Applying Convex Optimization in Solving Safety-Aware Deep Reinforcement Learning

Cong Zhang (G1802311E)

***Abstract*—In deep reinforcement learning (DRL), an agent's goal is to optimize a total expected long-term reward by exhaustively searching strategies in the hypothesis space, which is parameterised by θ and can be usually interpreted as neural networks, in a Markov decision process (MDP) environment while guided by a learning algorithm. However, safety requirement is often-overlooked during the exploration, even though it can play a saliency role in solving real-life problems, such as autonomous driving car. In this report, we are going to investigate the utilization of convex optimization to achieve safety-aware deep reinforcement learning. The problem to be solved is translated as a constrained Markov decision process (CMDP), where the safety requirement is encoded as a set of constrains. We will look at in detail how strong duality of Lagrangian inspires a solution and thus induces a practical algorithm. The referenced paper can be found at [1].**

***Index Terms*—Convex Optimization, Strong Primal-Duality, Deep Reinforcement Learning**

I. Introduction

This section will first introduce you some preliminary knowledge, then followed by introduction of the problem that the referenced paper tries to solve.

A. Preliminaries

A Markov decision process (MDP) is a random process where the state transition probability is only dependent on the previous state and action, i.e. $p(s_{t+1}) = p(s_{t+1}|a_t, s_t)$. After taking action a_t at state s_t, the agent will get an immediate scalar reward signal $r(s_t, a_t)$, and transit to next state s_{t+1} by $p(s_{t+1})$. The agent chooses actions by following some policy $\pi(a_t|s_t)$, which is a conditional probability given current state s_t. A *trajectory* or a *random walk* is a particular instance of the underlying Markov decision process. The goal of the agent is to optimize the total expected reward of any given initial state s_0, that is, to optimize $\mathbb{E}_{s_t\sim p(s_t|a_{t-1},s_{t-1}),a_t\sim\pi(a_{t-1}|s_{t-1})}(\sum_{t=1}\gamma r(s_t,a_t)), \forall s_0$ where $\gamma \in (0, 1)$ is a discounted factor measuring the importance of future reward. The initial state s_0 is followed some distribution $s_0 \sim p(s_0)$.

A *State Value Function* $V(s_t)$ which represents total discounted expected reward that the agent will get starting from state s_t by following policy π is,

$$V_\pi(s_t) = \mathbb{E}_{s_{t'+1}\sim p, a_{t'}\sim\pi}(\sum_{t'=t}\gamma r(s_{t'}, a_{t'})). \quad (1)$$

Cong Zhang is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. (email: cong030@e.ntu.edu.sg).

Similarly, we can define *State-Action Value Function* $Q(s_t, a_t)$ as,

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t'+1}\sim p, a_{t'+1}\sim\pi}(\sum_{t'=t}\gamma r(s_{t'}, a_{t'})). \quad (2)$$

which is the total discounted expected reward that the agent will get starting from state s_t and taking action a_t at that state, then by following policy π. The *Advantage Function* of pair (s_t, a_T) is just $A_\pi(s_t, a_T) = Q_\pi(s_t, a_T) - V_\pi(s_t)$, which reflects the difference between the average return over different actions a_t and a particular action a_T.

B. Problem Formulation

Let's consider a *constrained* Markov decision process (CMDP), where compared to original MDP, the agent will receive a penalty (cost) signal $c(s_t, a_t)$ along with reward signal. The goal of the agent is to maximize the $V_\pi(s_0)\ \forall s_0$ and keep expected long term cost below a given threshold τ, that is $G_\pi(s_0) < \tau\ \forall s_0$. There will be two formulation of the constrained Markov decision process problem, which are given by Definition.1 and Definition.2 using regularization and constrains respectively, where V_π is written as $V(\pi)$ for clarity.

Definition 1. `Regularization`: $\max_{\pi\in\Pi} V(\pi) - \lambda^\top G(\pi)$

Definition 2. `Constraint`: $\max_{\pi\in\Pi} V(\pi)\ \ s.t.\ G(\pi) < \tau$ (OPT)

It has been proved, under some conditions, these two definition are equivalent [1]. Of course there can be multiple constrains $\{G_i\}$. Now, many situations with safety requirement can be modeled as a CMDP. For example, in an water diving environment, the agent's goal is to dive as deeper as possible, whereas keeping oxygen enough for turning back to surface, and the penalty signal can be the oxygen consumed (although it will consume oxygen even doing nothing in the water, but you can imagine that the faster it dives, the quicker oxygen gone). The threshold τ can be the total oxygen capacity l in this case, i.e. we need $G_\pi < l$. To summary, the parameter τ is our tolerance of how risky we can endure. In what follows, the problem we would like to solve is `Constraint`, and we call it OPT.

II. Methodology

Now, let's start our most exciting part: how strong duality can inspire a theoretical solution and how a practical algorithm can be designed based on the theoretical result to solve the constrained MDP!

A. Theoretical Analysis

Convex Hull Hypothesis Space: Let the hypothesis space $\Pi_\theta = \{\pi_\theta\}$ parameterized by θ be a family of neural networks. Then a convex combination of π_θ given parameters $\{\alpha_i\}, \sum_i \alpha_i = 1$ is a convex hull, thus a linear space spanned by $\{\pi_\theta^i\}$ and the linear combination coefficients are restricted to $\sum_i \alpha_i = 1$. Because $\Pi_\theta \subset CH\{\Pi_\theta\} = \{\alpha_1\pi_\theta^1 + ... + \alpha_n\pi_\theta^n | \sum_i \alpha_i = 1\}$, it is easy to see the new hypothesis space $CH\{\Pi_\theta\}$ has solutions at least as good as the original hypothesis space.

Lagrangian Formulation: The Lagrangian of (OPT) is $L(\pi, \lambda) = V(\pi) + \lambda^\top(G(\pi) - \tau)$ for $\lambda \in \mathbb{R}_+^m$ where m is the number of constrains. Clearly OPT is equivalent to the max-min problem: $\max_{\pi\in\Pi} \min_{\lambda\in\mathbb{R}_+^m} L(\pi, \lambda)$, i.e., the primal problem.

Strong Duality Proof: We have established a convex hull as our new hypothesis space, so the domain is convex now. Then we assume OPT is feasible and that Slater's condition holds (otherwise, we can simply increase the constraint τ by a tiny amount). Slaters condition and policy class convexification ensure that strong duality holds [2]. Therefore, OPT is also equivalent to the min-max problem: $\min_{\lambda\in\mathbb{R}_+^m} \max_{\pi\in\Pi} L(\pi, \lambda)$, i.e., the dual problem, since the strong duality holds.

Meta-Algorithm Design: Since $L(\pi, \lambda)$ is linear in both π and λ, strong duality is also a consequence of von Neumanns celebrated convex-concave minimax theorem for zero-sum games [3]. From a game-thoeretic perspective, the problem becomes finding the equilibrium of a two-player game between the π-player and λ-player. In this repeated game, the π-player maximize the $L(\pi, \lambda)$ given the current λ, and the λ-player minimize the $L(\pi, \lambda)$ given the current π [4]. Based on this observation, we can design the meta-algorithm as presented in Algorithm.1, where ω is a convergence threshold as an hyper

Result: Optimal policy $\widehat{\pi}$
for *each round* t **do**
  $\pi_t \leftarrow$ `Best-response`(λ_t)
  $\widehat{\pi}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \pi_{t'}$, $\widehat{\lambda}_t \leftarrow \frac{1}{t}\sum_{t'=1}^t \lambda_{t'}$
  $L_{min} = \min_\lambda L(\widehat{\pi}_t, \lambda)$ //`Dual`
  $L_{max} = L($`Best-response`$(\widehat{\lambda}_t), \widehat{\lambda}_t)$ //`Primal`
  if $L_{min} - L_{max} \leq \omega$ **then**
    $\widehat{\pi} = \widehat{\pi}_t$
    `return` $\widehat{\pi}$
  end
  $\lambda_{t+1} \leftarrow$ `Online-algorithm`$(\pi_1, ..., \pi_t)$
end

Algorithm 1: The meta-level algorithm.

parameter, and `Best-response`(λ_t) is defined as:

$$\begin{aligned} \texttt{Best-response}(\lambda_t) &= \operatorname*{argmax}_{\pi\in\Pi} L(\pi, \lambda_t) \\ &= \operatorname*{argmax}_{\pi\in\Pi} V(\pi) - \lambda_t^\top G(\pi) \end{aligned} \tag{3}$$

Since the optimization problem has strong duality the difference $L_{min} - L_{max} \leq \omega$ is guaranteed to reach the given threshold, thus result an optimal policy $\widehat{\pi}$.

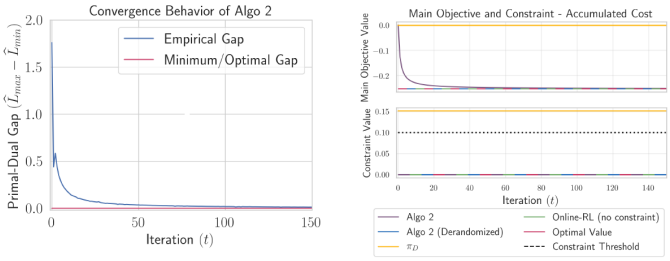

Fig. 1. Experiment results in `Constrained Car-Racing` environment.

B. Practical Algorithm

To design a practical algorithm, we need to design `Best-response` and `Online-algorithm` in Algorithm.1, where the former is known as a challenge in deep reinforcement learning which called *off-policy evaluation* meaning that we need to evaluate a policy π using data generated by another policy π', the latter is implemented as Online Gradient Descent (OGD) [5] in this paper. A Fitted Off-Policy Evaluation with Function Approximation (FQE) is proposed to solve the first challenge. I neglect details of algorithm design here as it requires further knowledge in deep reinforcement learning. Readers interested to this topic are referred to [1] for details.

III. Experiment and Evaluation

The test environment is a `Constrained Car-Racing` environment implemented in Openai Gym library [6], where a car need to follow a route as quicker as possible, and avoid obstacles on the road at the same time. The convergence result is given in Figure.1 left, and the total expected reward (where in the referenced paper it is cost, which can be set as reward by adding a negative sign, i.e., $reward = -cost$) is given in Figure.1 right.

As can be told from experiment result, the primal-dual gap shrinks very quickly at the beginning of learning and converges eventually, which validates our theoretical result.

IV. Conclusions

In this report, we review using convex optimization to solve a very challenging problem in deep reinforcement learning, i.e. safety-aware policy optimization. I need to stress that: *this work is very new (Mar.2019) and promising although it is a pre-print paper. The contribution of this paper is significant both in theoretical analysis and practical algorithm design.*

References

[1] H. M. Le, C. Voloshin, and Y. Yue, "Batch policy learning under constraints," *arXiv preprint arXiv:1903.08738*, 2019.

[2] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[3] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

[4] Y. Freund and R. E. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 79–103, 1999.

[5] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 928–936.

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.