

Sampling Beats Fixed Estimate Predictors for Cloning Stochastic Behavior in Multiagent Systems

Brian Hrolenok and Byron Boots and Tucker Hybinette Balch

School of Interactive Computing
Georgia Institute of Technology
{bhroleno,bboots,tucker}@cc.gatech.edu

Abstract

Modeling stochastic multiagent behavior such as fish schooling is challenging for fixed-estimate prediction techniques because they fail to reliably reproduce the stochastic aspects of the agents behavior. We show how standard fixed-estimate predictors fit within a probabilistic framework, and suggest the reason they work for certain classes of behaviors and not others. We quantify the degree of mismatch and offer alternative sampling-based modeling techniques. We are specifically interested in building executable models (as opposed to statistical or descriptive models) because we want to reproduce and study multiagent behavior in simulation. Such models can be used by biologists, sociologists, and economists to explain and predict individual and group behavior in novel scenarios, and to test hypotheses regarding group behavior. Developing models from observation of real systems is an obvious application of machine learning. Learning directly from data eliminates expensive hand processing and tuning, but introduces unique challenges that violate certain assumptions common in standard machine learning approaches. Our framework suggests a new class of sampling-based methods, which we implement and apply to simulated deterministic and stochastic schooling behaviors, as well as the observed schooling behavior of real fish. Experimental results show that our implementation performs comparably with standard learning techniques for deterministic behaviors, and better on stochastic behaviors.

Introduction

Executable models of agent behavior are increasingly common in the work of biologists, ethologists, and sociologists as part of the growing trend in Agent Based Modeling (ABM) within these sciences. In the study of collective decision making by social insects, ABMs are commonly used, such as in the work of Pratt et al. on the ability of *Temnothorax rugatulus* to collectively choose nest sites (2005), or the work of List, Elsholtz, and Seeley on forage site selection by honeybees (2009). Tunström et al. examine the schooling behavior of *Notemigonus crysoleucas* and classify emergent properties of the school using an ABM in simulation (2013). Hemelrijk has studied the social interactions and group dynamics of monkeys, and compared competing theories of behavior in simulation by examining the characteristics

of simulations run with different ABMs (Hemelrijk 2000; Hemelrijk, Wantia, and Gygax 2005).

The ABMs in each of these examples have varying degrees of complexity, in terms of number of parameters and the interactions between those parameters. Increasingly complex models allow researchers to study more complex individual behaviors, but at the same time require more tuning, which in turn requires more observational data.

With improvements in computational capacity, the ability to execute more and more complex simulations to explore the ramifications of particular theories of behavior has become part of the standard toolkit, and demands a corresponding increase in the ability of researchers to automate the process of fitting behavior models to observational data. However, as we discuss throughout the rest of this work, applying standard techniques from machine learning directly can lead to poor performance. In the remainder of this paper we present two novel contributions. First we show that some stochastic behaviors cannot be reproduced well by fixed-estimate predictors, and in the experiments section we introduce K-L divergence as a measure of similarity between observed and generated behavior. In the next section we discuss some previous work in this area.

Related Work

While the automatic construction of executable models of behavior is relatively new, it has precedents within the machine learning community under categories such as behavior or activity recognition. Ethograms, a concept from ethology, are one way of categorizing, describing, and understanding the relationships between various behaviors. Recent work on automating the process of collating observational data into these ethograms in the case of fruit flies illustrates the utility of machine learning approaches (Branson et al. 2009; Dankert et al. 2009), though the ethograms produced in this way are descriptive, rather than executable.

Egerstedt et al. present a technique for automatically constructing minimal control sequences from tracking data, and use this to build models of ant foraging behavior using tracks of ants (2005). These are executable models, however this approach does not address how the recovered control sequence should handle unexpected sensor values, and the authors note that this is an open question that needs to be answered if the learned controllers are to be used in a simula-

tion.

Oh et al. present a method for using switching linear dynamical systems (SLDS) to model and predict the distinct states in the “honey bee waggle dance” — a mechanism for communicating direction and quality of food sources — using Markov chain Monte Carlo (MCMC) sampling for approximate inference (2005). While SLDS models are generative, this work focuses only on the predictive capabilities of the model. In fact, most generative models in use in behavior recognition are not in fact used to *generate* behavior. Notable exceptions to this include work from the behavioral cloning, Learning from Demonstration (LfD), and robotics communities where hidden Markov models (HMMs) are used in action reproduction, however the resulting trajectories are often post-processed to ensure smoothness and continuity constraints (Inamura et al. 2004; Sugiura and Iwahashi 2008; Kulić, Takano, and Nakamura 2008).

More directly related to the construction of executable models is some recent preliminary work on learning models of fish schooling behavior, and assessing the quality of these learned models (Hrotenok and Balch 2013; 2014). It’s within the context of assessing executable models that the unique challenges of this domain make themselves most apparent, in that standard measures of predictive performance do not capture some important qualities of an executable model, especially in the case of unpredictable or *stochastic* behaviors. In the next section, we discuss the difference between deterministic and stochastic behaviors, and some specific consequences of that difference for a common class of predictors and the executable models that rely on them.

Deterministic and Stochastic Behaviors

Consider the behavior of an agent which reacts to its environment in a deterministic way. It is reasonable to cast the problem of predicting the behavior of the agent as a supervised learning problem, where the function to be predicted $f : \mathbb{R}^d \rightarrow \mathcal{B}$ is the behavior of the agent, which is a function of some (known) features $\phi : \mathcal{S} \rightarrow \mathbb{R}^d$ of the current state of the environment $s \in \mathcal{S}$ that are relevant to the agent’s behavior. We train a predictor \hat{f} of the true behavior f using a set of training data

$$D = \{(\phi(s_i), b_i) : b_i = f(\phi(s_i))\} \quad (1)$$

where each b_i in the dataset is the observed behavior of an agent, given state s_i . We can collect such a dataset by observing the behavior of the agent and the state of the environment. The key design decisions in this formulation are the features ϕ , the form of the model \hat{f} , and the learning procedure that fits \hat{f} to f given D . Although the selection of ϕ has a significant impact on the performance of \hat{f} , we will consider it as given for the remainder since it typically requires significant domain knowledge and has been discussed previously (Hrotenok and Balch 2014).

If we assume that our observations are noise free, we can collect sufficient data, and our predictor \hat{f} has the expressive capability to reproduce f , then we can simulate what

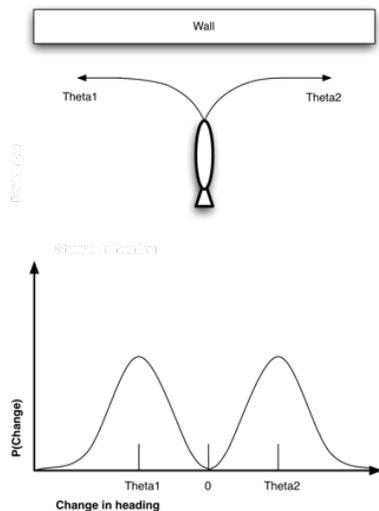


Figure 1: Graphical representation of a realistic distribution of behavior who’s expected value has low probability. The observed frequency of turning left versus right are equal, so the distribution of behavior has two equal modes. The mean θ , which minimizes expected error, has very low probability, and if used in an executable model would generate unrealistic behavior.

the agent would do for any given environmental state s , and this simulation should reproduce the observed behavior as well as accurately predict behavior in unobserved states. Of these three assumptions, the least reasonable is that of noise, which we address next.

In the typical supervised learning setting, training data D is assumed to be noisy in the sense that some uncontrollable aspect of the data collection process introduces an unknown (but small and normally distributed) amount of noise to the true output of f , that is,

$$D = \{(\phi(s_i), b_i) : b_i = f(\phi(s_i)) + \epsilon_i\} \quad (2)$$

where $\epsilon_i \sim \mathcal{N}(0, I\sigma)$. Frequently, it is assumed that variance in the observed behavior for a given point in the feature space is due to the noise process, and as a result many learning algorithms use estimates of central tendency which are robust to noise. The least-squares solution for linear regression is one such example, where our predicted output for an unobserved state s would be

$$\hat{f}(\phi(s)) = \langle \hat{W}, \phi(s) \rangle \quad (3)$$

where \hat{W} are the weights found by the least-squares solution. This estimate minimizes expected difference between the predicted output and the training data, which is achieved by predicting along a line as close to the central tendency of the training data as is possible, under the given assumptions that f is linear, and the noise process is normally distributed, and importantly that the underlying behavior is in fact deterministic.

If the behavior is instead *stochastic*, that is, it contains some unpredictable element that can be treated as non-deterministic, the variance in the observed behavior is not

completely due to the noise process and should be accounted for by \hat{f} . In this case it is more natural to think of f as a distribution over the behavior space \mathcal{B} . The observations of the agents behavior are then seen as samples drawn from f :

$$D = \{(\phi(s_i), b_i) : b_i \sim f(\phi(s_i))\} \quad (4)$$

Note that we can cast the ‘‘noisy observation’’ formulation in (2) to this stochastic behavior version by assuming that f takes a particular form. For example, in the case of least squares linear regression, $f(\phi(s)) = \mathcal{N}(\langle W, \phi(s) \rangle, I\sigma)$, while keeping \hat{f} as defined in (3). This highlights an issue with using fixed estimates as predictors for behavior. If f is symmetric, unimodal, and falls to zero rapidly as distance from the mode increases, having \hat{f} predict a behavior near the mode (as in (3)) will often be a good estimate, and so can be fixed for any specific $\phi(s)$. However, if f is multimodal, asymmetric, or has ‘‘fat tails’’, the probability of behaviors far from any central tendency of f can grow significantly, and while the sample mean may in fact minimize the predictive error, the probability of behaviors near the mean may be arbitrarily small.

To illustrate this, take for example the behavior of a fish swimming in a shallow tank of water, which we illustrate in Figure 1. As the fish approaches the wall of the tank, it can choose to turn left or right. From the perspective of the fish, it does not matter which direction it turns, so long as it does not collide with the wall. The argument could be made that a deterministic behavior is still feasible since the probability of approaching a wall exactly along the perpendicular is effectively zero, but it is also true that fish can only perceive their environment through the filter of imperfect sensors (which notably have a limited field of view with minimal overlap specifically along the perpendicular) and so may be prone to misperception or perceptual aliasing. It makes sense then that the distribution of behavior for a fish approaching a wall would exhibit two modes corresponding to both choices¹. It is also clear that in this scenario the mean or median behaviors correspond to a perfectly unnatural behavior (moving directly towards the wall) that nevertheless minimizes the expected difference between predicted and observed behavior training examples, and as such is the ‘‘best’’ fixed estimate of behavior. If the goal is to produce realistic predictions of behaviors of this kind, it’s clear we cannot rely on fixed-estimate predictors.

An obvious alternative is for \hat{f} to estimate the true density f directly, and then *sample* under \hat{f} . Since the benefit of this sampling is only evident in cases where the distribution of behaviors are not accurately modeled by Gaussian or similar distributions, and in general we do not know the form f takes *a priori*, we will rely on non-parametric density estimation. The standard approach to non-parametric density estimation is kernel density estimation, which takes the following form (for univariate b)

$$\hat{f}_h(b) = \frac{1}{|D|h} \sum_{b_i \in D} K\left(\frac{b - b_i}{h}\right) \quad (5)$$

where K is a chosen kernel function, and h is the bandwidth parameter. Note that this formulation is independent of ϕ , which is not useful if we want our agents to vary their behavior as a function of their environment, and so we augment the above definition by restricting the range of the sum:

$$\hat{f}_h(b; \phi(s)) = \frac{1}{|N(\phi(s))|h} \sum_{b_i \in N(\phi(s))} K\left(\frac{b - b_i}{h}\right) \quad (6)$$

where N is a neighborhood selection function that selects a subset of the b_i training examples from the training dataset D based on the distance of their corresponding $\phi(s_i)$ values to the given query point $\phi(s)$. We note that there exists an efficient method for approximating samples from this distribution if we use a kernel that decreases exponentially as distance increases, with small bandwidth, and that our training data is sampled from the same distribution as the testing data. We use k -Nearest Neighbors to obtain the k tuples whose $\phi(s_i)$ values are closest to $\phi(s)$, which will have the greatest effect on \hat{f} , and then sample uniformly from this set. In addition to being faster than rejection sampling under (6), it has the added benefit of not requiring a bandwidth parameter, the tuning of which is a problem of considerable complexity in itself. By contrast, the k parameter is a positive integer that is relatively straightforward to tune using cross validation.

In the next section we will use this approach to construct predictors for both stochastic and deterministic behaviors, and we will compare their performance with fixed-estimate predictors.

Experiments

To illustrate the difficulties in using fixed-estimate predictors and how sampling predictors can address these issues, we hand constructed simulations of a group of agents with homogeneous behaviors, both deterministic and stochastic, collected tracks of the agents performing these behaviors, and then compared the performance of two fixed-estimate predictors, linear regression and k NN regression, with a sampling predictor, neighborhood sampling k NN, in terms of both typical measures of prediction accuracy and similarity in terms of the distribution of behaviors. In brief we find that sampling based predictors perform on-par with or slightly worse than fixed-estimate predictors in terms of RMSE and trajectory error, but significantly outperform fixed-estimate predictors in terms of matching the distribution of behaviors.

Constructing an executable model using a trained predictor is straightforward: the output of the executable model is just the predicted output for the given feature values, which are in turn computed from the given environmental state. For each of these experiments we built such an executable model using each of our predictors and then ran these models in

¹It may even be advantageous for the fish to exhibit higher variance of behavior if we consider the context of fleeing in response to a perceived threat, since any highly predictable behavior could be exploited by a predator.

simulation. This allowed us to compute not just RMSE by comparing the predicted output on a hold-out set, but also end-point error by initializing the simulation at starting configurations from sequences in the hold-out set, and calculating the difference between the final pose in the simulation and the final pose in the withheld sequence, and a measure of the difference between the distribution of behavior generated by the executable model and the observed distribution of behavior generated by the original agent.

Motivated by our earlier example, and our overarching goal of learning complex multiagent behaviors of physical systems, we focus on two models of schooling behavior for a small number of fish confined to a shallow tank.

Deterministic Behavior

We base our deterministic model on the well known Boids (Reynolds 1987) model, which produces realistic flocking and schooling behaviors with each agent following a set of local rules which correspond to a linear combination of feature vectors. We choose a linear model specifically to focus on the differences between deterministic and stochastic behaviors, and to show that any limitations of a linear regression based predictor are not due to the expressiveness of the model or the complexity of the underlying behavior.

In this model, the behavior space ranges over desired velocities $b = (\dot{x}, \dot{y}, \dot{\theta})$, and the features correspond to vectors that capture characteristics of the school in a local area near the agent, which are summarized in Table 1 in the second column. These vectors are distance-weighted averages with the following form:

$$\phi_i(s) = \frac{1}{n} \sum_{j \neq i} \exp \left\{ \frac{d_j^2(s)}{2\sigma_i^2} \right\} v_{i,j}(s) \quad (7)$$

where n is the number of agents in the school, $d_j(s)$ is the distance to agent j , and $v_{i,j}(s)$ is a unit directional vector unique to ϕ_i . We use four ϕ_i : a *separation* component, an *orientation* component, a *cohesion* component, and an *obstacle* avoidance component. By choosing appropriate σ_i , we can tune the sensitivity of each component so that they respond to agents or obstacles at different ranges. The range of each σ_i is reported in the second column of Table 1.

The actual behavior f is defined as a simple linear combination of these features:

$$f(\phi(s)) = \langle W, \phi(s) \rangle \quad (8)$$

where $\phi(s)$ is the concatenation of the four feature vectors described above, along with a constant term: $\phi(s) = [\phi_{\text{sep}}(s), \phi_{\text{ori}}(s), \phi_{\text{coh}}(s), \phi_{\text{obs}}(s), 1]$, and W is described in Table 2. Both W and each σ_i were hand tuned to produce reasonable looking schooling behavior where agents avoided collisions with each other and obstacles and preferred to stay grouped and aligned when possible.

Given this model, we train a predictor \hat{f} with a set of observations collected from a simulation running the model. These observations are recorded directly by the simulator, and since all of the features can be computed from the pose (x, y, θ) of all the agents at any given time, they can stand

in for tracking information gathered from video of live animals, as we show in a subsequent section.

Training a linear regression model from these samples works well, unsurprisingly, and examining the learned weights shows that we’ve recovered the parameters of the generating model quite closely (Table 2, right). Similarly, k NN with neighborhood averaging (which we call k NN-Reg from here on) performs well, while k NN with neighborhood sampling (k NN-Sample) performs slightly worse. The first three rows in Table 3 show the performance of all three methods in terms of average prediction error (RMSE) and average end-point error in the first two columns. For this simple model, this result is as expected, since the benefit of sampling from the predicted distribution doesn’t come in to play for deterministic models and comes at the cost of increasing the variance of the error of the prediction.

Stochastic Behavior

In order to illustrate the benefit of sampling based predictors we introduce a slight variation of the model in the previous section, which incorporates an element of randomness into the behavior. We start with the same ϕ_i , the same linear combination of these weights, but we add a random variable sampled from an exponential distribution with mean $\beta = 0.05$ to the desired forward velocity (\dot{x}) . Since this random variable is non-negative, we adjust the bias term in W down to 0.00625 so that the mean forward velocity would remain approximately the same. This generates a behavior similar at a high level to what was generated before, but where the agents exhibit some unpredictable variation in forward speed. Since this variation is non-symmetric, we expect it to cause the fixed-estimate predictors Lin-Reg and k NN-Reg to produce behavior distributions that are significantly different from the generating behavior.

To highlight the effect of this inherent unpredictability, we introduce a third performance metric in addition to RMSE and end-point error, namely the Kullback-Leibler divergence which we apply to histograms of forward velocity under the generating behavior and the predictor behavior. The third column in Table 3 represents the degree of difference between the two distributions where a K-L divergence of zero means the two distributions match exactly. Figure 2 shows the histograms graphically for a more qualitative comparison. As these figures show, the sampling based predictor produces a distribution of behavior that more closely matches the generating behavior, while the fixed estimate predictors do not, even though the three predictors perform similarly in terms of end-point and RMSE error.

Live Fish Behavior

In the previous two experiments, we generate the training data from a simulation without noise, and for which we know the ground truth parameters. This setup is useful for illustrative purposes, but unrealistic. In this section we move to a more interesting domain that motivated our choice of behaviors and examples in previous sections. Given pose information (x, y, θ) from video of a small school of

Table 1: Description of features σ_i .

Component	$v_{i,j}$	σ_i
$v_{sep,j}$	Away from agent j ($-(x_j - x)$)	Near, 1-2 body lengths (0.1)
$v_{ori,j}$	Heading of agent j (θ_j)	Somewhat near, 2-3 body lengths (0.2)
$v_{coh,j}$	Towards agent j ($(x_j - x)$)	Majority of school (1.0)
$v_{sep,j}$	Away from obstacle j	Short, within 1 body length (0.05)

Table 2: Feature weights W for the deterministic behavior. *Left* are the hand-tuned parameters for the generating behavior, and *right* are the parameters recovered by linear regression. The recovered parameters nearly match the generating parameters, with the Frobenius norm of their difference being less than 0.894.

Generating $f(\phi(s))$	\dot{x}	\dot{y}	$\dot{\theta}$	Recovered $\hat{f}(\phi(s))$	\dot{x}	\dot{y}	$\dot{\theta}$
ϕ_{sep_x}	-1.0	0.0	0.0	ϕ_{sep_x}	-0.9998	0.0	0.0015
ϕ_{sep_y}	0.0	0.0	-20.0	ϕ_{sep_y}	-5.1436×10^{-4}	0.0	-19.9995
ϕ_{ori_x}	0.0	0.0	0.0	ϕ_{ori_x}	-1.3071×10^{-5}	0.0	-9.3309×10^{-6}
ϕ_{ori_y}	0.0	0.0	0.1	ϕ_{ori_y}	1.1432×10^{-7}	0.0	0.0998
ϕ_{coh_x}	0.0	0.0	0.0	ϕ_{coh_x}	-3.3333×10^{-6}	0.0	2.8271×10^{-4}
ϕ_{coh_y}	0.0	0.0	0.8	ϕ_{coh_y}	3.1213×10^{-5}	0.0	0.8004
ϕ_{obs_x}	-1.0	0.0	0.0	ϕ_{obs_x}	-0.9753	0.0	-0.3991
ϕ_{obs_y}	0.0	0.0	-40.0	ϕ_{obs_y}	4.6335×10^{-4}	0.0	-38.7059
bias	0.0125	0.0	0.0	bias	0.01250	0.0	-3.5565×10^{-5}

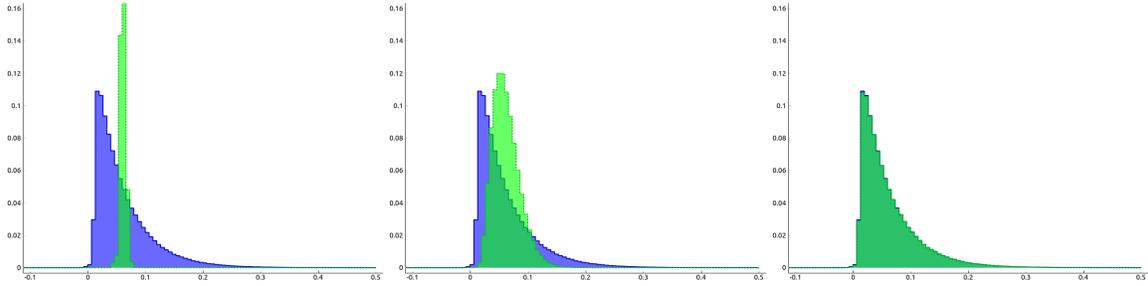


Figure 2: Comparison of predicted x-velocity distribution for (left) linear regression, (center) k NN-Reg, and (right) k NN-Sample in green (dotted line), versus actual x-velocity distribution in blue (solid line) for the stochastic simulated behavior. Note that while both of the fixed-estimate predictors produce behavior that has a similar *mean*, the shape of the distribution does not match the training data while k NN-sample does.

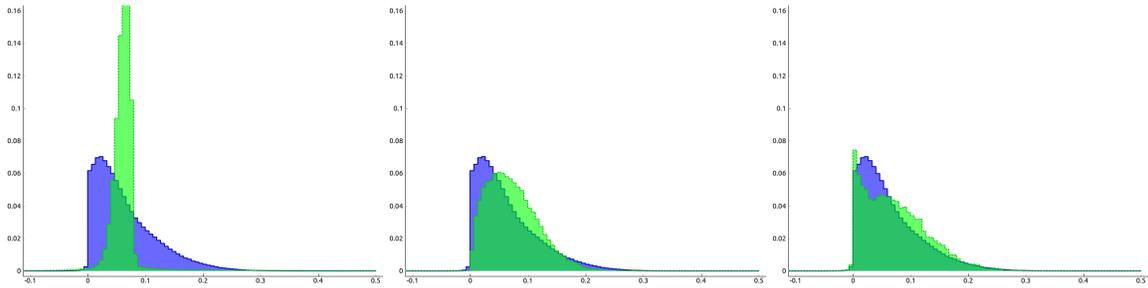


Figure 3: Comparison of predicted x-velocity distribution for (left) linear regression, (center) k NN-Reg, and (right) k NN-Sample in green (dotted line), versus actual x-velocity distribution in blue (solid line) for the data collected from real fish. Similar to the case with the stochastic simulated behavior, linear regression and k NN-Reg both do a better job of matching the mean value than the shape of the training distribution.

Table 3: Performance of linear regression, k NN-Reg, and k NN-Sample. Reported values are mean and standard deviation for 10-fold cross validation. Lower is better for all three error metrics.

Deterministic			
Predictor	RMSE	end-point	K-L divergence
Lin-Reg	0.0356 ± 0.0050	0.0023 ± 0.0006	—
k NN-Reg	5.9675 ± 0.3665	0.0070 ± 0.0021	—
k NN-Sample	4.4097 ± 0.3328	0.0069 ± 0.0020	—
Stochastic			
Predictor	RMSE	end-point	K-L divergence
Lin-Reg	6.1099 ± 0.0361	0.0067 ± 0.0009	482.6036 ± 0.7310
k NN-Reg	13.5425 ± 0.4549	0.0150 ± 0.0017	51.1153 ± 12.3740
k NN Sample	15.1933 ± 0.4574	0.0154 ± 0.0014	0.0465 ± 0.0282
Live Fish			
Predictor	RMSE	end-point	K-L divergence
Lin-Reg	308.6853 ± 22.4447	0.2164 ± 0.0148	1.5865 ± 0.1894
k NN-Reg	375.9885 ± 24.2010	0.1877 ± 0.0156	1.0020 ± 0.5021
k NN Sample	570.9620 ± 29.3125	0.2136 ± 0.0160	0.2957 ± 0.2497

Notemigonus crysoleucas in a shallow tank² we can apply the same prediction methods and performance metrics to create executable models of real fish schooling behavior. The selection of ϕ_i for this problem is by no means straightforward, as discussed in (Hrotenok and Balch 2014), but it is beyond the scope of this paper to perform a full ablative analysis of all reasonable features. So instead we use those ϕ_i described earlier, as well as a few additional features that we found improved the performance of *all* predictors, with the intuition that these features provide reasonable coverage of the information that is relevant to schooling behavior for an individual agent. To the set of ϕ_i already described we add the mean (ϕ_μ), variance (ϕ_σ), and maximum magnitude (ϕ_{\max}) of the forward velocity of all other fish in the school. For both k NN methods, the selection of k was handled by 10-fold cross validation on a parameter sweep from $k \in \{1 \dots 100\}$. We found that for both methods, the maximum performance was found at $k = 5$. The entire dataset was taken from a one hour video shot at 30 frames per second, with 30 fish visible in each frame, roughly 1.4 million data points after filtering.

The performance of linear regression, k NN-Reg, and k NN-Sample are given numerically in Table 3, and graphically in Figure 3. Note that the distribution of forward velocity for the real fish is an asymmetric Poisson-like curve, which is what motivated the stochastic model from the previous experiment. The form of the true behavior f for real fish is not known, and we have no strong evidence that it is linear, so it is not surprising that linear regression does not outperform non-parametric methods. As before, the increased variance in the expected difference between prediction and actual behavior for k NN-Sample leads to larger RMSE and end-point error than the fixed-estimate predictors, but a closer match between the distribution of predicted and actual behavior.

²Data was originally collected in (Katz et al. 2011), which details the specifics of the environment, and the tracking method used.

Conclusion

We’ve identified that stochastic agent behavior represents a challenging but important subset of agent behavior. By considering behaviors from a probabilistic viewpoint, we’ve explained why and under what circumstances using fixed-estimate predictors to construct executable models can lead to poor performance. This probabilistic viewpoint suggests a new class of predictors that match the distribution of behavior, and we’ve shown how to implement one such predictor in a simple and efficient way. Using this predictor, we conducted two simulation experiments to empirically validate our method, and to highlight the differences between deterministic and stochastic behaviors in terms of each predictor’s performance. In our third experiment, we’ve shown that our sampling predictor outperforms two fixed-estimate predictors in learning the schooling behavior of real fish from tracking data, suggesting that the behavior of these animals is in fact stochastic.

While we’ve used fairly simple learning methods in this work for the purposes of illustration, we are working to apply these same principles to more sophisticated methods due to recent developments in time-series prediction. Some preliminary results using methods based on work from Venktraman, Hebert, and Bagnell (2015) show improvement over the base methods presented here, and we are working to develop a more fully grounded theory to explain this effect. In this work, we use a measure of distribution similarity mainly to highlight the challenges unique to constructing executable models of inherently stochastic behaviors, but we believe that using distribution similarity as a measure of performance has more general applicability for the problem of constructing executable models. Specifically, distribution similarity provides a framework that allows us to assess high level qualitative similarity in a quantitative way. Using this measure of performance as the underlying cost function for novel machine learning algorithms is an interesting direction for future work that we are currently exploring.

References

- Branson, K.; Robie, A. A.; Bender, J.; Perona, P.; and Dickinson, M. H. 2009. High-throughput ethomics in large groups of drosophila. *Nature methods* 6(6):451–457.
- Dankert, H.; Wang, L.; Hoopfer, E. D.; Anderson, D. J.; and Perona, P. 2009. Automated monitoring and analysis of social behavior in drosophila. *Nature methods* 6(4):297–303.
- Egerstedt, M.; Balch, T.; Dellaert, F.; Delmotte, F.; and Khan, Z. 2005. What are the ants doing? vision-based tracking and reconstruction of control programs. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005)*, 18–22.
- Hemelrijk, C.; Wantia, J.; and Gyga, L. 2005. The construction of dominance order: comparing performance of five methods using an individual-based model. *Behaviour* 142(8):1037–1058.
- Hemelrijk, C. K. 2000. Towards the integration of social dominance and spatial structure. *Animal Behaviour* 59(5):1035–1048.
- Hrotenok, B., and Balch, T. 2013. Learning schooling behavior from observation. *Advances in Artificial Life, ECAL 12*:686–691.
- Hrotenok, B., and Balch, T. 2014. Assessing learned models of fish schooling behavior. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, AAMAS*, 1435–1436. International Foundation for Autonomous Agents and Multiagent Systems.
- Inamura, T.; Tushima, I.; Tanie, H.; and Nakamura, Y. 2004. Embodied symbol emergence based on mimesis theory. *The International Journal of Robotics Research* 23(4-5):363–377.
- 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.
- Kulić, D.; Takano, W.; and Nakamura, Y. 2008. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research* 27(7):761–784.
- List, C.; Elsholtz, C.; and Seeley, T. D. 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*, 944–949. Pittsburgh, Pennsylvania: AAAI Press.
- Pratt, S. C.; Sumpter, D. J.; Mallon, E. B.; and Franks, N. R. 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, 25–34.
- Sugiura, K., and Iwahashi, N. 2008. Motion recognition and generation by combining reference-point-dependent probabilistic models. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 852–857.
- Tunstrøm, K.; Katz, Y.; Ioannou, C. C.; Huepe, C.; Lutz, M. J.; and Couzin, I. D. 2013. Collective states, multistability and transitional behavior in schooling fish. *PLoS Comput Biol* 9(2):e1002915.
- Venkatraman, A.; Hebert, M.; and Bagnell, J. A. 2015. Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.