Inverse Reinforcement Learning of Bird Flocking Behavior

Robert Pinsler[1] and Max Maag[2] and Oleg Arenz[2] and Gerhard Neumann[2,3]

Abstract— **Birds within a flock are commonly assumed to be guided by simple rules, yet they show intelligent, collective behavior that is not entirely understood. We address this problem by modeling each bird as an agent of a separate Markov decision process, assuming that a bird makes decisions which maximize its own individual reward. By applying inverse reinforcement learning techniques to recover the unknown reward functions, we (1) were able to explain and reproduce the behavior of a flock of pigeons, and (2) propose a method for learning a leader-follower hierarchy. In the future, the learned reward representation could for example be used to teach a swarms of robots how to fly in a flock.**

I. INTRODUCTION

Flocks of birds can perform various complex maneuvers while maintaining highly synchronized motions. For example, in face of predators the evasion movement of one bird can be rapidly propagated through the flock, resulting in a coordinated turning maneuver [1]. The study of such collective behavior, as seen in bird flocks or school of fish, has spawned several mathematical models that are often inspired by biology [2], [3] or physics [4]. Various models assume that each individual only follows the basic principles of attraction, repulsion and alignment [5]–[7]. For example, Reynolds [5] was able to generate swarm-like behavior within computer simulation using these rules, suggesting that each bird follows the very same policy. If this policy is indecisive, conflicting actions are prioritized. Other models resolve this conflict by introducing different zones, within which each rule is effective [6], [8]. However, despite those attempts it is still largely unclear how exactly different rules interplay. Furthermore, the proposed mechanisms might not suffice to model the behavior of real birds accurately. In fact, Nagy et al. [9] were able to identify additional hierarchical patterns within small flocks of homing pigeons. The findings suggest that such dynamic leader-follower relationships play an important role for explaining flocking behavior.

Understanding the way birds interact is not merely of biological interest, however. One important field of application is swarm robotics, where self-organization between different autonomous agents is needed. Such robotic swarms can be used for environmental monitoring, rescue missions or for building up communication networks [10]. Our goal is to use insights from bird flocking to improve the coordination of such multi-agent systems. We are therefore interested in finding rules that explain the decisions of birds within a flock.

Markov Decision Processes (MDPs) are a powerful mathematical framework for modeling such decision making problems. We assume that each bird follows a (possibly different) policy that maximizes its long-term reward under the dynamics of the MDP. For instance, birds prefer to fly in a flock because it increases their chances of survival against predators. Assuming known dynamics, the problem of explaining the behavior of the birds then reduces to finding their reward function. By viewing each bird of a flock as an agent of a separate MDP, this issue can be formulated as an inverse reinforcement learning (IRL) problem [11], where the goal is to infer the underlying reward function of an agent from its observed actions.

Recently, there has been great interest [12]–[15] in devising IRL algorithms specifically tailored towards the multi-agent setting, often by exploiting shared structure among the agents. However, usually these methods make additional assumptions (e.g. the availability of a central controller, the possibility to collect more data using a learned policy, etc.) that are not suited for our application.

In this paper, we apply maximum entropy IRL to recover the reward functions of pigeons within a flock, where we used GPS data [9] from multiple flights of flocks of up to ten pigeons as expert trajectories. Furthermore, we show how to learn a leader-follower hierarchy from the recovered reward functions. The learned reward functions serve as succinct, transferable representations of the task. This does not only allow us to study the collective behavior of birds in a flock more closely but could also be used for apprenticeship learning [16] in swarms of robots.

II. BACKGROUND

This section fixes the notation and serves as an introduction for maximum entropy IRL in continuous MDPs.

A. Preliminaries

A finite MDP is a tuple $(S, A, \{P_{\boldsymbol{sa}}\}, R)$, where S is the state space, A is the action space, $\{P_{\boldsymbol{sa}}\}$ are the transition dynamics when taking action $\boldsymbol{a}$ in state $\boldsymbol{s}$, and $r(\boldsymbol{s}, \boldsymbol{a})$ is the reward function. In the IRL setting, the reward function is unknown. We assume that the reward function is a linear combination of features $\boldsymbol{\phi} \in \mathcal{R}^k$, i.e. $r(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{s}, \boldsymbol{a})$ with weights $\boldsymbol{\theta}$. The actions of the agent are selected according to policy $\pi(\boldsymbol{a}|\boldsymbol{s})$. An optimal policy π^* maximizes the expected return $J^\pi = \mathbb{E}_\pi[\sum_{t=0}^{T} r(s_t, a_t)]$, which denotes the sum of the expected rewards when following policy π, such that $\pi^* = \arg\max_\pi J^\pi$. By using the definition of the reward function, the expected return can be rewritten as

[1] Engineering Department, University of Cambridge, Cambridge, UK rp586@cam.ac.uk

[2] Fachbereich Informatik, Technische Universität Darmstadt, Germany

[3] School of Computer Science, University of Lincoln, Lincoln, UK

$J^\pi = \boldsymbol{\theta}^\top \tilde{\phi}^\pi$, where $\tilde{\phi}^\pi = \mathbb{E}_\pi[\sum_{t=0}^{T} \phi(s_t, a_t)]$ denotes the expected feature counts.

B. Maximum Entropy Inverse Reinforcement Learning

Maximum entropy IRL [17] chooses the least committed distribution over behaviors that still matches the expert feature counts. Under this model, the likelihood of a trajectory $\zeta_i = \{\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{s}_2, \boldsymbol{a}_2, \ldots, \boldsymbol{s}_T, \boldsymbol{a}_T\}$ is proportional to the exponential of the rewards obtained along the way:

$$P(\zeta_i|\boldsymbol{\theta}) = \frac{1}{Z} \exp\Big(\sum_t r(s_t, \boldsymbol{a}_t)\Big) \propto \exp \boldsymbol{\theta}^\top \boldsymbol{\phi}_{\zeta_i}. \quad (1)$$

However, evaluating the partition function Z is intractable for continuous domains. Levine and Koltun [18] therefore proposed to approximate the likelihood (1) using a Laplace approximation, yielding:

$$P(\zeta_i|\boldsymbol{\theta}) \approx e^{\frac{1}{2}\boldsymbol{g}^\top \boldsymbol{H}^{-1}\boldsymbol{g}} | - \boldsymbol{H}|^{\frac{1}{2}} (2\pi)^{-\frac{d_{\boldsymbol{a}}}{2}},$$

where $\boldsymbol{g} = \frac{\partial r}{\partial \boldsymbol{a}}$ and $\boldsymbol{H} = \frac{\partial^2 r}{\partial \boldsymbol{a}^2}$ are the gradient and Hessian of the sum of rewards along trajectory ζ_i w.r.t. action sequence $\boldsymbol{a} = [\boldsymbol{a}_0, \ldots, a_T]^1$. The approximation is equivalent to assuming that the expert trajectories are only locally optimal, eliminating the requirement of global optimality usually assumed in IRL. The approximate log likelihood objective is given by

$$\mathcal{L} = \frac{1}{2}\boldsymbol{g}^\top \boldsymbol{H}^{-1}\boldsymbol{g} + \frac{1}{2}\log|-\boldsymbol{H}| - \frac{d_a}{2}\log 2\pi, \quad (2)$$

which is maximized using gradient-based optimization.

III. Approach

In this section, we present our approach towards learning the reward function of pigeons. We use the pigeon flocking dataset of Nagy et al. [9] as training data. The position data was collected at a sampling rate of 0.2s during two different setups: free flights around their lair and homing flights. The provided data contains position, velocity and acceleration information. The GPS positions have a reported precision of 1-2m along the x- and y-coordinates and a substantially larger error in the z-direction.

A. Data preprocessing

Prior to the learning process, we conduct several preprocessing steps. First, time steps where data is missing for one or more bird are discarded. In addition, all time steps are filtered out where at least one bird is more than 200m away from the flock mean or has a velocity smaller than 1.5m/s. The remaining trajectory parts are split into several sub-trajectories of equal length. Velocity and acceleration information is re-computed at each sampled time step using forward differences, such that they are consistent with the proposed system dynamics.

B. Modeling

The decision making of the observed pigeons is modeled by bird-specific MDPs that only differ in the reward function of the respective bird. The problem of learning a reward function for each pigeon is thus decomposed into separate IRL problems. Because state and action spaces of birds are continuous, we follow [18] to approximate the log likelihood of the maximum entropy IRL objective. As system dynamics, we assume a double integrator:

$$\boldsymbol{s}_{t+1} = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{B}a_t$$

where $\boldsymbol{s}_t = \begin{bmatrix} x_1 & \dot{x}_1 & x_2 & \dot{x}_2 & x_3 & \dot{x}_3 \end{bmatrix}^\top$ and $\boldsymbol{a}_t = \begin{bmatrix} \ddot{x}_1 & \ddot{x}_2 & \ddot{x}_3 \end{bmatrix}^\top$ with position $\boldsymbol{x} \in \mathcal{R}^3$. $\boldsymbol{A}$ is a block-diagonal 6×6 matrix with blocks

$$\begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, \text{ and } \boldsymbol{B} = \begin{bmatrix} 0 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & dt \end{bmatrix}^\top,$$

where $dt = 0.2$. The reward function is modeled as a linear combination of k features as defined in Table I. The *Back Distance* and *Right Distance* features are based on observations from Nagy et al. [9], according to which leaders in pigeon flocks often fly in the front and to the left of the flock. Furthermore, the sum of each of the other birds' repulsions will be denoted as $\phi_{\sum \text{rep}} = \sum_{i=1}^{N_p} \phi_{\text{rep},i}$. Note that using both ϕ_{attr} and ϕ_{rep} (or $\phi_{\sum \text{rep}}$) allows to punish the agent when its distance to a flock member is either too small or too large. Finally, we define another bird $\bar{p}$ (in addition to the existing pigeons in the data set), which represents the flock mean. Its states are calculated as $\boldsymbol{s}_{\bar{p}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{s}_{p_i}$.

C. Hierarchy Learning

After learning the reward function for every pigeon we leverage the learned feature weights to infer a hierarchy that encodes leader-follower relationships. In order to compare the weights between pigeons, we apply the following normalization to each feature ϕ_k of bird a:

$$\hat{\phi}_{k,a} = \frac{\phi_{k,a} - \mu_{\phi_k}}{\sigma_{\phi_k}},$$

where μ_{ϕ_k} and σ_{ϕ_k} are the weight mean and standard deviation across the flock. We assume a pigeon a is following another pigeon p if the feature weight of a w.r.t. p is higher than some threshold $\tau = 1.0$. Intuitively, a high weight indicates that bird p has a high influence on the reward of agent a, therefore signalizing a stronger follower-leader relationship.

IV. Experiments and Results

We conducted different experiments with a varying set of reward features as summarized in Table II. A homing flight dataset was used that contained data of nine pigeons. The flight was split into consecutive training and test sets of length $T = 150$ each. First, we learned the reward function of one particular pigeon, A, on the training set. Since the true reward function is not known, we compared the original expert traces with the trajectories obtained from a policy that

[1] Under deterministic system dynamics and a fixed state distribution $\boldsymbol{s}_0$ a trajectory is completely determined by action sequence $\boldsymbol{a}$.

Feature	Expression	Description
Attraction	$\phi_{\text{attr}} = -\|\|\boldsymbol{x}_a - \boldsymbol{x}_p\|\|_2^2$	Stay close to others
Repulsion	$\phi_{\text{rep}} = -\exp\left(-\frac{(\boldsymbol{x}_a - \boldsymbol{x}_p)^2}{2\sigma^2}\right) \quad (\sigma = 3)$	Avoid crowding
Alignment	$\phi_{\text{align}} = \frac{\dot{x}_a^\top \dot{x}_p}{\|\|\dot{x}_a\|\|\|\|\dot{x}_p\|\|}$	Head in the same direction
Back Distance	$\phi_{\text{bdist}} = \log\left(1 + \exp\left(\frac{\dot{x}_m}{\|\|\dot{x}_m\|\|}(\boldsymbol{x}_p - \boldsymbol{x}_a)\right)\right)$	Avoid flying in the back of the flock ($\dot{x}_m$ denotes the mean flock velocity)
Right Distance	$\phi_{\text{rdist}} = \log\left(1 + \exp\left(\frac{\boldsymbol{e}_z \times \dot{x}_m}{\|\|\boldsymbol{e}_z \times \dot{x}_m\|\|}(\boldsymbol{x}_p - \boldsymbol{x}_a)\right)\right)$	Avoid flying on the right of the flock
Action Penalty	$\phi_{\text{act}} = -\|\|\boldsymbol{a}\|\|^2$	Avoid moving too much

TABLE I: Reward features of agent a with respect to another pigeon p.

Experiment	Features							Birds	
	ϕ_{attr}	ϕ_{rep}	ϕ_{align}	ϕ_{rdist}	ϕ_{bdist}	ϕ_{act}	$\phi_{\sum \text{rep}}$	All (except agent)	$\bar{p}$
AllAvg	X	X	X			X		X	X
All	X	X	X			X		X	
Base	X		X			X	X		X
R/B	X	X		X	X	X		X	

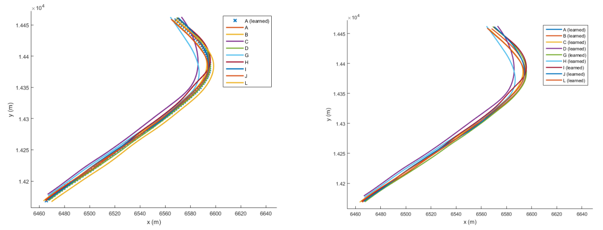
TABLE II: Summary of experiments. Each selected feature is created for every other pigeon of the flock (i.e. all birds except the agent itself), and optionally the mean flock $\bar{p}$. Note that ϕ_{act} is only relevant for the agent itself.

optimizes the learned reward function. The results are shown in Table III. While *AllAvg* and *R/B* yielded slightly smaller errors on the test data, the agent was in principle able to fly in the flock across all experimental settings. Fig. 1a shows some learned trajectories using the *Base* reward features. This suggests that the *Base* features already suffice to learn how to fly in a flock, although in reality it is unlikely that birds use features with respect to the flock average $\bar{p}$. Next, an individual reward function was learned for each flock member, one at a time. The resulting trajectories are depicted in Figure 1b. As can be seen, the different agents flew closely together during the complete time interval, suggesting they have learned to fly in a flock.

Next, we looked at more challenging free flights. In Fig. 2a, we successfully learned a reward function for each bird that jointly led to flocking behavior. Moreover, Fig. 2b shows that when the reward function was learned on a homing flight, we can still learn a sensible policy when transfered to a free flight.

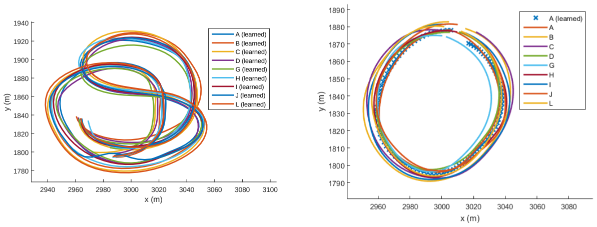
Metric	Experiment			
	AllAvg	All	Base	R/B
$\text{RMSE}_{\text{train}}(x_a, x_p)$	0.1449	0.1686	0.3946	0.2305
$\text{RMSE}_{\text{test}}(x_a, x_p)$	0.2949	0.3666	0.3947	0.2668

TABLE III: RMSE between learned trajectory for agent A



(a) (b)

Fig. 1: **(a)**: Learned trajectory of pigeon A along the original trajectories of the birds for *Base* experiment. **(b)**: Jointly learned trajectories for all pigeons.



(a) (b)

Fig. 2: **(a)**: Jointly learned trajectories for all pigeons on free flight. **(b)**: Reward function learned on homing flight and transfered to free flight.

Based on the learned weights, we then created a hierarchy as described in Section III-C. Since this requires features for individual birds, no hierarchies can be formed with the *Base* feature set. Furthermore, the *All* and *AllAvg* feature sets often result in hierarchies with cycles. Overall, hierarchies based on the ϕ_{rdist} and ϕ_{bdist} features led to more plausible results. This conforms to the observation that leader quality correlates with how far to the front and to the left a pigeon flies in the flock [9]. Fig. 3 shows an example of a learned hierarchy. However, in contrast to the findings by Nagy et al. [9] we were not able to produce a robust hierarchy that is consistent over several flights.

V. Conclusion

We formalized a flock of birds as a set of agents of separate MDPs, assuming each one of them attempts to maximize an internal reward function. Based on this for-

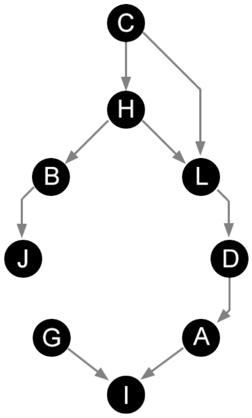


Fig. 3: Leader-follower network inferred from learned weights of ϕ_{rdist} features of every pigeon. A directed edge points from the follower to the leading bird.

mulation, we applied maximum entropy IRL to recover the reward functions of a set of pigeons. During multiple experiments with different sets of reward features we were able to produce flock-like behavior. We found that this works particularly well when taking the behavior of the flock mean as well as the repulsion to the other pigeons into account. Furthermore, we inferred a hierarchy based on the learned weights. However, we were not able to produce a consistent hierarchy across different flights.

In the future one could attempt to learn hierarchies based on the pigeons' temporal reaction delay as in [9]. Furthermore, it would be interesting to extend this work to more complex reward functions and unknown dynamics. However, doing so in light of limited data is still an open problem. Finally, an exciting future application of this work is to transfer the learned reward functions to control robot swarms. By choosing a leader, the agents would be able to follow that leader while staying in a flock. The flock can then be controlled by supplying a custom trajectory or goal position for the leader.

References

[1] W. K. Potts, "The chorus-line hypothesis of manoeuvre coordination in avian flocks," *Nature*, vol. 309, no. 5966, pp. 344–345, 1984.

[2] B. L. Partridge, "The structure and function of fish schools," *Scientific american*, vol. 246, no. 6, pp. 114–123, 1982.

[3] F. H. Heppner, "Three-dimensional structure and dynamics of bird flocks," *Animal groups in three dimensions*, pp. 68–89, 1997.

[4] 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*, vol. 75, no. 6, p. 1226, 1995.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Annual Conference on Computer Graphics and Interactive Techniques*, vol. 21, no. 4, 1987, pp. 25–34.

[6] I. Aoki, "A simulation study on the schooling mechanism in fish," *Bulletin of the Japanese Society of Scientific Fisheries*, vol. 48, no. 8, pp. 1081–1088, 1982.

[7] A. Huth and C. Wissel, "The simulation of the movement of fish schools," *Journal of theoretical biology*, vol. 156, no. 3, pp. 365–385, 1992.

[8] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of theoretical biology*, vol. 218, no. 1, pp. 1–11, 2002.

[9] M. Nagy, Z. Akos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 7290, p. 890, 2010.

[10] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International Workshop on Swarm Robotics*, 2004, pp. 10–20.

[11] A. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *International Conference on Machine Learning*, 2000, pp. 663–670.

[12] A. Šošić, W. R. Khuda-Bukhsh, A. M. Zoubir, and H. Koeppl, "Inverse reinforcement learning in swarm systems," in *Conference on Autonomous Agents and Multi-Agent Systems*, 2017, pp. 1413–1421.

[13] L. Dufton and K. Larson, "Multiagent policy teaching," in *(International Conference on Autonomous Agents and Multiagent Systems*, 2009.

[14] T. S. Reddy, V. Gopikrishna, G. Zaruba, and M. Huber, "Inverse reinforcement learning for decentralized non-cooperative multiagent systems," in *International Conference on Systems, Man, and Cybernetics*, 2012, pp. 1930–1935.

[15] S. Natarajan, G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting, and J. Shavlik, "Multi-agent inverse reinforcement learning," in *International Conference on Machine Learning and Applications*, 2010, pp. 395–400.

[16] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine Learning*, 2004.

[17] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, vol. 8, 2008, pp. 1433–1438.

[18] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *International Conference on Machine Learning*, 2012, pp. 475–482.