Learning governing equations of interacting particle systems using Gaussian process regression

Jinchao Feng
Johns Hopkins University
jfeng34@jhu.edu

Yunxiang Ren
Harvard University
yren@g.harvard.edu

Sui Tang
University of California, Santa Barbara
suitang@ucsb.edu

Abstract

Interacting particle or agent systems that display a rich variety of collection motions are ubiquitous in science and engineering. The fundamental and challenging goals are to infer individual interaction rules that yield collective behaviors and establish the governing equations. In this paper, we study the data-driven discovery of second-order interacting particle systems with distance-based interaction laws, which are known to have the capability to reproduce a rich variety of collective patterns. We propose a learning approach that models the latent interaction function as a Gaussian process, which can simultaneously fulfill two inference goals: one is the nonparametric inference of interaction function with the pointwise uncertainty quantification, and the other one is the inference of unknown parameters in the non-collective forces of the system. We test the learning approach on Dorsogma model and numerical results demonstrate the effectiveness.

1 Introduction

Interacting particle/agent systems are ubiquitous in science and engineering. The individual interactions among agents produce a rich variety of collective motions with visually compelling patterns, such as crystallization of particles, clustering of opinions on social events, and coordinated movements of ants, fish, birds, and cars. For these interacting particle/agent systems arising from numerous fields, it is a central subject to investigate the interaction laws and derive their governing equations.

There have been tremendous research efforts in using classical physical laws to model the collective dynamics. A common belief in scientific research is that the complicated collective behaviors are indeed consequences of rather simple interactions, for instances, the ones based on pairwise distances among particles/agents. Based on these ideas, one may write down a second-order system for N interacting particles $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ in $\mathbb{R}^d$ as follows:

$$m_i \ddot{\boldsymbol{x}}_i(t) = F(\boldsymbol{x}_i(t), \dot{\boldsymbol{x}}_i(t), \boldsymbol{\alpha}) + \sum_{i'=1}^{N} \frac{1}{N}\Big[\phi(\|\boldsymbol{x}_{i'}(t) - \boldsymbol{x}_i(t)\|)(\boldsymbol{x}_{i'}(t) - \boldsymbol{x}_i(t))\Big], \quad i = 1, \cdots, N. \tag{1}$$

The form of the above governing equation is indeed derived from Newton's second law: m_i is the mass of the agent i; $\ddot{\boldsymbol{x}}_i$ is the acceleration; $\dot{\boldsymbol{x}}_i$ is the velocity; F is a parametric function of position and velocities, modelling frictions of the particles with the environment, and the scalar parameters α describes their strength; the interaction force is the derivative of a potential energy function depending on pairwise distances:

$$\mathcal{U}(\boldsymbol{X}(t)) := \sum_{i,i'=1,1}^{N,N} \frac{1}{2N}\Phi(\|\boldsymbol{x}_{i'}(t) - \boldsymbol{x}_i(t)\|), \quad \Phi'(r) = \phi(r)r. \tag{2}$$

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

In other words, the interactions follow the rule of minimizing the energy function so that the particles will converge to the steady states that are local minimizers of the energy function.

For many systems arise in biology, ecology and social science, a grand challenging task is to find ϕ since there is no canonical choice. Remarkable progress has been made on qualitative analysis of (1) [1–5], which show that the solutions to (1) can reproduce a rich variety of macroscopic patterns when time goes to infinity, similar to those observed in practice.

In recent years, due to the rapid advancements in digital imaging and high-resolution lightweight GPS devices, the individual trajectory datasets of interacting particle systems are becoming increasingly available. This inspired the research of fitting trajectory data into governing equation of form (1) for scientific discovery [6, 2, 7, 8]. However, the estimation relied heavily on the experts' domain knowledge: one needs to select a small parametric family and then perform recovery with calibration by modellers. The goal is to explain the data qualitatively. Now days, machine learning methods have achieved great empirical success in many applications such as healthcare and computer vision, demonstrating the impressive power of extracting information from data. However, their application in data-driven modelling of dynamical systems is still in infancy. In particular, the machine learning literature towards the data-driven discovery of interacting particle system is still scarce.

In this paper, we consider the inverse problem and investigate whether the interaction kernel ϕ and α can be accurately estimated from the trajectory data generated by the system (1) by using machine learning methods. Our study will shed light on applying (1) for scientific discovery, by proposing a detailed methodology and providing the physical interpretation of estimators.

Problem Statement Without loss of generality, we assume that the masses of agents are the same and have been normalized to be 1. For the agent i, we denote its position and velocity at time t by $\boldsymbol{x}_i(t) \in \mathbb{R}^d$ and $\boldsymbol{v}_i(t) := \dot{\boldsymbol{x}}_i(t) \in \mathbb{R}^d$. Let $\boldsymbol{X}(t)$ be the $[\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \cdots, \boldsymbol{x}_N(t)] \in \mathbb{R}^{dN}$ and $\boldsymbol{V}(t) = \dot{\boldsymbol{X}}(t) \in \mathbb{R}^{dN}$ be defined in the similar way. Then we can rewrite the system (1)in a compact form:

$$\boldsymbol{Z}(t) = F(\boldsymbol{Y}(t), \boldsymbol{\alpha}) + \mathbf{f}_\phi(\boldsymbol{X}(t)), \tag{3}$$

where $\boldsymbol{Y}(t) := [\boldsymbol{X}(t), \boldsymbol{V}(t)]^T \in \mathbb{R}^{2dN}$ represents the state variable for the system, $\boldsymbol{Z}(t) = \dot{\boldsymbol{V}}(t)$, $\mathbf{f}_\phi(\boldsymbol{X}(t)) : \mathbb{R}^{dN} \to \mathbb{R}^{dN}$ represents the distance based interactions governed by the interaction kernel ϕ as in (1). We fix L time stamps with $0 = t_1 < t_2 < \cdots t_L = T$ on $[0, T]$ and obtain the trajectory data $\{\boldsymbol{Y}(t_l), \boldsymbol{Z}_{\sigma^2}(t_l) : 1 \le l \le L\}$ as one training instance, where σ^2 denotes the variance of additive Gaussian noise specified below. Furthermore, we hold the following two assumptions on training data of M training instances:

1. The M initial conditions $\{\boldsymbol{Y}^{(m)}(0) : 1 \le m \le M\}$ are drawn randomly from a probability measure $\boldsymbol{\mu}_0 = [\mu_0^{\boldsymbol{X}}, \mu_0^{\dot{\boldsymbol{X}}}]^T$ on $\mathbb{R}^{2dN}$.
2. The accelerations $\{\boldsymbol{Z}^{(m)}(t_l) : 1 \le l \le N, 1 \le m \le M\}$ are observed with i.i.d additive Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_{dN \times dN})$, so that the data is denoted by $\boldsymbol{Z}^{(m)}_{\sigma^2}(t_l)$.

In practical situations, there is often little information about the analytical form of the interaction kernel ϕ and we may have scarce noisy observation data. It will be very helpful to consider non-parametric inference of ϕ with uncertainty quantification of estimators, which quantifies the reliability of estimators. The GPR [9–11] is a non-parametric Bayesian machine learning technique with built-in quantification of uncertainty encoded in the posterior variances of estimators, which had many successful applications. In this paper, We propose a learning approach based on *Gaussian process regression* (GPR) to learn the interaction kernel ϕ and the scalar parameters $\boldsymbol{\alpha}$.

2 Methodology

We first model the interaction kernel function ϕ as a Gaussian process [9–11], namely, $\phi \sim \mathcal{GP}(0, K_\theta(r, r'))$, with mean zero and covariance kernel function K_θ, depending on the hyper-parameters θ. The Gaussian prior incorporates the prior knowledge about ϕ before seeing the observational data.

Given $\mathbb{Y} = [\boldsymbol{Y}^{(1,1)}, \ldots, \boldsymbol{Y}^{(M,L)}]^T$, it follows from Equation (3) and the properties of GPs that $\mathbb{Z} := [\boldsymbol{Z}^{(1,1)}_{\sigma^2}, \ldots, \boldsymbol{Z}^{(M,L)}_{\sigma^2}]^T$ follows a multivariate Gaussian distribution with the mean vector $F^{\boldsymbol{v}}_{\boldsymbol{\alpha}}(\mathbb{Y}) =$

$\mathrm{Vec}((F^{\boldsymbol{v}}_{\boldsymbol{\alpha}}(\boldsymbol{Y}^{(m,l)}))^{M,L}_{m=1,l=1}) \in \mathbb{R}^{dNML}$, and the covariance matrix $(K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X};\theta)+\sigma^2 I_{dNML}))$, where $K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X};\theta) = \big(\mathrm{Cov}(\mathbf{f}_\phi(\boldsymbol{X}^{(i,j)}),\mathbf{f}_\phi(\boldsymbol{X}^{(i',j')}))\big)^{M,M,L,L}_{i,i',j,j'=1,1,1,1}$ with (i,j)th block computed by

$$\mathrm{Cov}([\mathbf{f}_\phi(\boldsymbol{X})]_i,[\mathbf{f}_\phi(\boldsymbol{X}')]_j) = \frac{1}{N^2}\sum_{k\neq i,k'\neq j}\big(K_\theta(r^{\boldsymbol{x}}_{ik},r^{\boldsymbol{x}'}_{jk'})\boldsymbol{r}^{\boldsymbol{x}}_{ik}\boldsymbol{r}^{\boldsymbol{x}'\,T}_{jk'}\big), \tag{4}$$

where $r^{\boldsymbol{x}}_{ik} := \|\boldsymbol{r}^{\boldsymbol{x}}_{ik}\| := \|\boldsymbol{X}(t)_k - \boldsymbol{X}(t)_i\|$.

Then we train the hyper-parameters $\boldsymbol{\alpha}$ and θ by maximizing the probability of the observational data, which is equivalent to minimize the negative log marginal likelihood (see Chapter 4 in [9])

$$\begin{aligned}-\log P(\mathbb{Z}|\mathbb{Y},\boldsymbol{\alpha},\theta,\sigma^2) &= \frac{1}{2}(\mathbb{Z}-F^{\boldsymbol{v}}_{\boldsymbol{\alpha}}(\mathbb{Y}))^T(K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X};\theta)+\sigma^2 I)^{-1}(\mathbb{Z}-F^{\boldsymbol{v}}_{\boldsymbol{\alpha}}(\mathbb{Y}))\\ &\quad+\frac{1}{2}\log|K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X};\theta)+\sigma^2 I|+\frac{dNML}{2}\log 2\pi.\end{aligned} \tag{5}$$

To solve for the hyper-parameters $(\boldsymbol{\alpha},\theta)$, we can apply a gradient based method, Quasi-Newton optimizer L-BFGS [12], to minimize the negative log marginal likelihood. The marginal likelihood induces an automatic trade-off between data-fit and model complexity [13]. This flexible training procedure distinguishes Gaussian process from other kernel-based methods [14–16] and regularization based approaches [17–19].

After the training procedure, we obtain updated priors on the interaction kernel functions. We show how to predict the value $\phi(r^*)$ using the mean of its posterior distribution. Note that

$$\begin{bmatrix}\mathbf{f}_\phi(\mathbb{X})\\ \phi(r^*)\end{bmatrix} \sim \mathcal{N}\left(0,\begin{bmatrix}K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X}) & K_{\mathbf{f}_\phi,\phi}(\mathbb{X},r^*)\\ K_{\phi,\mathbf{f}_\phi}(r^*,\mathbb{X}) & K_\theta(r^*,r^*)\end{bmatrix}\right), \tag{6}$$

where $K_{\mathbf{f}_\phi,\phi}(\mathbb{X},r^*) = K_{\phi,\mathbf{f}_\phi}(r^*,\mathbb{X})^T$ denotes the covariance matrix between $\mathbf{f}_\phi(\mathbb{X})$ and $\phi(r^*)$. Conditioning on $\mathbf{f}_\phi(\mathbb{X})$, we obtain that

$$p(\phi(r^*)|\mathbb{Y},\mathbb{Z},r^*) \sim \mathcal{N}(\bar{\phi^*},\mathrm{Var}(\phi^*)), \tag{7}$$

where the mean and variance of the ϕ^* is given by

$$\bar{\phi^*} = K_{\phi,\mathbf{f}_\phi}(r^*,\mathbb{X})(K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X})+\sigma^2 I)^{-1}(\mathbb{Z}-F^{\boldsymbol{v}}_{\boldsymbol{\alpha}}(\mathbb{Y})), \tag{8}$$

$$\mathrm{Var}(\phi^*) = K_\theta(r^*,r^*) - K_{\phi,\mathbf{f}_\phi}(r^*,\mathbb{X})(K_{\mathbf{f}_\phi}(\mathbb{X},\mathbb{X})+\sigma^2 I)^{-1}K_{\mathbf{f}_\phi,\phi}(\mathbb{X},r^*). \tag{9}$$

The posterior variances $\mathrm{Var}(\phi^*)$ can be used as a good indicator for the uncertainty of the estimation $\hat{\phi} := \bar{\phi^*}$ based on our Bayesian approach. Moreover, using the estimated parameters $\hat{\boldsymbol{\alpha}}$ and interaction kernels $\hat{\phi}$, we can predict the dynamics based on the equations $\hat{\boldsymbol{Z}}(t) = F^{\boldsymbol{v}}_{\hat{\boldsymbol{\alpha}}}(\boldsymbol{Y}(t)) + \mathbf{f}_{\hat{\phi}}(\boldsymbol{X}(t))$.

3 Experiments

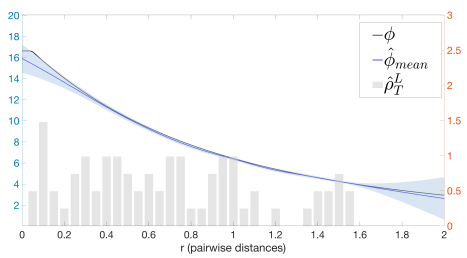
In this section, we report the empirical performance of our proposed approach with the Matérn covariance function in Dorsogma model [1, 5, 20], which describes the motion of N self-propelled particles powered by biological or mechanical motors, with $F(\boldsymbol{x}_i,\dot{\boldsymbol{x}}_i,\boldsymbol{\alpha}) = (\gamma-\beta|\dot{\boldsymbol{x}}_i|^2)\dot{\boldsymbol{x}}_i$ and $\phi(r) = \left[-e^{-2r}+e^{-\frac{r}{4}}\right]/r$, which is an instance of Morse potential. It is shown in [1, 20] that such system can produce a rich variety of dynamics such as double/single milling, swarming and ring.

The system studied in our experiment is a 10-agent system with $m_i = 1$ for all i, $\boldsymbol{\alpha} = (\gamma,\beta) = (1.5, 0.5)$ at the time interval $t \in [0,5]$, and we test our method on both noise free data and noisy data with the noise level $\sigma = 0.05, 0.1$. For each training instance, we generate the random initial condition $\boldsymbol{x}(0)$ from $[-0.5, 0.5]^2$ uniformly and set $\boldsymbol{v}(0) = 0$. Fix $M = 3$ and $L = 3$. The training data has M instances with L timestamps even spaced over the time interval $[0,5]$, while the test data is determined by another randomly generated M initial conditions.

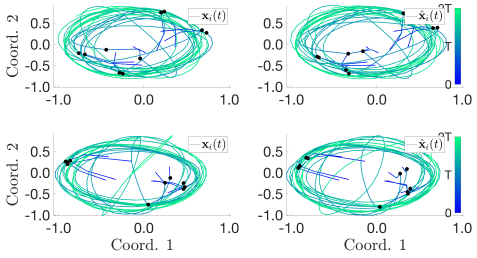
We obtained the estimators for the hyper-parameters from our learning approach and summarize the results of each model in Table 1. And in Figure 1, we show the comparison between the learned kernel and the true kernel, and their corresponding predictions for the model with $\sigma = 0.1$.

Table 1: Means and standard deviations of estimations for parameters σ and $\boldsymbol{\alpha}$, and predicted errors for ϕ on $[0, 1.76]$ in fishing milling dynamics with noise free data ($\sigma = 0$) and noisy data ($\sigma = 0.05, 0.1$) when $M = 3, L = 3$

Models	$\sigma = 0$	$\sigma = 0.05$	$\sigma = 0.1$
$\hat{\sigma}$	NA	$0.0495 \pm 2.8 \cdot 10^{-3}$	$0.0991 \pm 5.6 \cdot 10^{-3}$
$\hat{\gamma}$ (true γ = 1.5)	$1.4998 \pm 1.7 \cdot 10^{-4}$	$1.5030 \pm 1.3 \cdot 10^{-2}$	$1.5057 \pm 2.7 \cdot 10^{-2}$
$\hat{\beta}$ (ture β = 0.5)	$0.4999 \pm 7.3 \cdot 10^{-5}$	$0.5019 \pm 5.6 \cdot 10^{-3}$	$0.5036 \pm 1.1 \cdot 10^{-2}$
Relative L^∞-error of $\hat{\phi}$	$2.5 \cdot 10^{-2} \pm 3.4 \cdot 10^{-3}$	$4.1 \cdot 10^{-2} \pm 1.7 \cdot 10^{-2}$	$6.5 \cdot 10^{-2} \pm 3.2 \cdot 10^{-2}$
Errors on predictions for training $\mathbb{X}$ on $[0, 5]$	$1.4 \cdot 10^{-2} \pm 9.1 \cdot 10^{-3}$	$1.3 \cdot 10^{-1} \pm 4.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-1} \pm 1.0 \cdot 10^{-1}$
Errors on predictions for training $\mathbb{X}$ on $[5, 10]$	$4.4 \cdot 10^{-2} \pm 3.5 \cdot 10^{-2}$	$3.4 \cdot 10^{-1} \pm 1.8 \cdot 10^{-1}$	$7.0 \cdot 10^{-1} \pm 3.7 \cdot 10^{-1}$
Errors on predictions for testing $\mathbb{X}$ on $[0, 5]$	$1.3 \cdot 10^{-2} \pm 9.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-1} \pm 4.6 \cdot 10^{-2}$	$2.2 \cdot 10^{-1} \pm 1.1 \cdot 10^{-1}$
Errors on predictions for testing $\mathbb{X}$ on $[5, 10]$	$4.8 \cdot 10^{-2} \pm 3.2 \cdot 10^{-2}$	$3.2 \cdot 10^{-1} \pm 1.1 \cdot 10^{-1}$	$5.8 \cdot 10^{-1} \pm 2.4 \cdot 10^{-1}$



(a) learned kernel vs true kernel



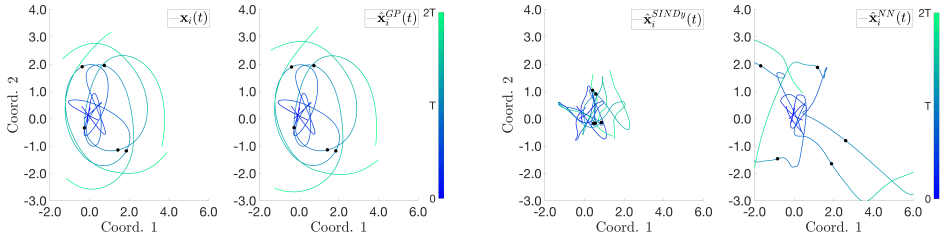
(b) trajectory prediction

Figure 1: Learning a fishing milling system (dim=20) using the Matérn kernel with noisy data, $\sigma = 0.1$. (a): predictive mean $\hat{\phi}_{mean}$ of the true kernel ϕ, and two-standard-deviation band (light blue color) around the mean. The grey bars represent the empirical density of pairwise distances of agents (computed from training data), on which ϕ is being learned. Our estimator enjoys extrapolation property outside its support. (b): the trajectory using true $\boldsymbol{\alpha}$ and ϕ (left) versus prediction using $\hat{\boldsymbol{\alpha}}$ and $\hat{\phi}_{mean}$ (right) for two sets of initial conditions.

Discussion In all cases, the estimation error for $\boldsymbol{\alpha}$ are very small (at most $O(10^{-2})$), and our estimators can produce faithful approximations to the true kernel for both noise free and noisy data as shown in Figure 1 (a). The uncertainty region in the area covered by the density of pairwise distance (see grey bar in Figure 1 (a)) is very small. When noise increased, the error around $r = 0$ is slightly larger, which is due to the fact that $\phi(r)$ is weighted by $\vec{r}$ in the model (1)(so the information of $\phi(0)$ is lost) and the area near 0 is not in the support of density of pairwise distance. Therefore, we need more data in the noisy case so that the pairwise distance can cover the part near 0. However, the true interaction kernel $\phi(r)$ is fully covered in the compact uncertainty region which we constructed using the posterior variances. The trajectory prediction errors can go up to $O(10^{-1})$ for few agents with the presence of a relatively large noise, but our estimators provided faithful predictions to most of the agents in the system, and the milling pattern as shown in Figure 1 (b). We also test our approach on other systems that exhibit clustering and flocking behaviour and the results demonstrate the effectiveness.

Baseline comparisons We perform comparisons with approaches that learn the right handside function of (3) directly from trajectory data: the first one is SINDy [21], which aims at finding a sparse representation for each row of governing equations in a (typically large) dictionary; the second

one is regression using the feed forward neural networks, for which we use the MATLAB® 2021a Deep Learning Toolbox™. To evaluate the performance, we compare the trajectory prediction errors of the estimators for a 5-agent system with same parameters and $\{M, L, \sigma\} = \{1, 9, 0.1\}$.



(a) True versus Our Model in FM dynamics

(b) Comparison with SINDy and FNN in FM dynamics

Figure 2: Learning FM dynamics from training data $\{N, M, L, \sigma\} = \{5, 1, 9, 0.1\}$. The true trajectory versus the prediction from our GP model (left), the SINDy model and the FNN model trajectories (right).

Baseline comparison tests illustrate the importance of exploiting the structure of the governing equation. SINDy treats each row individually and did not take the nonlocal interaction structure into account. FNN treats the right hand side function of governing equation as a high dimensional vector valued function. We see in the Figure 2 that our approach has the best performance with the limited training data. The specific errors are summarized in Table 2.

Table 2: Baseline comparison. The relative trajectory prediction errors.

Approach	Training time interval $[0, 5]$	Future time interval $[5, 10]$
GPs	$\mathbf{3.6 \cdot 10^{-3} \pm 2.5 \cdot 10^{-3}}$	$\mathbf{2.4 \cdot 10^{-1} \pm 3.1 \cdot 10^{-1}}$
SINDy	$9.4 \cdot 10^{-1} \pm 3.8 \cdot 10^{-1}$	$1.2 \cdot 10^{0} \pm 4.7 \cdot 10^{-1}$
FNN	$2.2 \cdot 10^{0} \pm 1.3 \cdot 10^{0}$	$3.1 \cdot 10^{0} \pm 1.7 \cdot 10^{0}$

4 Future work

In our ongoing work, we connect our learning problem with the statistical inverse problem and provide a systematic learning theory to provide a quantitative analysis of the estimators. Another directions include the improvement of the computational efficiency of the current approach to deal with abundant trajectory data, since a well-known computational limitation of GPs is that inverting dense covariance matrices scales cubically with the number of training data.

References

[1] Maria R D'Orsogna, Yao-Li Chuang, Andrea L Bertozzi, and Lincoln S Chayes. Self-propelled particles with soft-core interactions: patterns, stability, and collapse. *Physical review letters*, 96 (10):104302, 2006.

[2] Ryan Lukeman, Yue-Xian Li, and Leah Edelstein-Keshet. Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences*, 107(28):12576–12580, 2010.

[3] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[4] Fabian Baumann, Igor M Sokolov, and Melvyn Tyloo. A laplacian approach to stubborn agents and their role in opinion formation on influence networks. *Physica A: Statistical Mechanics and its Applications*, 557:124869, 2020.

[5] Yao-Li Chuang, Maria R D'orsogna, Daniel Marthaler, Andrea L Bertozzi, and Lincoln S Chayes. State transitions and the continuum limit for a 2d interacting, self-propelled particle system. *Physica D: Nonlinear Phenomena*, 232(1):33–47, 2007.

[6] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences*, 105(4):1232–1237, 2008.

[7] David JT Sumpter. *Collective animal behavior*. Princeton University Press, 2010.

[8] Yael Katz, Kolbjørn Tunstrøm, Christos C Ioannou, Cristián Huepe, and Iain D Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725, 2011.

[9] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[10] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[11] Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

[12] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[13] Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001.

[14] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[15] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[16] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[17] Andrei Nikolajevits Tihonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math.*, 4:1035–1038, 1963.

[18] Andrei Nikolaevich Tikhonov, AV Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013.

[19] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[20] Dhananjay Bhaskar, Angelika Manhart, Jesse Milzman, John T Nardini, Kathleen M Storey, Chad M Topaz, and Lori Ziegelmeier. Analyzing collective motion with machine learning and topology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(12):123125, 2019.

[21] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 1.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the Appendix C
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix B?
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the results of every experiment in 10 trials, see Appendix B
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We use university-sponsored HPC and can not reveal this information at review stage.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix B
   (b) Did you mention the license of the assets? [Yes] See Appendix B
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix B
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The real data set is public and we have included the details in Appendix B
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]