Technische Universität Berlin | Fakultät IV | Neuronale Informationsverarbeitung

Learning Notes on Safe Reinforcement Learning

Dr. Rong Guo

Berlin, 2022

Abstract

Learning notes on Safe Reinforcement Learning

Contents

1 Literature Review

1.1 A Comprehensive Survey on Safe Reinforcement Learning (?)

1.1.1 Catigorization of Safe RL Algorithms

Safe RL can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes.

The opposite of the concept *safety* is *risk*, which is related to the stochasticiy of the environment. Under the inherent uncertainty, an optimal policy w.r.t. the long-term reward maiximization may still perform poorly in some catastrophic situations.

Two fundamental categories of Safe RL algorithms:

1. Transforming the optimization criterion to include some risk measures.
   Risk-sensitive MDP and *Constrained MDP*

2. Modifying the exploration process through the incorporation of prior/external knowledge and/or the use of a risk metric, while the optimization criterion remains.

1.1.2 Optimization Criterion Considering Risk

Risk-neutral criterion

$$\max_{\pi\in\Pi} \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t r_t\Big]. \tag{1}$$

Risk-aware criterion

1. Worst-case criterion
   Minimax criterion: maximize the expectation of the return w.r.t the worst outcome under a policy or w.r.t the worst case policy over all possible models.
   - Minimax criterion under inherent uncertainty of the system:

$$\max_{\pi\in\Pi}\min_{w\in\Omega^\pi} \mathbb{E}_{\pi,w}\Big[\sum_{t=0}^{\infty}\gamma^t r_t\Big], \tag{2}$$

     where Ω^π is a set of trajectories that occurs under policy π.

     In general, the minimax criterion is too restrictive as it takes into account severe but extremely rare events which may never occur. The optimality criterion is exclusively focused on risk-avoidance or risk-averse policies.

     Example: β-pessimistic Q – learning, (**?**):

$$Q_\beta(s_t,a_t) = Q_\beta(s_t,a_t) + \alpha\Big(r_{t+1} + \gamma\Big((1-\beta)\max_{a_{t+1}\in\mathcal{A}} Q_\beta(s_{t+1},a_{t+1}) + \beta\min_{a_{t+1}\in\mathcal{A}} Q_\beta(s_{t+1},a_{t+1})\Big)\Big) \tag{3}$$

- Minimax criterion under parameter uncertainty (transition probabilities of the MDP is not know exactly):

$$\max_{\pi\in\Pi}\min_{p\in P}\mathbb{E}_{\pi,p}\Big[\sum_{t=0}^{\infty}\gamma^t r_t\Big], \tag{4}$$

where P is a set of possible transition matrices.

2. Risk-sensitive criterion
   - Exponential utility functions:

$$\max_{\pi\in\Pi}\tfrac{1}{\beta}\log\mathbb{E}_{\pi}(e^{\beta\sum_{t=0}^{\infty}\gamma^t r_t}), \tag{5}$$

where β is the risk sensitivity parameter. Taylor expansion of Formula 5 is:

$$\max_{\pi\in\Pi}\tfrac{1}{\beta}\log\mathbb{E}_{\pi}(e^{\beta R}) = \max_{\pi\in\Pi}\mathbb{E}_{\pi}(R) + \tfrac{\beta}{2}Var(R) + \mathcal{O}(\beta^2). \tag{6}$$

Risk-averse for $\beta < 0$, risk-seeking for $\beta > 0$ and risk-neutral for $\beta \to 0$

   - Weighted sum of return and risk:

$$\max_{\pi\in\Pi}(\mathbb{E}_{\pi}(R) - \beta\omega), \tag{7}$$

where ω refers to the consideration of the risk concept which can take various forms, e.g., $Var(R), TD, \rho^{\pi}(s) = \mathbb{E}(\sum_{i=1}^{\infty}\gamma^i\bar{r})$ in which $\bar{r} = 1$ if an error state occurs and $\bar{r} = 0$ if not.

3. Constrained criterion: constrained MDP (**?**)

$$\max_{\pi\in\Pi}\mathbb{E}_{\pi}(R), \text{ subject to } c_i \in C, c_i = \{h_i \leq \alpha_i\}, \tag{8}$$

where c_i represents the ith constraint in C that the policy π must fulfill, with $c_i = \{h_i \leq \alpha_i\}$ where h_i is a function related with the return and α_i is the threshold restricting the values of this function. Depending on the problem, the $\leq$ in the constraints may be replaced by $\geq$. The constraints can be seen as restrictions on the space of allowable policies. Formula 8 can be transformed into:

$$\max_{\pi\in\Gamma}\mathbb{E}_{\pi}(R), \tag{9}$$

where $\Gamma \subset \Pi$, each policy $\pi \in \Gamma$ satisfies the constraints $c_i \in C$. Γ is the space of safe policies, e.g., constraints to ensure that the expectation of the return exceeds some specific minimum threshold $P(\mathbb{E}(R) \leq \alpha) \leq (1-\epsilon)$, to ensure that the variance of the return does not exceed specific maximum threshold $Var(R) \leq \alpha$, to enforce ergodicity(that is, if it can reach any state it visits from any other state it visits, so that errors are reversible, which requires the model of the MDP to be known or learned), to ensure specific restrictions of the problem.

4. Other optimization criterion: risk measures in financial engineering, e.g., value-at-risk (VaR).

Note that a policy with a small variance can still have a large risk, because this policy can lead the agent to error states. Therefore, the variance or the worst-outcome criterion may not be generalizable to any risky domain.

1.1.3 Exploration Process

The use of random exploration would require an undesirable state to be visited before it can be labeled as undesirable. However, such visits to undesirable states may result in damage or injury to the agent, the learning system or external entities. Thus, it would be desirable to prevent the risk situations from the early steps in the learning process. The exploratory process is responsible for visits to undesirable states or risky situations but also for progressively improve the policies learned.

Examples of other risk metric based on the level of knowledge about a state: the distance between the known and the unknown space, the difference between the highest and lowest Q-values, the number of times an agent has made a non-trivial Q-value update in a state.

1. External knowledge.
   a) Providing initial knowledge, gathered from a teacher or previous information on the task, to bootstrap the learning algorithm. The learning algorithm is exposed to the most relevant regions of the state and action spaces from the earliest steps of the learning process, thereby eliminating the time need in random exploration for the discovery of these regions, e.g., evolutionary methods, transfer learning, initialization in RLM algorithms.
   b) Deriving a policy from a finite set of demonstrations, examples provided by the teacher, to learn a model from which to derive a policy in an off-line and, hence, safe manner, e.g., learning from demonstration (**?**), apprenticeship learning.
   c) Providing teach advice. The teacher may be a human or a simple controller, but in both cases it does not need to be an expert in the task. The teacher shares the goal of the agent and provides advice when it is necessary to avoid fatal states.
      - The learner agent asks for advice when a *confidence parameter* in a state is low. e.g., the confidence parameter can also be used to detect risky situations based on the definition of fatal transitions or unknown states, safe function.
      - The teacher provides action or information whenever the teacher feels its help is necessary, either using an interactive reward interface or sending human advice message (e.g., inductive logic programming).
      - Interactive learning process for both the teacher and the agent.
2. Risk-directed Exploration, while the classic optimization criterion remains. The exploration process is carried out by taking into account a defined risk metric, e.g.,*controllability*:

$$C(s_t, a_t) \leftarrow C(s_t, a_t) - \alpha'(|\sigma_t| + C(s_t, a_t)). \tag{10}$$

   The controllability is used as an exploration bonus.
   The *risk-adjusted utility*: $p(1 - U(s_t, a_t)) + (1 - p)Q(s,a_t)$, where $\in [0, 1]$.

1.2 Benchmarking Safe Exploration in Deep Reinforcement Learning (?)

Safety Gym Safety starter agent

The safety concept: avoiding hazards, that is, constraining the state and behavior of the system to stay away from the circumstances that lead to harm.

The safe exploration problem: Most of the RL agents so far are typically trained in simulations, where safety concerns are minimal. However, for many problems simulators might not be available or high-enough fidelity for RL to learn behaviours that succeed in the real word. The trial-and-error nature of RL in the real world might lead to unacceptable catastrophes.

How do we formulate safety specifications to incorporate them into RL, and how do we ensure that these specifications are robustly satisfied throughout exploration?

- Towards standardizing safety specifications: 1) safety specifications should be separate from task performance specifications, and 2) constraints are a natural way to encode safety specifications.
- Towards measuring progress: 1) task performance of the final policy, 2) constraint satisfaction of the final policy, and 3) average regret with respect to safety costs throughout training.
- Towards providing useful baselines: 1) task objectives and safety objectives have meaningful trade-offs, and 2) performing well at the task does not automatically result in safe behavior (**?**).

1.2.1 Constrained RL

The general problem of training an RL agent with the intention of satisfying constraints throughout exploration in training and at test time: to formulate safety requirements as constraints, and to attain constraint-satisfying exploration.

An optimal policy in constrained RL is given by:

$$\pi^* = \arg\max_{\pi \in \Pi_C} J_r(\pi), \tag{11}$$

where Π_C denote a feasible set of constraint-satisfying policies and $J_r(\pi)$ is a reward-based objective function, e.g., the infinite-horizon discounted return, the finite-horizon undiscounted return, or the infinite-horizon average reward.

The constrained Markov Decision Processes (CMDP) are equipped with a set of cost functions, $c_1, \cdots, c_k$, separate from the reward function. The feasible set in a CMDP is given by:

$$\Pi_C = \pi : J_{c_i}(\pi) \leq d_i, i = 1, \cdots, k, \tag{12}$$

where J_{c_i} is a cost-based constraint function defined the same way as an expected return or average return metric (using c_i instead of the reward r), and each d_i is a threshold (a human-selected hyperparameter). The CMDP framework can be extended to use different kinds of cost-based constraints. The use of constraints may improve 1) the ease with which safety specifications are learned and transferred between tasks, and 2) the robustness with which agents attain those safety requirements.

1.2.2 Evaluation protocol

- Optimization problem:

$$\begin{aligned} &\max_{\pi_\theta} \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} r_t\right] \\ \text{s.t. } &\max_{\pi_\theta} \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} c_t\right] \leq d, \end{aligned} \tag{13}$$

where c_t is the aggregate indicator cost function for the environment ($c_t = 1$ for an unsafe interaction, regardless of source) and d is a hyperparameter.

- Metrics:
  - The average episodic return, $J_r(\theta)$.
  - The average episodic sum of costs, $J_c(\theta)$.
  - The average cost over the entirety of training, ρ_c. If T_{ep} is the average episode length, the condition "the average episode during training satisfied constraints" can be written as $\rho_c T_{ep} \leq d$.
- Comparing training runs:
  - All agents that fail to satisfy constraints are strictly worse than all agents that satisfy constraints.
  - For two constraint-satisfying agents A_1 and A_2 that have been trained for an equal number of environment interactions, A_1 dominates A_2 ($A_1 \succ A_2$) if it strictly improves on either return or cost rate and does at least as well on the other. That is,

$$A_1 \succ A_2 \text{ if } \begin{array}{ll} J_r(A_1) \geq J_r(A_2) & \\ \rho_c(A_1) < \rho_c(A_2) & \end{array} \text{ or } \begin{array}{ll} J_r(A_1) > J_r(A_2) & \\ \rho_c(A_1) \leq \rho_c(A_2) & \end{array} \tag{14}$$

- Comparing Algorithms:
  - Normalized return

$$\bar{J}_r(\theta) = \frac{J_r(\theta)}{J_r^{\varepsilon}}, \tag{15}$$

  - Normalized constraint violation

$$\bar{M}_c(\theta) = \frac{\max(0, J_c(\theta) - d)}{\max(\epsilon, J_c^{\varepsilon} - d)}, \epsilon = 10^{-6}, \tag{16}$$

  - Normalized cost rate

$$\bar{\rho}_c(\theta) = \frac{\rho_c(\theta)}{\rho_c^{\varepsilon}}, \tag{17}$$

    where $J_r^{\varepsilon}, J_c^{\varepsilon}, \rho_c^{\varepsilon}$ are a set of characteristic metrics for each environment ε.

1.2.3 Algorithms

- **Unconstrained algorithms**: TRPO(**?**) and PPO(**?**).
- **Constrained algorithms**: TRPO-Lagrangian and PPO-Lagrangian

$$\max_{\theta} \min_{\lambda \geq 0} \mathcal{L}(\theta, \lambda) \doteq f(\theta) - \lambda g(\theta) \tag{18}$$

- **Constrained policy optimization**: CPO (**??**)

Note that standard model-free RL approaches without replay buffers are fundamentally limited in their ability to minimize constraint regret: they must continually experience unsafe events in order to learn about them. As a result, memory-based and model-based RL approaches might be particularly interesting here.

1.2.4 Research direction

- Safe transfer learning
  - Whether the agent can quickly adapt to new safety requirements: An agent is initially trained in one constrained RL environment, and then transferred to another environment where the task is the same but the safety requirements are different.
  - Whether the agent can remain constraint-satisfying despite the potential for catastrophic forgetting induced by the change in objective function: An agent is initially trained in one constrained RL environment, and then transferred to another environment where the safety requirements are the same but the task is different.
- Combining implicitly-specified objects with constrained RL, e.g., inverse reinforcement learning, learning from human preferences, and other heuristics for extracting value-aligned objectives from human data.

1.3 AI Safety Gridworlds (?)

gridworlds as minimal safety checks.

Gridworld setups:

- States and actions, an episodic $MDP < \mathcal{S}, \mathcal{A}, T, R, s_0 >$: A two-dimensional grid of cells (empty, or contain a wall or other objects), similar to a chess board. The agent always occupies one cell of the grid and can only interact with objects in its cell or move to the four adjacent cells.
- Reward function R: The nominal reinforcement signal observed by the agent.
- (Safety) performance function R^*: A second reward function hidden from the agent. The agent is evaluated on R^* instead of R.
- Objectives:
  - *Specification problem*: R and R^* differ. In a specification problem the challenge is to find an (a priori) algorithmic solution for each of these additional objectives that generalizes well across many environments.
    - Safe interruptibility: The off-switch environment.
    - Avoiding side effects: The irreversible side effects environment.
    - Absent Supervisor: The absent supervisor environment.
    - Reward Gaming: The boat race environment and the tomato watering environment.
  - *Robustness problem*: R and R^* are identical.

- Self-modification: The whisky and gold environment.
- Distributional Shift: The lava world environment
- Robustness to Adversaries: The friend or foe environment
- Safe Exploration: The island navigation environment.

2 Safe RL algorithms

3 Domain Background

1. Open AI Safety Gym
2. CoinRun
3. Safe Reinforcement Learning
4. Safe Exploration

References

Abbeel, P. and Ng., A. Y. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2005.

Achiam, J.; Held, D.; Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2017.

Altman, E. *Constrained Markov Decision Processes.* Bantam, London, 1988.

Chow, Y.; Nachum, O.; Faust, A.; Duenez-Guzman, E., and Ghavamzadeh, M. Lyapunov-based safe policy optimization for continuous control. In *ICML 2019 Workshop RL4RealLife*, 2019.

García, J. and Fernandez, F. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

Gaskett, C. Reinforcement learning under circumstances beyond its control. In *International Conference on Computational Intelligence for Modelling Control and Automation*, 2003.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P. A.; Everitt, T.; Lefrancq, A.; Orseau, L., and Legg, S. AI Safety Gridworlds. *arXiv:1711.09883 [cs]*, November 2017.

Ray, A.; Achiam, J., and Amodei, D. Benchmarking Safe Exploration in Deep Reinforcement Learning. 2019.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M., and Moritz, P. Trust region policy optimization. In Bach, F. and Blei, D., editors, *Proceedings of the International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, 2017.