Reinforcement Learning

Inverse Reinforcement Learning

Inverse RL, behaviour cloning, apprenticeship learning, imitation learning.

Vien Ngo
MLR, University of Stuttgart

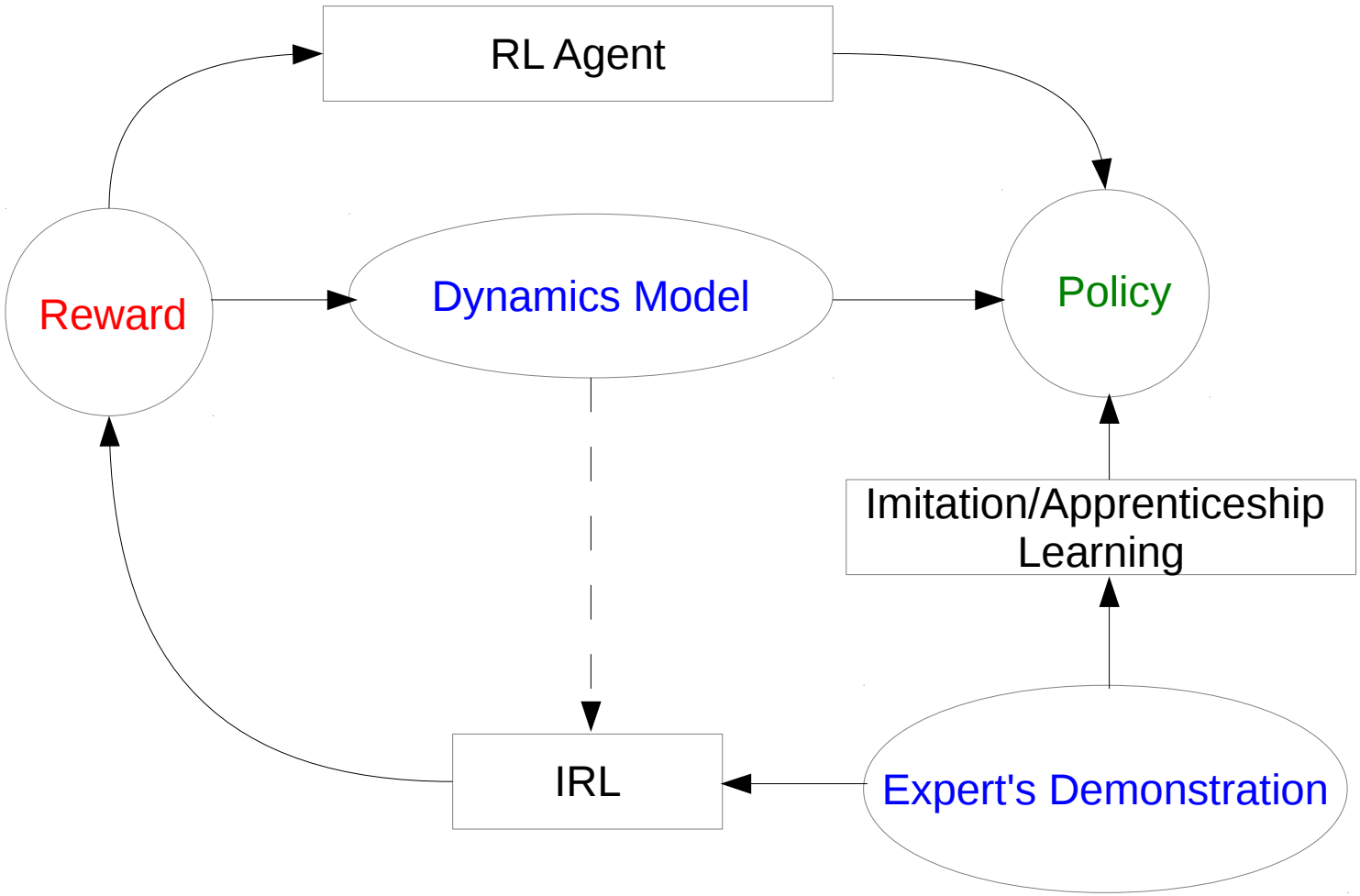
Outline

- Introduction to Inverse RL
- Inverse RL vs. behavioral cloning
- IRL algorithms
  (Inspired from a lecture from Pieter Abbeel.)

Inverse RL: Informal Definition

- **Given** Measurements of an agent's behaviour π over time (s_t, a_t, s'_t), in different circumstances. If possible, given transition model (not given reward function).
- **Goal**: Find the reward function $R^{\pi}(s, a, s')$.

Inverse Reinforcement Learning



adapted from a poster of Boularias, Kober, Peters.

Motivation: Two Sources

- The potential use of RL/related methods as computational model for animal and human learning: bee foraging (Montague et al 1995), song-bird vocalization (Doya & Sejnowski 1995), ...

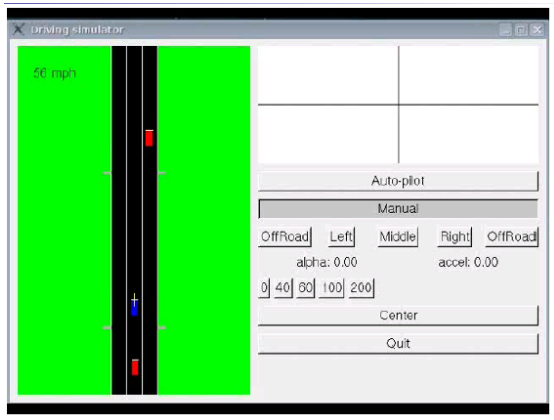
Motivation: Two Sources

- The potential use of RL/related methods as computational model for animal and human learning: bee foraging (Montague et al 1995), song-bird vocalization (Doya & Sejnowski 1995), ...
- Construction of an intelligent agent in a particular domain: Car driver, helicopter (Ng et al),... (imitation learning, apprenticeship learning)

Examples

- Car driving simulation

  Abbeel et al 2004, etc.



- Autonomous Helicopter Flight

  Andrew Ng et. al.

- Urban navigation

  Ziebart, Maas, Bagnell and Dey, AAAI 2008 (route recommendation, and destination prediction)

- etc.

Problem Formulation

- Given
  - State space $\mathcal{S}$, action space cA.
  - Transition model $T(s, a, s') = P(s'|s, a)$
  - not given reward function $R(s, a, s')$.
  - Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$
- Inverse RL:
  - Recover R.

Problem Formulation

- Given
  - State space $\mathcal{S}$, action space cA.
  - Transition model $T(s, a, s') = P(s'|s, a)$
  - not given reward function $R(s, a, s')$.
  - Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$
- Inverse RL:
  - Recover R.
- Apprenticeship learning via IRL
  - Use R to compute a good policy.
- Behaviour cloning:
  - Using supersived-learning to learn the teacher's policy.

IRL vs. behavioral cloning

IRL vs. Behavioral cloning

- Behavioral cloning: Formulated as a supervised-learning problem. (Using SVM, Neural networks, deep learning,...)
  – Given $(s_0, a_0), (s_1, a_1), \ldots,$ generated from a policy π^*.
  – Estimate a policy mapping s to a.

IRL vs. Behavioral cloning

- Behavioral cloning: Formulated as a supervised-learning problem. (Using SVM, Neural networks, deep learning,...)
  – Given $(s_0, a_0), (s_1, a_1), \ldots,$ generated from a policy π^*.
  – Estimate a policy mapping s to a.
- Behavioral cloning: can only mimic the trajectory of the teacher, then can not: with change of goal/destination, and non-Markovian environment (e.g. car driving).

IRL vs. Behavioral cloning

- Behavioral cloning: Formulated as a supervised-learning problem. (Using SVM, Neural networks, deep learning,...)
  – Given $(s_0, a_0), (s_1, a_1), \ldots,$ generated from a policy π^*.
  – Estimate a policy mapping s to a.
- Behavioral cloning: can only mimic the trajectory of the teacher, then can not: with change of goal/destination, and non-Markovian environment (e.g. car driving).
- IRL vs. Behavioral cloning is $\hat{R}^*$ vs. $\hat{\pi}^*$.

Inverse Reinforcement Learning

IRL: Mathematical Formulation

- **Given**
  - State space $\mathcal{S}$, action space cA.
  - Transition model $T(s, a, s') = P(s'|s, a)$
  - not given reward function $R(s, a, s')$.
  - Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$

IRL: Mathematical Formulation

- **Given**
  - State space $\mathcal{S}$, action space cA.
  - Transition model $T(s, a, s') = P(s'|s, a)$
  - not given reward function $R(s, a, s')$.
  - Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$
- **Find** R^*, such that

$$E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*\Big] \geq E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi\Big], \forall \pi$$

IRL: Mathematical Formulation

- **Given**
  – State space $\mathcal{S}$, action space cA.
  – Transition model $T(s, a, s') = P(s'|s, a)$
  – not given reward function $R(s, a, s')$.
  – Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$
- **Find** R^*, such that

$$E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*\Big] \geq E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi\Big], \forall \pi$$

- Challenges?
  – $R = 0$ is a solution (rewrad function ambiguity), and multiple R^* satisfy the above condition.
  – π^* is only given partially through trajectories, then how to evaluate the expectation terms.
  – must assume the expert is optimal

IRL: Mathematical Formulation

- **Given**
  - State space $\mathcal{S}$, action space cA.
  - Transition model $T(s, a, s') = P(s'|s, a)$
  - not given reward function $R(s, a, s')$.
  - Teacher's demonstration (from teacher's policy π^*): $s_0, a_0, s_1, a_1, \ldots,$
- **Find** R^*, such that

$$E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*\Big] \geq E\Big[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi\Big], \forall \pi$$

- Challenges?
  - $R = 0$ is a solution (rewrad function ambiguity), and multiple R^* satisfy the above condition.
  - π^* is only given partially through trajectories, then how to evaluate the expectation terms.
  - must assume the expert is optimal
  - R.H.S is computationally expensive, i.e enumerate all policies

IRL: Finite state spaces

- Bellman equations

$$V^{\pi} = (I - \gamma P^{\pi})^{-1} R$$

- Then IRL finds R such that

$$(P^{\pi^*} - P^{\pi})(I - \gamma P^{\pi^*})^{-1} R \geq 0, \forall \pi \in \Pi$$

(if consider only deterministic policies)

IRL: Finite state spaces

- Bellman equations

$$V^{\pi} = (I - \gamma P^{\pi})^{-1} R$$

- Then IRL finds R such that

$$(P^{\pi^*} - P^{\pi})(I - \gamma P^{\pi^*})^{-1} R \geq 0, \forall \pi \in \Pi$$

(if consider only deterministic policies)

- IRL as linear programming with l_1

$$\max \sum_{i=1}^{|S|} \min_{b \in \mathcal{A}/a^*} \left\{ (P^{a^*}(i) - P^{b}(i))(I - \gamma P^{a^*})^{-1} R \right\} - \lambda \|R\|_1$$

s.t.

$$(P^{\pi^*} - P^{\pi})(I - \gamma P^{\pi^*})^{-1} R \geq 0$$
$$R(i) \leq R_{max}$$

– Maximize the sum of differences between the values of the optimal action and the next-best.
– With l_1 penalty: simple reward functions (e.g. most states have zero rewards, some *good* states have rewards)

IRL: With FA in large state spaces

- Using FA: $R(s) = w^{\top}.\phi(s)$, where $w \in R^n$, and $\phi : \mathcal{S} \mapsto R$.

IRL: With FA in large state spaces

- Using FA: $R(s) = w^\top.\phi(s)$, where $w \in R^n$, and $\phi : \mathcal{S} \mapsto R$.
- Thus,

$$\begin{aligned} E\Big[\gamma^t R(s_t)|\pi\Big] &= E\Big[\gamma^t w^\top \phi(s_t)|\pi\Big] \\ &= w^\top E\Big[\gamma^t \phi(s_t)|\pi\Big] = w^\top.\eta(\pi) \end{aligned}$$

IRL: With FA in large state spaces

- Using FA: $R(s) = w^\top.\phi(s)$, where $w \in R^n$, and $\phi : \mathcal{S} \mapsto R$.
- Thus,

$$\begin{aligned} E\Big[\gamma^t R(s_t)|\pi\Big] &= E\Big[\gamma^t w^\top \phi(s_t)|\pi\Big] \\ &= w^\top E\Big[\gamma^t \phi(s_t)|\pi\Big] = w^\top.\eta(\pi) \end{aligned}$$

- The optimization problem: finding w^* such that

$$w^{*\top}.\eta(\pi^*) \geq w^{*\top}.\eta(\pi)$$

- $\eta(\pi)$ can be evaluated with sampled trajectories from π.

$$\eta(\pi) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_i} \gamma^t \phi(s_t)$$

Apprenticeship learning

Abbeel & Ng, 2004

Apprenticeship learning

- Finding a policy π whose performance is as close to the expert policy's performance as possible

$$\|w^{*\top}.\eta(\pi^*) - w^{\top}.\eta(\pi)\| \leq \epsilon$$

Apprenticeship learning

- Finding a policy π whose performance is as close to the expert policy's performance as possible

$$\|w^{*\top}.\eta(\pi^*) - w^\top.\eta(\pi)\| \leq \epsilon$$

1: Assume $R(s) = w^\top.\phi(s)$, where $w \in R^n$, and $\phi : \mathcal{S} \mapsto R$.
2: Initialize π_0
3: **for** $i = 1, 2, \ldots$ **do**
4: Find a reward function such that the teacher maximally outperforms all previously found controllers.

$$\max_{\gamma, \|w\| \leq 1} \|\gamma\|$$

s.t.

$$w^\top.\eta(\pi) \geq w^\top.\eta(\pi) + \gamma, \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

5: Find optimal policy π_i for the reward function R_w w.r.t current w

Examples

Simulated Highway Driving

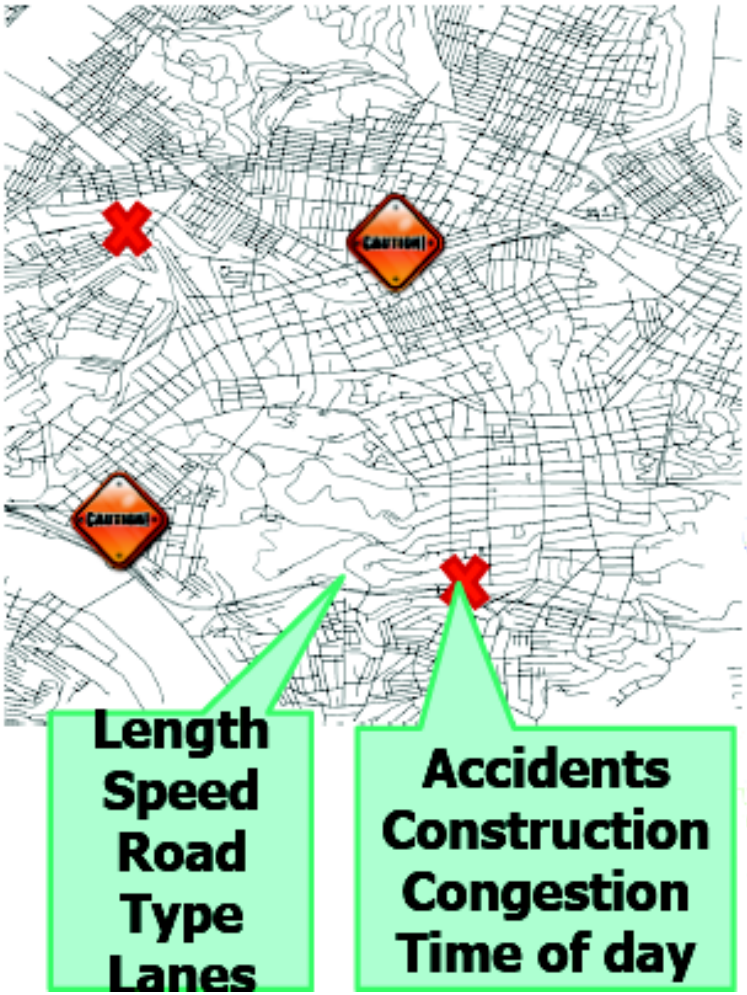
- Given dynamic model $T(s, a, s')$
- Each teacher demonstrates 1 minute.

Abbeel et. al. 2004

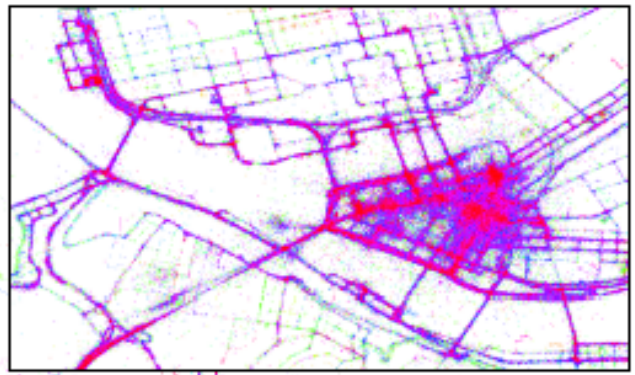
Simulated Highway Driving

expert demonstration (left), learned control (right)

Urban Navigation



25 Taxi Drivers



picture from a tutorial of Pieter Abbeel.

References

Andrew Y. Ng, Stuart J. Russell: Algorithms for Inverse Reinforcement Learning. ICML 2000: 663-670

Pieter Abbeel, Andrew Y. Ng: Apprenticeship learning via inverse reinforcement learning. ICML 2004

Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng: An Application of Reinforcement Learning to Aerobatic Helicopter Flight. NIPS 2006: 1-8

Adam Coates, Pieter Abbeel, Andrew Y. Ng: Apprenticeship learning for helicopter control. Commun. ACM 52(7): 97-105 (2009)