Inverse Reinforcement Learning in Swarm Systems

Adrian Šošić, Wasiur R. KhudaBukhsh, Abdelhak M. Zoubir and Heinz Koeppl

Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt, Germany

ABSTRACT

In recent years, inverse reinforcement learning (IRL) has attracted increasing attention as a new paradigm for learning behavioral models from expert demonstrations. However, despite its growing popularity, most IRL algorithms available to date are designed for single-agent environments, whereas the field of multi-agent IRL remains largely unexplored. In this paper, we summarize the work presented in [1], which extends the IRL principle to homogeneous large-scale systems. We briefly review the main principles introduced in this work and show some selected results.

Introduction

The work in [1] presents a new framework for multi-agent inverse reinforcement learning (IRL) with the incentive to make the IRL principle applicable to large-scale systems. This is of potential use for many applications, such as programmable matter [2] and self-assembly systems [3]. The proposed framework is based on a newly introduced system class, called the *swarMDP*, which constitutes a sub-class of decentralized partially observable Markov decision processes (Dec-POMDP) with a homogeneous agent architecture. The purpose of introducing the swarMDP is that, within this framework, existing IRL methods can be straightforwardly generalized to the multi-agent case (in a certain sense) without further modifications. The underlying key observation is that most existing IRL frameworks, such as [4, 5, 6, 7, 8, 9], share a common generic form [10] (see Algorithm 1), which directly translates to the multi-agent setting under the above-mentioned homogeneity assumption. In the following sections, we briefly recapitulate the swarMDP model and show its use in the context of multi-agent IRL.

Algorithm 1: GENERIC IRL

Input: expert data $\mathcal{D}$, MDP without reward function

0: Initialize reward function $R^{\{0\}}$
   for $i = 0, 1, 2, ...$
1: Policy update: Find optimal policy $\pi^{\{i\}}$ for $R^{\{i\}}$
2: Value estimation: Compute corresponding value $V^{\{i\}}$
3: Reward update: Given $V^{\{i\}}$ and $\mathcal{D}$, compute $R^{\{i+1\}}$

Appears in: *Proceedings of the 1st Workshop on Transfer in Reinforcement Learning (TiRL) at the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), A. Costa, D. Precup, M. Veloso, M. Taylor (chairs), May 8–9, 2017, São Paulo, Brazil.*

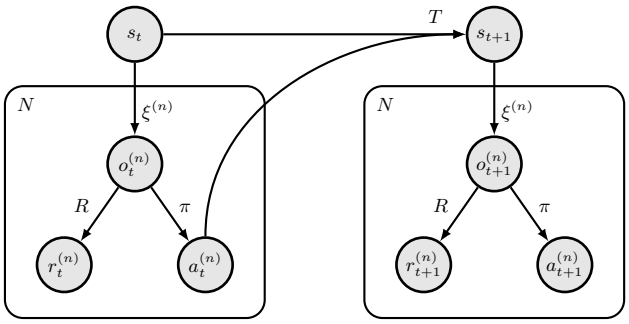

Figure 1: swarMDP visualized as a Bayesian network.

The swarMDP Model

The swarMDP model (Fig. 1) uses the concept of a *swarming agent*, which is defined as a tuple $\mathbb{A} \coloneqq (\mathcal{S}, \mathcal{O}, \mathcal{A}, R, \pi)$, where:

- $\mathcal{S}, \mathcal{O}, \mathcal{A}$ are sets of local states, observations and actions, respectively.
- $R : \mathcal{O} \to \mathbb{R}$ is an agent-level reward function.
- $\pi : \mathcal{O} \to \mathcal{A}$ is the local policy of the agent.

Taking the swarming agent as the basic building block, the swarMDP is constructed as a tuple $(N, \mathbb{A}, T, \xi)$, where:

- N is the number of agents in the system.
- $\mathbb{A}$ is a swarming agent prototype as defined above.
- $T : \mathcal{S}^N \times \mathcal{A}^N \times \mathcal{S}^N \to \mathbb{R}$ is the global transition model of the system.
- $\xi : \mathcal{S}^N \to \mathcal{O}^N$ is the observation model of the system.

To ensure the homogeneity of the system, T and ξ are required to respect the interchangeability of the agents, i.e. any permutation of the agents must not alter the dynamics of the model (see details in [1]).

IRL in Swarm Systems

The swarMDP model has several appealing properties, e.g. that the homogeneity of the agents also translates to their value functions, if defined appropriately. In [1], these properties are used to design a scalable multi-agent IRL procedure, based on Algorithm 1, whose main steps we briefly explain below. In this procedure, transfer learning takes place on two different levels: during the exchange of the local Q-values (Step 1) and at the construction of the global value estimate (Step 2). For details, we refer to the main paper.

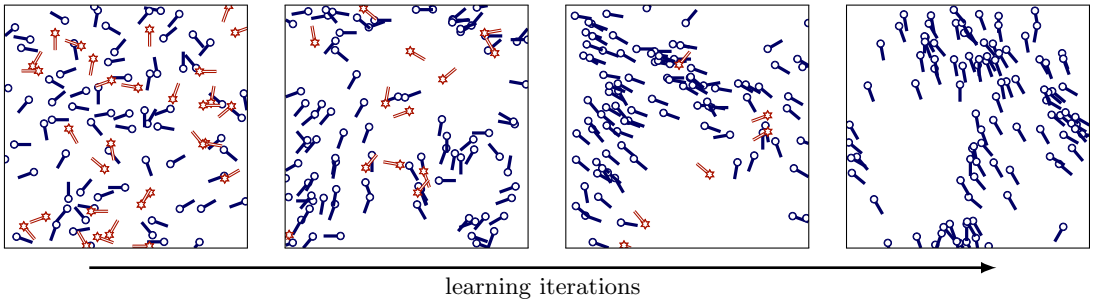

Figure 2: Snapshots of the proposed learning scheme applied to the Vicsek model. The agents are divided into a greedy set (—○) and an exploration set (=☆) to facilitate the exploration of locally desynchronized swarm states.

Step 1: Policy Update

The policy update step aims at optimizing the system's stationary behavior, whose performance is measured by the following agent-independent, stationary Q-function,

$$Q(o, a \mid \pi) = \mathbb{E}_{P(s|o^{(n)}=o,\pi)}\left[Q^{(n)}(s, a \mid \pi)\right],$$

which represents the local payoff for executing action a when making observation o in a stationary agent field generated by (and executing) policy π. The learning mechanism is designed as a repeated game played between a generic agent in this field and its surrounding (i.e. all other agents), where the agent tries to optimize its own local policy and the remaining agents react with a corresponding new swarm behavior. Algorithmically, the implementation of this game takes the form of a simple Q-learning procedure [11], where the agents exchange their local experiences in the form of Q-values,

$$\hat{Q}(o_t^{(n)}, a_t^{(n)}) \leftarrow (1-\alpha)\hat{Q}(o_t^{(n)}, a_t^{(n)}) + \alpha\big(r_t^{(n)} + \gamma \max_{a\in\mathcal{A}} \hat{Q}(o_{t+1}^{(n)}, a)\big).$$

In a rough analogy to the game, the agents are divided into a set of exploring agents, which take over the role of the generic agent, and a greedy set, which plays the role of the opponent targeting the behavior of the steady-state environment (Fig. 2). The result of the learning procedure is an optimized system policy that is obtained as

$$\pi^*(o) = \arg\max_{a\in\mathcal{A}} \hat{Q}(o, a \mid \pi).$$

Step 2: Value Estimation

Next, the performance of the returned policy is estimated based on the accumulated rewards of the agents,

$$V_{|\pi} \coloneqq \mathbb{E}_{P_0(s)}\left[V^{(n)}(s \mid \pi)\right] \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{\infty}\gamma^t r_t^{(n)}.$$

In contrast to the local Q-function shown above, this value is based on the global state configuration of the system and, therefore, it can be used to adapt the reward model in the last step of the algorithm so as to mimic the expert behavior.

Step 3: Reward Update

Lastly, the reward model is updated based on the estimated value. Interestingly, this update can be performed independently of the target system (see references in the introduction for algorithmic details). Hence, this step can be conducted by invoking any single-agent method that follows the basic procedure outlined in Algorithm 1. The results in [1] are obtained using the max-margin principle described in [4].

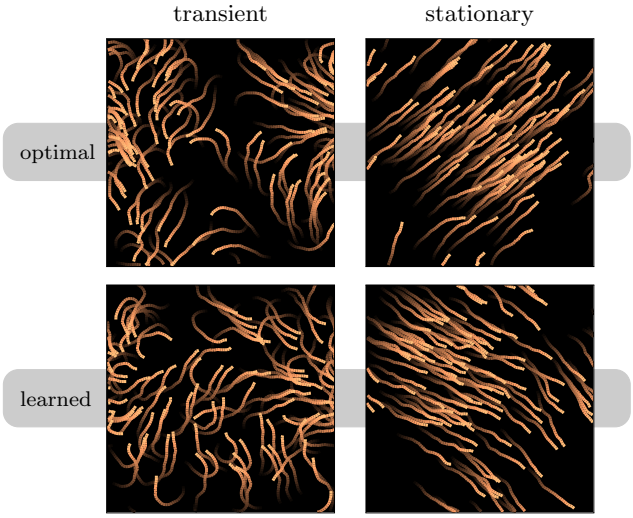

Figure 3: Trajectories of the Vicsek model. Note: the used observation model ignores the absolute direction of travel.

Simulation Results

As one of two example systems, [1] considers the Vicsek model [12], whose temporal dynamics are defined as

$$\theta_{t+1}^{(n)} = \langle\theta_t^{(n)}\rangle_\rho + \Delta\theta_t^{(n)},$$
$$x_{t+1}^{(n)} = x_t^{(n)} + v_t^{(n)}.$$

Herein, $x_t^{(n)} \in [0,1]\times[0,1]$ and $\theta_t^{(n)} \in [0, 2\pi)$ denote, respectively, the position and orientation of the nth agent at time t, $\langle\theta_t^{(n)}\rangle_\rho$ is the mean orientation of all agents within the ρ-neighborhood of agent n, $\Delta\theta_t^{(n)}$ represents an agent-specific noise term, and $v_t^{(n)} = v\cdot[\cos\theta_t^{(n)},\ \sin\theta_t^{(n)}]$ is the velocity vector of agent n, where v is the common absolute velocity of all agents. In order to provide the imitating system the necessary local information to copy the expert behavior, the observation model is chosen such that it enables the agents to monitor their local misalignment, i.e. $o_t^{(n)} = \xi^{(n)}(s_t) \coloneqq \langle\theta_t^{(n)}\rangle_\rho - \theta_t^{(n)}$. The available actions correspond to direction changes of various degrees.

Figure 3 shows some exemplary learning results which compare the learned behavior to the expert one. Further results, including the learned reward model, are provided in [1]. Videos of the learning process and the final policy can be found at `http://www.spg.tu-darmstadt.de/aamas2017`.

REFERENCES

[1] A. Šošić, W. R. KhudaBukhsh, A. M. Zoubir, and H. Koeppl. Inverse reinforcement learning in swarm systems. In *Proc. 16th International Conference on Autonomous Agents and Multiagent Systems*, 2017.

[2] S. C. Goldstein, J. D. Campbell, and T. C. Mowry. Programmable matter. *Computer*, 38(6):99–101, 2005.

[3] G. M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[4] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. 21st International Conference on Machine Learning*, page 1, 2004.

[5] Gergely Neu and Csaba Szepesvari. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. 23rd Conference on Uncertainty in Artificial Intelligence*, pages 295–302, 2007.

[6] A. Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proc. 17th International Conference on Machine Learning*, pages 663–670, 2000.

[7] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *Proc. 20th International Joint Conference on Artifical Intelligence*, pages 2586–2591, 2007.

[8] Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems*, pages 1449–1456, 2007.

[9] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. 23rd Conference on Artificial Intelligence*, pages 1433–1438, 2008.

[10] B. Michini and J. P. How. Bayesian nonparametric inverse reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2012.

[11] C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[12] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75(6):1226, 1995.