An Alternative to Variance: Gini Deviation for Risk-averse Policy Gradient

Yudong Luo[1,4], Guiliang Liu[2], Pascal Poupart[1,4], Yangchen Pan[3]
yudong.luo@uwaterloo.ca

[1]**University of Waterloo,** [2]**CUHK,Shenzhen,** [3]**University of Oxford,** [4]**Vector Institute**

Table of Contents

Reinforcement Learning (RL)

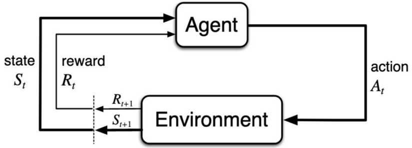

Figure 1: Markov Decision Process (MDP)

An agent interacts with environment using its policy $\pi(a|s)$.

- $\pi(a|s)$: mapping from state to action $\mathcal{S} \rightarrow \mathcal{A}$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

By interaction, a trajectory $\tau = (S_0, A_0, R_1, S_1, A_1, R_2, ...)$

- Total return random variable $G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + ...$

Reinforcement Learning (RL)

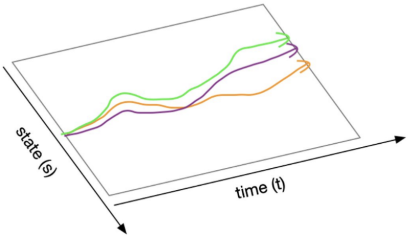

Figure 2: Random trajectories

- Traditional (Risk-neutral) RL: $\max_\pi \mathbb{E}[G_0]$
- Risk-averse RL: optimzie $\rho[G_0]$, where ρ is a risk measure
    - tail risk measure: VaR, CVaR
    - measure of variability: Variance, Standard Deviation
- For measure of variability, variance is a common choice.

Mean-Variance RL

Mean-Variance RL: maximize the expected return, minimize the return variance

$$\max_{\pi} \mathbb{E}[G_0] - \lambda \mathbb{V}[G_0] \tag{1}$$

How to maximize $\mathbb{E}[G_0] - \lambda \mathbb{V}[G_0]$ w.r.t. π ?

- $\mathbb{E}[G_0]$: time consistent, Bellman equation, dynamic programming
- $\mathbb{V}[G_0]$: time inconsistent, minimizing variance at each step is not minimizing variance of G_0

Consider Policy Gradient

- Parameterize π by θ (π_θ e.g. deep neural network)
- $\nabla_\theta \big(\mathbb{E}[G_0] - \lambda \mathbb{V}[G_0]\big)$, then gradient ascent.

Mean-Variance Policy Gradient

$$J(\theta) = \mathbb{E}[G_0] - \lambda \mathbb{V}[G_0] = \mathbb{E}[G_0] - \lambda\big(\mathbb{E}[G_0^2] - (\mathbb{E}[G_0])^2\big) \tag{2}$$

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}[G_0] - \lambda(\nabla_\theta \mathbb{E}[G_0^2] - 2\mathbb{E}[G_0]\nabla_\theta \mathbb{E}[G_0]) \tag{3}$$

Policy Gradient Theorem (Sutton and Barto (2018))

$$\nabla_\theta \mathbb{E}[G_0] = \nabla_\theta \mathbb{E}_\tau[R(\tau)] = \mathbb{E}_\tau\left[R(\tau)\nabla_\theta \sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t)\right] \tag{4}$$

where $R(\tau)$ is the return of trajectory τ

- $\nabla_\theta \mathbb{E}[G_0] : \mathbb{E}_\tau[R(\tau)\omega(\theta)] \quad \omega(\theta) = \nabla_\theta \sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t)$
- $\nabla_\theta \mathbb{E}[G_0^2] : \mathbb{E}_\tau[R^2(\tau)\omega(\theta)]$
- $\mathbb{E}[G_0]\nabla_\theta \mathbb{E}[G_0]$: requires double sampling

Mean Variance PG Issue?

Mainly due to the square term

- The variance of the gradient is very high
  - $R^2(\tau)$ in $\nabla_\theta \mathbb{E}[G_0^2] = \mathbb{E}_\tau[R^2(\tau)\omega(\theta)]$
- Sensitive to numerical scale
  - $\mathbb{E}[cG_0] = c\mathbb{E}[G_0]$, $\mathbb{V}[cG_0] = c^2\mathbb{V}[G_0]$. Change optimal solution

For double sampling $\mathbb{E}[G_0]\nabla_\theta \mathbb{E}[G_0]$

- not an issue if we can sample multiple τs.

Some works aim to do per trajectory update

- Tamar et al. (2012) used different learning rates for value and policy
- Xie et al. (2018) used Fenchel duality ($x^2 = \max_y(2xy - y^2)$) to avoid $(\mathbb{E}[G_0])^2$

Per-step Reward Variance

Consider $\mathbb{V}[R]$ as a proxy of $\mathbb{V}[G_0]$ due to the following inequality (Bisi et al. (2020))

$$\mathbb{V}[G_0] \leq \frac{\mathbb{V}[R]}{(1-\gamma)^2} \tag{5}$$

Change the objective function to

$$\max_{\pi} \mathbb{E}[R] - \lambda \mathbb{V}[R] = \max_{\pi} \mathbb{E}[R] - \lambda(\mathbb{E}[R^2] - (\mathbb{E}[R])^2) \tag{6}$$

Benefit of using $\mathbb{V}[R]$

- Eq 6 new reward $R - \lambda R^2 + \lambda(\mathbb{E}_\pi[R])^2$
- Fenchel duality (Zhang et al. (2021)): $\mathbb{E}[R] = \max_y(2\mathbb{E}[R]y - y^2)$
- New reward $R - \lambda R^2 + 2\lambda y R$ A risk neutral learning problem

Per-step Reward Variance Issue?

- $\mathbb{V}[R]$ is not an appropriate surrogate for $\mathbb{V}[G_0]$
  - In deterministic case, $\mathbb{V}[G_0] = 0$, while $\mathbb{V}[R] \neq 0$ in general
  - Shift a deterministic $r(s, a)$ may affect $\mathbb{V}[R]$ a lot
- Reward modification hinders policy learning
  - In $R - \lambda R^2 + 2\lambda y R$, $-\lambda R^2$ can make a positive reward to negative, even λ is small
  - Prevent agent from visiting the "good" state.

Gini Deviation

Random variable X, i.i.d. copies X_1, X_2. Variance is

$$\mathbb{V}[X] = \frac{1}{2}\mathbb{E}[(X_1 - X_2)^2] \tag{7}$$

Gini deviation (GD) is

$$\mathbb{D}[X] = \frac{1}{2}\mathbb{E}[|X_1 - X_2|] \tag{8}$$

Both consider the variability or dispersion of a random variable.

- Get rid of the square function
- Positive homogeneity $\mathbb{D}[cX] = c\mathbb{D}[X]$ for $c > 0$

New objective

$$\max_{\pi} \mathbb{E}[G_0] - \lambda\mathbb{D}[G_0] \tag{9}$$

Gini Deviation

Lemma 1 (Wang et al. (2020)) Gini deviation is a signed Choquet integral with a concave h given by $h(\alpha) = -\alpha^2 + \alpha, \alpha \in [0, 1]$.

$$\mathbb{D}[X] = \int_{-\infty}^{0} \Big(h\big(\Pr(X \geq x)\big) - h(1) \Big) dx + \int_{0}^{\infty} h\big(\Pr(X \geq x)\big) dx \qquad (10)$$

Lemma 2 (Wang et al. (2020), Lemma3) If F_X^{-1} is continuous, then $\mathbb{D}[X] = \int_0^1 F_X^{-1}(1-\alpha) dh(\alpha)$ (F_X^{-1} is the inverse CDF)

$$\mathbb{D}[X] = \int_{0}^{1} F_X^{-1}(\alpha)(2\alpha - 1) d\alpha \qquad (11)$$

Gini Deviation Gradient Formula

$$\mathbb{D}[X] = \int_0^1 F_X^{-1}(\alpha)(2\alpha - 1)d\alpha$$

Suppose the density function of X is $f_X(x;\theta)$ with parameter θ.

- Interested in computing $\nabla_\theta \mathbb{D}[X_\theta]$
- In RL, may think X is G_0, θ is policy.

Gini Deviation Gradient Formula

Define the α-level quantile value as $q_\alpha(X;\theta)$

Assumptions

X is a continuous random variable, and bounded in range $[-b, b]$ for all θ.
$\frac{\partial}{\partial\theta_i} q_\alpha(X;\theta)$ exists and is bounded for all θ, where θ_i is the i-th element of θ.
$\frac{\partial f_X(x;\theta)}{\partial\theta_i} / f_X(x;\theta)$ exists and is bounded for all θ, z. θ_i is the i-th element of θ.

$$\nabla_\theta \mathbb{D}[X_\theta] = -\mathbb{E}_{x\sim X_\theta}\left[\nabla_\theta \log f_X(x;\theta) \int_x^b \left(2F_{X_\theta}(t) - 1\right) dt\right] \tag{12}$$

Gini Deviation Gradient Formula

Get back to RL.

$$\nabla_\theta \mathbb{D}[G_0] = -\mathbb{E}_{R(\tau)\sim G_0}\left[\nabla_\theta \log f_{G_0}(R(\tau);\theta) \int_{R(\tau)}^{b} \left(2F_{G_0}(t) - 1\right) dt\right] \quad (13)$$

- $\nabla_\theta \log f_{G_0}(R(\tau);\theta)$: in policy gradient is $\nabla_\theta \sum_{t=0}^{T-1} \log \pi_\theta(a_t|s_t)$
- $\int_{R(\tau)}^{b} F_{G_0}(t)dt$: use ordered statistics, e.g., $R(\tau_1) \leq R(\tau_2) \leq ... \leq R(\tau_n)$.

Combine with mean, whole learning procedure

- Sample n trajectories $\{\tau_i\}_{i=1}^n$. Compute $\{R(\tau_i)\}_{i=1}^n$
- Update θ by $\nabla_\theta \mathbb{E}[G_0]$ Equation (4)
- Sort $\{R(\tau_i)\}_{i=1}^n$. Update θ by $-\lambda \nabla_\theta \mathbb{D}[G_0]$ Equation (13)

LunarLander

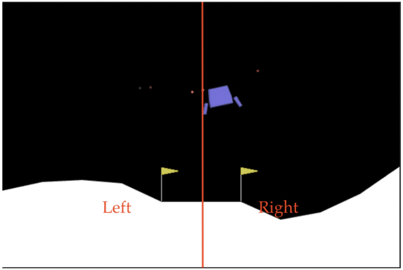

Figure 3: Modified LunarLander

- The goal is to land the agent on the ground without crashing.
- Reward is 100 if it comes to rest (unstable for total return variance and per-step variance).
- Give an additional noisy reward if agent lands in the right area.

LunarLander

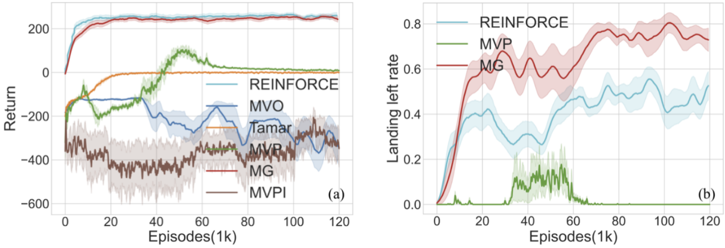

Figure 4: Return and landing at left rate

Mean-Gini Deviation (MG) compares with

- Risk-neutral (REINFORCE) Equation (4)
- Mean-Variance PG (MVO) Equation (3) ($\mathbb{V}[G_0]$, double sampling)
- Tamar et al. (2012) (Tamar) ($\mathbb{V}[G_0]$, per trajectory)
- Xie et al. (2018) (MVP) ($\mathbb{V}[G_0]$, per trajectory)
- Zhang et al. (2021) (MVPI) ($\mathbb{V}[R]$)

Thank you!

Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. *International Joint Conference on Artificial Intelligence*, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Aviv Tamar, Dotan Di Castro, and Shie Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the twenty-ninth international conference on machine learning*, pages 387–396, 2012.

Ruodu Wang, Yunran Wei, and Gordon E Willmot. Characterization, robustness, and aggregation of signed choquet integrals. *Mathematics of Operations Research*, 45(3):993–1015, 2020.

Tengyang Xie, Bo Liu, Yangyang Xu, Mohammad Ghavamzadeh, Yinlam Chow, Daoming Lyu, and Daesub Yoon. A block coordinate ascent algorithm for mean-variance optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10905–10913, 2021.