Parameter Inference with Bifurcation Diagrams

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimation of parameters in differential equation models can be achieved by applying learning algorithms to quantitative time-series data. However, sometimes it is only possible to measure qualitative changes of a system in response to a controlled condition. In dynamical systems theory, such change points are known as *bifurcations* and lie on a function of the controlled condition called the *bifurcation diagram*. In this work, we propose a gradient-based semi-supervised approach for inferring the parameters of differential equations that produce a user-specified bifurcation diagram. The cost function contains a supervised error term that is minimal when the model bifurcations match the specified targets and an unsupervised bifurcation measure which has gradients that push optimisers towards bifurcating parameter regimes. The gradients can be computed without the need to differentiate through the operations of the solver that was used to compute the diagram. We demonstrate parameter inference with minimal models which explore the space of saddle-node and pitchfork diagrams and the genetic toggle switch from synthetic biology. Furthermore, the cost landscape allows us to organise models in terms of topological and geometric equivalence.

1 Introduction

Inverse problems [1] arise in biology and engineering in settings when the model is not fully known and the desire is to match model behaviour to a given set of observations. This helps systematically guide both model and experimental design. While we would like to understand the quantitative details of a system, often only qualitative changes in response to varying experimental conditions can be robustly measured across independent studies [2, 3]. For example, several studies are likely to agree that the human immune system activates above a threshold concentration of a pathogen and deactivates at a lower threshold concentration, but may disagree on the exact quantities of the thresholds or the magnitudes of the immune response. Bifurcation theory provides us a framework for studying these transitions in a manner that is independent of quantitative details [4]. The emerging picture suggests that identification of the qualitative behaviour – the bifurcation diagram – should precede any attempt at inferring other properties of a system [5].

Inferring the parameters of a model directly from a bifurcation diagram is difficult because it is not obvious how multiple parameters in concert control the existence and position of a bifurcation. It could even be impossible for the model to bifurcate in the manner desired. Several approaches exist to place bifurcations to desired locations once a manifold is present [6–8] yet typically resort to sampling techniques to search for them in the first place [9, 10]. Progress has been made in cases where model structure and stability conditions are used to refine the search space [11, 12] yet the resulting objectives are still not explicit in the bifurcation targets and also not differentiable. In the emerging field of scientific machine learning [13–15], parameters of structured mechanistic models

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

are favoured over flexible models in larger parameter spaces. A scalable method for navigating the space of bifurcation diagrams would enable design of differential equations with high-level qualitative constraints. Furthermore one could begin organising models according to qualitatively distinct behaviours.

Back-propagation through differential equation solvers has been a breakthrough over the past couple of years [16, 17] that enabled scalable parameter inference for differential equations from trajectory data. Although one could use trajectory data to create the aforementioned qualitative constraints [18, 19] this would entail over-constraining information originating from the kinetics and dynamical transients of the model. Furthermore, such data usually does not contain sufficient information about dynamical transients in order to identify kinetic parameters. Techniques for back-propagating through implicit equation solvers have also been developed [20, 21] although to the best of the authors' knowledge have not been applied to bifurcation diagrams at the time of writing this paper.

The problem of inferring differential equation parameters against a user-specified bifurcation diagram decomposes into two parts: searching for bifurcating regimes and matching the locations of bifurcation points to desired values. Matching bifurcation locations is a supervised problem where the data are expressed as bifurcations points [7, 10]. Searching for bifurcations is an unsupervised problem because when bifurcations are not present, there is no distance defined between data and prediction [9]. Therefore only properties of the model can be used to start the search. We propose a semi-supervised approach for performing both tasks in an end-to-end fashion. The bifurcation diagram encodes high-level qualitative information defined by state space structures, rather than kinetics. We apply the strategy of implicit layers [20, 21] to calculate gradients. To compute the diagram we use a predictor-corrector method called deflated continuation [22, 23] developed for partial differential equations.

We find that the cost function landscape contains basins that not only allow us to synthesise models with a desired bifurcation diagram but also allow us to organise models in terms of topological and geometric equivalence. We discuss the relevance of this in model selection. In summary, our paper has the following main contributions:

- An end-to-end differentiable method for locating bifurcations in parameter space and then matching their dependency on a control condition to user-specified locations
- Implementation of the method as a Julia package
  `see attached zip file`
- Leveraging the cost landscape for a novel way of organising differential equation models in terms of geometric and topological equivalence

1.1 Preliminaries

Suppose we collected observations along a scalar control condition $p \in \mathbb{R}$ and conclude that there are specific values of p for which there are qualitative changes in system behaviour. Let $\mathcal{D}$ be the set of those values and let us hypothesise that these transitions occur due to bifurcations in the dynamics that drive the underlying mechanism. Let us model the mechanism with a parameterised set of differential equations for states $u \in \mathbb{R}^N$ with a vector function F_θ in a parameter space $\theta \in \mathbb{R}^M$.

For the purposes of introducing this work, we will consider the simplest class of bifurcations known as *co-dimension one* bifurcations not including limit cycles. Therefore $\mathcal{D}$ should contain conditions for which we hypothesise changes in multi-stable behaviour. Let the equations be

$$\frac{\partial u}{\partial t} = F_\theta(u, p) \qquad \text{where} \quad F_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^N \tag{1}$$

In the context of the differential equations, and not considering limit cycles for now, a co-dimension one bifurcation can be defined by a set of conditions on the determinant of the Jacobian $\left|\frac{\partial F_\theta}{\partial u}\right|$. The determinant of the Jacobian quantifies the rate at which trajectories in a local patch of state-space $u \in \mathbb{R}^N$ converge or diverge. The determinant approaching zero means that the dynamics of the system are slowing down, which is an important indicator for the onset of a transition between qualitative behaviours. Furthermore, the slowing down must necessarily be followed by breakdown of stability; for this to be true it is sufficient *but not necessary* to require that the determinant cross zero with a finite slope, meaning that its directional derivative along the diagram $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right|$ is not zero. The

set of predicted values for the control condition $\mathcal{P}(\theta) \subset \mathbb{R}$ at which bifurcations occur are defined as

$$\mathcal{P}(\theta) := \left\{ p \; \exists \; u : \; F_\theta(u,p) = 0, \; \left| \frac{\partial F_\theta}{\partial u} \right| = 0, \; \frac{d}{ds} \left| \frac{\partial F_\theta}{\partial u} \right| \neq 0 \right\} \tag{2}$$

The most common bifurcations between steady states, not including limit cycles, are saddle-nodes and pitchforks [24]. The relationships between our conditions for bifurcation (2) and the conditions for saddle-node are detailed in Appendix B. Saddle-node bifurcations, which often appear in pairs (Figure 1A) are defined by stable and unstable fixed points meeting and disappearing. Pitchfork bifurcations are special cases of the saddle-node where a single steady state splits into two stable and one unstable steady state (Figure 1B). To illustrate these bifurcations, we define minimal models (Figure 1) that span the space of saddle-node and pitchforks, where indeed zero crossings in the determinant with a finite slope define the set of prediction $\mathcal{P}(\theta)$. The location of these crossings in general may not match the targets $\mathcal{D}$.

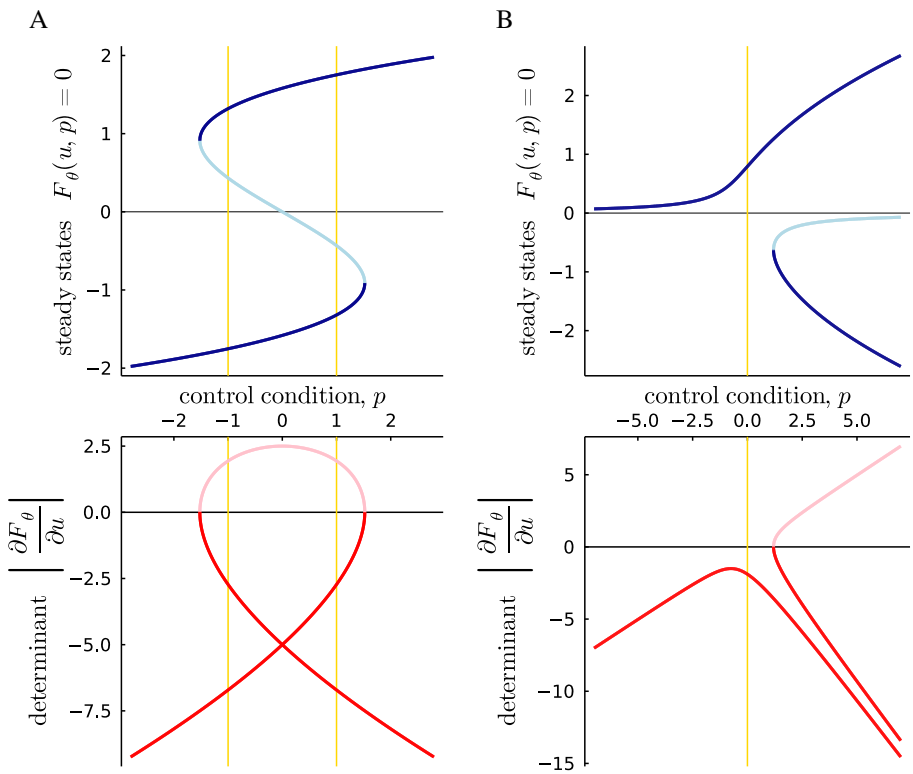

Figure 1: Illustration of bifurcation diagrams for minimal models of bifurcations. A. Saddle-node bifurcations arise for $F_\theta(u,p) = p + \theta_1 u + \theta_2 u^3$ when $\theta = (\frac{5}{2}, -1)$. B. Pitchfork bifurcations arise for $F_\theta(u,p) = \theta_1 + pu + \theta_2 u^3$ when $\theta = (\frac{1}{2}, -1)$. Targets are illustrated by light yellow vertical lines. Bifurcation curves are shown as solid blue and red lines, with lighter shades indicating the determinant crossing zero at locations $\mathcal{P}(\theta)$ giving rise to unstable solutions.

For a given set of parameters θ one could compute the set of predicted bifurcations $\mathcal{P}(\theta)$ using parameter continuation methods [23, 22]. Our goal is to find optimal parameters θ^* that match predictions $\mathcal{P}(\theta^*)$ to specified targets $\mathcal{D}$. We must design a suitable cost function L so that

$$\theta^* := \text{argmin}_\theta L(\theta|\mathcal{D}) \tag{3}$$

The optimal θ^* is not expected to always be unique, but is in general a manifold representing the space of qualitatively equivalent models. Ideally, the cost function L should reward θ for which the number of predicted bifurcations is equal to the number of targets, $|\mathcal{P}(\theta)| = |\mathcal{D}|$. This is especially important in the case where there are no predictions $|\mathcal{P}(\theta)| = 0$.

2 Proposed Method

2.1 Semi-supervised Cost Function

To identify parameter sets that give rise to bifurcation diagrams with specified bifurcation points, we propose a semi-supervised cost function that comprises two terms. The role of the supervised term is simply to reward predicted bifurcations to coincide with the specified target locations. This of course relies on such bifurcations existing. The role of the unsupervised term is to encourage an optimiser to move towards parameter regimes that do exhibit bifurcations.

2.1.1 Supervised term: matching bifurcations to target locations

In order for predicted bifurcations $p(\theta) \in \mathcal{P}(\theta)$ to match targets $p' \in \mathcal{D}$ we need to evaluate an error term $|p(\theta) - p'|$. A naive approach might take an average over the norms for all prediction-target pairs. However this gives rise to unwanted cross-terms and the possibility of multiple predictions matching the same target without any penalty for unmatched targets. Therefore, we choose a geometric mean over the predictions and an arithmetic mean over targets:

$$E(\theta, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{p' \in \mathcal{D}} \prod_{p(\theta) \in \mathcal{P}(\theta)} |p(\theta) - p'|^{\frac{1}{|\mathcal{P}|}} \tag{4}$$

The supervised term is only zero when each target is matched by at least one prediction and allows for cases where the number of predictions is greater than or equal to the number of targets $|\mathcal{P}| \geq |\mathcal{D}|$. An alternative approach, which undesirably introduces more hyper-parameters, would be to let each prediction $\mathcal{P}(\theta)$ represent the centroid of a mixture distribution and use expectation-maximisation to match the centroids to targets $\mathcal{D}$.

2.1.2 Unsupervised term: encouraging bifurcations

We can see from Figure 1 and definition (2) that predictions $p(\theta)$ can be identified by looking for points along the curve where the determinant crosses zero $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$ with a finite slope $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right| \neq 0$. Using these quantities we can define a positive semi-definite measure $\varphi_\theta(s)$ of zero crossings in the determinant along a curve parameterised by s which we define as

$$\varphi_\theta(s) := \left(1 + \left|\frac{\left|\frac{\partial F_\theta}{\partial u}\right|}{\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right|}\right|\right)^{-1} \tag{5}$$

The bifurcation measure $\varphi_\theta(s)$ is maximal at bifurcations and has finite gradients in non-bifurcating regimes (Figure 2). More specifically, the measure $\varphi_\theta(s)$ is one at bifurcation points and goes to zero an odd number of times between bifurcations. This is because $\left|\frac{\partial F_\theta}{\partial u}\right|$ must eventually turn around in order to return back to zero, resulting in the directional derivative $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right|$ going to zero. Hence the measure $\varphi_\theta(s)$ goes to zero for each turning point (see Figure 2).

On the other hand, as the determinant $\left|\frac{\partial F_\theta}{\partial u}\right|$ diverges, we approach regimes far away from any bifurcations and hence $\varphi_\theta(s) \to 0$. Since we would still like to have non-zero gradients with respect to θ in these regimes we designed the measure to go to zero sufficiently slowly.

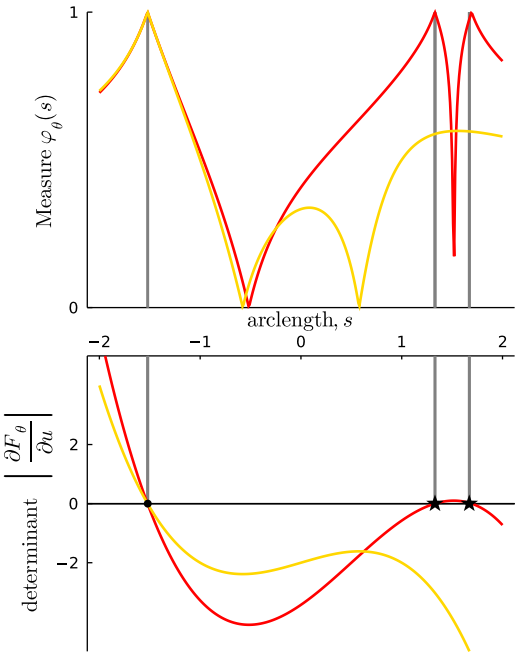

Figure 2: Bifurcation measure $\varphi_\theta(s)$ and determinant $\left|\frac{\partial F_\theta}{\partial u}\right|$ along the arclength s of two different bifurcation curves demonstrating how maximising the measure along the curve maintains the existing bifurcation marked by a circle, while encouraging new bifurcations marked by stars.

While the calculation of the determinant is straightforward, its directional derivative requires a tangent vector to the bifurcation curve. Fortunately the tangent vector $T_\theta(s)$ at the solution $u(s), p(s)$ anywhere along the curve s can be calculated as the nullspace of the rectangular $N \times (N+1)$ Jacobian

$$\left.\frac{\partial F_\theta}{\partial(u,p)}\right|_{F_\theta(u(s),p(s))=0} \cdot T_\theta(s) = 0 \tag{6}$$

This equation guarantees that the tangent vector $T_\theta(s)$ is orthogonal to all hyper-planes defined by the components of F_θ. In this setting the dimension of the nullspace is always known, and therefore can reliably be calculated using QR factorisation methods [25].

Equipped with a measure that quantifies the appearance of bifurcations along a bifurcation arc we can define the total measure for a bifurcation diagram as

$$\Psi(\theta) := \frac{\int_{F_\theta(u,p)=0} \varphi_\theta(s)\,\mathrm{d}s}{\int_{F_\theta(u,p)=0} \mathrm{d}s}. \tag{7}$$

Here we denote $\int_{F_\theta(u,p)=0} \mathrm{d}s$ as the sum of the line integrals in $(u,p) \in \mathbb{R}^{N+1}$ defined by the level set $F_\theta(u,p) = 0$ with s being an arbitrary parametrisation of the curves. The total measure $\Psi(\theta)$ is normalised such that $\Psi(\theta) \to 1$ in the regimes where the controlled condition region p is densely packed with bifurcations. The total measure $\Psi(\theta)$ is added to the supervised term as if it were a likelihood. This defines the semi-supervised cost function as

$$L(\theta|\mathcal{D}) := \big(|\mathcal{P}| - |\mathcal{D}|\big) \log \Psi(\theta) + E(\theta, \mathcal{D}), \tag{8}$$

The pre-factor $|\mathcal{D}| - |\mathcal{P}|$ in the unsupervised term ensures that the gradients are always pushing optimisers towards a state where $|\mathcal{D}| = |\mathcal{P}|$. This can be seen as a step-wise annealing of the unsupervised term until the desired state is reached.

2.2 Differentiating the semi-supervised cost function

To make use of gradient-based optimisers to locate desired bifurcation diagrams, we show here how to differentiate the cost function. First, we note that while individual bifurcations $p(\theta)$ depend smoothly on θ, the total number of predictions $|\mathcal{P}|$ does not have gradient contributions with respect to θ. Therefore, we can safely drop the dependency in the prediction counter and now proceed in taking gradients with respect to θ knowing that the only dependencies we need to track are for individual bifurcations $p(\theta)$ within the definition the supervised term (4) and the total measure (7). Therefore,

$$\frac{\partial L}{\partial \theta} = \big(|\mathcal{P}| - |\mathcal{D}|\big)\,\lambda\,\frac{\partial \Psi}{\partial \theta}\Psi(\theta)^{-1} + \frac{1}{|\mathcal{D}||\mathcal{P}|}\sum_{p'}\prod_{p(\theta)} |p(\theta) - p'|^{\frac{1}{|\mathcal{P}|}} \sum_{p(\theta)} \frac{\partial p}{\partial \theta}\,(p(\theta) - p')^{-1} \tag{9}$$

In a similar vein to back-propagation through neural differential equations [16] we would like to be able to calculate the gradient $\frac{\partial L}{\partial \theta}$ without having to differentiate through the operations of the solver that finds the bifurcation diagram $F_\theta(u,p) = 0$ and the bifurcation locations $p(\theta)$. To calculate the gradient of the measure $\frac{\partial \Psi}{\partial \theta}$ we need to differentiate line integrals that depend on θ. Fortunately this can be done by the application of the generalised Leibniz integral rule, details of which can be found in Appendix C.

The gradient of the bifurcation points $\frac{\partial p}{\partial \theta}$ is found by application of the implicit function theorem to a vector function $G_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ whose components represent the two constraints $F_\theta(u,p) = 0$ and $\left|\frac{\partial F_\theta}{\partial u}\right| = 0$. By following a similar strategy to that used by implicit layers [20] we yield an $(N+1) \times M$ Jacobian representing a deformation field [26] for each θ direction. The gradient we are looking for becomes

$$\frac{\partial p}{\partial \theta} = -\hat{p} \cdot \frac{\partial G_\theta}{\partial(u,p)}^{-1} \left.\frac{\partial G_\theta}{\partial \theta}\right|_{G_\theta(u,p)=0} \quad \text{where} \quad G_\theta(u,p) := \begin{bmatrix} F_\theta(u,p) \\ \left|\frac{\partial F_\theta}{\partial u}\right| \end{bmatrix} \tag{10}$$

Here $\hat{p}$ is a unit vector in $(u,p) \in \mathbb{R}^{N+1}$ that picks out the deformations along the p-direction. If we wanted to place the bifurcation at target steady state u' as well as target control condition p' we would use the full $(N+1) \times M$ deformation matrix. Calculation of this matrix involves inverting an

$(N+1) \times (N+1)$ Jacobian $\frac{\partial G_\theta}{\partial(u,p)}$. Instead of explicitly inverting the Jacobian the corresponding system of linear equations is solved. The determinant of this Jacobian goes to zero in the degenerate case where $\frac{d}{ds}\left|\frac{\partial F_\theta}{\partial u}\right| = 0$, further justifying our choice of measure $\Psi(\theta)$ which discourages the degenerate case.

The cost function is piece-wise smooth and differentiable with undefined gradients only in parameter contours where the number of predictions $|\mathcal{P}|$ changes; this is when $\Psi(\theta)$ is undefined and the inverse of $\frac{\partial G_\theta}{\partial(u,p)}$ does not exist. Given a set of solutions to $F_\theta(u,p) = 0$ and locations $p(\theta)$ the gradient $\frac{\partial L}{\partial \theta}$ can be evaluated using automatic differentiation methods [27–29] without needing to back-propagate through the solver that obtained the level set $F_\theta(u,p) = 0$ in the forward pass.

3 Experiments & Results

In this section, we apply the method first to minimal examples that can produce saddle-node and pitchfork bifurcations, and then a more complex model that has multiple parametric regimes that produce saddle-node bifurcations.

3.1 Minimal Models

Optimisations of two parameters (θ_1, θ_2) using simple gradient descent from `Flux.jl` with learning rate $\eta = 0.01$ for the minimal saddle-node and pitchfork models yield trajectories approaching lines of global minima in the cost function (Figures 3) which represent a set of geometrically equivalent models. Two bifurcation diagrams are geometrically equivalent if the number, type and locations of bifurcations match the specified targets $\mathcal{D}$.

We can see that the geometrically equivalent lines are contained within larger basins where the correct number and type of bifurcations are present but do not match the locations of targets $\mathcal{D}$. All models within this basin are in some sense topologically equivalent. This hierarchical classification allows us to identify the set of models that satisfy observed qualitative behaviour [5] before any attempt at inferring kinetic parameters, which is done by choosing a model along the line of geometrically equivalent models.

Optimisation trajectories for the two minimal models appear mostly circumferential. This is because the models were set up such that the radial direction from the origin in θ space mostly scale kinetics whereas the circumferential direction changes the bifurcation topology. This suggests that the gradients of our cost function seek to change model geometry over kinetics.

3.2 Genetic Toggle Switch

In this section we optimise a model where the states share a Hill function relationship with co-operatively $n = 2$; these models often emerge from mass action kinetics with quasi-steady state approximations and are used to model species concentrations. After re-scaling the equations governing the dynamics of concentrations, the simplified equations for state u_1 and u_2 become

$$\partial_t u_1 = \frac{a_1 + (pu_2)^2}{1 + (pu_2)^2} - \mu_1 u_1 \quad \partial_t u_2 = \frac{a_2 + (ku_1)^2}{1 + (ku_1)^2} - \mu_2 u_2 \tag{11}$$

where a_k is the baseline production rate for species k in the absence of the other species. Each species has a finite degradation rate μ_k. Finally we have two sensitivity constants p and k, one of which is chosen as our control condition. A baseline production rate $a_k > 1$ recovers an inhibitor type hill function for species k and is an activator otherwise. The sensitivities are proportional to the slope of the hill productions. Solving for the steady states, substituting the equation for u_1 into u_2 and rearranging gives rise to the relationship

$$\frac{k}{\mu_1} = \frac{(1 + (\frac{p}{\mu_2}u')^2)\sqrt{a_2 - u'}}{(a_1 + (\frac{p}{\mu_2}u')^2)\sqrt{u' - 1}} \quad \text{where} \quad u' := u_2\mu_2 \tag{12}$$

which reveals that only a_1, a_2 and the ratio between the sensitivity and degradation parameters, $\frac{k}{\mu_1}$, affect the solutions to this equation, and hence the locations of the bifurcations (Figure 4A). In 98% of 800 runs, optimisation using the ADAM optimiser [30] from `Flux.jl` with learning rate $\eta = 0.1$

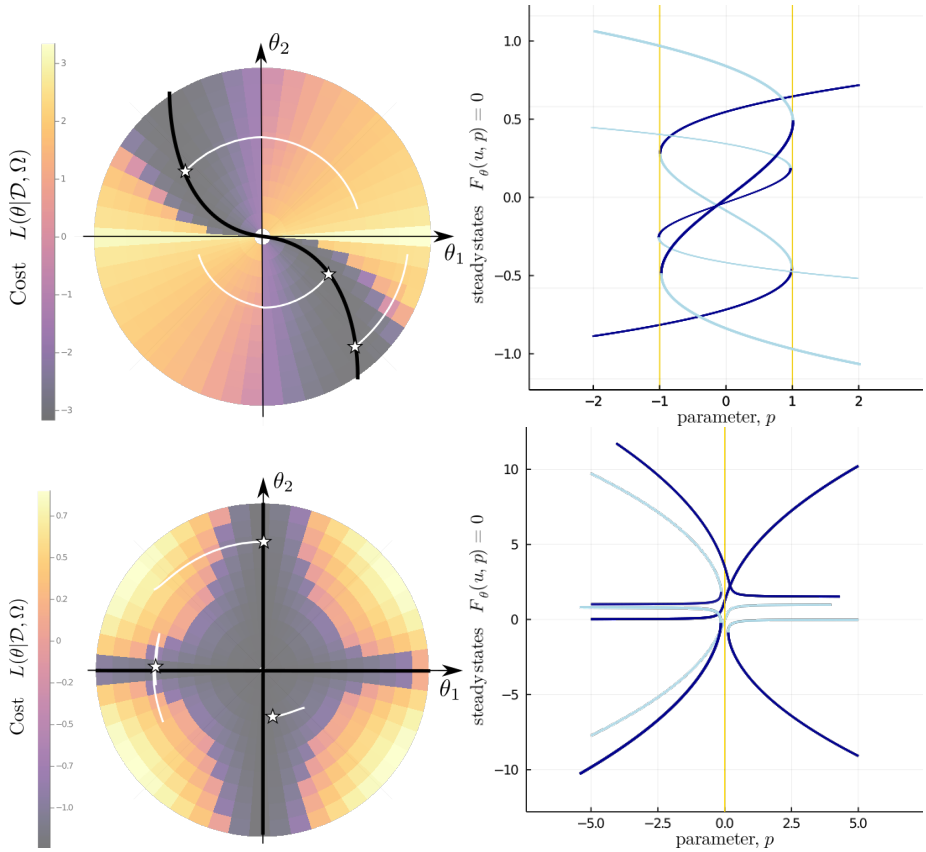

Figure 3: Saddle-node $F_\theta(u,p) = p+\theta_1 u+\theta_2 u^3$ and pitchfork $F_\theta(u,p) = \theta_1+up+\theta_2 u^3$ optimised with respect to θ so that predicted bifurcations $\mathcal{P}(\theta)$ match targets $\mathcal{D}$ in control condition p. The right panel shows bifurcations diagrams for the three optimal θ^* marked by stars on the left panel. The optimisation trajectories in white follow the gradient of the cost, approaching the black lines of global minima in the left panel

converged to one of two clusters: mutual activation ($a_1 < 1, a_2 < 1$; cluster 1) and mutual inhibition ($a_1 > 1, a_2 > 1$; cluster 2) regimes. Example bifurcation diagrams illustrate how the bifurcation curves of each species are positively correlated in mutual activation and negatively correlated for mutual inhibition (Figure 4B).

In order to maintain biological interpretability, optimisation was restricted to the positive parameter regime by transforming the parameters to log-space $\theta \to 10^\theta$. At the beginning of each optimisation run an initial θ was chosen in the log-space by sampling from a multivariate normal distribution with mean zero and standard deviation one.

3.3 Complexity

Performing one iteration of the optimisation requires the computation of the gradient of the cost (9), requiring a computation of the bifurcation diagram with parameter continuation methods, which includes the evaluation of matrix inversions (10). Instead of evaluating the inversions directly, we solve a system of linear equations, applying the same strategy as implicit layers [20, 21]. This leaves us with the computational bottleneck of calculating the determinant of the state space Jacobian, required in both the bifurcation measure (5) and gradient (10). This calculation scales like N^2 where N is the number of state space variables (Figure 5).

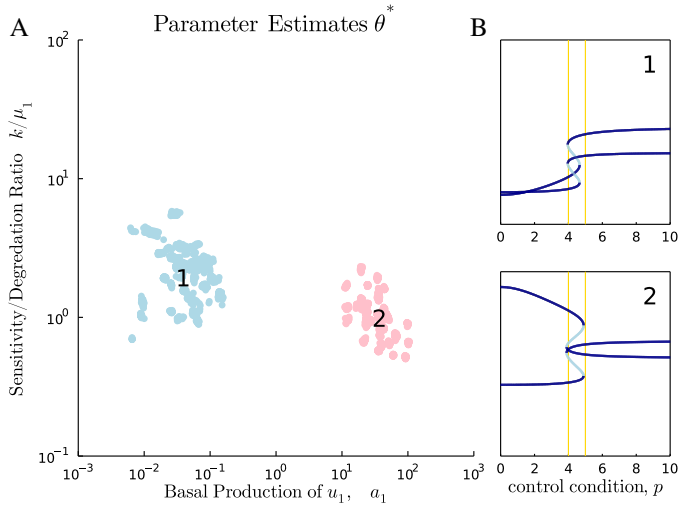

Figure 4: Bifurcation inference for the two-state model (11). A. Optimal parameter estimates θ^* for the targets $\mathcal{D} = \{4, 5\}$ reveal two clusters of qualitatively different regimes: mutual activation ($a_1 < 1$; cluster 1) and mutual inhibition ($a_1 > 1$; cluster 2). B. Example bifurcation diagrams indicate positively and negatively correlated dependencies between the two model states, as a function of the control condition.

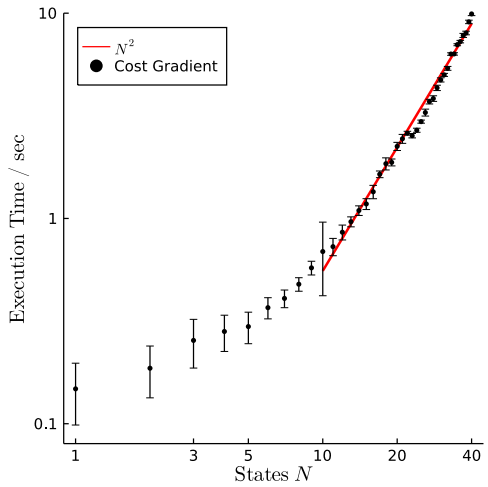

Figure 5: Complexity scaling of calculating the gradient of the cost function. Calculations were performed on an Intel Core i7-6700HQ CPU @ 2.60GHz x 8 without GPU acceleration

For this calculation a model (13) was chosen that so that it is extensible both in the number of parameters M and the number of states N.

$$\begin{cases} \partial_t u_1 = \sin^2 p - (\theta_1 \sin^2 p + 1) u_1 \\ \partial_t u_n = u_{n-1} - (\mu_n^2 + 1) u_n \qquad 2 \leq n \leq N \end{cases} \tag{13}$$

In this model only the first state u_1 defines the shape of the bifurcation diagram, while the remaining states are merely linearly proportional to the first. The parameters μ_n contain sums of θ_m allowing us a flexible choice on the number of parameters while maintaining stable solutions for the bifurcation diagram.

While still tractable on laptop computers for states $N < 100$ our implementation currently does not scale well for partial differential equations where a large the number of states N arises from discretisation of the spatial variables. The only reason we need this determinant is because it

is an indicator of bifurcations. We can address the computational bottleneck by finding a more computationally efficient way of calculating this indicator. One approach would be to take the product of a finite subset of eigenvalues of the system. Note that any more efficient calculation must still permit back-propagation through it.

4 Conclusion & Broader Impact

We proposed a gradient-based semi-supervised approach for inferring the parameters of differential equations that produce a user-specified bifurcation diagram. By applying implicit layers [20, 21] and the generalised Leibniz rule [31] to the geometry of the implicitly defined steady states [32] it is possible to use automatic differentiation methods to efficiently calculate gradients. We defined a bifurcation measure that uses the determinant of the state-space Jacobian as an indicator for bifurcating parameter regimes in the unsupervised term of the cost function. The gradients of the cost can be efficiently computed using automatic differentiation methods. The computational bottleneck is the evaluation of the state-space Jacobian determinant which limits the implementation to ordinary differential equations.

We demonstrated our approach on models with one bifurcation parameter that can give rise to pitchforks and saddle-nodes. The estimated parameters form distinct clusters, allowing us to organise models in terms of topological and geometric equivalence. In the case of the genetic toggle switch we recovered mutual activation and inhibition regimes.

Although we did not consider limit cycles, our approach can be modified using the conditions for Poincaré-Andronov-Hopf bifurcations instead of the conditions on the Jacobian determinant. Our approach generalises naturally to bifurcation manifolds such as limit point curves or surfaces. This is because the normal components of implicit derivatives can still be calculated for under-determined systems of equations [26, 33, 34]. In the case of manifolds it would be more appropriate to use isosurface extraction algorithms rather than continuation to obtain the steady-state manifold. Our approach does not depend on the details of the steady-state solver and therefore can still be applied.

In dynamical systems theory the geometry of state-space determines all of the qualitative behaviours of a system. Our work makes progress towards designing models directly in state-space, rather than the spatial or temporal domain. This is valuable to experimentalists who only have qualitative observations available to them and wish to navigate the space of qualitative behaviours of their system. Our work lies within a trend of progress in the scientific machine learning community, where structured domain-informed models are favoured over flexible models that live in large parameter spaces.

References

[1] U. G. Abdulla, R. Poteau, A. Binder, H. W. Engl, C. Flamm, P. K. Â. Ugler, J. Lu, S. M. Â. Uller, and P. Schuster, "Inverse problems in systems biology," *Inverse Problems*, vol. 25, p. 51, 2009.

[2] J. J. Tyson, K. Chen, and B. Novak, "Network dynamics and cell physiology," *Nature reviews Molecular cell biology*, vol. 2, no. 12, pp. 908–916, 2001.

[3] P. Grant, G. Szep, O. Patange, J. Halatek, V. Coppard, A. Csikász-Nagy, J. Haseloff, J. Locke, N. Dalchau, and A. Phillips, "Interpretation of morphogen gradients by a synthetic bistable circuit," *Nature Communications*, vol. 11, no. 1, 2020.

[4] Y. A. Kuznetsov, "Topological Equivalence, Bifurcations, and Structural Stability of Dynamical Systems," in *Elements of Applied Bifurcation Theory*, pp. 39–76, Springer New York, 2004.

[5] M. P. H. Stumpf and E. Roesch, "Parameter inference in dynamical systems with co-dimension 1 bifurcations," *Royal Society*, vol. 6, no. 10, 2019.

[6] K. Iwasaki and Y. Kamimura, "An inverse bifurcation problem and an integral equation of the Abel type," *Inverse Problems*, vol. 13, pp. 1015–1031, 1997.

[7] J. Lu, H. W. Engl, and P. Schuster, "Inverse bifurcation analysis: Application to simple gene systems," *Algorithms for Molecular Biology*, vol. 1, pp. 1–16, 7 2006.

[8] I. Dobson, "Distance to Bifurcation in Multidimensional Parameter Space: Margin Sensitivity and Closest Bifurcations," in *Bifurcation Control: Theory and Applications*, pp. 49–66, Springer, Berlin, Heidelberg, 4 2004.

[9] V. Chickarmane, S. R. Paladugu, F. Bergmann, and H. M. Sauro, "Bifurcation discovery tool," *BIOINFORMATICS APPLICATIONS NOTE*, vol. 21, no. 18, pp. 3688–3690, 2005.

[10] E. D. Conrad, J. Tyson, R. Laubenbacher, J. Phillips, and M. Renardy, "Bifurcation Analysis and Qualitative Optimization of Models in Molecular Cell Biology with Applications to the Circadian Clock," *Virginia Tech*, 4 2006.

[11] I. Otero-Muras and J. R. Banga, "Optimization-based prediction of fold bifurcations in nonlinear ODE models," *IFAC-PapersOnLine*, vol. 51, pp. 485–490, 1 2018.

[12] I. Otero-Muras, P. Yordanov, and J. Stelling, "A method for inverse bifurcation of biochemical switches: inferring parameters from dose response curves," *BMC Systems Biology*, vol. 8, p. 114, 2014.

[13] C. Rackauckas and Q. Nie, "Differentialequations.jl - a performant and feature-rich ecosystem for solving differential equations in julia," *Journal of Open Research Software*, vol. 5, no. 1, 2017.

[14] C. Rackauckas, Y. Ma, V. Dixit, X. Guo, M. Innes, J. Revels, J. Nyberg, and V. Ivaturi, "A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions," *arXiv preprint arXiv:1812.01892*, 2018.

[15] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. Ramadhan, "Universal Differential Equations for Scientific Machine Learning," *arXiv*, 2020.

[16] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," *NIPs*, vol. 109, pp. 31–60, 6 2018.

[17] C. Rackauckas, M. Innes, Y. Ma, J. Bettencourt, L. White, and V. Dixit, "DiffEqFlux.jl - A Julia Library for Neural Differential Equations," *ArXiv*, 2019.

[18] S. Ranciati, C. Viroli, and E. Wit, "Bayesian Smooth-and-Match strategy for ordinary differential equations models that are linear in the parameters," *ArXiv*, 2017.

[19] F. Khadivar, I. Lauzana, and A. Billard, "Learning dynamical systems with bifurcations," *Robotics and Autonomous Systems*, vol. 136, p. 103700, 2 2021.

[20] A. Look, S. Doneva, M. Kandemir, R. Gemulla, and J. Peters, "Differentiable Implicit Layers," *ArvXiv*, 10 2020.

[21] S. Bai, J. Z. Kolter, and V. Koltun, "Deep Equilibrium Models," *arXiv*, 9 2019.

[22] P. E. Farrell, C. H. L. Beentjes, and Ã. Birkisson, "The computation of disconnected bifurcation diagrams," *arXiv*, 3 2016.

[23] R. Veltz, "BifurcationKit.jl," tech. rep., Inria Sophia-Antipolis, 7 2020.

[24] M. Haragus and G. Iooss, *Local Bifurcations, Center Manifolds, and Normal Forms in Infinite-Dimensional Dynamical Systems*. London: Springer London, 2011.

[25] Z. Drmač and Z. Bujanović, "On the Failure of Rank-Revealing QR Factorization Software – A Case Study," *ACM Transactions on Mathematical Software*, vol. 35, 7 2008.

[26] S. Jos and R. Schmidt, "On the velocity of an implicit surface," *ACM Transactions on Graphics*, vol. 30, pp. 1–7, 5 2011.

[27] J. Revels, M. Lubin, and T. Papamarkou, "Forward-Mode Automatic Differentiation in Julia," *ArvXiv*, 2016.

[28] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable Modelling with Flux," *CoRR*, vol. abs/1811.01457, 2018.

[29] M. Innes, “Flux: Elegant Machine Learning with Julia,” *Journal of Open Source Software*, 2018.

[30] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv*, 12 2014.

[31] H. Flanders, “Differentiation Under the Integral Sign,” *The American Mathematical Monthly*, vol. 80, p. 615, 6 1973.

[32] R. Goldman, “Curvature formulas for implicit curves and surfaces,” in *Computer Aided Geometric Design*, vol. 22, pp. 632–658, Elsevier, 10 2005.

[33] M. Tao, J. Solomon, and A. Butscher, “Near-Isometric Level Set Tracking,” *Computer Graphics Forum*, vol. 35, pp. 65–77, 8 2016.

[34] M. Fujisawa, Y. Mandachi, and K. T. Miura, “Calculation of Velocity on an Implicit Surface by Curvature Invariance,” *Information and Media Technologies*, vol. 8, no. 4, pp. 674–680, 2013.

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] Section 4 concludes that the implementation only covers co-dimension one bifurcations, with straightforward extensions to limit cycles, but does not scale well for partial differential equations
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We believe this work is sufficiently small scale and abstract to have any direct negative social impacts
   (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? [N/A] In Section 2 and the Appendix we apply existing theorems and approaches to differentiate our cost function
   (b) Did you include complete proofs of all theoretical results? [N/A] While we do not have any theorems to prove, we include derivations in the Appendix to guide readers who are more theoretically inclined
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] Figures can be reproduced using unit tests in the GitHub repository for `BifurcationFit.jl`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Section 3 we details of optimisations with `Flux.jl`
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Section 3 we include error bars for time complexity estimates. We did not think that parameter uncertainties were relevant for the scope of this work
   (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)? [Yes] The caption of Figure 5 has the specifications of the laptop used for complexity measurements
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] `BifurcationKit.jl` [23] for continuation, `ForwardDiff.jl` [27] for automatic differentiation and `Flux.jl` [28, 29] for optimisation
   (b) Did you mention the license of the assets? [No] MIT Licences
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Julia package implementing our method `Julia package in attached zip file`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No data was collected
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No data was collected
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]