Probability Functional Descent: A Unifying Perspective on GANs, Variational Inference, and Reinforcement Learning

Chen Gong

23 September 2021

目录

1 Key idea **1**

2 概率泛函梯度下降 **1**

3 梯度下降 **3**

3.1 近似影响函数 4

4 生成对抗网络 **4**

4.1 最小最大形式下的 GAN 5

4.2 Non-saturating GAN 6

4.3 WGAN 7

5 变分法 **8**

5.1 Black-box 变分推断 8

5.2 对抗变分贝叶斯 9

6 强化学习 **9**

6.1 策略迭代 10

6.2 策略梯度和 AC 12

6.3 Dual AC 12

7 实验 **13**

8 结论 **14**

本文 “Probability Functional Descent: A Unifying Perspective on GANs, Variational Inference, and Reinforcement Learning” 于 2019 年发表于 ICML，是斯坦福大学的工作。这篇文章的思路与我的一篇工作的出发点非常的类似。都是从分布近似的角度分析强化学习问题。不过，机器学习中很多问题可以看成是最小化定义在概率测度空间的泛函，并且 GAN，变分推断和强化学习中的 AC 框架都可以看成是本文提出的框架的特例。

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，本文提出了一种新的框架，从最小化定义在概率测度空间的泛函来统一的解释当前机器学习中的优化问题（GAN，变分推断，AC 框架）。并且，本文给出了其框架通用的优化方法，并将其定义为概率泛函梯度，统一的解释了 GAN，变分推断，AC 框架的优化方法。

1 Key idea

本文主要是提出了一种框架来统一的解释机器学习中的一大类问题。此框架的核心思想是，定义概率度量空间中的一个抽象优化问题，并且，其优化过程是通过最小化概率泛函，其损失函数是将概率分布映射成一个实数。这个其实比较好理解，所谓泛函就是函数的函数，通俗的解释就是通过最小化泛函求解到一个目标的概率分布。而此框架中把机器学习中很多问题都当成求解一个分布。比如，RL 中求策略的分布，VI 中求近似后验，GAN 中求生成器分布。

了解泛函的朋友们应该清楚，函数可以简单的看成无穷维的向量，而无穷维的优化问题是不能简单的套用有限维中的算法，比如梯度下降。于是，接下来则是给出泛函导数的推导过程，所以，作者推导出了一种泛化的梯度方法，称为概率泛函梯度。并且通过概率泛函梯度可以推导出之前的很多算法，其本质的区别只有泛函导数不一样。所以，概率泛函梯度可以用来统一的分析现有的机器学习算法，并引导出新的算法。

2 概率泛函梯度下降

此部分涉及到很多实变函数中的概念，其实感觉不懂也没有太大关系。假设 $\mathcal{P}(X)$ 为拓扑空间 X 上的 Borel 概率测度空间。本文提出的概率密度的最小化问题可以写为，

$$\min_{\mu \in \mathcal{P}(X)} J(\mu) \tag{1}$$

其中，$J : \mathcal{P}(X) \to \mathbb{R}$ 被称为概率泛函。此公式也挺好理解的，我们想找到一个分布来使目标函数 J 最小，而 J 往往被定义为一种概率距离。我们现在利用 von Mises 微积分来精确地定义泛函的导数。这里不熟悉的朋友有参考一下加托微分的基本概念。

Definition 1 (Gateaux differential) 令 $J : \mathcal{P}(X) \to \mathbb{R}$ 为一个函数。而 J 函数关于 μ 在 $\mathcal{X}$ 的方向上的 Gateaux differential 被定义为，

$$dJ_\mu(\chi) = \lim_{\epsilon \to 0^+} \frac{J(\mu + \epsilon\chi) - J(\mu)}{\epsilon} \tag{2}$$

其中，$\mathcal{X} = \nu - \mu, \nu \in \mathcal{P}(X)$.

直觉告诉我们，Gateaux differential 是方向导数的推广，所以，$dJ_\mu(\mathcal{X})$ 表示为，概率测度 μ 在 $\mathcal{X}$ 方向向另一个测度 ν 受到无限小扰动时，$J(\mu)$ 值的变化。按公式 (2) 这样来，不太合适实际计算。而实际上，Gateaux differential $dJ_\mu(\mathcal{X})$ 可以被简洁地表示为一个影响函数的积分 $\Psi_\mu : X \to \mathbb{R}$，于是可以得到了影响函数的定义，

Definition 2 (Influence function). $\Psi_\mu : X \to \mathbb{R}$ 的定义满足，对于 J 函数在 μ 处的 Gateaux differential 可以积分的形式，

$$dJ_\mu(\chi) = \int_X \Psi_\mu(x)\chi(dx) \tag{3}$$

对于任意的 $\chi = \nu - \mu$，其中 $\mu, nu \in \mathcal{P}(X)$。

其中 χ 表示为两个分布之间的差距，将 $\chi = \nu - \mu$ 代入可以得到，

$$dJ_\mu(\chi) = \mathbb{E}_{x\sim\nu}[\Psi_\mu(x)] - \mathbb{E}_{x\sim\mu}[\Psi_\mu(x)]. \tag{4}$$

注意到如果 $\Psi_\mu(x)$ 是一个影响函数，那么 $\Psi_\mu(x) + \text{const}$ 也是。因为 χ 表示所有方向，所有方向求和就会相互抵消掉，所以有 $\int d\chi = 0$。这里用到了泛函导数的展开式，和泰勒公式有点像。将 μ_0 附近展开 $J(\mu)$，

$$\begin{aligned} \tilde{J}(\mu) &= J(\mu_0) + dJ_{\mu_0}(\mu - \mu_0) \\ &= J(\mu_0) + \mathbb{E}_{x\sim\mu}[\Psi_{\mu_0}(x)] - \mathbb{E}_{x\sim\mu_0}[\Psi_{\mu_0}(x)] \\ &= \text{constant} + \mathbb{E}_{x\sim\mu}[\Psi_{\mu_0}(x)] \end{aligned} \tag{5}$$

其中，$J(\mu_0) - \mathbb{E}_{x\sim\mu_0}[\Psi_{\mu_0}(x)]$ 是固定值。所以，我们发现对于 μ 有较小的扰动，而导致 $J(\mu)$ 的值变小，实际上是使 $\mathbb{E}_{x\sim\mu}[\Psi_{\mu_0}(x)]$ 的值变小。所以，优化 $J(\mu)$ 可以近似的等价于优化 $\mathbb{E}_{x\sim\mu}[\Psi_{\mu_0}(x)]$，这里有变分下界那味儿了。

基于这个理论，我们提出了概率泛函下降，此方法用类似于有限维空间中的一阶梯度下降的方法来计算概率泛函的梯度。首先，以影响函数 Ψ_{μ_0} 的形式在 μ_0 处计算函数 J 的线性近似，然后从 μ_0 处取一个局部步长，来减小线性近似的值。从而实现概率泛函下降的效果。简单的说就是下列伪代码：

Algorithm 1 Probability functional descent on $J(\mu)$

Initialize μ to a distribution in $\mathcal{P}(X)$
while μ has not converged **do**
  Set $\hat{\Psi} \approx \Psi_\mu$ (differentiation step)
  Update μ to decrease $\mathbb{E}_{x\sim\mu}[\hat{\Psi}(x)]$ (descent step)
end while

图 1: 概率泛函下降

实际上概率泛函下降作为许多现有算法的基础，在 GAN 网络中，differentiation 和 descent 步骤分别对应于的鉴别器和生成器更新; 在强化学习中，它们对应于策略评估和策略提升。接下来一个重要的问题是，解决如何进行梯度下降的问题。

3 梯度下降

首先将 μ 分布参数化，$\theta \mapsto \mu_\theta$，然后对 $\theta \mapsto \mathbb{E}_{x\sim\mu_\theta}[\hat{\Psi}(x)]$ 使用随机梯度下降，这里作者给出了证明，但是实际上我感觉观察公式 (5) 就很明显了。

Theorem 1 (Chain rule). 令 $J : \mathcal{P}(X) \to \mathbb{R}$ 是连续可微的，那么其影响函数 Ψ_μ 也是存在且 $(\mu, \nu) \mapsto \mathbb{E}_{x\sim\nu}[\Psi_\mu(x)]$ 也是连续的。令 $\theta \mapsto \mu_\theta$ 为可微的，那么，当 $h \to 0$ 时，$\frac{1}{\|h\|}(\mu_{\theta+h} - \mu_\theta)$ 将会收敛，

$$\nabla_\theta J(\mu_\theta) = \nabla_\theta \mathbb{E}_{x\sim\mu_\theta}[\hat{\Psi}(x)] \tag{6}$$

其中，$\hat{\Psi} = \Psi_{\mu_\theta}$ 作为一个不依赖于 θ 的函数：$X \to \mathbb{R}$.

Proof. 令 $\chi_\epsilon = \frac{1}{\epsilon}(\mu_{\theta+h}) - \mu_\theta$，并且令 $\chi = \lim_{\epsilon\to 0} \chi_\epsilon$，有

$$\begin{aligned}
\frac{d}{d\theta} J\left(\mu_\theta\right) &= \left.\frac{d}{d\epsilon} J\left(\mu_{\theta+\epsilon}\right)\right|_{\epsilon=0} \\
&= \left.\frac{d}{d\epsilon} J\left(\mu_\theta + \epsilon\chi_\epsilon\right)\right|_{\epsilon=0} \\
&= \left.\frac{d}{d\epsilon} J\left(\mu_\theta + \epsilon\chi\right)\right|_{\epsilon=0}
\end{aligned} \tag{7}$$

根据公式 (2) 和公式 (3)，易得，

$$\left.\frac{d}{d\epsilon} J(\mu + \epsilon\chi)\right|_{\epsilon=0^+} = \int_X \Psi(x)\chi(dx) \tag{8}$$

基于此，可以得到，

$$\begin{aligned}
\frac{d}{d\theta} J\left(\mu_\theta\right) &= \left.\frac{d}{d\epsilon} J\left(\mu_\theta + \epsilon\chi\right)\right|_{\epsilon=0} \\
&= \int_X \Psi(x)\chi(dx) \\
&= \int_X \hat{\Psi} d\chi \\
&= \int_X \hat{\Psi} d\left(\lim_{\epsilon\to 0} \frac{1}{\epsilon}\left(\mu_{\theta+\epsilon} - \mu_\theta\right)\right) \\
&= \lim_{\epsilon\to 0} \int_X \hat{\Psi} d\left(\frac{1}{\epsilon}\left(\mu_{\theta+\epsilon} - \mu_\theta\right)\right) \\
&= \frac{d}{d\theta} \int_X \hat{\Psi} d\mu_\theta
\end{aligned} \tag{9}$$

χ 本来就是加托导数里面的微分量，可以放进后面，本就是加托导数里面的方向向量，然后整个积分变成了一个向量积分。。由于 $\nabla_\theta J(\mu_\theta)$ 中 J 可能是一个非常复杂的非线性函数，其梯度很难计算。而把梯度的计算转换为一个期望问题，则有较好的解决方法了（参考强化学习中的策略梯度），关于求解公式 (6)，可以用到对数似然梯度估计，

$$\nabla_\theta \mathbb{E}_{x\sim\mu_\theta}[\hat{\Psi}(x)] = \mathbb{E}_{x\sim\mu_\theta}\left[\hat{\Psi}(x)\nabla_\theta \log \mu_\theta(x)\right] \tag{10}$$

3.1 近似影响函数

实际上很多算法看上去有较大的区别，都是由于影响函数的不同形式造成的。在某些情况下，通过近似，影响函数可以被估计的很准确。通常影响函数用一个神经网络来近似：$X \to \mathbb{R}$。但是，要想知道神经网络如何训练需要准确的分析影响函数。

那么，如果泛函 J 是凸的，那么可以推导出一种通用的近似方法。这里使用凸共轭的方法来统一解释，首先令 $\mathcal{P}(X)$ 为 Borel 测度空间的凸子集，$C(X)$ 为对偶空间，也是一个连续函数空间：$X \to \mathbb{R}$。最后，$\overline{\mathbb{R}}$ 表示为扩展的实值空间：$\mathbb{R} \cup \{-\infty, \infty\}$，凸共轭被定义为（这里的凸共轭的定义方法，逼格比较高，可以学习一下），

Definition 3. 令 $J : \mathcal{M}(X) \to \overline{\mathbb{R}}$，其凸共轭表示为，$J^* : \mathcal{C}(X) \to \overline{\mathbb{R}}$，

$$J^\star(\varphi) = \sup_{\mu \in \mathcal{M}(X)} \left[\int_X \varphi(x)\mu(dx) - J(\mu) \right] \tag{11}$$

凸共轭构成下面影响函数 Ψ 的表示的核心，

Theorem 2 (Fenchel-Moreau representation) 令 $J : \mathcal{M}(X) \to \overline{\mathbb{R}}$ 为适定的，凸的，下半连续函数。那么，如果最大化算子 $\varphi \mapsto \mathbb{E}_{x\sim\mu}[\varphi(x)] - J^*(\varphi)$ 存在，那么其为 J 函数在 μ 的影响函数，可得，

$$\Psi_\mu = \arg\max_{\varphi \in \mathcal{C}(X)} \left[\mathbb{E}_{x\sim\mu}[\varphi(x)] - J^\star(\varphi) \right] \tag{12}$$

那么，这样就实现了对影响函数的估计。

Proof. 下面将给出证明，最大化 φ 的结果就是影响函数，

$$J(\mu) = J^{\star\star}(\mu) = \sup_{\varphi \in \mathcal{C}(X)} \left[\int_X \varphi d\mu - J^\star(\varphi) \right] \tag{13}$$

并且，

$$J(\mu + \epsilon\chi) = \sup_{\varphi \in \mathcal{C}(X)} \left[\int_X \varphi d\mu + \epsilon \int_X \varphi d\chi - J^\star(\varphi) \right] \tag{14}$$

根据 Theorem 2 的结论，给出了计算影响函数的较好的方法，将模型 $\varphi : X \to \mathbb{R}$ 用参数 ϕ 表示。首先要使用随机梯度下降来优化 ϕ，其目标函数是令 $\mathbb{E}_{x\sim\mu}[\varphi(x)] - J^*(\varphi)$ 最大化。求得的 φ^* 即为影响函数，然后利用公式 (10) 来求对于 μ 的梯度就可以求解了。那么，此 PFD 可以被表达为一个鞍点优化问题，

$$\inf_{\mu} \sup_{\varphi} \left[\mathbb{E}_{x\sim\mu}[\varphi(x)] - J^\star(\varphi) \right] \tag{15}$$

其中，$J^\star(\varphi)$ 是 w.r.tμ 的常数。接下来将说明 GAN，VI 和 RL 中的 AC 框架都是 PFD 的一种特例。

4 生成对抗网络

GAN 使用参数化的概率分布 μ 来模拟一个数据分布 ν。实际上，GAN 的理论中已经清楚的描述了，GAN 的工作就是在最小化 $D(\mu\|\nu)$ 中各种不同的散度。许多 GAN 的变体可以看成 PFD 应用于不同散度。

4.1 最小最大形式下的 GAN

GAN 函数的最初表达式为，

$$\inf_{\mu}\sup_{D}\frac{1}{2}\mathbb{E}_{x\sim\nu}[\log D(x)]+\frac{1}{2}\mathbb{E}_{x\sim\mu}[\log(1-D(x))] \tag{16}$$

关于这个等式的解释，很多地方都有了，而且我相信看这篇文章的朋友，对 GAN 的理论有一定的认识，这里就不做过多的解释了。实际上，内层优化问题等价于优化 $D_{\mathrm{JS}}(\mu\|\nu)-\log 2$。实际算法中，令 ϕ 表示为判别器的参数，θ 表示为生成器的参数，而采用随机梯度下降主要优化下列两个损失函数，

$$\begin{cases}\phi\mapsto -\frac{1}{2}\mathbb{E}_{x\sim\nu}\left[\log D_{\phi}(x)\right]-\frac{1}{2}\mathbb{E}_{x\sim\mu_{\theta}}\left[\log\left(1-D_{\phi}(x)\right)\right]\\ \theta\mapsto \frac{1}{2}\mathbb{E}_{x\sim\mu_{\theta}}\left[\log\left(1-D_{\phi}(x)\right)\right]\end{cases} \tag{17}$$

实际上第一行描述的是近似影响函数的过程，而第二行描述的是更新目标函数的过程。下面将用 PFD 的观点来统一分析，

Proposition 1. PFD 关于 JS 散的的目标为，

$$J_{\mathrm{JS}}(\mu)=D_{\mathrm{JS}}(\mu\|\nu) \tag{18}$$

首先要知道，判别器实际上扮演着近似影响函数的作用，

Proposition 2. 假设 μ 为分布 $p(x)$，ν 为分布 $q(x)$，则 JS 散度的影响函数为，

$$\Psi_{\mathrm{JS}}(x)=\frac{1}{2}\log\frac{p(x)}{p(x)+q(x)} \tag{19}$$

Proof. 其证明思路也挺简单的，即为求泛函导数，

$$\begin{aligned}&\frac{d}{d\epsilon}J_{\mathrm{JS}}(\mu+\epsilon\chi)\Big|_{\epsilon=0}\\ &=\frac{1}{2}\int_X\frac{d}{d\epsilon}\left[(p+\epsilon\chi)\log\frac{p+\epsilon\chi}{\frac{1}{2}(p+\epsilon\chi)+\frac{1}{2}q}+q\log\frac{q}{\frac{1}{2}(p+\epsilon\chi)+\frac{1}{2}q}\right]_{\epsilon=0}dx\\ &=\frac{1}{2}\int_X\left[\log\frac{p}{\frac{1}{2}p+\frac{1}{2}q}+1-\frac{p}{p+q}-\frac{q}{p+q}\right]\chi dx\\ &=\frac{1}{2}\int_X\left[\log\frac{p}{p+q}+\log 2\right]\chi dx\end{aligned} \tag{20}$$

其中，第二行到第三行就是对 ϵ 求导，然后把 $\epsilon=0$ 带进去。而在 Definition 2 中，我们提到了，常数项可以省略，所以影响函数即为：$\frac{1}{2}\log\frac{p}{p+q}$。在 GAN 的基础理论中已经推导出，最优分类器满足：$D^*(x)=\frac{q(x)}{p(x)+q(x)}$。所以，影响函数可以用学到的判别器来表示，$\Psi_{\mathrm{JS}}(x)=\frac{1}{2}\log(1-D^*(x))$，实际上，这里充分说明了影响函数是由优化判别器得到的。但是公式 (19) 中的数据分布的具体形式是不知道的，所以需要用下面描述的方法来近似，

Proposition 3. J_{JS} 的凸共轭函数为，

$$J^{\star}_{\mathrm{JS}}(\varphi)=-\frac{1}{2}\mathbb{E}_{x\sim\nu}\left[\log\left(1-e^{2\varphi(x)+\log 2}\right)\right]-\frac{1}{2}\log 2 \tag{21}$$

Proof.

$$\begin{aligned}J^{\star}_{\mathrm{JS}}(\varphi)&=\sup_{\mu\in\mathcal{M}(X)}\left[\int_X\varphi d\mu-J_{\mathrm{JS}}(\mu)\right]\\ &=\sup_{p}\int_X\left[\varphi p-\frac{1}{2}p\log\frac{p}{\frac{1}{2}p+\frac{1}{2}q}-\frac{1}{2}q\log\frac{q}{\frac{1}{2}p+\frac{1}{2}q}\right]dx\end{aligned} \tag{22}$$

类似的对 p 求导，并令其等于 0，可得，

$$\varphi = \frac{1}{2}\log\frac{p}{\frac{1}{2}p+\frac{1}{2}q}$$

替换一下有，公式 (22) 中的第一项和第二项直接抵消掉了，留下第三项，

$$\frac{q}{\frac{1}{2}p+\frac{1}{2}q} = 2(1-2e^{2\varphi})$$

所以有，

$$J^{\star}_{\mathrm{JS}}(\varphi) = -\frac{1}{2}\int_X q\log\left(1-2e^{2\varphi}\right)dx - \frac{1}{2}\log 2 \tag{23}$$

那么，根据 Theorem 2，我们得到了 Ψ_{JS} 的优化方式，通过求解下列问题来近似公式 (19) 中的结果，

$$\Psi_{\mathrm{JS}} = \arg\max_{\varphi\in\mathcal{C}(X)}\left[\mathbb{E}_{x\sim\mu}[\varphi(x)] + \frac{1}{2}\mathbb{E}_{x\sim\nu}\left[\log\left(1-e^{2\varphi(x)+\log 2}\right)\right]\right], \tag{24}$$

那么可以先对公式 (24) 代替公式 (17) 中的第一行的表达式优化，然后对 $\mathbb{E}_{x\sim\mu}[\varphi(x)]$ 进行优化。

4.2 Non-saturating GAN

在 GAN 的原文中对公式 (17) 做了一些小的改变，主要是针对目标函数 μ 的优化部分，

$$\begin{cases} \phi \mapsto -\frac{1}{2}\mathbb{E}_{x\sim\nu}\left[\log D_\phi(x)\right] - \frac{1}{2}\mathbb{E}_{x\sim\mu_\theta}\left[\log\left(1-D_\phi(x)\right)\right] \\ \theta \mapsto -\frac{1}{2}\mathbb{E}_{x\sim\mu_\theta}\left[\log D_\phi(x)\right] \end{cases} \tag{25}$$

其中将对 $\log(1-D_\phi)$ 的优化，修改成了对 $-\log D_\phi$ 的优化。这种启发式修改是为了防止当鉴别器过于自信时，θ 的梯度收敛到 0，因此 θ 的损失称为非饱和损失。

本文中，考虑将对 θ 更新的原损失函数和非饱和损失合并起来，得到，

$$\begin{cases} \phi \mapsto -\frac{1}{2}\mathbb{E}_{x\sim\nu}\left[\log D_\phi(x)\right] - \frac{1}{2}\mathbb{E}_{x\sim\mu_\theta}\left[\log\left(1-D_\phi(x)\right)\right] \\ \theta \mapsto -\mathbb{E}_{x\sim\mu_\theta}\left[\log D_\phi(x)\right] + \mathbb{E}_{x\sim\mu_\theta}\left[\log\left(1-D_\phi(x)\right)\right] \end{cases} \tag{26}$$

本文做出的修改仍然可以和 non-saturation GAN 实现相同的效果，并且此过程可以看成最小化 $D_{\mathrm{KL}}(\mu\|\nu)$。

Proposition 4. PFD 定义下的 KL 散度目标为，

$$J_{\mathrm{NS}}(\mu) = D_{\mathrm{KL}}(\mu\|\nu) \tag{27}$$

Proposition 5. 令 μ 为 $p(x)$，ν 为 $q(x)$，J_{NS} 的影响函数被定义为，

$$\Psi_{\mathrm{NS}}(x) = \log\frac{p(x)}{q(x)} \tag{28}$$

Proof. 这个证明过程和前面的是一样的，

$$\begin{aligned}
&\frac{d}{d\epsilon}J_{\mathrm{NS}}(\mu+\epsilon\chi)\Big|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\int_X (p+\epsilon\chi)\log\frac{p+\epsilon\chi}{q}dx\Big|_{\epsilon=0} \\
&= \int_X \left[\chi\log\frac{p}{q}+\chi\right]dx \\
&= \int_X \left[\log\frac{p}{q}+1\right]d\chi \\
&= \int_X \left[\log\frac{p}{q}\right]d\chi
\end{aligned} \tag{29}$$

这里有点小伙伴可能会有一点困惑，这里的 χdx 怎么一下就变成了 $d\chi$，χ 本来就是加托导数里面的微分量，可以放进后面，本就是加托导数里面的方向向量，然后整个积分变成了一个向量积分。原来是逐元素积，现在变成积向量。其中，$D(x)$ 的最优解是 $D(x)=\frac{q(x)}{p(x)+q(x)}$，将此结果代入到公式 (28) 中，可得，

$$\Psi_{\mathrm{NS}}(x) \approx \log\frac{1-D_\phi(x)}{D_\phi(x)} \tag{30}$$

这也是为什么要将公式 (25) 改写成公式 (26) 的原因。所以，这么就可以将 non-saturation GAN 和 PFD 统一在一起分析了。

4.3 WGAN

WGAN 的目标形式为，

$$\inf_{\mu}\sup_{\|D\|_L\le 1}\left[\mathbb{E}_{x\sim\mu}[D(x)]-\mathbb{E}_{x\sim\nu}[D(x)]\right] \tag{31}$$

其中 $\|D\|_L$ 表示李普希斯约束，实际中参数更新方式为，

$$\begin{cases}
\phi \mapsto \mathbb{E}_{x\sim\mu_\theta}\left[D_\phi(x)\right]-\mathbb{E}_{x\sim\nu}\left[D_\phi(x)\right] \\
\theta \mapsto -\mathbb{E}_{x\sim\mu_\theta}\left[D_\phi(x)\right]
\end{cases} \tag{32}$$

有关 WGAN 的具体内容，这里不做过多的解释了，实际上 WGAN 也是 PFD 的一个特例，判别器起到了近似影响函数的作用，

Proposition 6. WGAN 的目标为，

$$J_W(\mu)=W_1(\mu,\nu) \tag{33}$$

实际上 WGAN 同样也可通过凸共轭推导出来，

Proposition 8. J_W 的凸共轭函数为，

$$J_W^*(\varphi)=\mathbb{E}_{x\sim\nu}[\varphi(x)]+\{\|\varphi\|\le 1\} \tag{34}$$

其中 $\{A\}$ 是指示函数，指示函数也是凸函数，其表示的意思为，当 A 为真时值为 0，否则为 ∞。

Proof. 在 WGAN 的原文中，根据 Kantorovich Rubinstein 对偶，WGAN 经常被写为，

$$\begin{aligned} J_{\mathrm{W}}(\mu) &= \sup_{\|\varphi\|_L \leq 1} \left[\int_X \varphi d\mu - \int_X \varphi d\nu \right] \\ &= \sup_{\varphi} \left[\int_X \varphi d\mu - \int_X \varphi d\nu - \{\|\varphi\|_L \leq 1\} \right] \end{aligned} \tag{35}$$

其中，第一行到第二行用的是冲击函数的性质，通过定义冲击函数可以将带约束的优化问题，转换为无约束的优化问题，大家想想就知道当 φ 不满足条件是，值为 $-\infty$，所以肯定取不到。而，

$$\begin{aligned} J_{\mathrm{W}}(\mu) &= \sup_{\varphi} \left[\int_X \varphi d\mu - J^*_{\mathrm{W}}(\mu) \right] \\ &= \sup_{\varphi} \left[\int_X \varphi d\mu - \int_X \varphi d\nu - \{\|\varphi\|_L \leq 1\} \right] \end{aligned}$$

所以，

$$J^*_W(\varphi) = \mathbb{E}_{x\sim\nu}[\varphi(x)] + \{\|\varphi\| \leq 1\}$$

那么，影响函数被定义为，

$$\Psi_{\mathrm{W}} = \underset{\varphi \in \mathcal{C}(X)}{\arg\max} \left[\mathbb{E}_{x\sim\mu_\theta}[\varphi(x)] - \mathbb{E}_{x\sim\nu}[\varphi(x)] - \{\|\varphi\|_L \leq 1\} \right]. \tag{36}$$

5 变分法

在贝叶斯推断里，我们经常需要求后验分布，

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)} = \frac{p(x \mid z)p(z)}{\int p(x \mid z)p(z)dz} \tag{37}$$

其中，$p(z)$ 为先验，$p(x|z)$ 为似然；但是分子的归一化因子非常难以计算，变分法提供了一种计算方法，用一个变分后验 $q(z)$ 来近似真实后验，那么变分问题被定义为，

$$\inf_q D_{\mathrm{KL}}(q(z) \| p(z|x))$$

5.1 Black-box 变分推断

直接优化比较困难，因为 $p(z|x)$ 根本就无法计算，而 VI 中采用优化变分下界（ELBO）的方法来优化，

$$D_{\mathrm{KL}}(q(z) \| p(z \mid x)) = \log p(x) - \underbrace{\mathbb{E}_{z\sim q(z)} \left[\log \frac{p(x \mid z)p(z)}{q(z)} \right]}_{\mathrm{ELBO}} \tag{38}$$

由于 $\log p(x)$ 是固定的，所以通过最大化 ELBO 就可以达到最小化 KL 散度的效果。使用蒙特卡罗采样和随机梯度下降就可以优化此目标，

$$\theta \mapsto -\mathbb{E}_{z\sim q_\theta(z)} \left[\log \frac{p(x \mid z)p(z)}{q_\theta(z)} \right] \tag{39}$$

Roeder 发现在求导的时候，将期望项中的 $q_\theta(z)$ 看成和 θ 无关的项，也可以计算出相同的梯度。所以，我们可以将 VI 统一到 PFD 中，

Proposition 9. 变分推断在 PFD 中目标函数为，

$$J_{\mathrm{VI}}(q) = D_{\mathrm{KL}}(q(z) \| p(z|x))$$

事实上，影响函数恰好等于 ELBO 期望内部的函数，

Proposition 10. J_{VI} 的影响函数为，

$$\Psi_{\mathrm{VI}}(z) = \log \frac{q(z)}{p(x \mid z)p(z)}$$

Proof.

$$\begin{aligned}
\frac{d}{d\epsilon} J_{\mathrm{VI}}(q + \epsilon\chi)\bigg|_{\epsilon=0} &= \frac{d}{d\epsilon} \int (q(z) + \epsilon\chi(z)) \log \frac{q(z) + \epsilon\chi(z)}{p(z \mid x)} dz \bigg|_{\epsilon=0} \\
&= \int \left[\chi(z) \log \frac{q(z) + \epsilon\chi(z)}{p(z \mid x)} + \chi(z) \right] dz \bigg|_{\epsilon=0} \\
&= \int \left[\log \frac{q(z)}{p(z \mid x)} + 1 \right] \chi(z) dz \\
&= \int \left[\log \frac{q(z)}{p(x \mid z)p(z)} + \log p(x) + 1 \right] \chi(z) dz \\
&= \int \log \frac{q(z)}{p(x \mid z)p(z)} \chi(z) dz
\end{aligned}$$

这里的推导和前面的基本是一样的，没什么可多说的。在这种情况下，可以准确地评估影响函数，因此可以在不近似的情况下执行 PFD 的微分步骤。

5.2 对抗变分贝叶斯

当先验 $p(z)$ 或者变分后验 $q(z|x)$ 未知时，对抗变分贝叶斯或许可以派上用场了。其中，用一个神经网络啦近似 $\log \frac{q(z)}{p(z)}$。那么求解过程可以写为，

$$\begin{cases} \phi \mapsto -\mathbb{E}_{q_\theta(z)} \left[\log \sigma \left(f_\phi(z) \right) \right] - \mathbb{E}_{p(z)} \left[\log \left(1 - \sigma \left(f_\phi(z) \right) \right) \right] \\ \theta \mapsto -\mathbb{E}_{q_\theta(z)} \left[-f_\phi(z) + \log p(x \mid z) \right] \end{cases} \tag{40}$$

类似的可以推导出其影响函数为，

$$\Psi_{\mathrm{VI}}(z) = \log \frac{q(z)}{p(x \mid z)p(z)} \approx f_\phi(z) - \log p(x \mid z)$$

6 强化学习

在 Markov 决策过程中，状态分布 $s = (s_0, s_1, \cdots)$，动作分布 $a = (a_1, a_2, \cdots)$，和奖励分布 $r = (r_1, r_2, \cdots)$ 取决于分布

$$\mathbb{P}(s, a, r) = p_0 \left(s_0 \right) \prod_{t=1}^{\infty} p \left(s_t, r_t \mid s_{t-1}, a_t \right) \pi \left(a_t \mid s_{t-1} \right)$$

其中，$p_0(s)$ 表示初始分布，$p(s', r \mid s, a)$ 表示给定一个动作状态对 (s, a)，转移到状态 s' 并获得奖励为 r 的概率分布。

6.1 策略迭代

策略迭代基本可以分成两步策略评估和策略提升。策略评估阶段，计算状态价值函数 $Q^\pi(s,a)$ 的值；策略提升阶段，策略使用 greedy 进行更新，其中动作选择采用 $\arg\max_a Q^\pi(s,a)$。

在给出统一结果之前，引入 $\pi(s)$ 表示关于状态的任意分布，并考虑 $\pi(s,a)=\pi(s)\pi(a|s)$ 的联合分布，使 π 是一个概率分布，而不是关于某个状态 s 的概率（这里文章说得有点迷惑），

Proposition 12. 强化学习在 PFD 中的定义为，

$$J_{\mathrm{RL}}(\pi)=-\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r_t\right], \tag{41}$$

利用精确的影响函数和线性逼近的全局最小化，得到策略迭代算法。

Proposition 13. J_{RL} 的影响函数定义为，

$$\Psi_{\mathrm{RL}}(s,a)=-\frac{\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s)}{\pi(s)\mid}\left(Q^\pi(s,a)-V^\pi(s)\right), \tag{42}$$

其中 p_t^π 表示为状态 s 的 t 步之后被访问的边缘概率。**Proof.** 首先有，

$$\begin{aligned}\frac{d}{d\epsilon}(\pi+\epsilon\chi)(a\mid s)\Big|_{\epsilon=0}&=\frac{d}{d\epsilon}\frac{\pi(a,s)+\epsilon\chi(s,a)}{\pi(s)+\epsilon\chi(s)}\Big|_{\epsilon=0}\\&=\frac{\chi(s,a)-\chi(s)\pi(a\mid s)}{\pi(s)}\end{aligned} \tag{43}$$

其中，令 $\chi(s)=\int\chi(s,a')da'$，而且目标函数可以写为轨迹概率的形式（这个积分符号放在这挺奇怪的，到底要对啥积分呢？实际上这里就是用积分符号表示一下求期望的意思），

$$-J_{\mathrm{RL}}=\int\sum_{t=1}^{\infty}\gamma^{t-1}r_t p_0\left(s_0\right)\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\prod_{k=1}^{\infty}\pi\left(a_k\mid s_{k-1}\right)$$

在计算 $\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)|_{\epsilon=0}$，观测上述形式发现，只要将 $\pi(a_k|s_{k-1})$ 进行替换为 $\frac{d}{d\epsilon}(\pi+\epsilon\chi)(a_k|s_{k-1})|_{\epsilon=0}$，那么有，

$$\begin{aligned}&-\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)\Big|_{\epsilon=0}\\&=\int\sum_{t=1}^{\infty}\gamma^{t-1}r_t p_0\left(s_0\right)\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\\&\times\sum_{k=1}^{\infty}\frac{\chi\left(s_{k-1},a_k\right)-\chi\left(s_{k-1}\right)\pi\left(a_k\mid s_{k-1}\right)}{\pi\left(s_{k-1}\right)}\prod_{\substack{\ell=1\\\ell\neq k}}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)\\&=\sum_{k=1}^{\infty}\int\sum_{t=1}^{\infty}\gamma^{t-1}r_t p_0\left(s_0\right)\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\\&\times\frac{\chi\left(s_{k-1},a_k\right)-\chi\left(s_{k-1}\right)\pi\left(a_k\mid s_{k-1}\right)}{\pi\left(s_{k-1}\right)}\prod_{\substack{\ell=1\\\ell\neq k}}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)\end{aligned}$$

其中用到了公式 (23) 中的结论，注意到当 $t<k$ 时，求和为 0，这个在基础的强化学习知识里面就有提到过，理由是小于 k 时刻的奖励不会对之后的决策产生影响。所以，上述公式被进一步化简为，

$$\begin{aligned}
&-\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)\Big|_{\epsilon=0}\\
&=\sum_{k=1}^{\infty}\int\sum_{t=k}^{\infty}\gamma^{t-1}r_t p_0\left(s_0\right)\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\\
&\times\frac{\chi\left(s_{k-1},a_k\right)-\chi\left(s_{k-1}\right)\pi\left(a_k\mid s_{k-1}\right)}{\pi\left(s_{k-1}\right)}\prod_{\substack{\ell=1\\\ell\neq k}}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)
\end{aligned}$$

令，

$$p_{k-1}^{\pi}\left(s_{k-1}\right)=\int\prod_{j=1}^{k-1}p\left(s_j,r_j\mid s_{j-1},a_j\right)\prod_{\ell=1}^{k-1}\pi\left(a_\ell\mid s_{\ell-1}\right)$$

原式将改写为，

$$\begin{aligned}
&-\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)\Big|_{\epsilon=0}\\
&=\sum_{k=1}^{\infty}\int\sum_{t=k}^{\infty}\gamma^{t-1}r_t p_{k-1}^{\pi}\left(s_{k-1}\right)\prod_{j=k}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\\
&\times\frac{\chi\left(s_{k-1},a_k\right)-\chi\left(s_{k-1}\right)\pi\left(a_k\mid s_{k-1}\right)}{\pi\left(s_{k-1}\right)}\prod_{\ell=k+1}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)
\end{aligned}$$

虽然，要用的时候这些公式都可以抄抄，然后改一下框架，但是大家注意作者的推导的思路是不断的将下标统一到和 k 相关，这样就比较好建立和 k 直接的关系。将所有的项都往前推 $k-1$ 可以得到，

$$\begin{aligned}
&-\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)\Big|_{\epsilon=0}\\
&=\sum_{k=1}^{\infty}\int\sum_{t=1}^{\infty}\gamma^{t+k-2}r_t p_{k-1}^{\pi}\left(s_0\right)\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\\
&\times\frac{\chi\left(s_0,a_1\right)-\chi\left(s_0\right)\pi\left(a_1\mid s_0\right)}{\pi\left(s_0\right)}\prod_{\ell=2}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)
\end{aligned}$$

其中，

$$\begin{aligned}
V^{\pi}\left(s_0\right)&=\int\sum_{t=1}^{\infty}\gamma^{t-1}r_t\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\prod_{\ell=1}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)\\
Q^{\pi}\left(s_0,a_1\right)&=\int\sum_{t=1}^{\infty}\gamma^{t-1}r_t\prod_{j=1}^{\infty}p\left(s_j,r_j\mid s_{j-1},a_j\right)\prod_{\ell=2}^{\infty}\pi\left(a_\ell\mid s_{\ell-1}\right)
\end{aligned}$$

这里将 $\chi\left(s_0,a_1\right)-\chi\left(s_0\right)\pi\left(a_1\mid s_0\right)$ 写开，就可以得到，

$$\begin{aligned}
&-\frac{d}{d\epsilon}J_{\mathrm{RL}}(\pi+\epsilon\chi)\Big|_{\epsilon=0}\\
&=\sum_{k=1}^{\infty}\int\gamma^{k-1}p_{k-1}^{\pi}\left(s_0\right)\frac{Q^{\pi}\left(s_0,a_1\right)\chi\left(s_0,a_1\right)-V^{\pi}\left(s_0\right)\chi\left(s_0\right)}{\pi\left(s_0\right)}
\end{aligned}$$

将 $\chi(s_0,a_0)$ 中的 s_0,a_0 换成 s,a 即可得到影响函数的表达形式，

$$\Psi_{\mathrm{RL}}(s,a)=-\frac{\sum_{k=0}^{\infty}\gamma^k p_k^{\pi}(s)}{\pi(s)}\left(Q^{\pi}(s,a)-V^{\pi}(s)\right)\tag{44}$$

而 PFD 的 descent 步骤对应着，更新 θ 参数，

$$\theta \mapsto -\mathbb{E}_{\pi_\theta(s,a)}\left[\frac{\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s)}{\pi(s)}\left(Q^\pi(s,a)-V^\pi(s)\right)\right] \tag{45}$$

令 $d^\pi(s)=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s)$，那么此求解过程可以简化为，

$$\begin{aligned}\theta &\mapsto -\frac{1}{1-\gamma}\mathbb{E}_{d^\pi(s)}\mathbb{E}_{\pi_\theta(a|s)}\left[Q^\pi(s,a)-V^\pi(s)\right]\\ \theta &\mapsto -\frac{1}{1-\gamma}\mathbb{E}_{d^\pi(s)}\mathbb{E}_{\pi_\theta(a|s)}\left[Q^\pi(s,a)\right]+\text{ constant}\end{aligned} \tag{46}$$

最简单的减小公式 (46) 的方法是全局最小化。对应的 $\pi_\theta(a|s)$ 采用 greedy 策略。实际上，$Q^\pi(s,a)$ 的评估表示 PFD 中计算影响函数的过程；而贪心策略则是对应着 PFD 中的 descent 过程。

6.2 策略梯度和 AC

在传统的策略迭代中通常采用蒙特卡罗方法来计算 Q 函数，实际上并没有参数化的估计。而实际上，通过采用 TD 的方法来估计 Q 函数，也就是 AC 方法。AC 算法描述了一类算法，其中用网络近似当前策略的值函数，然后使用估计的值函数对策略的参数进行梯度更新。

Proposition 14. 强化学习在 PFD 下的目标函数 J_{RL}，其中当影响函数用类似于蒙特卡罗，最小二乘类似的方法近似时，就导出了 AC 算法。

有很多估计影响函数的方法，这里只列举几种常见的算法。（这里对研究 RL 的同学来说就非常简单了）最经典的是策略梯度算法，记为 REINFORCE，其中直接采用蒙特卡罗方法计算 $Q^\pi(s,a)$。DDPG 中采用神经网络来近似 Q 函数，并且可以将其当做影响函数的近似。类似的还有 A2C 算法，其中引入了优势函数的概念，等等。所有这些算法传统上都是由著名的策略梯度定理导出的。

6.3 Dual AC

因为，J_{RL} 不是凸函数，所以不能直接用 PFD 中的凸共轭方法来近似影响函数。然而，Proposition 13 中的形式强烈建议将任意分布 $\pi(s)$ 固定为状态 $d_\pi(s)$ 的折扣边际分布。与强化学习的线性规划公式密切相关，这种选择结果是凸化 J_{RL}（这里说实话，没怎么理解），从而可以使用凸共轭逼近其影响函数。我们希望给出一种鞍点优化形式下的表达，而在 DAI Bo 的对偶 AC 里面给出了这个表达，

$$\sup_{\pi}\inf_{V}(1-\gamma)\mathbb{E}_{p_0(s)}[V(s)]+\mathbb{E}_{\pi(s,a)}[\mathcal{A}V(s,a)] \tag{47}$$

其中，$\mathcal{A}V(s,a)=\mathbb{E}_{p(s',r|s,a)}[r+\gamma V(s')]-V(s)$。

Proposition 16. J_{RL} 的凸函数是，

$$J_{\mathrm{RL}}^\star(\varphi)=(1-\gamma)\mathbb{E}_{p_0(s)}V_\varphi(s)+\{V_\varphi \text{ exists }\} \tag{48}$$

其中，V_φ 为 $\varphi=-\mathcal{A}V_\varphi$ 的唯一解。

Proof. 正如文章中之前提到的那样，我们定义一个任意的分布，$\pi(s)=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^\pi(s)$。那么，$\pi(s,a)$ 表示状态动作 occupancy measure，用以描述在此策略控制的轨迹上遇到状态-行动对 (s,a) 的概率。所以，很容易想到策略 $\pi(a|s)$ 和 occupancy measure 之间存在双射。

那么可以将此函数重新定义为，

$$J_{\mathrm{RL}}(\pi)=-\mathbb{E}\sum_{t=1}^{\infty}\gamma^{t-1}r_t+\left\{\forall s:\pi(s)=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t^{\pi}(s)\right\} \tag{49}$$

这相当于在原目标函数上加了一个约束条件。其中 $\{\cdot\}$ 是凸的指示函数，进一步此等式可以被重写为，

$$J_{\mathrm{RL}}(\pi)=-\mathbb{E}_{\pi(s,a)}R(s,a)+\left\{\forall s':\pi\left(s'\right)=(1-\gamma)p_0\left(s'\right)+\gamma\mathbb{E}_{\pi(s,a)}p\left(s'\mid s,a\right)\right\}$$

其中，$R(s,a)=\mathbb{E}_{p(s',r|s,a)}[r]$。而约束项是 Bellman flow equation，这个推导可见[DualDICE 解读]，实际上就是用了平稳状态方程。此公式是凸的，因为它是一个仿射函数和一个凸集的指示函数的和 (实际上是一个仿射子空间)。

根据定义哟，$-\varphi=\mathcal{A}V_\varphi$，其中，$\mathcal{A}V(s,a)=\mathbb{E}_{p(s',r|s,a)}[r+\gamma V(s')]-V(s)$。如果存在方程的解，$V_\varphi$ 是由 φ 唯一定义的。请注意 V_φ 是 Bellman 算子 $\mathcal{T}^a$ 的不动点，

$$\left(\mathcal{T}^aV\right)(s)=(R+\varphi)(s,a)+\gamma\mathbb{E}_{p(s'|s,a)}V\left(s'\right)$$

并且此算子是收缩算子，因此有唯一的不动点。其中对于任意的 a 都可以得到 V_φ 的表达式，

$$V_\varphi(s)=\lim_{k\to\infty}\left(\mathcal{T}^a\right)^k0=\mathbb{E}^a\sum_{t=1}^{\infty}\gamma^{t-1}(R+\varphi)\left(s_t,a\right)$$

其中期望基于确定性策略。那么，可以将 J_{RL} 用拉格朗日乘子 $V(s)$ 重写，

$$\begin{aligned}J_{\mathrm{RL}}(\pi)&=-\mathbb{E}_{\pi(s,a)}R(s,a)+\sup_V\int V\left(s'\right)\left[\pi\left(s'\right)-(1-\gamma)p_0\left(s'\right)-\gamma\mathbb{E}_{\pi(s,a)}p\left(s'\mid s,a\right)\right]ds'\\&=\sup_V-\mathbb{E}_{\pi(s,a)}R(s,a)+\mathbb{E}_{\pi(s)}V(s)-(1-\gamma)\mathbb{E}_{p_0(s)}V(s)-\gamma\mathbb{E}_{\pi(s,a)}\mathbb{E}_{p(s'|s,a)}V\left(s'\right)\\&=\sup_\varphi\mathbb{E}_{\pi(s,a)}\varphi(s,a)-(1-\gamma)\mathbb{E}_{p_0(s)}V_\varphi(s)-\{V_\varphi\text{ exists }\}\end{aligned}$$

上述公式中的，第二行到第三行的推导让人有点容易迷，这里用到了 $(1-\gamma)\mathbb{E}_{p_0(s)}V_\varphi(s)+\{V_\varphi\text{exists}\}$，而且关于目标函数的变换，用到了变量替换的技巧，也就是被替换的变量优化得到的最优解，和原变量一致。类似于 Proposition 8 中的证明，可以得到，

$$J_{\mathrm{RL}}^\star(\varphi)=(1-\gamma)\mathbb{E}_{p_0(s)}V_\varphi(s)+\{V_\varphi\text{ exists }\}$$

其中，关于 $\varphi=V_\varphi(s)-\mathbb{E}_{p(s',r|s,a)}[r+\gamma V(s')]$ 是凸函数，因为其满足，

$$V_{\alpha\varphi+(1-\alpha)\varphi'}=\alpha V_\varphi+(1-\alpha)V_{\varphi'}$$

那么根据凸共轭的公式，就可以推导出公式 (47) 中的对偶 AC 了，

$$\inf_\pi\sup_\varphi\mathbb{E}_{\pi(s,a)}[\varphi(s,a)]-J_{\mathrm{RL}}^\star(\varphi)=\inf_\pi\sup_\varphi\mathbb{E}_{\pi(s,a)}\left[-\mathcal{A}V_\varphi(s,a)\right]-(1-\gamma)\mathbb{E}_{p_0(s)}V_\varphi(s)$$

7 实验

本文是纯理论的文章，没有实验。

8 结论

本文提出了几种新的研究方向。因为，本文通过定义 PFD 证明了各种机器学习算法的通用性。那么，可以将 idea 和某些领域的方法从一个领域转移到另一个领域。比如在 GAN 中，利用 1 阶李普希斯约束来提升训练算法的稳定性，这样的方式是不是可以运用于强化领域，通过对优势函数的约束来提升强化学习算法的稳定性。

此外，本文采用的较为抽象观点来定义 GAN、变分推理和强化学习，这意味着底层理论的改进，可以同时改进这三个领域。基于凸共轭的一般性的影响函数逼近技术的改进可以同时改进这三个领域。在参数化概率分布的梯度下降之外，更复杂的下降技术，如 Frank-Wolfe 或信任区域方法，可以改进学习或产生更有效的收敛保证。**并且，小编认为对于极大极小问题的优化，还有很大的提升空间**。

最后，给出了概率泛函下降法应用于新问题的可能性。原则上，该算法可以简单地应用于任何想要优化概率分布的情况，这样可能引导出新的、更直接的方法来解决问题，例如，数学金融、平均场博弈或部分可观测 MDP 过程。

小编觉得，这套理论非常的有意思，统一的定义了 GAN，变分推断和强化学习。这里自卖自夸一下，小编早就觉得分布和分布之间的优化问题都是可以统一的。不过这里影响函数的定义，非常像得分函数的定义。不过，虽然 Dual AC 的推导非常厉害，小编还是觉得推导有点不太严谨，hhhhhh。读完这篇文章，让我对深度学习中分布引导建立的观点，有了更深刻的认识。希望以后在这方面做出更深刻的思考！还有这篇文章中有很多对于凸共轭，对偶性的定义和证明可以拿来抄抄。