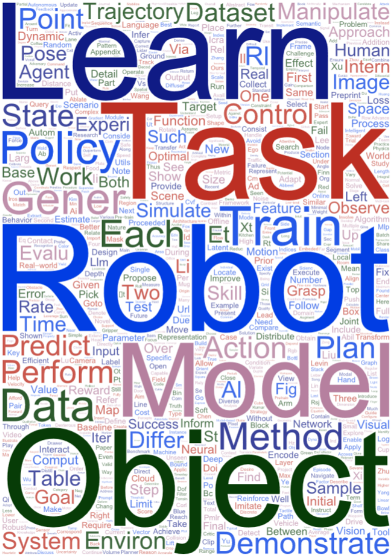


Robot Learning

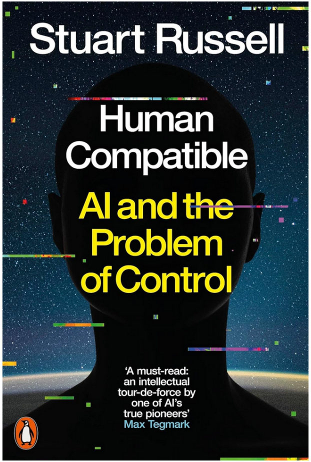
Inverse RL

Marc Toussaint
Technical University of Berlin

Summer 2024

Outline

- Value Alignment
- Inverse RL
- Preference-based RL



- Stuart Russell
  - Russell & Norvig: *Artificial Intelligence: A Modern Approach* (1995)
  - Decision & Game Theory

S. Russell. *Human compatible: AI and the problem of control.* 2019.
URL: https://books.google.com/books?hl=en&lr=&id=Gg-TDwAAQBAJ&oi=fnd&pg=PT8&dq=human+compatible+russell&ots=qoZKXK7gQ0&sig=p4x57HjxfMAVCpQ40_XcE7J4ECY

Russell: Value Alignment

- "Standard model of AI"
  - Define fixed objective; maximize

Russell: Value Alignment

- “Standard model of AI”
  - Define fixed objective; maximize

- Difficulty in defining objectives
  - Consequences (aspects of optimal behavior) unclear
  - Humans are bad at defining objectives

Russell: Value Alignment

- "Standard model of AI"
  - Define fixed objective; maximize

- Difficulty in defining objectives
  - Consequences (aspects of optimal behavior) unclear
  - Humans are bad at defining objectives

- Russell's proposal:
  - Systems should infer human preferences from behavior
  - Avoid overfitting
  - Large apriori uncertainty (incl. noise assumption in human behavior) to avoid overfitting

Cooperative Inverse Reinforcement Learning

Dylan Hadfield-Menell* **Anca Dragan** **Pieter Abbeel** **Stuart Russell**
Electrical Engineering and Computer Science
University of California at Berkeley
Berkeley, CA 94709

Abstract

For an autonomous system to be helpful to humans and to pose no unwarranted risks, it needs to align its values with those of the humans in its environment in such a way that its actions contribute to the maximization of value for the humans. We propose a formal definition of the value alignment problem as *cooperative inverse reinforcement learning* (CIRL). A CIRL problem is a cooperative, partial-information game with two agents, human and robot; both are rewarded according to the human's reward function, but the robot does not initially know what this is. In contrast to classical IRL, where the human is assumed to act optimally in isolation, optimal CIRL solutions produce behaviors such as active teaching, active learning, and communicative actions that are more effective in achieving value alignment. We show that computing optimal joint policies in CIRL games can be reduced to solving a POMDP, prove that optimality in isolation is suboptimal in CIRL, and derive an approximate CIRL algorithm.

D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning.
Advances in neural information processing systems, 29, 2016.
URL: https://proceedings.neurips.cc/paper_files/paper/2016/hash/c3395dd46c34fa7fd8d729d8cf88b7a8-Abstract.html

- Game-theoretic formalization of *Value Alignment*
  - ..is just one possible formulation
  - example for efforts to make “Value Alignment” more rigorous

Outline

- Value Alignment
- **Inverse RL**
- Preference-based RL

Inverse Reinforcement Learning

- Instance of **Imitation Learning**; recall:
  - Given expert demonstration data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n$ without external rewards/objectives/costs defined
  - Extract the "relevant information/model/policy" to reproduce demonstrations

Inverse Reinforcement Learning

- Instance of **Imitation Learning**; recall:
  - Given expert demonstration data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n$ without external rewards/objectives/costs defined
  - Extract the “relevant information/model/policy” to reproduce demonstrations

- Recap: Types of Imitation Learning
  - Behavior Cloning
  - Trajectory Distribution Learning (& Constraint Learning)
  - Direct (Interactive) Policy Learning (DAgger)
  - **Inverse Reinforcement Learning**
    - Builds on the full formalism of RL

Inverse Reinforcement Learning

- General Idea:
  - Given expert demonstration data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n$
  - **infer the reward function** assuming the demonstrated behavior is (approx.) optimal

Inverse Reinforcement Learning

- General Idea:
  - Given expert demonstration data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n$
  - **infer the reward function** assuming the demonstrated behavior is (approx.) optimal

- Benefits of understanding the reward function *behind* demonstrations:
  - Can apply and generalize to fully different domains, leading to different policy
  - Can be better than demonstrator

Inverse Reinforcement Learning

- Methods we discuss:
  - Max Margin IRL (Apprenticeship Learning)
  - Max Entropy IRL
  - Adversarial IRL

IRL: General Approach

- Recall the value of a policy π

$$J(\pi) = \mathbb{E}_{\xi \sim P_\pi}\{\textstyle\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\}$$

IRL: General Approach

- Recall the value of a policy π

$$J(\pi) = \mathbb{E}_{\xi\sim P_\pi}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\}$$

- Given a demonstration policy π^*, we want to find R such that for any other policy π:

$$J(\pi^*) \geq J(\pi) \quad \Leftrightarrow \quad \mathbb{E}_{\xi\sim P_{\pi^*}}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\} \geq \mathbb{E}_{\xi\sim P_\pi}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\}$$

IRL: General Approach

- Recall the value of a policy π

$$J(\pi) = \mathbb{E}_{\xi \sim P_\pi}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\}$$

- Given a demonstration policy π^*, we want to find R such that for any other policy π:

$$J(\pi^*) \ge J(\pi) \quad \Leftrightarrow \quad \mathbb{E}_{\xi \sim P_{\pi^*}}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\} \ge \mathbb{E}_{\xi \sim P_\pi}\{\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\}$$

- To simplify this, let's assume $R(s,a)$ is **linear in features** $\phi(s,a)$:

$$R(s,a) = w^\top \phi(s,a) = \sum_i w_i \phi_i(s,a) \tag{1}$$

$$\Rightarrow \quad J(\pi) = w^\top \mathbb{E}_\pi\{\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t)\} \overset{\Delta}{=} w^\top \mu(\pi) \tag{2}$$

and we want

$$\forall_{\pi \ne \pi^*}: \; w^\top \mu(\pi^*) \ge w^\top \mu(\pi)$$

Apprenticeship Learning

Apprenticeship Learning via Inverse Reinforcement Learning

Pieter Abbeel PABBEEL@CS.STANFORD.EDU
Andrew Y. Ng ANG@CS.STANFORD.EDU
Computer Science Department, Stanford University, Stanford, CA 94305, USA

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning.
In *Twenty-first international conference*, page 1, 2004.
URL: `http://portal.acm.org/citation.cfm?delete_doid=1015330.1015430`

Apprenticeship Learning

- First, π^* is not really given but
  - we estimate $\mu(\pi^*) = \mathbb{E}_{\pi^*}\{\sum_{t=0}^\infty \gamma^t \phi(s_t, a_t)\}$ from the demonstration data D
  - This $\mu(\pi^*)$ is the only information used from the demonstrations
- Second, we generate a series of other policies π_i against which we discriminate π^*
- Third, formulate "discrimination" as a max margin problem:

initialize π_0
for $i = 0, 1, 2, \dots$ **do**
  $w, t \leftarrow \text{argmax}_{w,t\in\mathbb{R}}\, t$ s.t. $\|w\| \le 1\ ,\quad \forall_{j\in\{0,..,i\}} :\ w^\top \mu(\pi^*) \ge w^\top \mu(\pi_j) + t$
  $\pi_{i+1} \leftarrow \text{argmax}_\pi\, J(\pi)$ **RL problem!**
end for

Maximum Entropy IRL

Maximum Entropy Inverse Reinforcement Learning

Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, and **Anind K. Dey**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bziebart@cs.cmu.edu, amaas@andrew.cmu.edu, dbagnell@ri.cmu.edu, anind@cs.cmu.edu

B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning

Maximum Entropy IRL

[skipping details]

- First, expert might be noisy, demonstrations ξ are assumed

$$P(\xi; w) = \frac{\exp\{w^\top \mu(\xi)\}}{\int \exp\{w^\top \mu(\xi')\}\, d\xi'}$$

- Second, find w that leads to max entropy $P(\cdot; w)$ but matches demonstrations:

$$\min_w \int P(\xi; w) \log P(\xi; w)\, d\xi$$

$$\text{s.t. } \mathbb{E}_{\xi \sim P(\xi; w)}\{\mu(\xi)\} = \mu(\pi^*)$$

Adversarial IRL

- Recall idea of GANs:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} \{\log D(x)\} + \mathbb{E}_{y=G(z), z \sim p_z} \{\log[1 - D(y)]\}$$

  - Train a discriminator D to label data positive, and generator's samples negative
  - Train a generator G to maximize likelihood of being classified data

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
URL: https://proceedings.neurips.cc/paper/5423-generative-adversarial-nets

Adversarial IRL

- Recall idea of GANs:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}\{\log D(x)\} + \mathbb{E}_{y=G(z),z\sim p_z}\{\log[1-D(y)]\}$$

  - Train a discriminator D to label data positive, and generator's samples negative
  - Train a generator G to maximize likelihood of being classified data

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets.
Advances in neural information processing systems, 27, 2014.
URL: https://proceedings.neurips.cc/paper/5423-generative-adversarial-nets

- The max margin idea is very similar:
  - Find a reward function that discriminates π^* optimal from all others
  - Find other policies π_i iteratively to discriminate against

Adversarial IRL

Learning Robust Rewards with Adversarial Inverse Reinforcement Learning

Justin Fu, Katie Luo, Sergey Levine
Department of Electrical Engineering and Computer Science
University of California, Berkeley
Berkeley, CA 94720, USA
justinjfu@eecs.berkeley.edu, katieluo@berkeley.edu,
svlevine@eecs.berkeley.edu

Abstract

Reinforcement learning provides a powerful and general framework for decision making and control, but its application in practice is often hindered by the need for extensive feature and reward engineering. Deep reinforcement learning methods can remove the need for explicit engineering of policy or value features, but still require a manually specified reward function. Inverse reinforcement learning holds the promise of automatic reward acquisition, but has proven exceptionally difficult to apply to large, high-dimensional problems with unknown dynamics. In this work, we propose AIRL, a practical and scalable inverse reinforcement learning algorithm based on an adversarial reward learning formulation. We demonstrate that AIRL is able to recover reward functions that are robust to changes in dynamics, enabling us to learn policies even under significant variation in the environment seen during training. Our experiments show that AIRL greatly outperforms prior methods in these transfer settings.

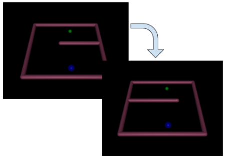


Figure 3: Illustration of the shifting maze task, where the agent (blue) must reach the goal (green). During training the agent must go around the wall on the left side, but during test time it must go around on the right.

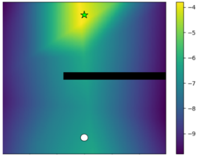


Figure 4: Reward learned on the point mass shifting maze task. The goal is located at the green star and the agent starts at the white circle. Note that there is little reward shaping, which enables the reward to transfer well.

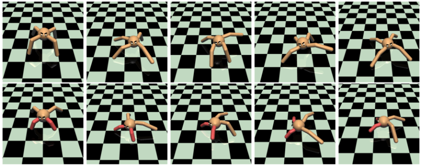


Figure 5: *Top row*: An ant running forwards (right in the picture) in the training environment. *Bottom row*: Behavior acquired by optimizing a state-only reward learned with AIRL on the disabled ant environment. Note that the ant must orient itself before crawling forward, which is a qualitatively different behavior from the optimal policy in the original environment, which runs sideways.

J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning, 2018-08-13.
URL: http://arxiv.org/abs/1710.11248, arXiv:1710.11248[cs]

Earlier similar work: [4]

Adversarial IRL

Algorithm 1 Adversarial inverse reinforcement learning

1: Obtain expert trajectories τ_i^E
2: Initialize policy π and discriminator $D_{\theta,\phi}$.
3: **for** step t in $\{1, \dots, \text{N}\}$ **do**
4: Collect trajectories $\tau_i = (s_0, a_0, ..., s_T, a_T)$ by executing π.
5: Train $D_{\theta,\phi}$ via binary logistic regression to classify expert data τ_i^E from samples τ_i.
6: Update reward $r_{\theta,\phi}(s, a, s') \leftarrow \log D_{\theta,\phi}(s, a, s') - \log(1 - D_{\theta,\phi}(s, a, s'))$
7: Update π with respect to $r_{\theta,\phi}$ using any policy optimization method.
8: **end for**

- The discriminator $D_{\theta,\phi}(s, a, s')$ operates on triplets and is parameterized as

$$D_{\theta,\phi}(s, a, s') = \frac{\exp\{f_{\theta,\phi}(s, a, s')\}}{\exp\{f_{\theta,\phi}(s, a, s')\} + \pi(a|s)}$$

$$\begin{aligned} f_{\theta,\phi}(s, a, s') &= g_\theta(s, a) + \gamma h_\phi(s') - h_\phi(s) \\ &\approx \underbrace{r(s, a) + \gamma V(s')}_{Q(s,a)} - V(s) = A(s, a) \end{aligned}$$

  - This particular decomposition is crucial!
  - Training this way $g_\theta(s, a)$ automatically gets “reward semantics”, and h_ϕ “value semantics”
  - $A(s, a)$ is called *advantage function*

Inverse RL Summary

- Conceptually highly interesting
- The max-margin/discrimination/adversarial idea is core to many approaches
  - Max entropy is alternative way of thinking

Outline

- Value Alignment
- Inverse RL
- **Preference-based RL**

Preference-based Learning

- In ML:
  - Given data of preference tuples $D = \{(x_1^i \succ x_2^i)\}_{i=1}^n$ (each tuple means a user preference)
  - learn a mapping $f : X \mapsto \mathbb{R}$ to minimize, e.g.

$$\sum_{i=1}^n [f(x_2^i) - f(x_1^i)]_+$$

Preference-based Learning

- In ML:
  - Given data of preference tuples $D = \{(x_1^i \succ x_2^i)\}_{i=1}^n$ (each tuple means a user preference)
  - learn a mapping $f : X \mapsto \mathbb{R}$ to minimize, e.g.

$$\sum_{i=1}^n [f(x_2^i) - f(x_1^i)]_+$$

  - Read about *label ranking, instance ranking, object ranking*

Preference-based RL

- Given *trajectory segment* data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n = \{\xi^i\}_{i=1}^n$ and *preferences* $\xi^i \succ \xi^j$ for some pairs (i,j), find a reward function s.t.

$$\xi^i \succ \xi^j \quad \Rightarrow \quad \sum_{t=1}^T R(s^i_t, a^i_t) > \sum_{t=1}^T R(s^j_t, a^j_t)$$

Preference-based RL

- Given *trajectory segment* data $D = \{(s^i_{1:T_i}, a^i_{1:T_i})\}_{i=1}^n = \{\xi^i\}_{i=1}^n$ and *preferences* $\xi^i \succ \xi^j$ for some pairs (i,j), find a reward function s.t.

$$\xi^i \succ \xi^j \quad \Rightarrow \quad \sum_{t=1}^T R(s^i_t, a^i_t) > \sum_{t=1}^T R(s^j_t, a^j_t)$$

- Long history, e.g.

R. Akrour, M. Schoenauer, and M. Sebag. APRIL: Active preference learning-based reinforcement learning.
In P. A. Flach, T. De Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524, pages 116–131. 2012.
URL: http://link.springer.com/10.1007/978-3-642-33486-3_8

Deep RL from Human Preferences

Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com

Jan Leike
DeepMind
leike@google.com

Tom B Brown
Google Brain*
tombbrown@google.com

Miljan Martic
DeepMind
miljanm@google.com

Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences.
Advances in neural information processing systems, 30, 2017.
URL: https://proceedings.neurips.cc/paper/7017-deep-reinforcement-learning-from-

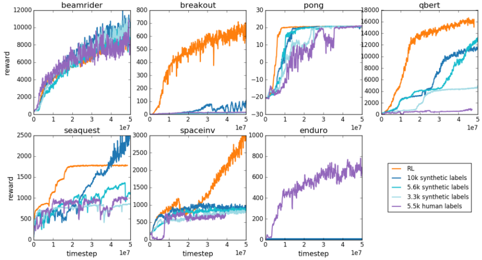


Figure 2: Results on Atari games as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 3 runs, except for the real human feedback which is a single run, and each point is the average reward over about 150,000 consecutive frames.

Deep RL from Human Preferences

- Iteratively update a policy π and reward function R_ψ:
  - Run RL algorithm to update π with R; collect episodes
  - Select segments ξ^i from these episodes; let a human specify preferences $\xi^i \succ \xi^j$
  - Update R to minimize "preference loss"
- Assume human preferences are noisy (Bradley-Terry model)

$$P(\xi^i \succ \xi^j; R) = \frac{\exp\{\sum_{t=1}^T R(s_t^i, a_t^i)\}}{\exp\{\sum_{t=1}^T R(s_t^i, a_t^i)\} + \exp\{\sum_{t=1}^T R(s_t^j, a_t^j)\}}$$

  - Maximize likelihood $\max_\psi \sum_{\xi^i \succ \xi^j} \log P(\xi^i \succ \xi^j; R_\psi)$ for all human provided preferences

Robotics Application

Few-Shot Preference Learning for Human-in-the-Loop RL

Joey Hejna
Stanford University
jhejna@cs.stanford.edu

Dorsa Sadigh
Stanford University
dorsa@cs.stanford.edu

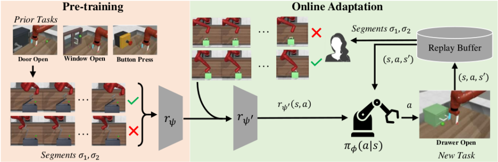


Figure 1: An overview of our method. **Pre-training (left):** In the pre-training phase we generate trajectory segment comparisons using data from a family of previously learned tasks and use them to train a reward model. **Online-Adaptation (Right)**: After pre-training the reward model, we adapt it to new data from human feedback use it to train a policy for a new task in a closed loop manner.

D. J. Hejna III and D. Sadigh. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning*, pages 2014–2025, 2023.
URL: https://proceedings.mlr.press/v205/iii23a.html

https://sites.google.com/view/few-shot-preference-rl/home

[1] P. Abbeel and A. Y. Ng.
Apprenticeship learning via inverse reinforcement learning.
In *Twenty-first international conference*, page 1, 2004.
URL: `http://portal.acm.org/citation.cfm?delete_doid=1015330.1015430`.

[2] R. Akrour, M. Schoenauer, and M. Sebag.
APRIL: Active preference learning-based reinforcement learning.
In P. A. Flach, T. De Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524, pages 116–131. 2012.
URL: `http://link.springer.com/10.1007/978-3-642-33486-3_8`.

[3] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei.
Deep reinforcement learning from human preferences.
Advances in neural information processing systems, 30, 2017.
URL: `https://proceedings.neurips.cc/paper/7017-deep-reinforcement-learning-from-`.

[4] C. Finn, S. Levine, and P. Abbeel.
Guided cost learning: Deep inverse optimal control via policy optimization.
In *International Conference on Machine Learning*, pages 49–58, 2016-06-11.
URL: `https://proceedings.mlr.press/v48/finn16.html`.

[5] J. Fu, K. Luo, and S. Levine.
Learning robust rewards with adversarial inverse reinforcement learning, 2018-08-13.
URL: `http://arxiv.org/abs/1710.11248`, `arXiv:1710.11248[cs]`.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative adversarial nets.
Advances in neural information processing systems, 27, 2014.
URL: `https://proceedings.neurips.cc/paper/5423-generative-adversarial-nets`.

[7] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan.
Cooperative inverse reinforcement learning.
Advances in neural information processing systems, 29, 2016.
URL:
https://proceedings.neurips.cc/paper_files/paper/2016/hash/c3395dd46c34fa7fd8d729d8cf88b7a8-Abstract.html.

[8] D. J. Hejna III and D. Sadigh.
Few-shot preference learning for human-in-the-loop rl.
In *Conference on Robot Learning*, pages 2014–2025, 2023.
URL: https://proceedings.mlr.press/v205/iii23a.html.

[9] S. Russell.
Human compatible: AI and the problem of control.
2019.
URL: https://books.google.com/books?hl=en&lr=&id=Gg-TDwAAQBAJ&oi=fnd&pg=PT8&dq=human+compatible+russell&ots=qoZKXK7gQ0&sig=p4x57HjxfMAVCpQ40_XcE7J4ECY.

[10] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey.
Maximum entropy inverse reinforcement learning.