Inverse Reinforcement Learning Using Policy Gradient Minimization

Gulshan Kumar, Nirvan Singhania, Aman Bansal

International Institute of Information Technology, Hyderabad

November 26, 2019

Outline

1. Introduction
2. Closed Form Solution
3. Main Approach in Paper
4. Pareto Optimality
5. Linear Reward Setting
6. Geometric Interpretation
7. Analysis of Linear Reward Setting
8. Experiments
9. References

Introduction

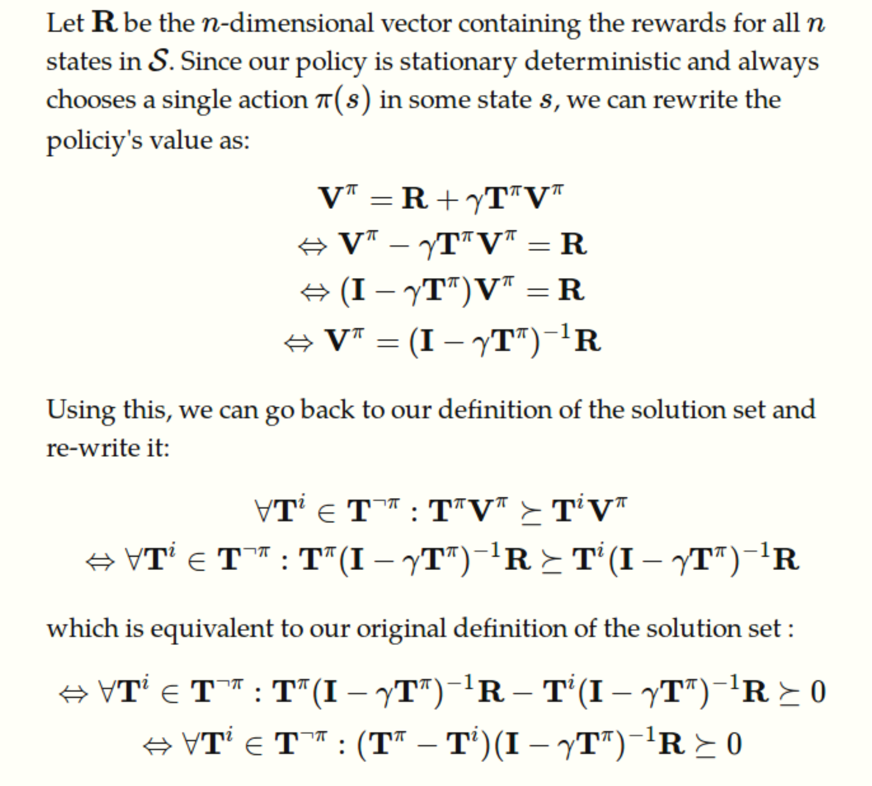
RL vs IRL

- Inverse reinforcement learning (IRL) is the field of learning an agent's objectives, values, or rewards by observing its behavior.
- In RL, our agent is provided with a reward function which, whenever it executes an action in some state, provides feedback about the agent's performance.
- In IRL, the setting is (as the name suggests) inverse. We are now given some agent's policy or a history of behavior and we try to find a reward function that explains the given behavior.

Why IRL

- No natural source for the reward signal
- Observe an expert (human) and get well fitting reward function and extract the respective rewards from the observations (trajectories).

Closed Form Solution



Let $\mathbf{R}$ be the n-dimensional vector containing the rewards for all n states in $\mathcal{S}$. Since our policy is stationary deterministic and always chooses a single action $\pi(s)$ in some state s, we can rewrite the policiy's value as:

$$
\begin{aligned}
\mathbf{V}^{\pi} &= \mathbf{R} + \gamma \mathbf{T}^{\pi}\mathbf{V}^{\pi} \\
\Leftrightarrow \mathbf{V}^{\pi} - \gamma \mathbf{T}^{\pi}\mathbf{V}^{\pi} &= \mathbf{R} \\
\Leftrightarrow (\mathbf{I} - \gamma \mathbf{T}^{\pi})\mathbf{V}^{\pi} &= \mathbf{R} \\
\Leftrightarrow \mathbf{V}^{\pi} &= (\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R}
\end{aligned}
$$

Using this, we can go back to our definition of the solution set and re-write it:

$$
\begin{gathered}
\forall \mathbf{T}^{i} \in \mathbf{T}^{\neg\pi} : \mathbf{T}^{\pi}\mathbf{V}^{\pi} \succeq \mathbf{T}^{i}\mathbf{V}^{\pi} \\
\Leftrightarrow \forall \mathbf{T}^{i} \in \mathbf{T}^{\neg\pi} : \mathbf{T}^{\pi}(\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R} \succeq \mathbf{T}^{i}(\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R}
\end{gathered}
$$

which is equivalent to our original definition of the solution set :

$$
\begin{gathered}
\Leftrightarrow \forall \mathbf{T}^{i} \in \mathbf{T}^{\neg\pi} : \mathbf{T}^{\pi}(\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R} - \mathbf{T}^{i}(\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R} \succeq 0 \\
\Leftrightarrow \forall \mathbf{T}^{i} \in \mathbf{T}^{\neg\pi} : (\mathbf{T}^{\pi} - \mathbf{T}^{i})(\mathbf{I} - \gamma \mathbf{T}^{\pi})^{-1}\mathbf{R} \succeq 0
\end{gathered}
$$

Figure 1: A solution using linear equations

Main Approach in Paper

We observe the behavior of an expert that follows a policy π^E that is optimal w.r.t. some reward function R^E. They assume that R^E can be represented through a linear or nonlinear function $R(s, a; \omega)$, where $\omega \epsilon R^q$

$$J_D(\pi) = \int_S d_\mu^\pi(s) \int_A \pi(a; s) R(s, a; \omega) dads \tag{1}$$

Given a parametric (linear or non linear) reward function $R(s, a; \omega)$,the associate policy gradient can be computed-

$$\Delta_\theta J(\pi_\theta^E, \omega) = \int_S d_\mu^\pi(s) \int_A \Delta_\theta \pi^E(a; s; \theta) Q^{\pi^E}(s; a; \omega) dads \tag{2}$$

Aim of the Gradient Inverse Reinforcement Learning

If the policy performance $J(\pi, \omega)$ is differentiable w.r.t. the policy parameters θ and the expert π_θ^E is optimal w.r.t. a parametrization $R(s, a, \omega^E)$, the associated policy gradient is identically equal to zero.

Clearly, the expert′s policy π_θ^E is a stationary point for $J(\pi, \omega^E)$. The Gradient IRL (GIRL) algorithm aims at finding a stationary point of $J(\pi_\theta^E, \omega)$ w.r.t. the reward parameter ω, that is, for any $x, y \geq 1$

$$\omega^A = argmin_w C_x^y(\pi_\theta^E, \omega) = argmin_w \frac{1}{y} \left\| \Delta_\theta J(\pi_\theta^E, \omega) \right\|_x^y \tag{3}$$

GIRL Property

Key property of GIRL is its **property of convexity**, which is state by the following lemma:

Lemma 1:*Given a convex representation of the reward function* $R(s, a; \omega)$ *w.r.t. the reward parameters, the objective function* C_x^y, *with* $x, y \geq 1$, *is convex.*

Pareto Optimality

- Consider the case with $N = 2$ agents, indexed by $i = 1, 2$ with actions a_1, a_2
- Define utility for agent 1 as $U_1(a1, a2)$, for agent 2 as $U_2(a1, a2)$

Definition

The set of feasible actions (a_1^P, a_2^P) is Pareto optimal if there does not exist another set of feasible actions $(a1^{'}, a2^{'})$ such that $U_1(a1^{'}, a2^{'}) \geq U_1(a_1^P, a_2^P) and U_1(a1^{'}, a2^{'}) \geq U_1(a_1^P, a_2^P)$ with at least one of the above inequality strict.

Pareto Optimality (Contd.)

Utility Set

the set of all pairs of utility (U_1, U_2) given by all of the different actions a_1 and a_2.The *utility possibility set* is that collection $U = \big\{(U_1, U_2) : U_1 = U_1(a_1, a_2), U_2 = U_2(a_1, a_2)\big\}$ which can usually be represented by a graph with U_1 on the $x - axis$ and U_2 on the $y - axis$

- Brute algorithm: Check if the upper right half of the point in the graph has no other points.

Linear Reward Setting

Using linear parametrization of the reward function $R(s, a, \omega)$, we have the following equation

$$J(\pi, \omega) = \sum_{i=1}^{q} \omega_i J_i(\pi) \tag{4}$$

where $J_(\pi) \epsilon R^q$ are the unconditional feature expectations under policy π.

The above equation is a weighted sum of the objective function. We are interested in searching for solutions which make the Pareto optimal.For this we have the following we have as a solution a non-convex vector α such that the equation holds

$$\alpha \epsilon Null(D_\theta J(\pi)) \tag{5}$$

where $(D_\theta J(\pi))$ is the Jacobian for J.

Pareto Optimal Solution

A point is Pareto−stationary if there exists a convex combination of the individual gradients that generates an identically zero vector.

$$(D_\theta J(\pi))\alpha = 0 \tag{6}$$

$$\sum_{i=1}^{q} \alpha_i = 1, \alpha_i \geq 0 \tag{7}$$

A solution that is Pareto optimal is also Pareto−stationary.

When the solution lies outside the frontier it is possible to identify a set of directions (ascent cone) that simultaneously increase all the objectives.

Pareto Optimal Solution (Contd.)

When the solution belongs to the Pareto frontier, the ascent cone is empty because any change in the parameters will cause the decrease of at least one objective.

The convex combination α represents the scalarization, i.e., the reward parameters. In particular, notice that $\alpha = \omega$ coincides with the solution computed by GIRL. This result follows easily by noticing that

$$||\Delta J(\pi_\theta^E, \omega)||_x = ||D_\theta J(\pi^E,)\omega||_x \tag{8}$$

where $J_\pi(s,a) \epsilon R^q$ is the vector of conditional feature expectations given $s_0 = s$ and $a_0 = a$.

Geometric Interpretation

- Geometrically, the reward weights are orthogonal to the hyper plane tangent to the Pareto frontier (identified by the individual gradients $\nabla_\theta J_i(\pi^E)$, i = (1, ...q)
- Compute tangent hyperplane can be identified by the q points associated to the Gram matrix of $D_\theta J(\pi^E)$.

$$G = (D_\theta J(\pi^E))^T D_\theta J(\pi^E) \tag{9}$$

- Property of Gram Matrix: Null($A^T A$) = Null(A)
- Since the expert$'$s weights are orthogonal to such hyperplane, they are obtained by computing the null space of the matrix $G : \omega \ \epsilon$ Null(G). Given the individual gradients, the complexity of obtaining the weights is $O(q^2 d + q^3)$. This version of it is called Plane GIRL (PGIRL).

Analysis of Linear Reward Setting : Handling Degeneracy

- Possible sources of degeneracy are constant reward functions, duplicated features or useless features.
- The batch nature of the GIRL problem allows us to eliminate linear dependent features.

We consider the rows of the Jacobian matrix to be LI. As long as the policy parameters d is greater or equal than the number q of reward parameters ($q \leq d$), any deficiency in the rank is due to linear dependence among the objectives.

Handling Degeneracy (Contd.)

As a consequence, the Jacobian matrix can be reduced in order to contain only columns that are linearly independent. The weights for the removed columns is set to zero for not affecting the reward partameterization.

When the Gram matrix G is not full rank we do not have a unique solution.In this case the null space operator returns an orthonormal basis Z such that $D_\theta J(\pi^E).Z$ is a null matrix.

Bound on Errors

- When using trajectories, the estimation errors of the gradients imply an error in the expert weights estimated by PGIRL

- The authors have derived an upper bound to the $L2$-norm of the difference between the expert weights ω_E and the weights ω_A estimated by PGIRL, given $L2$-norm upper bounds on the gradient estimation errors.

Bound on Errors

Error

Let denote with g^i the estimated value of $\nabla_\theta J_i(\pi^E)$. Under the assumption that the reward function maximized by the expert is a linear combination of basis functions $\phi(s,a)$ with weights ω_E and that, for each i, $||\nabla_\theta J_i(\pi^E)$ - $g^i\ ||_2 \leq \epsilon_i$

$$||\omega_E - \omega_A||_2 \leq \sqrt{(2(1 - \sqrt{1 - (\epsilon/\rho)^2}} \tag{10}$$

where $\epsilon = \max_i \epsilon_i$ and ρ is the radius of the largest $(q-1)$- dimensional ball inscribed in the convex hull of points g^i

Approximated Expert Policy Parameters

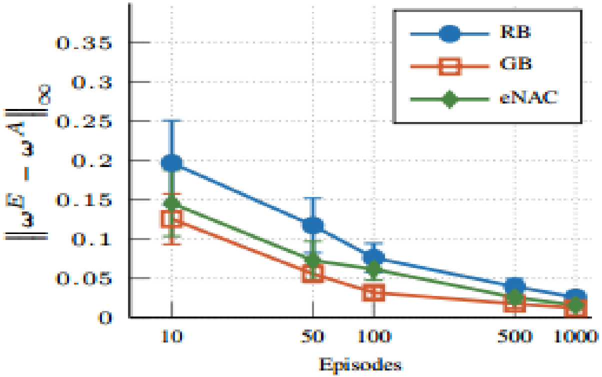
- When the expert′s policy is unknown, we need to infer a parametric policy model from a set of trajectories $\{\tau_i\}_{i=0}^{N}$ of length M
- Given a parametric policy model $\pi(a; s, \theta)$ a parametric density estimation problem can be defined as a Maximum Likelihood estimation (MLE) problem : $\widehat{\theta}_{MLE} = argmax_{\theta} \prod_{i=1}^{N.M} \pi(a_i; s_i, \theta)$

Paper Experiment

Linear Quadratic Regulator: They provide a set of experiments in the well−known Linear Quadratic Regulator (LQR) problem. Experiments are meant to be a proof of concept of algorithm behavior. We consider a linear parametrization of the reward function

$$R(s, a; \omega) = -\sum_{i=1}^{q} \omega_i (s^T Q_i s + a^T R_i a) \tag{11}$$

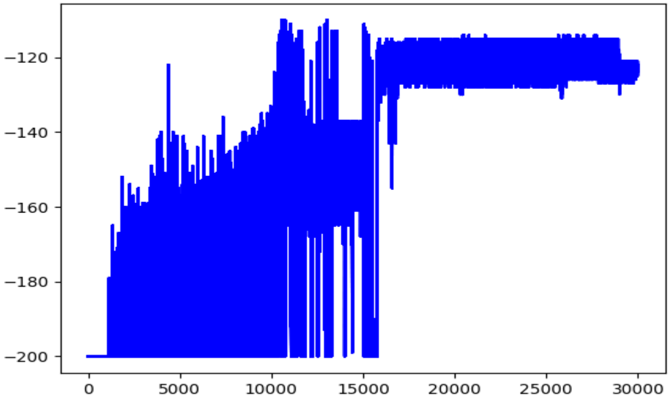
As the number of samples increases, the accuracy of the plane identified by the algorithm improves.



Our Results

Due to the complexity in the algorithm we couldn't evaluate this paper on some environment. We have written the code for sampling the expert trajectory for feeding into the network and also write the code for reward parameter estimation.

Simulated the *Maximum Entropy Inverse Reinforcement Learning* paper. We have evaluated the model on **MountainCar-v0**. The model is trained for 30000 episodes with $\gamma = 0.95$, $\eta = 0.05$.



References

- Matteo Pirotta, Marcello Restelli,Inverse Reinforcement Learning through Policy Gradient Minimization, Politecnico di Milano, Piazza Leonardo da Vinci, 32 I-20133, Milan, Italy
- Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, Anind K. Dey, Maximum Entropy Inverse Reinforcement Learning
- https://ai.stanford.edu/ ang/papers/icml00-irl.pdf
- https://github.com/albertometelli/crirl