# Trust Region Policy Optimization

# 信赖域策略优化

John Schulman: 伯克利的博士，2016毕业，导师为Pieter Abbeel

Pieter Abbeel: 是伯克利的副教授，同时也是OpenAI的研究科学家，是机器人强化学习领域最有影响力的人之一。毕业于斯坦福大学。

## 策略梯度的缺点：

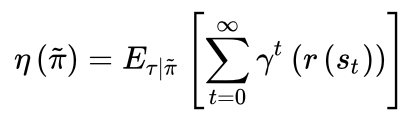
根据策略梯度方法，参数更新方程式为：

../../../../Desktop/屏幕快照%202018-02-26%20上午8.53.19.pn

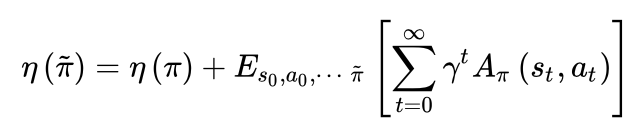
策略梯度算法的硬伤在于更新步长，当步长不合适时，容易导致越学越差，最后崩溃。所以，合适的步长对于强化学习非常关键。

所谓合适的步长实质当策略更新后，回报函数的值不能更差。如何选择步长，或者如何找到新的策略使得新的回报函数的值单调递增或单调不减，这是TRPO要解决的问题。

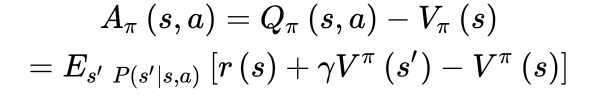
用表示一组状态-行为序列：s0, u0, … , sH, uH，则强化学习的回报函数为：



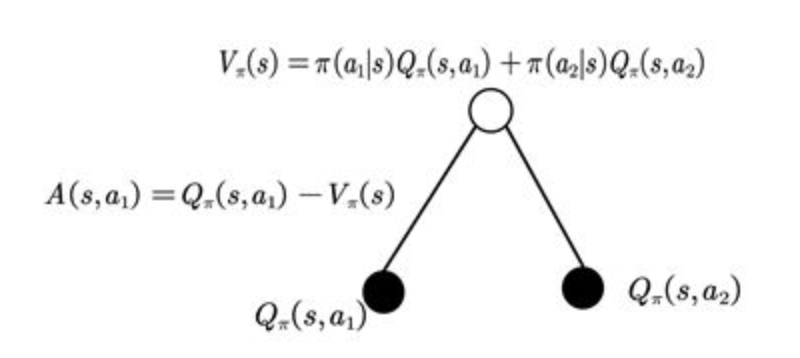
TRPO是找到新的策略，使得回报函数单调不减，想法为将新的策略所对应的回报函数分解成旧的策略所对应的回报函数+其他项。TRPO 的起点就是2002年Sham Kakade提出的这样一个等式：



其中：



称为**优势函数。**

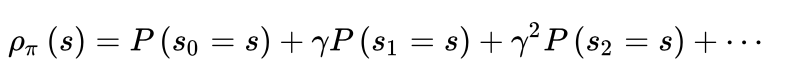


值函数V(s)为在该状态下所有可能动作所对应的动作值函数乘以采取该动作的概率的和，即时改状态下所有动作值函数关于动作概率的平均值。

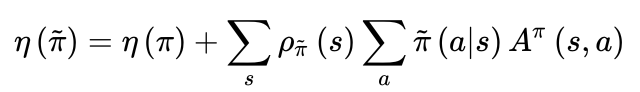
动作值函数Q(s,a)是单个动作所对应的值函数。

Q(s, a) – V(s)能评价当前动作值函数相对于平均值的大小。所以，这里的**优势指的是动作值函数相比于当前状态的值函数的优势。**如果优势函数大于零，则说明该动作比平均动作好，如果优势函数小于零，则说明当前动作还不如平均动作好。

定义：



则：

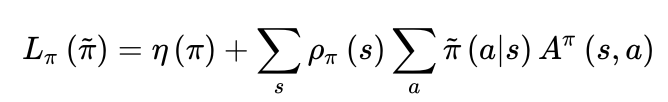


# ../../../../Desktop/屏幕快照%202018-02-26%20上午9.35.37.pn

### TRPO第一个技巧：

引入第一个技巧对状态分布进行处理。忽略状态分布的变化，依然采用旧的策略所对应的状态分布。当新旧参数很接近时，用旧的状态分布代替新的状态分布也是合理的。

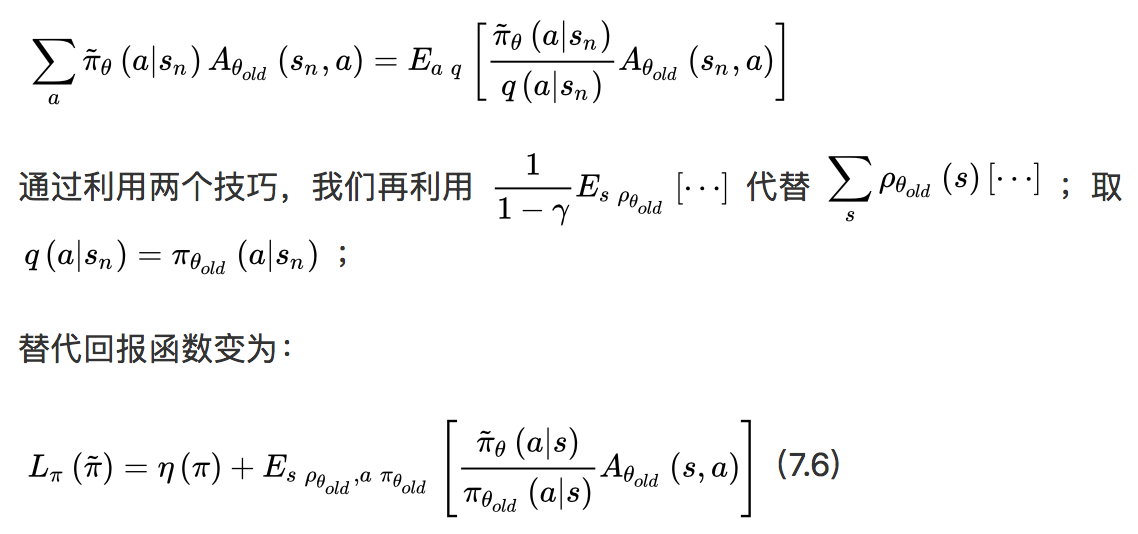
原代价函数变为：



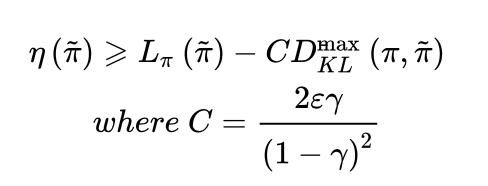
但其中第二项策略部分中的动作a是由新的策略生成，但此时新的策略是带未知参数的，无法用来产生动作。

### TRPO第二个技巧：

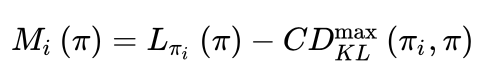
第二个技巧是利用重要性采样对动作分步进行处理。



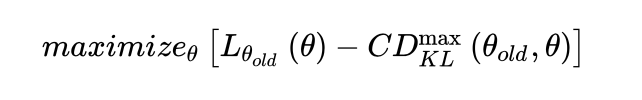
设备如第二个重量级的不等式：



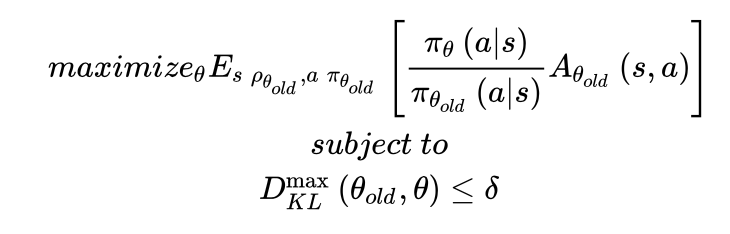
其中DKL() 是两个分布的KL散度。（相对熵，又称为KL散度，信息散度，相对熵是两个概率分布差别的非对称性的度量）该不等式给了的下界，定义为：



则使得Mi最大的新策略就是要更新的策略。该问题形式化为：



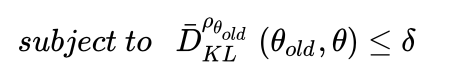
如果利用惩罚因子C则每次迭代步长很小，则问题转化为：



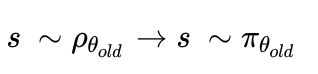
因为有无穷多的状态，因此约束条件DKL有无穷多个，问题不可解。

### TRPO第三个技巧

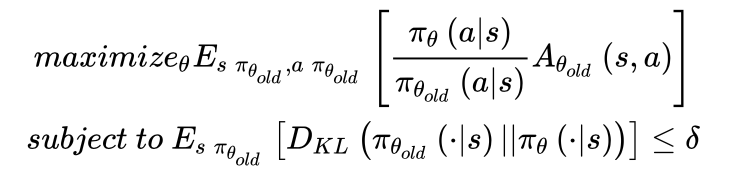
在约束条件中利用平均KL散度代替最大KL散度，即：



### TRPO第四个技巧



最终TRPO问题化简为：



接下来就是利用采样得到数据，然后求样本均值，解决优化问题即可。

## 原文：

# Trust Region Policy Optimization

加州大学伯克利分校电子工程与计算机科学系

2017.4.20

**Abstract**：我们描述了一个保证单调改进的优化策略的迭代过程，通过对理论上合理的过程进行几次近似，我们开发了一种称为信赖域策略优化的使用算法。该算法近似于自然策略梯度方法，对于优化大型非线性策略（例如神经网络）非常有效。我们的实验展示了它在各种任务上的鲁棒性：学习模拟机器人游泳，跳跃和行走步态；并使用屏幕图像作为输入来玩Atrai游戏。虽然它的近似值偏离了理论，但TRPO倾向于提供单调的改进，对超参数的调整很少。

1. **Introduction**

大多数策略优化算法可以分为三大类：1）策略迭代方法，在估计当前政策的价值函数和改进政策之间交替进行（2005）；2）策略梯度方法，使用从样本轨迹中获得的期望受益（总回报）梯度的估计值（2008）；3）无衍生优化算法，如交叉熵算法CEM和协方差矩阵自适应法CMA，将回报作为黑箱函数来处理，以便根据策略参数进行优化（2006）。

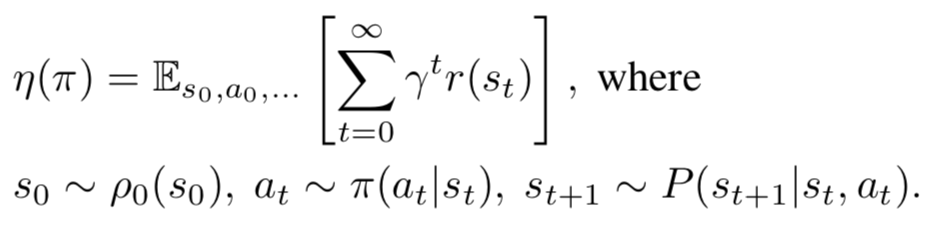
一般的无衍生随机优化法如CEM和CMA在很多问题上是首选，因为它们易于理解和实施且取得了良好的结果。例如，虽然俄罗斯方块是近似动态规划（ADP）的经典基准问题，但随机化方法难以胜任。对于连续控制问题像CMA这样的方法已经成功地学习了控制策略，以满足具有低维度参数化的手工策略类的运动等具有挑战性的任务。由于基于梯度的优化算法比无梯度方法具有更好的样本复杂性保证，所以ADP和基于梯度的方法一致击败无梯度随机搜索的能力并不令人满意。连续的基于梯度的优化在学习具有大量参数的监督学习任务的函数逼近方面非常成功，并且将其成功扩展到强化学习可以有效地训练复杂而强大的策略。

在本文中，我们首先证明，最小化某个特定的目标函数可以保证不平凡的步长大小的政策改进。然后，我们对理论上对齐的算法进行一系列的近似，得到一个实用的算法，我们称之为信赖域策略优化（TRPO）。我们描述了这种算法的两种变体：首先，单路径方法，可以在无模型设置中应用; 其次，vine法需要系统恢复到特定状态，这通常只能在模拟中实现。这些算法是可扩展的，可以优化带有数以万计参数的非线性策略，这对以前的无模型策略搜索提出了重大挑战。 在我们的实验中，我们展示了相同的TRPO方法可以学习游泳，跳跃和行走的复杂政策，以及直接从原始图像中玩Atari游戏。

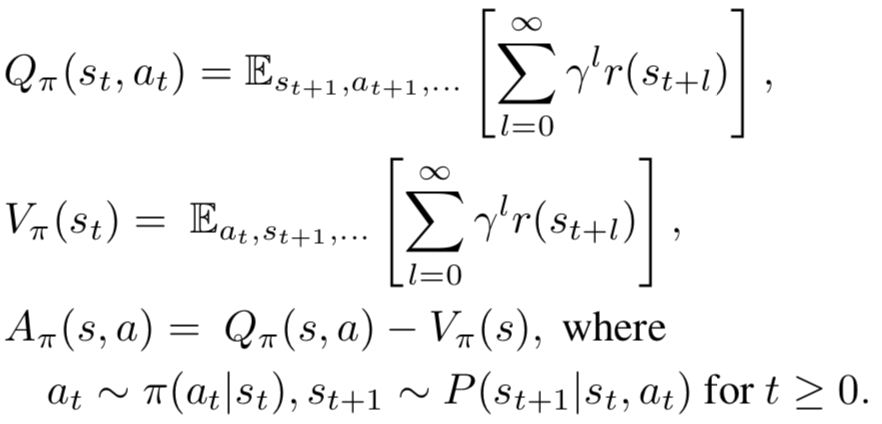
1. **Preliminaries**

考虑由(S, A, P, r, ρ0, γ)定义的MDP过程，其中S是有限状态集合，A是行为的有限行动的集合，P： S×A×S→R为转移概率分布。r：S→R为奖励函数，ρ0：S→R为初始状态s0的分布，γ∈（0,1）为折扣因子。

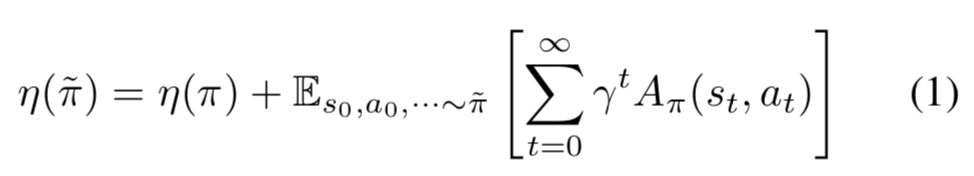
令π表示一个随机策略π：S×A→[0,1]，令η（π）表示其预期折扣奖励：



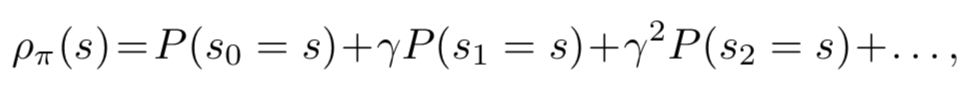
我们将使用状态-行动值函数Qπ，值函数Vπ和优势函数Aπ的以下标准定义：



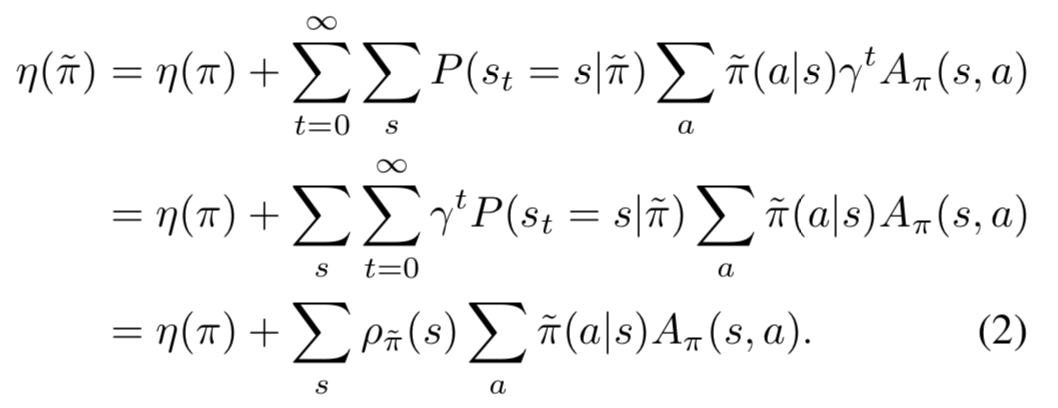
π ̃表示另一个政策的预期回报，其优势超过π，在时间步长上累积:



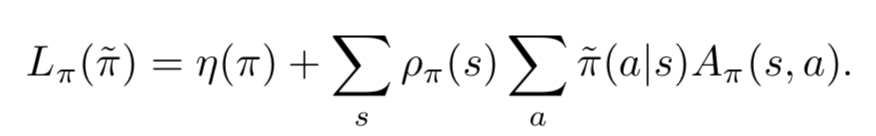
设ρπ为（非标准化）折扣访问频率:



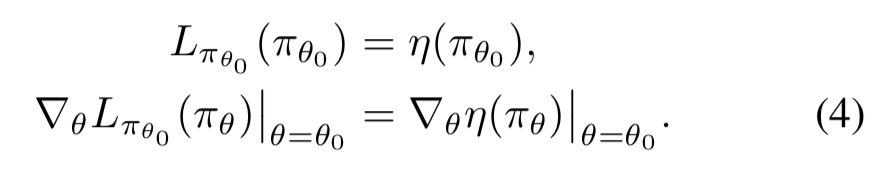
其中s0〜ρ0，动作根据π来选择。 我们可以用方程式（1）重写状态而不是时间步长：



这个方程意味着任何政策更新π→π ̃在每个状态s上具有非负的预期优势，即一个π ̃（a | s）Aπ（s，a）≥0，可以保证提高政策绩效η，或者在所有地方的预期优势都为零的情况下保持不变。这意味着经典结果是，如果至少有一个具有积极优势的状态动作对，那么使用确定性策略**π ̃（s）= argmaxaAπ（s，a）**的精确策略迭代执行的更新改进了策略值和非零状态访问概率，否则算法已经收敛到最优策略。然而，在近似的情况下，由于估计和逼近误差，通常将不可避免地存在一些状态，其中预期的优势是负的，ρπ（s）对π ̃的复数依赖性使方程（2）难以直接优化。 因此，我们引入下面的局部近似η：

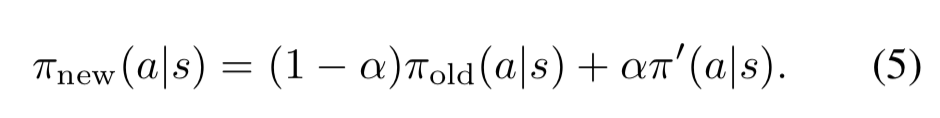


注意到L使用概率ρπ而不是ρπ ̃，忽略了策略变化导致的概率的变化。 然而，如果我们有一个参数化策略πθ，其中πθ（a | s）是参数向量θ的可微函数，则L将η与一阶导数相等。也就是说，对于任何参数值θ0:

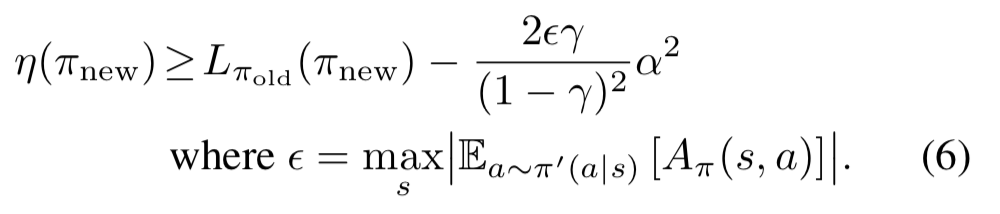


(4)意味着足够小的步长πθ0→π ̃可以改善η，但是对于步长取多大并没有指导。

为了解决这个问题，提出了一中称为**保守策略调整的策略更新方法**，可以为改进η提供明确的下限。为了定义保守的政策迭代更新，令πold表示当前的策略，π'= argmax π'L πold（π'）。新的策略πnew定义为：



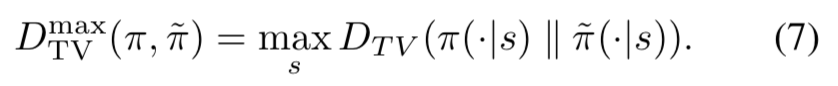
下界为：

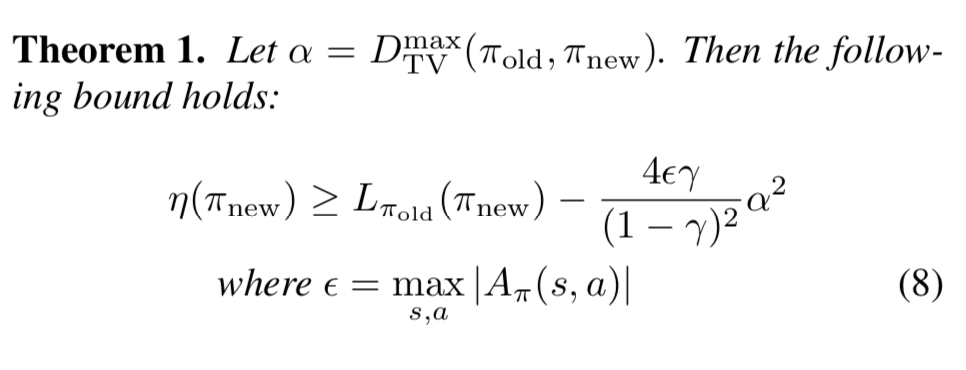


但是，请注意，迄今为止，此限制仅适用于由公式（5）生成的混合策略。 这个政策阶级在实践中是笨拙和限制性的，并且希望实际的政策更新方案适用于所有一般的随机政策类别。

1. **Monotonic Improvement Guarantee for General Stochastic Policies(一般随机策略的单调改进保证)**

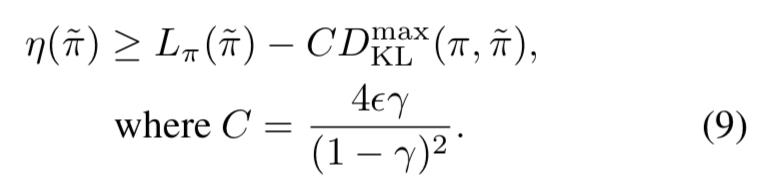
等式（6）适用于保守的策略迭代，意味着可以改进右侧的策略更新可以保证提高真实性能η。我们的主要理论结果是，方程（6）中的政策改进界限可以扩展到一般的随机政策，而不仅仅是混合政策，用π和π之间的距离度量替换α，常数ε适当。 由于实践中很少使用混合策略，因此这一结果对于将改进保证延伸到实际问题至关重要。 我们使用的特定距离度量是总变差发散度，../../../../Desktop/屏幕快照%202018-03-02%20上午9.32.15.pn,定义：





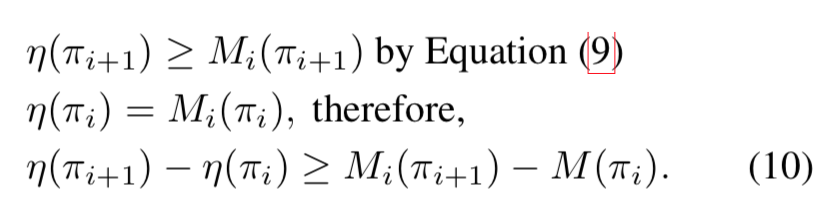
我们在附录中提供了两个证明。 第一个证明表明Kakade和Langford的结果使用了这样一个事实，即两个分布随机变量的总变化散度小于α可以耦合，因此它们的概率为1 - α。 第二个证明使用了扰动理论。

接下来，我们注意到总变差背离和KL散度之间的下列关系:DTV（p || q）2≤DKL（p || q）。 令Dmax KL（π，π ̃）= maxs DKL（π（·| s）||π ̃（·| s））。 下面的界限直接来自定理1：



算法1描述了一个基于方程（9）中的策略改进约束的近似策略迭代方案。 请注意，目前，我们假定对优势值Aπ进行精确评估。

从方程（9）得出，算法1保证产生一个单调改进的策略序列η（π0）≤η（π1）≤η（π2）≤....让Mi（π）= L（π） - CDmax（π，π）。 然后:



因此，通过在每次迭代中使Mi最大化，我们保证真实目标η是非递减的。 该算法是一种少数化最大化（MM）算法（Hunter＆Lange，2004），它是一类也包含期望最大化的方法。 在MM算法的术语中，Mi是替代函数，它在πi时使η等于零。 该算法也是近似梯度方法和镜像下降的重要方法。

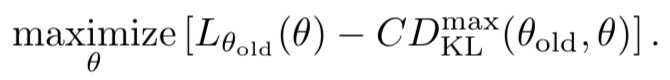
我们在后面的章节中提出的信任域策略优化是对算法1的近似，算法1对KL偏差采用约束而非惩罚来强有力地允许大量更新。

1. **Optimization of Parameterized Policies (参数化策略的优化)**

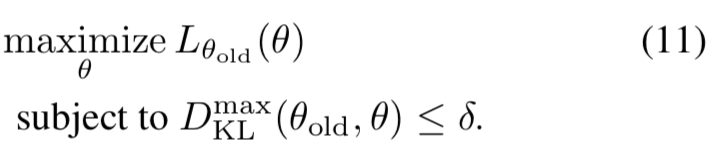
在前面的章节中，我们考虑了政策优化问题，而不考虑π的参数化，并且假设政策可以在所有状态进行评估。我们现在描述如何在有限的样本计数和任意参数化下从这些理论基础推导出实用的算法。

由于我们考虑参数化策略πθ（a | s）和参数向量θ，所以我们将使先前的符号过载以使用θ而不是π的函数，例如， η（θ）：=η（πθ），Lθ（θ ̃）：=Lπθ（πθ ̃），DKL（θ||θ ̃）：= DKL（πθ||πθ ̃）。 我们将使用θold来表示我们想要改进的以前的策略参数。

前面的部分表明η（θ）≥Lθold（θ） - CDmax（θ，θ），在θ=θold时相等。 因此，通过执行以下最大化，我们保证提高真实目标η：



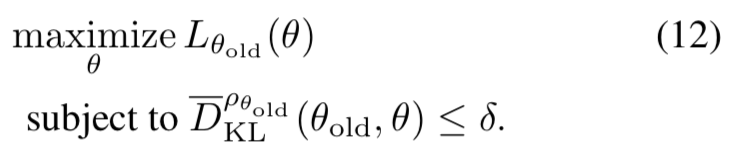
在实践中，如果我们使用上述理论推荐的惩罚系数C，步长将会非常小。 强有力地采取更大步骤的一种方法是对新政策和旧政策（即信托区域约束）之间的KL分歧施加约束：



这个问题强加了一个约束，即KL散度在状态空间的每个点都是有界的。 虽然它受到理论的激励，但由于约束条件较多，此问题不切实际。 相反，我们可以使用考虑平均KL散度的启发式近似：



因此，我们建议解决以下优化问题以生成策略更新：

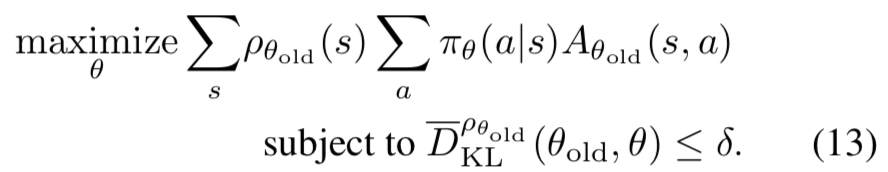


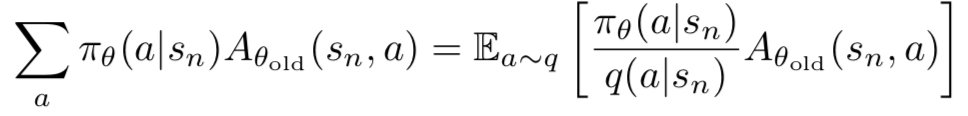
在以前的工作中已经提出了类似的政策更新（Bagnell＆Schneider，2003; Peters＆Schaal，2008b; Pers等，2010），我们将我们的方法与第7节中的现有方法和第8节中的实验 我们的实验也表明这种约束更新类型与经验公式（11）中的最大KL散度约束具有相似的经验性能。

1. **Sample-Based Estimation of the Objective and Constraint 基于样本的目标约束估计**

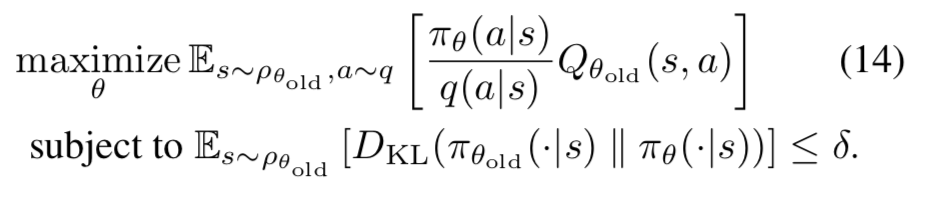
前一部分提出了一个关于政策参数的约束优化问题（方程（12）），该方案优化了预期总体回报η的估计，这受到每个更新期间政策变化的约束。本节介绍如何使用Monte Carlo模拟近似客观和约束函数。

我们试图解决下面的优化问题，通过扩展公式（12）中的Lθold得到：





在方程（13）中的优化问题完全等同于下面的一个，根据期望写成：



剩下的就是用样本平均值代替预期，并用经验估计代替Q值。 以下部分描述了执行此估算的两种不同方案。

我们称之为single路径的第一个抽样方案是通常用于策略梯度估计的方法（Bartlett＆Baxter，2011），并且基于抽样单个轨迹。 我们称之为vine的第二种方案涉及构建一个展示集，然后在展示集中从每个状态执行多个操作。 这种方法主要是在政策制定方法的背景下进行探索的（Lagoudakis＆Parr，2003; Gabillon等，2013）。

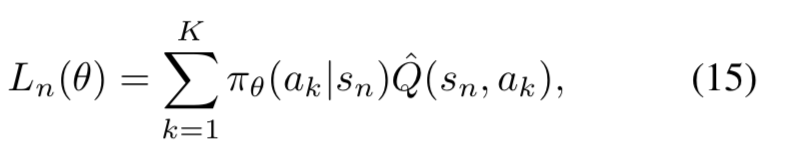
* 1. **Single Path**

在这个估计过程中，我们通过对s0〜ρ0进行采样，然后模拟时间周期的观测值s0，a0，s1，a1，...，sT-1，aT-1，sT的多项式时间序列来收集状态序列。 因此，q（a | s）=πθold（a | s）。 Qθold（s，a）在每个状态 - 动作对（st，at）处通过沿着轨迹获得未来奖励的折扣总和来计算。

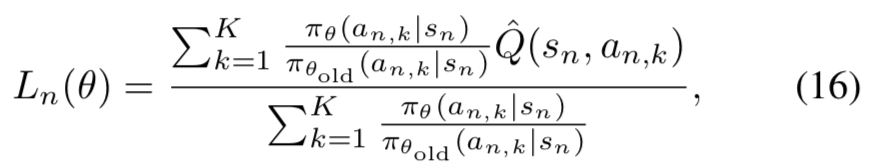
* 1. **Vine**

在这个估计过程中，我们首先对s0〜ρ0进行采样并模拟策略πθi以产生许多运动轨迹。 然后，我们沿着这些轨迹选择N个状态的子集，记为s1，s2，...，sN，我们称之为“推出设置”。 对于卷展栏集中的每个状态sn，我们根据ank〜q（·| sn）对K个动作进行采样。 任何包含πθi（·| sn）支持的支持q（·| sn）的选择将产生一致的估计量。 在实践中，我们发现q（·| sn）=πθi（·| sn）在连续问题（例如机器人运动）方面效果很好，而均匀分布在离散任务（例如Atari游戏） 它有时可以实现更好的探索。

在小而有限的动作空间中，我们可以从给定状态中为每个可能的动作生成一个展开。 来自单一状态sn对Lold的贡献如下：



动作空间是A = {a1，a2，...，ak}。 在大的或连续的状态空间中，我们可以使用重要性抽样构建替代目标的估计。 在单个状态s下获得的L的自归一化估计量为:



假设我们执行了K个动作an，1，an，2，... , anK。 这个自我归一化的估计器消除了使用Q值基线的需要（请注意，通过向Q值添加常数来改变梯度）。 对sn〜ρ（π）进行平均，我们得到Lθold的一个估计量，以及它的梯度。

图1说明了vine和single path的方法。我们使用vine，因为用于取样的轨迹可以比喻为葡萄藤的茎，葡萄藤在不同的点（分枝集）分枝成几个短的分枝展示轨迹）。

鉴于替代目标中相同数量的Q值样本，vine方法相对于我们的目标局部估计的single path方法的好处具有低得多的方差。也就是说，vine方法可以更好地估计优势值。vine方法的缺点是我们必须对这些优势估计中的每一个执行更多的模拟器调用。此外，vine方法要求我们在卷展栏集合中从每个状态生成多个轨迹，这将该算法限制在可将系统重置为任意状态的设置。相反，单路径算法不需要状态重置，可以直接在物理系统上实现（Peters＆Schaal，2008b）。

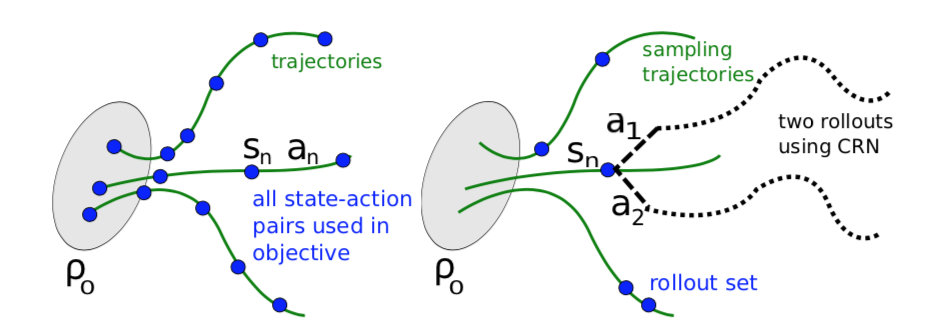


图1：左图：单路径程序的插图。 在这里，我们通过模拟政策生成一组轨迹，并将所有状态 - 行为对（sn，an）合并到目标中。 右图：vine程序的插图。 我们生成一组“干线”轨道，然后从达到状态的子集中生成“分支”卷展栏。 对于这些状态sn中的每一个，我们执行多个操作（此处为a1和a2），并在每个操作后执行一次转出，使用常见随机数（CRN）来减少方差。

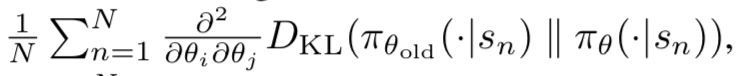
1. **Practical Algorithm 实用算法**

在这里我们提出两种基于上述想法的实用策略优化算法，它们使用前一节中的单路径或藤蔓采样方案。 算法重复执行以下步骤：

1.使用单一路径或藤蔓过程收集一组状态行为对以及蒙特卡罗估计的Q值。

2.通过对样本进行平均，构建公式（14）中的估计目标和约束。

3.近似解决这个约束优化问题来更新策略的参数向量θ。 我们使用共轭梯度算法，然后是行搜索，总共只比计算梯度本身稍微昂贵。 详情请参阅附录C.

关于（3），我们通过分析计算KL散度的Hessian而不是使用梯度的协方差矩阵来构造Fisher信息矩阵（FIM）。 也就是说，我们估计Aij为分析估计器集成了每个状态sn上的动作，而不依赖于被采样的动作。 如附录C所述，此分析估计器在大规模环境中具有计算优势，因为它不需要存储密集Hessian或所有策略梯度的批处理轨迹。 如实验所示，政策改进的速度与实证FIM相似。

让我们简要总结一下第3节的理论和我们所描述的实际算法之间的关系：

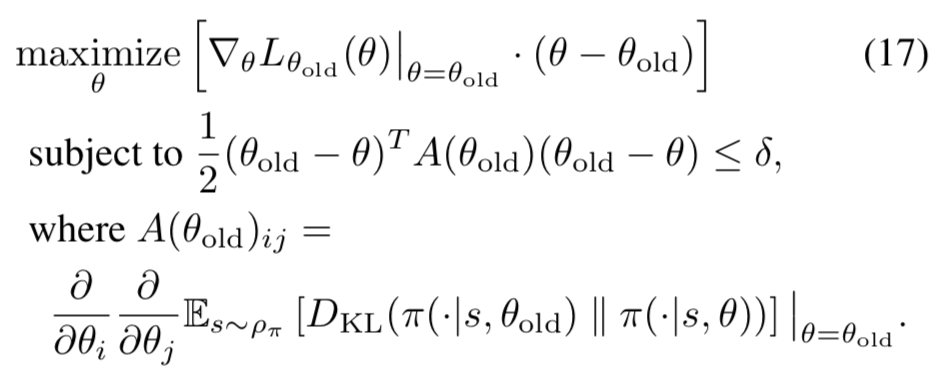
•该理论认为，优化替代目标与KL散度的惩罚。 然而，大的惩罚系数C导致禁止性的小步骤，所以我们想要降低这个系数。 从经验上讲，很难有力地选择惩罚系数，所以我们用一个硬约束代替惩罚，参数δ（KL散度的界限）。

•对DKLmax（θold，θ）的约束难以进行数值优化和估计，所以我们限制DKL（θold，θ）。

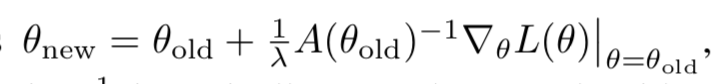
•我们的理论忽略了优势函数的估计误差。 卡卡德和兰福德（Kakade＆Langford，2002）在推导过程中考虑了这个错误，并且在本文的设置中也会有相同的论点，但为简单起见，我们忽略它们。

1. **Connections with Prior Work**

正如第4节中提到的，我们的推导导致了一个政策更新，与以前的几种方法相关，为多个政策更新计划提供了一个统一的视角。 通过使用L的线性逼近和DKL约束的二次逼近，可以获得自然政策梯度（Kakade，2002）作为方程（12）中更新的特例，导致以下问题：

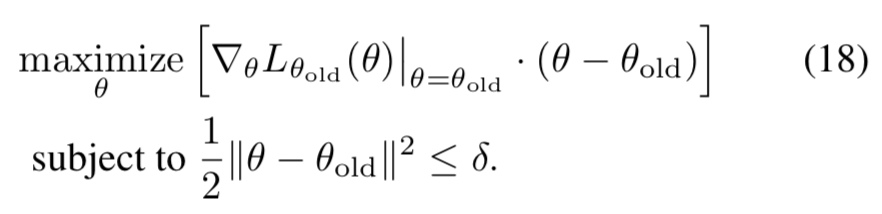


更新为：



其中步长1/通常被视为算法参数。 这与我们的方法不同，后者在每次更新时强制约束。 虽然这种差异可能看起来很微妙，但我们的实验证明，它能显着改善算法在较大问题上的性能。

我们还可以通过使用l2约束或惩罚来获得标准策略梯度更新：



策略迭代更新也可以通过使用方程（3）中定义的L来解决无约束问题maximumπ Lold（π）来获得。

其他几种方法采用类似于公式（12）的更新。 相对熵政策搜索（REPS）（Peters等，2010）约束状态行为边际p（s，a），而TRPO约束条件p（a | s）。 与REPS不同，我们的方法不需要在内部循环中进行昂贵的非线性优化。 Levine和Abbeel（2014）也使用KL分歧约束，但其目的是鼓励政策不偏离估计的动力学模型有效的区域，而我们不试图明确估计系统动力学。 Pirotta等人 （2013）也建立并推广了Kakade和Langford的结果，他们从这里推导出不同的算法。

1. **Experiments**

我们设计了我们的实验来调查以下问题：

1.单径和藤蔓采样程序的性能特征是什么？

2. TRPO与先前的方法（例如自然政策梯度）相关，但做了一些改变，最显着的是通过使用固定的KL分歧而不是固定的惩罚系数。 这如何影响算法的性能？

3.TRPO可以用来解决具有挑战性的大规模问题吗？ 在应用于大规模问题时，TRPO与其他方法相比，在最终性能，计算时间和样本复杂性方面如何？

为了回答（1）和（2），我们比较了TRPO的单一路径和藤蔓变体的性能，几种消除的变体和一些先前的策略优化算法。 关于（3），我们表明，单一路径和藤蔓算法都可以从头开始获得高质量的运动控制器，这被认为是一个难题。 我们还展示了这些算法在学习使用具有数万个参数的卷积神经网络从图像中播放Atari游戏的策略时产生了有竞争力的结果。

* 1. **Simulated Robotic Locomotion**

我们使用MuJoCo模拟器进行了机器人运动实验（Todorov等，2012）。 三个模拟机器人如图2所示。机器人的状态是它们的广义位置和速度，控制是关节转矩。 由于接触引起的不容忽视，高维度和不平滑的动态特性，使得这些任务非常具有挑战性。 我们的评估中包含以下模型：

1.游泳者。 10维状态空间，前向进展的线性回报和联合努力产生回报的二次惩罚r（x，u）= vx-10-5‖u‖2。 游泳者可以通过起伏运动向前推进。

2.跳跃者。 12维状态空间，与游泳者相同的奖励，在非终极状态下获得+1奖励。 当跳跃跌倒时，我们结束了这些事件，这是由躯干高度和角度的阈值定义的。

3.行走。 18维状态空间。 对于步行者，我们增加了对脚的强烈撞击的惩罚，以鼓励步行顺利，而不是跳步。

我们对所有实验使用δ= 0.01。 有关所用实验设置和参数的更多详细信息，请参阅附录中的表2。 我们使用神经网络来表示政策，其架构如图3所示，附录D中提供了更多细节。为了建立标准基线，我们还包括了基于Barto等人的公式。 （1983）使用了一个具有六个参数的线性策略，该策略易于使用无导数黑箱优化方法进行优化。

在比较中考虑了以下算法：单路径TRPO;藤TRPO;交叉熵法（CEM），无梯度法（Szita＆Lörincz，2006）;协方差矩阵自适应（CMA），另一种无梯度方法（Hansen＆Ostermeier，1996）;自然梯度，经典的自然策略梯度算法（Kakade，2002），它不同于单一路径，通过使用固定惩罚系数（拉格朗日乘子）而不是KL散度约束;经验FIM，与单一路径相同，除了FIM是使用梯度的协方差矩阵而不是分析估计来估计的; max KL，它只能用于推车杆问题，并且使用方程（11）中的最大KL散度而不是平均散度，从而使我们能够评估这种近似的质量。实验中使用的参数在附录E中提供。对于天然梯度方法，我们以3为因子扫描步长的可能值，并根据最终性能取最佳值。

显示每种算法五次平均总回报的学习曲线如图4所示。单一路径和藤TRPO解决了所有问题，产生了最佳解决方案。自然梯度在两个较容易的问题上表现良好，但无法产生促进进步的跳跃和步行步态。这些结果提供了经验证据，与使用固定惩罚相比，约束KL散度是一种更稳健的方式来选择步长和快速，一致的进展。 CEM和CMA是无衍生算法，因此它们的样本复杂度与参数数量不符，并且它们在较大的问题上表现不佳。由于约束条件更为严格，最大KL方法比我们的最终方法学得更慢，但总体而言，结果表明平均KL散度约束与理论上对齐的最大KL散度有相似的影响。 TRPO的政策视频可在项目网站上查看：http：//sites.google.com/site/trpopaper/。

请注意，TRPO使用通用策略和简单的奖励功能，使用最小的先验知识来学习所有步态。 这与大多数现有的学习移动方法形成鲜明对比，它们通常依赖于手工设计的策略类，这些策略类明确地编码了平衡和步进的数字（Tedrake et al。，2004; Geng et al。，2006; Wampler＆Popovic ，2009）。