Feature Construction for Inverse Reinforcement Learning

Sergey Levine
Stanford University

Zoran Popović
University of Washington

Vladlen Koltun
Stanford University

1. Introduction

Goal: given Markov Decision Process (MDP) $\mathcal{M}$ without its reward function R, as well as example traces $\mathcal{D}$ from its optimal policy, find R.

Motivations: learning policies from examples, inferring goals, specifying tasks by demonstration.

Challenge: many functions R fit the examples, but many will not generalize to unobserved states. Selecting compact set of features that represent R is difficult.

Solution: construct features to represent R from exhaustive list of *component features*, using **logical conjunctions** of component features represented as a **regression tree**.

2. Background

Markov Decision Process: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \theta, \gamma, R\}$

$\mathcal{S}$ – set of states
$\mathcal{A}$ – set of actions
γ – discount factor
R – reward function
θ – state transition probabilities: $\theta_{sas'} = P(s'|s,a)$

Optimal Policy: denoted π^*, maximizes $E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | \pi^*, \theta\right]$

Example Traces: $\mathcal{D} = \{(s_{1,1}, a_{1,1}), ..., (s_{n,T}, a_{n,T})\}$, where $s_{i,t}$ is the t^{th} state in the i^{th} trace, and $a_{i,t}$ is the optimal action in $s_{i,t}$.

Previous Work: most existing algorithms require a set of features Φ to be provided, and find a reward function that is a linear combination of the features [1, 2, 3, 4]. Finding features that are relevant and sufficient is difficult. Furthermore, a linear combination is not always a good estimate for the reward.

Component Features: instead of a complete set of relevant features, our method accepts an exhaustive list of component features $\delta : \mathcal{S} \rightarrow \mathbb{Z}$. The algorithm finds a regression tree, with relevant component features acting as tests, to represent the reward.

3. Algorithm

Overview: Iteratively construct feature set Φ and reward R, alternating between an **optimization phase** that determines a reward, and a **fitting phase** that determines the features.

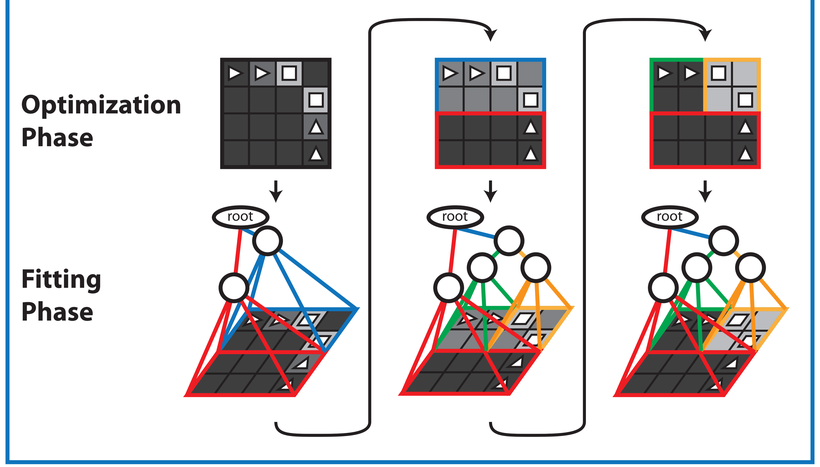
Optimization Phase: Find reward R "close" to current features Φ, under which examples $\mathcal{D}$ are part of the optimal policy. Letting $Proj_\Phi R$ denote the closest reward to R that is a linear combination of features Φ, we find R as:

$$\min_R \|R - Proj_\Phi R\|^2$$
$$\text{s.t.} \quad \pi^R(s) = a \qquad \forall (s,a) \in \mathcal{D}$$

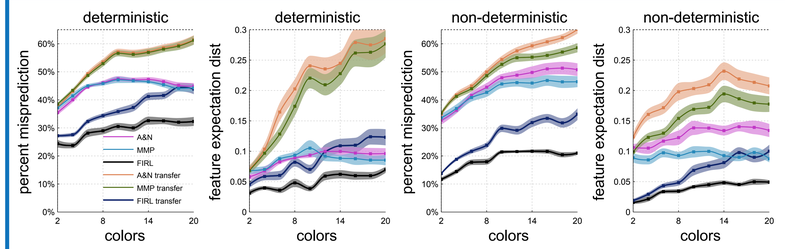
Note that R can "step outside" of the current features to satisfy the examples, if the current features Φ are insufficient.

Fitting Phase: Fit a **regression tree** to R, with component features δ acting as *tests* at tree nodes. Indicators for leaves of the tree are the new features Φ. Only component features that are **relevant** to the structure of R are selected, and leaves correspond to their **logical conjunctions**.

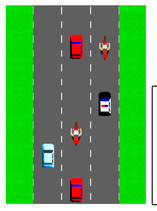
4. Illustrated Example



5. Experimental Results



Gridworld transfer comparison: 64×64 gridworld with colored objects placed at random. Component features give distance to object of specific color. Many colors are irrelevant. Transfer performance corresponds to learning reward on one random gridworld, and evaluating on 10 others (with random object placement). Comparing FIRL (proposed algorithm), Abbeel & Ng [1], MMP [3], LPAL [4]. FIRL outperforms prior methods, which cannot distinguish relevant objects from irrelevant ones.



	"Lawful" policies			"Outlaw" policies		
	percent mispred-iction	feature expect. distance	average speed	percent mispred-iction	feature expect. distance	average speed
Expert	0.0%	0.000	**2.410**	0.0%	0.000	**2.375**
FIRL	22.9%	0.025	**2.314**	24.2%	0.027	**2.376**
MMP	27.0%	0.111	1.068	27.2%	0.096	1.056
A&N	38.6%	0.202	1.054	39.3%	0.164	1.055
Random	42.7%	0.220	1.053	41.4%	0.184	1.053

Highway driving: "lawful" policy avoids going fast in right lane, "outlaw" policy drives fast, but slows down near police. Features indicate presence of police, current lane, speed, distance to cars, etc. Logical connection between speed and lanes/police cars cannot be captured by linear combinations, and prior methods cannot match the expert's speed while also matching feature expectations. Videos of the learned policies can be found at: `http://graphics.stanford.edu/projects/firl/index.htm`.

6. References

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML '04: Proceedings of the 21st International Conference on Machine Learning*. ACM, 2004.

[2] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *ICML '00: Proceedings of the 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000.

[3] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pages 729–736. ACM, 2006.

[4] U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 1032–1039. ACM, 2008.