

Parameter Reconstruction for Biochemical Networks Using Interval Analysis

WARWICK TUCKER

Department of Mathematics, Uppsala University, Box 480, Uppsala, Sweden,
e-mail: warwick@math.uu.se

and

VINCENT MOULTON

School of Computing Sciences, University of East Anglia, Norwich, Norfolk, UK
Abstract. Most optimization algorithms will rarely produce correct parameter values. Instead, almost all methods available utilize genetic/evolutionary algorithms to perform the non-linear parameter fitting. We propose a completely deterministic approach, which is based on interval analysis. This allows us to examine entire sets of parameters, and thus to exhaust the global search within a finite number of steps. The proposed method

the target system $\dot{x} =$

$f(x; p)$, see Figure 1. A trajectory of a d -dimensional system, sampled at N distinct times (excluding the initial point, which is assumed to be known at time t_0), produces the data $\{x(t_j)\}_j^N$

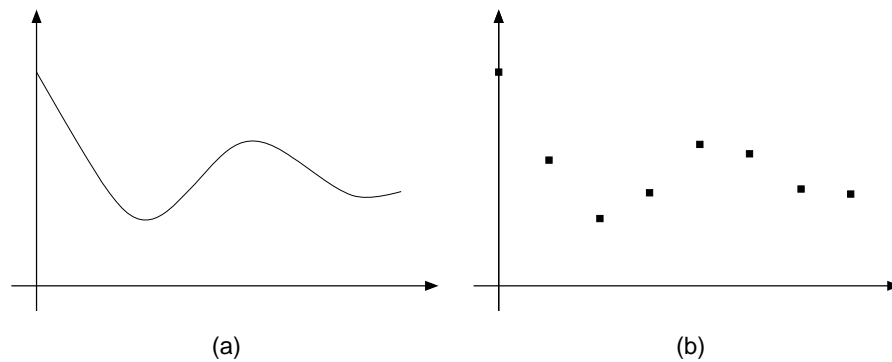


Figure 1. (a) One component of a trajectory. (b) Its sample data.

In what follows, we will use the short-hand notation $x = x_i(t_j)$. We will also assume that the problem of reconstructing the parameter is determined, that is, the cardinality of the data set is greater than the number of parameters to be determined.

1.1. SIMULTANEOUS RECONSTRUCTION VIA ODE SOLVING

Given a set of samples, one approach is to try to locate a point that minimizes the data error:

$$E^{(\text{data})}(p) = \sum_{i=1}^d E_i^{(\text{data})}(p) \stackrel{\text{def}}{=} \sum_{i=1}^d \sum_{j=0}^N |x_i(t_j; p) - x_{ij}|, \quad (1.1)$$

for some convenient metric (\cdot, \cdot) . Here, x solves the differential equation:

$$\frac{d}{dt} x(t; p) = f(t, x; p), \quad x(0; p) = x, \quad (1.2)$$

and can thus be approximated by numerical means, given the parameters. This approach is expensive seeing that, for each of the entire system of differential equations must be solved to compute the component errors $E_i^{(\text{data})}(p)$.

1.2. COMPONENTWISE RECONSTRUCTION VIA SLOPES

Rather than attempting to reconstruct the parameter by solving the entire system of differential equations, it may prove wiser to obtain more detailed information localized at the individual sample points. One possibility is to use the samples to reconstruct the trajectories (e.g. via piece-wise splines) with some degree of smoothness. This enables us to compute an approximation of the vector "eld at each sample point:

$$s_{ij} = f_i(t_j; p), \quad i = 1, f, d; \quad j = 0, f, N. \quad (1.3)$$

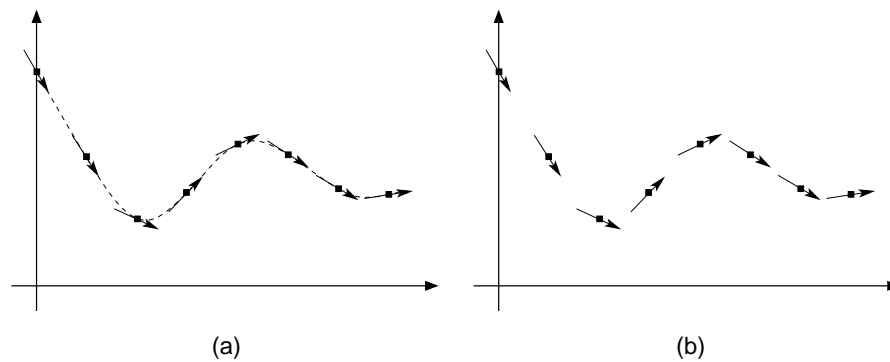


Figure 2 (a) One component of a trajectory. (b) enhanced sample data.

The number s_j corresponds to the slope of the trajectory's component at time t_j , see Figure 2.

Equipped with this enhanced sample data, we can try to locate a point P that minimizes the slope error:

$$E^{(\text{slope})}(p) = \sum_{i=1}^d E_i^{(\text{slope})}(p) \stackrel{\text{def}}{=} \sum_{i=1}^d \sum_{j=0}^N \tau_i(x(t_j); p, s_j), \quad (1.4)$$

for some convenient metric (\cdot, \cdot) . The major advantage with this approach is that the system decouples, that is, the computation of each $E_i^{(\text{slope})}(p)$ depends only on a fraction of the total number of parameters. $E^{(\text{slope})}(p) = E_i^{(\text{slope})}(p_i)$, where $p_i \in P_i$, and $P = P_1 \times \dots \times P_d$.

Assuming that each p_i has at most k non-zero components, the total dimension of the entire search space is dk . Rather than searching through the d -dimensional space, access to the enhanced sample data allows us to perform d independent searches in k -dimensions. The gain is immediate: introducing M grid-points in each component produces M^d points in the "rst case, but only dM^k points in the latter. This gives a speed-up factor of M^d / d .

Remark 1.1 As we only use approximations $s_j + \tau_i(x(t_j); p)$, we can not claim to be able to reconstruct the target parameter p with mathematical rigor. This limitation, however, is mild compared to the underlying assumption that the true trajectories are well-described by the data $\{x(t_j)\}_{j=0}^N$. In practice, given a reasonable amount of data, a good approximation of the target parameter is obtained. Poorly estimated slopes s_j will result in our algorithm dismissing all parameters at a very early stage, and are thus easily spotted.

Remark 1.2 When decoupling the system, information whether several components of the vector "eld share a common parameter is lost. Although it is possible to re-inject this information in the parameter reconstruction (e.g. communication between

parallel processes), it is not a trivial task. We use no such information in our algorithm.

1.3. INTERVAL-VALUED SLOPES

Our approach is a modification of the enhanced data method, and therefore shares the same attractive decoupling property of the system, as described above. The major improvement is that we now compute ranges of slopes for entire domains of parameters. In essence, we extend the vector field to a set-valued function, accepting solid boxes in parameter space as input. The mathematical justification for this type of extension is based on the theory of interval analysis. For a concise reference on this topic, see e.g. [1], [7]...[10]. For early papers, see [12], [15], and [16].

Let $[p_i]$ denote a box in P_i , that is, each component of $[p_i]$ is an interval. Then, for any point $p_i \in [p_i]$, we have

$$f_i(x(t_j); p_i) \in F_i(x(t_j); [p_i]), \quad (1.5)$$

that is, the set $F_i(x(t_j); [p_i])$ contains all possible slopes corresponding to parameters taken from the box $[p_i]$. This fact gives us a simple criterion for discarding portions of the search space: if a box $[p_i]$, at a sample point $x(t_j)$, produces a range of slopes such that $\dot{x}_j \notin F_i(x(t_j); [p_i])$, then no parameter in $[p_i]$ can have generated the sample data. If this situation occurs, we say that the parameter p_i violates the cone condition at time t_j , see Figure 3.

Our strategy in reconstructing the target parameter is to adaptively partition each space P_i into successively smaller sub-cubes, retaining only those who satisfy the cone condition at all times. At some level of resolution, we terminate the process, and are left with a collection of boxes $[p_i^{(1)}], \dots, [p_i^{(n)}]$, each of which satisfies $\text{Cone}([p_i^{(j)}]) = \text{true}$, where

$$\text{Cone}([p_i]) \stackrel{\text{def}}{=} \bigcap_{j=0}^N S_j(x(t_j); [p_i]) \quad (1.6)$$

is a boolean function that returns true if $[p_i] \subset P_i$ satisfies the cone condition at all sample times, and false otherwise.

The strategy of dismissing subsets that are inconsistent with some constraint lies at the heart of many interval methods. In the context of ODEs, see e.g. [3] and references within.

In Section 2, we will introduce a particular family of differential equations whose parameters we will reconstruct using the method described above.

12. Sunaga, T.: Theory of an Interval Algebra and Its Application to Numerical Analysis, *RAAG Memoirs* 2 (1958), pp. 29–46.
13. Voit, E. O.: *Computational Analysis of Biochemical Systems*, Cambridge University Press, 2000.
14. Voit, E. O. and Almeida, J.: Decoupling Dynamical Systems for Pathway Identification from Metabolic Profiles, *Bioinformatics* 20(11) (2004), pp. 1670–1681.
15. Warmus, M.: Calculus of Approximations, *Bulletin de l'Academie Polonaise de Sciences* 4:5 (1956), pp. 253–257.
16. Young, R. C.: The Algebra of Multi-Valued Quantities, *Mathematische Annalen* 104 (1931), pp. 260–290.

Table 6. The parameter values (and their reconstructions) of the S-system (4.5).

	i	g_1	g_2	g_3	i	h_{i1}	h_{i2}	h_{i3}
Original								
1	7.5	"	Š0.1	Š0.05	5.0	0.5	"	"
2	2.0	0.5	"	"	1.44	"	0.5	"
3	3.0	"	0.5	"	7.2	"	"	0.5
Reconstructed								
1	7.49	"	Š0.100	Š0.0503	4.99	0.501	"	"
2	2.00	0.501	"	"	1.44	"	0.502	"
3	3.00	"	0.500	"	7.20	"	"	0.500

Table 7. The computational effort for the "xed-topology S-system (4.5) with $\text{tol} = 1 \times 10^{-3}$.

component	boxes	F-evaluations	CPU-time
1	905	83,595	54s
2	34	11,073	7s
3	83	8,733	6s

$$\begin{aligned}
 x_1 &= 7.5x_2^{\text{Š}0.1} x_3^{\text{Š}0.05} \text{Š} 5x_1^{0.5}, \\
 x_2 &= 2x_1^{0.5} \text{Š} 1.44x_2^{0.5}, \\
 x_3 &= 3x_2^{0.5} \text{Š} 7.2x_3^{0.5}.
 \end{aligned} \tag{4.5}$$

This particular model is treated in [13], and differs from the two previous examples in that we are given, a priori, the network topology. This reduces the computational complexity significantly.

In Table 6, we present the target parameters together with the "nal result of our reconstruction. The reconstructed parameter values are simply the midpoint of the average over all parameter boxes produced by our search. We use the notation "Š" to indicate a non-present parameter.

For the computations, we used "ve sets of initial conditions, and each trajectory was sampled at 20 points in time. The search region for each of the kinetic orders g_{ij} , and h_{ij} was set to contain Š[1, +1], whereas the rate orders i and j were sought for within [0 15]. The stopping tolerance was set to 10^{-3} . Once again, the agreement is seen to be a good match. The computational effort is presented in Table 7.

5. Conclusions

We have presented a novel method for reconstructing parameters using interval analysis. In particular, we have applied it to reconstruct metabolic networks using S-systems, and obtained encouraging results. We stress that the proposed method

is quite general, and can (in principle) be applied to any system of "nitely parameterized differential equations.

Our method differs in a fundamental way from the main-stream reconstruction methods in that we solve the problem by a pruning scheme based on a boolean function (the cone condition), rather than recasting the parameter reconstruction as a global minimization problem. This has several advantages: "rst, it is well-known that global minimization is an intractable problem, in the sense that numerical solutions often converge to a local, rather than a global, minimum, and there is no way of telling the two cases apart. Second, the quantity to be minimized is often chosen to be a (weighted) least-square error. This implicitly pre-assumes rather strong statistical properties of the underlying data, assumptions that can not easily be veri"ed. Our method simply discards the parameters that are inconsistent with the underlying data, avoiding both above-mentioned problems.

The transition to set-valued vector "elds also allows us to dismiss unrealistic network topologies. In particular, this allows us to detect when the model we are trying to "t to the provided data is not appropriate. With a suf"ciently low stopping tolerance, our method would then discard parameter values.

In future work, we will re"ne the process of parameter exclusion, and exploit the problem's great potential for parallelization. This is an essential step towards exploring the scalability of our proposed method. We will also allow for noisy sample data, using statistical pre-processing in the generation of the slopes. We also plan to put our method to test on a larger class of problems (including generalized mass action models).

References

1. Alefeld, G. and Herzberger, J. Introduction to Interval Computation, Academic Press, New York, 1983.
2. Alves, R. and Savageau, M. A.: Comparing Systemic Properties of Ensembles of Biological Networks by Graphical and Statistical Methods, *Bioinformatics* 16 (6) (2000), pp. 527...533.
3. Deville, Y., Janssen, M., and van Hentenryck, C.: Consistency Techniques in Ordinary Differential Equations, *Constraints* 7 (2002), pp. 289...315.
4. Hlavacek, W. S. and Savageau, M. A.: Rules for Coupled Expressions of Regulator and Effector Genes in Inducible Circuits, *Mol. Biol.* 255 (1996), pp. 121...139.
5. de Jong, H.: Modeling and Simulation of Genetic Regulatory Systems: A Literature Review, *J. Comp. Biol.* 9 (1) (2002), pp. 67...103.
6. Kikuchi, S. et al.: Dynamic Modeling of Genetic Networks Using Genetic Algorithm and S-System, *Bioinformatics* 19 (5) (2003), pp. 643...650.
7. Kulisch, U. W. and Miranker, W. L. Computer Arithmetic in Theory and Practice, Academic Press, 1981.
8. Moore, R. E. Interval Analysis, Prentice Hall, Englewood Cliffs, 1966.
9. Moore, R. E. Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.
10. Neumaier, A.: Interval Methods for Systems of Equations, *Encyclopedia of Mathematics and Its Applications* 37, Cambridge Univ. Press, Cambridge, 1990
11. PROFIL/BIAS, Programmer's Runtime Optimized Fast Interval Library/Basic Interval Arithmetic Subroutines, <http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html>

