Sparsity-based algorithms for inverse problems

Ganguly, P.S.

Chapter 5

Learning cell–cell interactions for vascular network formation

5.1 Introduction

In many real-world applications, it is important to model dynamical equations that best describe the system studied. Dynamical equations may be constructed from first principles; the heat equation in physics is one such example. However, in other scenarios where first-principles methods may be insufficient or lacking, dynamical equations can be learned from data on the time evolution of a system.

A recent approach [108] formulates the discovery of dynamical equations as a sparse inverse problem. In this approach – known as Sparse Identification of Nonlinear Dynamics (SINDy) – the unknown dynamical equation is expressed as a linear combination of library functions, and a sparse combination of these functions able to explain the time evolution of the system is sought.

SINDy has been used to infer the dynamics of simulated and real data for a variety of canonical systems exhibiting nonlinear dynamics. In this chapter we adapt it to study vascular network formation in vertebrates.

Vascular network formation is the generation of a blood vessel network from cells that are initially separate. This process is responsible for the generation of a circulatory system during morphogenesis in vertebrates. The first step of this process is vasculogenesis, where a primary network is created. This network then sprouts and expands, in a process termed angiogenesis. Angiogenesis is also observed in cancer tumours, where it helps tumour maintenance and metastasis.

How endothelial cells organize to form a vascular network is still an open question. It has been proposed [109] that two main contributing factors are: 1) the intrinsic ability of

This chapter is based on:

Learning cell–cell interactions for vascular network formation. *P. S. Ganguly, K. A. E. Keijzer, D. Chen, T.M. Vergroesen, R. M. H. Merks and H. J. Hupkes.* (in preparation)

cells to form networks, and 2) environmental cues. The effects of both these factors have been studied using experimental and simulation studies of network formation. Although there have been extensive experimental investigations of angiogenesis [110]–[112], simulation studies are particularly effective in understanding how the interplay between different biological ingredients leads to network formation. This is because all the parameters of a simulated model can be adjusted and different parameter regimes, which may not be easy to probe in experimental studies, are easily simulated.

Different simulation paradigms have been used in the literature to study vascular network formation: one example is a lattice-free, particle-based approach [113], and another is the lattice-based cellular Potts model (CPM) [109].

The forward problem of network formation consists of modelling the cellular system using a Hamiltonian or a differential equation, followed by obtaining solutions that correspond to the steady state or have the lowest energy. However, it is not always clear which model is most suitable and which parameter regions are the most promising for observing network formation behaviours. Moreover, the correspondence between different simulation models is also not clear. For e.g. it is unknown whether there exist effective equations for stochastic Hamiltonian-based models like CPM.

In this chapter, we adapt the SINDy method to learn effective equations for vascular network formation directly from cell trajectories. In particular, we parametrize the pairwise interaction between cells instead of the vector field in our differential equation. This ensures that the number of parameters we learn remains the same despite an increase in the system size. A related work to ours is [114] where the authors adapt the SINDy framework for stochastic differential equations and parametrize the potential instead of the force vector. However, [114] considers only single particle systems in low dimensions, while we consider systems of many particles. Another related line of research is that that of learning force fields for molecular dynamics [115], where the task is to fit the energy of an atomic configuration obtained by solving the electronic Schroedinger equation. Starting with [116], the approach used is that of decomposing the energy into a sum of terms, one for each atom, and parametrizing each contribution via a neural network. While the idea of sharing parameters across particles is similar to our approach, the task in force fields parametrization is different from ours. Further, our optimization problem is similar to that of SINDy and has the advantage of being a convex optimization, while that of [116] is non-convex.

In this chapter, we focus on proof-of-concept studies, where ground-truth effective equations are available, in order to validate our approach and perform systematic numerical studies of the effect of system size, function library size and noise (Gaussian and stochastic) on the the accuracy of recovery. Our work is an important stepping stone towards applying such an approach to experimental data, where effective equations are unknown, or to other modelling paradigms like CPM, in order to find a correspondence between different simulation strategies. Effective differential equations are amenable to analysis and are much easier to simulate than cell models, thus providing much-needed analytical insight into biological systems.

This chapter is organized as follows. In Section 5.2 we review the SINDy method and provide some background on simulation methods for vascular network formation. In Section 5.3 we detail our method, which adapts the SINDy approach to learn pairwise

interactions. We give details of our numerical experiments and results in Section 5.4, and point to limitations and extensions in Section 5.5.

5.2 Background

5.2.1 SINDy

We consider the following ODE:

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}), \tag{5.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the system state at a certain time and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field that defines the dynamics of the system.

We only have data at discrete time points $\mathcal{T} := \{t_1, \ldots, t_m\}$, which we denote as X:

$$X := \begin{pmatrix} \boldsymbol{x}(t_1) \\ \boldsymbol{x}(t_2) \\ \vdots \\ \boldsymbol{x}(t_m) \end{pmatrix} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \ldots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \ldots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \ldots & x_n(t_m) \end{pmatrix}. \tag{5.2}$$

From X we can also approximate the time derivatives at $\mathcal{T}$, which we call $\dot{X}$. We shall use central differences

$$\dot{X}_{ij} := \frac{x_i(t_{j+1}) - x_i(t_{j-1})}{t_{j+1} - t_{j-1}}, \tag{5.3}$$

or forward differences

$$\dot{X}_{ij} := \frac{x_i(t_{j+1}) - x_i(t_j)}{t_{j+1} - t_j}, \tag{5.4}$$

depending on the application.

Learning problem The goal of SINDy is to learn the form of the function $\boldsymbol{g}$ from a library of basis functions, given data points X and $\dot{X}$.

First we define the library of K basis functions $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$, such that $\boldsymbol{\theta}_p : \mathbb{R}^n \to \mathbb{R}^n$. The unknown function $\boldsymbol{g}$ is approximated by a linear combination of these basis functions.

We evaluate the functions $\boldsymbol{\theta}_p$ at data points X by writing

$$\Theta(X) := \begin{pmatrix} \boldsymbol{\theta}_1(X) & \boldsymbol{\theta}_2(X) & \ldots & \boldsymbol{\theta}_K(X) \end{pmatrix}, \tag{5.5}$$

where

$$\boldsymbol{\theta}_p(X) = \begin{pmatrix} \boldsymbol{\theta}_p(\boldsymbol{x}(t_1)) \\ \boldsymbol{\theta}_p(\boldsymbol{x}(t_2)) \\ \vdots \\ \boldsymbol{\theta}_p(\boldsymbol{x}(t_m)) \end{pmatrix}.$$

We formulate the recovery of the function $\boldsymbol{g}$ as the following linear least-squares problem:

$$\underset{\boldsymbol{\xi} \in \mathbb{R}^K}{\text{minimize}} \quad \left\| \dot{X} - \Theta(X)\boldsymbol{\xi} \right\|_2^2. \tag{5.6}$$

Inducing sparsity Sparsity of the learnable coefficients is a regularization method used in machine learning to prevent overfitting, namely the fact that the model fits very well the training data but generalizes poorly to unseen data – in our case, to unseen time points. One way to induce sparsity in the coefficients is by solving the following optimization problem that has an ℓ^1 penalty:

$$\underset{\boldsymbol{\xi}\in\mathbb{R}^K}{\text{minimize}} \quad \left\|\dot{X} - \Theta(X)\boldsymbol{\xi}\right\|_2^2 + \alpha\left\|\boldsymbol{\xi}\right\|_1, \tag{5.7}$$

The above problem can be solved using LASSO. For large system sizes, LASSO is known to be computationally expensive and a sequentially thresholded least-squares (STLSQ) algorithm has been used in the literature as an alternative [108].

5.2.2 Particle-based model of vascular network formation

In this section we review the particle-based simulation paradigm that has been used in the literature to study vascular network formation.

This method was originally used to demonstrate that cell elongation and mutual attraction between endothelial cells was indeed sufficient for producing vascular networks [113], a claim that was first made using cellular Potts model (CPM) simulations [109].

In this lattice-free paradigm, each cell is represented with a particle that interacts with other particles in a predefined neighbourhood. The time evolution of the system is modelled with a Langevin equation:

$$\frac{d\boldsymbol{v}_i}{dt} = \frac{1}{m_i}\Big(-\tau\boldsymbol{v}_i + \sum_{j\neq i}\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}F_{ij} + \boldsymbol{\eta}\Big), \quad \boldsymbol{v}_i = \frac{d\boldsymbol{x}_i}{dt}, \tag{5.8}$$

where τ is the damping constant, F_{ij} is the pairwise interaction between cells and the last term is a stochastic noise term with correlation function

$$\mathbb{E}(\eta_a(t)\eta_b(t')) \propto \delta_{ab}\delta(t-t')\,. \tag{5.9}$$

The pairwise interaction F_{ij} is modelled with a short-range repulsive term and a long-range attractive term:

$$F_{ij} := \lambda_r A_r - \lambda_a A_a, \tag{5.10}$$

where A_r is the area of overlap between the smaller repulsive ellipses and A_a is the overlap between attractive ellipses (see Figure 5.1 (c)), and λ_r and λ_a are constants. The areas of overlap are usually computed in a Cartesian coordinate system and are functions of the locations, eccentricities and orientations of ellipses. We discuss this in more detail in the following section.

Without loss of generality we can set $m_i = 1$, so that the discrete time evolution, using forward differences, is:

$$\boldsymbol{a}_i(t+\Delta t) = -\tau\boldsymbol{v}_i(t) + \sum_{j\neq i}\frac{\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|}F_{ij}(t) + N_v\beta_v(t)\Delta t^{-0.5} \tag{5.11}$$

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \boldsymbol{a}_i(t+\Delta t)\Delta t \tag{5.12}$$

$$\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+\Delta t)\Delta t\,. \tag{5.13}$$

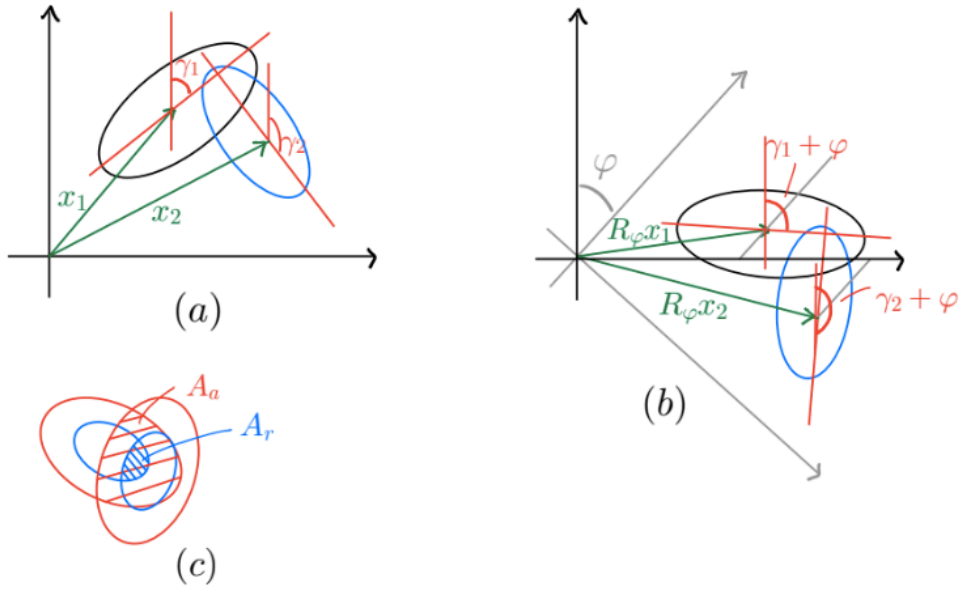

Figure 5.1: (a) The two ellipses model two cells, labelled 1 and 2. x_1, x_2 stand for the coordinates of the centers of the ellipses and γ_1, γ_2 for the angles the axis of the ellipses form with the y axis. (b) A global rotation of φ of the system. (c) Inner area of overlap A_r and outer area of overlap A_a.

Here we introduced the noise amplitude N_v and the Gaussian random vector β_v. In the overdamped regime, where the acceleration is negligible, setting $\tau = 1$, the discrete time evolution of the system reduces to

$$\boldsymbol{x}_i(t + \Delta t) - \boldsymbol{x}_i(t) = \Delta t \sum_{j \neq i} \frac{\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)}{\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|} F_{ij}(t) + N_v \beta_v(t) \sqrt{\Delta t}\,. \tag{5.14}$$

In addition to vectorial noise modulated by the amplitude N_v, the particle-based simulations make use of angular noise. This corresponds to random changes in the orientation of cells. A change in orientation of cell i is accepted with a turn probability

$$\Pi_i = \min\Big\{1, \exp\Big(\frac{1}{N_a} \sum_{j \neq i} F_{ij} - \sum_{i \neq j} F'_{ij}\Big)\Big\}, \tag{5.15}$$

where N_a is the angular noise amplitude, and F'_{ij} is the interaction between cells i and j if the orientation change is accepted.

In the following section, we show how we apply the method reviewed in Section 5.2.1 to the vascular network formation problem, and how this formulation leads us to discover cell–cell interactions from cell trajectories.

5.3 SINDy for pairwise interaction discovery

We now look at particle and lattice systems whose dynamics is governed by an interaction force between constituents. We first discuss particle systems, which is the primary focus of this chapter, and then comment on how to adapt the framework to lattice systems. In the vascular network formation problem, each of the particles represents a cell with coordinates $\boldsymbol{x}_i \in \mathbb{R}^d$, where d is the dimensionality of the problem. Then the number of variables is $n = d \times n_p$, where n_p denotes the number of particles.

We assume that $d = 2$ and that the dynamics of the system is given by (5.1) with

$$g_i(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) := \sum_{j \in \mathcal{N}_i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} F_{ij}\,, \quad i = 1, \ldots, n_p \tag{5.16}$$

$$F_{ij} := \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j, \gamma_i, \gamma_j), \tag{5.17}$$

where $\mathcal{N}_i$ is the set of particles that particle i interacts with, and γ_i denotes the angle that the i-th ellipse forms with y axis, see Figure 5.1 (a). Each ellipse is determined by $(\boldsymbol{x}_i, \gamma_i)$ and the two axes lengths which are assumed to be fixed for all cells, and therefore omitted from Φ. At this point Φ is a generic function and represents the interaction between the two ellipses. As such it should not change if we translate or rotate both ellipses w.r.t the origin. Translation by a vector $\boldsymbol{a}$ acts as $(\boldsymbol{x}_i, \gamma_i) \mapsto (\boldsymbol{x}_i + \boldsymbol{a}, \gamma_i)$. Rotation by an angle φ acts as $(\boldsymbol{x}_i, \gamma_i) \mapsto (R_\varphi \boldsymbol{x}_i, \gamma_i + \varphi)$, where R_φ is the 2×2 rotation matrix, see Figure 5.1 (b). Imposing translation invariance leads to

$$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(\boldsymbol{x}_i + \boldsymbol{a}, \boldsymbol{x}_j + \boldsymbol{a}, \gamma_i, \gamma_j) \tag{5.18}$$

whose solution is $\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j)$. Imposing rotation invariance leads to

$$\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(R_\varphi(\boldsymbol{x}_i - \boldsymbol{x}_j), \gamma_i + \varphi, \gamma_j + \varphi)\,, \quad \forall \varphi \in [0, 2\pi)\,. \tag{5.19}$$

First, we note that the following is invariant: $\Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j)$. However, this is too restrictive, as it satisfies the more general symmetry $\Phi(\boldsymbol{x}_i - \boldsymbol{x}_j, \gamma_i, \gamma_j) = \Phi(R_\varphi(\boldsymbol{x}_i - \boldsymbol{x}_j), \gamma_i + \varphi', \gamma_j + \varphi')$ even for $\varphi \neq \varphi'$. To simplify the parametrization we follow [113] and add a dependency on the areas of overlap, so that:

$$F_{ij} = \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j, A_{a,ij}, A_{r,ij})\,, \tag{5.20}$$

where A_a, A_r are as in (5.10). While these areas of overlap can be computed from $\boldsymbol{x}_i - \boldsymbol{x}_j$ and γ_i, γ_j, their expression is complicated and no simple analytical form is known [113]. We also note that this function is periodic in the second argument with period 2π.

We want to recover the function $\Phi : \mathbb{R}_+ \times [0, 2\pi) \times \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ given the trajectories of cells over time encoded in the matrix X of size $m \times n$, where n is the number of variables and m is the number of time samples. We shall now adapt the formalism described in Section 5.2.1 to this problem.

As a first step, we write down a set of basis functions $\{f_p(r, \gamma, a, b)\}_{p=1}^K$ to parametrize the unknown function $\Phi(r, \gamma, a, b)$ appearing in equation (5.20). These correspond to the following $\boldsymbol{\theta}_p$ in the formalism of Section 5.2.1:

$$(\boldsymbol{\theta}_p(\boldsymbol{x}))_i = \sum_{j \in \mathcal{N}_i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} f_p(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \gamma_i - \gamma_j, A_{a,ij}, A_{r,ij})\,. \tag{5.21}$$

Then we can plug in these values for $\boldsymbol{\theta}_p$ in equation (5.6) and solve the least square problem. The solution $\boldsymbol{\xi}$ will then describe the function Φ as:

$$\Phi(r, \gamma, a, b) = \sum_{p=1}^{K} \xi_p f_p(r, \gamma, a, b)\,. \tag{5.22}$$

If we take $\mathcal{N}_i$ in (5.16) to be the set of $n_p - 1$ points $j \neq i$, this implies that all particles interact with each other. To restrict particle interaction to within a neighbourhood, we can define a critical radius of interaction r_c, such that $f_p(r, \gamma, a, b) = 0$ if $r > r_c \ \forall p$. In the experiments shown later in the chapter, we do not learn the dynamics of the cell orientation parameters γ; instead, we treat them as known inputs.

5.4 Numerical experiments and results

5.4.1 1D lattice system

We can use the formulation of (5.16) for lattice systems as well by assigning the indices i to points on the lattice. For example in 1D, $i = 1, \ldots, n$ are the points on a line. $\boldsymbol{x}$ is then a field with values $\boldsymbol{x}_i$ at site i. In the case of lattice systems, we take the range of values $\mathcal{N}_i$ to be the neighbours on the grid. For example in 1D, $\mathcal{N}_i = \{i-1, i+1\}$ describes nearest-neighbour interactions.

In this section, we first describe our experiments on recovering the pairwise interaction between harmonic oscillators on a 1D lattice with nearest-neighbour interactions. The displacement of the ith particle is given by $x_i(t)$. We generated particle trajectories by evolving the system in the overdamped regime:

$$\dot{x}_i(t) = \sum_{j=i-1,i+1} \frac{x_i - x_j}{r_{ij}} F_{ij}, \quad F_{ij} := -k(r_{ij} - \rho), \tag{5.23}$$

where $r_{ij} = |x_i - x_j|$ and ρ is an offset. The initial configuration of oscillators and F_{ij} are shown in Figure 5.2, where we take $k = 2.0$, $\rho = 1.0$.

We integrated the dynamical equation numerically to obtain a matrix X for a discrete set of time points $\mathcal{T} = \{t_1, \ldots, t_m\}$. The matrix of time derivatives $\dot{X}$ was obtained using equation (5.3).

As the pairwise interaction between particles is a function of r_{ij}, we chose library functions that were polynomials of r_{ij}:

$$f_p(r) = r^p, \quad p \in \{0, 1, \ldots, K\} \tag{5.24}$$

and used these to solve the LASSO problem (5.7).

As a first experiment, we show how to determine the regularization parameter α in equation (5.7). For fixed K, n and m, we use LASSO to infer the parameters $\boldsymbol{\xi}$ for different values of α (see Figure 5.2). We choose the optimum α to be the one with the minimum number of non-zero terms in $\boldsymbol{\xi}$ for which the coefficient of determination

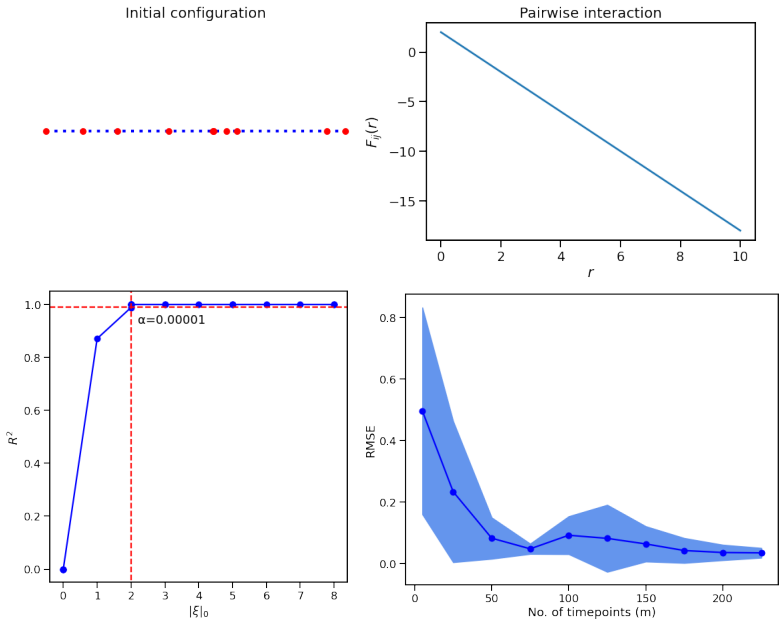

Figure 5.2: (*top left*) Initial configuration of the 1D lattice system with oscillators shown in red and connecting springs shown in blue; (*top right*) ground-truth pairwise interaction F_{ij} as a function of separation distance r. (*bottom left*) Plot of R^2 coefficient with respect to the number of non-zero parameters ξ for different values of the regularization parameter α at fixed $K = 10$, $m = 3$, $n = 1024$. The optimum regularization parameter, $\alpha = 10^{-5}$, is chosen such that $R^2 \geq 0.99$ for the least number of non-zero parameters. (*bottom right*) Plot of RMSE with respect to the number of timepoints m for noisy measurement data with $\sigma = 0.1$. The blue dots show mean values and the ribbons show standard deviations computed over 10 randomised noise seeds.

satisfies $R^2 \geq 0.99$, where:

$$R^2 = 1 - \frac{\|\dot{X} - \Theta(X)\boldsymbol{\xi}\|_2^2}{\|\dot{X} - \frac{1}{mn}\sum_{ij}\dot{X}_{ij}\|_2^2}. \tag{5.25}$$

In the absence of measurement noise, we can infer the correct coefficients for arbitrary K and n with as little as $m = 3$ timepoints (when time derivatives are computed using the central difference scheme (5.3)) and $\Delta t = 0.001\,\frac{1}{k}$.

As the next experiment, we investigate the effect of measurement noise on inference accuracy. In the most general setting, measurement noise affects both X and $\dot{X}$, the latter being numerical derivatives of the former. Applying SINDy to such data typically leads to large errors in the inferred parameters [108]. Instead as in [108], we choose to restrict measurement noise to observed values of $\dot{X}$. This translates to the forward problem:

$$\dot{X} = \Theta(X)\boldsymbol{\xi} + \eta\,, \tag{5.26}$$

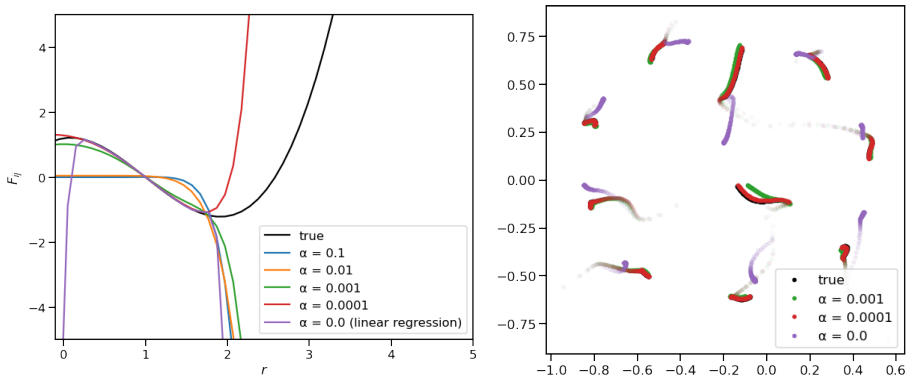

Figure 5.3: (*left*) Predicted interactions for various values of the regularization parameter α; in all inference experiments a library with $K = 11$ polynomial terms in r was used. (*right*) Ground truth trajectories (in black) overlaid with predicted trajectories for $n = 100$, $m = 100$; more transparent points are earlier in time.

where $\eta \sim \mathcal{N}(0, \sigma \mathbf{1})$.

We inferred parameters $\boldsymbol{\xi}$ for noisy measurement data using $\sigma = 0.1$ times the range of $\dot{X}$. We computed inference accuracy using the root mean squared error (RMSE) of the inferred parameters $\boldsymbol{\xi}_{\text{inf}}$ with respect to the ground truth $\boldsymbol{\xi}_{\text{gt}}$:

$$\text{RMSE} = \|\boldsymbol{\xi}_{\text{inf}} - \boldsymbol{\xi}_{\text{gt}}\|_2 \,. \tag{5.27}$$

In Figure 5.2, we observe that the RMSE is high for a small number of timepoints m and declines as m is increased.

5.4.2 2D particle system

Next we turn to a particle system in 2D, where each particle interacts with all others. This latter system brings us closer to the vascular network system, where 2D cell–cell interactions are at play.

For this system, we pick a cubic function to describe the ground-truth interaction between cells:

$$\dot{\boldsymbol{x}}(t) = \sum_{j \neq i} \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{r_{ij}} F_{ij}, \quad F_{ij} := k_1(r_{ij} - \rho)^3 - k_2(r_{ij} - \rho), \tag{5.28}$$

with $k_1 = 0.8$, $k_2 = 2.0$, $\rho = 1.0$. The inter-particle separation r_{ij} is now given by the Euclidean distance between particles i and j: $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$.

We performed simulations with $n = 10$ particles by integrating the above equation for $m = 100$ time points with time interval equal to $k_2/10$.

We inferred pairwise interactions between the particles by generating a library of polynomial terms (5.24) with $K = 11$. Using LASSO with regularization parameter α, we get different solutions for the inferred interaction in this case (shown in Figure 5.3).

High values of α lead to pairwise interactions where the cubic nature of the ground-truth function is not captured at all. Reducing α activates more and more terms in the function library. For $\alpha = 0.0$, where we effectively solve the linear least-squares problem (5.6), we get a poorer estimation for the interaction. The predicted trajectories overlaid on the ground-truth trajectories for $\alpha = 0.0001$ show a close match. This indicates that the part of the interaction that is not matched in this setting does not play a role in the data. This is not surprising as the part of the interaction that is not matched corresponds to the asymptotically increasing part of the cubic function in (5.28), and particles that experience this large force show exploding trajectories (x approaching infinity). Such particles were not included in the data in our simulations as such exploding trajectories are unphysical and unlikely to occur in a real experiment.

5.4.3 Particle-based simulations of vascular network formation

Finally we apply our method of interaction learning to simulated data of vascular network formation. For data generation in this part we used the particle-based simulation method described in Section 5.2.2, which has an open-source implementation in C++ [113].

We performed simulations with $n = 100$ elongated cells with fixed orientations. The ground-truth interaction between cells was given by equation (5.10) with $\lambda_r = 0.02$ and $\lambda_a = 0.0006$. We evolved the system using the discretized Langevin equation (5.11) for $m = 100$ time steps with time interval $\Delta t = 1.0$. The damping factor τ was set to 1.0 to simulate overdamped dynamics. To simulate vectorial stochastic noise in the locations of cells, we performed a series of simulations by modulating the noise amplitude N_v in equation (5.11). A network generated with the particle-based simulation method using noise amplitude $N_v = 0.0$ is shown in the top row of Figure 5.4.

Using our method, we then inferred cell–cell interaction terms from a library of $K = 15$ terms. The library terms used were polynomial functions of the areas of overlap A_r and A_a as well as those of the separation distance r. We also used two trigonometric terms for the relative orientation between cells γ. The full library used was:

$$\begin{aligned} f_1 = 1.0,\, f_2 = A_r,\, f_3 = A_a,\, f_4 = A_r^2,\, f_5 = A_a^2, \\ f_6 = A_r^3,\, f_7 = A_a^3,\, f_8 = A_r^4,\, f_9 = A_a^4,\, f_{10} = \cos(\gamma), \\ f_{11} = \sin(\gamma),\, f_{12} = r^1,\, f_{13} = r^2,\, f_{14} = r^3,\, f_{15} = r^4\,. \end{aligned}$$

In Figure 5.4, we plot inferred networks for noise amplitude $N_v = 0.0$. These networks were obtained by using the coefficients of the inferred terms as input to the particle-based simulations and integrating forward in time. The global structure of the inferred networks is qualitatively similar to that of the true networks. To quantify the similarity between networks at a given timepoint, we defined the deviation of the inferred network from the true network as

$$\epsilon = \frac{1}{n_p} \sum_{i=1}^{n_p} \|\boldsymbol{x}_i^{\text{inf}} - \boldsymbol{x}_i^{\text{gt}}\|_2, \tag{5.29}$$

where $\boldsymbol{x}_i^{\text{inf}}$ denotes the position of cell i in the inferred network and $\boldsymbol{x}_i^{\text{gt}}$ denotes its position in the ground truth.

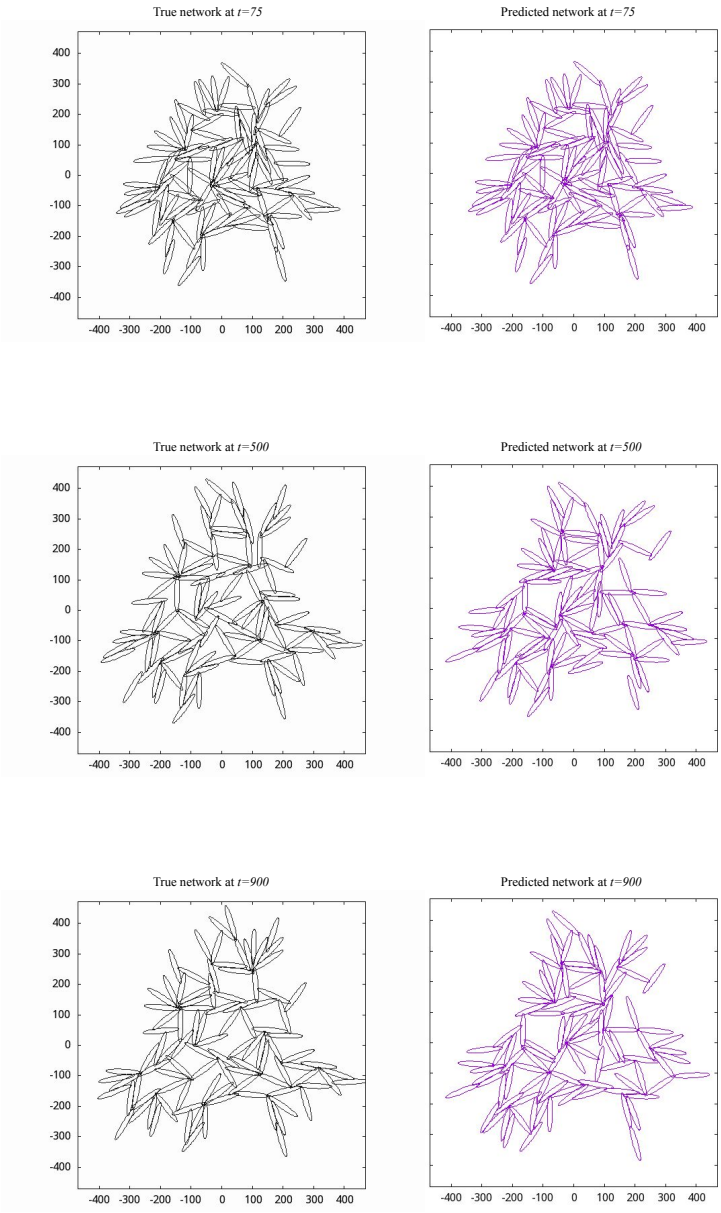

Figure 5.4: True and inferred networks of a particle-based simulation of angiogenesis using 100 elongated cells. Pairwise interactions were inferred for $N_v = 0.0$ using $m = 100$, $n = 100$; inferred networks were obtained by using the inferred interactions as input to an open-source C++ particle-based simulation code [113].

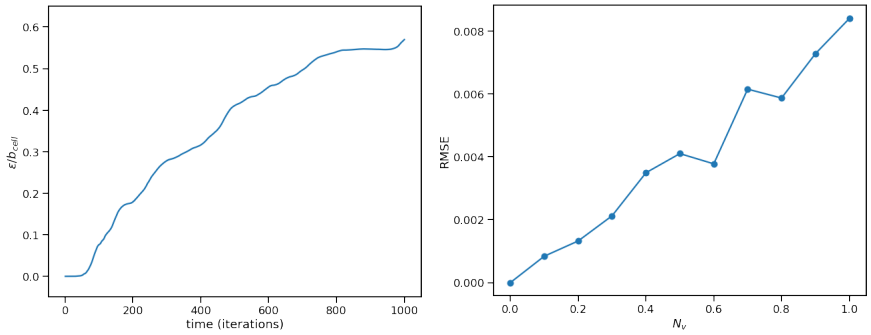

Figure 5.5: (*left*) Plot showing deviation of inferred network from the true network as a function of iterations for $N_v = 0.0$; (*right*) RMSE with respect to the stochastic noise amplitude N_v. For all inference experiments, we used a library with $K = 15$ terms, $m = 100$ and $n = 100$.

In Figure 5.5, we plot this deviation normalised by the major diameter of cells (b_{cell}, which is taken to be constant) as a function of iterations. The deviation is almost zero for iteration numbers lesser than 50 and is less than 10% of the cell diameter for the first 100 iterations. This indicates a good match with the data used for inference, given that we used the first 100 iterations for inferring pairwise interactions. The inferred network deviates from the true network to a greater extent for the next iterations; this is expected as small deviations in the earlier time points accumulate to larger differences later on. Qualitatively, we observe greater differences between the networks at $t = 500$ than between those at $t = 75$ in Figure 5.4. Interestingly, for longer iteration times (greater than 800), the deviation flattens out; this could be the result of the two networks (true and inferred) reaching separate steady states. Note, however, that the largest deviation in networks is still quite small – lesser than one cell diameter. For comparison, the field-of-view in the plots in Fig 5.4 is slightly greater than $8\, b_{\text{cell}}$.

In Figure 5.5, we also plot the RMSE (5.27) as a function of the noise amplitude N_v. For the noiseless simulation ($N_v = 0.0$), we were able to estimate the coefficients with high accuracy. The RMSE increased almost linearly for increasing amplitudes N_v of stochastic noise. Qualitatively, this observation is similar to one reported in [114], where the authors study homogeneous diffusion in the presence of thermal noise and report an increase in the percentage of function libraries that result in the correct solution with decreasing noise. In our experiments, adding stochastic noise did not have a large effect on inference accuracy, as evidenced by an increase in RMSE of less than 1%. This suggests that our deterministic method performs reasonably for the amounts of stochastic noise in such simulations.

5.5 Discussion and conclusions

In this chapter we discussed a method to learn pairwise interactions between cells from their trajectories. We adapted an existing equation learning method, SINDy, to our problem and demonstrated our approach on simulated lattice and particle data. On 1D lattice data we demonstrated the effect of Gaussian measurement noise on inference accuracy and presented a way to choose the optimum sparsity level by tuning the regularization parameter α. On 2D particle data, we further demonstrated the effect of the parameter α on the learned interaction and showed that parts of the interaction that are not matched correspond to specific regions that are not sampled in the data. On particle-based simulations of angiogenesis, we presented results on learning the interaction between elongated cells, and showed how the accuracy of inference degrades with stochastic noise. In the following, we briefly discuss how to apply our method to cellular Potts model (CPM) simulations.

The CPM is another simulation paradigm that has been used to elucidate mechanisms of vascular network formation. In particular, it was used to show that cell elongation was crucial to network generation [109], a claim that is supported by experimental observations. The CPM uses lattice spins to simulate biological cells. Each cell is a patch of identical spins, while the intercellular spaces are modelled by patches of the opposite spin. The interaction between neighbouring spins is used to generate an effective Hamiltonian, whose ground state is reached by performing Monte Carlo steps. To learn a CPM, we would use a library of Hamiltonian terms and coefficients. The observed data, analogous to the data obtained from particle-based simulations, would be the centres of mass and orientations of whole cells, which in the case of CPM correspond to patches of spins or Potts domains.

Applying our method to CPM is a stepping stone to inferring effective equations from experimental wet-lab data. This would enable a complementary approach to angiogenesis simulations, and pave the way to directly learning interactions that lead to network formation.