Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

QUANTILE-BASED POLICY OPTIMIZATION FOR REINFORCEMENT LEARNING

Jinyang Jiang
Yijie Peng

Guanghua School of Management
Peking University
5 Yiheyuan Road
Beijing 100871, P. R. CHINA

Jiaqiao Hu

Department of Applied Mathematics and Statistics
Stony Brook University
100 Nicolls Road
Stony Brook, NY 11794-3600, USA

ABSTRACT

Classical reinforcement learning (RL) aims to optimize the expected cumulative rewards. In this work, we consider the RL setting where the goal is to optimize the quantile of the cumulative rewards. We parameterize the policy controlling actions by neural networks and propose a novel policy gradient algorithm called Quantile-Based Policy Optimization (QPO) and its variant Quantile-Based Proximal Policy Optimization (QPPO) to solve deep RL problems with quantile objectives. QPO uses two coupled iterations running at different time scales for simultaneously estimating quantiles and policy parameters. Our numerical results demonstrate that the proposed algorithms outperform the existing baseline algorithms under the quantile criterion.

1 INTRODUCTION

In recent years, deep reinforcement learning (RL) has made significant achievements in games (Mnih et al. 2015; Silver et al. 2016), robotic control (Levine et al. 2016), recommendation (Zheng et al. 2018) and other fields. RL formulates a complex sequential decision-making task as a Markov Decision Process (MDP) and attempts to find an optimal policy by interacting with the environment. In the classical RL framework, the goal is to optimize an expected cumulative reward. However, expectation only reflects the average value of a distribution, but not the tail behavior. The tail of a distribution may capture catastrophic outcomes. For example, the joint defaults of many subprime mortgages led to the 2008 financial crisis, and in the post-crisis era, the Basel accord requires major financial institutes to maintain a minimal capital level for sustaining the loss under extreme market circumstances. In financial management, quantiles, also known as value-at-risk (VaR), can be directly translated into the minimal capital requirement.

Risk measures have been introduced into RL either in the forms of constraints or as the objectives, which are referred as risk-sensitivity RL in the literature. The tail behavior of the distribution can be better captured by risk measures, such as VaR and conditional value-at-risk (CVaR). With risk measures as the objective function of RL, a well-trained agent can be expected to perform more robustly under extreme events.

In this paper, we consider an RL setting where the goal is to optimize the quantiles of the cumulative rewards. In financial investment, investors may want to obtain the highest return subject to an acceptable level of risk; in system science, engineers may want to improve the system's most conservative output under extreme environment conditions. In Figure.1, although the mean values of the two distributions are the same, the 10%-quantile of distribution 1 is significantly larger than that of distribution 2. If the returns of two portfolios follow distributions 1 and 2, then it requires more capital to avoid bankruptcy in the presence of 90% possible returns for the second portfolio. For some distributions, e.g. the Cauchy distribution, the

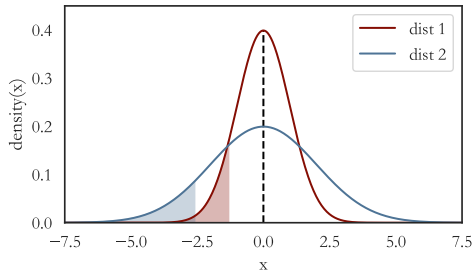

Figure 1: Probability density plots of two normal distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(0,2)$. The right boundaries of shaded areas represent the value of 10%-quantile.

expectation may not exist so that it cannot be used as a performance measure in this situation. In contrast, quantiles are always well-defined.

We parameterize the policy controlling actions by neural networks. For RL learning problems where the dimension of the policy parameter is typically very high, gradient-based optimization provides a feasible approach. The gradient of quantiles can be expressed as a function of gradients of the distribution and quantile value. However, unlike an expectation whose gradient can be estimated in a single simulation trajectory using techniques such as the likelihood ratio method, the gradient estimation for quantiles is much more complicated. We propose a novel policy gradient algorithm named Quantile-Based Policy Optimization (QPO) and its variant Quantile-Based Proximal Policy Optimization (QPPO) to solve deep RL problems with quantile objectives. QPO uses two coupled iterations running at different time scales for simultaneously estimating quantiles and policy parameters. Our proposed algorithms have been applied to financial investment examples, and numerical results demonstrate that the proposed algorithms outperform the existing baseline algorithms under the quantile criterion.

2 RELATED WORK

2.1 Mean-based RL

In RL with a mean-based criterion, there are two primary categories of methods. The first category is value-based method, which learns a Q function and selects the action with the best value. In deep RL, pioneer work includes Deep Q-learning (DQN) and its variants (Mnih et al. 2015; Hessel et al. 2018). The second category is based on policy optimization, where the policy is parameterized by certain basis functions and optimized by stochastic gradient ascent. REINFORCE is an early policy gradient algorithm (Williams 1992) . Trust Region Policy Optimization (TRPO) introduces the importance sampling technique into RL to improve data utilization efficiency (Schulman et al. 2015). By simplifying TRPO, Proximal Policy Optimization (PPO) achieves a significant improvement through optimizing a clipped surrogate objective and has become a commonly used baseline algorithm at present (Schulman et al. 2017). There are also some algorithms that combine the two categories, such as Deep Deterministic Policy Gradient (DDPG) and Soft Actor-Critic (SAC) (Lillicrap et al. 2015; Haarnoja et al. 2018).

2.2 Gradient Estimation of Risk Measure

Gradient estimation of risk measures has been studied actively. Classic gradient estimation problem considers expectation (Fu 2006), whereas estimating gradients of quantile and CVaR is much more complicated. To address the difficulties, different methods, such as infinite perturbation analysis (Hong 2009; Jiang and Fu 2015), kernel estimation (Liu and Hong 2009; Hong and Liu 2009), and measure-valued differentiation (Heidergott and Volk-Makarewicz 2016), have been developed. Recently, Glynn et al. (2021) use generalized likelihood ratio estimators to estimate the gradient for a rather general distortion risk measure. However,

these methods would require analytic forms of transition probabilities or rewards in the MDP, which is typically not satisfied in the RL setting. A well-known black-box gradient estimation approach is Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall 1992). However, tuning parameters such as the perturbation size and step size in optimization require care and it is typically hard to apply SPSA to high-dimensional optimization such as our deep RL problems where the policy is parameterized by neural networks.

2.3 Risk-sensitive RL

Risk-sensitive RL has recently gained steam. One can use a risk measure as the objective function of RL. For example, expected exponential utility is used as the objective in Borkar (2001); Petrik and Subramanian (2012) and Tamar et al. (2014) have studied CVaR-based objectives; Prashanth and Ghavamzadeh (2013) aim to optimize several variance-related risk measures by SPSA and a smooth function approach; Prashanth et al. (2016) apply SPSA to optimize an objective function in the cumulative prospect theory.

One can also impose a risk measure as the constraint of an RL problem. For example, dynamic and time-consistent risk constraints are considered in Chow and Pavone (2013); Borkar and Jain (2014) use CVaR as the constraint; Chow et al. (2017) develop policy gradient and actor-critic algorithms under VaR and CVaR constraints. A Lagrangian approach has been used to solve the RL problem subject to some risk measures (Bertsekas 1997). Tuning a multi-scale iterative algorithm to update Lagrange multipliers requires care.

To the best of our knowledge, our work is the first to propose a two-time-scale iterative algorithm to optimize quantiles, which can be applied to large-scale optimization for policy parameterized by neural networks in deep RL problems.

The rest of the paper is organized as follows: In Section 3, we describe the MDP settings, the classical framework of policy gradient algorithms, and our quantile optimization framework for RL. Section 4 introduces our new algorithms called QPO and its variant QPPO. In Section 5, we present theoretical results for QPO. Numerical results are presented in Section 6. We conclude the paper and discuss future direction in Section 7.

3 PROBLEM FORMULATION & PRELIMINARIES

3.1 Markov Decision Process

A Markov Decision Process (MDP) can be represented as a 5-tuple $(\mathscr{S}, \mathscr{A}, p, u, \eta)$, where $\mathscr{S}$ and $\mathscr{A}$ are state and action spaces, respectively; $p(s'|s,a)$ is the transition probability; $u(s',a,s)$ is the reward function; $\eta \in (0,1)$ is the reward discount factor. One may expect to optimize a parameterized policy function $\pi(a|s;\theta)$, which is a distribution of the action conditional on the state, through the interaction with the MDP system. We denote the state and action at time $t \in \{0,1,2,\cdots,T\}$ by s_t and a_t, where $s_t \sim p(\cdot|s_{t-1},a_{t-1})$ and $a_t \sim \pi(\cdot|s_t;\theta)$. Then the trajectory generated by the MDP can be defined as $\tau = \{s_0,a_0,s_1,\cdots,a_{T-1},s_T\} \sim \Pi(\cdot;\theta)$. Here we have $\Pi(\tau;\theta) = p(s_0)\prod_{t=0}^{T-1}\pi(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t)$, where s_0 is the initial state. The total reward of the trajectory is denoted as $R = \sum_{t=0}^{T-1}\eta^t u(s_{t+1},a_t,s_t) = U(\tau)$, which follows the distribution $F_R(\cdot;\theta)$. In an episode, a complete simulation trajectory of the action and state is generated by the interaction between the agent and environment.

3.2 Mean-based Criterion for RL

In the classical setting, the objective function of RL algorithms is to maximize the expected cumulative reward:

$$\max_{\theta\in\Theta}\mathbb{E}_{R\sim F_R(\cdot;\theta)}[R] = \max_{\theta\in\Theta}\mathbb{E}_{\tau\sim\pi(\cdot;\theta)}[U(\tau)]. \tag{1}$$

Policy gradient methods, an important class of RL algorithms, solve problem (1) by the stochastic gradient ascent. The key for implementing policy gradient methods is to estimate gradient, and the likelihood ratio method is a popular stochastic gradient estimation technique. Specifically, the gradient can be expressed as below:

$$\begin{aligned}\nabla_\theta \mathbb{E}[U(\tau)] &= \nabla_\theta \int_{\Omega_\tau} U(\tau)\Pi(\tau;\theta)d\tau = \int_{\Omega_\tau} U(\tau)\frac{\nabla_\theta \Pi(\tau;\theta)}{\Pi(\tau;\theta)}\Pi(\tau;\theta)d\tau \\ &= \mathbb{E}\left[U(\tau)\sum_{t=0}^{T-1}\nabla_\theta \log \pi(a_t|s_t;\theta)\right], \end{aligned} \tag{2}$$

where the interchange of the gradient and integral in the second equality can be justified by the dominated convergence theorem. The term $U(\tau)\sum_{t=0}^{T-1}\nabla_\theta \log \pi(a_t|s_t;\theta)$ inside the expectation on the right hand side of (2) is an unbiased stochastic gradient estimation of $\nabla_\theta \mathbb{E}[U(\tau)]$.

3.3 Quantile-based Criterion for RL

For a given probability level $\alpha \in (0,1)$, the α-quantile for distribution $F_R(\cdot;\theta)$ is defined as

$$q(\alpha;\theta) = \arg\inf\{r : P(R(\theta) \le r) = F_R(r;\theta) \ge \alpha\}.$$

We assume that $F_R(r;\theta)$ is continuously differentiable on $\mathbb{R}$, i.e. $F_R(r;\theta) \in C^1(\mathbb{R})$, so that the α-quantile can be written as an inverse of the distribution function $q(\alpha;\theta) = F_R^{-1}(\alpha;\theta)$. Our goal is to maximize the α-quantile of the distribution on a compact convex set $\Theta \subset \mathbb{R}^m$, i.e.,

$$\max_{\theta\in\Theta} q(\alpha;\theta) = \max_{\theta\in\Theta} F_R^{-1}(\alpha;\theta). \tag{3}$$

To solve problem (3), we consider a stochastic gradient ascent method. By definition, $F_R(q(\alpha;\theta);\theta) = \alpha$. Taking gradients on both sides with respect to θ, we have

$$\nabla_\theta q(\alpha;\theta) = -\frac{\nabla_\theta F_R(r;\theta)}{f_R(r;\theta)}\bigg|_{r=q(\alpha;\theta)}. \tag{4}$$

There are two difficulties in obtaining a single-run unbiased stochastic gradient estimator:

- The right hand side of equality (4) contains the α-quantile itself. The quantile would change in the sequential update of θ.
- The density function $f_R(\cdot;\theta)$ of the cumulative reward usually does not have an analytical form. We have $f_R(r;\theta) = \frac{\partial}{\partial r}F_R(r;\theta) = \frac{\partial}{\partial r}\mathbb{E}[\mathbf{1}\{R \le r\}]$, but the gradient and expectation operators cannot be interchanged since $\mathbf{1}\{R \le r\}$ is discontinuous.

4 QUANTILE-BASED POLICY OPTIMIZATION

In this section, we propose a new on-policy RL algorithm, named Quantile-based Policy Optimization (QPO) algorithm and its variant Quantile-based Proximal Policy Optimization (QPPO) algorithm to solve the MDP with quantile criterion (3). The computation complexity of our algorithms will also be discussed.

4.1 Quantile Optimization of REINFORCE Style

To optimize quantiles, Hu et al. (2021) proposed a three-time-scale iterative algorithm, which includes three coupled recursions to estimate gradient (4) and α-quantile, and do the gradient ascent. Although this algorithm performs well for relatively simple stochastic systems, it is not best suited for RL settings. First, it is typically difficult to tune step sizes with three time scales. In the RL context, the dimension of

θ could be extremely large, and the simulation may be very costly. Less hyper-parameters and higher data efficiency of the algorithm are critical for successful RL applications. Second, the density estimation in the denominator of (4) is generally difficult, because the transition probability and reward function may not be available in RL. Therefore, we provide a simpler algorithm with only two time scales, which does not require estimating $f_R(r;\theta)$.

Since the denominator of the α-quantile in equality (4) is non-negative, the ascent direction can be simplified as $d = -\nabla_\theta F_R(r;\theta)\big|_{r=q(\alpha;\theta)}$ in searching for the optimum. The likelihood ratio technique in (2) can be applied analogously to derive the gradient $\nabla_\theta F_R(r;\theta)$ as below:

$$\begin{aligned}\nabla_\theta F_R(r;\theta) &= \nabla_\theta \mathbb{E}[\mathbf{1}\{R \le r\}] = \nabla_\theta \mathbb{E}[\mathbf{1}\{U(\tau) \le r\}] = \nabla_\theta \int_{\Omega_\tau} \mathbf{1}\{U(\tau) \le r\}\Pi(\tau;\theta)d\tau \\ &= \mathbb{E}[\mathbf{1}\{U(\tau) \le r\}\nabla_\theta \log \Pi(\tau;\theta)] = \mathbb{E}\left[\mathbf{1}\{U(\tau) \le r\}\sum_{t=0}^{T-1}\nabla_\theta \log \pi(a_t|s_t;\theta)\right] \\ &\approx \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{1}\{U(\tau^n) \le r\}\sum_{t=0}^{T-1}\nabla_\theta \log \pi(a_t^n|s_t^n;\theta).\end{aligned}$$

Denote

$$D(\tau;\theta,r) = -\mathbf{1}\{U(\tau) \le r\}\sum_{t=0}^{T-1}\nabla_\theta \log \pi(a_t|s_t;\theta).$$

We propose a two-time-scale iterative algorithm as follows:

$$q_{k+1} = q_k + \beta_k(\alpha - \mathbf{1}\{U(\tau^k) \le q_k)\}), \tag{5}$$

$$\theta_{k+1} = \varphi(\theta_k + \gamma_k D(\tau^k;\theta_k,q_k)), \tag{6}$$

where $\varphi(\cdot)$ is a projection function to guarantee $\theta_k \in \Theta$. Recursion (5) is used to track the quantile of the current policy $\pi(\cdot|\cdot;\theta_k)$, i.e., a one-step search for the root of $F_R(q;\theta_k) = \alpha$.

In practice, we divide $D(\tau^k;\theta_k,q_k)$ by the estimator of $f_R(q_k;\theta_k)$ to speed up searching for the optimum. Here we offer two feasible approaches. One is to estimate the density by the empirical reward distribution. Using the well-known Kernel Density Estimation (KDE) approach (Scott 2015; Silverman 2018), we may construct the reward density function by a batch of episodes between two steps updating the parameters. An alternative approach is to approximate the indicator $\mathbf{1}\{\cdot\}$ by a smooth function that can be directly differentiated. For example, we have

$$f_R(r;\theta) = \frac{\partial \mathbb{E}[\mathbf{1}\{R \le r\}]}{\partial r} \approx \frac{\partial \mathbb{E}[\sigma(r;R)]}{\partial r} = \mathbb{E}\left[\sigma'(r;R)\right],$$

where $\sigma(\cdot)$ is the sigmoid function defined as

$$\sigma(r;R) = \frac{1}{1+e^{-(r-R)}}.$$

Then $\frac{1}{N}\sum_{n=0}^{N-1}\sigma'(q_k;U(\tau_k^n))$ can be used as an estimator of $f_R(q_k;\theta_k)$, which can be used as a factor of the step size γ_k to facilitate convergence in implementation. We present the pseudo code of QPO in Algorithm 1 as follows.

4.2 Quantile Optimization of PPO Style

To improve data utilization efficiency and robustness, we propose a variant of our QPO algorithm by using an importance sampling technique inspired by PPO. We consider a pair of policy networks $\pi(\cdot|\cdot;\theta)$ and

Algorithm 1 Quantile-Based Policy Optimization (QPO)

Input: Policy network $\pi(\cdot|\cdot;\theta)$, quantile parameter $\alpha \in (0,1)$, and batch size N.
Initialize: Policy parameter $\theta_0 \in \Theta$ and quantile estimator $q_0 \in \mathbb{R}$.
for $k = 0, \cdots, K-1$ **do**
  Generate N episodes $\{\tau_n^k\}_{n=0}^{N-1}$ following policy $\pi(\cdot|\cdot;\theta_k)$;
  $q_{k+1} \leftarrow q_k + \beta_k\big(\alpha - \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{1}\{U(\tau_n^k) \le q_k\}\big)$;
  $\theta_{k+1} \leftarrow \varphi\big(\theta_k + \gamma_k \frac{1}{N}\sum_{n=0}^{N-1} D(\tau_n^k;\theta_k,q_k)\big)$.
end for
Output: Trained policy network $\pi(\cdot|\cdot;\theta_K)$.

$\pi(\cdot|\cdot;\tilde{\theta})$, where only $\pi(\cdot|\cdot;\tilde{\theta})$ is used to interact with the environment. In each epoch, a batch of episodes are generated by following policy $\pi(\cdot|\cdot;\tilde{\theta})$. Then, the target policy $\pi(\cdot|\cdot;\theta)$ is updated according to the episodes generated by $\pi(\cdot|\cdot;\tilde{\theta})$, and the parameter $\tilde{\theta}$ is updated by the latest value of θ.

Denote the importance sampling ratio as

$$\rho(\tilde{a}_t,\tilde{s}_t) = \frac{\pi(\tilde{a}_t|\tilde{s}_t;\theta)}{\pi(\tilde{a}_t|\tilde{s}_t;\tilde{\theta})}, \quad \rho(\tilde{\tau}) = \frac{\Pi(\tilde{\tau};\theta)}{\Pi(\tilde{\tau};\tilde{\theta})} = \prod_{t=0}^{T-1}\rho(\tilde{a}_t,\tilde{s}_t), \tag{7}$$

where the trajectory $\tilde{\tau} = \{\tilde{s}_0,\tilde{a}_0,\cdots,\tilde{s}_{T-1},\tilde{a}_{T-1},\tilde{s}_T\}$ is generated by $\pi(\cdot|\cdot;\tilde{\theta})$. By noticing $\nabla_x f(x) = f(x)\nabla_x \log f(x)$ and $F_R(r;\theta) = \mathbb{E}_{\tilde{\tau}\sim\Pi(\cdot;\tilde{\theta})}[\mathbf{1}\{U(\tilde{\tau}) \le r\}\rho(\tilde{\tau})]$, we have

$$\nabla_\theta F_R(r;\theta) = \mathbb{E}_{\tilde{\tau}\sim\Pi(\cdot;\tilde{\theta})}[\rho(\tilde{\tau})\mathbf{1}\{U(\tilde{\tau}) \le r\}\nabla_\theta \log\Pi(\tilde{\tau};\theta)].$$

Thus, we can rewrite recursions (5) and (6) as follows:

$$q_{k+1} = q_k + \beta_k(\alpha - \rho(\tilde{\tau}^k)\mathbf{1}\{U(\tilde{\tau}^k) \le q_k)\}), \tag{8}$$

$$\theta_{k+1} = \varphi(\theta_k + \gamma_k\rho(\tilde{\tau}^k)D(\tilde{\tau}^k;\theta_k,q_k)). \tag{9}$$

In practice, the gradient term $\rho(\tilde{\tau}^k)D(\tilde{\tau}^k;\theta_k,q_k)$ can be calculated stepwisely, which is computationally more efficient. If $-\mathbf{1}\{U(\tilde{\tau}^k) \le q_k\}$ is used as the advantage function A_t^k for each action $\tilde{a}_t^k$ in the trajectory $\tilde{\tau}^k$, then the searching direction in recursion (9) can be replaced by

$$\hat{d}_k = \sum_{t=0}^{T-1}\rho(\tilde{a}_t^k,\tilde{s}_t^k)A_t^k\nabla_\theta\pi(\tilde{a}_t^k|\tilde{s}_t^k;\theta_k),$$

which can be obtained by differentiating the following surrogate objective:

$$\hat{\mathbb{E}}\left[\sum_{t=0}^{T-1}\rho(\tilde{a}_t,\tilde{s}_t)A_t\right] = \hat{\mathbb{E}}\left[-\sum_{t=0}^{T-1}\rho(\tilde{a}_t,\tilde{s}_t)\mathbf{1}\{U(\tilde{\tau}^k) \le q_k\}\right].$$

To constrain the difference between θ and $\tilde{\theta}$, we introduce a clip function $\mathrm{clip}(x,x^-,x^+)$, where x^- and x^+ are the lower and upper truncation bounds. The clipped surrogate objective is

$$\hat{\mathbb{E}}_{\tilde{\tau}\sim\Pi(\cdot;\tilde{\theta})}\left[\sum_{t=0}^{T-1}\min\{\rho(\tilde{a}_t,\tilde{s}_t)A_t, \mathrm{clip}(\rho(\tilde{a}_t,\tilde{s}_t),1-\varepsilon,1+\varepsilon)A_t\}\right]. \tag{10}$$

The pseudo code of QPPO is presented in Algorithm 2. To achieve a more stable performance, one may use an extra baseline network $B(\tilde{s}_t|w)$ and take $A_t = -\mathbf{1}\{U(\tilde{\tau}^k) \le q_k\} - B(\tilde{s}_t|w)$ as a new advantage function, where $B(\tilde{s}_t|w)$ is updated by minimizing the MSE of $-\mathbf{1}\{u(\tilde{\tau}^k) \le q_k\}$.

Algorithm 2 Quantile-Based Proximal Policy Optimization (QPPO)

Input: Policy network $\pi(\cdot|\cdot;\theta)$ and $\pi(\cdot|\cdot;\tilde{\theta})$, quantile parameter $\alpha \in (0,1)$, and batch size N.
Initialize: Policy parameter θ_0^0, $\tilde{\theta}_0 \in \Theta$, and quantile estimator $q_0^0 \in \mathbb{R}$.
for $k = 0, \cdots, K-1$ **do**
  Generate N episodes $\{\tilde{\tau}_n^k\}_{n=0}^{N-1}$ following policy $\pi(\cdot|\cdot;\tilde{\theta}_k)$;
  for $j = 0, \cdots, J-1$ **do**
    Use equality (7) to calculate $\{\rho(\tilde{\tau}_n^k)\}_{n=0}^{N-1}$;
    Update q_k^j to q_k^{j+1} by recursion (8);
    Update θ_k^j to θ_k^{j+1} by maximizing clipped surrogate objective (10);
  end for
  $q_{k+1}^0 \leftarrow q_k^{J-1}$, $\theta_{k+1}^0 \leftarrow \theta_k^{J-1}$, $\tilde{\theta}_{k+1} \leftarrow \theta_{k+1}^0$.
end for
Output: Trained policy network $\pi(\cdot|\cdot;\theta_K^0)$.

4.3 Computation Complexity

We have proposed two new RL algorithms based on the quantile maximization criterion, which are counterparts of two corresponding mean-based algorithms, i.e., REINFORCE and PPO. The computational cost of calculating the stochastic gradient estimate for the policy parameter is comparable to that of the mean-based algorithms. The main difference is that our algorithms need to estimate the quantile of the current policy. Recursion (5) or (8) is a one-step iteration for the one-dimensional root searching problem, which is computationally very cheap. Compared with the corresponding mean-based algorithm, the increase in computation burden of QPO and QPPO is negligible.

5 CONVERGENCE ANALYSIS

In this section, we establish the global convergence of the QPO algorithm. Let $(\Omega, \mathscr{F}, P)$ be a probability space. Define the filtration generated by our algorithm $\mathscr{F}_k = \{\theta_0, q_0, \cdots, \theta_k, q_k\}$ for $k = 0, 1, \cdots$. Here we introduce some assumptions before the analysis.

Assumption 1 For any $\alpha \in (0,1)$, $q(\alpha;\theta) \in C^1(\Theta)$.

Assumption 2 $\nabla_\theta F_R(q;\theta)$ is Lipschitz continuous with respect to both q and θ, i.e. there exists a constant C such that $\|\nabla_\theta F_R(q_1;\theta_1) - \nabla_\theta F_R(q_2;\theta_2)\| \le C\|(q_1,\theta_1) - (q_2,\theta_2)\|$ for any $(q_i,\theta_i) \in \mathbb{R}\times\Theta$, $i = 1,2$.

Assumption 3 The step-size sequences $\{\gamma_k\}$ and $\{\beta_k\}$ satisfy
(a) $\gamma_k > 0$, $\sum_{k=0}^{\infty}\gamma_k = \infty$, $\sum_{k=0}^{\infty}\gamma_k^2 < \infty$; (b) $\beta_k > 0$, $\sum_{k=0}^{\infty}\beta_k = \infty$, $\sum_{k=0}^{\infty}\beta_k^2 < \infty$; (c) $\gamma_k = o(\beta_k)$.

Assumption 4 The log gradient of the neural network output with respect to θ is bounded, i.e., for any state-action pair (s,a) and parameter θ, $\|\nabla_\theta \log\pi(a|s;\theta)\| < \infty$.

Assumption 1 requires that the objective function is smooth enough, which is commonly assumed in continuous optimization. Assumptions 2 and 3 are standard in stochastic approximation analysis. Assumption 4 is necessary to avoid computational overflow.

Since $\gamma_k = o(\beta_k)$ in Assumption 3 (c), recursion (5) updates on a faster scale than recursion (6). Let

$$g_1(q,\theta) = \alpha - F_R(q;\theta), \quad g_2(q,\theta) = -\nabla_{\theta'} F_R(q;\theta')\big|_{\theta'=\theta},$$

and we expect them to track a coupled ODE:

$$\dot{q}(t) = g_1(q(t),\theta(t)), \quad \dot{\theta}(t) = \tilde{\varphi}(g_2(q(t),\theta(t))), \tag{11}$$

where $\tilde{\varphi}(\cdot)$ is a projection function satisfying

$$\tilde{\varphi}(g_2(q(t),\theta(t))) = g_2(q(t),\theta(t)) + p(t),$$

where $p(t) \in -C(\theta(t))$ is the vector with the smallest norm needed to keep $\theta(t)$ in Θ, and $C(\theta)$ is the normal cone to Θ at θ. When $\theta(t) \in \partial\Theta$, $\tilde{\varphi}(\cdot)$ projects the gradient onto $\partial\Theta$.

Intuitively, $\theta(t)$ can be viewed as static for analyzing the dynamic of the process $q(t)$. Suppose that for some constant $\bar{\theta} \in \Theta$, the unique global asymptotically stable equilibrium of the ODE

$$\dot{q}(t) = g_1(q(t), \bar{\theta}) \tag{12}$$

is $q(\alpha; \bar{\theta})$. Recursion (6) can be viewed as tracking the ODE

$$\dot{\theta}(t) = \tilde{\varphi}(g_2(q(\alpha; \theta(t)), \theta(t))). \tag{13}$$

If $\theta^* = \arg\max_{\theta \in \Theta} q(\alpha; \theta)$ is the unique global asymptotically stable equilibrium of this ODE, then the global convergence of QPO to θ^* can be proved. Therefore, we first establish the unique global asymptotically stable equilibriums for ODE (12) and (13).

Lemma 1 $q(\alpha; \bar{\theta})$ is the unique global asymptotically stable equilibrium of ODE (12) for all $\bar{\theta} \in \Theta$.

Lemma 2 If $q(\alpha; \theta)$ is strictly convex on Θ, then θ^* is the unique global asymptotically stable equilibrium of ODE (13).

To prove that recursions (5) and (6) track coupled ODE (11), we apply the convergence theorem of the two-scale stochastic approximation as below:

Theorem 1 (Borkar 1997) Consider two coupled recursions:

$$q_{k+1} = q_k + \beta_k(g_1(q_k, \theta_k) + \varepsilon_{1,k}),$$
$$\theta_{k+1} = \varphi(\theta_k + \gamma_k(g_2(q_k, \theta_k) + \varepsilon_{2,k})),$$

where φ is a projection function, g_1 and g_2 are Lipschitz continuous, $\{\beta_k\}$ and $\{\gamma_k\}$ satisfy Assumption 3, $\{\varepsilon_{1,k}\}$ and $\{\varepsilon_{2,k}\}$ are random variable sequences satisfying

$$\sum_k \beta_k \varepsilon_{1,k} < \infty, \ \sum_k \gamma_k \varepsilon_{2,k} < \infty, \quad a.s.$$

If ODE (11) has a unique global asymptotically stable equilibrium $\lambda(\bar{\theta})$ for each $\bar{\theta} \in \Theta$, then the coupled recursions converge to the unique global asymptotically stable equilibrium of the ODE $\dot{\theta}(t) = \tilde{\varphi}(g_2(\lambda(\theta(t)), \theta(t)))$ a.s. conditional on that the sequence $\{q_k\}$ is bounded.

Next we show that the sequence $\{q_k\}$ is almost surely bounded under our assumptions. Then the conditions in Theorem 1 can be verified in Theorem 2.

Lemma 3 If Assumption 1 and 3(b) hold, then the sequence $\{q_k\}$ generated by recursion (5) is bounded w.p.1, i.e., $\sup_k |q_k| < \infty$ w.p.1.

Theorem 2 If Assumption 1-4 hold and $q(\alpha; \theta)$ is strictly convex on Θ, then the sequence $\{\theta_k\}$ generated by recursions (5) and (6) converges to the unique optimal solution $\{q(\alpha; \theta^*), \theta^*\}$ of problem (3) w.p.1.

The convexity assumption is for guaranteeing convergence to the global optimum; otherwise, it can be relaxed. The detailed proof of the lemmas and theorems mentioned above will be presented in a full journal version.

6 EXPERIMENTS

In this section, we conduct simulation experiments on different RL tasks to compare our QPO and QPPO with baseline RL algorithms REINFORCE and PPO. From our experiments, we find that the SPSA-based algorithms cannot train neural networks in the deep RL settings, so they are not compared with our methods.

The selection of $\{\beta_k\}$ will affect the convergence speed of the algorithm. We propose an adaptive updating approach for $\{\beta_k\}$: $\beta_k = c_0 k^{-\lambda} \text{clip}(|q_k|, c_1, c_2)$, where $\lambda \in (0.5, 1)$, and c_0, c_1 and c_2 are positive

constants. This choice of $\{\beta_k\}$ satisfies Assumption 3(b). The rationale of this learning rate is to scale up the searching step by the magnitude of the current quantile estimate clipped by some upper-and-lower bounds. In practice, to speed up searching further, $k^{-\lambda}$ can be replaced by a factor that decays at a constant rate after a certain number of iterations.

6.1 Toy Example: Zero Mean

In a simulation environment with T time steps, the reward of an agent at the t-th step is

$$R_t = X_t + h(\delta_t)(Y_t - \frac{1}{2}),\ X_t \sim \mathcal{N}(0,1),\ Y_t \sim B(1,\frac{1}{2}),$$

where δ_t is an implicit system parameter vector controlled by the agent, and $h(\delta_t)$ is an explicit state vector that can be observed. Then R_t follows a bimodal distribution with zero mean. Therefore, optimizing the expected cumulative reward is ineffective, whereas the quantile performance can be optimized by controlling δ_t. In the following results, we specify that $h : \mathbb{R} \to \mathbb{R}$.

The learning curves of REINFORCE, QPO, PPO and QPPO are presented in Figure 2. The policy is represented by a neural network consisting of two hidden layers with each containing 32 neurons. The initial learning rate is set as 5×10^{-4}, and it is multiplied by a decay factor 0.7 every 400 episodes. All algorithms update their policy parameters every 4 episodes. On the left hand side of the figure, we present the quantile estimated by total rewards in past 20 episodes. On the right hand side of the figure, we present the true quantiles estimated by 50 replications per 25 episodes. The shaded areas represent the 68% confidence intervals. We can see that QPO and QPPO significantly improve the agent's quantile performance. In contrast, the learning curves of REINFORCE and PPO keep oscillating around a low level. PPO is one of state-of-the-art policy-based RL algorithms for optimizing the expectation, but it is even harmful for training the agent in this quantile-based RL example.

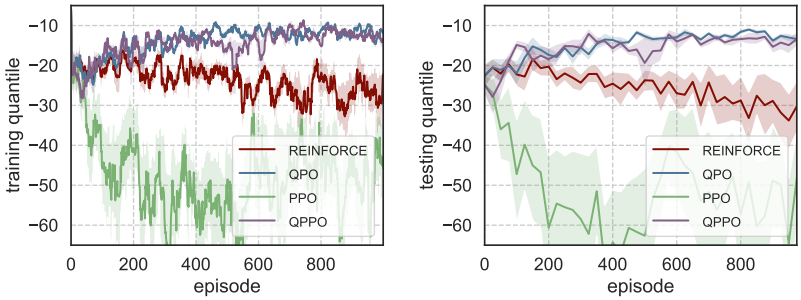

Figure 2: Learning curves for quantile ($\alpha = 0.1$) of REINFORCE, QPO, PPO and QPPO in the Zero Mean example estimated by 10 independent experiments.

6.2 Stock Trading with Simulated Prices: Sim Stock

We consider a simplified stock trading problem with N simulated stock prices and T time steps, where high returns are associated with high risks. The price of every stock is generated independently beforehand by a model in Moody and Saffell (2001). For the n-th stock, the log price x_{t+1} at time $t+1$ is

$$y^n_{t+1} = \eta_1 y^n_t + \varepsilon^n_t$$
$$x^n_{t+1} = x^n_t + y^n_t + \eta_2 v^n_t,$$

where η_1 and η_2 are constants, ε^n_t and v^n_t are noises which follow a bimodal distribution, more specifically, a summation of a zero-mean normal random variable and a zero-mean binomial random variable. A bimodal

distribution can capture distinctive statistical characteristics of the price movements under bull and bear market scenarios. Then the price sequences $\{p_t^n\}$ are simulated by

$$p_t^n = \exp\left(\frac{x_t^n}{\max_t x_t^n - \min_t x_t^n}\right).$$

The agent initially holds a random portfolio with a fixed total value 1 and can control the proportion of cash invested into each stock. At every time step, the agent first makes investment decisions, and calculates the total value of the portfolio under current stock prices and that under new stock prices. The agent is rewarded by the differences of two total values. In addition, cash held by the agent offers 0.2% risk-free return per time step. The observation state contains stock prices at the past few steps and the current portfolio.

The learning curves of REINFORCE, QPO and QPPO are presented in Figure 3. The policy and the baseline are represented by neural networks consisting of two hidden layers with each containing 64 neurons. The initial learning rate is set as 5×10^{-4}, and it is multiplied by a decay factor 0.7 every 500 episodes. The agent invests into 8 stocks in 200 time steps and can look back at stock prices within the past 5 steps. REINFORCE and QPO update their policy parameters every 2 episodes, while QPPO updates every 3 episodes. On the left hand side of the figure, we present the quantile estimated by total rewards in past 20 episodes. On the right hand side of the figure, we present the true quantiles estimated by 20 replications per 25 episodes. The shaded areas represent the 68% confidence intervals.

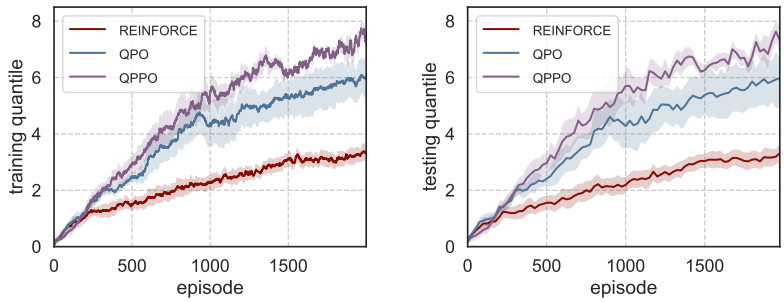

Figure 3: Learning curves for quantile ($\alpha = 0.3$) of REINFORCE, QPO and QPPO in the Sim Stock example estimated by 5 independent experiments.

Better quantile performance of the portfolio indicates that it is more robust under extreme market scenarios. The two quantile-based algorithms QPO and QPPO achieve much superior quantile performances than the mean-based algorithm REINFORCE in this example. In addition, QPPO converges faster than QPO.

6.3 Stock Trading with Real Prices: Dow Stock

We replace the simulated stock prices with the real data of Dow Jones 30 in Yang et al. (2020) and use the metric 'prccd' as a proxy of daily prices for the real stock market. We conduct experiments under the same setting as in simulated stock prices.

The learning curves for REINFORCE, QPO and QPPO are presented in Figure 4. In this example, QPO and QPPO still outperform the REINFORCE algorithm. QPPO converges faster than QPO at the beginning, but it lags slightly behind the latter at the end. This phenomenon may be attributed to the baseline network introduced in QPPO. Although baselines can speed up the convergence of policy-based algorithms, they also introduce biases. Unlike the mean-based baseline in PPO, the quantile-based baseline is not centralized so that it tends to be more difficult to learn.

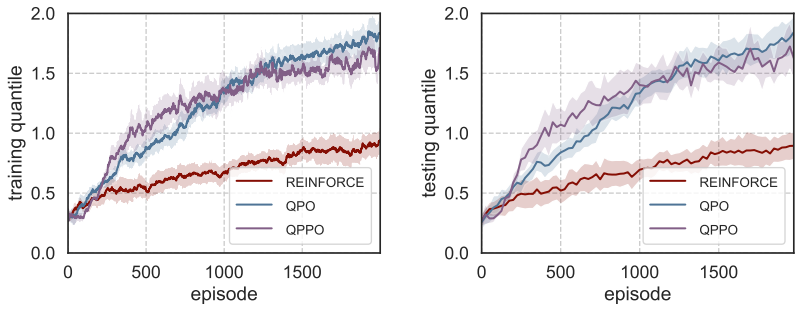

Figure 4: Learning curves for quantile ($\alpha = 0.3$) of REINFORCE, QPO and QPPO in the Dow Stock example estimated by 5 independent experiments.

7 CONCLUSION

In this paper, we propose a QPO algorithm and its variant QPPO for RL with a quantile criterion under a policy optimization framework. To the best of our knowledge, this is the first deep RL algorithm that directly optimizes quantile performance. The numerical experiments show that our proposed algorithms are effective for optimizing policy with a quantile criterion. In future work, the criterion can be extended to the distortion risk measure which is more general than quantiles. How to combine the quantile criterion with more effective policy optimization algorithms also deserves further study.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 71901003 and 72022001.

REFERENCES

Bertsekas, D. P. 1997. "Nonlinear programming". *Journal of the Operational Research Society* 48(3):334–334.

Borkar, V., and R. Jain. 2014. "Risk-constrained Markov decision processes". *IEEE Transactions on Automatic Control* 59(9):2574–2579.

Borkar, V. S. 1997. "Stochastic approximation with two time scales". *Systems & Control Letters* 29(5):291–294.

Borkar, V. S. 2001. "A sensitivity formula for risk-sensitive cost and the actor–critic algorithm". *Systems & Control Letters* 44(5):339–346.

Chow, Y., M. Ghavamzadeh, L. Janson, and M. Pavone. 2017. "Risk-constrained reinforcement learning with percentile risk criteria". *The Journal of Machine Learning Research* 18(1):6070–6120.

Chow, Y.-L., and M. Pavone. 2013. "Stochastic optimal control with dynamic, time-consistent risk constraints". In *2013 American Control Conference*, 390–395. IEEE.

Fu, M. C. 2006. "Gradient estimation". *Handbooks in operations research and management science* 13:575–616.

Glynn, P. W., Y. Peng, M. C. Fu, and J.-Q. Hu. 2021. "Computing sensitivities for distortion risk measures". *INFORMS Journal on Computing*.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine. 2018. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In *International conference on machine learning*, 1861–1870. PMLR.

Heidergott, B., and W. Volk-Makarewicz. 2016. "A measure-valued differentiation approach to sensitivities of quantiles". *Mathematics of Operations Research* 41(1):293–317.

Hessel, M., J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. 2018. "Rainbow: Combining improvements in deep reinforcement learning". In *Thirty-second AAAI conference on artificial intelligence*.

Hong, L. J. 2009. "Estimating quantile sensitivities". *Operations research* 57(1):118–130.

Hong, L. J., and G. Liu. 2009. "Simulating sensitivities of conditional value at risk". *Management Science* 55(2):281–293.

Hu, J., Y. Peng, G. Zhang, and Q. Zhang. 2021. "A stochastic approximation method for simulation-based quantile optimization". *Available at SSRN 3843441*.

Jiang, G., and M. C. Fu. 2015. "On estimating quantile sensitivities via infinitesimal perturbation analysis". *Operations Research* 63(2):435–441.

Levine, S., C. Finn, T. Darrell, and P. Abbeel. 2016. "End-to-end training of deep visuomotor policies". *The Journal of Machine Learning Research* 17(1):1334–1373.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. 2015. "Continuous control with deep reinforcement learning". *arXiv preprint arXiv:1509.02971*.

Liu, G., and L. J. Hong. 2009. "Kernel estimation of quantile sensitivities". *Naval Research Logistics (NRL)* 56(6):511–525.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. 2015. "Human-level control through deep reinforcement learning". *nature* 518(7540):529–533.

Moody, J., and M. Saffell. 2001. "Learning to trade via direct reinforcement". *IEEE transactions on neural Networks* 12(4):875–889.

Petrik, M., and D. Subramanian. 2012. "An approximate solution method for large risk-averse Markov decision processes". *arXiv preprint arXiv:1210.4901*.

Prashanth, L., and M. Ghavamzadeh. 2013. "Actor-critic algorithms for risk-sensitive MDPs".

Prashanth, L., C. Jie, M. Fu, S. Marcus, and C. Szepesvári. 2016. "Cumulative prospect theory meets reinforcement learning: Prediction and control". In *International Conference on Machine Learning*, 1406–1415. PMLR.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz. 2015. "Trust region policy optimization". In *International conference on machine learning*, 1889–1897. PMLR.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal policy optimization algorithms". *arXiv preprint arXiv:1707.06347*.

Scott, D. W. 2015. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. 2016. "Mastering the game of Go with deep neural networks and tree search". *nature* 529(7587):484–489.

Silverman, B. W. 2018. *Density estimation for statistics and data analysis*. Routledge.

Spall, J. C. 1992. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation". *IEEE transactions on automatic control* 37(3):332–341.

Tamar, A., Y. Glassner, and S. Mannor. 2014. "Policy gradients beyond expectations: Conditional value-at-risk". *arXiv preprint arXiv:1404.3862*.

Williams, R. J. 1992. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". *Machine learning* 8(3):229–256.

Yang, H., X.-Y. Liu, S. Zhong, and A. Walid. 2020. "Deep reinforcement learning for automated stock trading: An ensemble strategy". *Available at SSRN*.

Zheng, G., F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li. 2018. "DRN: A deep reinforcement learning framework for news recommendation". In *Proceedings of the 2018 World Wide Web Conference*, 167–176.

AUTHOR BIOGRAPHIES

JINYANG JIANG is a Ph.D. candidate in the Department of Management Science and Information Systems in Guanghua School of Management at Peking University, Beijing, China. He received the BS degree in information and computational sciences from School of Mathematics and Statistics, Wuhan University, and the double BS degree in computer science and technology from School of Computer Science, Wuhan University, in 2021. His research interests include machine learning and simulation optimization. His email address is jinyang.jiang@stu.pku.edu.cn.

JIAQIAO HU is an Associate Professor in the Department of Applied Mathematics and Statistics at the State University of New York, Stony Brook. He received the BS degree in automation from Shanghai Jiao Tong University, the MS degree in applied mathematics from the University of Maryland, Baltimore County, and the PhD degree in electrical engineering from the University of Maryland, College Park. His research interests include Markov decision processes, applied probability, and simulation optimization. He is currently Department Editor for IISE Transactions and Associate Editor for Operations Research. His email address is jqhu@ams.stonybrook.edu.

YIJIE PENG is an Associate Professor in Guanghua School of Management at Peking University. His research interests include stochastic modeling and analysis, simulation optimization, machine learning, data analytics, and healthcare. He is a member of INFORMS and IEEE, and serves as an Associate Editor of the Asia-Pacific Journal of Operational Research and the Conference Editorial Board of the IEEE Control Systems Society. His email address is pengyijie@pku.edu.cn.