Algorithms for Inverse Reinforcement Learning

Andrew Y. Ng ANG@CS.BERKELEY.EDU
Stuart Russell RUSSELL@CS.BERKELEY.EDU
Computer Science Division, U.C. Berkeley, Berkeley, CA 94720 USA

Introduction

What is IRL?

GIVEN

1. 다양한 상황에서 시간에 따른 에이전트의 행동들
2. 필요하다면, 에이전트들의 감각적 입력들
3. 가능하다면, 환경 모델이 주어질 때

DETERMINE

Reward function가 최적화된다.

IRL을 사용해야하는 2가지 이유

1. animal, human learning에 대한 computation model로서의 용도

agent : 벌
reward function : 꿀
reward function은 이미 고정, 알고 있음
그러나 reward를 받는 behavior를 조사할 때는 알고있지않은 reward function까지 고려해야한다.

2. 특정한 도메인에서 잘 행동할 수 있는 지능적인 에이전트 구성 가능

보통 에이전트를 만들 때, 디자이너들이 자기가 구성한 reward function의 optimization이 자신이 원한대로 행동을 만들것이라고 착각할 수 있음.
실제로는 그렇게 되지 않아서 디자이너들은 괴로움을 겪게된다.
이때 사용할 수 있는 것이 'expert's behavior'

+ expert behavior는 imitation learning과 apprenticeship learning에서 사용된다.

Section

finite
MDPs
&
IRL
problem

reward
function
&
optimal
policy

infinite
&
large
state
space

a finite
set of
observed
trajector
ies

Notation

2.1 Markov Decision Processes

A (finite) MDP is a tuple $(S, A, \{P_{sa}\}, \gamma, R)$, where

- S is a finite set of N **states**.
- $A = \{a_1, \ldots, a_k\}$ is a set of k **actions**.
- $P_{sa}(\cdot)$ are the state **transition probabilities** upon taking action a in state s.
- $\gamma \in [0, 1)$ is the **discount factor**.
- $R : S \mapsto \mathbb{R}$ is the **reinforcement function**, bounded in absolute value by $R_{\max}$.

* R(s,a)가 아닌 R(s)인 이유는
  R(s)가 더 확장성이 좋기 때문이다.

policy is defined as any map $\pi : S \mapsto A$,

value function for a policy π, evaluated at any state s_1 is given by

$$V^\pi(s_1) = \mathrm{E}\left[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + \cdots | \pi\right]$$

the **Q-function** according to

$$Q^\pi(s, a) = R(s) + \gamma \mathrm{E}_{s' \sim P_{sa}(\cdot)}\left[V^\pi(s')\right]$$

i. Finally, we let the symbols $\prec$ and $\preceq$ denote strict and non-strict vectorial inequality—i.e., $\boldsymbol{x} \prec \boldsymbol{y}$ if and only if $\forall i \; \boldsymbol{x}_i < \boldsymbol{y}_i$.

vector x와 y에 대한 부등식 기호로 $\prec$, $\preceq$ 사용.

Theorem 1) Bellman Equation

$$V^{\pi}(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s')V^{\pi}(s') \quad (1)$$

$$Q^{\pi}(s,a) = R(s) + \gamma \sum_{s'} P_{sa}(s')V^{\pi}(s') \quad (2)$$

Theorem 2) Bellman Optimality

$$\pi(s) \in \arg\max_{a \in A} Q^{\pi}(s,a) \quad (3)$$

IRL in Finite State Spaces

Theorem 3) reward R은 다음을 만족한다.

$$(\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \succeq 0 \qquad (4)$$

$\pi(s) \equiv a_1$기 때문에, (1)을 다음과 같이 쓸 수 있고 따라서 (5)가 된다.

$$V^\pi(s) \quad = \quad R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s') V^\pi(s') \quad (1)$$

$$\Downarrow$$

$$\boldsymbol{V}^\pi = \boldsymbol{R} + \gamma \boldsymbol{P}_{a_1} \boldsymbol{V}^\pi$$

$$\boldsymbol{V}^\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \qquad (5)$$

IRL in Finite State Spaces

문제점

1) R = 0은 항상 solution이 된다.
   즉, 어떤 행동을 취하더라도 reward가 같으면 그 policy가 항상 optimal하다는 것이다.

2) 대부분 MDPs에서는 (4)번 기준을 만족하는 R이 여러개 있다.

그렇다면 여러 개의 R들 중 어떤 것을 선택해야할까?

IRL in Finite State Spaces

여러 개들 중 max 값을 갖는 R을 선택하자

$$\sum_{s\in S}\left(Q^{\pi}(s,a_1)-\max_{a\in A\setminus a_1}Q^{\pi}(s,a)\right) \qquad (6)$$

여기에 $-\lambda||R||_1$ term을 두어서 차이를 더 극명하게 함

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{N}\min_{a\in\{a_2,\dots,a_k\}}\{(\boldsymbol{P}_{a_1}(i)-\boldsymbol{P}_{a}(i)) \\
& \quad (\boldsymbol{I}-\gamma\boldsymbol{P}_{a_1})^{-1}\boldsymbol{R}\}-\lambda||\boldsymbol{R}||_1 \\
\text{s.t.} \quad & (\boldsymbol{P}_{a_1}-\boldsymbol{P}_{a})(\boldsymbol{I}-\gamma\boldsymbol{P}_{a_1})^{-1}\boldsymbol{R}\succeq 0 \\
& \qquad \forall a\in A\setminus a_1 \\
& |\boldsymbol{R}_i|\le R_{\max},\ \ i=1,\dots,N
\end{aligned}$$

Linear Function Approximation in Large State Space

Infinite space에서의 경우,
reward function에 대한 linear approximation은 다음과 같다.

$$R(s) = \alpha_1 \phi_1(s) + \alpha_2 \phi_2(s) + \cdots + \alpha_d \phi_d(s) \qquad (8)$$

우리는 α를 fit해야함

value function에 대해서도 linear하게 표현할 수 있다.

$$V^\pi = \alpha_1 V_1^\pi + \cdots + \alpha_d V_d^\pi . \qquad (9)$$

Linear Function Approximation in Large State Space

(3)과 (9)를 사용하고, $\pi(s) \equiv a_1$ optimal하게 만들기위해 R에 대해 (4)의 적절한 일반화가 (10)의 조건임을 검증할 수 있다.

Theorem 2) Bellman Optimality

$$\pi(s) \in \arg\max_{a \in A} Q^{\pi}(s,a) \qquad (3)$$

$$V^{\pi} = \alpha_1 V_1^{\pi} + \cdots + \alpha_d V_d^{\pi}. \qquad (9)$$

$$(\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \succeq 0 \qquad (4)$$

$$\mathrm{E}_{s' \sim P_{sa_1}}[V^{\pi}(s')] \geq \mathrm{E}_{s' \sim P_{sa}}[V^{\pi}(s')] \qquad (10)$$

Linear Function Approximation in Large State Space

문제점

1) Infinite state여서 모든 state를 확인할 수 없다.

   → state sampling

2) R을 표현하기 위해서 linear function approximator를 사용하게되면, 어느 policy가 optimal한지 더이상 reward function을 표현할 수가 없게 된다.

   → linear function approximator 계속 사용

Linear Function Approximation in Large State Space

$$\begin{aligned}
\text{maximize } & \textstyle\sum_{s \in S_0} \min_{a \in \{a_2, \dots, a_k\}} \{ \\
& \quad p(\mathrm{E}_{s' \sim P_{sa_1}}[V^\pi(s')] - \mathrm{E}_{s' \sim P_{sa}}[V^\pi(s')])\} \\
\text{s.t. } & |\alpha_i| \leq 1, \quad i = 1, \dots, d
\end{aligned}$$

Linear Function Approximation in Large State Space

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{N} \min_{a \in \{a_2, \dots, a_k\}} \{ (\boldsymbol{P}_{a_1}(i) - \boldsymbol{P}_a(i)) \\
& \qquad (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \} - \lambda ||\boldsymbol{R}||_1 \\
\text{s.t.} \quad & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \succeq 0 \\
& \qquad \forall a \in A \setminus a_1 \\
& |\boldsymbol{R}_i| \leq R_{\max}, \;\; i = 1, \dots, N
\end{aligned}
$$

$$
\begin{aligned}
\text{maximize} & \textstyle\sum_{s \in S_0} \min_{a \in \{a_2, \dots, a_k\}} \{ \\
& p(\mathrm{E}_{s' \sim P_{sa_1}}[V^\pi(s')] - \mathrm{E}_{s' \sim P_{sa}}[V^\pi(s')]) \} \\
\text{s.t.} & \; |\alpha_i| \leq 1, \;\; i = 1, \dots, d
\end{aligned}
$$

IRL from Sampled Trajectories

오로지 trajectories의 set으로 policy에 접근함
그렇기 때문에 model이 필요하지 않음

초기 state distribution D를 고정시켜놓고 알지못하는 policy π에 대해서
π가 $\mathrm{E}_{s_0 \sim D}[V^\pi(s_0)]$ maximize하는 R을 찾자

일단 처음에 α_i의 setting으로 $V^\pi(s_0)$ estimate 해야한다.

IRL from Sampled Trajectories

estimate하기 위해서 m MC trajectories를 생성한다.
만약 Reward가 $R = \phi_i$면, $V^\pi(s_0)$m trajectories에 average empirical return이 얼마나 있었는지로 정의함.

ex) m = 1 trajectory, trajectory가 (s0, s1, s2, …)의 시퀀스라면

$$\hat{V}_i^\pi(s_0) = \phi_i(s_0) + \gamma\phi_i(s_1) + \gamma^2\phi_i(s_2) + \cdots$$

$\hat{V}^\pi(s_0)$ 다음과 같게 된다.

$$\hat{V}^\pi(s_0) = \alpha_1\hat{V}_1^\pi(s_0) + \cdots + \alpha_d\hat{V}_d^\pi(s_0) \qquad (11)$$

IRL from Sampled Trajectories

α_i의 setting은 귀납적으로 보면,
policy들의 집합 {π_1, ..., π_k}가 있고 reward function은 (12)수식을 만족하기 때문에 알 수 있다.

$$V^{\pi^*}(s_0) \geq V^{\pi_i}(s_0), \quad i = 1, \ldots, k \tag{12}$$

최종적은 수식은 section 4에서 변형하면 다음과 같다.

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

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{N} \min_{a \in \{a_2,\ldots,a_k\}} \{(\boldsymbol{P}_{a_1}(i) - \boldsymbol{P}_a(i)) \\
& \qquad (\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R}\} - \lambda ||\boldsymbol{R}||_1 \\
\text{s.t.} \quad & (\boldsymbol{P}_{a_1} - \boldsymbol{P}_a)(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1})^{-1} \boldsymbol{R} \succeq 0 \\
& \qquad\qquad \forall a \in A \setminus a_1 \\
& |\boldsymbol{R}_i| \leq R_{\max}, \quad i = 1, \ldots, N
\end{aligned}
$$

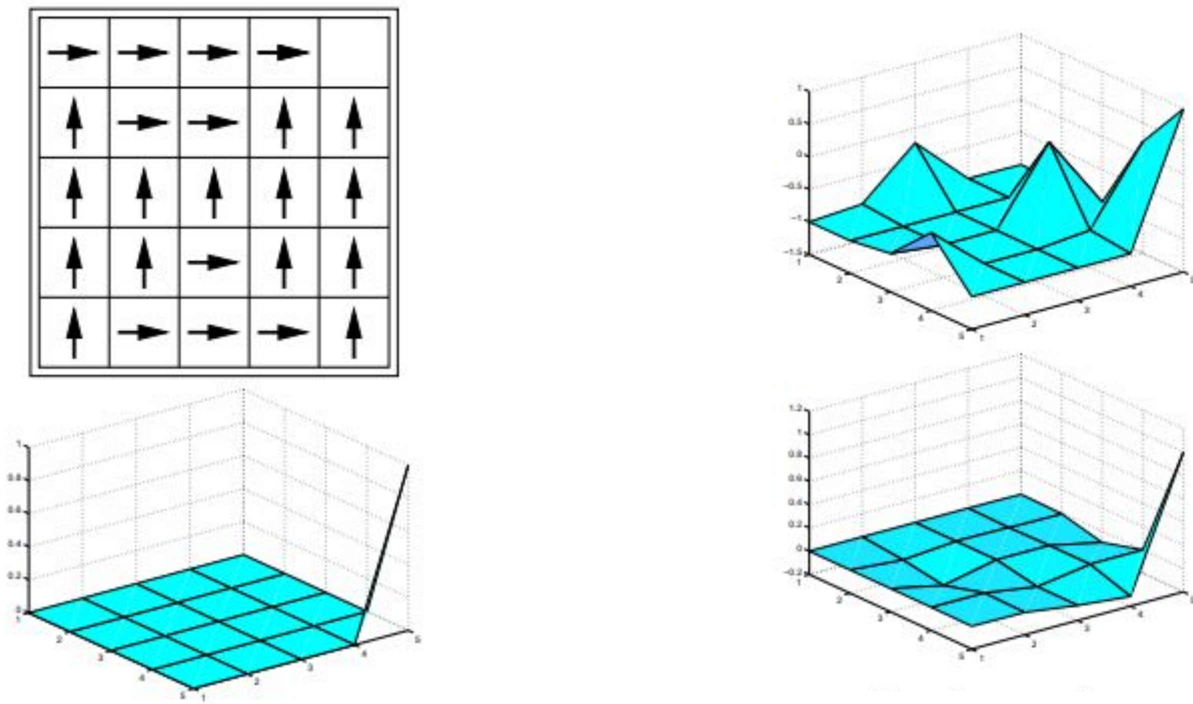
$$\Downarrow$$

$$
\begin{aligned}
\text{maximize} \ & \textstyle\sum_{s \in S_0} \min_{a \in \{a_2,\ldots,a_k\}} \{ \\
& \quad p(\mathrm{E}_{s' \sim P_{sa_1}} [V^{\pi}(s')] - \mathrm{E}_{s' \sim P_{sa}} [V^{\pi}(s')])\} \\
\text{s.t.} \ & |\alpha_i| \leq 1, \quad i = 1, \ldots, d
\end{aligned}
$$

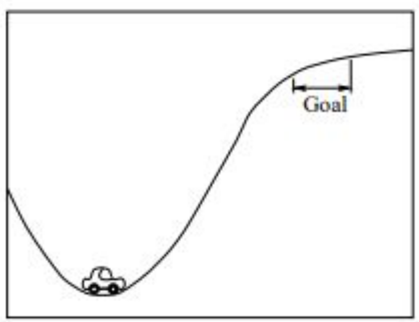
$$\Downarrow$$

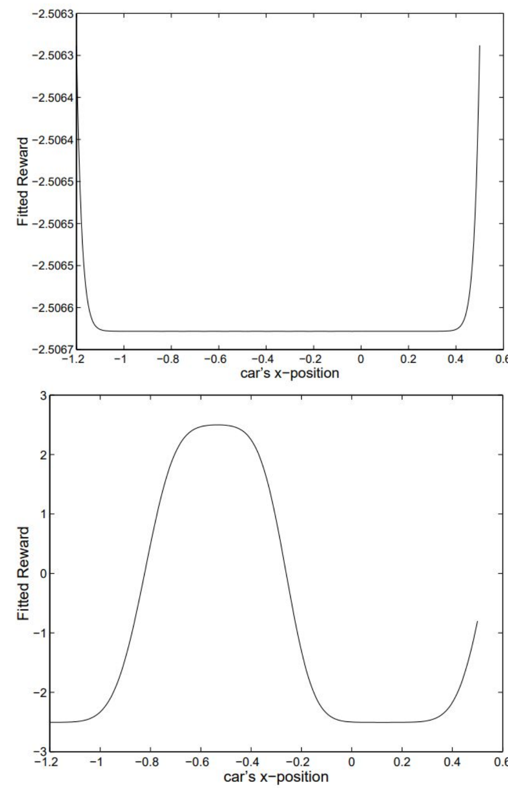
$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{i=1}^{k} p\left(\hat{V}^{\pi^*}(s_0) - \hat{V}^{\pi_i}(s_0)\right) \\
\text{s.t.} \quad & |\alpha_i| \leq 1, \quad i = 1, \ldots, d
\end{aligned}
$$

Experiments - finite state space

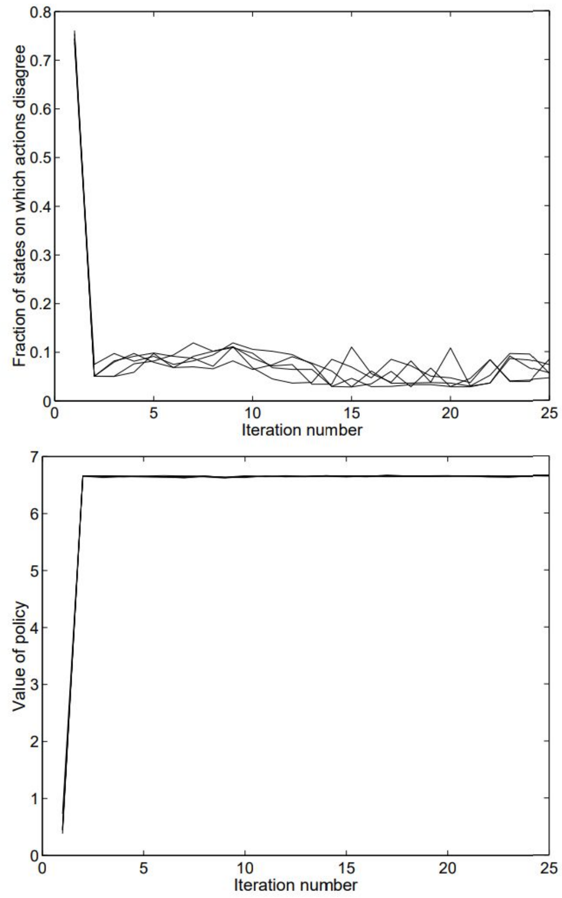


Experiments - continuous state space





Experiments - continuous version of 5x5 grid world



Conclusion

open question

1. potential-based shaping rewads는 MDP에서 학습시키기 위한 하나의 solution으로써 reward function을 더 쉽게 만들 수 있다.
   그렇다면 더 easy한 reward function을 만들기 위한 IRL 알고리즘들을 만들 수 있을까?
2. IRL을 현실 적용 측면에서 보면 sensor inputs and actions에 대해서 observe 측정에서 많은 noise가 있을지도 모른다. 게다가 많은 optimal policy들도 존재할 수 있다. 어떤 data를 noise없이 fit하게하는 적절한 metric이 무엇일까?
3. 행동이 절대로 optimality와 일치하지 않는다면 state space에 specific region에 대한 locally consistent reward function을 어떻게 알 수 있을까?
4. reward function의 identifiability를 최대화하기 위한 실험을 어떻게 고안할 수 있을까?
5. 본 연구의 알고리즘적인 접근이 partially observable한 경우에는 얼마나 잘 실행될까?