第 38 卷　第 4 期
2023 年 8 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 38　No. 4
Aug. 2023

DOI:10.13364/j.issn.1672-6510.20220246

基于最优向量基线的参数探索策略梯度算法

赵婷婷，李　坤，刘展硕，陈亚瑞，王　嫄，杨巨成
（天津科技大学人工智能学院，天津 300457）

摘　要：策略梯度算法是深度强化学习领域中广泛使用的一类无模型强化学习方法，在实际应用中取得了突破性进展。策略梯度算法一直受到梯度估计方差大的困扰，基于参数探索的策略梯度算法（policy gradients with parameter-based exploration，PGPE）从根本上缓解了该问题。通过最优基线技术的引入，策略梯度估计的方差进一步减小。然而，现有最优基线技术只使用标量值作为基线，忽略了策略梯度各维度之间的差异。针对此问题，本文提出一种向量基线概念并推导 PGPE 算法的最优向量基线表示，在理论上证明了引入最优向量基线的 PGPE 算法可以得到更小的梯度估计方差，并且实验验证了此算法的有效性。

关键词：深度强化学习；策略梯度；梯度估计；方差
中图分类号：TP391　　**文献标志码**：A　　**文章编号**：1672-6510（2023）04-0069-07

Policy Gradients with Parameter-Based Exploration Based on Optimal Vector Baseline

ZHAO Tingting，LI Kun，LIU Zhanshuo，CHEN Yarui，WANG Yuan，YANG Jucheng
（College of Artificial Intelligence，Tianjin University of Science & Technology，Tianjin 300457，China）

Abstract：Policy gradient methods are a kind of widely used model-free reinforcement learning methods in the field of deep reinforcement learning，which have made breakthrough in practical applications. However，policy gradient methods have been plagued by the large variance of gradient estimation. Policy gradients with parameter-based exploration（PGPE） has alleviated this problem fundamentally. By introducing the optimal baseline technique，the variance of policy gradient estimation has been further reduced. However，the existing optimal baseline techniques only use scalar values as baselines，ignoring the differences between the dimensions of the policy gradient. Therefore，in this article we propose a concept of vector baseline and derive the optimal vector baseline representation for PGPE. We theoretically proved that the PGPE with the optimal vector baseline could obtain a smaller gradient estimation variance. Moreover，the effectiveness of the proposed PGPE based on the optimal vector baseline was verified by experiments.

Key words：deep reinforcement learning；policy gradients；gradient estimation；variance

强化学习是机器学习领域的一个重要分支，智能体通过与未知环境交互找到一种最优策略，使累积回报最大化[1]。随着深度神经网络的引入，深度强化学习在游戏[2-3]、机器人控制[4-5]等领域都取得了巨大成功。

无模型强化学习是强化学习领域中灵活的框架，它无须对环境进行建模，直接学习决策策略。根据策略学习方式的不同，策略学习方法分为基于值函数的学习方法[6]和基于策略的学习方法[7]。基于值函数的学习方法通过值函数对动作进行评估并选择最佳动作，它可以有效处理离散动作空间问题[8]，但此类方法难以找到与动作相关的最大值函数来改进策略，因此无法处理连续动作空间问题。基于值函数的学习方法具有局限性，基于策略的学习方法则直接对策略

收稿日期：2022-11-04；**修回日期**：2023-02-02
基金项目：国家自然科学基金项目（61976156）；天津市企业科技特派员项目（20YDTPJC00560）
作者简介：赵婷婷（1986—），女（蒙古族），内蒙古赤峰人，副教授，tingting@tust.edu.cn

进行建模，利用参数化的函数表示策略，通过寻找最优参数确定最优策略，已广泛用于解决具有连续状态、动作空间的复杂决策强化学习任务[9]。基于演员–评论家（actor-critic，AC）架构[10]在基于策略的学习方法中引入了价值函数，兼备基于策略的学习方法和基于值函数方法两方面的优势，其中 Actor（演员）扮演策略这一角色，用于控制智能体生成动作，而 Critic（评论家）则根据值函数评估智能体动作的好坏并指导 Actor 对策略进行改进。此类方法可以有效解决包括离散动作空间和连续动作空间在内的各种决策问题。

策略梯度方法是基于策略的学习方法中最实用、最易于实现的一种算法[11-14]，它通过使用当前策略与环境交互得到的数据进行策略梯度估计，迭代更新策略参数，如传统策略梯度方法（REINFORCE）[11]、自然策略梯度方法（NPG）[12]等。REINFORCE 算法作为经典的策略梯度方法，在物理控制任务中表现突出，然而策略的随机性使得 REINFORCE 算法在估计梯度时具有较大的方差，导致收敛速度较慢[15-16]。NPG 算法[12]通过使用 KL 散度测量当前策略下路径分布与更新策略下路径分布的距离，使策略参数在得到最大程度的改变时，策略更新前后的路径分布只发生微小的改变，从而保证策略更新过程相对稳定。

为了减轻策略的随机性对策略梯度估计方差的影响，基于参数探索的策略梯度（policy gradients with parameter-based exploration，PGPE）算法[15]通过使用确定性策略函数，将探索引入策略参数的方式大幅度减少了决策过程中的随机扰动，即从策略参数的先验概率分布中抽取策略参数，然后确定性地选择动作，从而提高策略梯度估计的稳定性，从根本上解决了 REINFORCE 算法中梯度估计方差大的问题。然而，PGPE 算法依然需要大量样本才能保证策略梯度估计的稳定性及策略收敛速度。随机梯度下降的收敛速度主要取决于随机梯度的方差[17]，较低的策略梯度估计方差会有较高的采样效率。因此，长期以来人们一直在研究减少策略梯度估计方差的各种方法[18-20]。AC 算法和 λ 加权回归估计[21]使用基于抽样回归和函数近似器的估计代替高方差蒙特卡洛回归，有效地降低了策略梯度估计的方差。另外，统计学中的控制变量方法在不引入偏差的情况下可有效减少蒙特卡洛方法的估计量的方差，被广泛应用于策略梯度算法中，是减小梯度估计方差的代表性方法[22]。基于此，研究人员在策略梯度算法领域中通过基线函数构造控制变量。基线函数是在计算策略梯度时从收益估计中减去的函数，在实践中常通过减去移动平均基线减小策略梯度估计的方差。然而，研究[23]表明移动平均基线在梯度估计方差约减中并不是最优的。为了进一步减小梯度估计的方差，研究者提出了最优基线技术[16, 24-26]。但是，现有的最优基线技术只使用标量值作为基线，忽略了策略梯度各维度之间的差异。

在深度强化学习领域，深度确定性策略梯度（DDPG）方法[13]将 DQN（deep Q-learning）算法[27]中的经验回放机制和目标网络应用在策略搜索方法中，增加了算法的稳定性。DDPG 算法[13]需要对网络模型进行大规模的训练才能收敛，且交互环境中存在的环境噪声在一定程度上也会影响策略性能。在梯度更新时，策略梯度方法很难确定每步的更新步长，步长太小容易使算法陷入局部最优且收敛速度慢，步长过大会导致最终找不到最优策略。信赖域策略优化算法（TRPO）[14]通过引入 KL 散度定义的信赖域约束强制限定新旧策略之间的差异，选取合适的步长，避免因步长偏大或偏小导致的问题。然而，TRPO 算法[14]将 KL 约束独立出来的做法会导致计算过程复杂度提高。OpenAI 对 TRPO 算法的目标函数进行改进，提出了近端策略优化（PPO）算法[25]，该算法直接使用上下界常量对策略更新幅度进行裁剪，降低计算复杂度。此外，PPO 算法[25]还可以在一次采样后多次更新策略参数，从而提高样本利用率。

控制变量法多用于蒙特卡洛模拟[22, 28]和金融[29]等领域，旨在减少蒙特卡洛方法中的梯度估计方差且不会引入偏差。大量相关研究利用基线构造控制变量，理论上最佳标量值状态相关基线是策略函数梯度的平方范数加权的 Q 值[23-24, 30]，用来评估动作的价值。依赖于状态的基线函数易于实现且被证实非常有效，然而由此产生的策略梯度仍然可能具有高方差尤其是在高维环境中。动作相关基线[31-34]可以更好地与原始策略梯度估计器相关联，通过使用更精细的控制变量可以进一步减少由梯度估计中动作的随机性而导致的方差。最近，有研究者通过利用时间结构将动作相关基线扩展为轨迹相关基线进一步减小了方差[35]。

综上所述，以上相关工作对于控制变量的研究主要集中在标量值基线函数上。本文以高维空间为背景，从控制变量对梯度估计方差的影响的角度出发，将标量基线函数扩展到向量空间中，进一步探索减小

策略梯度估计的方差，稳定策略更新。以 PGPE 算法为基础，推导了其对应的最优向量基线表示并且在理论上证明了引入最优向量基线的 PGPE 算法可以得到更小的策略梯度估计的方差。通过实验验证本文所提出的基于最优向量基线的 PGPE 算法与传统的最优标量基线相比，可以进一步减小梯度估计的方差，其梯度更新更加稳定。

1　强化学习建模

强化学习任务通常可以描述成马尔可夫决策过程，用 (S,A,P_T,P_1,r,γ) 表示，其中：S 是环境状态集合，A 是智能体可执行的动作集合，$P_T(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$ 是采取动作 $\boldsymbol{a}$ 时从当前状态 $\boldsymbol{s}$ 到下一状态 $\boldsymbol{s}'$ 的状态转移概率密度，$P_1(\boldsymbol{s})$ 是初始状态的概率，$r(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$ 是通过采取动作 $\boldsymbol{a}$ 从 $\boldsymbol{s}$ 过渡到 $\boldsymbol{s}'$ 的即时奖励，$0<\gamma<1$ 是未来奖励的折扣因子。令 $p(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{\theta})$ 表示为带有参数 $\boldsymbol{\theta}$ 的随机策略，其代表在给定状态 $\boldsymbol{s}$ 下采取动作 $\boldsymbol{a}$ 的条件概率密度。

假设 $h=(\boldsymbol{s}_1,\boldsymbol{a}_1,\boldsymbol{s}_2,\boldsymbol{a}_2,\cdots,\boldsymbol{s}_T,\boldsymbol{a}_T)$ 是长度为 T 的路径，路径 h 的累积回报定义为

$$R(h)=\sum_{t=1}^{T}\gamma^{t-1}r(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) \tag{1}$$

目标函数即累积回报的期望可以表示为关于参数 $\boldsymbol{\theta}$ 的函数，即

$$J(\boldsymbol{\theta})=\int p(h|\boldsymbol{\theta})R(h)\mathrm{d}h \tag{2}$$

其中，$p(h|\boldsymbol{\theta})=p(\boldsymbol{s}_1)\prod_{t=1}^{T}p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)p(\boldsymbol{a}_t|\boldsymbol{s}_t,\boldsymbol{\theta})$ 表示在给定模型参数 $\boldsymbol{\theta}$ 的条件下，每条路径 h 出现的概率。

强化学习的目标是找到最优策略参数 $\boldsymbol{\theta}^*$，从而最大化期望回报 $J(\boldsymbol{\theta})$，即

$$\boldsymbol{\theta}^*:=\arg\max_{\boldsymbol{\theta}}J(\boldsymbol{\theta}) \tag{3}$$

策略梯度方法使用梯度下降法更新参数 $\boldsymbol{\theta}$。传统策略梯度算法的策略梯度估计方差大的根本原因在于策略的随机性，它在每个时间步上都要随机采取一个动作，使得策略梯度估计的方差很大[16]。

2　PGPE及其最优标量基线

PGPE 算法的提出有效地解决了策略梯度估计方差大这一问题。PGPE 算法摒弃了策略中不必要的随机性，采用确定性策略：$\boldsymbol{a}=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{s})$，其中 $\boldsymbol{\varphi}(\boldsymbol{s})$ 是基函数向量。PGPE 算法的随机性来自策略参数，策略参数 $\boldsymbol{\theta}$ 采用先验高斯分布，其由超参数 $\boldsymbol{\rho}$ 控制：$\boldsymbol{\rho}=(\boldsymbol{\eta},\boldsymbol{\tau})$，其中 $\boldsymbol{\eta}$ 为均值向量，$\boldsymbol{\tau}$ 为标准差向量。$\boldsymbol{\theta}$ 的每一维度的分布表示为

$$p(\theta_i|\rho_i)=\frac{1}{\tau_i\sqrt{2\pi}}\exp\left(-\frac{(\theta_i-\eta_i)^2}{2\tau_i^2}\right) \tag{4}$$

由此可见，在 PGPE 算法中，不考虑环境中状态转移带来的随机扰动下，每条路径 h 的产生仅由一个采样的策略参数 $\boldsymbol{\theta}$ 所决定。在 PGPE 框架下，基于超参数 $\boldsymbol{\rho}$ 的目标函数 $J(\boldsymbol{\rho})$ 定义为

$$J(\boldsymbol{\rho})=\iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})R(h)\mathrm{d}h\mathrm{d}\boldsymbol{\theta} \tag{5}$$

通过寻找最优超参数 $\boldsymbol{\rho}^*$，从而最大化目标函数，即

$$\boldsymbol{\rho}^*:=\arg\max_{\boldsymbol{\rho}}J(\boldsymbol{\rho}) \tag{6}$$

PGPE 算法通过梯度下降法更新超参数 $\boldsymbol{\rho}$，其梯度表示为

$$\nabla_{\boldsymbol{\rho}}J(\boldsymbol{\rho})=p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})\nabla_{\boldsymbol{\rho}}\log \iint p(\boldsymbol{\theta}|\boldsymbol{\rho})R(h)\mathrm{d}h\mathrm{d}\boldsymbol{\theta} \tag{7}$$

由于 $p(h|\boldsymbol{\theta})$ 未知，通过收集样本，利用经验平均值估计上述策略梯度。样本收集过程如下：首先根据策略参数的分布 $p(\boldsymbol{\theta}|\boldsymbol{\rho})$ 采样 N 个策略参数 $\{\theta_n\}_{n=1}^N$，然后利用策略参数生成对应的 N 条路径样本 $\{h_n\}_{n=1}^N$，将每次收集的样本记为 $\{(\theta_n,h_n)\}_{n=1}^N$。PGPE 算法中策略梯度的经验估计为

$$\nabla_{\boldsymbol{\rho}}\hat{J}(\boldsymbol{\rho})=\frac{1}{N}\sum_{n=1}^{N}\nabla_{\boldsymbol{\rho}}\log p(\boldsymbol{\theta}^n|\boldsymbol{\rho})R(h^n) \tag{8}$$

引入基线后的 PGPE 算法的梯度估计表示为

$$\nabla_{\boldsymbol{\rho}}\hat{J}^b(\boldsymbol{\rho})=\frac{1}{N}\sum_{n=1}^{N}\nabla_{\boldsymbol{\rho}}\log p(\boldsymbol{\theta}^n|\boldsymbol{\rho})(R(h^n)-b) \tag{9}$$

那么，使 PGPE 算法的梯度估计方差最小化的最优基线可定义为

$$b^*=\arg\max_{b}\mathrm{Var}[\nabla_{\boldsymbol{\rho}}\hat{J}^b(\boldsymbol{\rho})] \tag{10}$$

对应的策略梯度估计器表示为

$$\boldsymbol{g}=\nabla_{\boldsymbol{\rho}}\log p(\boldsymbol{\theta}|\boldsymbol{\rho})(R(h)-b) \tag{11}$$

对于多维向量的方差，将其定义为协方差矩阵的迹，即矩阵中主对角线上所有元素之和，用 tr 表示，那么 d 维策略梯度估计器 $\boldsymbol{g}$ 的方差为

$$\mathrm{V}[\boldsymbol{g}]=\mathrm{tr}\,\mathrm{Var}[\boldsymbol{g}]=\sum_{j=1}^{d}\mathrm{Var}[g_j]=\sum_{j=1}^{d}E[g_j^2]-E[g_j]^2 \tag{12}$$

因此，通过最小化 V[$\boldsymbol{g}$]，可以得到 PGPE 算法的最优标量基线[24]为

$$b^{*}=\frac{E\left[R(h)\left\|\nabla_{\boldsymbol{\rho}}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right\|^{2}\right]}{E\left[\left\|\nabla_{\boldsymbol{\rho}}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right\|^{2}\right]} \tag{13}$$

3　本文算法

传统最优基线都是标量基线，没有考虑到梯度向量每个维度之间的差异，这实质上是给梯度向量的每个维度分配了一个相同的控制变量。如果将基线函数空间扩展为向量值函数，为梯度向量的每一个维度分配一个单独的基线，梯度估计的方差可以得到进一步的减小。

命题 1　给定一个实值函数类 $F:S\mapsto\mathbb{R}$，其中，$b\in F$，$\boldsymbol{c}\in F^{d}$，c_j 为 $\boldsymbol{c}$ 的第 j 维表示，最优标量基线表示为

$$b^{*}=\arg\min_{b\in F}\mathrm{V}[\boldsymbol{g}^{b}] \tag{14}$$

最优向量基线表示为

$$\boldsymbol{c}^{*}=\arg\min_{\boldsymbol{c}\in F^{d}}\mathrm{V}[\boldsymbol{g}^{\boldsymbol{c}}] \tag{15}$$

则具有最优向量基线、最优标量基线的策略梯度估计量的方差满足关系

$$\mathrm{V}[\boldsymbol{g}^{\boldsymbol{c}^{*}}]\leqslant\mathrm{V}[\boldsymbol{g}^{b^{*}}] \tag{16}$$

证明：c_j 和 b 属于相同的实值函数类，$\boldsymbol{c}'=(b^{*},b^{*},\cdots,b^{*})$ 可以看作 $\boldsymbol{c}\in F^{d}$ 的一种特殊情况，可以得到 $\mathrm{V}[\boldsymbol{g}^{\boldsymbol{c}^{*}}]\leqslant\mathrm{V}[\boldsymbol{g}^{\boldsymbol{c}'}]=\mathrm{V}[\boldsymbol{g}^{b^{*}}]$。

基于上述命题[36]，现将 PGPE 算法的基线函数空间扩展为向量值函数，并给出 PGPE 算法的最优向量基线。

定理 1　设最优向量值基线函数表示为

$$\boldsymbol{c}^{*}=\left(c_1^{*}(\boldsymbol{\rho}),c_2^{*}(\boldsymbol{\rho}),\cdots,c_d^{*}(\boldsymbol{\rho})\right) \tag{17}$$

其中 d 为梯度的维度。PGPE 算法的策略梯度估计第 j 维对应的最优基线表示为

$$c_j^{*}(\boldsymbol{\rho})=\frac{E_{h\sim p(h\mid\boldsymbol{\theta}),\boldsymbol{\theta}\sim p(\boldsymbol{\theta}\mid\boldsymbol{\rho})}\left[\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}R(h)\right]}{E_{h\sim p(h\mid\boldsymbol{\theta}),\boldsymbol{\theta}\sim p(\boldsymbol{\theta}\mid\boldsymbol{\rho})}\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}} \tag{18}$$

证明：设向量基线表示为

$$\boldsymbol{c}(\boldsymbol{\rho})=\left(c_1(\boldsymbol{\rho}),c_2(\boldsymbol{\rho}),\cdots,c_d(\boldsymbol{\rho})\right) \tag{19}$$

则引入向量基线 $\boldsymbol{c}(\boldsymbol{\rho})$ 的 PGPE 算法的策略梯度估计表示为

$$\boldsymbol{g}^{\boldsymbol{c}}=(g_1^{c_1},g_2^{c_2},\ldots,g_d^{c_d}) \tag{20}$$

其中：$g_j^{c_j}=\dfrac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\left(R(h)-c_j(\boldsymbol{\rho})\right)$，$j=1,\cdots,d$。

以上为梯度向量的每一维度分配单独的基线函数 $c_j(\boldsymbol{\rho})$，即为整个梯度分配向量基线 $\boldsymbol{c}(\boldsymbol{\rho})$，对应的梯度向量第 j 维的方差为

$$\mathrm{Var}\left[g_j^{c_j}\right]=E\left[\left(g_j^{c_j}\right)^{2}\right]-E\left[g_j^{c_j}\right]^{2} \tag{21}$$

其中式(21)中只有第一项与 $c_j(\boldsymbol{\rho})$ 相关，而第二项与 $c_j(\boldsymbol{\rho})$ 无关，所以最小化 Var[$g_j^{c_j}$] 等同于最小化 $E\left[\left(g_j^{c_j}\right)^{2}\right]$，即

$$\arg\min_{c_j(\boldsymbol{\rho})}\mathrm{Var}[g_j^{c_j}]=\arg\min_{c_j(\boldsymbol{\rho})}E[(g_j^{c_j})^{2}] \tag{22}$$

$$E\left[\left(g_j^{c_j}\right)^{2}\right]=E\left[\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}\left(R(h)-c_j(\boldsymbol{\rho})\right)^{2}\right] \tag{23}$$

使方差最小的解 $c_j^{*}(\boldsymbol{\rho})$ 应满足如下条件：

$$\frac{\partial}{\partial c_j^{*}(\boldsymbol{\rho})}E[(g_j^{c_j})^{2}]=0 \tag{24}$$

$$\Rightarrow 2E\left[\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}\left(R(h)-c_j(\boldsymbol{\rho})\right)\right]=0$$

$$\Rightarrow c_j^{*}(\boldsymbol{\rho})=\frac{E_{h\sim p(h\mid\boldsymbol{\theta}),\boldsymbol{\theta}\sim p(\boldsymbol{\theta}\mid\boldsymbol{\rho})}\left[\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}R(h)\right]}{E_{h\sim p(h\mid\boldsymbol{\theta}),\boldsymbol{\theta}\sim p(\boldsymbol{\theta}\mid\boldsymbol{\rho})}\left(\frac{\partial}{\partial\boldsymbol{\rho}_j}\log p(\boldsymbol{\theta}\mid\boldsymbol{\rho})\right)^{2}}$$

由此得到 PGPE 算法的策略梯度估计第 j 维对应的最优基线表示。

4　实验结果与分析

为了验证本文方法的有效性，以 OpenAIGym 环境中的 Pendulum-v0 环境为算法验证任务，其示意图如图 1 所示。该环境包括一个钟摆，钟摆以随机位置开始，学习目标是令钟摆向上摆动，使其尽可能长时间保持直立。状态空间 S 是三维连续的，由钟摆的角 φ 的正弦值、余弦值及角速度 $\dot{\varphi}$ 组成。动作空间 A 是一维并且连续的，对应于控制钟摆转动的电机力矩。

本实验将从具体参数下梯度估计的方差和偏差、参数更新过程中方差的变化以及所学策略的性能探索本文算法有效性。具体对比算法：(1) PGPE：没有

任何基线的 PGPE 算法[15]；(2) PGPE-OB：基于最优标量基线的 PGPE 算法[24]；(3) PGPE-VOB：本文基于最优向量基线的 PGPE 算法。

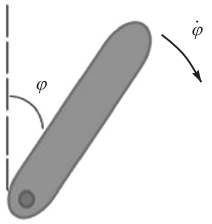

图 1　Pendulum-v0 任务示意图

Fig. 1　Schematic diagram of Pendulum-v0

4.1　方差和偏差

对算法 PGPE、PGPE-OB、PGPE-VOB 在具体参数下梯度估计的方差和偏差[37]进行对比。为保证实验的公平性，以上所有算法均采用相同的参数设置。高斯分布的初始均值均设置为 $\eta=0.3$，初始标准差均设置为 $\tau=0.5$，路径的最大长度设置为 $T=200$，奖励折扣因子 $\gamma=0.9$，路径样本的数量设置为 $N=250$。为了计算梯度估计的偏差，利用 600 条路径样本估计的梯度视为真实梯度。通过 80 次实验计算得到在参数 $\eta=0.3$ 及 $\tau=0.5$ 时梯度估计的方差和偏差结果见表 1。

表 1　3 种对比算法关于 $\rho=(0.3, 0.5)$ 的梯度估计的方差和偏差

Tab. 1　Variance and bias of gradient estimation of $\rho=(0.3, 0.5)$ for three comparison algorithms

算法	方差		偏差	
	$\eta=0.3$	$\tau=0.5$	$\eta=0.3$	$\tau=0.5$
PGPE	3.573 ± 0.272	6.825 ± 0.235	−0.159 ± 0.064	−0.364 ± 0.056
PGPE-OB	0.591 ± 0.126	1.018 ± 0.103	0.076 ± 0.038	0.128 ± 0.031
PGPE-VOB	0.485 ± 0.094	0.874 ± 0.106	−0.059 ± 0.028	0.103 ± 0.034

实验结果表明，PGPE-VOB 算法关于均值 η 和标准差 τ 的梯度估计方差均小于 PGPE-OB 算法和原始的 PGPE 算法，且引入向量基线不会增加偏差。

4.2　参数更新过程中的方差

在此实验中，路径的最大长度设置为 $T=100$，路径样本的数量设置为 $N=10$，参数的迭代次数为 50 次。若标准差参数 τ 在策略更新过程中变为负值，则将其设置为 0.05。通过 100 次运行得到的策略梯度值进行方差（对数标度）的计算，重复上述实验 10 次，观察 10 次实验中关于均值参数 η 的梯度估计的方差平均值，结果如图 2 所示。

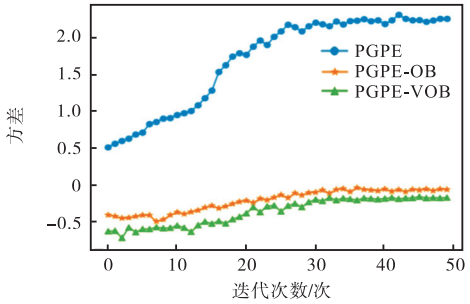

图 2　参数更新过程中的关于参数 η 的策略梯度估计的方差

Fig. 2　Variance of policy gradient estimation with respect to parameter η during parameter update

由图 2 可知，引入向量基线有效减小了参数更新过程中梯度估计的方差。因此，基于向量基线的 PGPE 算法在参数更新稳定性方面比其他算法更有优势。

4.3　策略性能

为保证公平性，本次实验中的初始均值和方差均设为一致，策略参数在高斯分布中随机选取，每次实验迭代次数为 600 次，每次迭代采样 250 条路径样本进行策略梯度的计算更新，计算 10 次实验的平均累积回报，实验结果如图 3 所示。

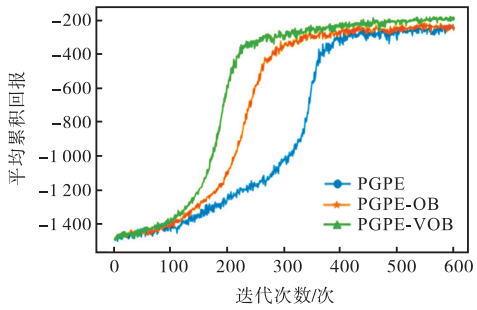

图 3　策略参数迭代过程中的平均累积回报

Fig. 3　Average cumulative return over the iteration of the policy parameters

由图 3 可知：PGPE-VOB 算法在第 200 次迭代后就开始收敛并取得了较好的效果，而 PGPE-OB 算法在大约 300 次迭代后开始收敛，且所得平均累积回报总体低于 PGPE-VOB 算法。原始 PGPE 算法在迭代第 400 次后开始收敛，所得累积回报总体上低于以上两种方法。实验结果表明 PGPE-VOB 算法比原始

PGPE 算法和 PGPE-OB 算法具有更好的性能，收敛速度更快。

综上所述，实验表明最优向量基线的引入进一步减小了策略梯度估计的方差，策略梯度的更新更加稳定，收敛速度更快且具有更好的性能。

5 结　语

策略梯度估计方差大是策略梯度算法领域中的共性问题，本文以减小策略梯度估计方差为研究目标，提出了最优向量基线概念并通过理论推导得到了 PGPE 算法的最优向量基线表示。通过实验证明，基于最优向量基线的 PGPE 算法具有更小的策略梯度估计方差并取得了最优性能。在未来研究中会将最优向量基线应用在高维复杂任务中进一步验证其有效性。

参考文献：

[1] SUTTON R S，BARTO A G. Reinforcement learning：an introduction[M]. 2nd ed. Cambridge：MIT Press，1998.

[2] RAJESWARAN A，MORDATCH I，KUMAR V. A game theoretic framework for model based reinforcement learning[EB/OL]. [2022-10-30]. http://arxiv.org/abs/2004.07804.

[3] 李茹杨，彭慧民，李仁刚，等. 强化学习算法与应用综述[J]. 计算机系统应用，2020，29(12)：13-25.

[4] IBARZ J，TAN J，FINN C，et al. How to train your robot with deep reinforcement learning：lessons we have learned[J]. The international journal of robotics research，2021，40(4/5)：698-721.

[5] 万里鹏，兰旭光，张翰博，等. 深度强化学习理论及其应用综述[J]. 模式识别与人工智能，2019，32(1)：67-81.

[6] 刘全，翟建伟，章宗长，等. 深度强化学习综述[J]. 计算机学报，2018，41(1)：1-27.

[7] 刘建伟，高峰，罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报，2019，42(6)：1406-1438.

[8] TEAZURO G. TD-Gammon，a self-teaching backgammon program，achieves master-level play[J]. Neural computation，1994，6(2)：215-219.

[9] NG A Y，JORDAN M I. PEGASUS：a policy search method for large MDPs and POMDPs[EB/OL]. [2022-10-30]. http://arxiv.org/abs/1301.3878.

[10] KONDA V，TSITSIKLIS J. Actor-critic algorithms[J]. Advances in neural information processing systems，1999，12：1008-1014.

[11] RONALD J，WILLIAMS. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning，1992，8(3)：229-256.

[12] KAKADE S M. A natural policy gradient[J]. Advances in neural information processing systems，2001，14：1531-1538.

[13] LILLICRAP T P，HUNT J J，PRITZEL A，et al. Continuous control with deep reinforcement learning[EB/OL]. [2022-10-30]. http://arxiv.org/abs/1509.02971v1.

[14] SCHULMAN J，LEVINE S，MORITZ P，et al. Trust region policy optimization[J]. Computer science，2015，6(4)：1889-1897.

[15] SEHNKE F，OSENDORFER C，RÜCKSTIEß T，et al. Parameter-exploring policy gradients[J]. Neural networks，2010，23(4)：551-559.

[16] PETERS J，SCHAAL S. Policy gradient methods for robotics[C]//IEEE. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. New York：IEEE，2006：2219-2225.

[17] GHADIMI S，LAN G，ZHANG H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization[J]. Mathematical programming，2016，155(1)：267-305.

[18] THOMAS P. Bias in natural actor-critic algorithms[C]//PMLR. International Conference on Machine Learning. New York：PMLR，2014：441-448.

[19] SILVER D，LEVER G，HEESS N，et al. Deterministic policy gradient algorithms[C]//PMLR. International Conference on Machine Learning. New York：PMLR，2014：387-395.

[20] SCHULMAN J，MORITZ P，LEVINE S，et al. High-dimensional continuous control using generalized advantage estimation[EB/OL]. [2022-10-30]. https://arxiv.org/pdf/1506.02438.pdf.

[21] TESAURO G. Temporal difference learning and TD-Gammon[J]. Communications of the ACM，1995，38(3)：58-68.

[22] RUBINSTEIN R Y，MARCUS R. Efficiency of multivariate control variates in monte Carlo simulation[J]. Operations research，1985，33(3)：661-677.

[23] GREENSMITH E，BARTLETT P L，BAXTER J. Vari-

ance reduction techniques for gradient estimates in reinforcement learning[J]. Journal of machine learning research, 2004, 5(9): 1471–1530.

[24] ZHAO T, HACHIYA H, NIU G, et al. Analysis and improvement of policy gradient estimation[J]. Neural networks, 2012, 26: 118–129.

[25] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. [2022–10–30]. http://arxiv.org/pdf/1707.06347.

[26] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]//PMLR. International Conference on Machine Learning. New York: PMLR, 2016: 1928–1937.

[27] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with deep reinforcement learning[EB/OL]. [2022–12–23]. http://arxiv.org/abs/1312.5602v1.

[28] GLYNN P W, SZECHTMAN R. Some new perspectives on the method of control variates[M]//NIEDDERREITER H, SHIUE P J. Monte Carlo and Quasi-Monte Carlo methods in scientific computing. Berlin: Springer, 2002: 27–49.

[29] GLASSERMAN P. Monte Carlo methods in financial engineering[M]. Berlin: Springer, 2004.

[30] WEAVER L, TAO N. The optimal reward baseline for gradient-based reinforcement learning[EB/OL]. [2022–10–30]. http://arxiv.org/abs/1301.2315.

[31] GU S, LILLICRAP T, GHAHRAMANI Z, et al. Q-prop: sample-efficient policy gradient with an off-policy critic[EB/OL]. [2022–10–30]. http://arxiv.org/pdf/1611.02247.

[32] LIU H, FENG Y, MAO Y, et al. Action-dependent control variates for policy optimization via stein's identity[EB/OL]. [2022–10–30]. http://arxiv.org/abs/1710.11198v4.

[33] GRATHWOHL W, CHO D, WU Y, et al. Backpropagation through the void: optimizing control variates for black-box gradient estimation[EB/OL]. [2022–10–30]. http://arxiv.org/abs/1711.00123v3.

[34] WU C, RAJESWARAN A, DUAN Y, et al. Variance reduction for policy gradient with action-dependent factorized baselines[EB/OL]. [2022–10–30]. http://arxiv.org/abs/1803.07246v1.

[35] CHENG C A, YAN X, BOOTS B. Trajectory-wise control variates for variance reduction in policy gradient methods[EB/OL]. [2022–10–30]. http://arxiv.org/abs/1908.03263v1.

[36] ZHONG Y, ZHOU Y, PENG J. Coordinate-wise control variates for deep policy gradients[EB/OL]. [2022–10–30]. https://arxiv.org/abs/2107.04987v1.

[37] ZHAO T, HACHIYA H, TANGKARATT V, et al. Efficient sample reuse in policy gradients with parameter-based exploration[J]. Neural computation, 2013, 25(6): 1512–1547.

责任编辑：郎婧