A Geometric Perspective on Optimal Representations for Reinforcement Learning

Marc G. Bellemare [1] Will Dabney [2] Robert Dadashi [1] Adrien Ali Taïga [1 3] Pablo Samuel Castro [1]
Nicolas Le Roux [1] Dale Schuurmans [1 4] Tor Lattimore [2] Clare Lyle [5]

【强化学习 75】AVF

张楚珩
清华大学 交叉信息院博士在读

AVF 是 adversarial value function 的缩写，这篇工作讲了一个学习 representation 的方法。

原文传送门

Bellemare, Marc G., et al. "A Geometric Perspective on Optimal Representations for Reinforcement Learning." arXiv preprint arXiv:1901.11530 (2019).

特色

这篇工作是专栏前一篇工作（polytope）的后续，前一篇工作从几何的视角分析了策略到价值函数的映射关系，这一篇工作就利用相关的分析来对于给定 MDP 学习一个表示（representation）。相比于之前一些比较 ad hoc 的表示学习工作，这篇工作更有理论保证。具体地，这篇工作要求，对于任意的策略，学习到的表示在一个简单的线性映射下，其对于价值函数的表示误差都不会太大。

过程

1. 价值函数近似

在价值函数近似上，文章使用了一个所谓的 two-part approximation：把一个状态先映射为一个 d 维的表示 $\phi: \mathcal{X} \to \mathbb{R}^d$ ，然后再使用一个线性映射得到相应的价值函数，

$$\hat{V}_\phi(x) := \phi(x)^\top \theta,$$

下图展示了这个 approximation 的过程。

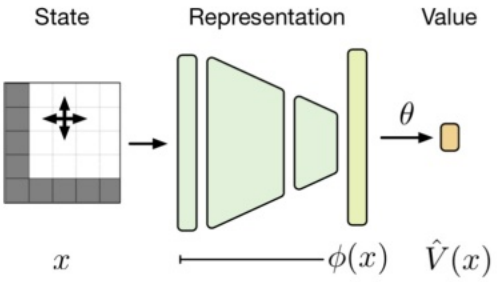

Figure 1. Many common deep reinforcement learning architectures can be viewed as a combination of two mappings $\phi : \mathcal{X} \to \mathbb{R}^d$ and $\theta : \mathbb{R}^d \to \mathbb{R}$.

当 representation 给定之后，后面一层参数 θ 的估计就只需要最小化如下这个（对于每个状态均匀加权的）误差即可

$$\left\|\hat{V}_\phi - V^\pi\right\|_2^2 = \sum_{x \in \mathcal{X}} (\phi(x)^\top \theta - V^\pi(x))^2.$$

当 representation 给定之后，如果把价值函数看做是 $n=|\mathcal{X}|$ 维空间中的一个点，那么 θ 所能表示的价值函数在都在如下这个超平面中

$$H = \left\{\Phi\theta : \theta \in \mathbb{R}^d\right\}$$

最小化上述误差相当于在找 V^π 往这个超平面上的投影，即 $\hat{V}_\phi^\pi = \Pi_\phi V^\pi$ 。

Representation 的学习可以通过一些 auxiliary task 来得到，比如之前的工作使用 $\phi(x)$ 外接神经网络去预测下一个状态。这里的学习方法也类似，只不过外接了一个线性映射去预测相应的价值函数。后面会看到，文章要求对于一族特殊的策略，希望都能找到参数 θ 使得相应的价值函数估计接近真实的价值函数。

Representation 的学习一般可以分为两种，一种是事先学习好（pretrained），另一种是随着策略的学习同时来学习（concurrently）。这里主要讲 pretrained 的情况。

2. 价值函数 polytope

由于上一篇才讲了，这里就不复习 polytope 了，只说下面这个引理

Lemma 1. *Let $\delta \in \mathbb{R}^n$ and define the functional $f(V) := \delta^\top V$, with domain $\mathcal{V}$. Then f is maximized by an extremal vertex. Furthermore, the set of directions $\delta \in \mathbb{R}^n$ for which the maximum of f is achieved by multiple extremal vertices has Lebesgue measure zero in $\mathbb{R}^n$.*

其中 extremal vertex 的定义如下

> $V \in \mathcal{V}$ is an *extremal vertex* if it is a vertex of the convex hull of $\mathcal{V}$ (if $\mathcal{V}$ is a convex, non-intersecting polytope, then all vertices of $\mathcal{V}$ are extremal vertices)

这个比较显然，即对于一个 polytope 在某个方向上投影最远的点一定是其 convex hull 的顶点。

3. 表示学习的目标

希望学习到一个这样的 representation：对于任意的策略，其表示经过线性映射都能尽可能得到该策略下的价值函数。即可以写出如下优化目标

$$\min_{\phi \in \mathscr{R}} \max_{\pi \in \mathcal{P}} \left\| \hat{V}_{\phi}^{\pi} - V^{\pi} \right\|_2^2. \tag{1}$$

其中

$$\mathscr{R} \equiv \mathbb{R}^{n \times d} \text{ be the set of } d\text{-dimensional representations.}$$

一个 d 维的 representation 能表示出来的价值函数组成一个 d 维的超平面 $H = \{\theta\theta : \theta \in \mathbb{R}^d\}$，上述目标就是希望 ν 中的离该超平面最远的一点也不要太远。形象地说，就是希望找到一个超平面尽可能多地在 ν 里面。

该目标要求对于所有的策略求最大值，这显然不容易做到，但是可以证明其实只需要对于所有的 extremal policies （即 $\mathcal{P}_v$）求最大值即可，这大大减小了搜索的空间。即，有如下定理。

> **Theorem 1.** *For any representation $\phi \in \mathscr{R}$, the maximal approximation error measured over all value functions is the same as the error measured over the set of extremal vertices:*
>
> $$\max_{\pi \in \mathcal{P}} \left\| \hat{V}_{\phi}^{\pi} - V^{\pi} \right\|_2^2 = \max_{\pi \in \mathcal{P}_v} \left\| \hat{V}_{\phi}^{\pi} - V^{\pi} \right\|_2^2.$$
>
> *As a consequence, the optimal representation ϕ^* is also the minimizer of*
>
> $$\min_{\phi \in \mathscr{R}} \max_{\pi \in \mathcal{P}_v} \left\| \hat{V}_{\phi}^{\pi} - V^{\pi} \right\|_2^2.$$

更进一步，如果一个表示是最优的，那么由它确定的 d 维超平面会对于 $d+1$ 个 extremal policies 上价值函数的误差相同，并且这 $d+1$ 个 extremal policies 产生的误差也等于所有 v 上产生的误差。

Proposition 1. *For each $\phi : \mathcal{X} \to \mathbb{R}^d$, there exist $d+1$ extremal policies $\mu_i \in \mathcal{P}_v$, $i \in \{1, \ldots, d+1\}$ such that*

$$\max_{\pi \in \mathcal{P}_v} \left\| \hat{V}_\phi^\pi - V^\pi \right\|_2^2 = \max_{\mu_i} \left\| \hat{V}_\phi^{\mu_i} - V^{\mu_i} \right\|_2^2.$$

Furthermore, for the representation ϕ minimizing the above all approximation errors are the same:

$$\left\| \hat{V}_\phi^{\mu_i} - V^{\mu_i} \right\|_2^2 = \left\| \hat{V}_\phi^{\mu_j} - V^{\mu_j} \right\|_2^2 \quad \forall i, j.$$

4. 表示学习算法

表示学习算法主要分为两步，即先找到 k 个特殊的策略及其对应的价值函数（adversarial value functions，AVFs），再利用 AVF 计算 representation 的损失函数，通过梯度下降的方法来训练一个神经网络编码的 representation。算法表示如下

Algorithm 1 Representation learning using AVFs

input k – desired # of AVFs, d – desired # of features.

Sample $\delta_1, \ldots, \delta_k \sim [-1, 1]^n$

Compute $\mu_i = \arg\max_\pi \delta_i^\top V^\pi$ using policy gradient

Find $\phi^* = \arg\min_\phi \tilde{L}(\phi; \mu_1, \ldots, \mu_k)$ (Equation 4)

先讲如何得到 AVFs。前面说到给定一个方向 δ，使得 $\delta^\top V$ 最大的点是一个 extremal point，也就是我们感兴趣的一个点。因此这里随机生成一些方向，然后又策略梯度去得到这些我们感兴趣的策略，把这些策略及其对应的价值函数作为 AVF。

1. Sample δ uniformly in $[-1,1]^n$,
2. Use policy gradient to find $\arg\max_{V^\mu \in \mathcal{V}} \delta^\top V^\mu$.

接下来就使用得到的 AVF 来更新 representation。Loss function 可以写为

$$L(\phi;\boldsymbol{\mu}) = \max_{\mu_i \in \boldsymbol{\mu}} \left\| \Pi_\phi V^{\mu_i} - V^{\mu_i} \right\|_2^2. \qquad (3)$$

其中 $\mu := \mu_1, \cdots, \mu_k$ 是 k 个策略，对应的价值函数为 AVFs。

由于 max 函数不方便求导（当两个策略产生的 loss 相同时梯度就会不确定性通过哪一项往回传），因此把上述的 max 改写为 softmax，即

$$\tilde{L}(\phi;\boldsymbol{\mu}) = \log \sum_{\mu_i \in \boldsymbol{\mu}} \exp\left\{ \left\| \Pi_\phi V^{\mu_i} - V^{\mu_i} \right\|_2^2 \right\}. \qquad (4)$$

对该损失函数做梯度下降即可求到相应的 representation。

最大化 $\delta^\top V$ 究竟是在做什么呢？下面定理告诉我们最大化该目标相当于是在同时最大化或者最小化每个状态下的目标函数。

Theorem 2. *We have the following equivalence:*

$$\max_{\pi \in \mathcal{P}} \delta^\top V^\pi = \max_{\pi \in \mathcal{P}} d_\pi^\top r,$$

where the vector d_π satisfies the reverse Bellman equation

$$d_\pi = \delta + \gamma P^{\pi\top} d_\pi.$$

In particular, for the policy $\tilde{\pi}$ maximizing the above we have

$$V^{\tilde{\pi}}(x) = r(x) + \gamma \begin{cases} \max_{a \in \mathcal{A}} \mathbb{E}_{x' \sim P} V^{\tilde{\pi}}(x') & d_{\tilde{\pi}}(x) > 0, \\ \min_{a \in \mathcal{A}} \mathbb{E}_{x' \sim P} V^{\tilde{\pi}}(x') & d_{\tilde{\pi}}(x) < 0. \end{cases}$$

可以看出 $d_\pi = (I - \gamma P^\pi)^{-1}\delta$ ，如果 $\delta = I$ （全 1 向量），那么 d_π 就是前面讲到的 successor representation，这时该目标就是一个正常的强化学习目标。

实验

文章在一个有四个房间的格子世界上做了实验，下图画出了随机生成的一个 δ 及其对应的 d_π 和 AVF。

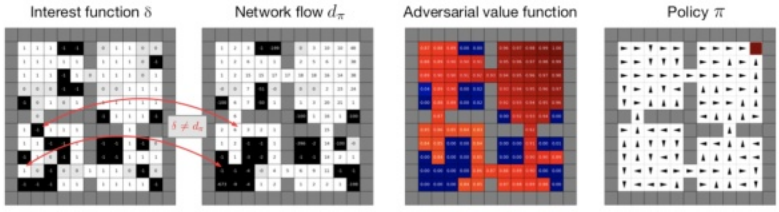

Figure 4. An interest function δ in the four-room domain and the corresponding adversarial value function. Red arrows highlight states where the sign of the interest function is opposite to that of the network flow d_π.

经验上来说，对于某个状态 s，如果 $\delta(s) > 0$，那么大概率地其对应的 $d_\pi(s) > 0$，当然也会有例外，比如图中红色箭头所示的状态。

文章还对比了另外两种做法形成的价值函数，一种是随机生成价值函数，另一种是随机生成策略然后求得对应的价值函数。可以看到 AVF 生成的价值函数（也可以看出相应的策略）更具有代表性。

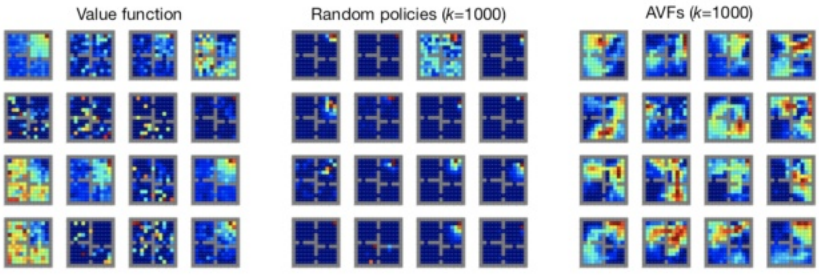

Figure 5. 16-dimensional representations learned by predicting a value function (left), the value functions of 1000 random policies (middle), or 1000 AVFs sampled using our method (right). Each element of a panel depicts the activation of a given feature at each state.

下图展示了不同数目的 AVF 学习得到 representation 每一维的图像，可以看到 $k = 1000; 4000$ 时，学习到的 representation 每一维都基本上能够反映一些结构信息（比如在是否在某个房间里面）。

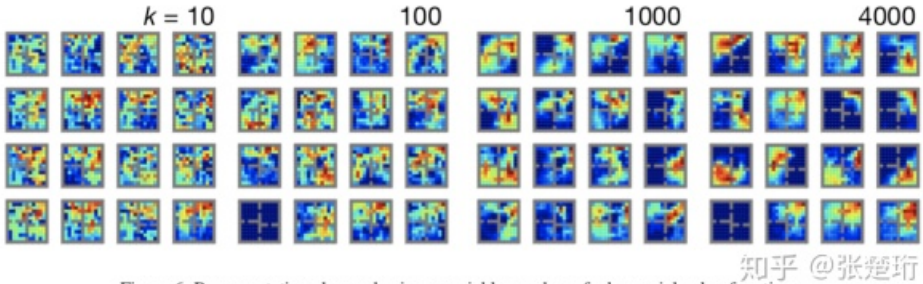

Figure 6. Representations learned using a variable number of adversarial value functions.

相应地，使用学到的 representation 来完成 RL 任务，需要的样本数目也会更少。

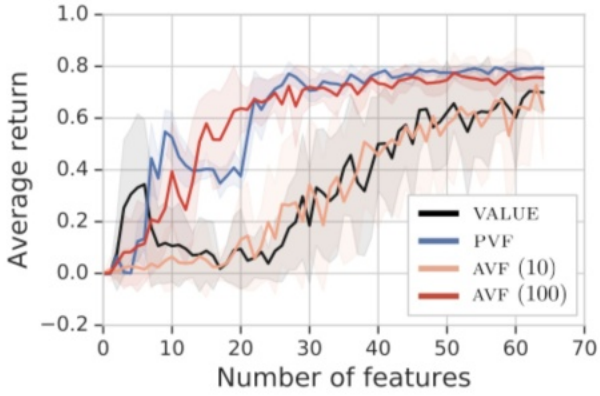

Figure 7. Average discounted return achieved by policies learned using a representation, with given number of features, produced by VALUE, AVF, or PVF. Average is over 20 random seeds and shading gives standard deviation.

一个猜想：如果一个 MDP 有 bisimulation，那么其价值函数的 polytope 在一个低维子空间上。

发布于 2019-06-21

强化学习 (Reinforcement Learning)