ALGORITHMS FOR INVERSE REINFORCEMENT LEARNING
A. NG, S. RUSSEL

GABRIELE MARINO

INTRODUCTION TO IRL

- IRL Problem:
  - Given the policy of an agent determine the reward function being optimezed by the agent behavior.
  - Formally, given a set of $\boldsymbol{N}$ states $\boldsymbol{S}$, a set of $\mathbf{k}$ actions $\boldsymbol{A}$, the set of transition probabilities $\boldsymbol{P_{sa}}$, a discount facor $\boldsymbol{\gamma}$ and a policy $\boldsymbol{\pi}$, we want to find the reward function(s) $\boldsymbol{R}$ such that $\boldsymbol{\pi}$ is optimal for the MDP defined by $(\boldsymbol{S}, \boldsymbol{A}, \{\boldsymbol{P_{sa}}\}, \boldsymbol{\gamma}, \boldsymbol{R})$.

- Applications:
  - Imitation and apprenticeship learning: learning by looking at the behavior of expert agents.
  - Explain human or animal behavior by ascertaining their reward function (in particular multivariate ones).

IRL IN FINITE STATE SPACES

- Theorem (Characterization of the Solution Set):
  - Let a finite state space $\boldsymbol{S}$, a set of actions $\boldsymbol{A} = \{\boldsymbol{a_1}, \dots, \boldsymbol{a_k}\}$, transition probability matrices $\{\boldsymbol{P_a}\}$, and a discount factor $\boldsymbol{\gamma} \in (\mathbf{0}, \mathbf{1})$ be given. Then the policy $\boldsymbol{\pi}$ given by $\boldsymbol{\pi}(\boldsymbol{s}) \equiv \boldsymbol{a_1}$ is optimal if and only if, far all $\boldsymbol{a} = \boldsymbol{a_2}, \dots, \boldsymbol{a_k}$, the reward $\boldsymbol{R}$ satisfies:

$$\left(\boldsymbol{P_{a_1}} - \boldsymbol{P_a}\right)\left(\boldsymbol{I} - \boldsymbol{\gamma P_{a_1}}\right)^{-1}\boldsymbol{R} \succcurlyeq \mathbf{0}.$$

  - Proof: It follows directly from Bellman equations and Bellman optimality with some simple algebra.
  - Notes:
    - $\boldsymbol{\pi}(\boldsymbol{s}) \equiv \boldsymbol{a_1}$ is not restrictive, as it is possible to rename actions w.l.o.g.
    - $\left(\boldsymbol{I} - \boldsymbol{\gamma P_{a_1}}\right)$ is always invertible: $\boldsymbol{P_{a_1}}$ eigenvalues have modulus $\leq \mathbf{1}$ (it is a transition matrix), so $\boldsymbol{\gamma P_{a_1}}$ eigenvalues have modulus $< \mathbf{1}$, from which it follows that $\left(\boldsymbol{I} - \boldsymbol{\gamma P_{a_1}}\right)$ has no zero eigenvalues, and thus it is not singular.
    - From an existence theorem about the reward function of finite state spaces MDP, the solution set is never empty.
  - Many different $\boldsymbol{R}$ (also not significant ones: every constant $\boldsymbol{R}$, included $\boldsymbol{R} = \mathbf{0}$). How to choose among them?

LINEAR PROGRAMMING FORMULATION OF IRL IN FINITE STATE SPACES

$$\text{maximize} \quad \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \min_{\boldsymbol{a\in\{a_2,\ldots,a_k\}}} \{\left(P_{a_1}(i) - P_a(i)\right)\left(I - \gamma P_{a_1}\right)^{-1} R)\} - \lambda \|R\|_1$$

$$\text{subject to} \quad \begin{cases} \boldsymbol{\left(P_{a_1} - P_a\right)\left(I - \gamma P_{a_1}\right)^{-1} R \succcurlyeq 0,} & \boldsymbol{\forall a \in A \setminus \mathrm{a}_1} \\ \boldsymbol{|R_i| < R_{max},} & \boldsymbol{i = 1, 2, \ldots, N} \end{cases}$$

- The objective function, apart from the penalization term, is equivalent to $\sum_{\boldsymbol{s\in S}} \left(\boldsymbol{Q^{\pi}(s, a_1)} - \max_{\boldsymbol{a\in A\setminus a_1}} \boldsymbol{Q^{\pi}(s, a)}\right)$. The idea is to favor solutions that make any singe-step deviation from $\boldsymbol{\pi}$ as costly as possible.
- The penalty term takes into account the preference for simple reward functions. Norm-1 is used to induce as many null entries as possible in the reward function, as it happens in most real-case scenarios.
- The feasibility conditions enforces the belonging to the solution set of the previous theorem and the boundedness of the reward function.
- The optimization problem can be solved by standard linear programming techniques.

LINEAR PROGRAMMING FORMULATION OF IRL IN INFINITE STATE SPACES

- $S = \mathbb{R}^n$. We assume $R(s) = \alpha_1\phi_1(s) + \ldots + \alpha_d\phi_d(s)$, for some fixed basis $\phi_1, \ldots, \phi_d$ and unknown $\alpha_1, \ldots, \alpha_d$.
- By the linearity of expectations: $V^\pi = \alpha_1 V_1^\pi + \ldots + \alpha_d V_d^\pi$, where V_i^π is the value function of policy π when $R = \phi_i$. We also assume the availability of a subroutine for approximating V^π.
- The appropriate generalization to this hypothesis of the membership condition to the solution set of the IRL Problem is given by: $\mathbb{E}_{s' \sim P_{sa_1}}[V^\pi(s')] \geq \mathbb{E}_{s' \sim P_{sa}}[V^\pi(s')]$ for all states s and all actions $a \in A \setminus a_1$ (it also follows directly from Bellman equations and Bellman optimality with some algebra). This represent a set of linear constraints on the α_i's.
- As the number of states is infinite, we consider a large and finite subsample of states S_0.
- A linear reward function does not necessarily exists, so we relax the constraints but penalize their violations.
- The optimization problem, whose unknown variables are the α_i's, still solvable by standard linear programming techniques, is then:

$$maximize \sum_{s \in S_0} \min_{a \in \{a_2, \ldots, a_k\}} \left\{ p\left(\mathbb{E}_{s' \sim P_{sa_1}}[V^\pi(s')] - \mathbb{E}_{s' \sim P_{sa}}[V^\pi(s')] \right) \right\} \qquad s.t. \quad |\alpha_i| \leq 1, \qquad i = 1, 2, \ldots, d;$$

where $p(x) = \begin{cases} x \text{ } if \text{ } x \geq 0 \\ 2x \text{ } if \text{ } x < 0 \end{cases}$ is the penalization function (2 is a heursitically chosen penalty weight).

IRL FROM SAMPLED TRAJECTORIES

- Our aim is to find $\boldsymbol{R}$ s.t. the unknown policy $\boldsymbol{\pi}$ maximizes $\mathbb{E}_{s_0 \sim D}[V^{\pi}(s_0)]$ for some fixed initial state distribution $\boldsymbol{D}$. We then assume w.l.o.g. that there is only one fixed start state s_0 whose next-state distribution is $\boldsymbol{D}$. We are given a set of trajectories of an assumed optimal expert policy $\boldsymbol{\pi}^*$.
- $\boldsymbol{R}$ is still a linear approximator in α_i's, and we assume to have the ability of simulate trajectories in the MDP for any given $\boldsymbol{\pi}$.
- To estimate $V^{\pi}(s_0)$ for a given $\boldsymbol{\pi}$, we run $\boldsymbol{m}$ Monte Carlo trajectories under $\boldsymbol{\pi}$ and define $\hat{V}_i^{\pi}(s_0)$ to be what the average empirical return would have been on these $\boldsymbol{m}$ trajectories if the reward had been $R = \phi_i$, for each $i = 1, 2, \dots, d$. Thus, we can estimate $V^{\pi}(s_0)$ by $\hat{V}^{\pi}(s_0) = \alpha_1 \hat{V}_1^{\pi}(s_0) + \cdots + \alpha_d \hat{V}_d^{\pi}(s_0)$.
- We construct an algorithm whose idea is to <u>iteratively</u> improve $\boldsymbol{R}$ by comparing V^{π^*} with the value functions of a sequence of policy generated by the algorithm which hopefully converges to $\boldsymbol{\pi}^*$. We start the algorithm by computing V^{π^*} and V^{π_1} for a random initial policy π_1 and then repeat, at each step $\boldsymbol{k}$, until convergence:

1. Solve for the α_i's:

$$\text{maximize} \sum_{i=1}^{k} p\left(\hat{V}^{\pi^*}(s_0) - \hat{V}^{\pi_i}(s_0)\right)$$
$$\text{s.t.} \ |\alpha_i| \leq 1, i = 1, \dots, d.$$

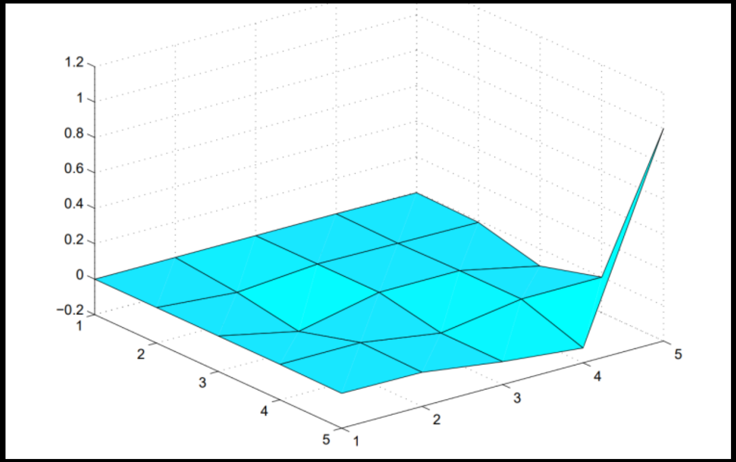
2. Update $R = \alpha_1 \phi_1 + \ \dots \ + \alpha_d \phi_d$.
3. Compute π_{k+1}, the optimal policy under $\boldsymbol{R}$.

EXPERIMENTAL RESULTS

Finite State Space IRL

5x5 Gridworld Benchmark

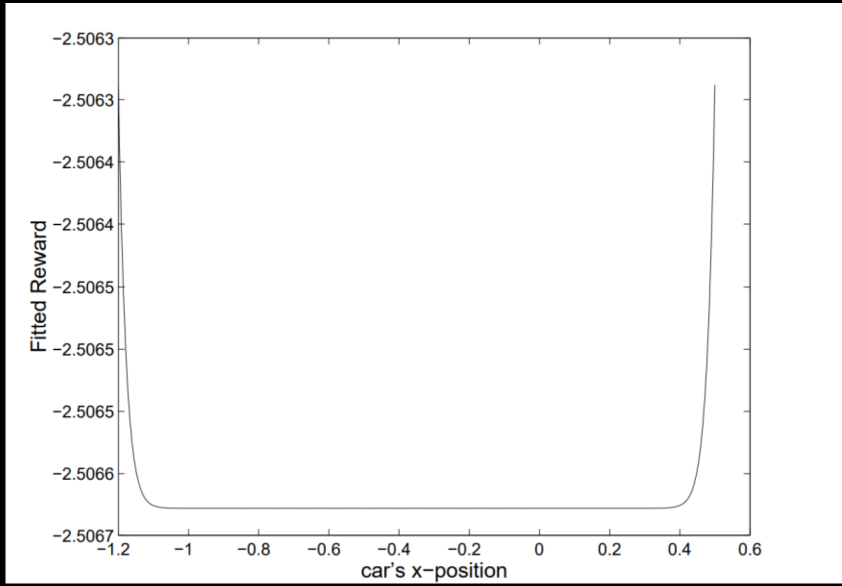
The proposed algorithm manages to approximate the reward function of the problem with fairly good precision, apart from some negligible oscillations along z-axis.



Continuos State Space IRL

Mountain-Car Benchmark

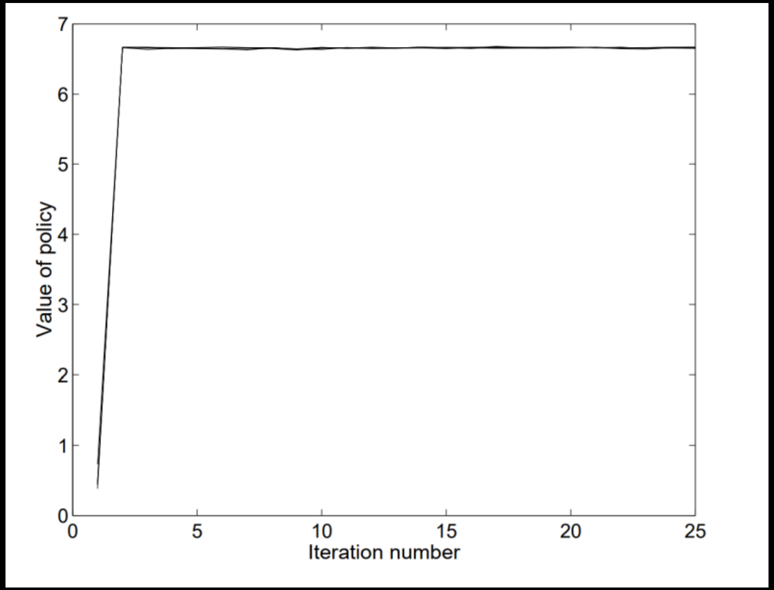
The proposed algorithm manages to capture perfectly the $\boldsymbol{R} = -\boldsymbol{c}$ structure of the true reward function.



IRL From Trajectories

$[\mathbf{0}, \mathbf{1}] \times [\mathbf{0}, \mathbf{1}]$ *Gridworld Benchmark*

The proposed algorithm manages to find a reward function whose optimal policy value has no statistically significant differences with the value of the optimal policy under the true reward function.



CONCLUSIONS AND CONSIDERATIONS

- The experimental results show that IRL is soluble, at least for moderate-sized state spaces.

- Further research topics:
  - How the generalize the presented approach to the case of Partially Observable MDP?
  - Some particular reward functions (for instance those produced by potential-based shaping rewards) makes the RL problem dramatically easier to solve. Can we modify the presented approach to improve the performances on such simple problems?
  - How can we deal with the intrinsic noise in the observer's measurements and with the suboptimality of the agent's policy? What are appropriate metrics for representing and analyzing such situations?