Learning Schooling Behavior from Observation

Brian Hrolenok and Tucker Balch

Georgia Institute of Technology, Atlanta, GA 30332
{bhroleno, balch}@cc.gatech.edu

Abstract

Agent-based simulation is a valuable tool for biologists studying animal behavior, however constructing models for simulation is often a time-consuming manual task, and validation of these models requires a principled approach. We present a framework for using machine learning techniques to automatically construct behaviors from tracking data of live animals from video that can be run in a simulated environment. Using this framework, we provide results for automatically learning the schooling behavior of *Notemigonus crysoleucas*.

Introduction

The motivation for this work has been to enable the work of biologists that study collective behavior through agent-based models. Agent-based models have been successful in analyzing the behavior of social insects such as ants and bees (Pratt et al., 2005; List et al., 2009), although currently such models are constructed after manual processing of video of the collective behavior of the animal. An automated method for constructing these models would enable more rapid iterative refinement of biological theories by allowing researchers to test hypotheses *in silico* with parameters that would be difficult to manage in real animals, as well as provide a tool for performing principled validation as outlined by Yang et al. (2012).

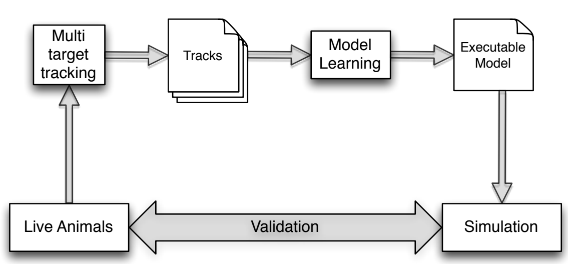

Figure 1: Workflow for automatically constructing executable behaviors from observation

The manual process of model creation usually consists of frame-by-frame annotation of video of the animals in question and statistical analysis of the resulting data, and the automation of this process can be decomposed into two corresponding subproblems: multi-target tracking of animals in video, and learning an executable model from those tracks. This workflow is outlined in Figure 1. The computer vision community has developed a number of algorithms for solving the multi-target tracking problem in specific domains, including tracking biological agents such as humans and ants (Feldman et al., 2012). Given a tracking algorithm that can produce tracks of individual agents with reasonable accuracy, the task is then to construct an executable agent-based model of behavior from the given data.

Learning fish schooling behaviors

The schooling of *Notemigonus crysoleucas* is an interesting collective behavior, one example of many types of "flocking" behavior found in nature. While the motion of the group as a whole is generally very complex, Reynolds (1987) has shown that individuals following fairly simple local rules can result in global flocking behavior. If we can then correctly learn a model of how the fish react to the features of their local surroundings, we should be able to reproduce the global schooling behavior by simulating fish in a similar environment that react according to the learned model. This means we need to identify which features of the environment the fish are reacting to, compute those features for each track in the tracking data, learn a mapping from features to reactions, and compute the identified features as part of the simulation.

Fish sensor features

There are several important features of the environment that effect how individual fish act as part of a school, and how the schooling phenomenon arises in groups of fish. We took inspiration from both classic flocking literature (Reynolds, 1987) and more recent work by Katz et al. (2011) in determining which features to include. From the collected tracking data we compute 13 features: 8 proximity sensors, the

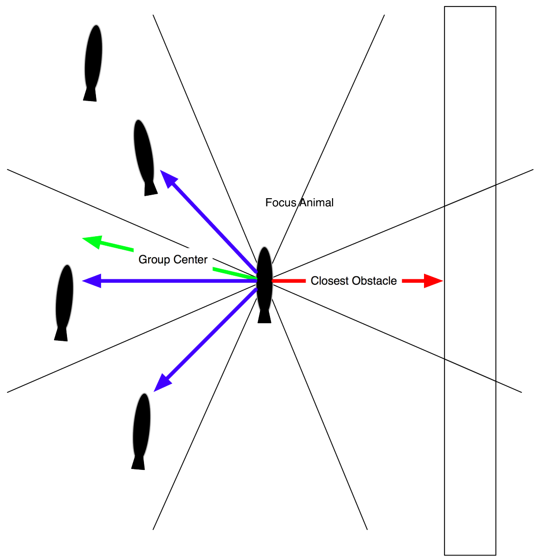

Figure 2: Sensor model for *Notemigonus crysoleucas*

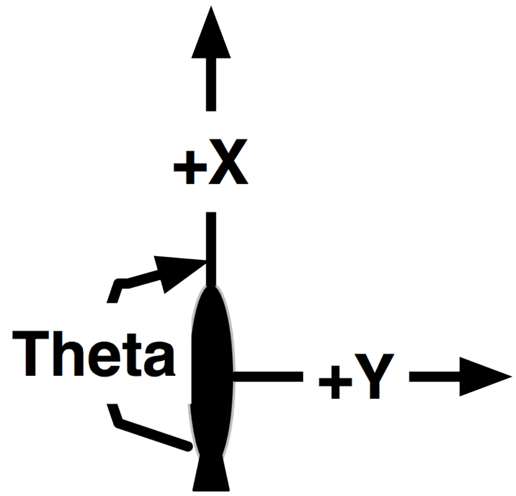

Figure 3: Actuator model for *Notemigonus crysoleucas*

x and y components of the normalized vector to the school center, the x and y components to the nearest obstacle, and a binary (near-far) distance to the school centroid that is one if the fish is within 3 body lengths of the school centroid and zero otherwise. The proximity sensors are thresholded at 4 body lengths, and the obstacle vector and school centroid calculation are both limited to objects within 1m.

These specific features can be thought of roughly corresponding with three of the classic components proposed by Reynolds: the 8 proximity sensors are useful for determining *separation*, the obstacle vector provides a mechanism for avoiding environmental obstacles, and the group center vector influences *cohesion*. Notice that we do not include any alignment term, and as Katz et al. suggest, the apparent group alignment is an emergent phenomenon, not a determining feature that the individual fish react to.

Fish actuators

In order to learn how the fish should react to a given feature vector, we must also quantify how the observed fish actually moved in response to the computed features. The tracking data includes the position (x, y) and orientation (θ) of each fish at each time step (see Figure 3). From consecutive time steps, we can calculate the change in position and orientation as a rough estimate of the velocity of the fish in reaction to its local environment, as long as the time interval is relatively short (the tracking data we use is computed frame-to-frame from video running at 30Hz).

Learning

Using the paired feature vector and velocity estimate as training data, we can construct a k-NN which maps any new feature vector to the k most similar training instances and the associated observed motions. One interesting difference from the standard k-NN in this instance is that the output associated with each feature vector is a continuous set of values describing how the fish moved, rather than a discrete class. In the standard k-NN with discrete output, each of the k nearest neighbors to a given query q votes for one of the possible discrete outputs and the output with the highest number of votes is returned as the class for the query. We can generalize this by returning the output of an arbitrary function g of the k nearest neighbors for a given q:

$$f(q) = g(\{q_i \mid d(q_i, q) <= d(q_j, q), \forall i < j, i = 1 \dots k\})$$

where $d(q_i, q)$ is the distance between q and q_i. In the standard k-NN, the function g just returns the class with the maximum number of votes, or the mode of the classes of the k neighbors. Other choices for g include the mean, or median. Empirically, we've found that sampling randomly from the k neighbors works better than taking the mean. This might be due to the fact that the animals do not behave in a completely deterministic manner: In the case where the fish is approaching a wall head-on it may turn left or right to avoid it, but the average of both cases would be to head straight forward, leading to a collision with the wall. On the other hand, sampling randomly from the neighbors would produce both left and right turns, and in the proportion that they are represented in the data.

Simulations of learned behavior

For our training set, we used tracking data collected from a 54 minute video of 30 *Notemigonus crysoleucas* schooling in a shallow tank 2.1 meters long by 1.2 meters wide[1]. From

[1]This data was one replicant from the experiments performed in Katz et al. (2011).

the tracks we computed the 13 features and 3 velocities described previously. The collected data amounted to roughly 2.6 million input/output pairs, which we used as training data for a k-NN. We constructed a simulation with 30 fish in a similar environment using BioSim, a freely available simulation toolkit[2]. At each time step, each fish computes the 13 features described earlier and sets its velocity by selecting from the k nearest neighbors. Figure 4 shows screenshots from the simulation and the resulting schooling behavior.

The fish are initially placed in the environment at random locations moderately spread out, but they quickly form into a single dense school. The school tends to stay close to the boundaries, tightly clustered. This is very similar to the behavior of the real fish in the training data, as shown in Figure 5.

As discussed in our motivation, one reason such agent-based simulations are useful is the ability to run experiments in simulation that would be difficult or time consuming to perform using live test animals. To illustrate this capability, we ran a simulation of 300 fish in a larger (3m by 5m) tank.

Figure 6 shows a screenshot of the 300 fish simulation. Notice how the fish have separated into several distinct schools.

Conclusion and Future work

This work has illustrated how the process outlined initially can be applied to learn the schooling behavior of fish from video: by applying a standard multi-target tracking algorithm to video to produce tracks of position and orientation, then computing a set of input/output (features/motions) pairs from the tracking data, then using those pairs as training data for a learning algorithm (k-NN) to construct a mapping between observed features and agent output, and finally using that mapping as the basis for a simulation. Our experimental results show that the collective behavior of agents following the learned behavior is qualitatively similar to the schooling behavior which generated the training data.

It's important to note that the choice of algorithm for both the tracking and learning components are crucial. The noise inherent in the tracks produced by the tracking algorithm must be relatively small, otherwise the training data used by the learning algorithm may be so noisy as to not permit an accurate mapping. The tracking algorithm must also be able to account for all the variables of interest, such as orientation. The choice of learning algorithm also has a profound effect. In the case of schooling fish, it is apparent from flocking models that the collective schooling behavior can arise from purely local and *reactive* rules. In other words, the mapping we've discussed so far is *stateless* in that the output is dependent only on the observed features, and not any internal memory or state. However there are many interesting types of behavior that are not stateless in this sense, such as foraging in ants (Yang et al., 2012) or the honey bee "waggle dance" (Oh et al., 2005). Learning these types of behaviors requires an algorithm that can handle state such as presented by Balch et al. (2006), and such algorithms are a focus of our current and future work.

[2] `https://github.com/biotracking/biosim2`

Acknowledgements

We would like to thank Iain Couzin and the Collective Animal Behavior lab at Princeton University for the use of their tracking data.

References

Balch, T., Dellaert, F., Feldman, A., Guillory, A., Isbell, C. L., Khan, Z., Pratt, S. C., Stein, A. N., and Wilde, H. (2006). How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(7):1445–1463.

Feldman, A., Hybinette, M., and Balch, T. (2012). The multi-iterative closest point tracker: An online algorithm for tracking multiple interacting targets. *Journal of Field Robotics*, 29(2):258–276.

Katz, Y., Tunstrøm, K., Ioannou, C. C., Huepe, C., and Couzin, I. D. (2011). Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725.

List, C., Elsholtz, C., and Seeley, T. (2009). Independence and interdependence in collective decision making: an agent-based model of nest-site choice by honeybee swarms. *Philosophical transactions of The Royal Society B: biological sciences*, 364(1518):755–762.

Oh, S. M., Rehg, J. M., Balch, T., and Dellaert, F. (2005). Data-driven mcmc for learning and inference in switching linear dynamic systems. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2*, pages 944–949. AAAI Press.

Pratt, S., Sumpter, D., Mallon, E., and Franks, N. (2005). An agent-based model of collective nest choice by the ant *Temnothorax albipennis*. *Animal Behaviour*, 70(5):1023–1036.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM.

Yang, Y., Quitmeyer, A., Hrolenok, B., Shang, H., Nguyen, D., Balch, T., Medina, T., Sherer, C., and Hybinette, M. (2012). Ant hunt: Towards a validated model of live ant hunting behavior. In *Twenty-Fifth International FLAIRS Conference*.

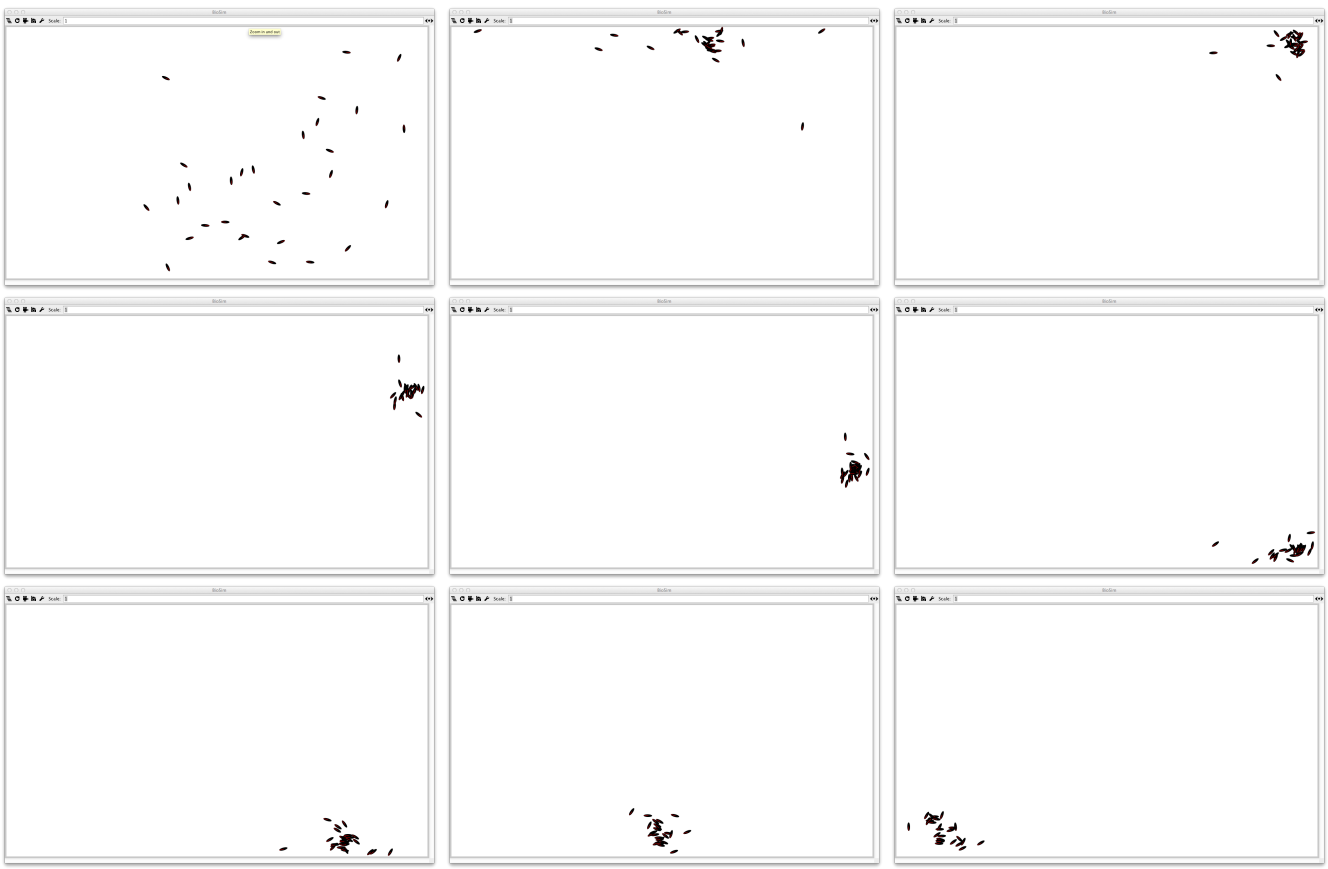

Figure 4: Simulated fish at consecutive intervals. The fish have a strong tendency to stay with the school, and congregate near the walls of the tank much like the real fish.

Figure 5: Replayed tracking data of real fish.

Figure 6: Simulation of 300 fish in a large tank using the same 30 fish training data