OptiDICE: Offline Policy Optimization via Stationary Distribution Correction Estimation

Chen Gong

04 October 2021

目录

本文 “OptiDICE: Offline Policy Optimization via Stationary Distribution Correction Estimation” 于 2021 年发表于 ICML，是韩国科学技术高级研究院的工作。和别人讨论时，有人提出此工作的复现效果还不错，而且此方法有希望可以广泛的运用于各类 offline RL 算法中，所以仔细阅读了一下此文章。此文章的基本思路也不难，感觉是 AlgaeDICE 的翻版，不过其运用非常巧妙。在 DAI Bo 描述的 DICE 中，DICE 的最大好处是可以从满足任意分布的数据集中求解出平方分布修正因子来对当前策略就行评估，在 OPE 问题中取得了很大的成功。

看到了知乎中的 “一个科学家” 的文章：https://zhuanlan.zhihu.com/p/377994641，其中提到了欣赏和提炼他人文章 insight 的重要性。此篇文章对我启发很大，于是决定之后的文章都思考其 insight。

本文的 insight 在于，OptiDICE 通过直接对平稳分布修正因子 ω 建模，得到数据集中的行为策略分布和最优策略分布之间的关系。从而通过 ω^* 还原出最优策略分布。从而避免了传统的从价值函数角度进行优化，而导致的分布偏移问题。同时本文第一次从优化平稳分布修正因子的角度出发解决 RL 问题。

1 Key idea

在传统的机器学习方法中，其标准的 workflow 是在数据集上训练和验证模型，然后在训练完成之后固定参数部署 offline 模型。这样的离线训练使我们能够在不需要实际部署的情况下解决系统的各种操作需求。但是这样的训练方式，并不适合于强化学习，因为强化学习中需要不断地探索环境，并且从不断地试错中学习到理想的状态。这也是强化学习在实际应用中的一个难点，因为有点探索行为非常的昂贵，或者危险。

Offline RL 中将 RL 问题转换到 offline 的设定中。但是，offline RL 中一个非常重要的问题是分布偏移（给定的 offline dataset 分布不能很好地衡量当前的模型，贝尔曼方程中的期望的分布是当前策略实际运行产生的数据分布）的问题，这是由于我们训练得到的策略偏离了收集数据的策略而导致的。如果，没有持续的 online 收集数据，将导致分布偏移无法被校正，将导致使用 bootstrapped（bootstrapped 简单的可以理解为动态规划中使用贝尔曼方程迭代的过程）和函数近似的 RL 算法中的重要问题：**由于预测 OOD 的动作价值来更新 Q 函数，会造成 model free 中的动作价值过估计问题的恶化。（这里为什么会用过估计来描述呢？因为过估计实际上就是由于 Q 函数估计的误差所导致的，这里的过估计问题很大，也可以等价的理解为 Q 函数的误差很大。所谓 OOD 问题，指的是训练用的数据集和目标分布的数据集不一致。而 OOD 动作价值，就是对于一个策略关于状态 s 的输出 a，如果 (s,a) 在训练数据集中不存在的话，那么在使用贝尔曼方程更新时误差会非常大。而在 online 的 RL 中，可以通过和环境的交互去估计 $Q(s,a)$，并且 replay buffer 中的数据会随着目标策略的更新，不断地更新，这就是一个修正的过程。）**为了缓解此问题，大部分目前的 offline RL 算法通过复杂的技术来鼓励动作价值欠估计，但是这又将引入一系列需要调节的超参数。

本文中，提出了一种 Offline RL 算法，从本质上跳过了评估 OOD 动作价值的需要，从而避免了价值函数过估计的问题。作者将此算法命名为：Offline Policy Optimization via Stationary DIstribution Correction Estimation (OptiDICE)。OptilDICE 通过估计平稳分布率，来修正数据分布与最优策略的平稳分布之间的差异。这种最优平稳分布修正可以通过不涉及从目标策略采样的极小极大优化估计。然后，我们推导和探索了上述极小极大优化子问题的 closed-from 解，将整个问题简化为无约束凸优化，从而极大地稳定了 OptilDICE 的优化过程。OptiDICE 是第一个纯粹在平稳分布空间中优化策略

的深度 Offline RL 算法，而不是在 Q 函数或策略空间中优化策略。

2 Background

其他的部分这里不做过多的解释了，主要解释一些关于平稳分布 d^π 的知识，d^π 被定义为，

$$d^\pi(s,a) = \begin{cases} (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t=t, a_t=a) & \text{if} \quad \gamma<1 \\ \lim_{T\to\infty}\frac{1}{T+1}\sum_{t=0}^{\infty}\Pr(s_t=t,a_t=a) & \text{if} \quad \gamma=1 \end{cases} \tag{1}$$

其中，强化学习的学习目标是，$\max_\pi \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)]$。在 offline RL 的设定中，智能体从静态的数据集 $D=\{(s_i,a_i,r_i,s_i')\}_{i=1}^N$ 中优化策略。我们用 $d^{\mathcal{D}}$ 表示数据集的经验分布，为了简单表示，用 $d^{\mathcal{D}}$ 统一表示，$s\sim d^{\mathcal{D}}$，$(s,a)\sim d^{\mathcal{D}}$ 和 $(s,a,s')\sim d^{\mathcal{D}}$。之前的 offline 无模型 RL 算法依靠估计 Q 值来优化目标策略。这个过程通常会产生 Q 值过估计的问题，这是由于从目标策略中采样到的 OOD 动作的 bootstrapped 估计的复合误差造成的（不知道大家好不好理解这句话，可以参考[Offline RL 综述]中，对 Offline RL 中的分布偏移问题有较为详细的分析，通俗的说 OOD 就是 offline RL 中的数据集所代表的策略和正在学习的目标策略不一致。）

3 OptilDICE

不同于传统的 online RL 中的对不确定性保持乐观的态度，在 offline RL 中，我们不鼓励不确定性。因为可能导致数据采集策略无法得到改善，甚至性能严重下降。我们考虑正则化的策略优化框架（下面的过程和 AlgaeDICE 中一样，这里就当复习一下），

$$\pi^* := \arg\max_\pi \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)] - \alpha D_f(d^\pi \| d^{\mathcal{D}}) \tag{2}$$

其中，$D_f(d^\pi\|d^{\mathcal{D}}) := \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[f\left(\frac{d^\pi(s,a)}{d^{\mathcal{D}}(s,a)}\right)\right]$，$d^\pi$ 为目标策略的平稳分布，数据集分布为 $d^{\mathcal{D}}$。其中 $\alpha>0$，为正则化因子。假设，$d^{\mathcal{D}}>0$，并且 f 函数是严格凸的和连续可微的。注意，我们在平稳分布的空间而不是策略的空间增加正则化项。显然，公式 (2) 是不能直接优化的，因为其中涉及到对 d^π 的评估，这个在 offline RL 的设定中不能直接得到。

为了让优化问题可解，将公式 (2) 转换为优化一个平稳分布 $d: S\times A\to\mathbb{R}$。首先考虑为带折扣的 MDP ($\gamma<1$)，然后扩展到无折扣的情况 ($\gamma=1$)，

$$\begin{aligned} &\max_d \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f\left(d\|d^D\right) \\ &\text{s.t.} \quad \left(\mathcal{B}_* d\right)(s) = (1-\gamma)p_0(s) + \gamma\left(\mathcal{T}_* d\right)(s)\ \forall s, \quad d(s,a)\geq 0 \quad \forall s,a \end{aligned} \tag{3}$$

其中，根据 $\pi^*(a|s) = \frac{d^*(s,a)}{\sum_{\bar{a}} d^*(s,\bar{a})}$ 来计算出最优策略。其中，$\mathcal{B}_* d := \sum_{\bar{a}} d(s,\bar{a})$ 表示边缘化算子，$(\mathcal{T}_* d)(s) := \sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a})d(\bar{s},\bar{a})$ 为转置贝尔曼算子；其中公式 (3) 的约束条件为 Bellman flow 约束。

使用拉格朗日算子，可以将公式 (3) 的约束问题转换为，

$$\max_{d\geq 0}\min_\nu \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f\left(d||d^D\right) + \sum_s \nu(s)\left((1-\gamma)p_0(s) + \gamma\left(\mathcal{T}_* d\right)(s) - \left(\mathcal{B}_* d\right)(s)\right) \tag{4}$$

其中，$\nu(s)$ 是拉格朗日乘子。然后，我们通过重新排列公式 (4) 中的项来消除对 d 和 $\mathcal{T}_*$ 的直接依赖（因为感觉对 d 求期望还是有点不太方便，因为 d 本就是要优化的目标，对其求期望并不好计算），最终要优化的是分布率 $\omega = \frac{d(s,a)}{d^\pi(s,a)}$，而不是 d，但是显然有 $d^* = \omega^*$。

$$
\begin{aligned}
&\mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f\left(d \| d^D\right) + \sum_s \nu(s)\left((1-\gamma)p_0(s) + \gamma\left(\mathcal{T}_* d\right)(s) - \left(\mathcal{B}_* d\right)(s)\right) \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[-\alpha f\left(\frac{d(s,a)}{d^D(s,a)}\right)\right] + \sum_{s,a} d(s,a)\underbrace{(R(s,a) + \gamma(\mathcal{T}\nu)(s,a) - (\mathcal{B}\nu)(s,a))}_{=:e_\nu(s,a)(\text{ advantage' using } \nu)} \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[-\alpha f\left(\frac{d(s,a)}{d^D(s,a)}\right)\right] + \mathbb{E}_{(s,a)\sim d^D}[\underbrace{\frac{d(s,a)}{d^D(s,a)}}_{=:w(s,a)}\left(e_\nu(s,a)\right)] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}[-\alpha f(w(s,a))] + \mathbb{E}_{(s,a)\sim d^D}\left[w(s,a)\left(e_\nu(s,a)\right)\right] \\
&= L(w,\nu)
\end{aligned}
\tag{5}
$$

其中，第一个等式用到了 $\beta_*, \mathcal{T}_*$ 的性质，$\mathcal{B}_* d := \sum_{\bar{a}} d(s,\bar{a})$，$(\mathcal{T}_* d)(s) := \sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a}) d(\bar{s},\bar{a})$，代入可以得到，

$$
\begin{aligned}
\sum_s \nu(s)\left(\mathcal{B}_* d\right)(s) &= \sum_{s,a} d(s,a)(\mathcal{B}\nu)(s,a) \\
\sum_s \nu(s)\left(\mathcal{T}_* d\right)(s) &= \sum_{s,a} d(s,a)(\mathcal{T}\nu)(s,a)
\end{aligned}
\tag{6}
$$

其中，$(\mathcal{T}\nu)(s,a) = \sum_{s'} T(s'|s,a)\nu(s'), (\mathcal{B}\nu)(s,a) = \nu(s)$。显然，最终的目标函数 $L(\omega,\nu)$ 不依赖于求关于 d 的期望，只需要求关于 p_0 和 d^D 的期望，而这两个都是已知的，

公式 (5) 中只需要从数据分布 d^D 中采样来完成估计，

$$
\hat{L}(w,\nu) := (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a,s')\sim d^D}\left[-\alpha f(w(s,a)) + w(s,a)\left(\hat{e}_\nu\left(s,a,s'\right)\right)\right] \tag{7}
$$

这里，$\hat{e}_\nu\left(s,a,s'\right) := R(s,a) + \gamma\nu(s') - \nu(s)$。这样来的，将 $\mathcal{T}\nu, \mathcal{B}\nu$ 的关系代入有，$e_\nu(s,a) = R(s,a) + \gamma\mathbb{E}_{s'\sim T(s'|s,a)}[v(s')] - v(s')$，而此处为了简便就用了一次对 s' 的采样。之前的 offline RL 算法中经常含有对从目标策略中采样出的 OOD 动作的估计。显然，optiDICE 中没有这个过程，就不会因为使用了 OOD 动作而在 bootstrapped 过程中有较大的误差。

实际上，OptiDICE 解决了这样一个问题，

$$
\max_{\omega\geq 0}\min_{\nu} L(\omega,\nu) \tag{8}
$$

其中，公式 (8) 的最优解 ω^* 代表最优策略的平稳分布与数据集分布之间的平稳分布修正：$w^*(s,a) = \frac{d^{\pi^*}(s,a)}{d^D(s,a)}$。

3.1 A closed-form solution

在上面已经完整的给出了 optIDICE 的定义，上面这部分进步上是照抄的 AlgaeDICE 中的东西吧。而本篇文章的一个重要创新也是给出了一个 closed-form 的解，不然还真够不上 ICML 的 bar。我也挺好奇这个 closed-form 是何方神圣。

当状态或动作空间，非常大且连续的时候，通常采用函数近似的方法来逼近 ω 和 ν，然后使用关于 L 的梯度来进行优化。然而，这可能破坏优化 L 的良好性质，如 ω 的凹性和 ν 的凸性，导致数值不

稳定和较差的收敛性。实际上就是鞍点优化问题的不稳定性，我在最终看 DICE 的时候就觉得有这个问题了，但是作者好像没考虑这个。本文通过获得内层优化问题的 close-form 解来缓解这个问题，将整个问题简化为无约束凸优化。

由于公式 (3) 中定义的优化问题是一个凸优化问题，通过证明其满足 Slater 条件，可以证明其是强凸的。所以，可以将其 min 和 max 的顺序换一下，

$$\min_{\nu} \max_{\omega \geq 0} L(\omega, \nu) \tag{9}$$

对于任意的 ν，公式 (9) 中内层算子的 closed-form 的解为，

Proposition 1. 公式 (9) 中内层算子的 closed-form 的解为，$\omega_\nu^*(s,a) := \arg\max_{\omega \geq 0} L(\omega, \nu)$，

$$\omega_\nu^*(s,a) := \max\left(0, (f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right)\right) \tag{10}$$

其中，$(f')^{-1}$ 表示为 f 的导数 f' 的逆函数，并且由于 f 是严格凸函数，$(f')^{-1}$ 是单调增的。
Proof.

Lemma 5. 公式 (3) 中定义的优化问题是一个凸优化问题。

$$\begin{aligned}
&\max_d \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f\left(d \| d^D\right) \\
&\text{s.t. } \left(\mathcal{B}_* d\right)(s) = (1-\gamma)p_0(s) + \gamma\left(\mathcal{T}_* d\right)(s)\ \forall s, \quad d(s,a) \geq 0 \quad \forall s, a
\end{aligned}$$

Proof. 首先 $\mathbb{E}_{(s,a)\sim d}[R(s,a)]$ 是关于 d 的线性算子，肯定是凸函数，或者也可以说成是凹函数。而关于 $D_f\left(d \| d^D\right)$，对于任意 $d_1 : S \times A \to \mathbb{R}, d_2 : S \times A \to \mathbb{R}$，

$$\begin{aligned}
D_f\left((1-t)d_1 + t d_2 \| d^D\right) &= \sum_{s,a} d^D(s,a) f\left((1-t)\frac{d_1(s,a)}{d^D(s,a)} + t\frac{d_2(s,a)}{d^D(s,a)}\right) \\
&< \sum_{s,a} d^D(s,a)\left\{(1-t) f\left(\frac{d_1(s,a)}{d^D(s,a)}\right) + t f\left(\frac{d_2(s,a)}{d^D(s,a)}\right)\right\} \\
&= (1-t) D_f\left(d_1 \| d^D\right) + t D_f\left(d_2 \| d^D\right)
\end{aligned}$$

其中，第一行到第二行用到 f 是严格凸的。而关于公式 (3) 中的约束条件，前者是关于 d 的仿射变换，后者的不等式约束是线性的，所以显然是关于 d 的凸函数。所以，本文的问题一定是凸优化问题。

Lemma 6. 假设在给定的 MDP 中所有状态 $s \in S$ 都可达，有

$$\max_{\omega \geq 0} \min_{\nu} L(\omega, \nu) = \min_{\nu} \max_{\omega \geq 0} L(\omega, \nu)$$

Proof. 首先定义约束优化问题的拉格朗日形式，

$$\begin{aligned}
\mathcal{L}(d, \nu, \mu) := \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha D_f\left(d || d^D\right) &+ \sum_s \nu(s)\left((1-\gamma)p_0(s) + \gamma \sum_{\bar{s},\bar{a}} T(s \mid \bar{s}, \bar{a}) d(\bar{s}, \bar{a}) - \sum_{\bar{a}} d(s, \bar{a})\right) \\
&+ \sum_{s,a} \mu(s,a) d(s,a)
\end{aligned}$$

其中，$\nu(s), \forall s$ 和 $\mu(s,a), \forall s, a$ 是拉格朗日乘子，原始问题可以等价的表示为，

$$\max_{d \geq 0} \min_{\nu} L(d, \nu, 0) = \max_{d} \min_{\nu, \mu \geq 0} L(d, \nu, \mu)$$

实际上就是多写了一层而已。而对于所有状态 $s \in S$ 都可达，存在 d 使得 $\forall s, a,\ d(s,a) > 0$，这个应该是客观事实。由 Slater 条件 (存在严格可行 d 的条件) 可知，强对偶性成立，即可以改变优化顺序：

$$\max_{d} \min_{\nu,\mu \geq 0} L(d,\nu,\mu) = \min_{\nu,\mu \geq 0} \max_{d} L(d,\nu,\mu) = \min_{\nu} \max_{d \geq 0} L(d,\nu,0)$$

其中，最后一个等式用到了强凸的性质，对于一个固定的 ν 有 $\max_{d\geq 0} L(d,\nu,0) = \max_d \min_{\mu\geq 0} L(d,\nu,\mu) = \min_{\mu\geq 0} \max_d L(d,\nu,\mu)$。最后将 d 换成我们要优化的变量 $\omega = \frac{d}{d^D}$，有

$$\max_{\omega \geq 0} \min_{\nu} L(\omega,\nu) = \min_{\nu} \max_{\omega \geq 0} L(\omega,\nu)$$

最后，根据上述的推导可以求解出内层最大化算子的解。令 $\max_{\omega\geq 0} L(\omega,\nu)$ 作为原问题，将 $\omega \geq 0$ 用拉格朗日乘子写开，可以得到其对应的对偶问题，

$$\max_{\omega} \min_{\mu \geq 0} L(\omega,\mu) + \sum_{s,a} \mu(s,a)\omega(s,a)$$

由于强对偶性成立，满足 KKT 条件是原始问题和对偶问题解 ω^* 和 μ^* 的充分必要条件，

Condition 1 (Primal feasibility). $\omega^* \geq 0\ \forall s, a$。

Condition 2 (Dual feasibility). $\mu^* \geq 0\ \forall s, a$。

Condition 3 (Stationarity). $d^D(s,a)(-\alpha f'(w^*(s,a)) + e_\nu(s,a) + \mu^*(s,a)) = 0\ \forall s, a$。

Condition 4 (Complementary slackness). $\omega^*(s,a)\mu^*(s,a) = 0\ \forall s, a$。

其中，Stationarity 是上述的对偶问题对 ω 求偏导导出的，其他的条件都比较简单，没什么好说的。从 Stationarity 性质中，可以得到，

$$f'(w^*(s,a)) = \frac{e_\nu(s,a) + \mu^*(s,a)}{\alpha}\ \forall s, a$$

由于 f' 是可逆的，可得，

$$w^*(s,a) = (f')^{-1}\left(\frac{e_\nu(s,a) + \mu^*(s,a)}{\alpha}\right)\ \forall s, a$$

对于固定的 $(s,a) \times S \times A$，考虑两种情况 $\omega^*(s,a) > 0$ 和 $\omega^*(s,a) = 0$。

Case 1 $(\omega^*(s,a) > 0)$，根据 Complementary slackness 条件，有 $\mu(s,a) = 0$ 必然成立，

$$w^*(s,a) = (f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right) > 0$$

由于，f' 是严格增函数（**这是因为严格凸函数的二阶导数恒大于 0，所以其一阶导数恒为增函数**。）

$\frac{e_\nu(s,a)}{\alpha} > f'(0)$，$e_\nu(s,a) > \alpha f'(0)$

Case 2 $(\omega^*(s,a) = 0)$，由 Stationarity 和 Dual feasibility 可以得出，$\mu^*(s,a) = \alpha f'(0) - e_\nu(s,a) \geq 0$。所以有，$e_\nu(s,a) \leq \alpha f'(0)$。

实际上，小编感觉上述的分析纯属凑页数，这些在 KKT 条件里都被定义烂了，实际上就是分析边界条件起作用和不起作用的情况。所以，综上所述，

$$\omega^*_\nu(s,a) = \max\left(0, (f')^{-1}\left(\frac{e_\nu(s,a)}{\alpha}\right)\right)$$

□

进一步研究 (10) 可以发现，对于固定的 ν，具有较大优势 $e_\nu(s,a)$ 的状态-行为的最优平稳分布修正 $w^*_\nu(s,a)$ 较大。当 $\alpha \to 0$ 时，公式 (5) 将回归为成为求解 MDP 的原始线性规划问题的拉格朗日方程，其中 $d(s,a)$ 作为拉格朗日算子来约束 $R(s,a)+\gamma(\mathcal{T}\nu)(s,a) \le \nu(s)$。实际上，$\nu(s)$ 作为优化变量，表示最优状态值函数。这样，$e_{\nu^*}(s,a) = Q^*(s,a) - V^*(s,a)$，表示最优策略的优势函数。并且 f 确定优势与平稳分布修正之间的关系。

将 w_* 的结论融入到公式 (9) 的问题中，可得此优化问题的最新表现形式，

$$\begin{aligned}\min_\nu L\left(w^*_\nu,\nu\right) = (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[-\alpha f\left(\max\left(0,(f')^{-1}\left(\frac{1}{\alpha}e_\nu(s,a)\right)\right)\right)\right] \\ + \mathbb{E}_{(s,a)\sim d^D}\left[\max\left(0,(f')^{-1}\left(\frac{1}{\alpha}e_\nu(s,a)\right)\right)(e_\nu(s,a))\right]\end{aligned} \tag{11}$$

Proposition 2. $L(\omega^*_\nu,\nu)$ 是关于 ν 的凸函数。

Proof. 这个凸函数的证明有点意思，小编第一遍看还有点懵逼。作者这里还给了两种证明方法，第一种是采用拉格朗日对偶来证明的，第二种证明方法则是采用证明二阶导数恒大于 0 和凸函数的基本性质。

Proof by Lagrangian duality. 首先考虑拉格朗日对偶函数，

$$g(\nu,\mu) := \max_d L(d,\nu,\mu)$$

通常来说 L 函数关于 ν,μ 算子一定是凸的，因为是其相关的仿射运算。那么，对于任意的 $\mu_1,\mu_2 \ge 0$，和它们的凸组合 $(1-t)\mu_1 + t\mu_2$，有

$$\min_{\mu\ge 0} g((1-t)\nu_1 + t\nu_2,\mu) \le g((1-t)\nu_1 + t\nu_2,(1-t)\mu_1 + t\mu_2) \le (1-t)g(\nu_1,\mu_1) + tg(\nu_2,\mu_2)$$

同样，利用 $g(\mu,\nu)$ 的凸性，

$$\max_{\mu\ge 0} g((1-t)\nu_1 + t\nu_2,\mu) \le (1-t)\min_{\mu_1\ge 0} g(\nu_1,\mu_1) + t\min_{\mu_2\ge 0} g(\nu_2,\mu_2).$$

所以，显然 $g(\nu,\mu)$ 关于 ν 是凸函数，那么有，对于函数

$$G(\nu) := \min_{\mu\ge 0} g(\nu,\mu) = \min_{\mu\ge 0}\max_d L(d,\nu,\mu) = \max_{d\ge 0} L(d,\nu,0)$$

关于变量 ν 来说，也是凸的。那么，变换变量可以得到，

$$\max_{d\ge 0} L(d,\nu,0) = \max_{\omega\ge 0} L(\omega,\nu) = L(\arg\max_{\omega\ge 0} L(\omega,\nu),\nu)$$

关于 ν 也是凸函数。（**看了好几遍，小编才慢慢的看懂他的证明思路，先证明 $\min_{\mu\ge 0} g(\nu,\mu)$ 是一个凸函数，然后把 $\min_{\mu\ge 0}$ 拿掉，直接留下了 $\min_{d\ge 0} L(d,\nu)$，做一下变量替换得到最终要证的目标**。实际上，这就像是把拉格朗日算子的性质证明抄了一遍，关于拉格朗日算子的函数，肯定是凸函数，这还要证吗？23333）

Proof by exploiting second-order derivative. 首先定义一个函数，

$$h(x) := -f\left(\max\left(0,(f')^{-1}(x)\right)\right) + \max\left(0,(f')^{-1}(x)\right)\cdot x,$$

那么，$L(\omega_\nu^*, \nu)$ 可以表示为，

$$L(\omega_\nu^*, \nu) = (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\left(\frac{1}{\alpha}e_\nu(s,a)\right)\right] \tag{12}$$

$\mathbb{E}_{s\sim p_0}[\nu(s)]$ 不用看肯定是凸的。而证后面那部分是凸函数的关键就是证明 $h(x)$ 是凸函数。其中强调一下，f' 是严格单增的函数，这意味着 f'^{-1} 也是单增的函数。那么，接下来，老老实实的求二阶导数，

$$\begin{aligned}
h(x) &= -f\left(\left(f'\right)^{-1}(x)\right) + \left(f'\right)^{-1}(x)\cdot x \\
h'(x) &= -\underbrace{f'\left(\left(f'\right)^{-1}(x)\right)}_{\text{(identity function)}}\left(\left(f'\right)^{-1}\right)'(x) + \left(\left(f'\right)^{-1}\right)'(x)\cdot x + \left(f'\right)^{-1}(x) \\
&= -x\cdot\left(\left(f'\right)^{-1}\right)'(x) + \left(\left(f'\right)^{-1}\right)'(x)\cdot x + \left(f'\right)^{-1}(x) \\
&= \left(f'\right)^{-1}(x) \\
h''(x) &= \left(\left(f'\right)^{-1}\right)'(x)
\end{aligned}$$

那么，下面的重点即为判断 $\left(\left(f'\right)^{-1}\right)'(x)$ 的正负性。由于，f'^{-1} 是严格单调递增的函数，这不必然有 $((f')^{-1})'(x) > 0$，感觉这里作者将 $(f')^{-1}(x)$ 的情况分成大于 0 和小于 0 去证明，纯属没必要。

□

优化公式 (11) 的凸目标的极小值比嵌套极小极大优化问题更可靠。实际中，我们使用以下目标，可以通过在 D 中的采样来达到优化的目的：

$$\begin{aligned}
\tilde{L}(\nu) :=& (1-\gamma)\mathbb{E}_{s\sim p0}[\nu(s)] + \mathbb{E}_{(s,a,s')\sim d^D}\left[-\alpha f\left(\max\left(0, \left(f'\right)^{-1}\left(\frac{1}{\alpha}\hat{e}_\nu(s,a,s')\right)\right)\right)\right. \\
&\left. + \max\left(0, \left(f'\right)^{-1}\left(\frac{1}{\alpha}\hat{e}_\nu(s,a,s')\right)\right)\left(\hat{e}_\nu(s,a,s')\right)\right]
\end{aligned} \tag{13}$$

但是，由于，$(f')^{-1}$ 是非线性的，并且 $L(\omega_\nu^*)$ 中有 double sampling 问题 **（所谓 Double Sampling 问题简单解释为，从每个状态出发需要独立采集两个 “下一个状态 (s')”，这要求模拟环境能够重置回上一个状态 s，大多数情况比较难实现。）** $\tilde{L}(\nu)$ 并不是一个对 $L(\omega_\nu^*, \nu)$ 的无偏估计。本文中证明了，$\tilde{L}(\nu)$ 是 $L(\omega_\nu^*, \nu)$ 的上界。所以，此最小化问题才比较好优化，可以保证优化 $\tilde{L}(\nu)$ 可以达到优化 $L(\omega_\nu^*, \nu)$ 的效果。小编在这里思考了一下，为什么是上界呢？误差从何而来还挺好想象的，但是上界就想不通了。待我仔细看下证明，或许可以获得一些启示。

Corollary 3. $\tilde{L}(\nu)$ 是公式 (11) 中 $L(\omega_\nu^*, \nu)$ 的上界，即为 $L(\omega_\nu^*, \nu) \le \tilde{L}(\nu)$ 总是成立的，当 MDP 过程确定时，等号成立。

Proof.

$$
\begin{aligned}
L\left(w_{\nu}^{*}, \nu\right) =& (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\left(\frac{1}{\alpha}e_{\nu}(s,a)\right)\right] \\
=& (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}\left[\alpha h\left(\frac{1}{\alpha}\mathbb{E}_{s'\sim T(s,a)}\left[\hat{e}_{\nu}\left(s,a,s'\right)\right]\right)\right] \\
\leq& (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D, s'\sim T(s,a)}\left[\alpha h\left(\frac{1}{\alpha}\hat{e}_{\nu}\left(s,a,s'\right)\right)\right] \quad \text{(by Jensen' s inequality)} \\
=& (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a,s')\sim d^D}\left[-\alpha f\left(\max\left(0,(f')^{-1}\left(\frac{1}{\alpha}\hat{e}_{\nu}\left(s,a,s'\right)\right)\right)\right)\right. \quad \text{(by definition of } h\text{)} \\
&\left. + \max\left(0,(f')^{-1}\left(\frac{1}{\alpha}\hat{e}_{\nu}\left(s,a,s'\right)\right)\right)\left(\hat{e}_{\nu}\left(s,a,s'\right)\right)\right] = \tilde{L}(\nu)
\end{aligned}
$$

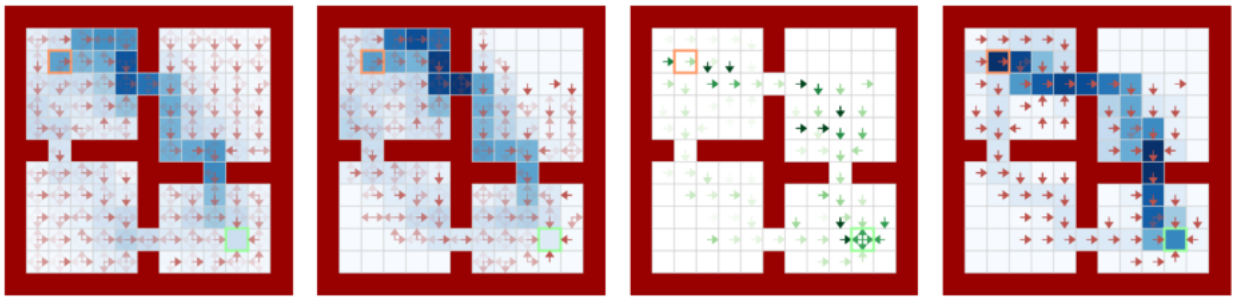
第二行到第三行等号成立的条件是，当状态转移过程 T 确定的时候，

$$
h\left(\frac{1}{\alpha}\mathbb{E}_{s'\sim T(s,a)}[\hat{e}(s,a,s')]\right) = \mathbb{E}_{s'\sim T(s,a)}\left[h\left(\frac{1}{\alpha}\hat{e}(s,a,s')\right)\right]
$$

看完此证明，基本理解了，前文中说的 double-sampling 的困难点了，在第二行涉及到利用 (s,a) 去计算 $\mathbb{E}_{s'\sim T(s,a)}[\cdot]$，这里只能进行一次采样，从 $(s,a)\to s'$，然后要想再从 $(s,a)\to s'$，需要回退到 s 状态就很困难。而这里的不等式关系来自于琴生不等式的放缩。

3.2 A example of finite MDPs

本文中接着，讲述了一个 Four Rooms 的有限状态 MDP 的简单应用，实际是简单的迷宫游戏。如下图所示，



(a) Stationary Dist. d^{π_D} (b) Empirical Dist. d^D (c) OptiDICE w^* (d) Estimated $\hat{d}^{\pi^*}$

图 1: 初始化状态和目标状态分别用橘色和绿色的方框。基于次优的数据收集策略 π_D，并且根据 π_D 可以引导出图 (a) 中的 d^{π_D}。静态数据集被采样，其经验分布为 d^D 为图 (b)。每个方格的不透明度由每个平稳分布的状态分布边际决定，箭头的不透明度表示每个平稳分布诱导的策略的值，箭头颜色越深，代表选择此动作的概率越大。通过求解目标函数，得到 ω^*，可以通过 $\hat{d}^{\pi^*} = d^D(s,a)\omega^*(s,a)$ 来计算最优策略 π^*。

其中，数据收集策略为 $\pi_D = 0.5\pi^*_{\text{true}} + 0.5\pi_{\text{rand}}$，$\pi^*_{\text{true}}$ 是底层 true MDP 的最优策略，π_{rand} 是从 Dirichlet 分布中抽样的随机策略，收集了包含 50 个最大时间步长为 50 的静态数据集 D。再此例子中，基于数据集 D，我们明确构建了最大似然估计 (MLE) MDP。由于是确定的 MDP 过程，直接用公式 (11) 中的表达式可以求解出 ν^*，其中利用 $\hat{M}$ 可以用来准确的计算 $e_{\nu}(s,a)$。然后，用公式 (10)

中 ω^* 的 closed-form，可以准确的求解出 $\omega^*_{\nu^*}$。利用 $\omega^*_{\nu^*}$，可以通过 $\hat{d}^{\pi^*} = d^D(s,a)\omega^*(s,a)$ 来计算最优策略 π^*。在表格的 MDP 中，全局最优解 (ν^*, ω^*) 通常是可以计算出来的。接下来，将较为具体的描述其求解过程。

第一步，我们使用给定的 offline 数据集构建了一个最大后验估计 MDP，$\hat{M} = \langle S, A, T, R, p_0, \gamma \rangle$。然后，我们计算出了数据收集策略 π_D 的平稳分布，并表示为 d^{π_D}。最后，将求解此 MLE MDP 下的策略优化问题，

$$\pi^* := \arg\max_{\pi} \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)] - \alpha D_f(d^\pi \| d^{\pi_D})$$

此问题考虑为关于拉格朗日乘子 ν 的优化平稳分布修正 ω：

$$\min_{\nu} \max_{w \geq 0} L(w,\nu) = (1-\gamma)\mathbb{E}_{s\sim p_0(s)}[\nu(s)] + \mathbb{E}_{(s,a)\sim\mathcal{D}}[-\alpha f(w(s,a)) + w(s,a)(R(s,a) + \gamma(\mathcal{T}\nu)(s,a) - (\mathcal{B}\nu)(s,a))]$$

对于，状态和动作有限的 MDP，MDP 中的所有元素都可以用矩阵描述。$\nu \in \mathbb{R}^{|S|}, \omega \in \mathbb{R}^{|S||A|}, R \in \mathbb{R}^{|S||A|}$，$D$ 是对角矩阵：$D = \mathrm{diag}(d^{\pi_D}) \in \mathbb{R}^{|S||A|\times|S||A|}$，并且 $\mathcal{T} \in \mathbb{R}^{|S||A|\times|S|}$，$\mathcal{B} \in \mathbb{R}^{|S||A|\times|S|}$ 满足以下条件，

$$\mathcal{T}\nu \in \mathbb{R}^{|S||A|} \text{ s.t. } (\mathcal{T}\nu)((s,a)) = \sum_{s'} T(s' \mid s,a)\,\nu(s')$$

$$\mathcal{B}\nu \in \mathbb{R}^{|S||A|} \text{ s.t. } (\mathcal{B}\nu)((s,a)) = \nu(s)$$

为了简单表示，只考虑 $f(x) = \frac{1}{2}(x-1)^2$ 的情况，将上述优化问题用矩阵的形式表示为，

$$\min_{\nu} \max_{\omega \geq 0} L(\omega,\nu) = (1-\gamma)p_0^\top \nu - \frac{\alpha}{2}(\omega-1)^\top D(\omega-1) + \omega^\top D(R + \gamma\mathcal{T}\nu - \mathcal{B}\nu) \tag{14}$$

并且，在 Proposition 1 中，已经得到了内层优化算子的最优解形式，$\omega^*_\nu = \max(0, \frac{1}{\alpha}(R+\gamma\mathcal{T}\nu-\mathcal{B}\nu)+1)$，由于 $(f')^{-1}(x) = x+1$，将 ω^*_ν 融入到 $L(\omega,\nu)$ 中，可以得到，

$$\min_{\nu} L(\omega^*_\nu, \nu) = L(\nu) := (1-\gamma)p_0^\top \nu - \frac{\alpha}{2}(\omega^*_\nu - 1)^\top D(\omega^*_\nu - 1) + {\omega^*_\nu}^\top D(R + \gamma\mathcal{T}\nu - \mathcal{B}\nu) \tag{15}$$

由于 $L(\nu)$ 是关于 ν 的凸函数，所以，可以采用二阶导数的方法来快速优化，比如，Newton 法，

$$\begin{aligned}
e_\nu &:= R + \gamma\mathcal{T}\nu - \mathcal{B}\nu && \text{(advantage using } \nu\text{)}\\
m &:= \mathbb{I}\left(\tfrac{1}{\alpha}e_\nu + 1 \geq 0\right) && \text{(binary masking vector)}\\
w^*_\nu &:= \left(\tfrac{1}{\alpha}e_\nu + 1\right) \odot m \quad \text{(where } \odot m \text{ denotes element-wise masking)} && \text{(closed-form solution)}\\
J &:= \tfrac{\partial w^*_\nu}{\partial \nu} = \tfrac{1}{\alpha}(\gamma\mathcal{T} - \mathcal{B}) \odot m \quad \text{(where } \odot m \text{ denotes row-wise masking)} && \text{(Jacobian matrix)}\\
g &:= \tfrac{\partial L(\nu)}{\partial \nu} = (1-\gamma)p_0 - \alpha J^\top D\,(w^*_\nu - 1) + J^\top D e_\nu + (\gamma\mathcal{T} - \mathcal{B})^\top D w^*_\nu && \text{(first-order derivative)}\\
H &:= \tfrac{\partial^2 L(\nu)}{\partial \nu^2} = -\alpha J^\top D J + J^\top D(\gamma\mathcal{T} - \mathcal{B}) + (\gamma\mathcal{T} - \mathcal{B})^\top D J && \text{(second-order derivative).}
\end{aligned}$$

迭代的更新 v 知道其收敛，梯度方向为 $-H^{-1}g$。最后，使用 $\omega^*_{\nu^*}$ 来表示最优策略 $\pi^*(a|s) \propto \omega^*_{\nu^*}(s,a) \cdot d^{\pi_D}(s,a)$。伪代码如下所示，

Algorithm 2 Tabular OptiDICE ($f(x) = \frac{1}{2}(x-1)^2$)

Input: MLE MDP $\hat{M} = \langle S, A, \mathcal{T}, r, \gamma, p_0 \rangle$, data-collection policy π_D, regularization hyperparameter $\alpha > 0$.

$d^{\pi_D} \leftarrow \text{COMPUTESTATIONARYDISTRIBUTION}(\hat{M}, \pi_D)$

$D \leftarrow \text{diag}(d_D^{\pi})$

$\nu \leftarrow$ (random initialization)

while ν is not converged **do**

$e \leftarrow r + \gamma \mathcal{T} v - \mathcal{B} v$

$m \leftarrow \mathbb{1}\left(\frac{1}{\alpha} e_\nu + 1 \geq 0\right)$

$w \leftarrow \left(\frac{1}{\alpha} e + 1\right) \odot m$

$J \leftarrow \frac{1}{\alpha}\left(\gamma \mathcal{T} - \mathcal{B}\right) \odot m$

$g \leftarrow (1-\gamma) p_0 - \alpha J^\top D(w-1) + J^\top D e + (\gamma \mathcal{T} - \mathcal{B})^\top D w$

$H \leftarrow -\alpha J^\top D J + J^\top D(\gamma \mathcal{T} - \mathcal{B}) + (\gamma \mathcal{T} - \mathcal{B})^\top D J$

$\nu \leftarrow \nu - \eta H^{-1} g$ (where η is a step-size)

end while

$$\pi^*(a|s) \leftarrow \frac{w(s,a) d^{\pi_D}(s,a)}{\sum_{a'} w(s,a') d^{\pi_D}(s,a')} \quad \forall s, a$$

Output: π^*, w

图 2: 表格环境下 OptIDICE 伪代码

3.3 函数近似下的平稳分布修正估计

将 ν 和 ω 参数化为 θ 和 ϕ，并且都用深度神经网络表示。我们的主要目的是优化 θ,

$$\min_\theta J_\nu(\theta) := \min_\theta \tilde{L}(\nu_\theta) \tag{16}$$

当获得最优解 θ^* 之后，需要估计 $w^*(s,a) = \frac{d^{\pi^*}(s,a)}{d^D(s,a)}$，以获得最优策略 π^*。但是，$\omega^*_{\nu_\theta}(s,a)$ 的解析解只能用来估计数据集 D 中存在的 (s,a)，因为需要 $R(s,a)$ 和 $\mathbb{E}_{s' \sim T(s,a)}[\nu_\theta(s')]$ 来估计 e_ν，那么对于数据集中没有的 (s,a)，无法估计其对应的 $e_\nu(s,a)$。（**这里其实也并不难理解，数据集中保持的数据形式为：(s,a,s',r)，对于一个 (s,a)，要想计算 e_ν，需要 s', r。而对于不在数据集中的 (s,a)，显然，我们无法获得 (s',r)**）。所以，我们使用参数化的模型 e_ϕ 来近似优势函数，

$$\omega_\phi(s,a) := \max\left(0, (f')^{-1}\left(\frac{e_\phi(s,a)}{\alpha}\right)\right)$$

当我们从公式 (16) 中得到 ν_θ 时，我们考虑了两种优化 ϕ 的方法。首先，ϕ 可以通过

$$\min_\phi J_\omega(\phi; \theta) := \min_\phi -\hat{L}(\omega_\phi, \nu_\theta) \tag{17}$$

这就相当于退回到了公式 (9) 中的 min-max 问题。第二种就比较粗暴了，即为用训练 ν_θ 导出的 $\hat{e}_{\nu_\theta}$ 直接和 $e_\phi(s,a)$ 做近似，

$$\min_\phi J_\omega^{\text{MSE}}(\phi; \theta) := \min_\phi \mathbb{E}_{(s,a,s') \sim d^D}\left[\left(e_\phi(s,a) - \hat{e}_{\nu_\theta}(s,a,s')\right)^2\right], \tag{18}$$

作者观察到使用 J_ω 或 J_ω^{MSE} 都是有效的，这将在实验中详细说明。在作者的实现中，采用对 θ 和 ϕ 进行联合训练，而不是在 θ 收敛后对 ϕ 进行优化，也就是选择第一种优化方式。（小编觉得这样又回到了之前的 AlgaeDICE 了，作者搞出那么多花里胡哨的，还是写作功底比较强啊，小编赶紧学习学习，hhhh）

3.4 策略提取

最后一步是从最优平稳分布修正 $\omega_\phi(s,a) = \frac{d^{\pi^*}(s,a)}{d^D(s,a)}$ 中提取最优策略 π^*。在表格环境下，使用 $\pi^*(a|s) = \frac{d^D(s,a)\omega_\phi(s,a)}{\sum_{\bar{a}} d^D(s,\bar{a})\omega_\phi(s,\bar{a})}$ 非常简单，但是在连续环境中，并不能直接使用。处理连续域的一种方法是使用重要性加权行为克隆：我们通过最大化从最优策略 π^* 中采样的 (s,a) 的对数似然值来优化参数化策略 π_φ：

$$\min_\varphi \mathbb{E}_{(s,a)\sim d^{\pi^*}}[\log \pi_\varphi(a|s)] = \max_\varphi \mathbb{E}_{(s,a)\sim d^D}[\omega_\phi(s,a)\log \pi_\varphi(a|s)]$$

尽管这样的方法看似很简单，但是实际中使用并不实用（既然不实用，那你说什么呀，，，，而且接下来引出来的方法好像也没什么关系，而且我并没有太看懂怀疑这里是作者的一种写作技巧，可以学习一下下）。

所以，我们使用 information projection（I-projection）来训练策略，

$$\min_\varphi \mathbb{KL}(d^D(s)\pi_\varphi(a|s)\|d^D(s)\pi^*(a|s)) \tag{19}$$

此等式的思路在于最小化 $\pi_\psi(a|s)$ 和 $\pi^*(a|s)$ 有关 π_D 的平稳分布的的距离。这种方法的动机是，策略 π_ψ 应该至少在 D 中存在的状态上进行训练，以便在部署时具有鲁棒性，

$$\begin{aligned}
&\mathbb{KL}\left(d^D(s)\pi_\psi(a \mid s)\|d^D(s)\pi^*(a \mid s)\right) \\
&= -\mathbb{E}_{\substack{s\sim d^D \\ a\sim\pi_\psi(s)}}[\log \underbrace{\frac{d^*(s,a)}{d^D(s,a)}}_{=w_\phi(s,a)} - \log \frac{\pi_\psi(a \mid s)}{\pi_D(a \mid s)} - \underbrace{\log \frac{d^*(s)}{d^D(s)}}_{\text{constant for } \pi}] \\
&= -\mathbb{E}_{\substack{s\sim d^D \\ a\sim\pi_\psi(s)}}[\log w_\phi(s,a) - \mathbb{KL}\left(\pi_\psi(\bar{a} \mid s)\|\pi_D(\bar{a} \mid s)\right)] + C \\
&=: J_\pi\left(\psi;\phi,\pi_D\right)
\end{aligned} \tag{20}$$

我们可以将 I-projection 目标函数看成带 KL 正则化项的框架，其中 $\log\omega_\phi(s,a)$ 为 critic，π_ψ 为 actor。由于，$\pi_D(a|s)$ 是什么也不太清楚，但是这个比较好求，将 $\pi_D(a|s)$ 参数化为 π_β，直接使用简单的行为克隆（BC）来求。

3.5 泛化到 $\gamma = 1$ 的情况

对于 $\gamma = 1$，需要增加了一个条件 $\sum_{s,a} d(s,a) = 1$，至于为什么加这个条件，之前的 reinforcement learning via fenchel Rockafellar 对偶。使用类似的推导方法，可以得到目标函数为，

$$\begin{aligned}
\min_{\nu,\lambda}\max_{w\geq 0} L(w,\nu,\lambda) :=& L(w,\nu) + \lambda\left(1 - \mathbb{E}_{(s,a)\sim d^D}[w(s,a)]\right) \\
=&(1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^D}[-\alpha f(w(s,a))] - \mathbb{E}_{(s,a)\sim d^D}\left[w(s,a)\left(e_\nu(s,a) - \lambda\right)\right] + \lambda
\end{aligned} \tag{21}$$

用类似的方法可以得到关于 $L(w,\nu,\lambda)$ 函数的估计 $\hat{L}(w,\nu,\lambda)$，

$$\hat{L}(\omega,\nu,\lambda) := (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \lambda + \mathbb{E}_{(s,a,s')\sim d^D}[-\alpha f(\omega(s,a)) + \omega(s,a)(\hat{e}_{\nu,\lambda}(s,a,s'))] \tag{22}$$

其中，$\hat{e}_{\nu,\lambda}(s,a,s') = \hat{e}_\nu(s,a,s') - \lambda$。然后，我们就可以推导出内层最大化运算 closed-form 的解，

Proposition 4. 公式 (21) 中内层最优化的最大值 $\omega^*_{\nu,\lambda} : S\times A \to \mathbb{R}$，被定义为：$\omega^*_{\nu,\lambda} := \arg\max_{\omega\geq 0} L(\omega,\nu,\lambda)$，

$$\omega^*_{\nu,\lambda}(s,a) = \max\left(0, (f')^{-1}\left(\frac{e_\nu(s,a) - \lambda}{\alpha}\right)\right) \tag{23}$$

类似于公式 (13) 中的推导，我们最小化一个有偏估计 $\tilde{L}(\nu,\lambda)$，其是 $L(\omega^*_{\nu,\lambda},\nu,\lambda)$ 的上界，这里都很前面是一样的，

$$\begin{aligned}\tilde{L}(\nu,\lambda) := (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a,s')\sim d^D}\Bigg[-\alpha f\left(\max\left(0,(f')^{-1}\left(\frac{1}{\alpha}\hat{e}_{\nu,\lambda}(s,a,s')\right)\right)\right) \\ + \max\left(0,(f')^{-1}\left(\frac{1}{\alpha}\hat{e}_{\nu,\lambda}(s,a,s')\right)\right)\left(\hat{e}_{\nu,\lambda}(s,a,s')\right)\Bigg] + \lambda\end{aligned} \tag{24}$$

使用上述的估计器，对应的更新之前的目标 θ 和 ϕ。首先，关于 θ 的目标函数修改为，

$$\min_\theta J_\nu(\theta,\lambda) := \min_\theta \tilde{L}(\nu_\theta,\lambda)$$

对于 ϕ 的优化，包含拉格朗日 $\lambda' \in \mathbb{R}$:

$$\omega_{\phi,\lambda'}(s,a) := \max\left(0,(f')^{-1}\left(\frac{e_\phi(s,a)-\lambda'}{\alpha}\right)\right)$$

注意 $\lambda' \neq \lambda$ 经常用于稳定学习过程。训练过程几乎一样，

$$\min_\phi J_\omega(\phi,\lambda';\theta) := \min_\phi -\hat{L}(\omega_{\phi,\lambda'},\nu_\theta,\lambda') \tag{25}$$

当时，多增加了一项关于 λ 的优化，这是本节讨论的标准化约束所需要的:

$$\min_\lambda J_\nu(\theta,\lambda) \qquad \text{and} \qquad \min_{\lambda'} J_\omega(\phi,\lambda';\theta),$$

最后，利用上述目标和公式 (20) 中的 BC 目标和策略提取目标，来求解得到我们的算法，OptiDICE，

Algorithm 1 OptiDICE

Input: A dataset $D := \{(s_i,a_i,r_i,s'_i)\}_{i=1}^N$, a set of initial states $D_0 := \{s_{0,i}\}_{i=1}^{N_0}$, neural networks ν_θ and e_ϕ with parameters θ and ϕ, learnable parameters λ and λ', policy networks π_β and π_ψ with parameter β and ψ, a learning rate η

1: **for** each iteration **do**
2: Sample mini-batches from D and D_0, respectively.
3: Compute θ-gradient to optimize (23):
$$g_\theta \approx \nabla_\theta J_\nu(\theta,\lambda)$$
4: Compute ϕ-gradient for either one of objectives:
$$g_\phi \approx \nabla_\phi J_w(\phi,\lambda';\theta) \quad (\textit{minimax obj. (24)})$$
$$g_\phi \approx \nabla_\phi J_w^{\mathrm{MSE}}(\phi;\theta) \quad (\textit{MSE obj. (17)})$$
5: Compute λ and λ' gradients to optimize (25):
$$g_\lambda \approx \nabla_\lambda J_\nu(\theta,\lambda), g_{\lambda'} \approx \nabla_{\lambda'} J_w(\phi,\lambda';\theta)$$
6: Compute β-gradient g_β for BC.
7: Compute ψ-gradient via (19) (*policy extraction*):
$$g_\psi \approx \nabla_\psi J_\pi(\psi;\phi,\pi_\beta)$$
8: Perform SGD updates:
$$\theta \leftarrow \theta - \eta g_\theta, \quad \lambda \leftarrow \lambda - \eta g_\lambda, \quad \beta \leftarrow \beta - \eta g_\beta,$$
$$\phi \leftarrow \phi - \eta g_\phi, \quad \lambda' \leftarrow \lambda' - \eta g_{\lambda'}, \quad \psi \leftarrow \psi - \eta g_\psi.$$
9: **end for**

Output: $\nu_\theta \approx \nu^*, w_{\phi,\lambda'} \approx w^*, \pi_\psi \approx \pi^*,$

图 3: OptiDICE 算法流程图

这里标号有点和我的 notes 中的标号有点不太一样，对着论文中看看就没什么问题了。吸取到教训了，以后写 notes 的时候，尽量把 notes 中的公式标号和文章中的统一一下，方便表示一点。

4 Discussion

当前，DICE 算法中除了 AlgaeDICE 以外，其他的算法都只解决了策略评估或者模仿学习类似的问题，并没有解决策略优化方面的问题。

尽管，algaeDICE 和 OptiDICE 目的都是为了求解 f-divergence 正则化 RL，每个算法求解问题的方式不同。AlgaeDICE 本质上还是先策略评估后策略提升，依赖 off-policy evaluation，利用 DICE 先做策略评估（求解一个 $\min_\nu \max_\omega$ 问题）；然后通过策略提升求策略梯度，最终问题为 $\max_\pi \min_\nu \max_\omega$。实际中，AlgaeDICE 实现使用了额外的近似来进行实际优化，也是为了规避 double-sampling 的问题，利用有偏估计 $\mathbb{E}_{s',a'}[f_*(\cdot)]$ 来估计 $f_*(\mathbb{E}_{s',a'}[\cdot])$，它仍然涉及到嵌套的 $\max_\pi \min_\nu$ 优化，容易受到不稳定性的影响。但是在 OptiDICE 中，则是通过直接优化 ν，利用 ν 来导出平稳分布修正因子 ω，然后利用 ω^* 来还原 π^*，所以作者觉得他们的优化方式更具有稳定性。而作者通过一个简单实验也来证明了这一点，我并没有仔细看。**但是，我看作者的做法，在连续环境下，无法用 ω 直接导出 π，还是要做一步估计。而且当状态空间过大的时候，无法估计 $e(s,a)$，从而无法通过 $e_\nu(s,a)$ 来求得 ω，还是要同时优化 ν,ω，所以，我觉得 optiDICE 只适用于简单环境，其在复杂环境下的情况，有点怀疑。我看实验结果，确实只是在简单环境下比较好，在复杂环境下，表现得没有那么惊艳。**

$$
\begin{aligned}
&\text{AlgaeDICE} \quad \left(e_\nu^\pi(s,a) := r(s,a) + \gamma \mathbb{E}_{s'\sim T(s,a),a'\sim\pi(s')}\left[\nu\left(s',a'\right)\right] - \nu(s,a),\right.\\
&\qquad\qquad \left.\hat{e}_\nu\left(s,a,s',a'\right) := r(s,a) + \gamma\nu\left(s',a'\right) - \nu(s,a)\right)\\
&\max_\pi \min_\nu \max_w \mathbb{E}_{(s,a)\sim d^D}\left[e_\nu^\pi(s,a)w(s,a) - \alpha f(w(s,a))\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0, a_0\sim\pi(s_0)}\left[\nu\left(s_0,a_0\right)\right]\\
&= \max_\pi \min_\nu \alpha \mathbb{E}_{(s,a)\sim d^D}\left[f_*\left(\frac{1}{\alpha}e_\nu^\pi(s,a)\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0, a_0\sim\pi(s_0)}\left[\nu\left(s_0,a_0\right)\right]\\
&\approx \max_\pi \min_\nu \alpha \mathbb{E}_{(s,a,s')\sim d^D, a'\sim\pi(s')}\left[f_*\left(\frac{1}{\alpha}\hat{e}_\nu\left(s,a,s',a'\right)\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0, a_0\sim\pi(s_0)}\left[\nu\left(s_0,a_0\right)\right]\\
&\text{OptiDICE} \quad \left(e_\nu(s,a) := r(s,a) + \gamma\mathbb{E}_{s'\sim T(s,a)}\left[\nu\left(s'\right)\right] - \nu(s), \hat{e}_\nu\left(s,a,s'\right) := r(s,a) + \gamma\nu\left(s'\right) - \nu(s), x_+ := \max(0,x)\right)\\
&\min_\nu \max_{w\geq 0} \mathbb{E}_{(s,a)\sim d^D}\left[e_\nu(s,a)w(s,a) - \alpha f(w(s,a))\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0}\left[\nu\left(s_0\right)\right]\\
&= \min_\nu \mathbb{E}_{(s,a)\sim d^D}\left[e_\nu(s,a)\left(f'\right)^{-1}\left(\frac{1}{\alpha}e_\nu(s,a)\right)_+ - \alpha f\left(\left(f'\right)^{-1}\left(\frac{1}{\alpha}e_\nu(s,a)\right)_+\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0}\left[\nu\left(s_0\right)\right]\\
&\approx \min_\nu \mathbb{E}_{(s,a,s')\sim d'}\left[\hat{e}_\nu\left(s,a,s'\right)\left(f'\right)^{-1}\left(\frac{1}{\alpha}\hat{e}_\nu\left(s,a,s'\right)\right)_+ - \alpha f\left(\left(f'\right)^{-1}\left(\frac{1}{\alpha}\hat{e}_\nu\left(s,a,s'\right)\right)_+\right)\right] + (1-\gamma)\mathbb{E}_{s_0\sim p_0}\left[\nu\left(s_0\right)\right]
\end{aligned}
\tag{26}
$$