Safe Policy Learning with Constrained Return Variance

Arushi Jain[1,2]

[1] McGill University, Montreal, Canada
[2] Mila, Montreal, Canada
arushi.jain@mail.mcgill.ca

Abstract. It is desirable for a safety-critical application that the agent performs in a reliable and repeatable manner which conventional setting in reinforcement learning (RL) often fails to provide. In this work, we derive a novel algorithm to learn a safe hierarchical policy by constraining the direct estimate of the variance in the return in the Option-Critic framework [2]. We first present the novel theorem of safe control in the policy gradient methods and then extend the derivation to the Option-Critic framework.

Keywords: Safety · Policy Gradient · Option-Critic

1 Introduction

RL agents learn to solve a task by optimizing the observed return in a conventional setting. While this approach produces the highest return in expectation, it does not provide any constraints on the distribution of the return, making it a vulnerable strategy for the risk-sensitive domains. Safety in AI systems can be defined in several ways [1] - safe exploration, reward hacking, etc. Our notion of safety emphasizes on minimizing the erratic or harmful behavior of an agent - by introducing the constraints on the variance in the return. The variance in the return reflects the uncertainty in the value function which makes an agent behave inconsistently. Therefore, the unsafe states which exhibit harmful or abrupt behavior would have a higher variance in the return.

[6, 9, 11, 10, 5] used the estimate of the variance in λ-return by the indirect second-order moment method or directly estimated the cost-to-go returns with the updates provided after completing the entire trajectory. [8] came up with a direct estimation of the variance in the λ-return using a Bellman operator in the policy evaluation methods. [4] used the variance in the temporal difference (TD) error to identify the controllable states in the option-critic framework.

In our preliminary work, we first came up with a Bellman operator to directly estimate the variance in the return given a state-action pair and learn a safe policy in control setting for actor-critic methods. Taking inspiration from this work, we extended the Bellman operator of variance to option-critic setting and introduce a safe hierarchical policy learning approach.

2 Background

In a Markov Decision Process (MDP), an agent interacts with the environment in discrete time steps t, where the agent takes an action $a \in A$, transitions from state S_t to state S_{t+1}, and receives an immediate reward R_{t+1} from the environment. The expected reward is $R(S_t, A_t) = \sum_{r \in \mathbb{R}} r \sum_{s'} P(s', r | S_t, A_t)$ where $R : S \times A \to \mathbb{R}$. The environment dynamics is modeled by $P(S_{t+1}|S_t, A_t)$, where $P : S \times A \times S \to [0, 1]$. A stochastic policy $\pi(A_t|S_t)$ determines the probability of taking an action in a given state. The MDP is represented by a tuple $\langle S, A, P, R, \gamma \rangle$, where $\gamma \in [0, 1]$ is a discount factor.

3 Safe Actor-Critic (Safe AC)

Let the policy be given by $\pi_{\boldsymbol{\theta}}(a|s)$, where θ is the parameter of the policy. Extending the work by [8], the Bellman of the variance in the return given a state-action pair $\sigma_\pi(s, a)$ is similarly given by:

$$\sigma_\pi(s, a) = \mathbb{E}_\pi \big[\delta_t^2 + \bar{\gamma}\sigma_\pi(S_{t+1}, A_{t+1}) \big| S_t = s, A_t = a.\big] \tag{1}$$

where $\bar{\gamma} = \gamma^2\lambda^2$, λ is the trace-decay parameter and δ_t is the one-step TD error. The proof for the above equation is left because of the space limitation. The new safe objective function now becomes:

$$J_d(\theta) = \mathbb{E}_{d,\pi} \Big[\sum_a \pi_\theta(a|s) \big(Q_\pi(s, a) - \boldsymbol{\psi}\sigma_\pi(s, a) \big). \Big] \tag{2}$$

where $\boldsymbol{\psi}$ is the penalty coefficient. Here we aim to maximize the mean of the return along with minimizing the variance in the return in order to learn consistently behaving policy. Following the policy gradient theorem, the update for the gradient of the new safe objective function is:

$$\nabla_\theta J_{d,\pi}(\theta) = \mathbb{E}_\pi \left[\nabla_\theta \log \pi_\theta(A_t|S_t) \{ Q_\pi(S_t, A_t) \boldsymbol{-\psi\sigma_\pi(S_t, A_t)} \} \right] \tag{3}$$

4 Safe Option-Critic (Safe OC)

Keeping similar notions to Option-Critic Architecture [2], an option $w \in W$ is defined as a tuple (I_w, π_w, β_w); where I_w contains the initial states set where an option can start, π_w is the option policy defining a distribution over action space and β_w determines the termination probability of an option in a state. The policy over the options is denoted by $\mu(w|s)$. Let $\Theta = [\theta, \nu, \kappa]$ be the parameters of intra-option policy, termination condition and policy over options respectively.

Let us consider $Z_t = (S_t, W_t)$ as an augmented state space, a space of state-option pair. Similar to the (1), the variance given a state-option-action is denoted by:

$$\sigma_{\pi,\mu}(z, a) = \mathbb{E}_{\pi,\mu}[\delta_t^2 + \bar{\gamma}\sigma_{\pi,\mu}(Z_{t+1}, A_{t+1}) | Z_t = z, A_t = a] \tag{4}$$

The safe objective for the option-critic is similar to (2) where state is replaced with augmented states space and $\boldsymbol{\psi_z}$ represents the regularizer for the variance. The updates for the parameters are shown below where blue color highlights the change from [2] due to the safe objective function. The intra-option gradient is:

$$\nabla_\theta J(\Theta) = \mathbb{E}_{d,\Theta}\Big[\sum_a \nabla_\theta \pi_\theta(a|z)\big(Q_{\pi,\mu}(z,a) \boldsymbol{-\psi_z\, \sigma_{\pi,\mu}(z,a)}\big)\Big] \tag{5}$$

The termination function gradient update is given by:

$$\nabla_\nu J(\Theta) = \mathbb{E}_{d,\Theta}[-\nabla_\nu \beta_\nu(s',w)\big(A_{\pi,\mu,Q}(s',w) \boldsymbol{-\psi_z\, A_{\pi,\mu,\sigma}(s',w)}\big)] \tag{6}$$

where $\boldsymbol{A_{\pi,\mu,\sigma}(s',w)} = \sigma_\Theta(s',w) - \sigma_\Theta(s')$ is the *advantage function* for the variance similar to the value function. The update for the policy over options is provided by:

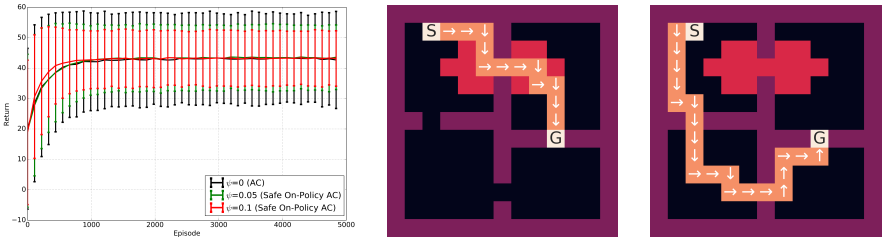
$$\nabla_\kappa J(\Theta) = \mathbb{E}_{d,\Theta}[\beta_\nu(s',w)\sum_{w'} \nabla_\kappa \mu_\kappa(w'|s')\big(Q_\Theta(s',w') \boldsymbol{-\psi_{s',w'}\, \sigma_\Theta(s',w')}\big)] \tag{7}$$

5 Experiments

We first performed the experiments in Safe Actor-Critic to see the effect of the safety and later show a tabular experiment in Safe Option-Critic to show the concreteness of the proposed algorithm. We added an unsafe frozen region (F) in one of the hallway in the discrete tabular four rooms (FR) environment [2] which has a normal reward distribution from $\mathcal{N}(\mu = 0, \sigma = 8)$. A different action from the one intended by the policy is taken with 0.2 probability. The agent can be initialized from anywhere in the state space and the reward of 50 is received when the agent reaches the goal state denoted by G in the Fig. 1 The reward for all the other states is kept 0. In expectation, the reward for both the hallways is 0.

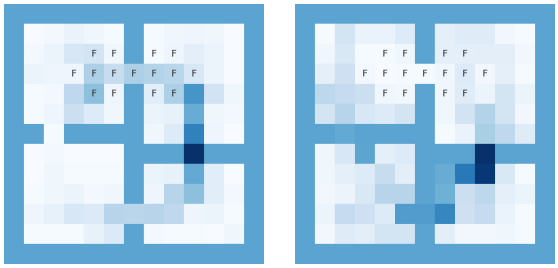
Safe AC: Fig. 1 depicts that using the safe framework (red plot), the variation in the return decreases highlighting that the agent reduces the visits to the variable reward region. The sampled policy from both the baseline and the safe method shows that agent takes a round about path to reach the goal state to avoid the frozen region. The risk-averse policy would generally exhibit faster convergence speed due to decrease in visits to inconsistent regions. If the learning curve is extended over a period of time, the risk-neutral would achieve higher or at par mean performance compared to the risk-averse as the penalty term is not part of the baseline. The learning curve for the experiments using non-linear approximation in Mujoco OpenAI Gym [3] environments are shown in the Appendix to show the scalability of the algorithm.

Safe OC: Using the same four rooms environment, we compare the performance of the Safe OC with the baseline OC using 4 options. The code was built on top of Option-Critic baseline. Fig. 2 represents the state visitation frequency of the agent, where the Safe OC shows a lower frequency in the frozen hallway. This depicts that safe agent learns to avoid erratically behaving region. The performance using the learning curve and the absolute TD error are shown in the appendix.



(a) Learning Curve (b) AC Policy (c) Safe-AC Policy

Fig. 1: **Safe AC in FR domain**: a) Averaged performance over 50 trials where the vertical bands depict the standard deviation demonstrating that safe methods ($\psi > 0$) have lower variation in the return. Sampled policy using b) baseline actor-critic, c) safe actor-critic method. The unsafe region is depicted by the red color.



(a) OC State Frequency (b) Safe-OC State Frequency

Fig. 2: **State Frequency in FR domain**: Shows the state visitation count in 100 sampled trajectories in testing phase. The darker shade represents the higher frequency. Safe OC learns to avoid the frozen hallway showcasing lighter shade in F region.

6 Conclusion & Future Work

In this work, we presented a generic safe policy learning framework which learns a consistently behaving policy by constraining the direct estimate of the variance in the return. We first presented the safe policy gradient style update in the primitive action space and then extended this framework to the hierarchical option-critic format. Our approach provides an incentive to the agent which minimize the visits to the inconsistently behaving regions. This framework provides the capability to overcome the variability introduced by the environment dynamics.

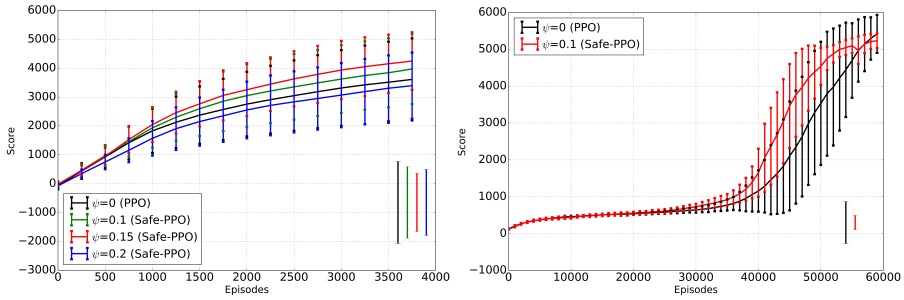
The future step is to experiment with the safe option-critic framework using the non-linear function approximation in problems like Mujoco and ALE domains. The other step is to explore in the direction of variable value of ψ_z, $\forall z \in Z$ such that each options can have a different value of ψ, the importance factor for learning a safe policy.

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P.F., Schulman, J., Mané, D.: Concrete problems in AI safety. CoRR (2016)
2. Bacon, P.L., Harb, J., Precup, D.: The option-critic architecture. In: AAAI. pp. 1726–1734 (2017)
3. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI gym. arXiv preprint arXiv:1606.01540 (2016)
4. Jain, A., Khetarpal, K., Precup, D.: Safe option-critic: Learning safety in the option-critic architecture. arXiv preprint arXiv:1807.08060 (2018)
5. Prashanth, L., Ghavamzadeh, M.: Actor-critic algorithms for risk-sensitive MDPs. In: Advances in neural information processing systems. pp. 252–260 (2013)
6. Sato, M., Kimura, H., Kobayashi, S.: TD algorithm for the variance of return and mean-variance reinforcement learning. Transactions of the Japanese Society for Artificial Intelligence **16**(3), 353–362 (2001)
7. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
8. Sherstan, C., Bennett, B., Young, K., Ashley, D.R., White, A., White, M., Sutton, R.S.: Directly estimating the variance of the λ-return using temporal-difference methods. arXiv preprint arXiv:1801.08287 (2018)
9. Tamar, A., Di Castro, D., Mannor, S.: Policy gradients with variance related risk criteria. In: Proceedings of the twenty-ninth international conference on machine learning. pp. 387–396 (2012)
10. Tamar, A., Di Castro, D., Mannor, S.: Learning the variance of the reward-to-go. Journal of Machine Learning Research **17**(13), 1–36 (2016)
11. Tamar, A., Xu, H., Mannor, S.: Scaling up robust MDPs by reinforcement learning. arXiv preprint arXiv:1306.6189 (2013)
12. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5026–5033. IEEE (2012)

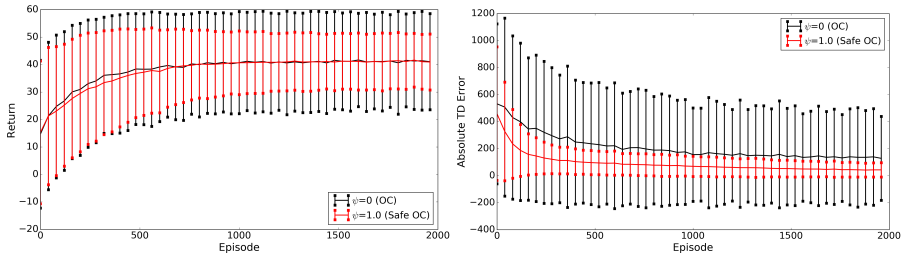
Appendix

We experimented with the safe objective with the proximal policy optimization (PPO) [7] algorithm using safe actor-critic framework in Mujoco environments [12]. In all the experiments for Safe-OC, ψ_z is kept to a constant value for all $z \in Z$.



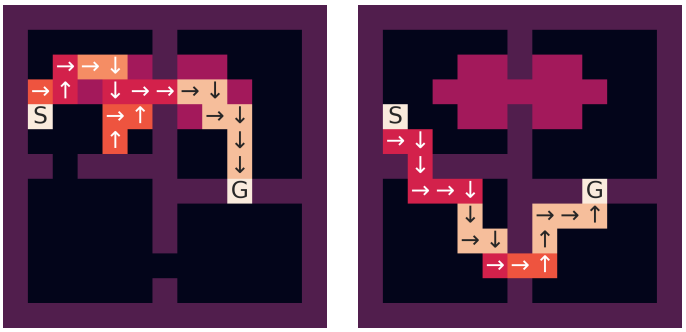
(a) Half Cheetah (b) Humanoid

Fig. 3: **Learning curve in Mujoco environments for Safe AC**: The graphs depict the scores over multiple trials: a) 20 trials for Half Cheetah; b) 5 trials for Humanoid environment. The bands represent the standard deviation of score across different trials. $\psi = 0$ (black) represents the baseline PPO without safety. For the best performing safe architectures (red) a) $\psi = 0.15$, b)$\psi = 0.1$, we observe a reduction in the standard deviation of score compared with baseline. The comparison across the averaged standard deviation of score for different ψ are shown in bottom right of the graph where reduced variance indicates the consistency in the performance.



(a) Learning Curve (b) Absolute TD Error

Fig. 4: **Safe OC in FR domain**: The graphs shows the averaged performance over 100 runs. Shows a) return, b) sum of absolute TD error. The safe policy (red) has smaller standard deviation (vertical bands) compared to the baseline.



(a) OC (b) Safe OC

Fig. 5: **Safe OC in FR domain**: The change in color represents the switch among 4 options. The purple region in upper hallway represents the frozen hallway. The graphs shows the sampled trajectory of a) baseline OC, b) Safe OC. The model without safety takes the shortest path to reach the goal state depicted by G, whereas the safe model takes a round about path to reach the goal.