Learning safe policies with expert guidance

Jessie Huang, Fa Wu, Doina Precup, Yang Cai

McGill University

NIPS 2018

Problem Definition

- Given:
  - demonstrations from expert
- Asked:
  - safe policy
- Method:
  - ellipsoid based optimization
- Assumption:
  - $R(s) = w.\phi(s)$, where $\phi(s)$ is a vector of features
- Value Funciton:

$$
\begin{aligned}
\mathbb{E}_{s_0 \sim D}[V^{\pi}(s_0)|M] \quad &= \mathbb{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t w.\phi(s_t)|\pi] \\
&= w.\mathbb{E}[\textstyle\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi] \\
&= w.\Psi(\pi)
\end{aligned}
$$

Background

- Robust MDP:
  - Which policy gives us the most in the worst condition?
  - Maxmin learning

$$\max_{\mu \in P_F} \min_{w \in P_R} \mu^\top w$$

  - By strong duality

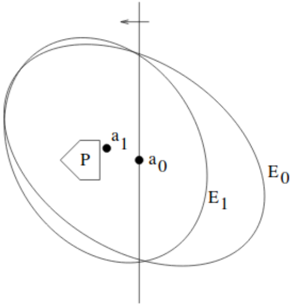
$$\begin{aligned} \max \quad & z \\ s.t. \quad & z \leq \mu^\top w, \quad \forall w \in P_R \\ & \mu \in P_F \end{aligned}$$

Background

- Problem?
  - Coming up with a reasonable P_F
  - Too many possible rewards $w \in P_R$
- Idea?
  - Every Linear Program can be turned into a series of feasibility problem
- How?
  - $$\begin{aligned} \max \quad c^\top x \\ Ax \le b \\ x \ge 0 \end{aligned} \quad \equiv \quad \begin{aligned} c^\top x \ge c_0 \\ Ax \le b \\ x \ge 0 \end{aligned}$$
  - if feasible, the optimum is smaller than c_0. So, we decrease c_0 by a factor of 2
  - if not feasible, the optimum is in the interval of $[c_0/2, c_0]$
  - initial c_0 should be sufficiently large

Optimization vs. Feasiblity

- It turns into a binary search
- Solves in polynomial time in the input size
- An intuitive way of addressing feasibility problem is called Ellipsoid Algorithm



Ellipsoid Algorithm

Input: Bounding ellipsoid E_0 for S, Lower bound V_l on $Vol(S)$.
Output: "yes" if the linear program is feasible, "no" otherwise.
Algorithm:

```
i=0;
while(Vol(E_i) ≥ V_l){
  p = Center of E_i;
  (ans,H) = SepOracle(p);
  if(ans==yes)
    return "yes";
  else{
    Take the separating hyperplane H and let
    E_{i+1} = minimum volume ellipsoid containing E_i ⋂ H^+;
    i = i + 1;
  }
}
return "no";
```

Separation Oracle

Algorithm 1 Separation Oracle SO_R for the reward polytope P_R

---

input $w' \in \mathbb{R}^k$

1: Let $\mu_{w'} := \text{argmax}_{\mu \in P_F} \mu \cdot w'$. Notice that $\mu_{w'}$ is the feature vector of the optimal policy under reward weights w'. Hence, it can be computed by our MDP solver ALG.

2: **if** $\mu_{w'} \cdot w' > \mu_E \cdot w' + \epsilon$ **then**

3: output "NO" , and $(\mu_E - \mu_{w'}) \cdot w + \epsilon \geq 0$ as the separating hyperplane, since for all $w \in P_R, \mu_E \cdot w \geq \mu_{w'} \cdot w - \epsilon$.

4: **else**

5: output "YES".

6: **end if**

Separation Oracle

Algorithm 2 Separation Oracle for the feasible (μ, z) in LP 1

input $(\mu', z') \in \mathbb{R}^{k+1}$

Query $SO_F(\mu')$.
if $\mu' \notin P_F$ **then**
  output "NO" and output the same separating hyperplane as outputted by $SO_F(\mu')$.
else
  Let $w^* \in \text{argmin}_{w \in P_R} \mu' \cdot w$ and $V = \mu' \cdot w^*$. This requires solving a linear optimization problem over P_R using the ellipsoid method with the separation oracle SO_R.
  if $z' \leq V$ **then**
    output "YES"
  else
    output "NO", and a separating hyperplane $z \leq \mu \cdot w^*$, as $z' > \mu' \cdot w^*$ and all feasible solutions of LP 1 respect this constraint.
  end if
end if

A problem, and a solution!

- Problem:
  - Despite the mathematical proof of polynomial time complexity, it does worse than simplex
- Solution:
  - Using follow-the-pertubed-leader

Thank You!