Implicit Quantile Networks

Distributional Reinforcement Learning

또봇 조

DaeukJeong, SejunPark, VictorCappa

Outline

1. Implicit Quantile Network(IQN) is one of distributional RL algorithms.
2. Important papers of distributional RL is the followings :
   1. A Distributional Perspective on Reinforcement Learning - Marc G.Bellemare, Will Dabney, Remi Munos. (2017)
   2. An Analysis of Categorical Distributional Reinforcement Learning - Mark Rowland, Marc G.Bellemare, Will Dabney, Remi Munos, Yee Whye Teh. (2018)
   3. Distributional Reinforcement Learning with Quantile Regression - Will Dabney, Mark Rowland, Marc G. Bellemare, Remmi Munos. (2017)
   4. Implicit Quantile Networks for Distributional Reinforcement Learning - Will Dabney, Georg Ostrovski, David Silver, Remi Munos. (2018)
3. Our paper is the last paper.

Contents

Value Distribution

Recall the (expected) Bellman equation

$$\begin{aligned} Q(x,a) &= E[G_t|S_t = x, A_t = a] \\ &= E[R_t + \gamma Q(S_{t+1}, A_{t+1})|S_t = x, A_t = a] \\ &= ER(x,a) + \gamma EQ(X', A') \end{aligned}$$

where $(x,a) \to (X', A')$ is a random transition.
$(X' \sim p(\cdot|x,a), A' \sim \pi(\cdot|x))$

Value Distribution

Now we are interested in the distribution sense. Define the distributional Bellman equation

$$Z(x,a) =_d R(x,a) + \gamma Z(X', A')$$

where $(x,a) \to (X', A')$ is a random transition.
$(X' \sim p(\cdot|x,a), A' \sim \pi(\cdot|x))$ Z is called the value distribution.

Mathematical Setting

We use a time-homogeneous MDP $\langle \mathcal{X}, \mathcal{A}, R, p, \gamma \rangle$.

1. The return Z^π is the sum of discounted rewards along the agent's trajectory, i.e.
$$Z^\pi(x,a) = \sum_{t=0}^{\infty} \gamma^t R(x_t, a_t).$$
We can consider Z^π as a mapping
$$Z^\pi : \mathcal{X} \times \mathcal{A} \to \{\text{functions} : \Omega \to \mathbb{R}\}.$$

2. The value function is
$$Q^\pi(x,a) = EZ^\pi(x,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(x_t, a_t)\right]$$
$$= ER(x,a) + \gamma E_{\substack{x' \sim p(\cdot|x,a) \\ a' \sim \pi(\cdot|x')}}[Q^\pi(x', a')].$$
where $x_t \sim p(\cdot|x_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|x_t), x_0 = x, a_0 = a.$

Example : Flip a Coin

1. Consider coin flip situation. It has only one-state/one-action. The random reward is Bernoulli(0.5) and the discount factor is 1/2.
2. Then the (expected) Bellman equation is $V = \frac{1}{2} + \frac{1}{2}V$, thus $V = 1$.
3. The return $Z = \sum_{k=0}^{\infty} 2^{-k} R_k$. Thus the value distribution is $Z = Unif(0, 2)$.[1]
4. The distributional Bellman equation is $Z = Bernoulli(\frac{1}{2}) + \frac{1}{2}Z$, i.e.

$$\eta(z) = \frac{1}{2}(\delta(0) + \delta(1)) * 2\eta(2z) = \eta(2z) + \eta(2(z-1)).$$

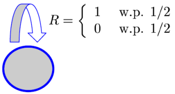


Figure: Example : Flip a Coin

[1] Consider binary expression

Why consider distribution sense?

1. Performance improves.
2. It can be better than approximating only the expected value.
3. We get much information, e.g. risk-sensitive learning.

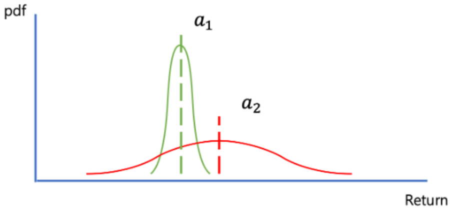
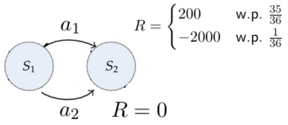


Figure: Risk of Action

Example of Risk-sensitive Learning : Roll two Dice

1. Consider rolling two dice situation. It has two-state/two-action.



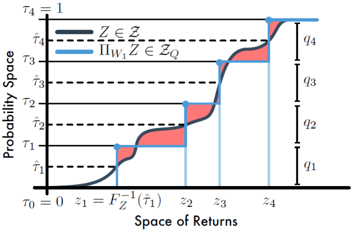
2. When we roll two dice, if the numbers are all 1, then we lose 2000 dollars and otherwise we get 200 dollars.
3. Then the expected immediate reward is

$$ER(x) = \frac{1}{36} \times (-2000) + \frac{35}{36} \times 200 = 138.89.$$

   However the action 1 is risky.
4. Thus we choose action 2 if we want our policy is risk-averse.

Quantile-Regression Distributional RL Framework

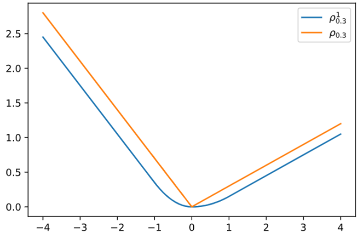
1. Model is $Z_\theta(x,a) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i(x,a)}$ where $\theta_i : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^N$.
2. For quantiles $\tau_1, ..., \tau_N$, $\{\theta_1, ..., \theta_n\}$ is support of the optimal function minimizing 1-Wasserstein distance from Y where $\hat{\tau} = \frac{\tau_{i-1}+\tau_i}{2}$ and $\theta_i = F_Y^{-1}(\hat{\tau}_i)$.
3. Note that sample Wasserstein loss is biased, but the quantile regression loss is unbiased. Thus we can use gradient-based method for finding parameters.



Quantile-Regression Distributional RL Framework

1. The quantile regression of minimizer also minimize the quantile regression loss $L^{\tau}_{QR}(\theta) = \sum_{i=1}^{n} E_{\hat{Z}\sim Z}[\rho_{\hat{\tau}_i}(\hat{Z} - \theta)]$ where $\rho_{\tau}(u) = u(\tau - \delta_{\{u<0\}})$.
2. Since the quantile regression loss is not smooth, construct the smooth loss called quantile Huber loss $\rho^{\kappa}_{\tau}(u) = |\tau - \delta_{\{u<0\}}|L_{\kappa}(u)$ where

$$L_{\kappa}(u) = \begin{cases} \frac{1}{2}u^2 & \text{if } |u| \leq \kappa \\ \kappa(|u| - \frac{1}{2}\kappa) & \text{otherwise.} \end{cases}$$



Implicit Quantile Networks

1. Take $\tau \sim U(0,1)$. Set $Z_\tau = F_Z^{-1}(\tau)$, then $Z_\tau(x,a) \sim Z(x,a)$.
2. Let $\beta : [0,1] \to [0,1]$ be a distortion risk measure. ($\beta = \text{id}$ for risk-neutrality) The distorted expectation $Q_\beta(x,a) = E_\tau[Z_{\beta(\tau)}(x,a)]$.
3. Let $\pi_\beta = \operatorname{argmax}_{a\in\mathcal{A}} Q_\beta(x,a)$ be the risk-sensitive greedy policy. So

$$\tilde{\pi}_\beta(x) = \operatorname{argmax}_{a\in\mathcal{A}} \frac{1}{K}\sum_{k=1}^{K} Z_{\beta(\tilde{\tau}_k)}(x,a)$$

approximates $\pi(x)$. In this paper, use $K = 32$. Note that IQN is not sensitive to K.

Implicit Quantile Networks

4 TD error is

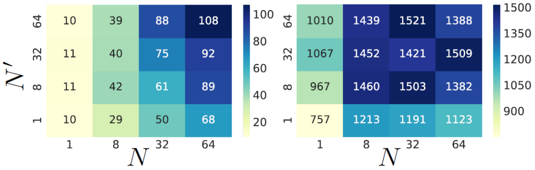
$$\delta_t^{\tau,\tau'} = r_t + \gamma Z_{\tau'}(x_{t+1}, \pi_\beta(x_{t+1})) - Z_\tau(x_t, a_t).$$

Note that this is decorrelated.

5 IQN loss function is Huber loss, i.e.

$$L(x_t, a_t, r_t, x_{t+1}) = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa}(\delta_t^{\tau_i,\tau_j'}).$$

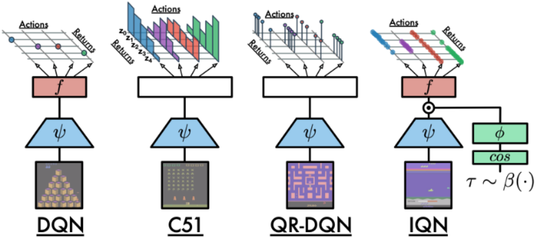
6 Using experimental result, $N = N' = 8$ is enough.



Implicit Quantile Networks

7 Output of QR-DQN is multi-valued for τ (objective is fixed quantile statistics), but output of IQN is $f(\psi(x), \phi(\tau))_a$ (objective is sampled quantile statistics) where ψ is the convolution layers and f is the fully-connected layers.

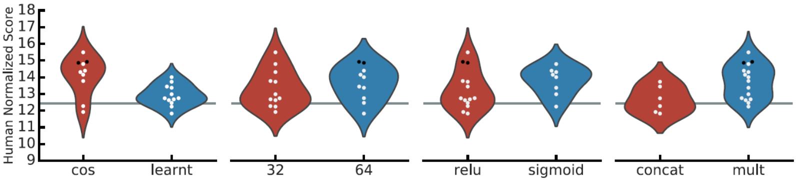
8 Use experience replay, target network, ϵ-greedy policy as same as original DQN.



Implicit Quantile Networks

9 Use $Z_\tau(x,a) \approx f(\psi(x) \odot \phi(\tau))_a$ where

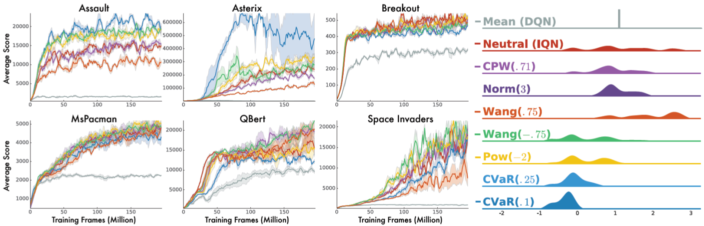
$$\phi_j(\tau) = \mathrm{ReLU}\left(\sum_{i=0}^{n-1} \cos(\pi i \tau) w_{ij} + b_j\right). \quad (n = 64)$$



Risk-Sensitive Reinforcement Learning

1. $CPW(\eta, \tau) = \frac{\tau^\eta}{(\tau^\eta + (1-\tau)^\eta)^{1/\eta}}$
2. $Wang(\eta, \tau) = \Phi(\Phi^{-1}(\tau + \eta))$: risk-averse if $\eta < 0$ and risk-seeking if $\eta > 0$
3. $Pow(\eta, \tau) = \mathbf{1}_{\{\eta \geq 0\}} \tau^{1/(1+|\eta|)} + \mathbf{1}_{\{\eta < 0\}} (1 - (1-\tau)^{1/(1+|\eta|)})$: risk-averse if $\eta < 0$ and risk-seeking if $\eta > 0$
4. $CVaR(\eta, \tau) = \eta\tau$
5. Roughly, in Atari-57 environment, performance rank :

Risk-averse $>$ Risk-neutral $>$ Risk-seeking



Implementation

Algorithm 1 Implicit Quantile Network Loss

Require: N, N', K, κ and functions β, Z

input $x, a, r, x', \gamma \in [0, 1)$

Compute greedy next action

$a^* \leftarrow \arg\max_{a'} \frac{1}{K} \sum_k^K Z_{\tilde{\tau}_k}(x', a'), \quad \tilde{\tau}_k \sim \beta(\cdot)$

Sample quantile thresholds

$\tau_i, \tau'_j \sim U([0, 1]), \quad 1 \leq i \leq N, 1 \leq j \leq N'$

Compute distributional temporal differences

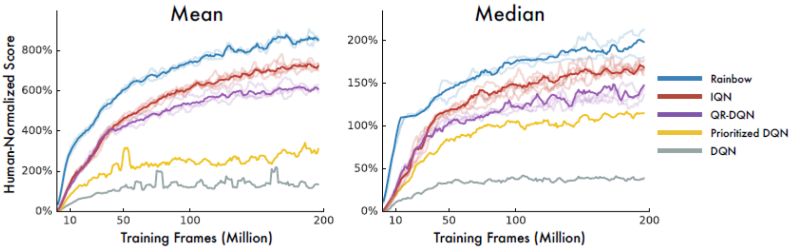
$\delta_{ij} \leftarrow r + \gamma Z_{\tau'_j}(x', a^*) - Z_{\tau_i}(x, a), \quad \forall i, j$

Compute Huber quantile loss

output $\sum_{i=1}^N \mathbb{E}_{\tau'} \left[\rho^\kappa_{\tau_i}(\delta_{ij}) \right]$

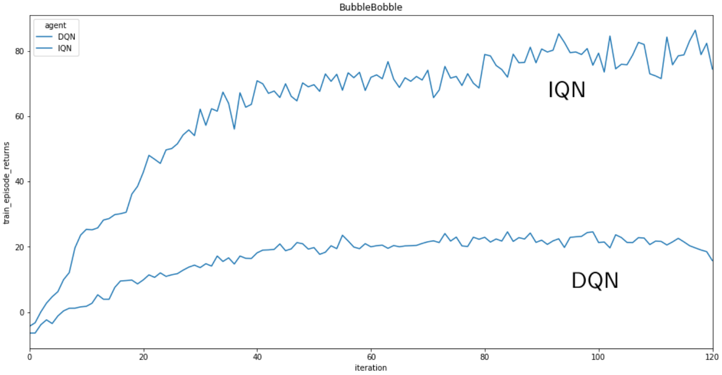
Full Atari-57 Results

1. Compare with DQN, DQN with prioritized experience replay, Rainbow, QR-DQN.
2. IQN has better performance than others except Rainbow.
3. However, Rainbow is a combination of techniques. Rainbow uses C51, the first distributional RL algorithm.



BubbleBobble : Compare with DQN

1. Not random level. Rewards are our custom.
2. IQN setting : $\kappa = 1, N = N' = 32, K = 32$, risk-neutral
3. Learning curves are following :



BubbleBobble : Compare with DQN

1. Dopamine : https://github.com/google/dopamine
2. Playing video (DQN) : https://www.youtube.com/watch?v=QDXTZ2bXii8
3. Playing video (IQN) : https://youtu.be/BEen9P1n-r0