Quantile Regression Hindsight Experience Replay

Qiwei He[1], Liansheng Zhuang[1(✉)], Wei Zhang[2,3], and Houqiang Li[1]

[1] University of Science and Technology of China, Hefei 230027, China
hqw1996@mail.ustc.edu.cn, lszhuang@ustc.edu.cn
[2] Peng Cheng Laboratory, Shenzhen 518000, China
[3] Science and Technology on Electronic Information Control Lab., Chengdu 610036, China

Abstract. Efficient learning in the environment with sparse rewards is one of the most important challenges in Deep Reinforcement Learning (DRL). In continuous DRL environments such as robotic manipulation tasks, Multi-goal RL with the accompanying algorithm Hindsight Experience Replay (HER) has been shown an effective solution. However, HER and its variants typically suffer from a major challenge that the agents may perform well in some goals while poorly in the other goals. The main reason for the phenomenon is the popular concept in the recent DRL works called intrinsic stochasticity. In Multi-goal RL, intrinsic stochasticity lies in that the different initial goals of the environment will cause the different value distributions and interfere with each other, where computing the expected return is not suitable in principle and cannot perform well as usual. To tackle this challenge, in this paper, we propose Quantile Regression Hindsight Experience Replay (QR-HER), a novel approach based on Quantile Regression. The key idea is to select the returns that are most closely related to the current goal from the replay buffer without additional data. In this way, the interference between different initial goals will be significantly reduced. We evaluate QR-HER on OpenAI Robotics manipulation tasks with sparse rewards. Experimental results show that, in contrast to HER and its variants, our proposed QR-HER achieves better performance by improving the performances of each goal as we expected.

Keywords: Deep reinforcement learning · Robotic manipulation · Multi-goal

1 Introduction

Reinforcement learning (RL) [10] is designed to predict and control the agent to accomplish different kinds of tasks from the interactions with the environment by receiving rewards. RL combined with Deep Learning [5] has been shown to be an effective framework in a wide range of domains. However, many great challenges still exist in Deep Reinforcement Learning (DRL), one of which is to

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 824–831, 2020.
https://doi.org/10.1007/978-3-030-63820-7_94

make the agent learn efficiently with sparse rewards. In the environment with sparse rewards, rewards are zero in most transitions and non-zero only when the agent achieves some special states. This makes it extremely difficult for the policy network to inference the correct behavior in the long-sequence decision making. To tackle this challenge, Universal Value Function Approximator (UVFA)[9] is proposed to sample goals from some special states, which extends the definition of value function by not just over states but also over goals. This is equivalent to giving the value function higher dimensional states as parameters for the gain of extra information in different episodes. Lillicrap et al. developed the Deep Deterministic Policy Gradient (DDPG) [6] by utilizing Gaussian Noise for exploration, which significantly improves the performance in continuous control tasks such as manipulation and locomotion. Experience Replay (ER) [7] is a technique that stores and reuses past experiences with a replay buffer. Inspired by the above methods, Hindsight Experience Replay (HER) [1] replaces the desired goals of training trajectories with the sampled goals in the replay buffer and additionally leverage the rich repository of the failed experiences. Utilizing HER, the RL agent can learn to accomplish complex robotic manipulation tasks [8], which is nearly impossible to be solved with general RL algorithms.

Nevertheless, the above methods based on maximizing the expected return still has its problem called intrinsic stochasticity [2]. This phenomenon occurs because the return depends on internal registers and is truly unobservable. On most occasions, the return can be regarded as a constant value function over states, for instance, the maze. In this way, the optimal value of any state should also be constant after long time of training. However, in some occasions, different initial states of the environment will cause significantly different value functions that form the value distributions, which called parametric uncertainty [3]. Furthermore, the MDP process itself does not include past rewards for the current state so that it cannot even distinguish the predictions for different steps of receiving the rewards, which called MDP intrinsic randomness [3]. The above two reasons are the main sources of intrinsic stochasticity.

In the environment with sparse rewards, the intrinsic stochasticity exists mainly due to the parametric uncertainty. The initial goals in Multi-goal RL, as part of the environment, may be completely different from each other and the value distributions are significantly affected by the distribution of goals. However, from the perspective of the expected return, HER and its variants ignore the intrinsic stochasticity caused by the distribution of initial goals and mix the parameters of different value distributions in the training process. In principle, it may cause the instability and degradation of performance especially when the number of goals is large.

Inspired by the above insights, in this paper, we propose a novel method called Quantile Regression Hindsight Experience Replay (QR-HER) to improve the intrinsic stochasticity of the training process in Multi-goal RL. On the basis of Quantile Regression, our key idea is to reduce the interference between different goals by selecting the proper returns for the current goal from the similar goals in the replay buffer. We evaluate QR-HER on the representative OpenAI

Robotics environment and find that QR-HER can achieve better performance compared to HER and its state-of-the-art variant CHER [4]. Furthermore, we infer that the performance improvement of QR-HER is due to the enhancement of the policy for each goal.

2 Preliminary

2.1 Universal Value Function Approximators

UVFA [9] proposed utilizing the concatenation of states $s \in \mathcal{S}$ and goals $g \in \mathcal{G}$ as higher dimensional universal states (s, g) such that the value function approximators $V(s)$ and $Q(s, a)$ can be generalized as $V(s, g)$ and $Q(s, a, g)$. The goals can also be called goal states since in general $\mathcal{G} \subset \mathcal{S}$.

2.2 Multi-goal RL and HER

In Multi-goal RL, random exploration is unlikely to reach the goals. Even if the agent is lucky enough to reach a goal, it does not have enough experience to reach the next one. To address the challenge, [1] proposed Hindsight Experience Replay (HER) including two key techniques, *reward shaping* and *goal relabelling*. The key technique called *reward shaping* is to make the reward function dependent on a goal $g \in G$, such that $r_g : S \times A \times G \to R$. The formula is given by:

$$r_t = r_g\left(s_t, a_t, g\right) = \begin{cases} 0, \text{ if } |s_t - g| < \delta \\ -1, \text{ otherwise} \end{cases} \tag{1}$$

where we can figure out that this trick brings much more virtual returns to support the training. The other technique called *goal relabelling* is to replay each trajectory with different goals sampled from the intermediate states by special schemes. For the transition $(s_t \| g, a_t, r_t, s_{t+1} \| g)$, we will store its hindsight transition $(s_t \| g', a_t, r', s_{t+1} \| g')$ in the replay buffer instead.

3 Quantile Regression Hindsight Experience Replay

3.1 Distributional Mutil-Goal RL Objective

For convenience, we replace the state s with x, the Multi-goal Bellman operator is given as:

$$\mathcal{T}^{\pi} Q(x, a, g) = \mathbb{E}[R(x, a, g)] + \gamma \mathbb{E}_{P,\pi}\left[Q\left(x', a', g\right)\right]. \tag{2}$$

Using the above formula, the distributional Bellman operator is given as:

$$\begin{aligned} \mathcal{T}^{\pi} Z(x, a, g) &:\overset{D}{=} R(x, a, g) + \gamma Z\left(x', a', g\right) \\ x' &\sim P(\cdot|x, a, g), a' \sim \pi\left(\cdot|x', g\right), \end{aligned} \tag{3}$$

where Z denotes the value distribution of Q, $Z :\overset{D}{=} U$ denotes equality of probability laws, that is the random variable Z is distributed according to the sam law as U.

3.2 The Wasserstein Metric

For different goals g, different value distributions Z will be produced. In order to minimize the gap among different value distributions, utilizing the inverse CDF (Cumulative Distribution Function) F^{-1}, we introduce the p-Wasserstein metric, which is written as:

$$W_p(Z_G, Z_{G'}) = \left(\int_0^1 \left| F^{-1}_{(Z_G}(\omega) - F^{-1}_{Z_{G'}}(\omega) \right|^p d\omega \right)^{1/p}, \tag{4}$$

where we use G to represent the initial desired goals which are generated by the environments to be separated from the total sampled goals g, $G \subset g$. G is the main source of intrinsic stochasticity in Multi-goal RL while not the goals generated by hindsight replay. In the cases of RL with expected return, for the current state, we assume that there is only one value distribution and calculate the average probability of its different values. While in quantile distributional RL, we prefer to divide the probability space into different and identical small blocks. For each block, we find out all the corresponding returns of different value distributions utilizing the inverse CDF F^{-1}. Then when making action decisions, we consider all the return values in the blocks to select one from a comprehensive view rather than just averaging.

The quantile distributional parameters and corresponding inverse CDF are not available, so we introduce Quantile Regression network as the function to be learned from the samples. The number of blocks called Quant is fixed and the output of the regression network is the Z vector consisting of the returns of the quantiles. In the Bellman update, the Bellman operator continuously changes the value of the Z vector until convergence. In this way, we can use the Wasserstein Metric to calculate the Quantile Regression loss between the Z vectors of the current state and the next state for the network training, given by:

$$\begin{gathered} \mathcal{L}^{\tau}_{\mathrm{QR}}(\theta) := \mathbb{E}_{\hat{Z} \sim Z} \left[\rho_\tau(\hat{Z} - \theta) \right], \text{ where} \\ \rho_\tau(u) = u \left(\tau - \delta_{\{u<0\}} \right), \forall u \in \mathbb{R} \end{gathered} \tag{5}$$

where θ is the parameter to fit the unknown inverse CDF, for minimizing a step of Bellman update $\int_\tau^{\tau'} \left| F^{-1}(\omega) - \theta \right| d\omega$, it can be deduced mathematically that:

$$\left\{ \theta \in \mathbb{R} | F(\theta) = \left(\frac{\tau + \tau'}{2} \right) \right\}, \tag{6}$$

then if F^{-1} is continuous at $(\tau + \tau')/2$, we can use $\theta = F^{-1}\left((\tau + \tau')/2\right)$ as the unique minimizer.

However, the Quantile Regression loss is not smooth at zero, we will consider use the Huber loss $\rho^\kappa_\tau(u)$ to replace $\rho_\tau(u)$, given by:

$$\mathcal{L}_\kappa(u) = \begin{cases} \frac{1}{2}u^2, & \text{if } |u| \le \kappa \\ \kappa\left(|u| - \frac{1}{2}\kappa\right), & \text{otherwise} \end{cases}, \rho^\kappa_\tau(u) = \left| \tau - \delta_{\{u<0\}} \right| \frac{\mathcal{L}_\kappa(u)}{\kappa} \tag{7}$$

3.3 Return Selection for Multi-goal RL

As demonstrated in the Introduction and Preliminary, the different initial goals mainly cause the intrinsic stochasticity in RL with sparse rewards. When updating the parameters for the current goal, we should exclude interference from other less relevant goals as much as possible. Hence, we propose using the Wasserstein Metric to eliminate the interference of the value distributions of goals with low correlation as the following formula:

$$Z_\theta(x,a,g,G) := \frac{1}{N}\sum_{i=1}^{N}\sum_{G_\epsilon}\delta_{\theta_i(x,a,g,G_\epsilon)}, G_\epsilon \subset G, \frac{W_p(Z_G, Z_{G_\epsilon})}{W_p(Z_G, 0)} < \epsilon, \quad (8)$$

where we only adopt G_ϵ as the subset of G for the return selection to update the value network. Therefore the value distributions significantly different from the current initial goal will not be selected in the replay buffer in a self-attention way. This method is somewhat similar to knowledge distillation in deep learning.

3.4 Algorithm

Utilizing the above derivations, we propose the algorithm for Quantile Regression Multi-goal RL as Algorithm 1.

4 Experiments

4.1 Environments

We evaluate QR-HER and compare QR-HER to HER and its SOTA variant on several challenging robotic manipulation tasks in simulated Mujoco environments Robotics [8] as the Fig. 1 shows, including two kinds of tasks, Fetch robotic arm tasks and Shadow Dexterous Hand tasks. Both two kinds of tasks have sparse binary rewards and follow a Multi-goal RL framework. We choose the most challenging tasks, FetchSlide and HandManipulatePen to carry out our experiments.

4.2 Implementation Details

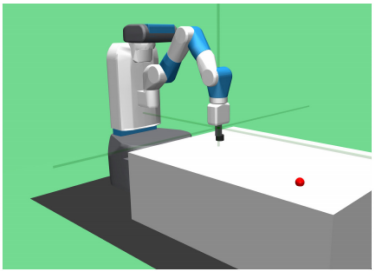
We run the experiments using PyTorch on a machine with 2 14-cores Intel Xeon E5-2690 v4 CPUs and 4 TITAN X(Pascal) GPUs. To make a fair comparison, for all algorithms, each off-policy algorithm is implemented with identical hyperparameters. In the experiments, one epoch is equivalent to 500 episodes with a unique seed(one goal). 10 percent of the episodes are used for testing set to get the mean success rate. The seeds are different in different epochs. Both policy networks and value networks are using MLP with three hidden layers (256,256,256) and optimized using Adam optimizer with critic learning rate of 0.001 and actor learning rate of 3×10^{-4}. The replay buffer size is 10^6 and the batch size is 64. The γ for the Bellman backup is 0.97 and the polyak for target network updating is 0.95. The distributional parameters Quant is choosed from [20,50,100,200,500] and the range of return selection parameter ϵ is [0.1, 0.3].

4.3 Benchmark Performance

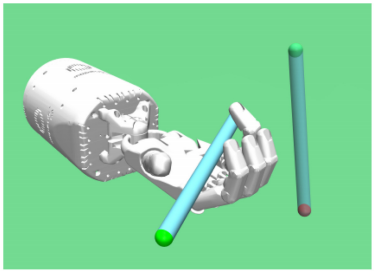
In the benchmark experiments, the better mean success rate represents for better performance to accomplish robotic manipulation tasks. Now we compare the mean success rates in Fig. 1, where the shaded area represents the standard deviation since we use different seeds. Actually, the training process is extremely unstable but we use filters to smooth the curve.

Algorithm 1. Quantile Regression Hindsight Experience Replay

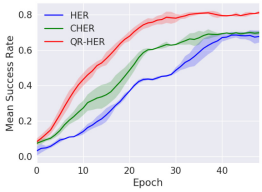
Input: initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, V-function parameters ψ, empty replay buffer $\mathcal{R}$, a strategy $\mathcal{S}$ for sampling goals for replay, distributional parameter Quant(related to ρ_τ^κ), return selection parameter ϵ
Initialize replay buffer $\mathcal{R}$, Set target parameters equal to main parameters $\psi_{\text{targ}} \leftarrow \psi$
for episode = 1,M **do**
  Sample an initial goal G, initial state s_0, $g = G$
  for $t = 0, T-1$ **do**
    Sample an action from
    $a = \text{clip}\left(\mu_\theta(s_t, g) + \delta, a_{Low}, a_{High}\right)$, where $\delta \sim \mathcal{N}$
    Execute a_t in the environment and get next state s_{t+1}
  end for
  for $t = 0, T-1$ **do**
    $r_t := r\left(s_t, a_t, g\right)$
    Store the transition $(s_t \| g, a_t, r_t, s_{t+1} \| g, G)$ in replay buffer $\mathcal{R}$
    Sample a set of additional goals for replay $\mathcal{G} := S$
    for $g' \in \mathcal{G}$ **do**
      $r' := r\left(s_t, a_t, g'\right)$
      Store the transition $(s_t \| g', a_t, r', s_{t+1} \| g', G)$ in $\mathcal{R}$
    end for
  end for
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer $\mathcal{R}$
    for each transition in B **do**
      Calculate the return selection goals set G_ϵ from the initial goals set G
      for each goal in G_ϵ **do**
        Quantile Regression Q-targets updating
        $\mathcal{T}y_q(r, s', g, G) = \mathbb{E}[r(s, g, G_\epsilon)] + \gamma(1-d)\mathbb{E}[V_{\psi_{\text{targ}}}(s', g, G)]$
        Quantile Regression Q-loss and π-loss gradient descent
        $\nabla_\psi \frac{1}{|B|} \sum_{s \in B} [\rho_\tau^\kappa \left(Q_\psi(s, g, G) - \mathcal{T}y_q(r, s', G)\right)]^2$
        $\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \rho_\tau^\kappa Q_{\phi,1}(s, g, \tilde{a}_\theta(s), G)$
        Target value network updating
        $\psi_{\text{targ}} \leftarrow \rho\psi_{\text{targ}} + (1-\rho)\psi$
      end for
    end for
  end for
end for



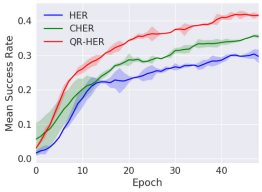
(a) FetchSlide



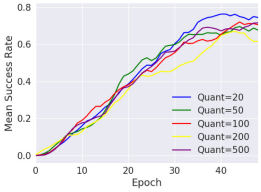
(b) HandManipulatePen



(c) FetchSlide



(d) HandManipulatePen-Rotate



(e) Quant

Fig. 1. The open AI robotics experiments for QR-HER

The agent trained with QR-HER shows the best benchmark performance at the end of the training. The value of Quant is the key hyper-parameter of QR-HER, as is shown in Fig. 1. The performance at Quant = 20 is 20% higher than at Quant = 200. Our conclusion is that the agent has its best performance when the Quant is about half of the number of goals(epochs).

4.4 Performance Analysis of Each Goal

According to our assumption, the quantile regression method with return selection is supposed to reduce the interference between different goals to improve the performance of each goal. The corresponding result is shown in Table 1. From the table, we can infer that QR-HER improves the overall performance through the optimization of the policy of each goal as we expected.

Table 1. Final success rates in HandManipulatePenRotate with different goals(seeds)

Method	seed = 0	1000	10000	20000	100000
HER	0.315	0.307	0.282	0.293	0.296
CHER	0.346	0.325	0.336	0.311	0.325
QR-HER (Ours)	**0.457**	**0.422**	**0.434**	**0.418**	**0.420**

5 Summary

The main contributions of this paper are summarized as follows: (1) We raise the issue of performance instability and performance degradation in Multi-goal RL, and attribute the cause to intrinsic stochasticity; (2) We introduce Wasserstein Metric and Quantile Regression into Multi-goal RL to derive QR-HER; (3) We show that QR-HER can exceed HER and its variants to achieve the state-of-the-art performance on OpenAI Robotics; (4) We show that QR-HER improves the performance of each goal to become the powerful evidence for the correctness of our theory.

Acknowledgments. This work was supported in part to Dr. Liansheng Zhuang by NSFC under Grant contract No. 61976199, in part to Dr. Houqiang Li by NSFC under Grant contract No. 61836011.

References

1. Andrychowicz, M., et al.: Hindsight experience replay. In: Advances in Neural Information Processing Systems, pp. 5048–5058 (2017)
2. Bellemare, M.G., Dabney, W., Munos, R.: A distributional perspective on reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 449–458. JMLR. org (2017)
3. Dabney, W., Rowland, M., Bellemare, M.G., Munos, R.: Distributional reinforcement learning with quantile regression (2017)
4. Fang, M., Zhou, T., Du, Y., Han, L., Zhang, Z.: Curriculum-guided hindsight experience replay. In: Advances in Neural Information Processing Systems, pp. 12602–12613 (2019)
5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
6. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
7. Lin, L.J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn. **8**(3–4), 293–321 (1992)
8. Plappert, M., et al.: Multi-goal reinforcement learning: challenging robotics environments and request for research. arXiv preprint arXiv:1802.09464 (2018)
9. Schaul, T., Horgan, D., Gregor, K., Silver, D.: Universal value function approximators. In: International Conference on Machine Learning, pp. 1312–1320 (2015)
10. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. MIT press (2018)