H}_t\boldsymbol{F}_t+\boldsymbol{N}_t \tag{16}$$

其中,$\boldsymbol{Y}_t=(Y(x_1),\ldots,Y(x_t))^T$,$\boldsymbol{H}_t$ 是一个 $t\times t$ 维的矩阵,$\boldsymbol{F}_t=(F(x_1),\ldots,F(x_t))^T$,$\boldsymbol{N}_t=(N(x_1),\ldots,N(x_t))^T$.

3 基于高斯过程的带参近似策略迭代算法

本文主要考虑在确定环境问题中,利用带参线性函数对值函数进行建模,再利用高斯过程对值函数空间进行建模,并给定某一先验信息,构造合适的概率生成模型,根据贝叶斯理论,求得高斯过程的后验,即值函数空间的后验;并在学习的过程中,利用值函数的后验分布,求得动作的信息价值增益,结合值函数的期望选择相应动作,平衡学习过程中的探索和利用,加快算法的收敛.

3.1 基于高斯过程的值函数参数估计

本文主要是利用线性函数对动作值函数进行建模,公式(17)给出动作值函数的形式化表示方法:

$$Q(x,u)=\sum_{i=1}^n\phi_i(x,u)\omega_j=\Phi(x,u)^TW \tag{17}$$

其中,x 是当前的状态,u 是当前的动作,ϕ_i 是第 i 个基函数,ω_i 是第 i 个参数,$\Phi=\{\phi_1,\phi_2,\ldots,\phi_n\}^T$ 是由 n 个基函数所构造的特征向量,$W=\{\omega_1,\omega_2,\ldots,\omega_n\}$是一个 n 维的参数向量.

假设 1. 假设 W 中任意两个参数之间相互独立,且 W 先验服从标准 n 维高斯分布,即 $W\sim N(0,\boldsymbol{I})$,$\boldsymbol{I}$ 是 $n\times n$ 的单位矩阵.

利用高斯过程对值函数空间进行建模,为了与强化学习中的符号表示一致,高斯过程用 $\boldsymbol{Q}$ 表示,即 $Q(z)$是当

前状态动作对 z 的动作值函数,其中,$z\in Z$.根据假设 1 及公式(17),给出动作值函数的期望以及任意两个动作值函数的协方差,如公式(18)和公式(19)所示:

$$E\{Q(z)\}=E\{\Phi(z)^TW\}=\Phi(z)^TE\{W\}=0 \tag{18}$$

$$Cov\{Q(z),Q(z')\}=Cov\{\Phi(z)^TW,\Phi(z')^TW\}=\Phi(z)^T\Phi(z') \tag{19}$$

其中,$m(z)\triangleq E\{Q(z)\}$,$k(z,z')\triangleq Cov\{Q(z),Q(z')\}=\Phi(z)^T\Phi(z')$.即在先验高斯过程中,对于$\forall z\in Z$,

$$Q(z)\sim N(m(z),k(z,z')).$$

因此,关于动作值函数的高斯过程的形式化表示如公式(20)所示:

$$\boldsymbol{Q}\sim GP(m,k) \tag{20}$$

在确定环境问题的在线学习过程中,公式(9)可以改写为公式(21):

$$Q(z)=\hat{r}(z)+\gamma Q(z') \tag{21}$$

其中,z'是在当前策略下,状态动作对 z 的后续状态动作对;$\hat{r}(z)$ 是当前状态动作对下立即奖赏 $r(z)$的期望.因此,$\hat{r}(x)$ 可以用公式(22)来表示:

$$\hat{r}(z)=r(z)-N(z) \tag{22}$$

其中,N 是噪声项.将公式(22)带入公式(21),可得公式(23):

$$r(z)=Q(z)-\gamma Q(z')+N(z) \tag{23}$$

假设 2. 假设各状态动作对的立即奖赏的噪声项相互独立且服从高斯分布,均值为 0,方差为$\sigma^2(z)$,即

$$N(z)\sim N(0,\sigma^2(z)).$$

假设给定一组包含 t+1 个状态的样本序列 $\{z_i,r(z_i),z_{i+1}\}_{i=0}^{i=t-1}$,其中,$z_{i+1}$ 是 z_i 的后续状态动作对,则对于第 i 个样本,$\hat{r}(z_i)=r(z_i)-N(z_i)$.将这一组样本的动作值函数、立即奖赏以及噪声分别写成向量的形式,如公式(24)~公式(26)所示:

$$\boldsymbol{Q}_t=(Q(z_0),Q(z_1),\ldots,Q(z_t))^T \tag{24}$$

$$\boldsymbol{r}_{t-1}=(r(z_0),r(z_1),\ldots,r(z_{t-1}))^T \tag{25}$$

$$\boldsymbol{N}_{t-1}=(N(z_0),N(z_1),\ldots,N(z_{t-1}))^T \tag{26}$$

根据假设 2 以及公式(26),噪声向量 $\boldsymbol{N}_{t-1}$ 的分布形式如公式(27)所示:

$$\boldsymbol{N}_{t-1}\sim N(\boldsymbol{0},\Sigma_t)\ \text{其中},\Sigma_t=\begin{pmatrix}\sigma_0^2 & 0 & \cdots & 0\\ 0 & \sigma_1^2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \sigma_{t-1}^2\end{pmatrix} \tag{27}$$

其中$\sigma_t=\sigma(z_t)$.根据这组样本序列及公式(23),可得一个包含 t 个等式的线性方程组,如公式(28)所示:

$$\boldsymbol{r}_{t-1}=\boldsymbol{H}_t\boldsymbol{Q}_t+\boldsymbol{N}_{t-1} \tag{28}$$

其中,$\boldsymbol{H}_t$ 是一个 $t\times(t+1)$的矩阵,如公式(29)所示:

$$\boldsymbol{H}_t=\begin{pmatrix}1 & -\gamma & 0 & \cdots & 0\\ 0 & 1 & -\gamma & \cdots & 0\\ \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & \cdots & 1 & -\gamma\end{pmatrix} \tag{29}$$

将公式(17)带入公式(28),可得关于参数向量 W 的线性方程组,如公式(30)所示:

$$\boldsymbol{r}_{t-1}=\boldsymbol{H}_t\boldsymbol{\Phi}_t^TW+\boldsymbol{N}_{t-1} \tag{30}$$

其中,$\boldsymbol{\Phi}_t^T$ 是一个$(t+1)\times n$ 的矩阵,即$\boldsymbol{\Phi}_t^T=(\Phi(z_0),\Phi(z_2),\ldots,\Phi(z_t))^T$,$\Phi(z_i)=(\phi_1(z_i),\phi_2(z_i),\ldots,\phi_n(z_i))^T$.公式(30)是在进行贝叶斯推理过程中所用到的概率生成模型.

定理 1. 在假设 1、假设 2 成立的条件下,对于一组给定的样本序列 $\{z_i,r(z_i),z_{i+1}\}_{i=0}^{i=t-1}$,其中,立即奖赏向量 $\boldsymbol{r}_{t-1}$ 的分布满足公式(31):

$$\boldsymbol{r}_{t\text{-}1}\sim N(0,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t) \tag{31}$$

证明:根据假设 1、假设 2 可知,$W\sim N(0,\boldsymbol{I})$,$\boldsymbol{N}_{t-1}\sim N(\boldsymbol{0},\Sigma_t)$.结合公式(30)可知:

$$E\{\boldsymbol{r}_{t-1}\}=E\{\boldsymbol{H}_t\boldsymbol{\Phi}_t^TW+\boldsymbol{N}_{t-1}\}=0,$$

$$Cov(\boldsymbol{r}_{t-1},\boldsymbol{r}_{t-1})=Cov(\boldsymbol{H}_t\boldsymbol{\Phi}_t^TW+\boldsymbol{N}_{t-1},\boldsymbol{H}_t\boldsymbol{\Phi}_t^TW+\boldsymbol{N}_{t-1})=\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t.$$

因此,$\boldsymbol{r}_{t-1}\sim N(0,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t)$. □

定理 2. 假设 A 是一个 $n\times m$ 的矩阵,B 是一个 $m\times n$ 的矩阵,且 $BA+\boldsymbol{I}$ 是非奇异矩阵,那么 $AB+\boldsymbol{I}$ 也是非奇异矩阵,且 $A(BA+\boldsymbol{I})^{-1}=(AB+\boldsymbol{I})^{-1}A$.

证明:利用反证法证明.假设 $AB+\boldsymbol{I}$ 是奇异矩阵,则存在一个向量 $a\neq0$,使得公式(32)成立:

$$(AB+\boldsymbol{I})a=0 \tag{32}$$

在公式(32)两边同时左乘 B,得到公式(33):

$$B(AB+\boldsymbol{I})a=(BA+\boldsymbol{I})Ba=0 \tag{33}$$

令 $b\triangleq Ba$,根据公式(32)可知,$a=-ABa=-Ab$.若 $b=0$,则 $a=0$,因此,$b\neq0$.再根据公式(33)可知,$BA+\boldsymbol{I}$ 也是奇异矩阵,与条件不符.因此,如果 $BA+\boldsymbol{I}$ 是非奇异矩阵,则 $AB+\boldsymbol{I}$ 也是非奇异矩阵.

当 $AB+\boldsymbol{I}$ 是非奇异矩阵,即 $AB+\boldsymbol{I}$ 存在逆矩阵时,可得:

$$\begin{aligned}A(BA+\boldsymbol{I})^{-1}&=(AB+\boldsymbol{I})^{-1}(AB+\boldsymbol{I})A(BA+\boldsymbol{I})^{-1}\\&=(AB+\boldsymbol{I})^{-1}A(BA+\boldsymbol{I})(BA+\boldsymbol{I})^{-1}\\&=(AB+\boldsymbol{I})^{-1}A.\end{aligned}$$

因此,$A(BA+\boldsymbol{I})^{-1}=(AB+\boldsymbol{I})^{-1}A$. □

定理 3. 在假设 1、假设 2 成立的条件下,对于一组给定的样本序列 $\{z_i,r(z_i),z_{i+1}\}_{i=0}^{i=t-1}$,参数向量 W 与立即奖赏向量 $\boldsymbol{r}_{t-1}$ 的联合概率分布如公式(34)所示,且参数向量的后验 $W|\boldsymbol{r}_{t-1}$ 满足公式(35):

$$\begin{pmatrix}W\\\boldsymbol{r}_{t-1}\end{pmatrix}\sim N\left\{\begin{pmatrix}\boldsymbol{0}\\\boldsymbol{0}\end{pmatrix},\begin{pmatrix}\boldsymbol{I}&\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\\\boldsymbol{H}_t\boldsymbol{\Phi}_t^T&\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t\end{pmatrix}\right\} \tag{34}$$

$$W\,|\,\boldsymbol{r}_{t-1}\sim N((\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{r}_{t-1},(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}) \tag{35}$$

证明:根据 W 的先验以及概率生成模型公式(30)可知:

$$Cov(W,\boldsymbol{r}_{t-1})=Cov(W,\boldsymbol{H}_t\boldsymbol{\Phi}_t^TW+\boldsymbol{N}_{t-1})\ =\boldsymbol{\Phi}_t\boldsymbol{H}_t^T.$$

结合定理 1,可得:

$$\begin{pmatrix}W\\\boldsymbol{r}_{t-1}\end{pmatrix}\sim N\left\{\begin{pmatrix}\boldsymbol{0}\\\boldsymbol{0}\end{pmatrix},\begin{pmatrix}\boldsymbol{I}&\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\\\boldsymbol{H}_t\boldsymbol{\Phi}_t^T&\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t\end{pmatrix}\right\}.$$

因此,公式(34)得证.

根据公式(34)、条件高斯分布的后验形式以及定理 2,可知:

$$E\{W\,|\,\boldsymbol{r}_{t-1}\}=\boldsymbol{\Phi}_t\boldsymbol{H}_t^T(\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t)^{-1}\boldsymbol{r}_{t-1}=(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{r}_{t-1},$$

$$Cov(W\,|\,\boldsymbol{r}_{t-1},W\,|\,\boldsymbol{r}_{t-1})=\boldsymbol{I}-\boldsymbol{\Phi}_t\boldsymbol{H}_t^T(\boldsymbol{H}_t\boldsymbol{\Phi}_t^T\boldsymbol{\Phi}_t\boldsymbol{H}_t^T+\Sigma_t)^{-1}\boldsymbol{H}_t\boldsymbol{\Phi}_t^T=(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}.$$

因此,$W\,|\,\boldsymbol{r}_{t-1}\sim N((\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{r}_{t-1},(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\,\Sigma_t^{-1}\,\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1})$. □

3.2 基于VPI的动作选择方法

探索的目的是为了找到更好的策略,因此在学习过程中,需要格外关注能够改变动作选择策略的信息.通常有两种情况能够导致策略发生变化:

1) 新信息显示当前状态下的非最优动作要优于最优动作;

2) 新信息显示当前状态下的最优动作要劣于次优动作.

对于第 1 种情况,假设动作 u_1 是最优动作,即 $\forall u'\neq u_1,E\{Q(x,u_1)\}\geqslant E\{Q(x,u')\}$,且新信息显示动作 u 可能是一个更优的动作,即 $E\{Q(x,u_1)\}\leqslant Q(x,u)$,其中,$Q^*(x,u)$是在状态 x 下可能获得的较优动作值函数.因此,在执行动作 u 时,可以获得额外的奖励,即 $Q^*(x,u)-E\{Q(x,u_1)\}$.

对于第 2 种情况,假设动作 u_1 是最优动作,u_2 是次优动作,即$\forall u'\neq u_1,u'\neq u_2,E\{Q(x,u_1)\}\geqslant E\{Q(x,u_2)\}\geqslant E\{Q(x,u')\}$,且信息显示当前的最优动作 u_1 可能会劣于次优动作 u_2.即在动作 u_1 下,可能获得某一次的值函数 $Q'(x,u_1),Q'(x,u_1)<E\{Q(x,u_2)\}$,即通过执行动作 u_2 可以得到额外的奖励,即 $E\{Q(x,u_2)\}-Q'(x,u_1)$.因此,根据以上两种情况,可以得到公式(36):

$$Info_{x,u}(Q(x,u))=\begin{cases}Q(x,u)-E\{Q(x,u_1)\}, & 当u\neq u_1,且Q(x,u)=Q^*(x,u)>E\{Q(x,u_1)\}\\ E\{Q(x,u_2)\}-Q(x,u), & 当u=u_1,且Q(x,u)=Q'(x,u)<E\{Q(x,u_2)\}\\ 0, & 其他\end{cases} \tag{36}$$

其中,$u_1,u_2\in U$,且 u_1 是最优动作,u_2 是次优动作.然而,强化学习是一种在线的学习方法,在学习过程中,Agent 无法事先知道所采取的动作可能获取的较优或者较劣的值函数——$Q^*(x,u)$或者 $Q'(x,u)$,因此只能根据先验信息进行估计,计算当前动作下的额外奖赏的期望,即信息价值增益(value of perfect information,简称 VPI).

定义 2(信息价值增益). 信息价值增益是指针对所考虑的动作,根据先验信息所能获取关于该动作的额外信息,并可以利用该额外信息指导动作的选择.信息价值增益的计算如公式(37)所示:

$$VPI(x,u)=\int_{-\infty}^{\infty}Info_{x,u}(y)P(Q(x,u)=y)\mathrm{d}y \tag{37}$$

其中,$P(Q(x,u)=y)$是当前状态动作对(x,u)的值函数为 y 时的概率密度.

在在线学习的过程中,信息价值增益主要用于指导动作的选择,不参与任何值函数的修正.在获取信息价值增益的情况下,动作的选择需满足公式(38):

$$u=\arg\max_u\{E\{Q(x,u)\}+VPI(x,u)\} \tag{38}$$

3.3 GPPAPI

结合第 3.1 节的参数估计方法以及第 3.2 节的动作选择策略,给出基于高斯过程的带参近似策略迭代算法(Gaussian process-based parametric approximation policy iteration,简称 GPPAPI).由于所设计的生成模型具有局限性,算法只能用于确定环境下的问题求解.接下来将详细介绍算法的求解方法.为了便于计算,做如下定义,令:

$$P_t=Cov(W\mid \boldsymbol{r}_{t-1},W\mid \boldsymbol{r}_{t-1})=(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\Sigma_t^{-1}\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1},$$
$$W_t=E\{W\mid \boldsymbol{r}_{t-1}\}=(\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\Sigma_t^{-1}\boldsymbol{H}_t\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1}\boldsymbol{\Phi}_t\boldsymbol{H}_t^T\Sigma_t^{-1}\boldsymbol{r}_{t-1},$$

其中,$\boldsymbol{\Phi}_t$ 是 $n\times(t+1)$的矩阵,$\boldsymbol{\Phi}_t^T=(\Phi(z_0),...,\Phi(z_t))^T,\Phi(z_i)=(\phi_1(z_i),...,\phi_n(z_i))^T$,$\boldsymbol{H}_t$ 是$(t+1)\times n$ 的矩阵.

再令 $\Delta\Phi_t=\Phi(z_{t-1})-\gamma\Phi(z_t),\Delta\boldsymbol{\Phi}_t=\boldsymbol{\Phi}_t\boldsymbol{H}_t^T=(\Delta\Phi_1,\Delta\Phi_2,...,\Delta\Phi_t)$, 则 P_t,W_t 可分别由公式(39)、公式(40)表示:

$$P_t=(\Delta\boldsymbol{\Phi}_t\Sigma_t^{-1}\Delta\boldsymbol{\Phi}_t^T+\boldsymbol{I})^{-1} \tag{39}$$

$$W_t=E\{W\mid \boldsymbol{r}_{t-1}\}=P_t\Delta\boldsymbol{\Phi}_t\Sigma_t^{-1}\boldsymbol{r}_{t-1} \tag{40}$$

再分别令:

$$B_t=\Delta\boldsymbol{\Phi}_t\Sigma_t^{-1}\Delta\boldsymbol{\Phi}_t^T=\sum_{i=1}^{t}\frac{1}{\sigma_{i-1}^2}\Delta\Phi_i\Delta\Phi_i^T=B_{t-1}+\frac{1}{\sigma_{t-1}^2}\Delta\Phi_t\Delta\Phi_t^T,$$
$$b_t=\Delta\boldsymbol{\Phi}_t\Sigma_t^{-1}\boldsymbol{r}_{t-1}=\sum_{i=1}^{t}\frac{1}{\sigma_{i-1}^2}\Delta\Phi_i r(z_{t-1})=b_{t-1}+\frac{1}{\sigma_{t-1}^2}\Delta\Phi_t r(z_{t-1}),$$

则公式(39)、公式(40)可以写成公式(41)、公式(42):

$$P_t=(B_t+\boldsymbol{I})^{-1} \tag{41}$$

$$W_t=P_tb_t \tag{42}$$

对公式(41)做如下变化:

$$P_t=(B_t+\boldsymbol{I})^{-1}=\left(B_{t-1}+\boldsymbol{I}+\frac{1}{\sigma_{t-1}^2}\Delta\Phi_t\Delta\Phi_t^T\right)^{-1}=P_{t-1}-(\sigma_{t-1}^2+\Delta\Phi_t^TP_{t-1}\Delta\Phi_t)^{-1}P_{t-1}\Delta\Phi_t\Delta\Phi_t^TP_{t-1},$$

令 $k_t=(\sigma_{t-1}^2+\Delta\Phi_t^TP_{t-1}\Delta\Phi_t)^{-1}P_{t-1}\Delta\Phi_t\Delta\Phi_t^T$, 则公式(41)可以写成公式(43):

$$P_t=P_{t-1}-k_tP_{t-1} \tag{43}$$

对公式(42)做如下变化,可以写成公式(44):

$$W_t = P_t b_t = \left(P_{t-1} - k_t P_{t-1} \right)\left(b_{t-1} + \frac{1}{\sigma_{t-1}^2} \Delta \Phi_t r(z_{t-1}) \right) = (I - k_t) W_{t-1} + \frac{1}{\sigma_{t-1}^2} P_t \Delta \Phi_t r(z_{t-1}) \tag{44}$$

接下来,给出完整的 GPPAPI 算法.算法的执行流程如算法 1 所示.

算法 1. GPPAPI 算法.

Step 1. 初始化.令 $W_0=\mathbf{0}$,$P_0=\boldsymbol{I}$,且 $\forall i \in \mathbb{R}, \sigma_i^2 = \sigma$.

Step 2. 令当前的采样序号为 t,初始 t=1,且当前初始状态为 $x_{t-1}=x_0$.

Step 3. 根据公式(37)以及当前参数向量的分布,计算当前状态 x_{t-1} 下所有动作的 VPI,并根据公式(38)选择相应的动作 u_{t-1}.

Step 4. 执行动作 u_{t-1},立即奖赏 $r(x_{t-1},u_{t-1})$,后续状态 x_t.计算 x_t 下所有动作的 VPI,并选择相应的动作 u_t.

Step 5. 计算 $\Delta \Phi_t = \Phi(x_{t-1},u_{t-1}) - \gamma \Phi(x_t,u_t)$ 及 $k_t = (\sigma_{t-1}^2 + \Delta \Phi_t^T P_{t-1} \Delta \Phi_t)^{-1} P_{t-1} \Delta \Phi_t \Delta \Phi_t^T$.

Step 6. 根据公式(43)、公式(44)计算 P_t 和 W_t.

Step 7. 令 $t=t+1$,并跳转至 Step 4.

4 实验结果分析

为了验证算法的有效性,将 GPPAPI 算法、最小二乘策略迭代算法以及基于函数近似的 Sarsa 算法用于强化学习中的 Mountain Car 问题.Mountain Car 是强化学习中一个经典的连续状态空间、情节式任务,如图 1 所示.

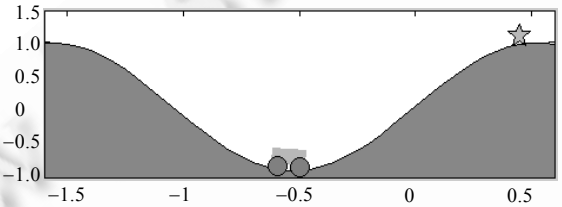


Fig.1 Mountain Car problem

图 1 Mountain Car 问题

Mountain Car 描述的是一个动力不足的小车爬坡的过程,如图 1 所示.假设小车处于坡底,由于动力不足,无法再直接加速冲上坡顶,因此必须借助惯性到达坡顶,即图 1 中最右侧星形标记的位置.其中,状态包含两个维度——位置和速度(为了简化问题,仅考虑小车的水平位置),用 p 和 v 表示,则状态 $x=[p,v]^T$;在任意时刻,小车存在 3 个可选动作,分别是向右加速,向左加速和不加速,分别用+1,−1 和 0 表示,即动作 u={+1,−1,0};路面可以用函数 h=sin(3p)表示;状态转移函数如公式(45)所示:

$$\begin{cases} v_{t+1} = bound[v_t + 0.001u_t + g\cos(3p_t)] \\ p_{t+1} = bound[p_t + 1] \end{cases} \tag{45}$$

其中,*bound* 是限界函数,即 $bound(v_t) \in [-0.07,+0.07]$,$bound(p_t) \in [-1.2,+0.5]$;$g$ 是重力加速度,且 $g=-0.0025$.

在情节开始时,给定小车一个随机位置和速度,然后进行交互,当小车到达目标位置或者当前执行的时间步超过 1 000 时,情节终止并开始新的情节.当小车到达目标位置时,立即奖赏是 1;在其他情况下,小车的立即奖赏是 0.具体的奖赏函数如公式(46)所示:

$$\rho_t(x_t,u_t,x_{t+1}) = \begin{cases} 0, & p_{t+1} = p^* \\ -1, & -1.2 < p_{t+1} < p^* \end{cases} \tag{46}$$

在实验过程中,利用高斯径向基函数进行编码,将状态的两个维度分别分成 6 份,一共包含 6×6 个径向基函数,其中,高斯径向基函数如公式(47)所示:

$$\phi(x,\bar{x}_i) = \exp\left(-\frac{(p-\bar{p}_i)(v-\bar{v}_i)}{2 \times \tau^2} \right) \tag{47}$$

其中,$\bar{x}_i$ 是第 i 个基函数的中心,即 $\bar{x}_i=[\bar{p}_i,\bar{v}_i]^T$, 且 τ^2=0.82.

图 2 是将 GPPAPI,LSPI 和 Sarsa(λ)用于 Mountain Car 问题的性能比较.为了增加算法比较的合理性,实验中 3 种算法都采用高斯径向基函数进行编码,且都包含 6×6 个径向基函数,基函数如公式(47)所示.此外,在 GPPAPI 中,σ的值是 1;在 LSPI 中,α,ε的值分别是 0.05 和 0.3;在 Sarsa(λ)中,α,ε以及λ的值分别是 0.1,0.3 和 0.9.从图中可以看出,GPPAPI 和 LSPI 的收敛性能要明显优于 Sarsa(λ).其中,GPPAPI 收敛后,小车到达目标位置所需要的时间步大约是 83 步;在 LSPI 中,小车大约需要 87 步到达目标位置,而 Sarsa(λ)收敛后,小车到达目标位置所需的时间步稳定在 135 步.另外,GPPAPI 大约在 47 个情节之后,算法可以收敛;LSPI 大约在 73 个情节之后,算法可以认为收敛(该算法在收敛后,依然会出现一定的波动);而 Sarsa(λ)大约在 97 个情节之后才能收敛.因此,从收敛性能和收敛速度的角度来看,GPPAPI 算法都要优于 LSPI 以及 Sarsa(λ).另外,GPPAPI 在收敛前期,算法的波动比较大,主要是由于动作选择策略以及值函数的方差较大导致的,当值函数的方差较大时,可以认为对于值函数的估计不够准确,且 VPI 值的波动较大.因此,根据公式(38)选择的动作具有较大的探索性能,反映在算法的收敛曲线上,就是曲线的波动性比较大;但是当算法趋于收敛时,值函数的估计趋于准确,VPI 的值将逐渐减少到 0,算法的动作选择策略趋于贪心,因此在算法收敛后,具有较为平稳的收敛性能.

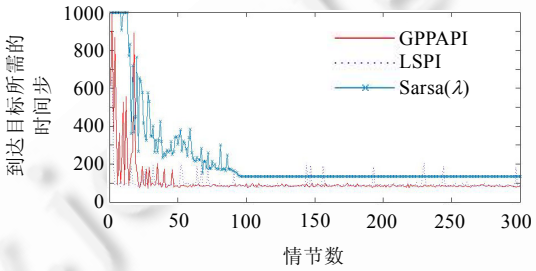


Fig.2 Convergence performance of algorithms on Mountain Car problem

图 2 算法在 Mountain Car 问题上的收敛性能

表 1 主要用于分析 GPPAPI,LSPI 以及 Sarsa(λ)在 Mountain Car 问题上的计算性能.表中主要列举两个性能指标——算法收敛所需的时间以及计算每个样本所需的时间.通过多次重复实验,GPPAPI 大约需要 47 个情节可以收敛,总计约有 10 293 个样本,算法收敛大约需要 20s,平均每个样本的计算时间是 0.00 2s;LSPI 大约需要 73 个情节可以收敛,总计约有 8 660 个样本,算法收敛大约需要 44s,每个样本的平均计算时间是 0.005s;而 Sarsa(λ)大约需要 97 个情节可以收敛,总计约有 37 985 个样本,每个样本的平均计算时间是 0.000 05s.因此,在计算性能的角度上考虑,GPPAPI 要优于 LSPI,劣于 Sarsa(λ),但是结合图 2 可以得出,与 LSPI 相比,GPPAPI 在需要较少计算量的情况下,可以获得更优的收敛性能.从算法执行的角度分析各个算法的计算性能,LSPI 在算法执行过程中需要进行矩阵的求逆操作;GPPAPI 仅需要进行矩阵的乘法;而 Sarsa(λ)是利用梯度下降的算法更新参数.因此,从计算量的角度分析,Sarsa(λ)要优于 GPPAPI,而 GPPAPI 要优于 LSPI.此外,在实验的过程中发现:当径向基函数的数量超过 10×10 时,LSPI 的计算速度已经远远不能满足收敛要求,而 GPPAPI 依然可以收敛.

Table 1 Computation performance comparison of algorithms on Mountain Car

表 1 算法在 Mountain Car 问题上的计算性能比较

	收敛所需时间(s)	计算每个样本所需时间(s)
GPPAPI	20	0.002
LSPI	44	0.005
Sarsa(λ)	2	0.000 05

图 3 主要通过与传统ε-greedy 方法比较(其中,ε的值是 0.3)来分析基于 VPI 的动作选择方法对算法收敛性能的影响.从图 3 中可以直观地看出:利用基于 VPI 的动作选择方法的 GPPAPI 算法收敛所需要的情节数较少,而利用ε-greedy 的 GPPAPI 算法大约需要 86 个情节才能收敛;且收敛之后存在一定的波动,而前者的收敛曲线

相对平稳.在算法的执行过程中,需要计算参数向量的期望以及协方差矩阵,而在算法收敛前期,参数向量的期望以及协方差矩阵的变化较大,在算法收敛后期,期望趋于稳定,且每个参数的方差逐渐趋于 0.根据 VPI 的计算规则可以得出:在算法收敛前期,VPI 的值相对较大,有利于探索;而在算法收敛后期,VPI 的值也将逐渐趋于 0,动作选择策略也逐渐趋于贪心策略,因此在该阶段,策略相对比较稳定,小车达到目标位置所需要的时间步也趋于稳定.与ε-greedy 方法相比,基于 VPI 的动作选择方法对于平衡探索和利用的问题是一种更好的解决方案,其中,ε-greedy 方法本质上是一种盲目、随机的动作选择方法,而基于 VPI 的动作选择方法可以看成是一种自适应的动作选择方法.因此,与基于ε-greedy 的 GPPAPI 相比,基于 VPI 的动作选择方法的 GPPAPI 算法能够在需要较少情节数的情况下达到收敛,且收敛曲线相对更为稳定,收敛性能更为理想.

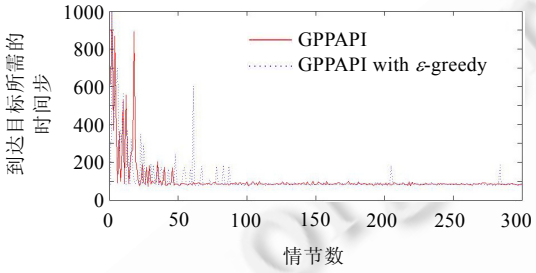


Fig.3 Effect of action selection based on VPI on GPPAPI

图 3 基于 VPI 的动作选择方法对 GPPAPI 算法的影响

5 结 论

本文主要针对值函数的参数估计以及学习过程中的探索和利用问题,提出一种基于高斯过程的带参近似策略迭代算法.该算法利用高斯过程对带参的值函数进行建模,并根据 Bellman 公式建立确定环境问题下的概率生成模型,利用贝叶斯推理求解值函数参数的后验分布.与传统的点估计方法相比,该方法具有更优的计算性能,不仅可以获得参数向量的期望,而且可以给出参数向量相应的协方差矩阵,提高计算的精确性.同时,在算法执行过程中,通过求解动作的信息价值增益指导动作的选择,在一定程度上可以解决探索和利用的平衡问题.以 Mountain Car 问题作为实验平台,从多个角度对 GPPAPI 算法的收敛性能进行分析,通过与 LSPI 以及 Sarsa(λ)进行比较,得出 GPPAPI 算法具有较优的收敛性能.与 LSPI 方法相比,GPPAPI 方法具有更好的计算性能,且参数较少,对于不同的问题具有更好的鲁棒性.

本文主要关注确定环境、连续状态空间且离散动作空间的强化学习问题的求解,但在很多实际问题中需要考虑随机环境的问题,因此,下一步的工作就是要考虑在随机环境中如何建立合理的概率生成模型,利用基于高斯过程的值函数估计方法求解.另外,本文考虑的动作空间是离散的,因此,给出的策略也是基于离散动作空间的.如何将本文提出的方法进一步延伸至连续动作空间问题,也是下一步需要做的工作.

References:

[1] Sutton RS, Barto GA. Reinforcement Learning. Cambridge: MIT Press, 1998.

[2] Sutton RS. Learning to predict by the method of temporal differences. Machine Learning, 1988,3(1):9−44.

[3] Liu Q, Fu QM, Gong SR, Fu YC, Cui ZM. Reinforcement learning algorithm based on minimum state method and average reward. Journal on Communications, 2011,32(1):66−71 (in Chinese with English abstract). [doi: 10.1109/CSIE.2009.433]

[4] Tadic V. On the convergence of temporal-difference learning with linear function approximation. Machine Learning, 2001,42(3): 241−267. [doi:10.1023/A:1007609817671]

[5] Shah H, Gopal M. Fuzzy decision tree function approximation in reinforcement learning. Int'l Journal of Artificial Intelligence and Soft Computing, 2010,2(1-2):26−45. [doi: 10.1504/IJAISC.2010.032511]

[6] Precup D, Sutton RS, Dasgupta S. Off-Policy temporal-difference learning with function approximation. In: Proc. of the 18th Int'l Conf. on Machine Learning. Morgan Kaufmann Publishers, 2001. 417−424.

[7] Lagoudakis M, Parr R. Least squares policy iteration. Journal of Machine Learning Research, 2003,4(12):1107−1149.

[8] Engel Y, Mannor S, Meir R. Reinforcement learning with Gaussian processes. In: Proc. of the 22nd Int'l Conf. on Machine Learning. ACM Press, 2005. 201−208. [doi: 10.1145/1102351.1102377]

[9] Sutton RS, Szepesvári C, Maei HR. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In: Proc. of the 22nd Annual Conf. on Neural Information Processing Systems. Cambridge: MIT Press, 2008. 1609−1616.

[10] Sutton RS, Hamid RM, Precup D, Bhatnagar S, Silver D, Szepesvári C, Wiewiora E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proc. of the 26th Int'l Conf. on Machine Learning. ACM Press, 2009. 993−1000. [doi: 10.1145/1553374.1553501]

[11] Dearden R, Friedman N, Russell S. Bayesian Q-learning. In: Proc. of the 15th National/10th Conf. on Artificial Intelligence/ Innovative Applications of Artificial Intelligence. AAAI Press, 1998. 761−768.

[12] Ghavamzadeh M, Engel Y. Bayesian actor-critic algorithms. In: Proc. of the 24th Int'l Conf. on Machine Learning. ACM Press, 2007. 297−304. [doi: 10.1145/1273496.1273534]

[13] Dimitrakakis C, Rothkopf CA. Bayesian multitask inverse reinforcement learning. In: Proc. of the 9th European Workshop on Reinforcement Learning. Springer-Verlag, 2012. 273−284. [doi: 10.1007/978-3-642-29946-9_27]

[14] Xu X, Hu DW, Lu XC. Kernel-Based least squares policy iteration for reinforcement learning. IEEE Trans. on Neural Networks, 2007,18(4):973−992. [doi: 10.1109/TNN.2007.899161]

[15] Wingate D, Goodman ND, Roy DM, Kaelbling LP, Tenenbaum JB. Bayesian policy search with policy priors. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence. AAAI Press, 2011. 1565−1570.

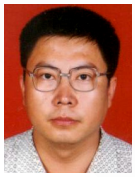
[16] Ross S, Pineau J. Model-Based Bayesian reinforcement learning in large structured domains. In: Proc. of the 24th Conf. on Uncertainty in Artificial Intelligence. AUAI Press, 2008. 476−483.

[17] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. Cambridge: MIT Press, 2006.

附中文参考文献:

[3] 刘全,傅启明,龚声蓉,伏玉琛,崔志明.一种最小状态变元平均报酬的强化学习方法.通信学报,2011,32(1):66−71.

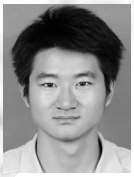
傅启明(1985－),男,江苏淮安人,博士生,CCF 学生会员,主要研究领域为强化学习,贝叶斯推理,遗传算法.
E-mail: fqm_1@126.com



刘全(1969－),男,博士,教授,博士生导师,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

伏玉琛(1968－),男,博士,副教授,主要研究领域为机器学习,数据挖掘.
E-mail: yuchenfu@suda.edu.cn

周谊成(1990－),男,硕士生,主要研究领域为强化学习.
E-mail: zyc9012@163.com



于俊(1989－),男,硕士生,主要研究领域为强化学习.
E-mail: preferyu@gmail.com