Loaded DiCE: Trading off Bias and Variance in Any-Order Score Function Estimators for Reinforcement Learning

针对强化学习的任意阶 **得分函数估计器(Score Function Estimators)** 中的偏差方差权衡

作者:JohnJim

> **Loaded DiCE**: 即本文提出的objective或者说目标函数，也就是本文第3节中兼容了优势估计器后的$J_{\diamond}$以及加入了折扣因子后最终的Loaded DiCE(3.2节中的J_{λ})

> **优势估计器**: 见2.3节，也是一种梯度估计器

> **Score Function Estimators** 是一种梯度估计方法，有人将梯度估计大致分为两类，一类是求解分布测度的导数，典型如 score function gradient estimator(如强化学习policy gradient中一个经典的算法REINFORCE)，一类是求解代价函数的导数，如pathwise gradient estimator

> **为什么要梯度估计?** 机器学习大多都是求最值的过程，中学时我们知道导数为0的地方有极值，扩展到大学就是梯度，即极值所在的地方就是梯度下降最快的地方。但是对于目标函数是如下形式的随机函数时，即常规求导方法无法求出梯度，就需要进行梯度估计，至于以下函数具体啥意思暂不深究，参考知乎 $$ F(\theta)=\int p(x ; \theta) f(x ; \phi)=E_{p(x ; \theta)}[f(x ; \phi)] $$ **什么是代价函数?** 机器学习我们希望最小化的函数叫目标函数，当我们最小化的时候又叫代价函数或损失函数或误差函数，参考CSDN

摘要

在具有未知或不可追踪的动态随机环境中优化目标所用的基于梯度的方法，需要导数估计器。本文推出一个目标(objective)，即在自动微分下，可以以任意阶生成低方差的无偏导数估计量。我们的目标与任意优势估计器(advantage estimators)兼容，该目标可在使用函数逼近时控制任意阶导数的偏差和方差。此外，本文提出了一种通过折扣更远的因果关系的影响(discounting the impact of more distant causal dependencies)来权衡高阶导数偏差与方差的方法。 并且展示了本文的目标在分析上易于控制的MDP和用于连续控制的元强化学习中的正确性和实用性。

> 这里的objective指的应该是目标函数的意思

> 总结本文的创新点，一是提出了新的目标函数也就是这里的objective，能够进行任意阶低方差无偏导数估计，其中2.3节引入优势估计器中的τ用来控制方差与偏差，下文也会提本文提出的估计器，实际上建立目标函数然后基于此进行梯度估计是一回事，二是引入一个类似于经典RL的折扣因子即3.2节中的λ，来discounting the impact of more distant causal dependencies

> 这里的discounting the impact of more distant causal dependencies跟经典强化学习类似，就是对过去行为的贡献作一个衰减或者说折扣

1 引言

在随机环境(stochastic settings)中，比如强化学习，通常不可能计算目标(objectice)的微分，因为它们的分布不可知或难处理(比如强化学习中的转换函数transition function)。在这种情况下，基于梯度的优化只可能通过随机梯度估计器。

> 这里也提到了为什么要梯度估计?

[François-Lavet et al., 2018]等人通过建立适合自动微分的一阶微分估计器取得了巨大成功，并用于优化DNN参数。

但是在很多应用方面一阶导数是不够的，比如元学习和多智能体学习，它们常涉及通过基于梯度的学习器来微分[Finn et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Foerster et al., 2018a]。高阶方法也可以提高采样效率 [Furmston et al., 2016]，但是在自动微分的情况下以地方查正确估计这些高阶导数很困难。

[Foerster et al.,2018b]提出了一个易于使用的构建任意阶导数估计器的工具，因为它们避免了繁琐的操作，而这些操作又需要考虑梯度估计值对它们的采样分布的依赖性。但是，它们的表述公式依赖于目标的纯蒙特卡洛估计，在一阶和高阶导数的估计中引入了不可接受的方差，比较理想化，这就限制了依赖于这些导数的方法的使用。

与此同时，随机目标的一阶微分估计器也有长足的进展。在强化学习中，值函数(value functions)用作critics和baseline已有广泛研究。在混合目标中，可以将梯度估计量的偏差和方差之间的权衡明确化，该目标将目标的蒙特卡洛样本与值函数结合在一起[Schulman et al., 2015b]。这些技术构成优势估计器家族，可用于减少方差并加速一阶优化中的信用分配(credit assignment)，但尚未完全通用地应用于高阶导数。

> “François-Lavet et al., 2018]等人...高阶导数"这段内容在介绍目前高阶梯度估计还不成熟

在这项工作中，我们得出了一个目标，该目标可以进行任意次微分，以产生具有Markov属性的随机计算图(SCG)中的高阶导数的正确估计量，例如RL和序列建模中的那些。 与先前的工作不同，该目标与优势估计器的任意选择完全兼容。 当使用近似值函数时，这允许使用已知技术（或使用未来的为一阶导数设计的任何优势估计方法）在任意阶导数估计中的偏差和方差之间进行明确的权衡。 此外，我们提出了一种通过折衷更远的因果关系的影响来权衡高阶导数的偏差和方差的方法。

> 计算图是一种数学表达形式，参见百易教程和机器之心

根据经验，我们首先使用小的随机MDP来接受解析解，以展现本文的估计器在使用理想值函数时是无偏且方差低的，并且偏差和方差可以使用两个超参数灵活地进行权衡。 我们进一步研究了在更具挑战性的元强化学习问题中进行模拟连续控制的目标，并展示了各种参数选择对训练的影响。

2 背景

2.1 梯度估计器

我们通常面对的目标形式是随机变量的期望。为了计算出带有感兴趣参数梯度的期望，必须采用梯度估计器，因为梯度不能被精确计算出。比如强化学习环境动态(environment dynamics)是未知的并且组成目标(objective)即期望回报的一部分。如下： $$ \nabla_{\theta} \mathbb{E}*{x}[f(x, \theta)]=\mathbb{E}*{x}\left[f(x, \theta) \nabla_{\theta} \log p(x ; \theta)+\nabla_{\theta} f(x, \theta)\right] $$

> 这是一个经典的蒙特卡洛梯度估计公式

等式右边通常由蒙特卡洛样本估计出，通常f独立于θ，第二项省略，如果f依赖于θ，那么第一项可能就省去，[Fu, 2006]或[Mohamed et al.,2019]有更详尽的阐述。

2.2 随机计算图和MDP

> 即通过随机计算图表示MDP过程

随机计算图(SCG)是有向无环图，其中节点是确定性函数或随机函数，边表示函数依赖性[Schulman et al。，2015a]。 上面描述的梯度估计器可以用于估计目标相对于参数θ的梯度(**成本节点**之和)。[Schulman et al, 2015a]提出了*替代损失*，这是一个在微分下产生所需梯度估计的单一目标。

[Weber et al.,2019]在SCG上应用了更为高级的一阶梯度估计器。 他们将SCG的Markov属性公式化，最开始应用于强化学习中。 本文将在后面小节中描述这些估计量，首先定义SCG的相关子集。 为了使本文的主体保持简单并突出本文方法中最重要的已知用例，我们采用强化学习的表示法，而不是通用SCG的较繁琐的表示法。

强化学习中的图描述了马尔可夫决策过程(MDP)。首先给定一个$t=0$时刻的初始状态s_0。在每个时间步(timestep)中，动作(action)a_t由随机策略π_{θ}取样给出，参数为θ，即将状态映射到动作。这就会添加一个随机节点a_t到图中。状态-动作对(state-action pair)会导致一个奖励r_t，和下一个状态s_{t+1}，这会使过程持续下去。如图Fig1是一个简单的MDP图，在许多问题中，奖励只取决于状态而不是状态和动作。考虑一个经T时间步终止的回合问题(episodic problems)，尽管结果可能会扩展到无终止的情况。(衰减后的)奖励是这个图的成本节点，构成奖励之和的强化学习目标：$J=\mathbb{E}\left[\sum_{t=0}^{T} \gamma^{t} r_{t}\right]$, 其中$\gamma$是衰减系数，该期望跟策略和未知的转换动态(transition dynamics)都有关。

本文结果的泛化也适用于更通用SCG，其目标仍然是随着时间的推移的总回报。在每个时间步t内，会有任意数量的随机和确定节点$\mathcal{X}*{t}$。然而，这些节点可能通过影响下一个时间步的回报。马尔科夫属性表明对任意节点w，都存在一个从w到任何$r*{t^{\prime}}$的定向路径，其中$t^{\prime}>t$且不会被$\mathcal{X}*{t}$阻挡，即w的子节点都不会在$\mathcal{X}*{t}$中([Weber et al., 2019]中的定义6)。此类SCG可以捕获各种类似于MDP的模型，如图Fig1所示。

2.3 优势梯度估计器

SCG中一组节点的值函数是目标对其他随机变量（不包括那组节点）的期望。 这些可以通过充当控制变量（“基准”）或作为评论家(critics)来减少差异，这些评论家还以相应的随机节点所采样的值（即采样的动作）为条件。 评论者值函数和基准值函数之差被称为优势，它取代了梯度估计器中的采样成本。

> **关于基准(baseline)。** 简单例子就是因为score function(前面提到的得分估计器)的期望为0，这个具体证明略，一种降低估计方差的思路就是将代价函数$f(x)$改为$f(x)-b$,这个b就是baseline，这里基准值函数就指的是无方差的$f(x)-b$，当然形式要更复杂

> 这里的评论家值函数(critic value function)可以参考Qlearning的值函数，它跟Actor-Critic中的Critic部分是类似的，所以这里称作critic value function，可以把这里的值函数理解为常规意义的目标函数

基线值函数仅影响梯度估计量的方差[Weaver and Tao，2001]。 但是，使用完善的评论家价值函数会导致梯度估计量出现偏差。我们可以通过使用抽样成本（无偏，高方差）和评论家价值函数（有偏，低方差）的不同组合权衡偏见和方差。 优势估计量及其超参数的这种选择可用于调整所得梯度估计量的偏差和方差，以适应当前的问题。

> 此处也提到baseline影响方差，但是值函数本身会产生偏差，因此才有了方差偏差权衡问题

有许多方式可以对RL中的优势函数建模，如下是[Schulman et al.，2015b]提出的一个简单流行的优势估计器：$$ A^{G A E(\gamma, \tau)}\left(s_{t}, a_{t}\right)=\sum_{t^{\prime}=t}^{\infty}(\gamma \tau)^{t^{\prime}-t}\left(r_{t^{\prime}}+\gamma \hat{V}\left(s_{t^{\prime}+1}\right)-\hat{V}\left(s_{t^{\prime}}\right)\right) $$ 其中参

数τ权衡方差与偏差：当τ等于1时，公式只有采样回报且无偏，但是高方差。当$\tau=0$，公式只有下一个采样回报r_t且重度依赖于估计的值函数$\hat{V}$，该情况下减少了方差但牺牲了偏差。

> 此处τ等于1时，说明优势即值函数和基准值函数存在差值，所以有方差，等于0时，优势为0说明值函数跟基准值函数误差，这样就完全依赖于对值函数本身的估计，这个估计过程就会产生偏差。

2.4 高阶估计器

> 本节主要提到前人的一些方法比如J_{LVC}和DiCE即$J_{\odot}$

为了构建高阶梯度估计器，再次应用上述技巧，将梯度估计视作一个新SCG的目标。[Foerster et al.，2018b]等人指出了[Schulman et al.,2015a]的替代损失方法应用于高阶微分的几个缺点。替代损失本身不能再次加以微分以产生正确的高阶估计量。甚至使用替代损失产生的估计值也不能被视为新SCG中的目标，因为替代损失将采样成本与采样分布的依赖关系分开了。

为了处理这个问题，[Foerster et al.，2018b]引入了DiCE，即可以反复微分（使用自动微分）以生成任意阶导数的无偏估计量的单个目标。对于强化学习，DiCE目标如下： $$ J_{\odot}=\sum_{t=0}^{T} \gamma^{t} {\odot}(a \leq t) r_{t} $$ 其中$a \leq t$指在时间步t或之前的随机节点的集合。$\odot$是一个作用于随机节点集合$\mathcal{W}$特殊的操作算子。即$\odot(.)$总是估计为1，但是有在微分下有一个特殊的操作如下： $$ \nabla_{\theta} \odot(\mathcal{W})=\odot(\mathcal{W}) \sum_{w \in \mathcal{W}} \nabla_{\theta} \log p(w ; \theta) $$ 该算子使针对期望微分的似然比技巧自动化，同时保持依赖性即使在计算高阶导数时也能应用相同技巧。为了方便起见，再拓展空集上的操作：$\odot(\varnothing)=1$，即有零阶导数。

与上文描述的用于估计随机目标的一阶导数的现有技术方法相比，DiCE的原始版本有两个关键缺陷。第一，首先，它没有使用基线来减少高阶导数估计量方差的机制。[Mao et al., 2019]，以及[Liu et al.,2019]（随后但独立）提出了针对该问题的相同局部解决方案，但均未提供超出二阶的估计量的无偏证明。第二，DiCE（以及[Mao et al., 2019]和[Liu et al.,2019]的估计器）需要使用蒙特卡洛抽样成本。但不能使用评论家价值函数(critic value functions)的形式，就无法利用所有可能的优势估算器。

在精确计算高阶导数估计量时，给定奖励对所有先前动作的依赖性会导致在先前时间步上的嵌套总和。 如[Furmston et al., 2016]和[Rothfuss et al., 2018]所指出的，当使用数据进行估算时，这些项往往具有较高的方差，并且在局部最优值附近变小。他们使用此观察结果提出了DiCE目标的简化版本，删除了这些依赖项，如下： $$ J_{L V C}=\sum_{t=0}^{T}\odot\left(a_{t}) R_{t}\right. $$ 该估计器比一阶微分估计器偏差要高，并且[Rothfuss et al., 2018]并没有推出一个正确的对于任意阶的无偏估计器，没有在这个目标中使用优势估计器，也没有拓展MAML式元学习的应用[Finn et al., 2017]。

在下一节中，我们将介绍一个新的目标，该目标可以利用评论家(critics)以及基线值函数，从而可以通过选择优势估计量来权衡任意阶导数的偏差和方差。 此外，我们引入了对过去依赖性的衰减，从而可以平滑地权衡偏差和[Furmston et al., 2016]产生的高方差。

3 方法

> 本节主要讲了本文提出的Loaded DiCE，是建立在2.4节中的DiCE上的，首先重写了一下DiCE即$J_{\odot}$，然后用优势A替代Q，具体推导也可以见原文的附录，这是本文的一个贡献。

DiCE目标是奖励的总和，为了使用评论家值函数(critic value functions),必须对回报使用先验总和(forward-looking sums)。如果图相对其目标保持上文第2.2节中定义的Markov属性，那么就可能对成本节点进行顺序分解，比如回报r_t和动作a_t。我们首先给定目标是所有奖励的衰减和(discounted sum)，但实际目标则是策略π_{θ}下奖励的期望衰减和。

如经典的RL一样，定义回报$R_{t}=\sum_{t^{\prime}=t}^{T} \gamma^{t^{\prime}-t} r_{t^{\prime}}$，其中$r_{t}=R_{t}-\gamma R_{t+1}$，因此有： $$ \begin{aligned} J_{\odot} &=\sum_{t=0}^{T} \gamma^{t}{\odot}\left(a_{\leq t}\right)\left(R_{t}-\gamma R_{t+1}\right) \ &=\sum_{t=0}^{T} \gamma^{t}{\odot}\left(a_{\leq t}\right) R_{t}-\sum_{t=0}^{T} \gamma^{t+1}{\odot}\left(a_{\leq t}\right) R_{t+1} \end{aligned} $$ 令等式第二项$t+1$换成t，就有： $$ \begin{aligned} J_{\odot}=& \sum_{t=0}^{T} \gamma^{t} {\odot}\left(a_{\leq t}\right) R_{t}-\sum_{t=1}^{T+1} \gamma^{t} \odot\left(a_{\leq t-1}\right) R_{t} \ =& \sum_{t=0}^{T} \gamma^{t}{\odot}\left(a_{\leq t}\right) R_{t}-\sum_{t=1}^{T+1} \gamma^{t}{\odot}\left(a_{<t}\right) R_{t} \ =& \sum_{t=0}^{T} \gamma^{t} {\odot}\left(a_{\leq t}\right) R_{t}-\sum_{t=1}^{T+1} \gamma^{t}{\odot}\left(a_{<t}\right) R_{t} \ =& \sum_{t=0}^{T} \gamma^{t} {\odot}\left(a_{<0}\right) R_{t}-\sum_{t=0}^{T} \gamma^{t} {\odot}\left(a_{<t}\right) R_{t} \ &+\gamma^{0}{\odot}\left(a_{<0}\right) R_{0}-\gamma^{T+1} {\odot}\left(a_{<T+1}\right) R_{T+1} \ =& R_{0}+\sum_{t=0}^{T} \gamma^{t}\left({\odot}\left(a_{\leq t}\right)-{\odot}\left(a_{<t}\right)\right) R_{t} \end{aligned} $$ 其中最后一项$\gamma^{T+1} {\odot}\left(a_{<T+1}\right) R_{T+1}$中$R_{T+1}=0$。 现在有了前瞻性回报的目标，它通过$({\odot}\left(a_{\leq t}\right)-{\odot}\left(a_{<t}\right)t$捕获了对抽样分布的依赖性。由于这只是DiCE目标的重新表达（适用于具有Markov属性的有限类SCG），仍然可以保证其导数将是真实目标的导数的无偏估计量，并且上升到任意阶 。 最初的DiCE目标的证明由[Foerster et al., 2018b]等人提供。因为R_0导数为0，接下来的估计其中将直接省略掉。 现在介绍值函数。R_t条件独立于，每个${\odot}(a_{\leq t})$和${\odot}(a_{\lt t})$(以及它们的导数)。由于SCG的马尔可夫属性，这等同于条件独立于给定的s_t和a_t。考虑本文目标或者说期望的新形式$J_{\odot}$，推出评论家值函数(critci value function)如下，其中定义$Q\left(s_{t}, a_{t}\right)=\mathbb{E}{\pi}\left[R{t} | s_{t}, a_{t}\right]$： $$ \begin{aligned} \mathbb{E}{\pi}\left[J{\odot}\right] &=\mathbb{E}{\pi}\left[\sum{t=0}^{T} \gamma^{t}\left({\odot}\left(a_{\leq t}\right)-{\odot}\left(a_{<t}\right)\right) \mathbb{E}\left[R_{t} | s_{t}, a_{t}\right]\right] \ &=\mathbb{E}{\pi}\left[\sum{t=0}^{T} \gamma^{t}\left({\odot}\left(a_{\leq t}\right)-{\odot}\left(a_{<t}\right)\right) Q\left(s_{t}, a_{t}\right)\right] \end{aligned} $$ 强化学习中，通常使用期望状态值$V\left(s_{t}\right)=\mathbb{E}{a{t}}\left[Q\left(s_{t}, a_{t}\right)\right]$表示最优基线的近似。那么估计器可以用$A\left(s_{t}, a_{t}\right)=Q\left(s_{t}, a_{t}\right)-V\left(s_{t}\right)$来代替上面的$R_t$，现在就推出了包含优势$A(s_{t}, a_{t})$在任意阶上的无偏估计器： $$ J_{\diamond}=\sum_{t} \gamma^{t}\left({\odot}\left(a_{\leq t}\right)-{\odot}\left(a_{<t}\right)\right) A\left(s_{t}, a_{t}\right) $$ 在实践中，通常会省略γ^{t}，从而优化未折扣(discount)的回报，但仍使用折扣回报作为减少方差的工具。 参见[Thomas, 2014]。

3.1 函数估计

在实践中，必须从有限数据中得到优势估计。评论家价值函数的不精确模型（由于数据有限，模型类规格不正确或学习效率低下）在梯度估计器中引入了偏差。 如[Schulman et al., 2015b],，我们可能会结合使用抽样成本和估计值来形成权衡偏差和方差的优势估计量。 但是，由于有了新的估计器，该估估器充分体现了优势对采样分布的依赖性，因此这些权衡可以立即应用于高阶导数。
近似基准值函数仅影响估计方差。 尽管如此，谨慎选择此基准可能仍具有重要意义（例如，通过利用策略的因式分解[Foerster et al., 2018c])。 本文对目标的表述将此类方法拓展一阶优势估计以及高阶导数。

3.2 高阶依赖导致的方差

> 引入折扣因子λ是本文的另一个贡献

本文提出一个折扣因子$\lambda \in[0,1]$来限制过去动作对高阶导数估计的影响，跟MDP中的折扣因子γ的作用类似。

首先对于任意随机节点集$\mathcal{W}$，算子${\odot}$将其分解为连乘的形式：${\odot}(\mathcal{W})=\prod_{w \in \mathcal{W}}{\odot}(w)$。通过对过去的贡献进行指数衰减完善折扣(discounting)如下： $$ \left.J_{\lambda}=\sum_{t=0}^{T}\left(\prod_{t^{\prime}=0}^{t}{\odot}\left(a_{t^{\prime}}\right)\right)

\lambda^{t-t^{\prime}}-\prod_{t^{\prime}=0}^{t-1} {\odot}\left(a_{t^{\prime}}\right) \lambda^{t-t^{\prime}}\right)\right) A_{t} $$ 这就是本文的最终目标(obejective)，即"loaded DiCE"。这些乘积最好在对数空间操作，这样能得到稳定方便的求和。下列算法1展示了如何计算该目标。

> **Algorithm 1** Computer Loaded DiCE Objective
> **Require:** trajectory of states s_t, actions $a_t,t=0...T$
> $J \leftarrow 0$ // J accumulates the final objective
> $w \leftarrow 0$ // w accumulates the λ-weighted stochastic dependencies
> **for** $t \leftarrow 0$ to T **do**
> $w \leftarrow \lambda w+\log \left(\pi\left(a_{t} | s_{t}\right)\right)$ // w has the dependencies including a_t
> $v \leftarrow w-\log \left(\pi\left(a_{t} | s_{t}\right)\right)$ // v has the dependencies excluding a_t
> $deps\leftarrow f(w)-f(v)$ // f applies the $\odot$ operator on the log-probabilities
> $J \leftarrow J+deps \cdot A\left(s_{t}, a_{t}\right)$ //The dependencies are weighted by the advantage $A\left(s_{t}, a_{t}\right)$
> **end for**
> **return** J

> **function**$f(x)$
> **return** $exp(x-stop_gradient(x))$
> **return function**

> 本文提出的objective兼容了上面的J_{LVC}和$J_{\diamond}$，具体就是调整λ的值

当$\lambda=0$时，该估计器跟J_{LVC}相同，尽管其利用了优势，可能有较低的方差但是与优势估计器的选择无关，并且有偏差。当$\lambda=1$时，估计器就涵盖了$J_{\diamond}$，当优势估计器自身无偏时也无偏。λ的中间值则能够权衡方差与偏差，具体在第四节中展示。本目标的新形式能够平滑地降低[Furmston et al., 2016] 和[Rothfuss et al., 2018]中高方差的影响，而不是突然砍去这些方差项。

4 实验

4.1 任意阶微分下的偏差与方差

为了开始的分析简便易懂，先使用小的随机的具有五个state的MDP，其中每个state包含四个action，并且reward只依赖于state。对于这些MDP折扣指可能通过分析计算得出，如下。
P^{π}是由MDP的转变函数(transition function)$P\left(s, a, s^{\prime}\right)$以及表格策略(tabular policy)$\pi$推出，如下： $$ P_{s s^{\prime}}^{\pi}=\sum_{a} P\left(s, a, s^{\prime}\right) \pi(a | s) $$ 令P_0为初始状态分布，然后t时刻的状态概率分布为$p_{s_{t}}=\left(P^{\pi}\right)^{t} P_{0}$，$t$时刻的平均奖励为$r_{t}=R^{T} p_{s_{t}}$，$R$是每个状态对应的奖励向量，最后得出： $$ \begin{aligned} V^{\pi} &=\sum_{t=0}^{\infty} \gamma^{t} r_{t}=R^{T} \sum_{t=0}^{\infty}\left(\gamma P_{\pi}\right)^{t} P_{0} \\ &=R^{T}\left(I-\gamma P_{\pi}\right)^{-1} P_{0} \end{aligned} $$ 这里V^{π}关于π可微并且可通过自动微分软件包容易地计算得出。

低方差任意阶无偏估计器。 图Fig2展示了不同目标以及不同阶下估计导数与真实导数相关性随样本数(batch size)的变化。具体比较了原始的DiCE估计器，本文的loaded DiCE，以及[Mao et al., 2019]提出的只包含一个基线(baseline)的目标。对于本文的loaded DiCE使用一个$\tau=0$的优势估计器A^{GAE}(可见2.3节)也就等价于一个精确的值函数，然后令$\lambda=1$使得它无偏。 Fig2第一张图可以明显看出由于都是无偏的，所以当

batch size足够大的时候最终都会收敛于真实导数。然而对比使用了优势估计器的loaded DiCE，其方差会急剧减少并且估计量会很快收敛。本文展示了LVC的效果，在一阶下，它跟[Mao et al., 2019]但是比Loaded DiCE效果要差，因为没使用优势估计器(advantage)。对于高阶情况，方差低但是有偏。

Fig2

包含优势估计的方差偏差权衡。 图Fig3a分别展示了估计导数在τ范围内的偏差和标准差，以及使用一个不太精确的值函数(将每个状态的真实值函数与高斯噪声混合来模拟函数近似)。可以看到有时估计器的选择不仅在一阶影响方差偏差权衡，在任意阶同理。
折扣因子下的方差偏差权衡。 图Fig3b别展示了估计导数在λ范围内的偏差和标准差。为了孤立λ的影响使用精确的值函数并且令$\tau=0$，因此它的绝对方差和偏差要比Fig3a小。可以看到，一阶导数下不受λ影响。然后，对于高阶情况，λ则有很大影响。并且对于三阶当$\lambda=0.75$时，偏差不再保证单调性，但是二阶以下这种情况要少，可以看作是MDP的特殊情况。

Fig3

4.2 模型不可知(MAML)的元强化学习和loaded DiCE

现在接着[Finn et al.，2017]，将提出的新估计器应用到连续控制的元强化学习中。

> 为什么会将本文的objective或者说梯度估计器应用到这里MAML，重点就是因为内循环中关键的梯度估计过程。

该方法是对多个任务进行采样，并在内环策略梯度优化步骤中调整策略。 然后，在外循环中，更新初始参数，以使自适应后策略的奖励最大化。 外循环优化取决于自适应后的参数，而后者取决于内循环中估计的梯度。 因此，在外循环优化中存在重要的高阶项。 因此，使用正确的估计器进行内循环优化会影响整体元训练过程的效率以及最终解决方案的质量。

对于内循环的优化，使用本文提出的新目标，并且τ和λ可变。我们首先令$\lambda=0$改变τ，然后使用最佳的τ改变λ。对于外循环的优化，就使用带有一个基线的普通策略梯度(vanilla policy gradient)。外循环也可以用其他基于梯度的策略优化算法，但是本文就选择一个简单的版本，在某种程度上孤立内循环估计器的影响。

Fig4展示了结果。在CheetahDir任务中，如果τ过大，那么会导致估计器方差很大效果就很差。而τ在CheetahVel任务中影响很小。注意在[Finn et al., 2017]这些任务的回合(episode)都较短，γ较低，值函数也是简单的线性函数，这些都支持一个较高的τ。但是对于较高的方差回报或更好的值函数，更多地依赖于学习值函数(通过使用较低的τ)可能是有效的。

在两种任务环境中，$\lambda=1$都会导致高方差。当值函数更好时我们目标的无偏版本($\lambda=1$)将更有价值并且能够有效减轻方差。在CheetahVel任务中，当λ低至接近于0但不等于0时学习速度更快。[Furmston et al., 2016]的分析表明当经λ折扣后的高阶项数值会随策略接近局部最优时变小。这与本文的经验发现一致，即非零λ可能学习得更快，并且效果还可处于相似水平。到这里总结本文的Load DiCE提供了对高阶估计量的有意义的控制，并且对实际用例产生了重大影响。

5 结论

> 基本把上文概括了一下。。。

本文推出了一个理论上合理的目标，该目标可以将通用优势函数应用于强化学习等一系列问题中的任意阶导数的估计。 在函数逼近的情况下，该目标可提高在高阶导数中权衡偏差和方差的能力。 重要的是，就像潜在的DiCE目标一样，本文的单个目标会在重复自动微分的情况下为任意阶导数生成估计量。 此外，本文提出了一种简单的方法，可以将更远的因果相关性对高阶导数的估计的影响进行折现，从而为偏差和方差的折衷提供了另一个轴。 根据经验，我们使用小的随机MDP来证明高阶导数估计的偏差和方差行为，并进一步证明其在元强化学习中的效用。

本文或可在元学习、多智能体学习以及更高阶优化的应用中使用本文提出的目标。未来工作中还希望重新考虑折扣因子λ的选择，这是一种启发式方法，可以限制高方差项的影响。进一步的理论分析也可能有助于确定语境(context), 其中的高阶依赖对于优化很重要。最后，甚至有可能对超参数τ和λ本身进行元学习。

6 复现