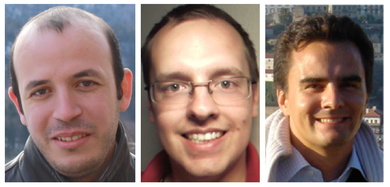
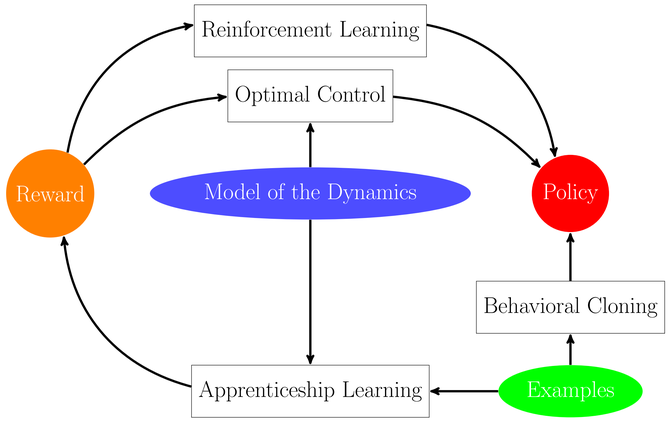
Structured Apprenticeship Learning

Abdeslam Boularias[1] Oliver Krömer[2] Jan Peters[1,2]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Darmstadt University of Technology, Darmstadt, Germany



1 Overview



- Problem: apprenticeship learning with **noisy** or **missing reward features**. This problem often occurs when the features are obtained through **sensors**.
- Proposed solution:
  1. define a **similarity measure** in the state-action space (domain knowledge),
  2. construct a k-nearest neighbors graph of states and actions,
  3. learn a distribution on policies such that the probability of a policy is proportional to its value and to the number of **adjacent states** that have **the same action**.

2 Background

2.1 Markov Decision Process (MDP)

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, T, R)$, where

- $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions,
- T are transition probabilities with $T(s,a,s') = P(s'|s,a)$ for $s, s' \in \mathcal{S}, a \in A$,
- R is a reward function where $R(s,a)$ is the reward given for action a in state s.

2.2 Policies and Value Functions

- A policy π is a function that maps every state into an action.
- The value of policy π is given by $V(\pi) = \mathbb{E}[\sum_{t=0}^{H} \gamma^t R(s_t, a_t)|\mu_0, \pi, T]$, where μ_0 is an initial state distribution and H is a horizon.

2.3 Apprenticeship Learning

- Designing a reward function for matching a complex behavior can be a challenging problem. It is often easier to provide examples of the desired behavior [1].
- Apprenticeship Learning via Inverse Reinforcement Learning (IRL) consists in learning a reward function that explains an observed behavior.
- The reward is usually assumed to be a linear function of state-action features ϕ_i,

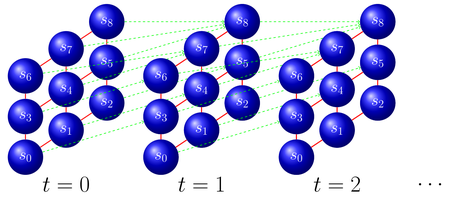
$$R(s,a) = \sum_i \theta_i \phi_i(s,a).$$

- The learned reward function, parameterized by θ, is used to find a policy that generalizes the observed behavior.

3 Structured Apprenticeship Learning

3.1 Key Insight

- In practice, it is often difficult to find appropriate reward features. These features are usually obtained from empirical data and are subject to measurement errors.
- Most real-world problems have a certain structure wherein states that are close to each other tend to have the same optimal action.
- We create a graph that connects similar states together. This graph can be constructed by using the Euclidean distance, for instance, as a similarity measure.



Markov Decision Process with a Similarity Graph

- We denote the edges of this graph by $\mathcal{E}$ and the edge features by ψ.

3.2 Problem Statement

Structured Apprenticeship Learning is formulated as the problem of maximizing the entropy of a distribution on policies P,

$$\max_{P,\mu^\pi} \Big(- \sum_{\pi \in \mathcal{A}^{|\mathcal{S}|}} P(\pi) \log P(\pi) \Big), \tag{1}$$

subject to the following constraints

$$\sum_{\pi \in \mathcal{A}^{|\mathcal{S}|}} P(\pi) = 1,$$
$$\forall \phi_k : \sum_{\pi \in \mathcal{A}^{|\mathcal{S}|}} P(\pi) \sum_{s \in \mathcal{S}} \mu^\pi(s) \phi_k(s, \pi(s)) = \hat{\phi}_k,$$
$$\forall \psi_k : \sum_{(s_i,s_j) \in \mathcal{E}} \psi_k(s_i, s_j) \sum_{\pi, \pi(s_i) = \pi(s_j)} P(\pi) = \hat{\psi}_k.$$

- $\hat{\phi}$ and $\hat{\psi}$ are the empirical averages of the reward and graph features respectively, calculated from the demonstration, and $\mu^\pi(s) = \mu_0(s) + \gamma \sum_{s'} T(s, \pi(s), s') \mu^\pi(s')$.

3.3 Solution

$$P(\pi) \propto \exp\Big(\underbrace{\sum_s \mu^\pi(s) \sum_k \theta_k \phi_k(s, \pi(s))}_{\text{usual value function}} + \overbrace{\sum_{\substack{(s_i,s_j) \in \mathcal{E} \\ \text{s.t. } \pi(s_i) = \pi(s_j)}} \sum_k \lambda_k \psi_k(s_i, s_j)}^{\text{structure reward}} \Big).$$

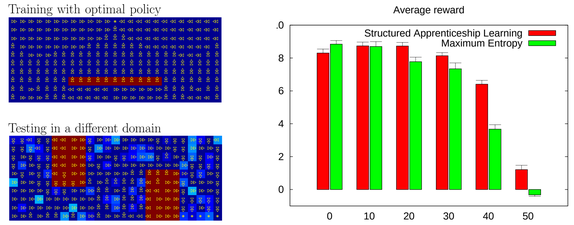
- Parameters θ and λ are learned by maximizing the likelihood of the demonstration.
- An optimal policy $\pi^* = \arg\max_{\pi \in \mathcal{A}^{|\mathcal{S}|}} P(\pi)$ is found by dynamic programing.

3.4 Relation to other methods

	$\lvert\mathcal{E}\rvert = 0$	$\lvert\mathcal{E}\rvert \neq 0$
$\gamma = 0$	Logistic regression	Associative Markov Networks (AMN) [4]
$\gamma \neq 0$	Maximum Entropy IRL [2]	Structured Apprenticeship Learning

4 Experiments

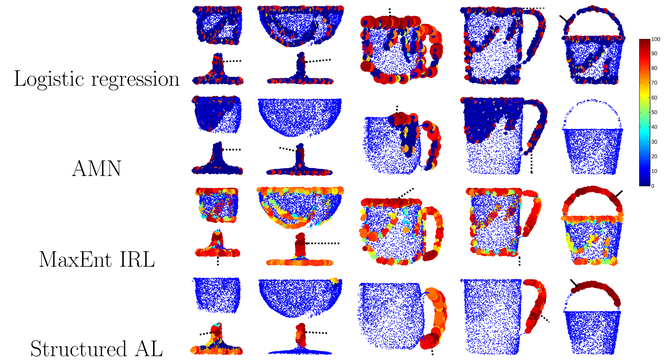
4.1 Gridworlds



Blue indicates a low value of a feature associated with negative reward, and red indicates a higher value of that feature. In the testing domain, a noise is added to the negative-weighted feature of randomly chosen states. The figure shows the average true rewards of MaxEnt and SAL as a function of the percentage of noisy states.

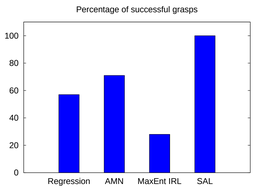
4.2 Grasping New Objects

From a high-level point of view, grasping an object can be seen as an MDP with three time-steps: reaching the object, preshaping the hand, and grasping.



Learned Q-values at $t = 0$. Each point on an object corresponds to a reaching action. Blue indicates low values and red indicates high values. The black arrow indicates the approach direction in the optimal policy according to the learned reward function.

Notice how Structured Apprenticeship Learning improves over the other methods by generally giving high values to handle points only. The confusion in the other methods comes from noise features and self-occlusions of the 3D point clouds.



References

[1] Pieter Abbeel and Andrew Ng. Apprenticeship Learning via Inverse Reinforcement Learning. ICML 2004.
[2] Ziebart, B., Maas, A., Bagnell, A., and Dey, A. Maximum Entropy Inverse Reinforcement Learning. AAAI 2008.
[3] Nathan Ratliff, J. Andrew Bagnell and Martin Zinkevich. Maximum Margin Planning. ICML 2006.
[4] Taskar, B.: Learning Structured Prediction Models: A Large Margin Approach. PhD thesis, Stanford University, 2004.

EWRL 2012, Edinburgh, June 2012