Inverse Reinforcement Learning

CS4375 Artificial Intelligence Techniques

Luciano Cavalcante Siebert

Assistant Professor

Interactive Intelligence (II) Group, INSY, EEMCS, TU Delft

l.cavalcantesiebert@tudelft.nl

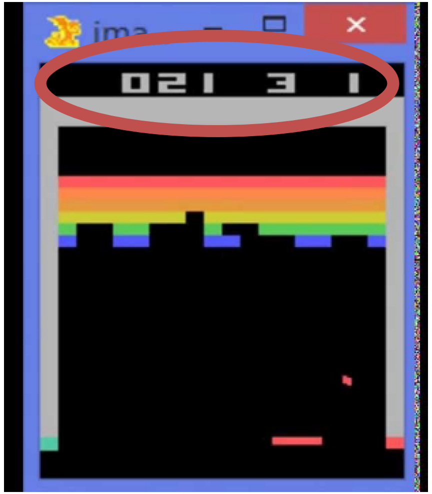
TU Delft

What we are going to talk about…

- Introduction to Imitation learning
- Inverse Reinforcement Learning (IRL)
  - Problem definition
  - Linear formulation and example
  - Other approaches: Max Entropy IRL, Adversarial IRL, Collaborative IRL
  - Limitations and critiques

Introduction

For computer games, the reward is usually quite clear:



However, in real world applications this is often not the case. Often a proxy is used as reward:

TU Delft

Imitation learning

- Instead of trying to learn from sparse rewards or manually specified reward function, an expert provides a set of demonstrations

- Agent then tries to learn from these demonstrations

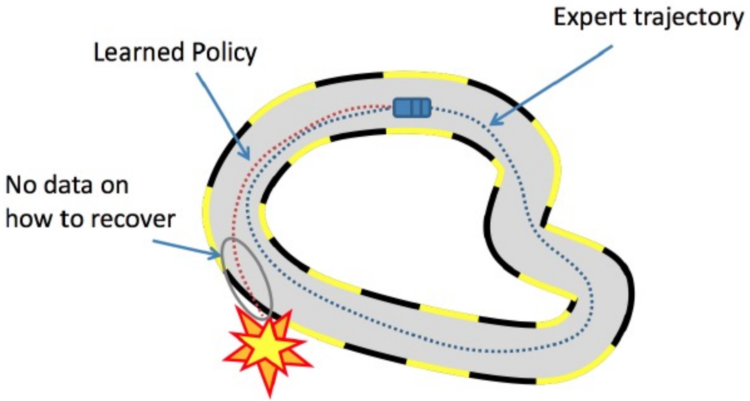
- Approaches to imitation learning include:
  - Behavioral cloning
  - Direct policy learning
  - **Inverse reinforcement learning (focus of today's lecture)**

Behavioral cloning

- Simple and efficient form of imitation learning
- Learn the expert policy directly with supervised learning

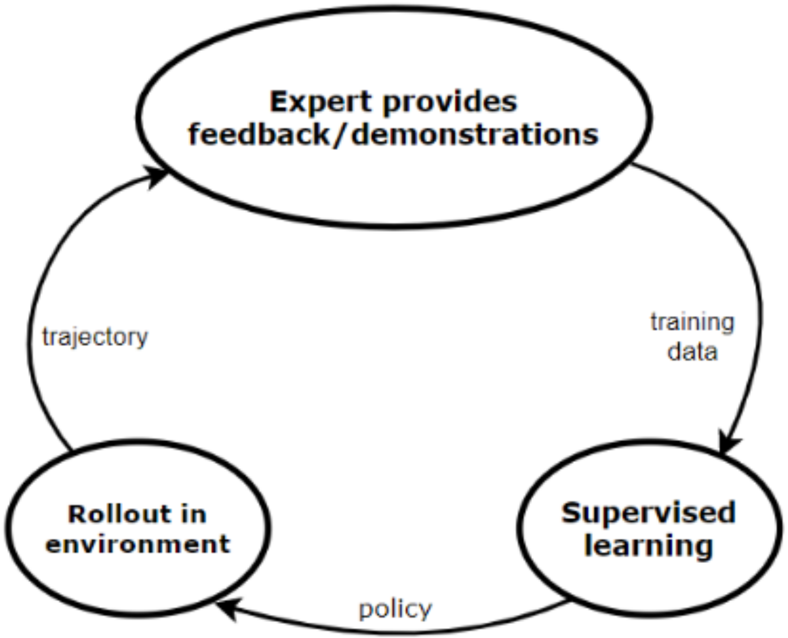
https://youtu.be/H0igiP6Hg1k?t=464

- Problem of “brittleness”: 1-step deviations can lead to huge errors
- Behavioral cloning will at best duplicate the expert’s performance, not exceed it



Source: http://web.stanford.edu/class/cs234/slides/lecture7.pdf

Direct policy learning (via interactive demonstrator)



- Does not suffer from the same problems as behavioural cloning
- But requires an interactive demonstrator/expert at all times

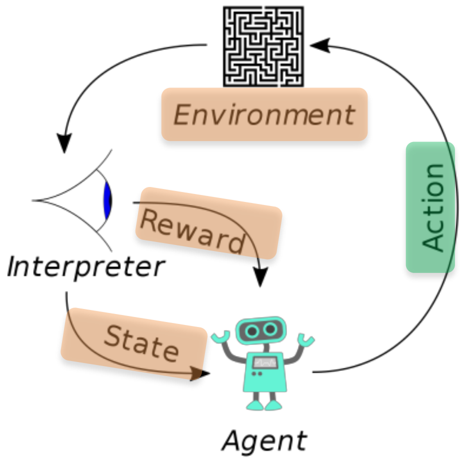
- Still not clear *why* the agent should perform any given action

TU Delft

Inverse Reinforcement Learning (IRL)

- What if, instead of learning the policy, we learn the reward?

IRL: An informal definition



RL

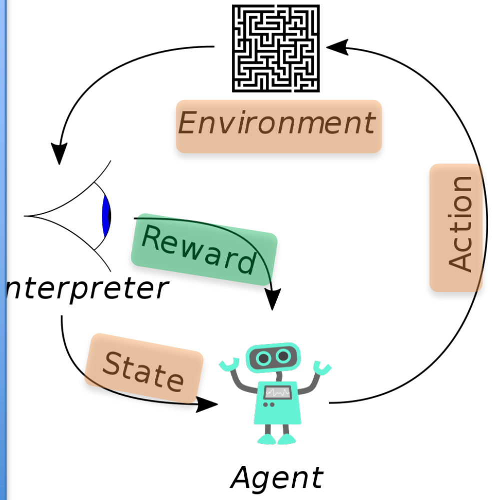
Given: Reward

Find: Optimal policy

IRL

Given: Demonstrations (or optimal policy)

Find: Reward



TU Delft

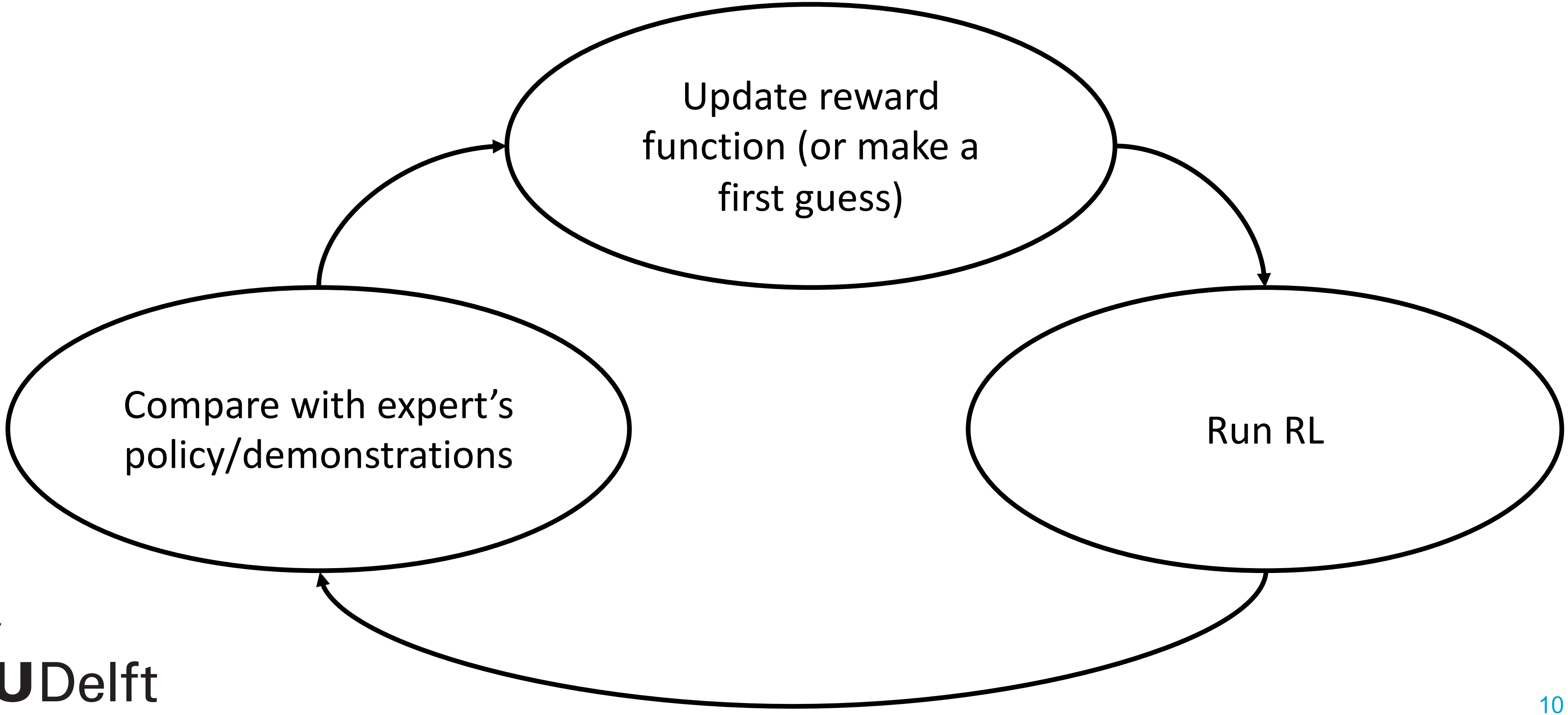
	Direct Policy Learning	**Reward Learning**	**Access to Environment**	**Interactive Demonstrator**	**Pre-collected demonstrations**
Behavioral cloning (BC)	Yes	No	No	No	Yes
Direct policy learning (interactive IL)	Yes	No	Yes	Yes	Optional
Inverse Reinforcement Learning (IRL)	**No**	**Yes**	**Yes**	**No**	**Yes**
Preference-based RL	No	Yes	Yes	Yes	No

Source: Adapted from ICML 2018 Imitational Learning Tutorial. Yisong Yue, Hoang M. Le
https://drive.google.com/file/d/12QdNmMlI-bGISWnm8pmD_TawuRN7xagX/viewv
https://sites.google.com/view/icml2018-imitation-learning/

* For more information about Preference-based RL, more specifically about the technique reinforcement learning from human feedback RLFH, see: https://arxiv.org/pdf/1706.03741.pdf

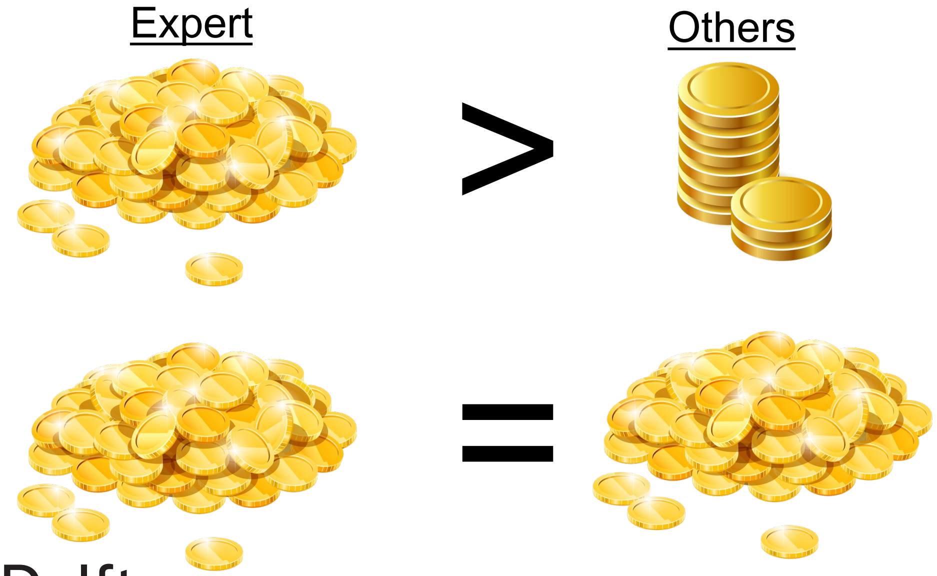
TUDelft

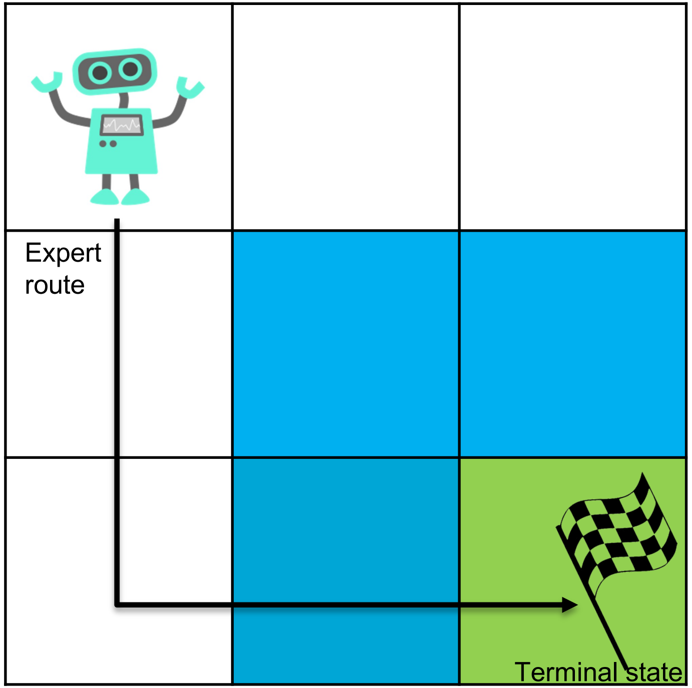
IRL: An informal definition



IRL Gridworld example

Let's assume that: "Experts" achieve identical or higher rewards than others





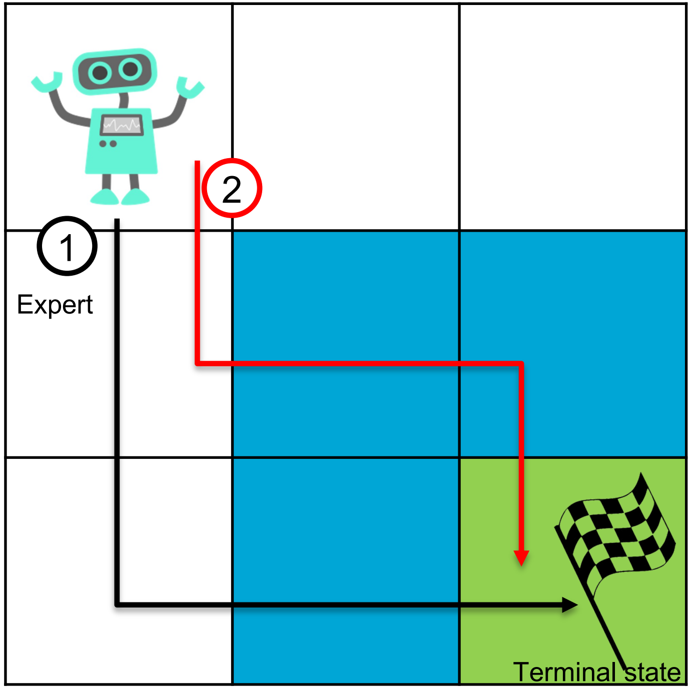
Gridworld example

First guess:
- White = 0
- Blue = 1
- Green = 3

Route 1: 0 + 0 + 1 + 3 = 4
Route 2: 0 + 1 + 1 + 3 = 5

Second guess:
- White = 0
- Blue = -1
- Green = 2

Route 1: 0 + 0 - 1 + 2 = 1
Route 2: 0 - 1 - 1 + 2 = 0



TUDelft

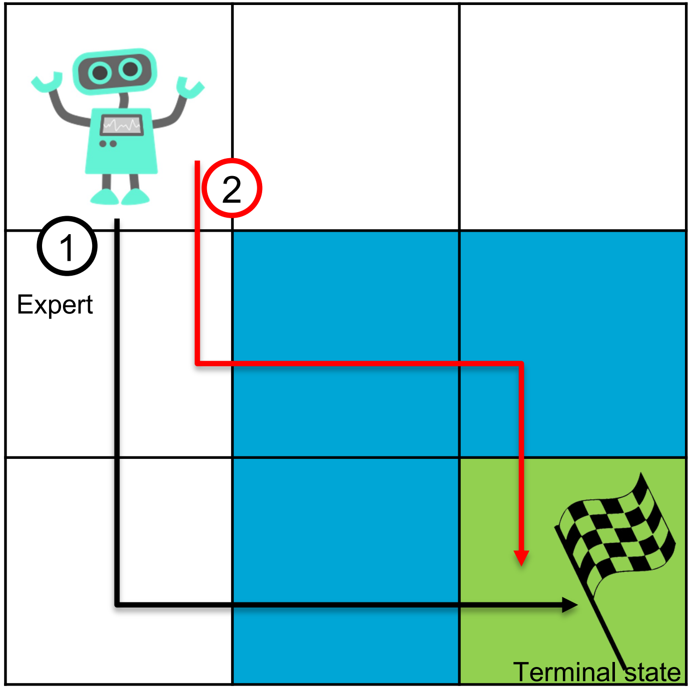
Gridworld example

Third guess:
- White = 0
- Blue = 0
- Green = 1

Route 1: 0 + 0 + 0 + 1 = 1
Route 2: 0 + 0 + 0 + 1 = 1

Fourth guess:
- White = 0
- Blue = 0
- Green = 0

Route 1: 0 + 0 + 0 +0 = 0
Route 2: 0 + 0 + 0 +0 = 0



IRL

- Goal:
  - Find R where π provided by the expert is optimal (let us start with that.. more advanced methods consider demonstrations instead of the policy π)

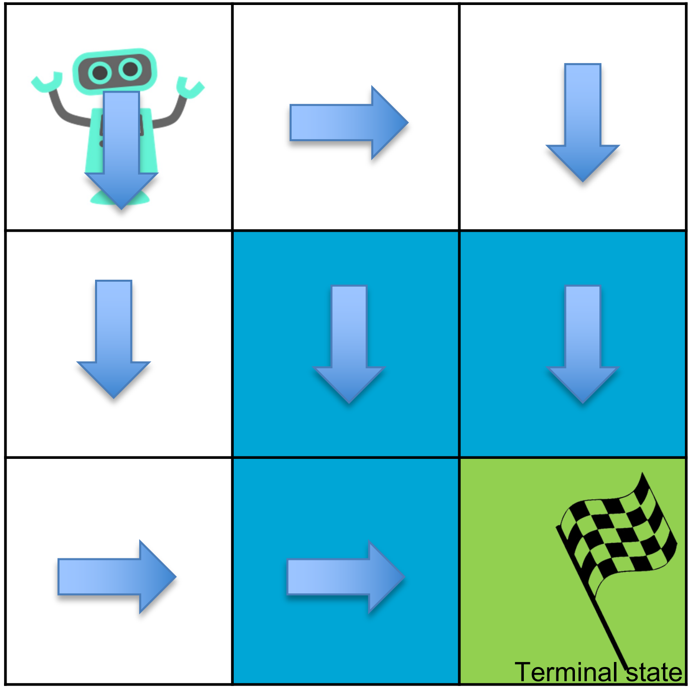
- But, this is an underdetermined problem. We need some heuristics:
  - Prefer solutions where the expert policy has the largest difference to the other ones

$$max(value^{*}\ - value^{2nd\ best})$$

  - Prefer solutions with smaller rewards

$$min\,Reward$$

$$max(-Reward)$$



TUDelft

Formalizing

- Bellman equation: $$V^{\pi}(s) = R(s) + \gamma \sum P_{s\pi(s)}(s')V^{\pi}(s')$$
- Given that $\pi(s) \equiv a$
- We can rewrite the equation above as:

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

$$\boldsymbol{V}^{\pi} - \gamma \boldsymbol{P}_{a^*}\boldsymbol{V}^{\pi} = \boldsymbol{R}$$

$$\boldsymbol{V}^{\pi}(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*}) = \boldsymbol{R}$$

$$\boxed{\boldsymbol{V}^{\pi} = (\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1}\boldsymbol{R}}$$

Where:
$\boldsymbol{P}_{a^*}$ is the transition probability matrix ,$N \times N$
$\boldsymbol{V}^{\pi}$ and $\boldsymbol{R}$ (reward) are N x 1 vectors
γ is the discount factor

Note: In the previous lecture *Ta* was used for the probability transition matrix. Other notations can also be slightly different. In this this and the following slides I use the notation from Ng and Russel (2000)

TU Delft

A. Y. Ng and S. J. Russell. 2000. Algorithms for inverse reinforcement learning. In: *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, Stanford University, Stanford, CA, USA.
https://ai.stanford.edu/~ang/papers/icml00-irl.pdf

Formalizing

- Now let's formalize our assumption that π^* achieves identical or higher expected value then all other policies:

$$\boldsymbol{P}_{a^*}\boldsymbol{V}^{\pi} \succcurlyeq \boldsymbol{P}_a\boldsymbol{V}^{\pi}, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*$$

$$\boldsymbol{P}_{a^*}\boxed{\boldsymbol{V}^{\pi}} - \boldsymbol{P}_a\boxed{\boldsymbol{V}^{\pi}} \succcurlyeq 0, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*$$

$$\boxed{\boldsymbol{P}_{a^*}}(\boldsymbol{I} - \gamma\boldsymbol{P}_{a^*})^{-1}\boldsymbol{R} - \boxed{\boldsymbol{P}_a}(\boldsymbol{I} - \gamma\boldsymbol{P}_{a^*})^{-1}\boldsymbol{R} \succcurlyeq 0, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*$$

$$\boxed{(\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma\boldsymbol{P}_{a^*})^{-1}\boldsymbol{R} \succcurlyeq 0, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*}$$

Formalizing

Heuristics:

- Prefer solutions where the expert policy performs better than the other ones
  - Maximize the gap of expected value of acting optimally and the best expected value acting suboptimally

$$maximize \sum_{i=1}^{N} min_{a \in \boldsymbol{A} \setminus \boldsymbol{a}^*} (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R}$$

- Prefer solutions with smaller rewards
  - Add a penalty term

$$maximize \sum_{i=1}^{N} min_{a \in \boldsymbol{A} \setminus \boldsymbol{a}^*} \{(\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R}\} \boxed{- \boldsymbol{\lambda} || \boldsymbol{R} ||_{\mathbf{1}}}$$

TU Delft

Formal definition

Linear programming formulation:

$$maximize \sum_{i=1}^{N} min_{a \in \boldsymbol{A} \setminus \, \boldsymbol{a}^*} (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R} \; - \; \boldsymbol{\lambda} \|\boldsymbol{R}\|_{\boldsymbol{1}}$$

$$s.t. \,(\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R} \succcurlyeq 0, \forall a \in \boldsymbol{A} \; \setminus \; \boldsymbol{a}^*$$

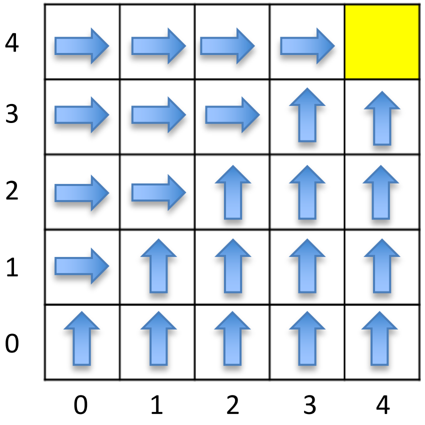
$$|R_i| \leq \; R_{max}, i = 1, \ldots, N$$

A practical example

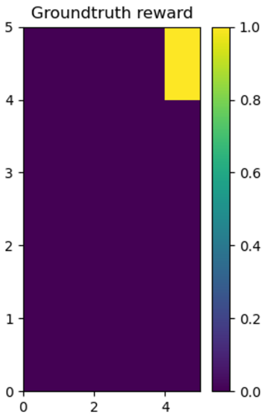
$$\mathit{maximize} \sum_{i=1}^{N} \mathit{min}_{a \in \boldsymbol{A} \setminus \boldsymbol{a}^*} (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R} \; - \; \lambda \|\boldsymbol{R}\|_1$$

$$s.t.\ (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1} \boldsymbol{R} \succcurlyeq 0, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*$$
$$|R_i| \leq R_{max}, i = 1, \ldots, N$$

5x5 gridworld environment, where π^*:



Actions: (↑, →, ↓, ←)



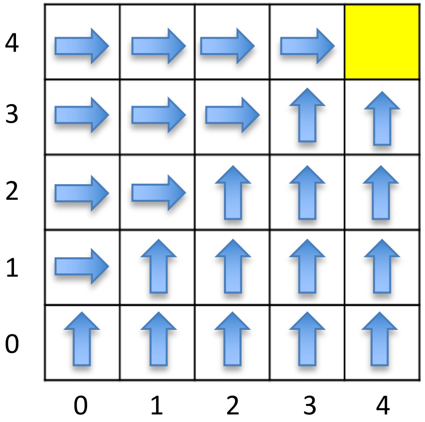
- Agents start from the lower-left grid square (0,0), and finish on the upper-right grid square (4,4)
- Agents move (up, down, left, right) just one square at a time
- Actions have a 30% chance of moving in a random direction (wind)
- Discount factor $\gamma = 0.2$
- Penalty factor $\lambda = 1.05$
- $R_{max} = 1$

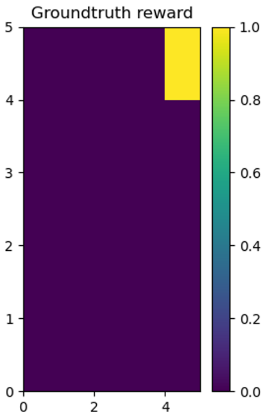
A practical example

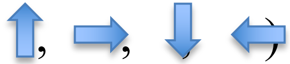
$$\text{maximize} \sum_{i=1}^{N} \min_{a\in \boldsymbol{A}\setminus \boldsymbol{a}^*} (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1}\boldsymbol{R} \;-\; \lambda \|\boldsymbol{R}\|_1$$

$$s.t.\ (\boldsymbol{P}_{a^*} - \boldsymbol{P}_a)\,(\boldsymbol{I} - \gamma \boldsymbol{P}_{a^*})^{-1}\boldsymbol{R} \succcurlyeq 0, \forall a \in \boldsymbol{A} \setminus \boldsymbol{a}^*$$
$$|R_i| \leq R_{max}, i = 1, \ldots, N$$

5x5 gridworld environment, where π^*:





Actions: ()

- Step 1: Calculate $\boldsymbol{P}_a$

$\boldsymbol{P}_{a=\uparrow}(0,0)(0,0) = 0$

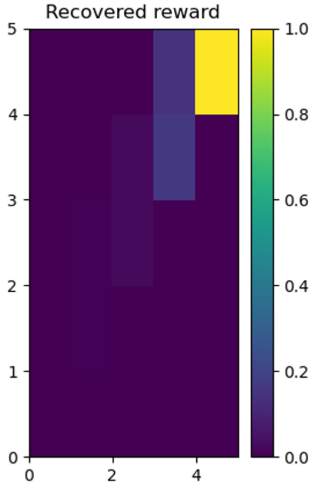
$\boldsymbol{P}_{a=\uparrow}(0,0)(0,1) = 1 - 0.3 + 0.3/4$

$\boldsymbol{P}_{a=\uparrow}(0,0)(1,0) = 0.3/4$

$\boldsymbol{P}_{a=\uparrow}(0,0)(0,2) = 0$

….

- Step 2: Run the linear IRL algorithm



Exercise sheet on Brightspace:

A working example of IRL is provided in the tutorial sheet

Explore the code and answer the questions

TUDelft

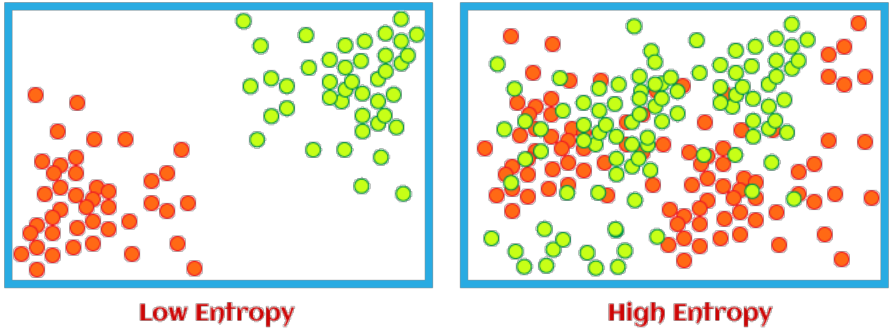
- This linear formulation requires an optimal policy, but we usually all we have access is a set of trajectories (demonstrations)

- Can we do better than the heuristics we used?

Maximum Entropy IRL

- Handle ambiguity using a probabilistic model of behavior
- Employs the *principle of maximum entropy* to resolve the ambiguity in choosing a distribution over decisions in a principled way

Principle of maximum entropy: the probability distribution which best represents the current state of knowledge about a system is the one with largest entropy.



Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In Aaai (Vol. 8, pp. 1433-1438). https://www.aaai.org/Papers/AAAI/2008/AAAI08-227.pdf

Maximum Entropy IRL

- Notion of suboptomality
  - If the demonstrator has some random behavior at times, this might mean that they don’t care much about specific actions in this setting → Lower/No reward
  - If the demonstrator consistently does a specific action in a given setting, it probably means that they really care about it → Larger reward

- The idea is to match the most relevant features, but besides that be as random as possible

Maximum Entropy IRL

- A bit more formal.. We want to learn a reward function that yields the distribution over all trajectories with the maximum entropy:

$$\max_{D} \; - \sum_{\pi \in (S \times A)} Pr(\pi) \log Pr(\pi)$$

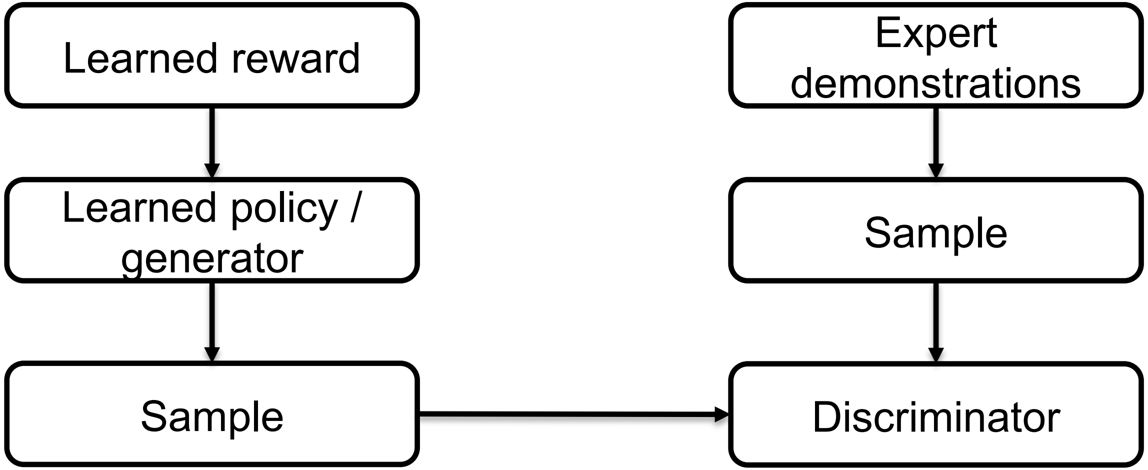
- While respecting two constraints:
  - The distribution over all trajectories should be a probability distribution
  - The expected feature count of the demonstrated trajectories must match the empirical feature

TUDelft

- Maximum Entropy IRL deals with the ambiguity problem in a more principled manner
- But, still it only consider linear reward functions
- Other approaches can deal with non-linear reward functions, for example…

Adversarial IRL

- Trains a policy against a discriminator that aims to distinguish the expert demonstrations from the learned policy
- Adversarial IRL can deal with non-linear rewards



Fu, Justin, Katie Luo, and Sergey Levine. "Learning robust rewards with adversarial inverse reinforcement learning." arXiv preprint arXiv:1710.11248 (2017).

- Finally, this process could be done in a more collaborative manner
- Humans teach each other all the time… and someone that is teaching something will not necessarily behave “optimally”, but will try to modify the behavior to support learning

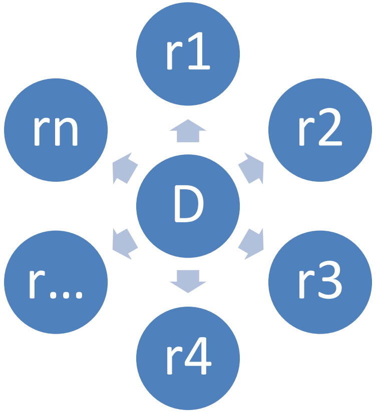
Collaborative IRL

- Cooperative, partial information game with two agents, human and robot
  - Human knows the reward function
  - Robot does not know the reward function

- The human and the robot get rewards determined by the same reward function → incentivizes the human to teach and the robot to learn

Hadfield-Menell, D., Russell, S. J., Abbeel, P., & Dragan, A. (2016). Cooperative inverse reinforcement learning. Advances in neural information processing systems, 29.

Critiques and limitations



- Underdetermined problem (ambiguity)
  - Wrong guesses can lead to high regret

- Difficult to evaluate a learned reward → we do not have access to a “ground truth reward”

- Demonstrations may not be optimal

TU Delft

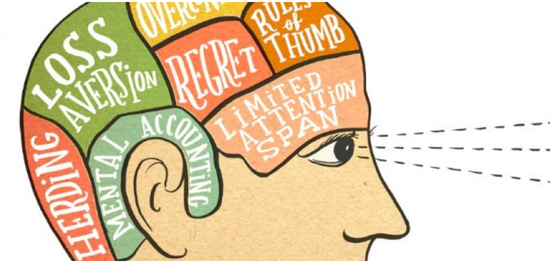
Critiques and limitations

- IRL assumes that human behavior is optimal or noisily optimal

- However, humans often deviate from such rationality assumptions:
  - In systematic, non-random ways: Biases such as time inconsistency, loss aversion and anchoring
  - But also due to cognitive aspects such as forgetfulness, limited planning and false beliefs

Critiques and limitations

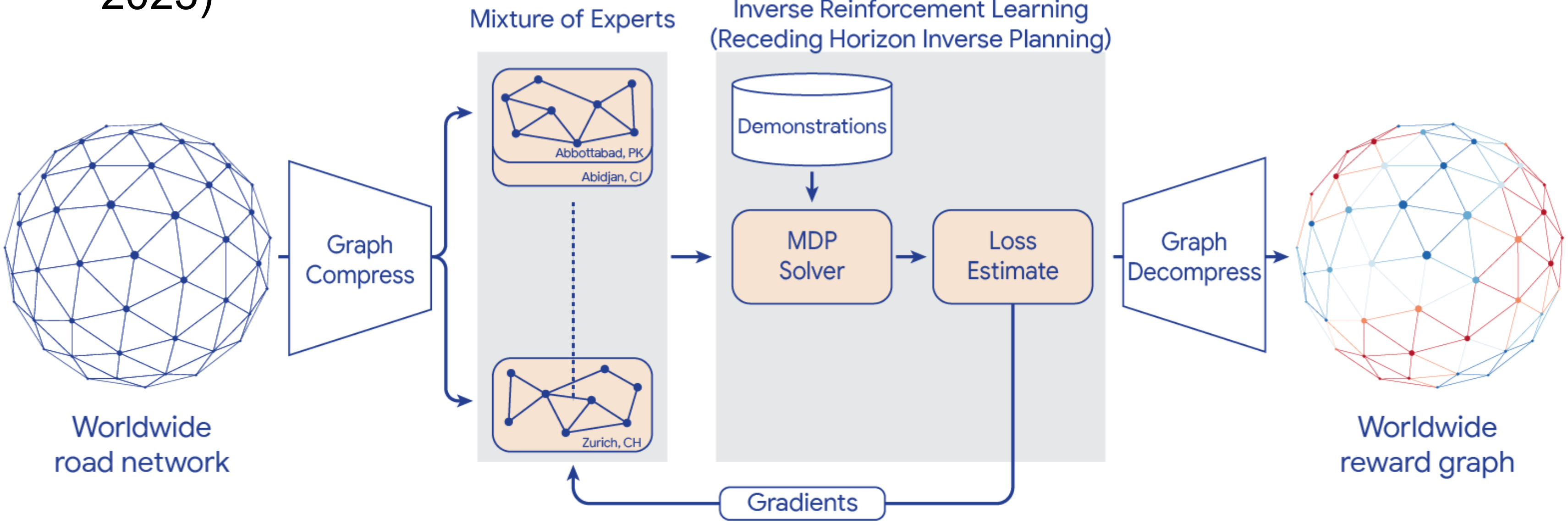
- Methods have been proposed to account for specific human deviations from such rational expectations e.g. specific biases or noise rationality, but no general framework has been proposed
- Mindermann and Armstrong (2018) demonstrated that:
  - It is impossible to uniquely decompose a policy (or demonstrations) into a planning algorithm (i.e. human thinking/”rationality”) and a reward function
  - Normative assumptions are needed, which cannot be deduced exclusively from observations



Armstrong, S., & Mindermann, S. (2018). Occam's razor is insufficient to infer the preferences of irrational agents. Advances in neural information processing systems, 31.

A recent real-world application…

- World scale inverse reinforcement learning in Google Maps (September 12, 2023)

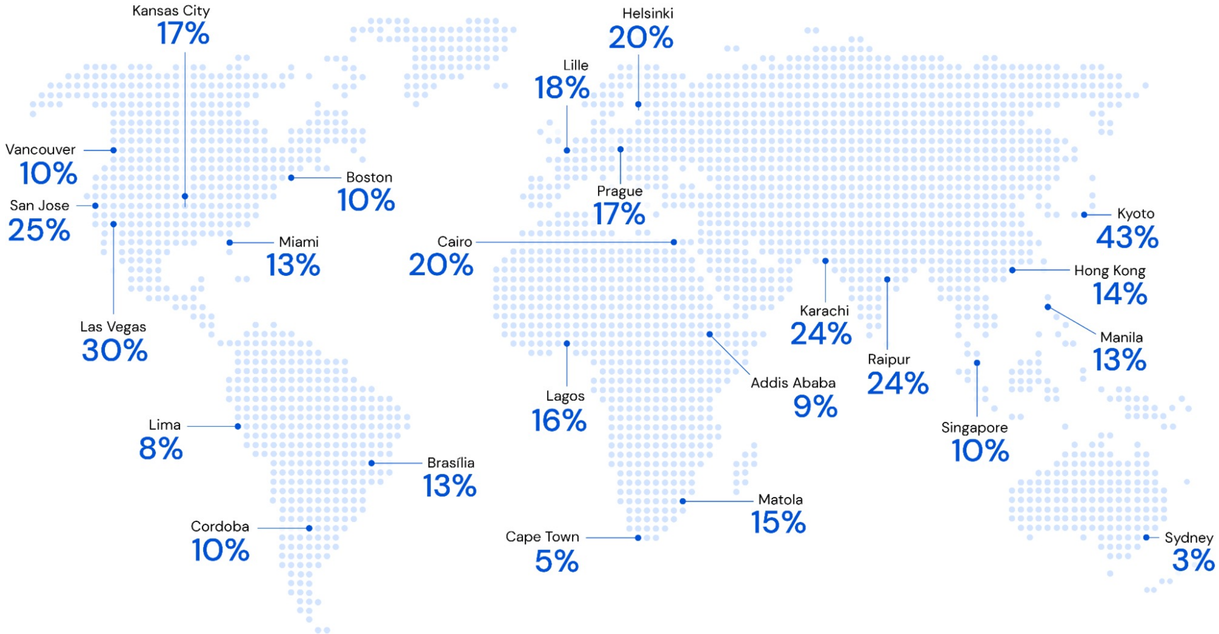


TUDelft

https://blog.research.google/2023/09/world-scale-inverse-reinforcement.html
https://arxiv.org/pdf/2305.11290.pdf

A recent real-world application…

- World scale inverse reinforcement learning in Google Maps (September 12, 2023)



TU Delft

https://blog.research.google/2023/09/world-scale-inverse-reinforcement.html
https://arxiv.org/pdf/2305.11290.pdf

Wrapping up

- IRL methods can learn a reward function from human demonstrations

- However, especially in social complex environments, it is important to take very well into consideration that such algorithms might not model all nuances and particularities of human behavior -> Model ≠ Reality

TUDelft

References

IRL:

- A. Y. Ng and S. J. Russell. 2000. Algorithms for inverse reinforcement learning. In: *Proceedings of the 17th International Conference on Machine Learning (ICML ’00)*, Stanford University, Stanford, CA, USA. https://ai.stanford.edu/~ang/papers/icml00-irl.pdf
- Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In Aaai (Vol. 8, pp. 1433-1438).
- Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. Artificial Intelligence, 297, 103500.

RL text book:

- R. S. Sutton and A. G. Barto. 2018. Reinforcement learning: An introduction (2nd ed). MIT press. https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf

Text book on linear programming:

- R. J. Vanderbei. 2020. Linear programming: foundations and extensions (5th ed). Springer Nature.

Inverse Reinforcement Learning

CS4375 Artificial Intelligence Techniques

Luciano Cavalcante Siebert

Assistant Professor

Interactive Intelligence (II) Group, INSY, EEMCS, TU Delft

l.cavalcantesiebert@tudelft.nl

TU Delft