Policy Transfer in Apprenticeship Learning

Abdeslam Boularias and Brahim Chaib-draa

Laval University, Canada

{boularias;chaib}@damas.ift.ulaval.ca

DAMAS Décision, Adaptation, Multi-AgentS www.damas.ift.ulaval.ca

UNIVERSITÉ LAVAL

1. Overview

- We cast the problem of Apprenticeship Learning (Imitation Learning) as a classification problem.
- We use a modified version of the k-nearest neighbors method.
- The distance between two vertices is the distance between the graphs defined around these vertices.
- The distance between two graphs is the largest error of a homomorphism between the two graphs.

2. Markov Decision Process (MDP)

A Markov Decision Process (MDP) is defined by:

- $\mathcal{S}$: a finite set of states.
- $\mathcal{A}$: a finite set of actions.
- T: a transition function, where $T(s,a,s')$ is the probability of ending up in state s' after taking action a in state s.
- R: a reward function, $R(s,a)$ is the immediate reward that the agent receives for executing action a in state s.
- $\gamma \in (0,1]$ is a discount factor.

3. Policies

- A policy π is a function that maps every state into a distribution over the actions:

$$\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$$
$$\pi(s,a) = Pr(a_t = a | s_t = s)$$

- The value of a policy π is the expected sum of the rewards that an agent receives by following this policy.

$$V(\pi) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | \pi]$$

- Solving an MDP consists in finding an optimal policy.

4. Apprenticeship Learning

- Specifying a reward function by hand is not easy in most of the practical problems [Abbeel & Ng, 2004].
- It is often easier to demonstrate examples of a desired behavior than to define a reward function.
- In apprenticeship learning, we assume that the reward function is unknown.

There are two parts involved in apprenticeship learning:

1. An expert agent demonstrating an optimal policy π^E for some states.
2. A apprentice agent trying to learn a generalized policy π^A by observing the expert.

5. Problem of Policy Transfer

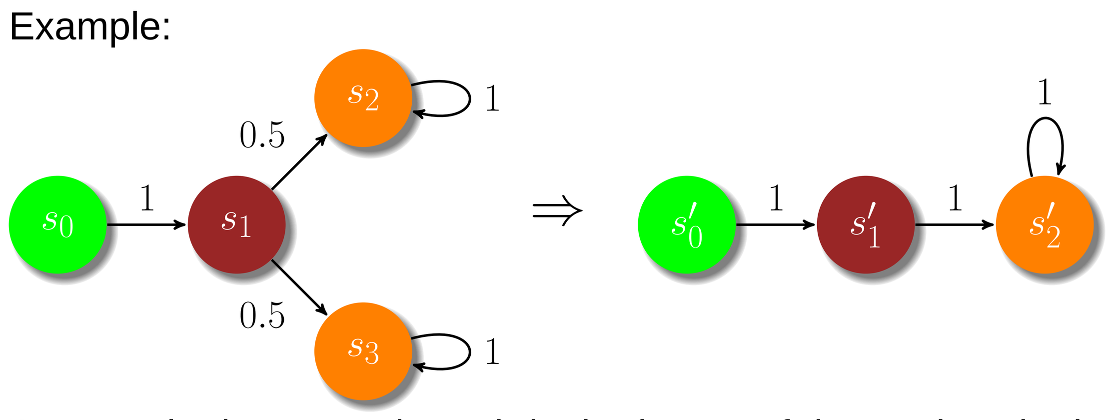
- Problem: How to generalize the expert's policy to states that have not been encountered during the demonstration.
- Previous works have attempted to solve this problem by representing the states as vectors of features, and classifying the states accordingly.
- Inverse reinforcement learning algorithms learn a reward function from the demonstration of the expert policy, and use it to find a generalized policy [Abbeel & Ng, 2004].
- These algorithms assume that the reward function can be expressed by considering only the features of the states.
- However, the reward function may depend on the topology of the graph rather than the features of the states.

6. MDP Homomorphism [Ravindran, 2004]

A homomorphism from MDP M to MDP M' is a surjective function f that maps every state in M to a state of M' such that:

$$T'(f(s_t), a, s'_{t+1}) = \sum_{s_{t+1} \in \mathcal{S}, f(s_{t+1}) = s'_{t+1}} T(s_t, a, s_{t+1})$$

Example:



A vertex in the second graph is the image of the vertices in the first graph that have the same color.

7. Soft MDP Homomorphism [Sorg & Singh, 2009]

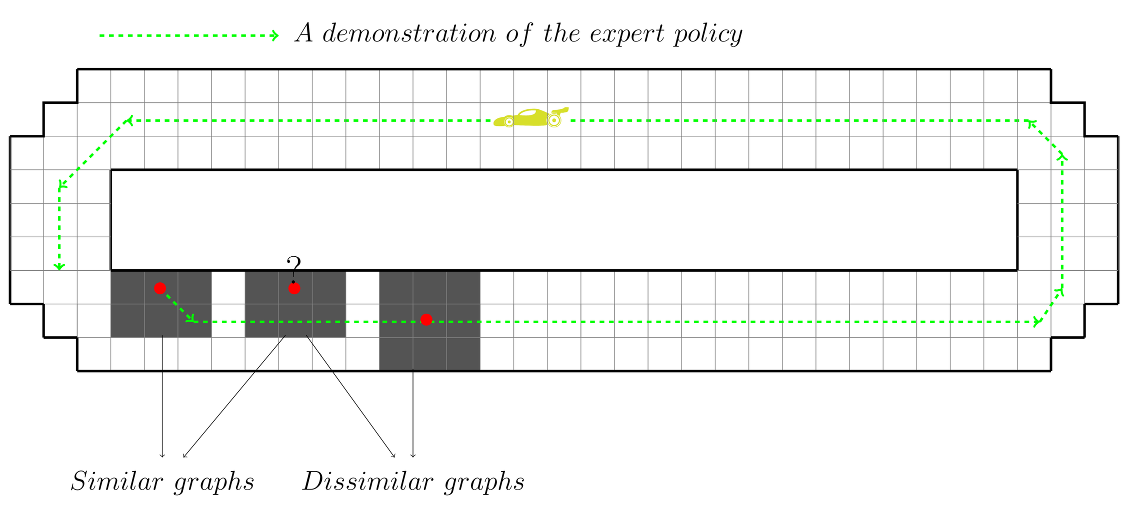
A soft homomorphism is a function f that maps every state in M to a distribution over the states of M' such that:

$$\sum_{s'_t \in \mathcal{S}'} f(s_t, s'_t) T'(s'_t, a, s'_{t+1}) = \sum_{s_{t+1} \in \mathcal{S}} T(s_t, a, s_{t+1}) f(s_{t+1}, s'_{t+1})$$

Finding a soft homomorphism can be casted as a linear program.

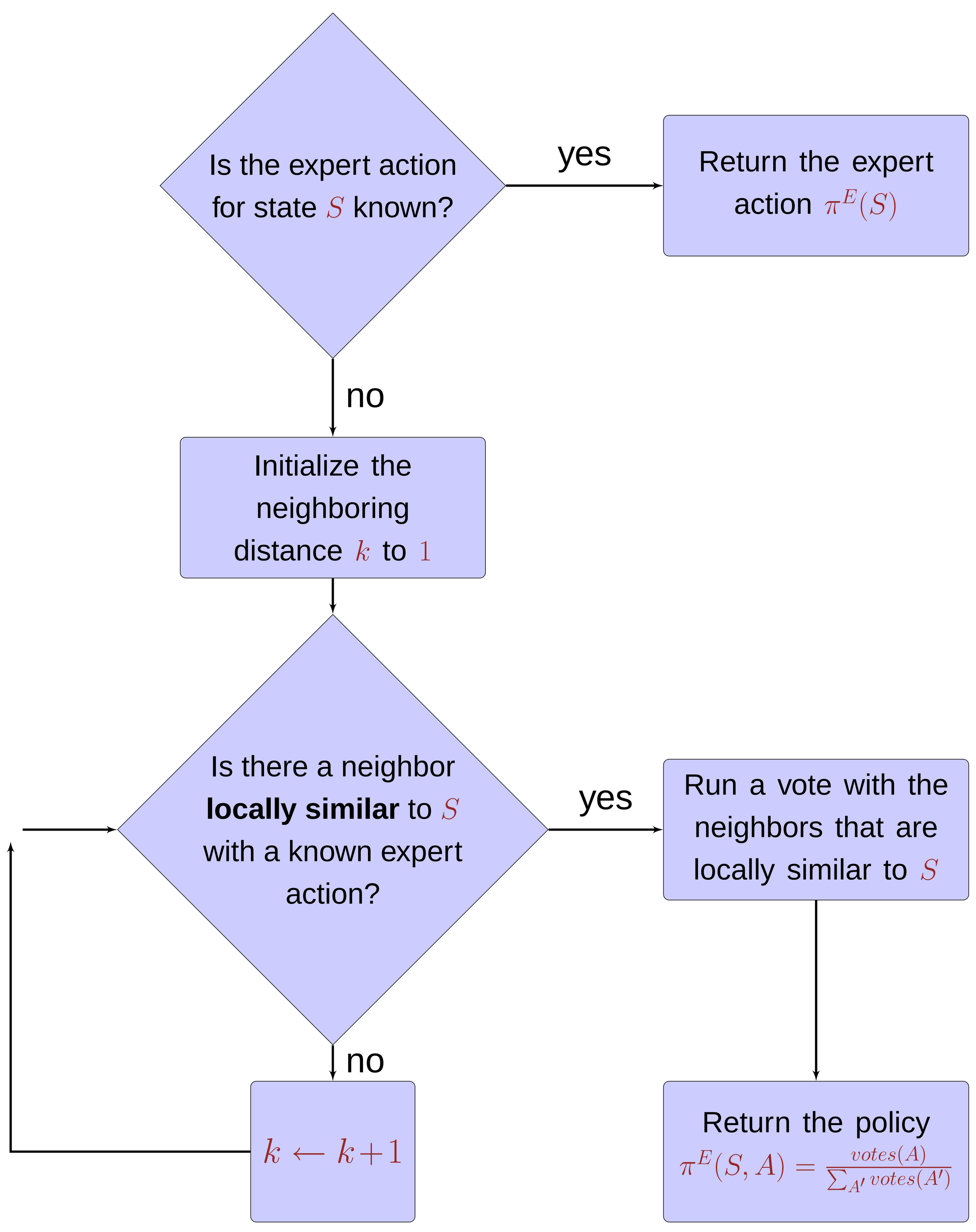
Definition. *Two states are locally similar if there is a soft homomorphism from the MDP defined by the neighbors (within a given distance d) of the first state to the MDP defined by the neighbors of the second.*

8. Racetrack Example

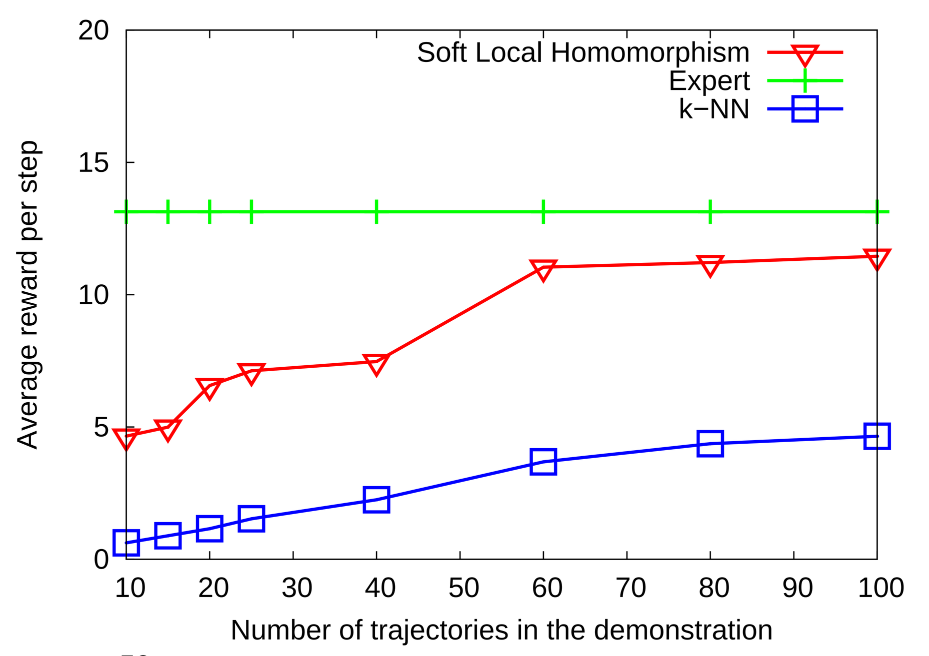


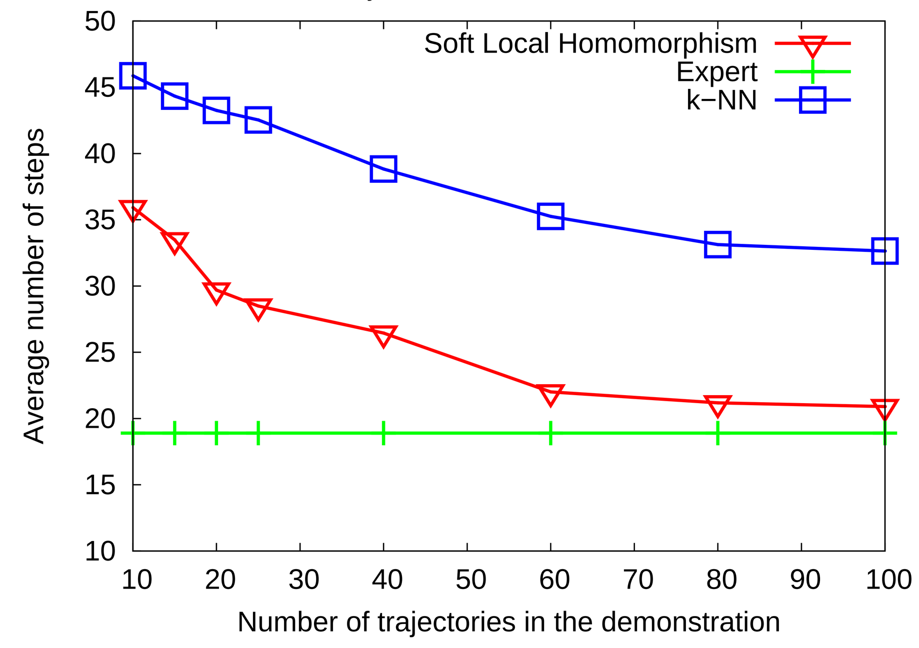
- There are two possible speeds in each direction of the vertical and horizontal axis, in addition to the zero speed in each axis.
- Actions: accelerate or decelerate in each axis, or do nothing.
- Actions succeed with probability 0.9 in low speeds and only 0.5 in high speeds.
- The cost of an off-road is -5 and the reward for reaching the finish line is 200.

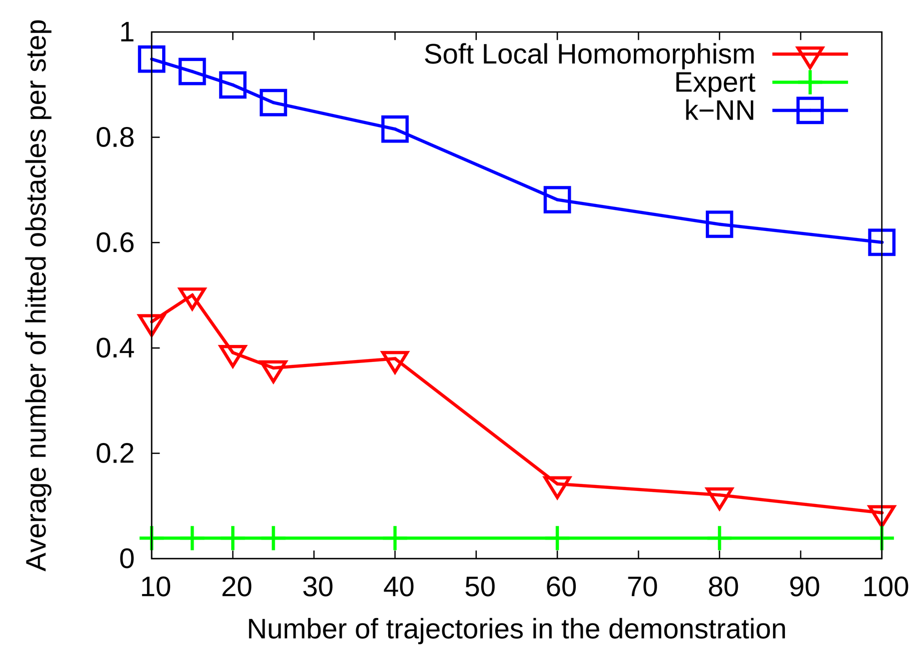
9. Algorithm



10. Results of the Racetrack Simulation







11. Conclusion and Future Work

- ✓ Policy transfer by soft local homomorphisms is well-suited for problems where the rewards depend on the topology of the graph.
- ✓ Using homomorphisms leads to a significant improvement in the quality of the policies learned by imitation.
- ✗ This approach involves solving $O(|\mathcal{S}|^2)$ linear programs, though the number of variables is bounded by the maximal distance.
- ✗ There are no guarantees about the optimality of the solution.

As a future work, we target to use random walk kernels as a measure of similarity between graphs, and find the theoretical guarantees about the optimality of the solution.

References

[Abbeel & Ng, 2004] Abbeel, P., & Ng, A. Y. (2004). Apprenticeship Learning via Inverse Reinforcement Learning. *Proceedings of the Twenty-first International Conference on Machine Learning (ICML'04)* (pp. 1–8).

[Ravindran, 2004] Ravindran, B. (2004). *An Algebraic Approach to Abstraction in Reinforcement Learning*. Doctoral dissertation, University of Massachusetts, Amherst MA.

[Wolfe & Barto , 2006] Wolfe, A., & Barto, A. (2006). Decision Tree Methods for Finding Reusable MDP Homomorphisms. *Proceedings of The Twenty-first National Conference on Artificial Intelligence (AAAI'06)* (pp. 530–535).

[Sorg & Singh, 2009] Sorg, J., & Singh, S. (2009). Transfer via Soft Homomorphisms. *Proceedings of The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)* (pp. 741–748).