Estimation of Parametric Nonlinear ODEs for Biological Networks Identification

Florence d'Alché-Buc, Nicolas Brunel

Estimation of parametric nonlinear ODEs for biological networks identification*

Florence d'Alché–Buc and Nicolas Brunel
IBISC fre CNRS 3190 , Université d'Evry-Val d'Essonne,
Genopole, Evry, FRANCE
Email: florence.dalche@ibisc.fr, nicolas.brunel@ibisc.fr

March 18, 2009

Abstract

Ordinary Differential Equations (ODEs) provide a theoretical framework for a mechanistic description of biological networks (*e.g.* signalling pathway, gene regulatory network, metabolic pathway) as continuous time dynamical systems. Relevant ODEs are often nonlinear because they are derived from biochemical kinetics and based on law of mass action and its generalizations or Hill kinetics. We present two approaches devoted to the identification of parameters from time-series of the state variables in nonlinear ODEs. The first approach is based on a nonparametric estimation of the trajectory of the variables involved in the ODE. The parameters are learned in a second step by minimizing a distance between two estimates of the derivatives. In the second approach, dedicated to Bayesian estimation, we build a nonlinear state-space model from the ODEs and we estimate both parameters and hidden variables by approximate nonlinear filtering and smoothing (performed by the unscented transform).The two approaches are illustrated on numerical examples and discussed.

1 Introduction

Reverse-modeling of biological networks such as gene regulatory networks, signaling pathways or metabolic pathways has recently witnessed a surge of interest due to the widespread availability of large scale measurement techniques. From over 20 years now, many mathematical models including discrete-time models and continuous time ones, deterministic models and stochastic ones, have been used to study and analyze the dynamical behavior of biological networks. Among all these frameworks, Ordinary Differential Equations (ODEs) are certainly one of the most powerful ones, providing a theoretical framework for the

*Draft version of a chapter to appear in Learning and Inference in Computational Systems Biology, N.Lawrence, M. Girolami, M. Rattray, G. Sanguinetti, MIT press, to appear in 2009.

description of biological networks as continuous time dynamical systems. ODEs are usually employed in systems biology to describe in a deterministic way how components in the cell interact with each other, offering either a very precise mechanistic view of the various components such as Michaelis-Menten equations or a more general idea of regulation through some generic model of interaction such as linear models proposed in D'Haeseleer et al. [14], generalized linear models studied in [9] and S-systems introduced and studied by [39, 49]. Once a given family of ODEs is chosen and when time series of state-variable observations are available, reverse-modeling of biological networks becomes possible. This identification task then boils down to the estimation of the ODEs' structure and parameters. In case of some simple models, such as linear ones, the structure is explicitly encoded into the parameters and the estimation of all parameters provides thus the graph of interactions, assuming that the model complexity is controlled either by the use of appropriate penalizations such as encountered in ℓ_1 regularized criteria and AIC or BIC criteria. However, in some cases, for instance for complex models like Michaelis-Menten or Hill equations, the graph of interactions has to be given before the definition of the ODE because learning a semiparametric model that encompasses various structures would be too difficult. In this case, structure inference and parameters estimation can be solved by the use of two coupled methods: a structure learning method and a parameter learning method. The structure learning method explores the space of graphs and selects good candidates with a score based on the estimated parameter. Assuming that parameter estimation can be combined to structure estimation algorithms like those developed in [4], we focus on the sole problem of parameter estimation. Regarding this last issue, least-squares methods developed so far provide solutions but require the solution of the ODEs and face difficult global optimization problem depending on implicit definition of the cost function. Meanwhile, the noisy nature of available data and their limited number, as well as their potential incompleteness, leads us to explore other learning paradigms that encompass these constraints. In this chapter, we present two approaches inspired from two different views of the estimation problem, each presenting their own advantages. The first approach is based on a two-step estimation procedure of the parameters of the ODE. In a first step, a nonparametric estimation of the trajectory of the variables involved in the ODE is made. Then, parameters are learned in a second step by minimizing the distance between the derivative of the approximation obtained in the first step and the derivative estimated from the parametric vector field defining the ODE.

In the second approach, assuming that the true biological process is not fully observed, we build a nonlinear state-space model from the ODEs and we estimate both parameters and hidden variables by nonlinear Bayesian filtering and smoothing.

The first approach gives more importance to the nature of the ODE's (approximated) solution, easily allowing us to incorporate some qualitative constraint on its shape whereas the second approach benefits from the probabilistic framework of graphical models in which hidden variables can be easily modeled.

The chapter is structured as follows. First we introduce in Section 2 examples of ODEs used in systems biology, taking the examples of gene regulatory networks and signaling pathways. Then we present the issues of statistical learning of ODEs. This introduces the next part of the paper devoted to the two-step estimation procedure (Section 3), and its properties. We show also some refinements for the use of prior qualitative knowledge. In Section 4, we describe a class of state-space models defined from nonlinear ODEs. We then present the estimation of parameters and hidden state in the framework of Bayesian inference. In order to overcome the difficulty induced by the nonlinearity of the studied process, filtering and smoothing algorithms based on the *unscented transform* introduced in [21] are used to implement the Bayesian inference. Then, we discuss in Section 5 the differences between the two estimation procedures presented in 3 and 4, which rely on two points of view, each one bringing its own advantages. Finally, we evocate further possible improvements for both approaches.

2 Modeling biological systems with Ordinary Differential equations

Ordinary Differential Equations (ODEs) have been mainly developed to represent biochemical networks that involve interactions between various chemical species. First defined for metabolic reaction networks, ODEs have also been used to describe gene interactions in gene regulatory networks. Biologically relevant ODEs reflect nonlinearities that occur in biological systems in which saturated signals for instance are observed. Usually derived from biochemical kinetics, they encompass models like laws of mass action, Hill kinetics as previously introduced in **??**. When the graph of interactions is known, reverse-modeling of the network boils down to the estimation of the parameters indexing the nonlinear ODEs. In this modeling framework, a biological system that involves p species is described by a system of p coupled equations. In first-order differential equations, the i^{th} equation expresses how some species concentrations affect the evolution of the i^{th} species at any time t. Here, we do not consider delays and we restrict ourselves to the study of first-order ODEs.
A system of p coupled differential equations can be written as follows:

$$\dot{X}_t = f(X_t, t, \theta),\ \forall t \in [0,1], \tag{1}$$

where f is a time-dependent vector field from $^p\times$ to ^{p}, $p \in$ and $\theta \in \Theta$, Θ being a subset of a ^{d}, where d is the number of parameters. Moreover, the vector field $f\ :\ \mathcal{X} \times [0,1] \times \Theta \to^p$ $(\mathcal{X} \subset^p)$ is a smooth function of class C^m w.r.t X and θ, $m \geq 1$. This smoothness condition ensures then the existence and uniqueness of a solution for given initial values $X_0 \in \mathcal{X}$ on a neighborhood of 0 for each θ. We consider that the solution X_t that represents the vector of the concentrations of the p species at time t, exists on $[0,1]$ and is itself a smooth function of degree $m+1$ in θ and X_0 (the reader can refer to [19] for deeper details about these assumptions). So, a solution X_t is in fact indexed by a parameter vector, $X_t(\eta)$, where the parameter vector is given by $\eta = (X_0, \theta) \in \Theta \times \mathcal{X}$, but most of the time, one is interested only in the parameters θ which characterize the network under study, and one is not so much interested in the initial value X_0. The latter can be considered as a nuisance parameter during the estimation, that corresponds to the experimental conditions (possibly random).

2.1 Modeling transcriptional regulatory networks with Hill equations

In transcriptional regulatory networks, variables of interest are mRNA and protein concentrations, denoted respectively by m_i and $p_i, i = 1, \ldots, d$ (p equals $2d$ in this case). Let us make the assumption here that one gene can only produce one protein. We consider transcription and translation as dynamical processes, in which the production of mRNAs depends on the concentrations of protein transcription factors (TFs) and the production of proteins depends on the concentrations of mRNAs. Hence, we have $X_t = (\mathbf{m}(t)^\top, \mathbf{p}(t)^\top)^\top$ with $\mathbf{m}(t) = (m_1(t), ..., m_d(t))^\top$ and $\mathbf{p}(t) = (p_1(t), ..., p_d(t))^\top$. Equation (1) can be split into the following equations. Transcription is described as

$$\frac{\mathrm{d}m_i(t)}{\mathrm{d}t} = g_i(\mathbf{p}(t)) - k_i^g m_i(t), \tag{2}$$

whilst translation may be modeled as

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = k_i m_i(t) - k_i^p p_i(t). \tag{3}$$

where k_i^g and k_i^p are respectively the degradation rates of mRNA i and protein i. The function g_i describes how TFs regulate the transcription of gene i and equation (3) describes the production and the degradation of protein i as linear functions where k_i is the translational constant for gene i [9].

Various forms have been proposed for $g_i(\mathbf{p})$, like linear approaches as described in [9] and nonlinear approaches presented in [16, 43, 13, 28, 11]. Experimental evidence has suggested that the response of mRNA to TFs concentrations has a *Hill curve* form [13, 16]. The reader may find a detailed presentation of Michaelis-Menten kinetics as well as Hill kinetics in Chapter **??**. The regulation function of transcription factor p_j on its target gene i can be described by

$$g^+(p_j; v_i, k_{i,j}, h) = v_i \frac{p_j^h}{k_{i,j}^h + p_j^h}$$

for the activation case and

$$g^-(p_j; v_i, k_{i,j}, h) = v_i \frac{k_{i,j}^h}{k_{i,j}^h + p_j^h}$$

for the inhibition case. Here v_i is the maximum rate of transcription of gene i, $k_{i,j}$ is the concentration of protein p_j at which gene i reaches half of its

[width=0.49]chapters/BrunelDalcheBuc/ElowitzStructure.pdf
[width=0.49]chapters/BrunelDalcheBuc/jakstatstruct2.pdf

Figure 1: *Left*: Repressilator. The first repressor protein, LacI inhibits the transcription of the second repressor gene TetR whose protein product in turn inhibits the expression of a third gene cI. Finally, CI inhibits lacI expression, completing the cycle. *Right*: JAK-STAT signaling pathway. JAK protein binds to the Erythropoietin Receptor (EpoR) and causes the phosphorylation of STAT5 protein. Phosphorylated STAT5 protein then forms a dimer and moves into the nucleus. In the nucleus, phosphorylated STAT5 dimer is dephosphorylated and forms a STAT5 monomer, which finally goes back to the cytoplasm.

maximum transcription rate and h is a steepness parameter describing the shape of sigmoid responses. The parameter vector $\theta = (v_i, k_{i,j}, k_i^g, k_i, k_i^g, h)$ for $i, j = 1, ..., d$ is the set of kinetic constants to be estimated. Note that if a gene has several regulators, the regulatory part of the equation (2) can be extended into a product of functions g^+ and g^- that expresses the combined effect of regulators. However, we consider here examples where the genes have only one regulator.

The complexity of the dynamics that can be described by such equations is highlighted in the example of the repressilator, a synthetic network based on three transcriptional repressors that was implemented by Elowitz and Leibler in 2000 (see [16]) to implement some desired dynamical behavior (e.g. sustained oscillations) as illustrated in Figure 1. This system was also built experimentally by genetic engineering with mutated *E. coli* strains. Despite the simplicity of the transcriptional regulation model, the negative feedback loop implemented by the three genes that act as three inhibitors leads to oscillating concentrations confirmed by experiments. The kinetics of the system can be described by six coupled ODEs which exactly fit the framework previously described:

$$\frac{\mathrm{d}m_1(t)}{\mathrm{d}t} = g_1^-(p_2; v_1, k_{1,2}, h) - k_1^g m_1(t) \quad (4)$$

$$\frac{\mathrm{d}m_2(t)}{\mathrm{d}t} = g_2^-(p_3; v_2, k_{2,3}, h) - k_2^g m_2(t) \quad (5)$$

$$\frac{\mathrm{d}m_3(t)}{\mathrm{d}t} = g_3^-(p_1; v_3, k_{3,1}, h) - k_3^g m_3(t) \quad (6)$$

$$(7)$$

The equations for each protein concentration, for $i = 1, \ldots, 3$ remain exactly under the form of equation decsribed in 3.

2.2 Modeling signaling pathways: the JAK-STAT example

Signaling pathways are other candidates for ODEs modeling. They usually involve numerous and various intermediate products in a complex sequence of transformations. Depending on the types of signals and intermediary components, and the localization of the pathways, there exist several relevant types of ODEs. Consequently, it seems rather difficult to give the same wide picture as for transcriptional regulatory networks, but most of the time we can say that the system of ODES involves nonlinear reaction rates derived from mass action law and Michaelis-Menten (or Hill) kinetics. We provide here a description of the JAK-STAT signaling pathway involved in the cellular response to cytokines and growth factors, which involves Janus kinases (JAKs) and Signal Transducers and Activators of Transcription (STATs), see the graph on the right of Figure 1. This pathway transduces the signal carried by these extracellular polypeptides to the cell nucleus, where activated STAT proteins modify gene expression. In both cases, there may be some difficulties in observing the variables of the pathway, and this gives rise to different observation functions from gene regulatory networks. This is particularly emphasized in the JAK-STAT pathway, for which it is difficult to discriminate between several intermediates in the pathway. In [45], Swameye et al. suggested an ODE linking the Erythropoietin receptor (EpoR) to the various forms of the STAT5 protein such as dephosphorylated STAT5 monomer (x_1) and phosphorylated STAT5 dimer (x_2) in the cytoplasm, phosphorylated STAT5 dimer (x_3) and STAT5 monomer (x_4) in the nucleus. Another variable of interest is the concentration of EpoR which is considered as an exogenous variable of the system. Finally, as proposed by Zi and Klipp [51], the evolution of this network can be described by the following system of coupled differential equations with an input variable $u(t)$ (EpoR), which can be considered as an adaptation of the system proposed by Swameye et al. [45] :

$$\begin{aligned}
\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} &= -a_1 x_1(t)u(t) + 2a_4 x_4(t) 1_{\{t\geq\tau\}} \\
\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= a_1 x_1(t)u(t) - 2a_4 x_2^2(t) \\
\frac{\mathrm{d}x_3(t)}{\mathrm{d}t} &= -a_3 x_3(t) + x_2^2(t) \\
\frac{\mathrm{d}x_4(t)}{\mathrm{d}t} &= a_3 x_3(t) - a_4 x_4(t) 1_{\{t\geq\tau\}}
\end{aligned} \tag{8}$$

where $1_{\{t\geq\tau\}}$ denotes the indicator function, equals to 0 for $t \leq \tau$ and equals to 1 otherwise. The concentrations and constants $a_i, i = 1, 3, 4$ in (8) stand for normalized quantities as described in [50]. The vector $\theta = (a_1, a_3, a_4)^\top$ contains the parameters to be estimated. As pointed out by Swameye2003, the individual STAT5 population is difficult to access experimentally, and only the following variables could be measured: $y_1 = (x_2 + 2x_3)$, the concentration of phosphorylated STAT5 in the cytoplasm and $y_2 = (x_1 + x_2 + 2x_3)$, the total amount of STAT5 in the cytoplasm. As we shall see in the section **??**, the estimation of such a system in the context of hidden variable fits the framework of state-space models based on ODEs.

2.3 Statistical learning of ODEs with constraints and hidden variables

As previously noted, we focus on the estimation of θ (and not of X_0, nor of the structure). We now consider the problem of learning parameters θ of the equations in (1) from data. We can first make an simple assumption: the data are simply noisy observations of the states, *i.e.* $Y_{t_0}, \ldots, Y_{t_T}$ with

$$Y_{t_i} = X_{t_i}(\eta) + \epsilon_i,\; i = 0, \ldots, T, \tag{9}$$

and $0 \leq t_0 < \ldots < t_T \leq 1$ are $T+1$ observation times in $[0,1]$. The random variables ϵ_i, $i = 0, \ldots, T$ are observation noise, and we suppose that they are simply spherical Gaussian independent variables $(0, \sigma^2 I_p)$. The observation equation (9) corresponds to the particular case where the states X_t are observed, but in some situations the system can only be partly observed as for the Repressilator or the JAK-STAT pathway described previously. Hence, generally, we have the following observation equation

$$Y_{t_i} = H(X_{t_i}(\eta)) + \epsilon_i,\; i = 0, \ldots, T, \tag{10}$$

where $H :^p \rightarrow^d$ is a smooth (possibly nonlinear) function with $1 \leq d \leq p$.

From (9) or (10), it is clear that the estimation of η or θ corresponds to a classical problem of multivariate parametric nonlinear regression. We know that the Maximum Likelihood Estimator (MLE) or Least Squares Estimator are good statistical estimators in this case, with desirable properties such as consistency and efficiency. Consequently, if we suppose that identifiability problems (due to partial observation or over-parameterization of the model) are ruled out, the estimation of η boils down to the following optimization problem :

$$\hat{\eta} = \arg \min_{\eta \in \Theta \times^d} \sum_{i=1}^{T} \left\| Y_{t_i} - X_{t_i}(\eta) \right\|^2. \tag{11}$$

In the remaining part of the chapter, $\eta^* = (X_0^*, \theta^*)$ denotes the "true" parameter of the ODE and X_t^* the corresponding solution. There exists many variants of this estimation problem. Examples include special boundary values instead of the simple initial value problem (a function $z(\cdot)$ links the values at the boundary *i.e.* $z(X_0, X_1) = 0$), random initial values or random parameters [15], or noisy observation times [23]. Nevertheless, the fundamental difficulty of the estimation of ODEs already appears in the simple setting we have just presented and lies in the implicit definition of the model. Indeed, the least squares criterion in (11) can be only computed by numerically solving the system of ODEs for a given set of parameter η, which can be computationally prohibitive. Moreover, the Jacobian $\nabla_\eta X_t$ does not have a closed-form expression, and must be computed by solving the first variational equations (or sensitivity equations).

In case of a large number of parameters such that it could be encountered in mechanistic models, the corresponding optimization problem may be hard to solve. The task becomes even harder because the least-squares criterion possesses numerous local minima, as it has been emphasized by [33]. As a matter of fact, one of the best estimation methods proposed so far relies on more global optimization algorithm, that enables a more efficient exploration of the parameter space for η, see for instance Chapter **??**, [27]. Despite their satisfactory theoretical properties, the efficiency of the MLE may be dramatically degraded in practice by computational problems that arise from the implicit and nonlinear definition of the model. If regularization can help in solving the issue of model complexity, the implicit definition of the model still makes the learning task difficult.

Finally, the main difficulty of the estimation of η, whatever the method, is related to the notion of parameter identifiability. In a brief setting, a parameter of a dynamical system is said to be identifiable given some data if only one value of this parameter can produce the observed behavior. Although there exist practical tools like sensitivity analysis to study parameters identifiability given a dynamical system, to our knowledge very few estimation methods take into account information about identifiability, within the learning process. A recent work of De Pauw et al. [29] exploits such constraint in the parameters space exploration using evolutionary algorithms.

Related to the issue of identifiability, the analysis of the qualitative behavior of the dynamical system under study could surely be helpful in the estimation process. Although models are usually built to be able to explain some particular qualitative behavior (sustained oscillations or convergence to an equilibrium point), the link between the parameters η and the shape of the functions $X_t(\eta)$ is hard to decipher and thus never used in the estimation process. Classical tools of dynamics analysis come from bifurcation theory (see for instance [11, 22]), but they remain devoted to small systems.

The two methods we present respectively in Sections 3 and 4 can be considered as alternative solutions to the complex optimization task described previously. They both take into account explicitly the fact that $X_t(\eta)$ is the solution of an ODE. The first one, called two-step estimation, uses a direct and global reconstruction of the unobserved solution with nonparametric estimation. We give a detailed account of the theoretical properties of two step estimators because they are relatively unknown and do not share the same properties as other parametric estimators such as MLE. This will also motivate potential improvements for practical implementation by using qualitative constraints. A second estimation method is based on the incorporation of a system of ODEs in the definition of a state-space model, enabling the estimation of both parameters and hidden variables using recursive inference algorithms. These methods are well-known in machine learning, but they are not commonly used in the context of ODE estimation.

3 Learning with two-step estimators

As we have previously noted, the only difficulty with respect to the regression setting is that the function $X_t(\eta)$ (solution of the ODE for parameter η) needs to be computed numerically, which causes important difficulties for exploring the sets of (parametric) candidate functions. The idea is to replace the function $X_t(\eta)$ by a function close to the data (and to the true solution), that can be computed easily from the data. At the same time, we expect from the nonparametric approach enough versatility for the incorporation of constraints.

3.1 Rationale and consistency

We present here the idea of the two-step estimator and we give also a relatively detailed description of its statistical mechanism, as it relies on different ideas of the Maximum Likelihood Estimator or Bayesian estimators. Whereas Bayesian estimators are self-justified (they minimize the Bayesian risk), the quality of frequentist estimators is usually assessed by controlling that they converge to the true parameter as the number of observations T tends to infinity. In particular, the MLE ($\hat{\theta}_{MLE}$) is a consistent estimator because it converges to the true parameter θ^* for a wide range of model as we get more and more observations [8]. Moreover, the rate of convergence of the MLE (and its asymptotic distribution) is known and can be written on the following form

$$\sqrt{T}\left(\hat{\theta}_{MLE} - \theta^*\right) \sim (0, \mathcal{I}(\theta^*))$$

where $\mathcal{I}(\theta^*)$ is the so-called Fisher information of the model (9). This result is central for the construction of confidence sets or for the testing of statistical hypothesis. It also enables to quantify the rate at which the statistical procedure extracts information on the parameter from the observations. For parametric estimators, the classical rate is "in $\sqrt{T}$" and the MLE is usually considered as the best (asymptotically) estimator because it has the best attainable constant (or asymptotic variance) given by the Cramér-Rao bound. Nevertheless, the small sample properties of the MLE are not easy to grasp in general, and the heavy use of the MLE in applications comes partly of its simple definition. Another reason is that the rationale of the MLE is rather satisfying, as the maximization of the likelihood is equivalent to the minimization of the Kullback-Leibler distance between the estimated distribution and the true distribution. Since the definition of the two-step estimator is distribution-free, we describe its mechanism in order to explain why the two-step estimator remains consistent but can have an asymptotic behavior slightly different from the MLE. This latter comes from the fact that it uses functional estimators (which have different rates of convergence than parametric estimators).

When all the system is observed (*i.e.* (9) is satisfied), we can build on the theory of nonparametric regression and functional estimation to compute a proxy $\hat{X}_t$ for the solution of the ODE and its derivative $\dot{\hat{X}}_t$. Based on $\hat{X}_t$, we can look for parameters θ' such that $\hat{X}_t$ is nearly solution of the ODE $\dot{X}_t = f(X_t, t, \theta')$. This is possible because we can compute the derivative $\dot{\hat{X}}_t$, which is also an estimator of the true derivative, $\dot{X}_t^*$. Obviously, there is room for defining several notions of closeness to a solution of an ODE, but a rather straightforward one is to use the L_2 norm. More precisely, we propose the following two-step procedure (estimator):

Functional Estimation Estimate the ODE solution

1. For $j = 1, \ldots, p$, estimate $x_j^*(t)$ with a consistent estimator $\hat{x}_j(t)$ from observations $y_{j,0}, \ldots, y_{j,T}$
2. For $j = 1, \ldots, p$, estimate $\dot{x}_j^*(t)$ by differentiating $\hat{x}_j(t)$: $\hat{\dot{x}}_j(t) = \dot{\hat{x}}_j(t)$

Parameter Identification Solve the optimization problem

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \|\dot{\hat{X}}_t^* - f(\hat{X}_t^*, t, \theta)\|_{L_2}^2.$$

Such a procedure has already been proposed by [24], [48], [49, 10] essentially with splines, but they can be replaced with ones preferred nonparametric estimators to get a correct estimator. A detailed analysis of two-step procedures and of their potential improvements are discussed in [6].
The two-step estimator $\hat{\theta}$ is a generalization of the maximum Likelihood estimator: it is a M-estimator [46] *i.e.* a generalization of the maximum Likelihood estimator. Instead of minimizing the sums of squared errors (or maximizing the log-likelihood), we minimize

$$\mathcal{R}_{2,T}(\theta) = \|\dot{\hat{X}}_t - f(\hat{X}_t, t, \theta)\|_{L_2}^2$$

which is the empirical counterpart of the criterion

$$\mathcal{R}_2(\theta) = \|f(X_t^*, t, \theta) - \dot{X}_t^*\|_{L_2}^2,$$

where

$$\|z\|_{L_2}^2 = \int_0^1 \|z(t)\|^2 \mathrm{d}t.$$

This family of estimators do provide consistent statistical estimators under general conditions, and it has been studied in this particular setting in [6]. Obviously, the consistency of $\hat{\theta}$ relies heavily on the properties of $\hat{X}_t$:at least $\hat{X}_t$ must be chosen such that $\hat{X}_t$ and $\dot{\hat{X}}_t$ are consistent estimators of X_t^* and $\dot{X}_t^*$ in the L_2 norm. Moreover, a critical property needed for $\mathcal{R}_{2,T}(\theta)$ to be a good contrast is the following appropriate identifiability condition, that ensures that the minimum of the contrast is reached only for the true parameter θ^*:

$$\forall \epsilon > 0, \inf_{\|\theta - \theta^*\| \geq \epsilon} \mathcal{R}_2(\theta) > \mathcal{R}_2(\theta^*), \tag{12}$$

This criterion means roughly that there exists no other parameter θ' such that $\dot{X}_t(\eta) = f(X_t(\eta), t, \theta')$, but with the additional constraint that this equality must not hold approximately for some θ' far from θ^*. This condition is a bit more stringent than the classical identifiability that requires the map $\eta \mapsto X_t(\eta)$

to be one-to-one [2]. Even if the solution to this problem is still open, we can partially answer this problem by computing the Hessian of the asymptotic criterion $R(\theta)$ evaluated in θ^*:

$$J^* = \int_0^1 D_\theta f(X_t^*, t, \theta^*)^\top D_\theta f(X_t^*, t, \theta^*) \mathrm{d}t.$$

where $D_\theta f$ is the Jacobian of the vector field f with respect to θ. Hence, if J^* is nonsingular, one can derive a local identifiability criterion; indeed, $\mathcal{R}_2(\theta)$ behaves like a positive definite quadratic form on a neighborhood $\mathcal{V}(\theta^*)$ of θ^*, so that condition (12) is true on $\mathcal{V}(\theta^*)$. Obviously, this condition is hard to check in practice, but it provides a hint for detecting possible identifiability problems. We do not further address this aspect of the estimation problem and we shall consider now only the computational estimation problem.

Note that the minimization of the discrepancy between estimates of the derivatives has been exploited in another way by [34] in the functional data approach. This is based on the fact that smoothing splines are obtained by solving the trade-off between adequacy to data and smoothness of the solution as measured by linear differential operators. It was extended more recently in [33] to the case of nonlinear differential operators, and gives rise to a different cost function and a pragmatic method for parameter estimation.

3.2 Computational advantages

Now, we briefly describe the computational advantage derived from this modification of the cost function with respect to (11). The first gain is that we do not need now to solve the ODE for the computation of the criterion $\mathcal{R}_{2,T}(\theta)$, nor its minimization: this dramatically reduces the computational load of the estimation algorithm. Another significant gain concerns the complexity of the optimization task because we can decouple the estimation of p differential equations so that we reduce the size of the parameter space to explore. Indeed, let us decompose the equation (1)

$$\forall i = 1, \ldots, p, \ \frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = f_i(X_t, t, \theta_{[i]})$$

where $\theta_{[i]} \in \Theta_{[i]}$ denotes the subset of the parameters effectively involved in equation i ($\theta = \cup_{i=1}^p \theta_{[i]}$). If the parameters $\theta_{[i]}$ are non-overlapping, the effective optimization takes place in small spaces

$$\hat{\theta}_{[i]} = \arg \min_{\theta_{[i]} \in \Theta_{[i]}} \mathcal{R}^i_{2,N}(\theta).$$

where $\mathcal{R}^i_{2,T}(\theta) = \|f(\hat{X}_t, t, \theta) - \dot{\hat{X}}_t\|_2^2$. Finally, as we do not estimate the initial states, the size of the parameter set is dramatically decreased, as is the computation time. In this case we can easily compute the gradient in closed-form (for each dimension):

$$\forall i = 1, \ldots, p, \ \nabla_{\theta_{[i]}} \mathcal{R}^i_{2,T}(\theta) = \int_0^1 (f_i(X_t, t, \theta_{[i]}) - \dot{\hat{x}}_i(t)) \nabla_{\theta_{[i]}} f_i(X_t, t, \theta_{[i]})) \mathrm{d}t.$$

The two-step procedure gives then a consistent and computationally fast estimation method for the parameters of an ODE. We provide then in the next section an analysis of the rate of convergence.

3.3 Asymptotics

As we can see, the reason why a two-step procedure does work is different from a likelihood-based procedure such as the MLE. In [6], a local study of the criterion $\mathcal{R}_{2,T}$ enables to derive an asymptotic expansion for the two-step estimators similar to 3.1. Indeed, the criterion $\mathcal{R}_{2,T}$ is linearized thanks to a Taylor expansion of the vector field f around the true solution X_t^* and the true parameter θ^* and gives

$$\hat{\theta} - \theta^* = \int_0^1 A(t)\left(\hat{X}_t - X_t^*\right)\mathrm{d}t + \alpha\left(\hat{X}_1 - X_1^*\right) - \beta\left(\hat{X}_0 - X_0^*\right) \tag{13}$$

where $t \mapsto A(t)$ is a smooth matrix-valued function (in ${}^{p\times p}$) that depends on f and its derivatives (w.r.t. θ^* and X^*), and α, β are two matrices in ${}^{p\times p}$. The rate of $\hat{\theta}$ depends on the rates of convergence of the linear functionals:

Evaluation functionals $T_0(\hat{X}_t) = \hat{X}(0)$ and $T_1(\hat{X}_t) = \hat{X}_1$,

Smooth functional $\Gamma(\hat{X}_t)$ with $\Gamma(Z(t)) = \int_0^1 A(t)Z(t)dt$ (for any function $t \mapsto Z(t)$).

It is well known that the pointwise evaluation of a regression function cannot be done at a better rate than $O_P\left(T^{-\frac{\beta}{2\beta+1}}\right)$, where β is a measure of the smoothness of the function X^* [44] (in our case β is the degree of differentiability). If the function X^* is reasonably smooth, it can be shown that the smooth functional $\Gamma(\hat{X}_t)$ converges to $\Gamma(X^*)$ at a rate equals to $\sqrt{T}$ for wide families of nonparametric estimators (the so-called plug-in property). In particular, this is possible for Nadaraya-Watson estimators ([17]) or when the estimator is decomposed in a basis of function $\psi_{i,T}$, $i = 1, \ldots, K_T$ (with good approximation properties) i.e. $\hat{X} = \sum_{i=1}^{K_T} c_{i,T}\psi_{i,T}$ [1]. In particular, (cubic) splines fulfil these requirements. Moreover, it is possible to derive asymptotic normality of $\hat{\theta}$ at least for series-estimators, using results of [1]. Finally this detailed decomposition of $\hat{\theta} - \theta^*$ allows one to show that a two-step estimator has a rate of convergence lower than the classical one (*i.e.* $\sqrt{T}$) due to the presence of the (slow rate) evaluation functionals. Nevertheless, this (relative) weakness of two-step procedures can be corrected by modifying the definition of $\mathcal{R}_{2,T}$ with a suitable weight function [6]. In general, the (asymptotic) confidence interval constructed from the two-step estimator will be larger than the one constructed from the MLE. This indicates that the statistical optimality has been sacrificed for computational simplicity. This result concerning the possible slow rate of convergence indicates that some care must be taken when using nonparametric estimators in parametric procedures. This puts emphasis on the necessity to construct reliable and close estimators $\hat{X}$. Despite this potential limitation, empirical studies performed with cubic splines by [48, 31, 6] show satisfying results, even on relatively small samples. Moreover, nonparametric estimators are constructed in practice with adaptive procedures for selecting the basis $\psi_{k,T}, 1 \leq k \leq K_T$ (or the number K_T), or the bandwidth for Nadaraya-Watson. One can expect to construct "relatively" close functional estimates of X^*, so that the pointwise distance between $\hat{X}$ and X^* remains small even for finite sample size. An interesting family of adaptive nonparametric estimators can be derived in the framework of the so-called Support Vector Regression (SVR) [40, 42]. These estimators are functions belonging to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ associated to a kernel $k(\cdot,\cdot)$. The estimator (for each dimension) is then characterized as the solution of the following optimization problem

$$\hat{x}(t) = \arg\min_{x(t)\in\mathcal{H}} \|x(t)\|_{\mathcal{H}} + C\sum_{i=0}^{T} L_\epsilon(y_i - x(t_i))$$

where $L_\epsilon(\cdot) = \max(|\cdot| - \epsilon, 0)$ is the ϵ–loss function, $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm and C a regularization parameter. The solution exists and is unique, defined as

$$\hat{x}(t) = b + \sum_{i=0}^{T} c_i k(t_i, t)$$

and the coefficients b, c_i, $i = 0,\dots,T$ are computed by solving the constrained quadratic program

$$\{ \min_{c_i,b,\xi,\xi^*} \frac{1}{2}c^\top Kc + C\sum_{i=0}^{T}(\xi+\xi^*) s.t. \{ \quad ((Kc)_i + b - y_i \leq \epsilon + \xi_i y_i - ((Kc)_i + b) \leq \epsilon + \xi_i^* \xi, \xi^* \geq 0,\ i = 0,\dots,T$$

(14)

where $K = (k(t_i, t_j))_{i,j}$ is the kernel matrix. There is a great deal of choice for the functional specification of the kernel matrix, but the usual ones in univariate regression are the Gaussian or spline kernels [47]. We consider here only Gaussian kernels,

$$(t, t') \mapsto k(t, t') = \exp\left(-\gamma\frac{(t-t')^2}{2}\right)$$

with a fixed scale parameter γ, which possesses good approximation properties (it is a universal kernel [26], *i.e.* the RKHS is dense for the uniform norm in the set of continuous functions on $[0, 1]$). Among other adaptive methods, we can select the hyperparameters C and ϵ in (3.3) by minimizing the generalized cross validation criterion

$$GCV = \frac{\sum_{i=0}^{T}(y_i - \hat{x}(t_i))}{T - \hat{df}}$$

where $\hat{df}$ is the effective degrees-of-freedom. $\hat{df}$ is a generalization of the number of variables in linear regression, and it has been introduced in nonparametric regression for the estimation of the prediction error. In the framework of SVR, it can be approximated by counting the number of observations in the ϵ–tube, as proposed by [18]. Since only a subset of the coefficients of SVR are not null, and since the Gaussian kernel is quite close to splines for particular values of the scale parameter γ, the SVR estimator $\hat{x}(t)$ is quite close to a least-squares splines estimator with an adaptive selection of the knots. The estimated parameters do not seem to be sensitive to the family of the estimator, but rather on the smoothness in $x^*(t)$ and the underlying approximating power of the estimator. As a result, one can ask for more adapted nonparametric estimators, possibly using more information on x^*. We propose in the next section a slight modification of SVR to be able to use qualitative information for $\hat{x}(t)$.

3.4 Qualitative constraints and semiparametric estimator

We have seen in the linear expansion of $\hat{\theta}-\theta$ in equation (13) that the quality of estimation is directly related to the quality of approximation of $X^*(t)$ by $\hat{X}(t)$. We therefore propose to use prior knowledge from the solution of the ODE, and in particular qualitative knowledge, to have a better estimates $\hat{x}$ and $\hat{\theta}$. To do this, we suggest modifying the nonparametric estimation into a semiparametric estimation.

Our first approach is to introduce prior knowledge about the qualitative behavior of the solution. The idea is to decompose the solution in the following manner:

$$X_t^* = S_t + N_t$$

where $S_t = (s_1(t), \dots, s_p(t))$ represents the shape or the main pattern of the solution X_t^*, and $N_t = (n_1(t), \dots, n_p(t))$ is a noise or at least an unknown part. In that case, we identify the functions S_t with the qualitative behavior of the solution, and they serve to represent the information we have about the behavior of the system, such as the convergence to an equilibrium point, or to periodic solution. This latter situation particularly motivates this decomposition when the function X_t^* converge to a periodic solution $Z_t = (z_1, \dots, z_p)$ with (componentwise) Fourier decomposition

$$z_i(t) = \sum_{k=0}^{\infty} b_k \cos(2\pi k\omega t + \phi_k).$$

Hence, we can look for a decomposition

$$x_i^*(t) = \sum_{j=0}^{\ell} b_{j,i} \cos(2\pi j\omega_i t + \phi_{j,i}) + n_i(t)$$

with finite ℓ, and n might appear as the rest of the Fourier series plus a transient part. In all generality, we need to get a (precise) identifiability criterion for the couple (S_t, N_t). A possible one is that both parts belong to a RKHS $\mathcal{H}$ generated by a universal kernel $k(\cdot,\cdot)$ (and scalar product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$). We propose that s is in $\mathcal{S}$, the vector space spanned by $\psi_j, j = 1,\ldots,\ell$ where ψ_js are shape functions, *i.e.* $s = \sum_{j=1} \ell s_k \psi_k$. n belongs to the orthogonal of $\mathcal{S}$. This means that any continuous solution X_t^* and shape function S_t can be well approximated by a function in $\mathcal{H}$ (in the uniform norm). In that case, we search a semiparametric estimator by using the so-called semiparametric SVR. The representer theorem can be adapted in this setting ([40]), and the form of the semiparametric estimator is:

$$\hat{x}(t) = \sum_{l=1}^{\ell} b_l \psi_l(t) + \sum_{i=0}^{T} c_i k(t_i, t) = \hat{s}(t) + \hat{n}(t)$$

where $\hat{s}(t) = \sum_{l=0}^{\ell} b_l \psi_l(t)$ and $\hat{n}(t) = \sum_{i=1}^{T} c_i k(t_i, t)$ are the estimators of the shape and "noise" parts. The coefficients b_l, $l = 0,\ldots,\ell$ and c_i, $i = 1,\ldots,N$ are computed by solving the following optimization problem:

$$\hat{x}(t) = \arg\min_{b_k, c_i} C \sum_{i=1}^{T} L_\epsilon(y_i - \hat{s}(t_i) - \hat{n}(t_i)) + \|\hat{n}(t)\|_{\mathcal{H}}.$$

When we solve this problem, it shows up in the dual form that the parameters b_l, $l = 1,\ldots,\ell$ and c_i, $i = 1,\ldots,T$ are computed such that

$$\forall l = 0,\ldots,\ell, \langle \psi_l, \hat{n}\rangle_{\mathcal{H}} = 0.$$

This implies the decomposition of $\hat{x}(t) = \hat{s}(t) + \hat{n}(t)$ in two orthogonal parts and only the norm of $\hat{n}(t)$ is minimized while regularizing. It is possible to use (nonlinear) parameters ϖ in the parametric part by putting $\psi_l(\cdot) = \psi(\cdot, \varpi_l)$, $l = 0,\ldots,\ell$. For instance, in the case of sustained oscillations, the parameters (representing a prior information) are the frequency ω and the phases ϕ_k. Note that the use of a semiparametric estimation for imposing shape constraints has been used by [3] (by using regularization with linear operators), or [31] in the context of the estimation of differential equations.

There is room for important improvements of two-step estimators by using a well-adapted parametric part, which then highly depends on the context. Nevertheless we can propose another general way of computing sensible parametric priors. Indeed, in all the cases, we can try to describe X_t^* with the use of several others solutions of the differential equation $X_t(\eta_0),\ldots,X_t(\eta_\ell)$. This means that we consider that the parameters $\eta_0,\ldots,\eta_\ell$ are reasonably close to η^* such that $X_t(\eta_i)$ should behave approximately as X_t^*, hence instead of using only the parameter values η_i, $i = 0\ldots\ell$ as initial values for the optimization of the criterion $\mathcal{R}_{2,N}(\eta)$ in the second step, we use them also for a better approximation during the first step. This approach can be partly justified by the generalization of well-known properties of linear ODEs. Indeed, the set of solutions of a linear ODE (with fixed and known parameter θ^*) is a vector space generated by the (linearly independent) solutions $X_t([\theta^*, X_0^1]),\ldots,X_t([\theta^*, X_0^d])$ having p different initial conditions x_0^j, $j = 1, \ldots, p$. In this very simple situation, the estimation of θ boils down to the estimation of the initial parameters X_0, *i.e.* of the linear combination $\sum_{k=1}^{d} b_k X\left(t, (\theta^*, X_0^k)\right)$, and the final estimate of initial condition is simply $\hat{X}_0 = \sum_{k=1}^{d} b_k X_0^k$. The nonparametric part s is here to account for the fact that the differential equation is nonlinear and that we do not know θ^*. In practice, there is no particular constraint on the number ℓ of solutions that are used in the parametric part.

Another piece of prior information that can be easily incorporated in the first step is the knowledge of some values taken by the true solution X_t^*. Typically, it is possible to know the initial value of the system X_0^* and one would like to have a proxy $\hat{X}_t$ such that $\hat{X}_0 = X_0^*$. This can be done straightforwardly with SVR, because it suffices to add this constraint in the optimization program (3.3), which can still be solved (it remains quadratic convex with linear constraints); SVR enables a mix of approximation and interpolation. This can be generalized to a series of known values $\hat{X}_{t_k} = d_k$, because this is equivalent to have $\left\langle \hat{X}, k(t_k, \cdot)\right\rangle_{\mathcal{H}} = d_k$. A particularly interesting situation is the case where we know the initial and final conditions X_0^* and X_1^*. According to equation (13), the contribution of the evaluation functionals $\hat{X}_1 - X_1^*)$ and $\hat{X}_0 - X_0^*)$ vanishes. Hence, there are two practical consequences to this slight modification of $\hat{X}_t$: on one hand, $\hat{X}_t$ is closer to the observations since we have the exact values at the boundary; on the other hand, the slow rate part in the asymptotic expansion has vanished, so the rate of $\hat{\theta} - \theta^*$ is governed only by the smooth functional $\Gamma(\hat{X} - X^*)$, so it has the parametric rate of convergence $\sqrt{T}$.

3.5 Experiments

We present experiments on data simulated from the repressilator model. The situation that we consider here is more favorable than the one described in Section 2.1 because we assume that all the states are observable, which means that we observe simultaneously the mRNA and protein concentrations. It is then possible to compute the 6 functions $\{\hat{x}_i\}_{i=1}^{6}$ that correspond to the estimation of the 3 true concentrations of mRNAs m_1, m_2, m_3 and 3 true concentration of proteins p_1, p_2, p_3 as well as the criterion $\mathcal{R}_{2,T}(\theta)$. The two-step procedure is illustrated in Figure 3.5, which shows how well the concentration of protein p_1 and its derivative have been approximated.

We consider observation noise $\epsilon \sim (0, \sigma_p^2)$ that is spherical with $\sigma^2 = 4$ and an ODE with initial conditions $m_1(0) = 100, m_2(0) = 100, m_3(0) = 150$ and $p_1(0) = 1, p_2(0) = 2, p_3(0) = 3$. The true parameter vector θ^* is given in Tables 1 and 2. These parameter values give sustained oscillations for the concentrations. Below we describe how different kinds of prior knowledge can be used for the construction of $\hat{X}_t$:

1. a completely nonparametric estimator $\hat{X}_t$ (no prior knowledge),

2. a semiparametric estimator $\hat{X}_t^{\text{period}}$ with a periodic function

$$s_i(t) = b_0 + b_1 \cos(2\pi\omega_i t + \phi_{i,1}) + b_2 \cos(2\pi \times 2\omega_i t + \phi_{i,2}),$$

3. a semiparametric estimator $\hat{X}_t^{\mathrm{ode}}$ with $S_t = b_1 X_t(\{x_0, \theta_1\}) + b_1 X_t(\{x_0^*, \theta_2\})$),
4. a nonparametric estimator $\hat{X}_t^{\mathrm{cons}}$ constrained to be such that $\hat{X}^{\mathrm{cons}}(0) = X^*(0)$ and $\hat{X}^{\mathrm{cons}}(1) = X^*(1)$.

We compare the mean performances of the two-step estimators $\hat{\theta}$, but we insist on the fact that we essentially compare then our ability to estimate the solution with an adaptive procedure. Indeed, the first step should be done by an automatic and general method that adaptively selects the parameters C, ϵ and γ from the data, hence a comparison of the two-step estimator will essentially compare the adaptive procedure for computing the first step. When T is large, these different estimators converge to give equivalent estimates. Hence, we are interested in the size of the finite sample, and in particular in the case of small T because one can expect to have a gain in using qualitative/prior knowledge when observations are limited. As previously described, it is possible to adapt $\hat{X}$ to the use of different prior knowledge in the system. In the case of a periodic shape (situation 2), we take advantage of the convergence to a periodic solution: we suppose that the first 3 components of the Fourier series are known (*i.e.* ω, ϕ_1, ϕ_2). In the case of the repressilator, the frequency (common to all the dimensions) is approximately equal to $\omega = 0.55$, but the phases are different along the dimension and they are roughly estimated from the data. The semiparametric estimator is computed by estimating first the parametric part (by nonlinear regression), and then the semiparametric SVR with the estimated (fix) shape parameter. For $\hat{X}^{\mathrm{ode}}$, we use two parameters which are close to the true value θ^* and having initial conditions equals to $x^*(0)$. The parameters we used are $\theta_1 = \theta^* + 0.2$ and $\theta_2 = \theta^* - 0.1$. They give sustained oscillations. One can see that $\hat{X}_t, \hat{X}_t^{\mathrm{ode}}, \hat{X}^{\mathrm{cons}}(t)$ gives roughly the same estimators from Tables 1 and 2 but the behavior of these estimators are slightly different, as it is shown in Table 3. From the comparison of the estimated mean square error (MSE) and variance of the three estimators, it is clear that the $\hat{X}^{\mathrm{cons}}(t)$ gives the best estimator with a smaller bias and a smaller variance. The second semiparametric estimator $\hat{X}_t^{\mathrm{ode}}$, gives intermediate results between $\hat{X}_t$ and $\hat{X}_t^{\mathrm{cons}}$ (except for $k_{3,1}$) (the same remarks can be done for parameters $k_{1m}, k_{2m}, k_{3m}, k_{1p}, k_{2p}, k_{3p}, h$). This shows that the use of additional information during the first step can ameliorate significantly the statistical performance of the two-step estimators. Nevertheless, the results of $\hat{X}_t^{\mathrm{period}}$ shows that the use of prior knowledge must be done with care. Despite a reasonable fit in the first step, the underlying parametric structure is too strong and causes then an important bias in the estimation. In our experiment, it comes from the fact the solution is transient in $[0, 10]$ which induces an artefact in the estimation of the parametric part, but we remark that better adaptive semiparametric methods can be used, see for instance [37]).

[Estimation of the true solution p_1^* (hard line) by a Support Vector Regression $\hat{p_1}$ (dashed-line), computed from 50 noisy observations (stars).] [width=.9]chapters/BrunelDalcheBuc/2step/Estimation$_fonction_spline.pdf[Estimation of the derivative \dot{p}_1$ by $\dot{\hat{p_1}}$ (dashed-line) and by $f(X_t, t, \theta)$ (dash-dot line) (true derivative is hard-line). The two-step estimator $\hat{\theta}$ is the parameter that minimizes (locally) the L_2 distance between the dashed line and the dash-dot line.] [width=.9]chapters/BrunelDalcheBuc/2step/Estimation$_derivee_spline2.pdf$

Figure 2: Estimation of the protein p_1 concentration and its derivative for the Repressilator model

Table 1: Mean of the two-step estimator computed with different estimators of the true solution of the data when $T = 40$ observations, computed with 100 Monte Carlo runs.

	True Parameter	$\hat{X}$	$\hat{X}^{period}$	$\hat{X}^{ode}$	$\hat{X}^{cons}$
v_1	150	150.0	113.2	149.1	149.17
v_2	80	79.99	87.6	78.2	78.4
v_3	100	101.98	81.2	101.8	100.6
$k_{1,2}$	50	50.5	66.6	50.4	50.5
$k_{2,3}$	40	40.4	53.2	40.1	40.5
$k_{3,1}$	50	49.65	39.0	48.9	50.2
k_1	1	0.98	1.24	1.23	0.99
k_2	2	1.96	1.9	1.91	1.95
k_3	3	2.85	3.21	3.18	2.8

Table 2: Comparison for different prior knowledge.

	True Parameter	$\hat{X}$	$\hat{X}^{period}$	$\hat{X}^{ode}$	$\hat{X}^{cons}$
k_{1m}	1	0.98	0.34	0.99	0.99
k_{2m}	1	0.97	0.84	0.96	0.96
k_{3m}	1	0.99	0.85	0.98	0.99
k_{1p}	1	0.98	1.31	1	0.99
k_{2p}	1	0.98	1.11	0.98	0.98
k_{3p}	1	0.99	1.0	0.98	0.93
h	3	2.89	2.81	2.88	2.92

4 Learning state space models defined from ODEs

Another point of view for reverse-modeling of biological networks consists of choosing the probabilistic framework of graphical models to represent the dynamical processes at work in the cell. Graphical models (GM) allows the representation and factorization of a joint distribution of variables of interest, taking into account conditional independences: they also benefit from the statistical estimation linked to generative models, regardless of the framework being frequentist or Bayesian. Dynamical Bayesian networks, and more specifically state-space models, are good candidates to represent interactions between components that occur through time as emphasized in Chapter **??**. This choice gives rise to several advantages: first, variables of interest, for instance the mRNA concentrations or the protein concentrations in the case of regulatory interactions, are seen as random variables, allowing the representation of some stochasticity, which could arise from either the measurement process or to the nature of the biological process. Second, in this framework, it is possible to treat some variables as "hidden" and estimate them as parameters. Note that the incompletness of the observations is quite realistic. For instance, when studying transcriptional regulations, we usually do not observe proteins concentrations together with mRNA concentrations because of the technical difficulty in performing such joint measurements. So the available data often reduce to transcriptome data measured through DNA chips or qPCR. In this case, we can consider that the observations are noisy measurements of mRNA concentrations, whose dynamics can be described by some hidden processes which involve protein concentrations and mRNA concentrations.

4.1 Definition of state-space models based on ODEs

Motivated by these remarks, we focus on the rich framework of state-space models [36, 7] that will also be considered in Chapter **??** in the form of linear Gaussian models. Since 2003, several authors have proposed linear state-space models mainly to represent gene regulatory networks [12, 30, 35]. One way to define a model [12, 30] is to start from the definition of a linear ODE system, discretize time and add noise to get a probabilistic model of dynamics that takes into account intrinsic and extrinsic noises. All the true state-variables are represented as hidden processes while the observations are assumed to be produced by these hidden processes with the addition of noise. In this case, biological relevance and limits of technical measurement dictate the choice of hidden variables and observed variables. First results with such linear models are encouraging but regulations and interactions between macromolecules exhibit saturated behavior, so it is quite natural to turn to nonlinear models. Meanwhile, one can argue that discretizing a continuous time-model requires a too big an assumption about the time intervals between which measure. In this chapter [32, see also], we propose a state-space model whose hidden process is defined through an integration of the ODES, classically used to represent biological dynamical systems, avoiding a too big dependency on the choice of the time interval used to acquire data. The idea consists of building a nonlinear state-space model that benefits from the two frameworks: that of ODEs which allows us to describe, in a time-continuous setting, the dynamical behavior of the network [11, 13], and the one of state-space models [7, 21, 41] that allows us to deal with hidden variables and is also associated with a large family of estimation methods. We define a state-space model whose transition function for the hidden states is based on the integration of the function f defining the ODEs as introduced in Section 2. Moreover, the framework of Bayesian estimation allows for the possibility of adding constraints by the use of prior distributions on parameters.

Our model is defined from the following assumptions: First, the state of the network satisfies the following ODE:

$$\dot{X}_t = f(X_t, t, \theta) \tag{15}$$

Second, the vector X_t is not observed and we will now refer to it as the *hidden state* of the network at time t, with t varying from t_0 to t_T. We assume that the variables $\mathbf{Y} = (Y_{t_1}, \ldots, Y_{t_{p_y}})$ can only be observed through the observation function H. Then, a state-space model can be defined with functions F_t and functions H and using the notation $X_{t_i} = X^i$ for $i \geq 0$ (and similar for Y):

$$X^{i+1} = F_i(X^i; \theta). \tag{16}$$

$$Y^i = H(X^i; \theta) + \epsilon_i. \tag{17}$$

with ϵ_i being a measurement noise chosen as a centered Gaussian noise and the following definition of $F_i(\cdot)$ based on ODEs [41, 32]:

$$F_i(X^i; \theta) = X^i + \int_{t_i}^{t_{i+1}} f(X_u, u, \theta)\mathrm{d}u. \tag{18}$$

The state-space model defined by equations (16) and (17) is frequently encountered in engineering but had not yet been exploited in reverse-modeling of biological networks . Let us discuss the properties of the model.

The variables $\mathbf{X}$ can have a stochastic evolution, so equation (16) may be replaced by the more general one $X^{i+1} = F_i(X^i; \theta) +_i$, with $(_i)^{i \geq 0}$ being a white noise. This assumption also has a biological motivation; for instance, [25] have shown the intrinsic stochasticity of gene regulatory networks, where X_t represents gene expression levels and concentrations of transcription factors in the cell. However for sake of simplicity we will keep here a deterministic hidden process. One of the most interesting feature of this model is its ability to deal with irregular measurement time intervals, because the transition function in the hidden process is based on an integration. Assuming the integrability of function f is checked, a numerical integration is feasible (such as Runge-Kutta integration **??**) leading to a computationally efficient transition step. Let us notice that we make a rather parsimonious use of ODE integration in our model since the integration is processed between a restricted interval of time. During the inference, we use the Markov property for the decomposition of the global least-squares criterion. Indeed, we can propose recursive estimators where the quality of a given parameter is re-computed for each new observation and can be abandoned then, whereas a classical least-squares integrates the ODE for the interval $[0, 1]$ before evaluating the quality of the solution (for all the observations).

The transition from X^i to X^{i+1} is time-dependent: first, because of uneven sampling times (even if the ODE (15) is autonomous) and second, because of the presence of a time-dependent input variable. In the following, we show that this framework encompasses both models of transcriptional regulatory networks and models of signaling pathways, including the presence of an input signal.

4.2 Learning states and parameters with a recursive Bayesian inference algorithm

Learning the parameters (and the initial state) of a state-space model can be tackled using different points of view. As a probabilistic model, the parameters can be estimated through a frequentist approach such as likelihood maximization as well as Bayesian approaches such as Maximum a posteriori (that can be seen as penalized likelihood maximization) or full Bayesian approaches. In the case of Bayesian approaches, parameters are assumed to be random variables and the estimation process aims at learning the posterior distribution of parameters and initial state, $p(\theta, X^0|Y^{0:T})$, given a prior distribution $\pi(\theta, X^0)$. The major difficulties here come from the fact that the true state of the system is hidden and moreover, that in our case, the model is assumed to be nonlinear. As the true state of the system is hidden, it is not possible to estimate posterior probability of parameters without estimating the conditional probability distribution of states given the observations, also referred as posterior probability distribution of states. The probability distribution of states X_{t_i} can take two forms: the filtering probability $p(X^i|Y^{0:i})$ if we consider only observations until time t_i and the smoothing probability $p(X^i|Y^{0:T}$ if we take into account the full observed process $Y_{t_0:t_T}$. The theory of optimal filtering and smoothing [see for instance [7]] defines recursive convergent algorithms to proceed to this estimation. In each case, the method provides an approximation of the posterior probability that also gives an approximation of the minimum mean squared error estimator (MMSE).

In this framework, if one uses the so-called augmented state vector approach that consists of assimilating parameters to an additional hidden variable, the state and parameters estimation issue can be addressed through the same recursive Bayesian inference algorithm applied to a new joint state variable. The dynamics evolution of the new system can be described as follows:

$$\theta_{i+1} = \theta_i \tag{19}$$

$$X^{i+1} = F_i(X^i; \theta_i) \tag{20}$$

$$Y^i = H(X^i; \theta_i) + \epsilon_i, \tag{21}$$

where the parameter is considered as a hidden state without any temporal evolution. We shall notice that the fact that the parameters are constant in the dynamical system does not mean that during the recursive algorithm they will not be affected by the corrections. This point will be highlighted when we will recall the equations of filtering and smoothing.

Finally, the last difficulty is encountered when the function F_i is nonlinear which is exactly the case for most of the biological networks equations we described in Section 2. In this case, the computations involved in filtering or smoothing become intractable and adaptation of classical filtering or smoothing well adpated to linear systems have to be derived. In this work, we have chosen to focus on the use of unscented transform introduced in [20] to overcome these difficulties.

Let us now recall the general principle of optimal filtering and smoothing, then we describe the extension of the Kalman filter and smoother to the case of nonlinear evolution equations, using the unscented transformation (UT). We have chosen to use a Rauch-Tung-Striebel smoother that presents the advantage of starting with a forward filtering pass and then using a separate backward smoothing pass, as described by [7, 38, 5].

4.2.1 Filtering

Using the augmented state vector approach, we can now remove parameter θ in equations (16) and (17) and only describe the estimation of hidden (augmented) states that include parameters θ and noise parameters. Filtering is the sequential computation of the *posterior* (or filtering) probability $\alpha_i(X) = p(X^i|Y^{0:i})$ for $i = 0, \dots, T$ [7]. Without loss of generality, the complete process $\mathbf{X} = \left\{X^i\right\}_{i=0}^{T}$ may be a Markov (nondeterministic) chain, with values in $\mathcal{X}$ (here $\mathcal{X} \subset^p$). The computation of the filtering probability consists of the alternate and sequential computation of the prediction probability $p(X^i|Y^{0:i-1}), i \geq 0$, in the so-called *prediction step*:

$$p(X^i|Y^{0:i-1}) \quad = \quad \int_{\mathcal{X}} p(X^i|X^{i-1})\alpha_{i-1}(X^{i-1})\mathrm{d}X^{i-1} \tag{22}$$

and its “correction” into $\alpha_i(\cdot)$ (the so-called *correction step*) by:

$$\alpha_i(X^i) \quad = \quad \frac{p(Y^i|X^i)p(X^i|Y^{0:i-1})}{\int_{\mathcal{X}} p(Y^i|X^i)p(X^i|Y^{0:i-1})\mathrm{d}X^i}. \tag{23}$$

We can then derive the sequence of most likely current states characterized by $\hat{X}^i = \arg\max_{X\in\mathcal{X}} \alpha_i(X^i), i = 0, \dots, T$. Note that at $i = 0$, the prediction step is replaced by setting $p(X^0|Y^{0:-1})\pi(X^0)$, where $\pi(X^0)$ is our prior distribution on the initial state.

4.2.2 Smoothing

In smoothing, one wishes to benefit from the whole observed sequence $Y^{0:T}$ and thus the distribution of interest is $p(X_i|Y^{0:T})$, called the *s*moothing distribution.

Using the following decomposition of this smoothing distribution,

$$p(X^i|Y^{0:T}) = p(X^i|Y^{0:i}) \int \frac{p(X^{i+1}|X^i)p(X^{i+1}|Y^{0:T})}{p(X^{i+1}|Y^{0:i})} \mathrm{d}X^{i+1} \tag{24}$$

where $p(X^i|Y^{0:i})$ is the *filtering* distribution of the time step i and $p(X^{i+1}|Y^{1:i})$ is the predicted distribution at time step $i+1$ which is estimated by optimal filtering described in previous section, the recursive smoother proceeds backward in time from last time step $i = T$.

There exist several ways to implement such a computation. We use here the Rauch-Tung-Striebel smoother type, and a recent unscented version described by [38], that can be decomposed into three steps:

- Using $p(X^i|Y^{0:i})$, the filtering distribution of the current time step i, compute the joint distribution of the hidden states X^i and X^{i+1} given the sequence of observations $Y^{1:i}$:

$$p(X^i, X^{i+1}|Y^{0:i}) = p(X^{i+1}|X^i)p(X^i|Y^{1:T}). \tag{25}$$

- Then, condition the joint distribution of current hidden state X^i and X^{i+1} to $Y^{0:i}$ in order to compute the conditional distribution of the state at current time step i given the next state X^{i+1} and the sequence $Y^{0:i}$

$$p(X^i|X^{i+1}, Y^{1:i}) = \frac{p(X^i, X^{i+1}|Y^{0:i})}{p(X^{i+1}|Y^{0:i})},$$

  where the denominator can be expressed as:

$$p(X^{i+1}|Y^{0:i}) = \int p(X^{i+1}|X^i)p(X^i|Y^{0:i})\mathrm{d}X^i.$$

- Using the Markov properties, $p(X^i|X^{i+1}, Y^{0:T})$ reduces to $p(X^i|X^{i+1}, Y^{0:i})$ and thus the joint distribution of X^i and X^{i+1} given the sequence $Y^{0:T}$ can finally be written as

$$p(X^i|X^{i+1}, Y^{0:T}) = p(X^i|X^{i+1}, Y^{0:T})$$

4.3 Approximate Filtering and Smoothing

When $\mathbf{X}$ is a Gaussian and linear Markov process ($F(\cdot)$ and $H(\cdot)$ are linear), the prediction-correction algorithm is the well-known Kalman filter which consists of a recursive computation of the mean and covariance of the (Gaussian) distribution $\alpha_i(\cdot)$ while the forward-backward smoothing corresponds to the Kalman smoother. However, when these equations involve nonlinearities or when noise is not Gaussian, the computations become intractable. Among several extensions that have been proposed to tackle nonlinearities or non-Gaussianity, we have here focused on the unscented transform that allows the approximation of the filtering distribution and the smoothing distribution by Gaussian densities with the help of deterministic sampling. We denote μ_i, the mean of the Gaussian approximation of $\alpha_i(\cdot)$ and Σ_i, its covariance. Unscented Kalman filtering (UKF) relies on small deterministic sets of appropriately chosen points used to mimic the nonlinear evolution of the state variable: the so-called sigma points $\xi_{0,i}, \ldots, \xi_{2p,i}$. The key idea in UKF lies in the *prediction step*, where the "unscented transformation" allows one to compute an approximation of the mean $\mu_{i+1|i}$ and covariance $\Sigma_{i+1|i}$ of the prediction probability. The mean and covariance of the transformed variable $F(X_i)$ (when X_i has the posterior distribution $\alpha_i(\cdot)$) can indeed be approximated simply by using the first empirical moments of transformed sigma points chosen as

$$\xi_{0,i} = \mu_i, \xi_{j,i} = \mu_i + Q_{j,i}, \xi_{i+p,i} = \mu_i(\mathbf{x}) - Q_{j,i},$$

where Q_i is a square root matrix of $(2p + 1/2)\Sigma_i$. Other interesting choices of sigma points are given in [21]. Then, the *correction step* is carried out in a way similar to Kalman filtering using the classical filtering equations and the (approximate) covariance of the p.d.f $p(Y^{i+1}|Y^{0:i})$. The sequence of estimates (*filtered process*) $\hat{X}^i$ of the hidden variables is the sequence of means μ_i.

For the case of unscented RTS Smoothing (RTS-UKS), the same idea is used to build an approximation to the optimal smoothing by estimating the mean $\mathbf{m}^s_{i+1}$ and the covariance Σ^s_{i+1} of the smoothing distribution, supposed here to be Gaussian. Details of this approach can be found in [38].

In practice, we can use UKF (resp. RTS-UKS) in order to compute the approximation of $p(X^i|Y^{0:i})$ (resp. $p(X^i|Y^{0:T})$) and hence derive a sequence of estimates $(\hat{X}^i, \hat{\theta}_i)$.

The minimizer of the squared error is approximated by μ_i. Nevertheless, in both cases these approximations can be spurious minimizers, and it is recommended to perform several sweeps of the algorithm on the data.

For the filtering case, as described for hidden states, at $i = 0$, the prediction step is replaced by setting

$$p(X_0, \theta_0|Y^{0:-1}) = \pi(X_0, \theta, X).$$

We propose using the rather noninformative hierarchical prior $\pi(X^0, \theta) = \pi(\theta)\pi(X^0)$, where $\pi((X^0)_i) = \prod_i \mathcal{N}(\mu_{x_i}, \sigma^2_{x_i})$, with $\mu_{x_i} \sim U([0, \lambda_i])$, *i.e.* all of the components of the vector are independent and Gaussian, with a mean drawn according to a uniform distribution whose support is determined by an hyperparameter λ_i computed from the data, and the variance $\sigma^2_{x_i}$ is a fixed value (depending on the data). However, if a certain constraint concerning the initial value is made available, more informative prior could be used.

4.4 Results

We first illustrate our approach on artificial data generated from the repressilator model and second, on experimental data of the JAK-STAT pathway. Other simulation studies that provide useful insights into the strengths and weaknesses of learning algorithms, such as robustness against numerous choices of settings,

Table 3: Mean Square Error and Variance of two-step estimators with different prior knowledge, computed with 100 Monte Carlo runs for $T = 100$.

	$\hat{X}$		$\hat{X}^{period}$		$\hat{X}^{ode}$		$\hat{X}^{cons}$	
	MSE	Var	MSE	Var	MSE	Var	MSE	Var
v_1	10.62	10.45	$2.1\ 10^{+3}$	753.4	15.1	14.7	10.5	10.5
v_2	12.84	12.84	187	128.7	11.2	10.4	12.4	9.4
v_3	19.83	17.68	726.5	374	15.8	11.8	14.9	14.8
$k_{1,2}$	0.44	0.23	320	44.9	0.36	0.28	0.45	0.24
$k_{2,3}$	1.72	1.56	344	168.5	1.52	1.52	1.76	1.41
$k_{3,1}$	1.8	1.71	219	99.3	2.6	1.41	1.48	1.44
k_1	1.10^{-3}	1.10^{-3}	0.1	0.04	5.10^{-4}	5.10^{-4}	4.10^{-4}	5.10^{-4}
k_2	3.10^{-3}	2.10^{-3}	0.1	0.08	3.10^{-3}	3.10^{-3}	4.10^{-3}	3.10^{-3}
k_3	3.10^{-2}	1.10^{-2}	0.2	0.18	5.10^{-3}	4.10^{-3}	7.10^{-2}	3.10^{-3}

including the quantity of observed data, the sampling interval for observing the data, the number of time points in the observed time series, can be found in [32] and its supplementary material.

For the simulation presented in this chapter, we made use of the code of Sarkka08[1] [38]. The size of the systems used in the experiments are quite representative of the size of the nonlinear models that the proposed method can efficiently handle, without identifiability problems, *i.e.* around ten variables and parameters. In higher dimensions, the recursive optimization using the unscented approximation can lead to spurious minimizers, but simulating more initial values from the prior distribution could help to improve the algorithm. One of the main limitations of the approach is closely related to the respective sizes of the observed and hidden parts of the system.

4.4.1 Parameter and hidden state estimation of the Repressilator

We start from the equations given in (2.1) and fix the following values of the parameters according to the stability study presented in [16]: $k_1^p = 1$, $k_2^p = 2$, $k_3^p = 3$, $k_1^g = k_2^g = k_3^g = 1, v_1 = 50, v_2 = 80, v_3 = 100, k_{1,2} = 50, k_{2,3} = 30, k_{3,1} = 40$ and $h = 3$. The components of the initial state are drawn independently from a uniform distribution on $[0, 100]$ (arbitrary units). Simulations are performed using the MATLAB numerical integrator *ode45* over the time interval $[0, T]$, with $T = 20$. The observation noises ϵ_i are added to three observed variables to mimic gene expression data and the standard deviation of ϵ_i shown in the experiments is chosen to be equal to 20% of the standard deviation of the states. The robustness of the method has been tested with respect to a higher noise level (30%, 40%), and similar results for the estimation for the states and parameters have been obtained. The estimated predicted variance and the variance of the estimators increase, although no systematic divergence of the method has been detected.

[1]EKF/UK toolbox, V1.2, http://www.lce.hut.fi/research/mm/ekfukf/

During the simulation, measurements are sampled at a fixed interval Δ_t, so that for each experiment a time series containing T/Δ_t time points is collected. We assume that the learning problem consists of identifying the following 6 parameters: $v_1, v_2, v_3, k_{1,2}, k_{2,3}, k_{3,1}$ while the degradation rates for proteins and mRNAs are known. In order to learn the true parameters, we use a multi-start approach by sampling $I = 50$ different initial states and parameters from our prior $\pi(x_0, \theta)$, so that we compute 50 filters or smoothers in parallel. Our final state and parameter estimates are simply the mean of the prediction of the 50 different filters or smoothers (an alternative way to combine the different filters would be to select the filter with the lowest prediction error). The Gaussian priors for the parameter are such that $\lambda_i = 2 \times \theta_i^*$ and $\sigma_{\theta_i} = 0.2 \times \theta_i^*$, and for the unobserved variables $\lambda_i = 2 \times X0^i$ and $\sigma_{\underset{0}{i}}$ is set to 20% of the standard deviation of the state $_i$. For the observed variables, the prior is also Gaussian with mean $\mu_{\underset{0}{i}} = y_0^i$ and the same formula as for the unobserved variables is used for the standard deviation.

Evaluation of estimation The filtered protein concentrations as well as the smoothed protein concentrations using the unscented transform are shown in Figure **??** for $\Delta_t = 0.2$. Among 50 runs started with different random initialisations, the sequence of hidden states that best fits the observations has been chosen. It is quite difficult to distinguish the sequences of estimated state-variables using the two approaches while in each case, the estimation is successful. To get a more precise idea about the contribution of the smoothing compared to filtering, the reader may have a look at Table **??**. For 100 different samples, UKF and UKS have been run from 50 random initialisations of the parameters. Each result of the multi-start approach is an average of the final estimation of each of the 50 runs. The obtained empirical mean and standard deviation computed from this scheme is figured in the table. Smoothing performs generally slightly better than filtering.

Additional simulations about the influence of the number of different experiments (*i.e.* time series corresponding to the observation of the same system but with different initial conditions) can be found in [32]: we showed that estimated parameters tends to their true values with smaller standard deviations when the number of observations increases.

4.4.2 Parameter estimation for the JAK-STAT pathway model using experimental data

Experimental data of JAK-STAT pathways from [45] was used. Time series of two observed variables $y_1(t)$ (the total concentration of phosphorylated STAT5) and $y_2(t)$ (the total concentration of STAT5 in the cytoplasm) are measurable. Each time series contains 16 time points sampled at $t = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 40, 50, 60]$ minutes. Data for the input EPoR phosphorylation is also available. Here we use a linear interpolation in order to obtain a continuous time input. We initialize the parameters described in Section 2 a_1, a_3, a_4 and the initial condition X_0 independently with a uniform distribution on $[0, 5]$. The reader can refer to

[32] to see the curve corresponding to normalized MSE between the predicted time series and the data in Figure 4 of the *Supplementary Material* of [32]. The convergence of this curve shows the stability of the learning algorithm, and ensures that a local minima has been reached. Eventually, the parameter estimates (with standard deviations) are $\hat{a}_1 = 0.0515 \pm 0.0055$, $\hat{a}_3 = 3.39 \pm 0.45$ and $\hat{a}_4 = 0.35 \pm 0.047$, and the prediction for the observed variables $y_1(t)$ and $y_2(t)$ are shown in Figure **??**, which shows a good fit of the learned model. We also check the coherence of the estimation by simulating the JAK-STAT pathway with these estimates. A new time series is simulated from (8) with initial conditions $x_1 = 0.2$, $x_2 = x_3 = x_4 = 0$ and the estimated parameters. The result in Figure **??** showed that the learned model is able to predict well the four unobserved components of $X^\star$, so we may have a higher confidence in the prediction of the unobserved variables.

5 Conclusion and perspectives

We have presented two approaches in order to learn parameters of nonlinear ODEs devoted to biological networks modeling. These two approaches appear as alternatives to classical least-squares method: a two-step estimator based on functional estimation and a recursive Bayesian method applied on a state-space model based on the integration of a system of ODEs. As we have seen, the two-step estimator consists of a new kind of parametric estimator, which constructs a functional proxy from the data. This particular feature limits the method to completely observed systems, which might restrict its applicability in practical situations with true data (even if some extensions can be considered [6]), but this approach is highly related to the biological interpretation of the models because it allows the visualization of the curves and the possibility of imposing some constraints. On the other hand, the state-space formulation of ODE models is well-adapted to partially observed systems or to repeated measurements thanks to the versatile probabilistic interpretation of the state-space framework. Moreover, some prior knowledge can be used by sampling parameters from appropriate prior distributions, but the shape of the solution here is not directly controllable. Both algorithms can be seen as a particular formulation of the estimation problem for deriving fast algorithms, and they rely fundamentally on the ODE model. Indeed, the two methods emphasize the duality of the definition of a dynamical system or process which can be described either as a function of time (solution of the differential equation) or as a transition system. In other words, the two methods reflect respectively a bottom-up and a local description of a system. One simulates the observed system with the state equation and tries to fit a model with respect to the data; whereas the two-step estimator is top-down approach: it starts from a reconstructed global behavior and then identifies the corresponding transition rules between the states from the smooth reconstructed function.

Several extensions can be drawn from this work. The idea of constraining the shape of the ODE solution can also be implemented in the framework of state-space model by imposing that the estimated hidden states be decomposed into a spline basis.

Another extension of this work could be the modification of the deterministic ODE into a stochastic differential equation (by adding a noise in the ODE) that would lead to the definition of a stochastic state process as it is usually the case in state-space models. The previous notion of duality is well-known in a probabilistic context, where one can describe a Markov continuous-time stochastic process either with a transition kernel, or as a random variable in function space (with some pathwise properties). This last interpretation is of course related to Gaussian processes and their use in machine learning. This can be related to the work described in Chapter **??**.

Another important direction is now to scale these approaches to larger networks. Simpler models such as generalized linear models can surely help here. Decomposition into modules either by the way of a mixture (see for instance Chapter **??**) or by clustering and dimension reduction may also provide the key to addressing this issue.

Finally, a complete and generic approach for reverse-engineering of biological networks would consist in combining parameters estimation algorithms with structure learning procedures even in the case of nonlinear models.

Funding

This research has been mainly funded by Genopole^{TM}through an ATIGE and a postdoc grant and partially by an ANR (National Agency for Research) grant (project GD2GS, ARA Call 2005).

References

[1] D. K. Andrews. Asymptotic normality of series estimators for nonparametric and semiparametric regression models. *Econometrica*, 59(2):307–345, 1991.

[2] Milena Anguelova. *Observability and identifiability of nonlinear systems with applications in biology.* PhD thesis, Chalmers university, 2007.

[3] C. F. Ansley, R. Kohn, and C-M Wong. Nonparametric spline regression with prior information. *Biometrika*, 80(1):75–88, 1993.

[4] C. Auliac, V. Frouin, X. Gidrol, and F. d'Alch Buc. Evolutionary approaches for the reverse-engineering of gene regulatory networks: a study on a biologically realistic dataset. *BMC Bioinformatics*, 9(91), February 2008.

[5] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. *Trans. on Signal Processing*, submitted.

[6] N. Brunel. Parameter estimation of ode's via nonparametric estimators. *Electronic Journal of Statistics*, 2008.

[7] O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer-Verlag, 2005.

[8] G. Casella and R. Berger. *Statistical Inference*. Duxbury Resource Center, 2003.

[9] T. Chen, H.L. He, and G.M. Church. Modeling gene expression with differential equations. In *Pacific Symposium of Biocomputing*, 1999.

[10] I.-C. Chou, H. Martens, and Eberhard O. Voit. Parameter estimation in biochemical systems models with alternating regression. *Theor Biol Med Model.*, 3(4):1–15, 2006.

[11] A. Csikasz-Nagy, D. Battogtokh, K.C. Chen, B. Novk, and J.J Tyson. Analysis of a generic model of eukaryotic cell cycle regulation. *Biophysical Journal*, (90):4361–4379, 2006.

[12] Michael J. L. de Hoon, Seiya Imoto, Kazuo Kobayashi, Naotake Ogasawara, and Satoru Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. In *Proceedings of the Pacific Symposium on Biocomputing*, 2003.

[13] Hidde de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[14] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mrna expression levels during cns development and injury. *Pacific Symposium of Biocomputing*, 4:41–52, 1999.

[15] S. Donnet and A. Samson. Estimation of parameters in incomplete data models defined by dynamical systems. *Journal of Statistical Planning and Inference*, 137(9):2815–2831, 2006.

[16] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

[17] L. Goldstein and K. Messer. Optimal plug-in estimators for nonparametric functional estimation. *The annals of statistics*, 20(3):1306–1328, 1992.

[18] L. Gunter and J. Zhu. Computing the solution path for the regularized support vector regression. In *Advances in Neural Information Processing Systems (NIPS'05)*, 2005.

[19] M.W. Hirsch, S. Smale, and R. Devaney. *Differential equation, Dynamical Systems and an Introduction to Chaos*, volume 60 of *Pure and Applied Mathematical series*. Elsevier Academic Press, 2nde edition edition, 2003.

[20] S. Julier and J. Uhlmann. A new approach for filtering nonlinear systems. In *Proc. 1995 American Control Conference, Seattle*, 1995.

[21] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Automat. Contr.*, 45:477–482, 2000.

[22] Y.A. Kuznetsov. *Elements of Applied Bifurcation Theory.* Springer-Verlag, New York, 2004.

[23] N. Lalam and C. Klaassen. Pseudo-maximum likelihood estimation for differential equations. Technical Report 2006-18, Eurandom, 2006.

[24] J. Madar, J. Abonyi, H. Roubos, and F. Szeifert. Incorporating prior knowledge in cubic spline approximation - application to the identification of reaction kinetic models. *Industrial and Engineering Chemistry Research*, 42(17):4043–4049, 2003.

[25] Harley H. McAdams and Adam Arkin. It's a noisy business! genetic regulation at the nanomolar scale. *Trends in Genetics*, 15(2):65–69, June 1999.

[26] C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.

[27] C. G. Moles, Pedro Mendes, and J. R. Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.*, 13(11):2467–2474, 2003.

[28] I. Nachman, A. Regev, and Nir Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20:248–256, 2004. Suppl. 1.

[29] Dirk J.W. De Pauw and Bernard De Baets. Incorporating model identifiability into equation discovery of ode systems. In *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 2135–2140, New York, NY, USA, 2008. ACM.

[30] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and Florence d'Alche Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19:S138–S148, 2003.

[31] A. A. Poyton, M. S Varziri, K. B. McAuley, P. J McLellan, and J. O. Ramsay. Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers and Chemical engineering*, 30:698–708, 2006.

[32] M. Quach, N. Brunel, and F. d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23:3209 – 3216, 2007.

[33] J. O. Ramsay, G. Hooker, J. Cao, and D. Campbell. Parameter estimation for differential equations: A generalized smoothing approach. *Journal of the Royal Statistical Society (B)*, 2007. To appear.

[34] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer series in statistics. Springer, 1997.

[35] C. Rangel, J. A., Zoubin Ghahramani, M. Li, E. Sotheran, A. Gaiba, David L. Wild, and F. Falciani. Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361–1372, 2004.

[36] A. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2), 1999.

[37] D. Ruppert. *Semiparametric regression*. cup, 2003.

[38] S. Sarkka. Unscented rauch-tung-striebel smoother. *IEEE Trans. on Automatic Control*, 53(3):845–849, 2008.

[39] M. A. Savageau. Biochemical systems analysis. 3. dynamic solutions using a power-law approximation.. *J Theor Biol*, 26:215–226, 1970.

[40] Bernhard Schölkopf and Alex Smola. *Learning with kernels*. MIT Press, 2002.

[41] A. Sitz, U. Schwarz, J. Kurths, and H.U. Voss. Estimation of parameters and unobserved components for nonlinear systems from noisy time series. *Physical review E*, 66:016210, 2002.

[42] Alex Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2003.

[43] Paul Smolen, Douglas A. Baxter, and John H. Byrne. Modeling transcriptional control in gene networks - methods, recent results, and future directions. *Bulletin of Mathematical Biology*, 62:247 – 292, 2000.

[44] C. J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10(4):1040–1053, 1982.

[45] I. Swameye, T.G. Mulller, J. Timmer, O. Sandra, and U. Klingmuller. Identification of nucleocytoplasmic cycling as remote sensor in signaling by databased modeling. *PNAS*, 100:1028–1033, 2003.

[46] A.W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilities Mathematics. Cambridge University Press, 1998.

[47] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[48] J. M. Varah. A spline least squares method for numerical parameter estimation in differential equations. *SIAM J.sci. Stat. Comput.*, 3(1):28–46, 1982.

[49] E. O. Voit and J. Almeida. Decoupling dynamiclal systems for pathway identification from metabolic profiles. *Bioinformatics*, 20:1670–1681, 2004.

[50] Z. Zi. *User guide: SBML-PET, A sytems biology markup language based estimation parameter tool.* Max Planck Institute for Molecular Genetics, 2006. Available from www.sysbio.molgen.mpg.de/SBML-PET/.

[51] Z. Zi and E. Klipp. SBML-PET: a systems biology markup language-based parameter estimation tool. *Bioinformatics*, 22(21):2704–2705, 2006.