Inference of Switched Biochemical Reaction Networks Using Sparse Bayesian Learning

Wei Pan[1], Ye Yuan[2], Aivar Sootla[1], and Guy-Bart Stan[1]

[1]Centre for Synthetic Biology and Innovation and Department of Bioengineering, Imperial College London, {w.pan11, a.sootla, g.stan}@ imperial.ac.uk
[2]Department of Engineering, University of Cambridge, yy311@cam.ac.uk

Abstract. This paper proposes an algorithm to identify biochemical reaction networks with time-varying kinetics. We formulate the problem as a nonconvex optimisation problem casted in a sparse Bayesian learning framework. The nonconvex problem can be efficiently solved using Convex-Concave programming. We test the effectiveness of the method on a simulated example of DNA circuit realising a switched chaotic Lorenz oscillator.

Keywords: Switched system, Sparse Bayesian Learning, Convex-Concave programming.

1 Motivation, background and related work

Identification of switched systems, which are characterised by the interaction of both continuous and discrete dynamics, is widely used in many different fields such as systems/synthetic biology, econometrics, finance, biochemical engineering, social networks, etc. In this paper, we are interested in the identification of switched biochemical reaction networks. Biochemical processes can go through different phases in time; for example, a cell cycle in bacteria or diurnal alternations in plants. These switches are typically triggered by time dependent processes or by some external force. Therefore, the dynamics of biochemical reactions can be modelled as a collection of submodels amongst which switches occur over time. For biochemical reaction networks, the submodels are typically nonlinear due to mass action kinetics.

In classical switched system identification, the submodels are typically assumed to be linear or of the switch affine type [1], which is often used to approximate nonlinear dynamics. In [2], the structure of submodels is fixed and a minimal number of switches between submodels is inferred. However, these techniques are not generally applicable to biochemical kinetics due to highly nonlinear terms and model complexity. In the nonlinear case, there is an additional problem of nonlinear basis selection, which is fixed and predefined in the

linear case. Unlike the linear case, the number of nonlinear basis functions can be infinite and one might have to use complicated nonlinear functions to model the dynamics without any switches. In practice, if one is interested in obtaining the least number of switches, the number of nonlinear basis functions will typically grow, and vice versa, a small number of nonlinear basis functions will result in many switches. Hence there are two different and competing minimisation criteria: the number of switches between submodels and the number of basis functions in each submodel.

In this paper, we cast the problem of identification of switched biochemical reaction networks as a linear regression problem by defining a set of nonlinear basis functions based on mass action kinetics. Minimising the number of switches and/or the number of basis functions is typically addressed in such problems by an ℓ_1 or ℓ_2 regularisation approach. In this paper, however, we take a sparse Bayesian learning approach, which is shown to promote sparsity better than to ℓ_1 methods [3–5]. By specifying sparse priors on the number of parameters and the number of switches in this sparse Bayesian learning framework, the identification problem is formulated as a nonconvex optimisation problem. By exploiting the structure of the nonconvex optimisation problem, one can use Convex-Concave programming techniques to solve the problem efficiently. One illustrative example from DNA computation is used to show the effectiveness of the proposed method.

2 Preliminaries on biochemical reaction networks

Consider a biochemical system with n species $X_1, \ldots, X_n$. We denote the concentration of species X_j as x_j. Let $\mathcal{U}$ be the set of uni-species reactions and $\mathcal{B}$ be the set of bi-species reactions. A uni-species reaction $i \in \mathcal{U}$ is defined by the index $r_i \in \{1, \ldots, n\}$ of its single reactant species, the associated real-valued rate constant $k_i > 0$, and the integer product coefficients for each species $c_{i,j} \geq 0$: $m_i X_{r_i} \xrightarrow{k_i} c_{i,1}X_1 + \ldots + c_{i,n}X_n$. A bi-species reaction $i \in \mathcal{B}$ is defined by the indices $r_{i,1}, r_{i,2} \in \{1, \ldots, n\}$ of its two reactant species, the real-valued rate constant $k_i > 0$, and the integer product coefficients for each species $c_{i,j} \geq 0$: $m_i X_{r_{i,1}} + n_i X_{r_{i,2}} \xrightarrow{k_i} c_{i,1}X_1 + \ldots + c_{i,n}X_n$. Using the law of mass action, the dynamics of the concentrations $x_j \geq 0$ of species X_j are given according to the ordinary differential equations

$$\begin{aligned} \dot{x}_j = &- \sum_{i \in \mathcal{U} | r_i = j} k_i x_j^{m_i} - \sum_{i \in \mathcal{B} | r_{i,1} = j} k_i x_j^{m_i} x_{r_{i,2}}^{n_i} - \sum_{i \in \mathcal{B} | r_{i,2} = j} k_i x_{r_{i,1}}^{m_i} x_j^{n_i} \\ &+ \sum_{i \in \mathcal{U}} c_{i,j} k_i x_{r_i} + \sum_{i \in \mathcal{B}} c_{i,j} k_i x_{r_{i,1}} x_{r_{i,2}}, \end{aligned} \tag{2.1}$$

We can expand (2.1) for more than two species, though this can be rarely found in reality due to highly improbable simultaneous three-species molecular collision mechanisms.

Eq. (2.1) can be modelled using the general form: $\dot{\mathbf{x}} = \mathbf{S}\mathbf{v}(\mathbf{x})$, where $\mathbf{x}$ is the vector of species whose elements are x_j, $\mathbf{S}$ is the stoichiometry matrix and $\mathbf{v}(\mathbf{x})$ is a vector of propensity functions. The matrix $\mathbf{S}$ and the propensity vector $\mathbf{v}(\mathbf{x})$ can be built based on the biochemical reactions and their rates. Hence, without loss of generality we can assume that $\mathbf{S}$ is a matrix whose elements are real constants and $\mathbf{v}(\mathbf{x})$ is a vector whose elements are nonlinear functions of $\mathbf{x}$ as in (2.1). Biochemical processes can go through different phases in time; for example, a cell cycle in bacteria or diurnal alternations in plants. These switches, which are typically triggered by time dependent processes or by some external force, can be fitted into our model as follows: $\dot{\mathbf{x}} = \mathbf{S}^{\alpha(t)}\mathbf{v}(\mathbf{x})$, where $\alpha(t)$ is a sequence of integers in a bounded set and $\mathbf{S}^{\alpha(t)}$ takes values from an unknown set $\{\mathbf{S}_1, \ldots, \mathbf{S}_{N_{modes}}\}$ depending on time.

In what follows, we consider the system dynamics expressed in discrete-time and subjected to additive i.i.d. Gaussian noise $\xi(k)$ with known statistics.

$$\mathbf{x}(k+1) = \mathbf{S}^{\alpha(k)}\mathbf{v}(\mathbf{x}(k)) + \xi(k). \tag{2.2}$$

3 Problem formulation

3.1 Linear Regression Problem Formulation

Taking the transpose of both sides of (2.2) and considering the i^{th} state variable x_i of (2.2), we can obtain

$$\begin{aligned} x_i(k+1) &= v_i^\top(x(k)) \cdot \left(\mathbf{S}_{i,:}^{\alpha(k)}\right)^\top + \xi_i(k), \\ &= \big(f_{i1}(\mathbf{x}(k)) \ldots f_{iN}(\mathbf{x}(k)\big) \cdot \mathbf{w}_i(k) + \xi_i(k), \end{aligned} \tag{3.1}$$

where $\mathbf{S}_{i,:}^{\alpha(k)}$ represent the i^{th} row of $\mathbf{S}^{\alpha(k)}$; and f_{ij} represent the basis functions we use to reconstruct the model. The form of these functions can be any of those described in (2.1). In (3.1), $\mathbf{w}_i(k) = [w_{i1}(k), \ldots, w_{iN}(k)]^\top$, and the noise $\xi_i(k)$ is assumed to be i.i.d. Gaussian distributed: $\xi_i(k) \sim \mathcal{N}(0, \sigma_i^2)$, with $\mathbb{E}(\xi_i(p)) = 0$, $\mathbb{E}(\xi_i(p)\xi_i(q)) = \sigma_i^2\delta_{pq}$, with $\delta_{pq} = \begin{cases} 1, p = q \\ 0, p \neq q \end{cases}$. Now, let's assume that time-series measurements from a biochemical network are collected in a vector $\mathbf{y}_i$, where $\mathbf{y}_i = \big(x_i(2) \ldots x_i(M+1)\big)^\top$. We state the problem as identifying the system (2.2) based on these measurements. That is, our goal is to find all matrices $\mathbf{S}_1, \ldots, \mathbf{S}_{N_{modes}}$ and the switching sequence $\alpha(k)$ from the measurements stored in $\mathbf{y}_i$. Since the formulation in (3.1) is similar for all the state variables x_i, $i = 1, \ldots, N$, in what follows we drop the subscript i to ease the notation.

By defining the following block matrix and vectors

$$\begin{aligned}
\mathbf{A} &\triangleq \left[\begin{array}{ccc|c|ccc} f_1(\mathbf{x}(1)) & & & \cdot & & f_N(\mathbf{x}(1)) & \\ & \ddots & & \ddots & & \ddots & \\ & & f_1(\mathbf{x}(M)) & \cdot & & & f_N(\mathbf{x}(M)) \end{array}\right] \\
&= \left[\, A_1 \middle| \ldots \middle| A_N \,\right] \in \mathbb{R}^{M \times MN}, \\
\mathbf{w} &\triangleq \left[\, w_1(1), \ldots, w_1(M) \middle| \ldots \middle| w_N(1), \ldots, w_N(M) \,\right]^\top \\
&= \left[\, \mathbf{w}_1^\top \middle| \ldots \middle| \mathbf{w}_N^\top \,\right]^\top \in \mathbb{R}^{MN}, \\
\boldsymbol{\Xi} &\triangleq [\xi(1), \ldots, \xi(M)]^\top \in \mathbb{R}^M.
\end{aligned} \tag{3.2}$$

we can reformulate the linear regression equations in (3.1) as

$$\mathbf{y} = \mathbf{A}\mathbf{w} + \boldsymbol{\Xi}. \tag{3.3}$$

There are two issues that needs to be considered at this stage. First, each block $\mathbf{w}_i = [w_1(1), \ldots, w_1(M)]$ is associated only with certain basis function. The solution $\mathbf{w}$ to (3.3) is therefore typically going to be block sparse, which is mainly due to the potential introduction of non-relevant and/or non-independent basis functions in $\mathbf{A}$. Second, in the switched case, we have to penalise the number of switches from t_1 to t_M and/or the number of modes N_{modes}, which can be fixed in advance or set equal to M. Clearly such a problem has an infinite number of solutions, especially in the noisy setting. Therefore, we refine the problem statement to identify the system (2.2) with the least number of non-zero blocks in $\mathbf{w}$ and the least number of switches in the sequence $\alpha(k)$.

These are actually two different and competing criteria: if we want the least number of switches, the number of non-zero parameters in $\mathbf{w}$ will grow, and vice versa, a small number of non-zero parameters in $\mathbf{w}$ will result in many switches.

3.2 Minimising the Number of Switches

To limit the number of switches, we need to ensure that $S^{\alpha(k)}$ stays the same from time k to time $k+1$. Hence we need to add a condition maximising the sparsity of $S^{\alpha(k+1)} - S^{\alpha(k)}$ for all k. This leads to the following problem statement:

Problem 1. Given $\mathbf{y}$ and $\mathbf{A}$ and the block partitions formulated in (3.3), find a $\mathbf{w}$ that can explain the data with the minimal number of switches and the minimal number of non-zero blocks in $\mathbf{w}$.

If we index the vector $\mathbf{w}$ appropriately, the problem of minimising the number of switches can be formulated by enforcing $\mathbf{D}_j\mathbf{w}_j$ sparse, where the matrix $\mathbf{D}_j$ is defined as follows:

$$\mathbf{D}_j \triangleq \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(M-1)\times M}. \tag{3.4}$$

If we further define

$$\mathbf{B}_j \triangleq \begin{bmatrix} I \\ \rho D_j \end{bmatrix} \in \mathbb{R}^{(2M-1)\times M,}, \mathbf{B} \triangleq \begin{bmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_N \end{bmatrix} \in \mathbb{R}^{N(2M-1)\times MN}, \tag{3.5}$$

Problem 1 can be formulated as follows

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \lambda\|\mathbf{B}\mathbf{w}\|_{\ell_0}, \tag{3.6}$$

where ρ in (3.5) is a trade-off parameter between the number of switches and the number of non-zero parameters, while λ in (3.6) is known as a regularisation parameter in penalised linear regression problems. Using the specially designed matrix $\mathbf{B}$ defined in (3.5), we can penalise a) the number of switches that occur and b) the number of non-zero element in every identified model.

For matrices $\mathbf{B}$ with the special form given in (3.5), algorithms minimising the number of non-zero elements and the number of switches belong to the class of so-called *fused LASSO algorithms* [6]. For general $\mathbf{B}$ matrices, the problem defined in (3.6) would be solved using *generalised LASSO algorithms* [7].

Overall, instead of employing LASSO-type algorithms to obtain an approximated solution, we are going to tackle the problem from a sparse Bayesian learning perspective [3, 4] as this gives much sparser solutions.

4 Sparse Bayesian Learning

In order to estimate $\mathcal{P}(\mathbf{w}|\mathbf{y})$, firstly the prior distribution over $\mathbf{w}$ should be specified. In problem (3.6), we not only want to minimise the number of basis functions but also the number of switches. Therefore, sparsity promoting priors should be specified for $\mathcal{P}(\mathbf{B}_{j,:}\mathbf{w}_j)$, $\forall j$, where $\mathbf{B}_{j,:}$ is the j^{th} row of $\mathbf{B}$. These priors can be chosen as *super-Gaussian* [8]. It means that for every parameter $\mathbf{B}_{j,:}\mathbf{w}_j$, we define a hyper-parameter γ_j such that $\mathcal{P}(\mathbf{B}_{j,:}\mathbf{w}_j) = \max_{\gamma_j>0} \mathcal{N}(\mathbf{B}_{j,:}\mathbf{w}_j|0,\gamma_j)\varphi(\gamma_j)$. In this case the priors $\mathcal{P}(\mathbf{B}\mathbf{w})$ can be computed as follows:

$$\mathcal{P}(\mathbf{B}\mathbf{w}) = \max_{\boldsymbol{\gamma}>\mathbf{0}} \prod_j \mathcal{N}(\mathbf{B}_{j,:}\mathbf{w}_j|0,\gamma_j)\varphi(\gamma_j). \tag{4.1}$$

where $\boldsymbol{\gamma}$ is a vector of γ_j and $\varphi(\cdot)$ is a nonnegative function of the hyperparameters, which can be given depending on a selection specific sparsity promoting distribution, such as a Laplace distribution, a Student's t distribution, etc. Note that, if the parameter vector $\boldsymbol{\gamma}$ is known, we can estimate $\mathcal{P}(\mathbf{B}\mathbf{w}|\mathbf{y};\boldsymbol{\gamma})$ instead of computing $\mathcal{P}(\mathbf{B}\mathbf{w}|\mathbf{y})$. Therefore, the problem should be recasted in terms of finding the most appropriate hyperparameters of the priors: $\hat{\boldsymbol{\gamma}}$. A good way of selecting $\hat{\boldsymbol{\gamma}}$ is to choose it as the minimiser of the sum of the misaligned probability

mass, e.g.,

$$\begin{aligned}\hat{\boldsymbol{\gamma}} &= \underset{\boldsymbol{\gamma}>\mathbf{0}}{\operatorname{argmin}} \int \mathcal{P}(\mathbf{y}|\mathbf{w})\,|\mathcal{P}(\mathbf{B}\mathbf{w}) - \mathcal{P}(\mathbf{w};\boldsymbol{\gamma})|\,d\mathbf{w} \\ &= \underset{\boldsymbol{\gamma}>\mathbf{0}}{\operatorname{argmax}} \int \mathcal{P}(\mathbf{y}|\mathbf{w})\mathcal{P}(\mathbf{B}\mathbf{w};\boldsymbol{\gamma})d\mathbf{w}.\end{aligned} \tag{4.2}$$

The procedure in (4.2) is referred to as evidence/marginal likelihood maximisation [3, 4]. It means that the marginal likelihood can be maximised by selecting the most probable hyperparameters able to explain the observed data. Defining $\boldsymbol{\Gamma}$ as a diagonal matrix with diagonal entries γ_j, the parameters $\mathbf{w}$ and $\boldsymbol{\gamma}$ can be estimated by solving the optimisation problem in Proposition 1:

Proposition 1. *The optimisation problem in* (4.2) *is equivalent to the following non-convex problem*

$$\begin{aligned}\min_{\boldsymbol{\gamma}>\mathbf{0},\mathbf{w}} \{&\frac{1}{\sigma^2}\|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \mathbf{w}^\top \mathbf{B}^\top \boldsymbol{\Gamma}^{-1}\mathbf{B}\mathbf{w} \\ &+ \log|\boldsymbol{\Gamma}| + \log|\mathbf{B}^\top\boldsymbol{\Gamma}^{-1}\mathbf{B} + \sigma^{-2}\mathbf{A}^\top\mathbf{A}| + \sum_{j=1}^{N} p(\gamma_j)\}\end{aligned} \tag{4.3}$$

where $\boldsymbol{\Gamma}$ *is a diagonal matrix with entries* $\boldsymbol{\gamma}$ *on the diagonal and* $p(\cdot) = \log(\varphi(\cdot))$.

Proof. The proof is similar to that derived in [4, 5]. Therefore, we omit it due the space limitation.

We approach the solution to this problem by separating the objective function into the following parts:

$$\begin{aligned}f(\mathbf{w},\boldsymbol{\gamma}) &= \frac{1}{\sigma^2}\|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 + \mathbf{w}^\top \mathbf{B}^\top \boldsymbol{\Gamma}^{-1}\mathbf{B}\mathbf{w} \\ g(\boldsymbol{\gamma}) &= \log|\boldsymbol{\Gamma}| + \log|\mathbf{B}^\top\boldsymbol{\Gamma}^{-1}\mathbf{B} + \sigma^{-2}\mathbf{A}^\top\mathbf{A}| + \sum_{j=1}^{N} p(\gamma_j).\end{aligned}$$

Proposition 2. *The function* $f(\mathbf{w},\boldsymbol{\gamma})$ *is jointly convex in* $\mathbf{w}$ *and* $\boldsymbol{\gamma}$*, while the function* $g(\boldsymbol{\gamma})$ *is concave.*

Proof. It is easy to verify the first part of this proposition. A proof on concavity of the sum of log-determinant functions in the second part for general matrices $\mathbf{B}$ can be found in [5, Theorem 3.1 (3)].

Proposition 2 allows us to use Convex-Concave Programming [9] in order to find a stationary point, which results in:

$$\left(\mathbf{w}^{k+1}, \boldsymbol{\gamma}^{k+1}\right) = \underset{\mathbf{w},\boldsymbol{\gamma}>\mathbf{0}}{\operatorname{argmin}} f(\mathbf{w},\boldsymbol{\gamma}) + \nabla_{\boldsymbol{\gamma}}(g(\boldsymbol{\gamma}^k))^\top\boldsymbol{\gamma} \tag{4.4}$$

In order to make the algorithm more transparent we also separate the minimisation into separate minimisation programmes over $\mathbf{w}$ and $\boldsymbol{\gamma}$: By defining $\boldsymbol{\epsilon}^{k+1} \triangleq \nabla_{\boldsymbol{\gamma}}(g(\boldsymbol{\gamma}^k))$, the optimal solution in (4.4) over $\boldsymbol{\gamma}$ can be computed analytically as $\gamma_j = \mathbf{B}_{j,:}\mathbf{w}/\sqrt{\epsilon_j}$, $\forall j$ and for every fixed $\mathbf{w}$. Now we only need to minimise in (4.4) over $\mathbf{w}$ as follows:

$$\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A}\mathbf{w} - \mathbf{y}\|_2^2 + \sigma^2 \sum_j \|\epsilon_j^k \cdot \mathbf{B}_{j,:}\mathbf{w}\|_1,$$

while the hyperparameters are updated as $\gamma_j^{k+1} = \mathbf{B}_{j,:}\mathbf{w}^{k+1}/\sqrt{\epsilon_j}$, $\forall j$. To summarise the algorithm, one can initialise γ_j^0 at any positive real scalar. Some additional insight can be obtained by initialising $\epsilon_j^0 = 1$, $\forall j$ instead. In that case, the first iteration becomes a linear regression problem with ℓ_1 penalty on the parameters $\mathbf{B}\mathbf{w}$:

$$\mathbf{w}^1 = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A}\mathbf{w} - \mathbf{y}\|_2^2 + \sigma^2 \|\mathbf{B}\mathbf{w}\|_{\ell_1}.$$

Then we update γ_j^1 using $\gamma_j^1 = \mathbf{B}_{j,:}\mathbf{w}^1/\sqrt{\epsilon_j^0}$. Using this initialisation, we provably get results at least not worse than the generalised LASSO algorithm. Algorithm 1 summarises this approach, which converges to a stationary point in $\mathbf{w}$ and $\boldsymbol{\gamma}$ [9]. Algorithm 1 can be seen as a particular version of the reweighted LASSO approach with a Bayesian update on the weights. The program (4.4) is

Algorithm 1 Switched Systems Identification Algorithm

1: Initialise $\epsilon_j^0 = 1,\ \forall j = 1, \ldots, N(2M-1)$,
2: **for** $k = 0, \ldots, k_{\max}$ **do**
3: Update the parameters as follows:

$$\mathbf{w}^{k+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A}\mathbf{w} - \mathbf{y}\|_2^2 + \sigma^2 \sum_j \|\epsilon_j^k \cdot \mathbf{B}_{j,:}\mathbf{w}\|_1$$

$$\boldsymbol{\gamma}_j^{k+1} = \frac{\mathbf{B}_{j,:}\mathbf{w}^{k+1}}{\sqrt{\epsilon_j^k}}$$

$$\epsilon_j^{k+1} = -\frac{\mathbf{B}_{j,:}(\mathbf{B}^\top(\boldsymbol{\Gamma}^{k+1})^{-1}\mathbf{B} + \rho^k \mathbf{A}^\top\mathbf{A})^{-1}\mathbf{B}_{j,:}^\top}{(\gamma_j^{k+1})^2} + \frac{1}{\gamma_j^{k+1}}$$

4: **if** a stopping criterion is satisfied **then**
5: Break
6: **end if**
7: **end for**

convex, quadratic and uncostrained; however, the size of the problem can be extremely large. Two techniques to speed-up the solution can be adopted: pruning

the parameter $\mathbf{w}$ space after each iteration as in [10], and/or using distributed computation methods such as ADMM, e.g. [11]).

5 Results

In this section, we consider time-series data obtained from a chaotic Lorenz Oscillator implemented *in vitro* using DNA computations [12]. From the associated biochemical reactions, a polynomial ODE can be derived using the law of mass action. We artificially generate data using this oscillator model but change certain parameters at certain time. This can be realised *in vitro* by changing experiment conditions or enzyme concentrations. The Lorenz oscillator can be described by the discretised differential equations

$$\begin{aligned}&\left[\frac{y_1(k+1)-y_1(k)}{\delta t}, \frac{y_2(k+1)-y_2(k)}{\delta t}, \frac{y_3(k+1)-y_3(k)}{\delta t}\right]\\ =&\left[p_1(k)(y_2(k)-y_1(k)), y_1(k)(p_2(k)-y_2(k)), y_1(k)y_2(k)-k_2(k)y_3(k)\right].\end{aligned}$$

where we the fix sampling time to $\delta t = 0.02$ (arbitrary units).

Initially ("Mode 1"), the parameters are $p_1 = 10, p_2 = 30, p_3 = 8/3$. From $k = 201$ to $k = 400$ ("Mode 2"), the parameters are changed to $p_1 = 10, p_2 = 30, p_3 = 4$. For the kinetics of y_1 and y_3, the nonlinear dynamics change after switching from Mode 1 to Mode 2. For y_2, the parameters do not switch. We construct the basis functions in (3.1) as

$$\left(y_1^0(k)y_2^0(k)y_3^0(k),\, y_1^0(k)y_2^0(k)y_3^1(k),\, \ldots,\, y_1^{n_1}y_2^{n_2}(k)y_3^{n_3}(k)\right).$$

We index the parameter vector $\mathbf{w}(k)$ as $[w^{000}(k), w^{001}(k), \ldots, w^{n_1n_2n_3}(k)]$, choose $\lambda = 1$ and $\rho = 100$ and set the initial condition to $[y_1(1), y_2(1), y_3(1)] = [0.2444, -2.217, 2.314]$. Finally, we set $n_1 = 1$, $n_2 = 1$ and $n_3 = 1$. The true and estimated parameters' evolution over time are shown in Figure1.

6 Conclusion

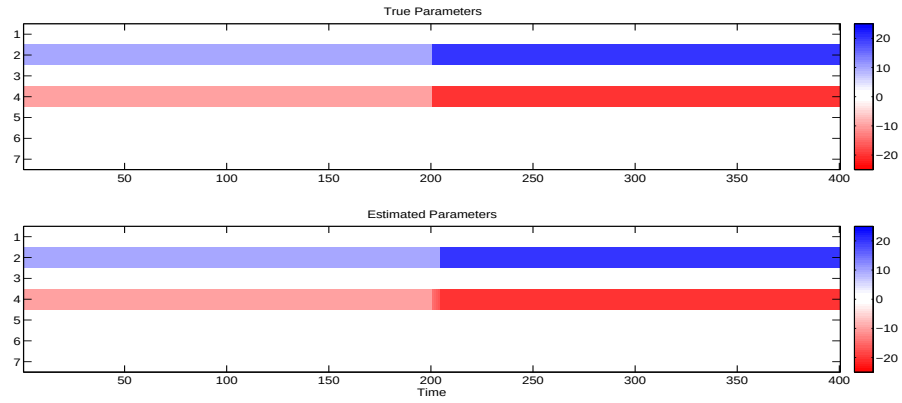
In this paper we proposed an efficient way to solve the switched system identification problem for biochemical reaction networks. For this purpose, an efficient framework based on sparse Bayesian learning has been proposed to solve this problem by specifying sparse priors on the number of parameters and the number of switches. Future work lies in the identifiability of such switching systems and how to design proper excitation signals to guarantee identifiability of the switching systems from output data.

7 Acknowledgement

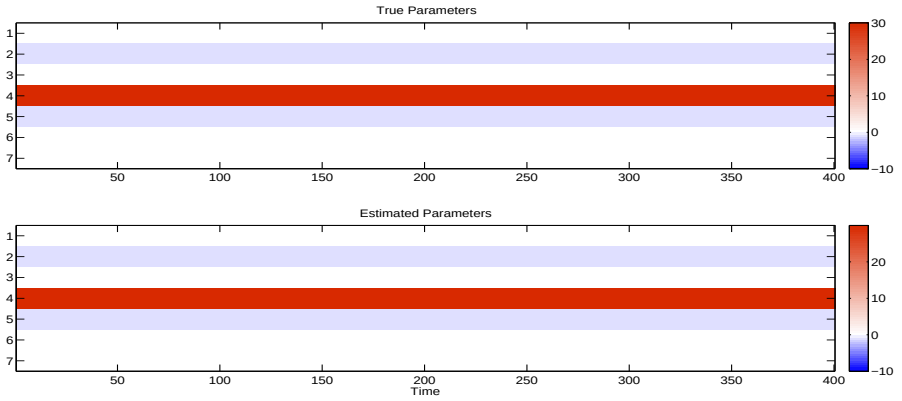
Mr W. Pan gratefully acknowledges the support of Microsoft Research through the PhD Scholarship Program. Dr A. Sootla and Dr G.-B. Stan acknowledge the support of EPSRC through the project EP/J014214/1 and the EPSRC Science and Innovation Award EP/G036004/1. Dr Y. Yuan acknowledge the support from EPSRC (project EP/I03210X/1).

References

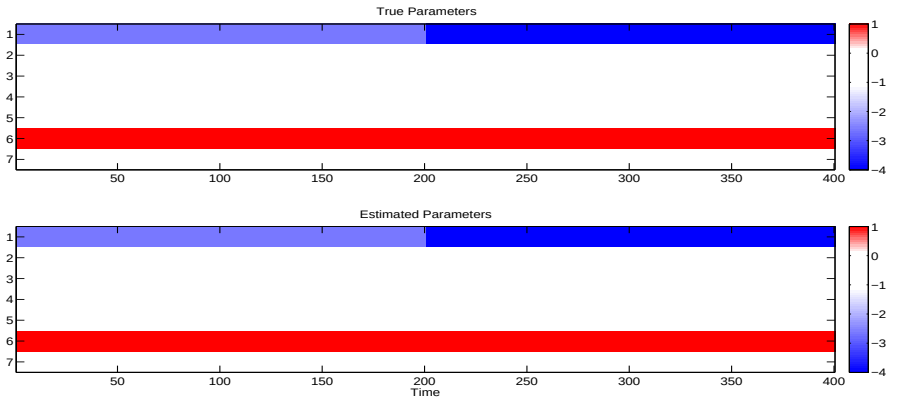
1. S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *European journal of control*, vol. 13, no. 2, pp. 242–260, 2007.
2. N. Ozay, M. Sznaier, C. M. Lagoa, and O. I. Camps, "A sparsification approach to set membership identification of switched affine systems," *Automatic Control, IEEE Transactions on*, vol. 57, no. 3, pp. 634–648, 2012.
3. M. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
4. D. Wipf, B. Rao, and S. Nagarajan, "Latent variable bayesian models for promoting sparsity," *Information Theory, IEEE Transactions on*, vol. 57, no. 9, pp. 6236–6255, 2011.
5. M. W. Seeger and H. Nickisch, "Large scale bayesian inference and experimental design for sparse linear models," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 166–199, 2011.
6. R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.
7. R. J. Tibshirani, *The solution path of the generalized lasso*. Stanford University, 2011.
8. J. Palmer, K. Kreutz-delgado, B. D. Rao, and D. P. Wipf, "Variational em algorithms for non-gaussian latent variable models," in *Advances in neural information processing systems*, 2005, pp. 1059–1066.
9. B. K. Sriperumbudur and G. R. Lanckriet, "On the convergence of the concave-convex procedure." in *NIPS*, vol. 9, 2009, pp. 1759–1767.
10. W. Pan, Y. Yuan, J. Gonçalves, and G.-B. Stan, "Bayesian approaches to nonlinear network reconstruction," *submitted*, 2013.
11. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
12. D. Soloveichik, G. Seelig, and E. Winfree, "Dna as a universal substrate for chemical kinetics," *Proceedings of the National Academy of Sciences*, vol. 107, no. 12, pp. 5393–5398, 2010.



(a) True and estimated parameters for y_1.



(b) True and estimated parameters for y_2.



(c) True and estimated parameters for y_3.

Fig. 1: True (upper panel) and estimated (lower panel) parameters' evolution over time. The horizontal axis represents time, whereas the vertical axis represents the estimated coefficients. From top to bottom, the index goes from 001 to 111.