Distributional Reinforcement Learning with Monotonic Splines

Yudong Luo[1,4], Guiliang Liu[1,4], Haonan Duan[2,4], Oliver Schulte[3], Pascal Poupart[1,4]

[1]University of Waterloo [2]University of Toronto [3]Simon Fraser University [4]Vector Institute

Abstract

One key challenge in quantile based distributional RL lies in how to parameterize the quantile function when minimizing the Wasserstein metric of temporal differences. Existing algorithms use step functions or piece-wise linear functions.
We propose to learn smooth continuous quantile functions represented by monotonic rational-quadratic splines

- Efficiently learned by deep neural network
- Naturally solves the quantile crossing problem
- Outperforms its counterparts in stochastic domains

Distributional RL

Distributional RL algorithms characterize the total return as a random variable and estimate its underlying distribution, so that the intrinsic uncertainty of MDP is captured. In contrast, traditional value-based RL algorithms focus only on the mean of the random variable.

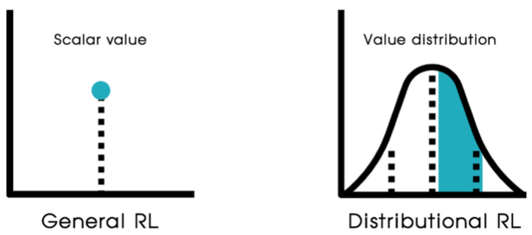

Figure 1: Traditional RL VS. Distributional RL

Distributional Bellman Operator

Traditional Bellman equation

$$Q(s,a) = \mathbb{E}R(s,a) + \gamma \mathbb{E}Q(S',A') \tag{1}$$

Distributional Bellman operator [1]

$$\mathcal{T}^{\pi} Z(s,a) \overset{D}{=} R(s,a) + \gamma Z(S',A') \tag{2}$$

with $S' \sim P(\cdot|s,a)$, $A' \sim \pi(\cdot|S')$, and $X \overset{D}{=} Y$ indicates that random variables X and Y follow the same distribution.

Two Key Questions

- Which distance metric to optimize
- How to parameterize the return distribution

Wasserstein Metric

The distributional Bellman operator is a contraction under **p-Wasserstein metric** [1].

$$W_p(X,Y) = \left(\int_0^1 |F_X^{-1}(\omega) - F_Y^{-1}(\omega)|^p d\omega\right)^{1/p} \tag{3}$$

where F^{-1} is the quantile function (inverse cumulative distribution function).
Directly minimize Wesserstein loss from samples suffers from biased gradients. **Quantile Regression** (QR) offers unbiased gradient estimation [2]. The loss function is Huber quantile regression loss.

QR-based Methods

1. **QR-DQN** [2] and **NC-QR-DQN** [3] represents return distribution by a uniform mixture of N Diracs

$$Z_\theta(s,a) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i(s,a)} \tag{4}$$

2. **IQN** [4] takes arbitrary $\tau \sim U([0,1])$, and outputs the corresponding quantile value.
3. **FQF** [5] learns a separate network to propose τ taking state as input.
4. **NDQFN** [6] uses piece-wise linear function.

Learning Quantile Function with Monotonic Splines

Limitations of the existing methods: 1) a precise approximation for quntile function may need infinite τs if use discretization (QR-DQN, NC-QR-DQN, IQN, FQF) 2) quantile crossing problem (QR-DQN, IQN, FQF) 3) piece-wise linear function has limited approximation ability (NDQFN).

Monotonic Splines

Monotonic splines produce a monotonic interpolant to a set of monotonic data points (called knots). The monotonicity of this kind of splines naturally fits the non-decreasing property of the quantile function.

Monotonic Rational Quadratic Splines

Suppose we learn splines for K bins. Neural net gives knots $\{(x_k, y_k)\}_{k=0}^{K}$, and derivatives $\{d_k\}_{k=0}^{K}$.
$x_0 < ... < x_k < ... < x_K$. $x_0 = 0$, $x_K = 1$
$y_0 < ... < y_k < ... < y_K$. d_k is positive.
Denote $g_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$ and $h_k(x) = (x - x_k)/(x_{k+1} - x_k)$ for $x \in [x_k, x_{k+1}]$. Use h_k for short of $h_k(x)$, we have two quadratic functions

$$\begin{aligned} O_k(h_k) &= g_k y_{k+1} h_k^2 + (y_k d_{k+1} + y_{k+1}) h_k (1-h_k) \\ &\quad + g_k y_k (1-h_k)^2 \\ P_k(h_k) &= g_k + (d_{k+1} + d_k - 2g_k) h_k (1-h_k) \end{aligned} \tag{5}$$

$$f_k(h_k) = \frac{O_k(h_k)}{P_k(h_k)} \tag{6}$$

$$f_k(h_k) = y_k + \frac{(y_{k+1} - y_k)[g_k h_k^2 + d_k h_k (1-h_k)]}{g_k + (d_{k+1} + d_k - 2g_k) h_k (1-h_k)} \tag{7}$$

Learned Quantile Function

- x_k: sigmoid + cumulative sum
- y_k: sigmoid + rescale (multiply α then plus β)
- d_k: softplus

Stochastic Environment Most previous distributional RL algorithms were evaluated with deterministic Atari games. It is problematic since the resulting value distributions tend to be simple and close to deterministic.

Windy Gridworld Some columns are affected by some wind. A reward of -1 is earned at each step. Have probability 0.1 of moving in a random direction without wind effect, otherwise the wind pushes the agent northward.

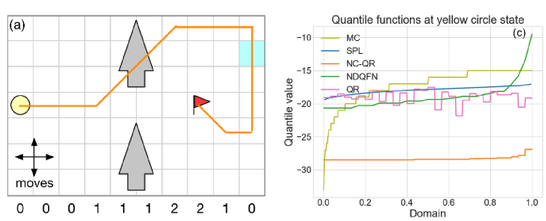

Figure 2: Learned quantile in stochastic Windy Gridworld

Stochastic PyBulletGym

Add noise $\epsilon \sim \mathcal{N}(0, \sigma)$ both the location and velocity of each part of the robot. Combine distributional RL with DDPG and SAC. Check results details in our paper.

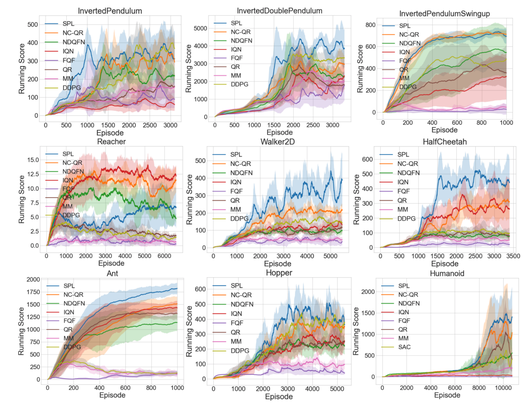

Figure 3: Performance in stochastic PyBulletGym

References

[1] Marc G Bellemare, Will Dabney, and Rémi Munos.
A distributional perspective on reinforcement learning.
In *ICML*, pages 449–458, 2017.

[2] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos.
Distributional reinforcement learning with quantile regression.
In *AAAI*, volume 32, 2018.

[3] Fan Zhou, Jianing Wang, and Xingdong Feng.
Non-crossing quantile regression for distributional reinforcement learning.
In *NeurIPS*, volume 33, 2020.

[4] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos.
Implicit quantile networks for distributional reinforcement learning.
In *ICML*, pages 1096–1105, 2018.

[5] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu.
Fully parameterized quantile function for distributional reinforcement learning.
In *NeurIPS*, volume 32, 2019.

[6] Fan Zhou, Zhoufan Zhu, Qi Kuang, and Liwen Zhang.
Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning.
In *IJCAI*, 2021.