Estimating the probability of success of a simple algorithm for switched linear regression

Fabien Lauer

Estimating the probability of success of a simple algorithm for switched linear regression

Fabien Lauer

Université de Lorraine, LORIA, UMR 7503, F-54506 Vandœuvre-lès-Nancy, France
CNRS
Inria

Abstract

This paper deals with the switched linear regression problem inherent in hybrid system identification. In particular, we discuss k-LinReg, a straightforward and easy to implement algorithm in the spirit of k-means for the nonconvex optimization problem at the core of switched linear regression, and focus on the question of its accuracy on large data sets and its ability to reach global optimality. To this end, we emphasize the relationship between the sample size and the probability of obtaining a local minimum close to the global one with a random initialization. This is achieved through the estimation of a model of the behavior of this probability with respect to the problem dimensions. This model can then be used to tune the number of restarts required to obtain a global solution with high probability. Experiments show that the model can accurately predict the probability of success and that, despite its simplicity, the resulting algorithm can outperform more complicated approaches in both speed and accuracy.

Keywords: switched regression, system identification, switched linear systems, piecewise affine systems, sample size, nonconvex optimization, global optimality

1. Introduction

This paper deals with hybrid dynamical system identification and more precisely with the switched linear regression problem. In this framework, a set of linear models are estimated in order to approximate a noisy data set with minimal error under the assumption that the data are generated by a switched system. We consider that the number of linear models is fixed a priori. Note that, even with this assumption, without knowledge of the classification of the data into groups associated to each one of the linear models, this problem is NP-hard [1] and is thus naturally intractable for data living in a high dimensional space. However, even for data in low dimension, algorithmic efficiency on large data sets remains an open issue.

Related work. As witnessed by the seminal works [2, 3, 4, 5, 6] reviewed in [1], the switched regression problem has been extensively studied over the last decade for the identification of hybrid dynamical systems. While methods in [3, 4] focus on piecewise affine systems, the ones in [2, 5, 6] equivalently apply to arbitrarily switched systems. Due to the NP-hard nature of the problem, all methods, except the global optimization approach of [4], yield suboptimal solutions in the sense that they are based on local optimization algorithms or only solve an approximation of the original nonconvex problem. However, each approach increased our understanding of hybrid system identification with a different point of view. By studying the noiseless setting, the algebraic approach [2, 7] showed how to cast the problem as a linear system of equations. The clustering-based method [3] proposed a mapping of the data into a feature space where the submodels

Email address: fabien.lauer@loria.fr (Fabien Lauer)
URL: http://www.loria.fr/~lauer (Fabien Lauer)

become separable. The Bayesian approach [5] analyzed the problem in a probabilistic framework, while the bounded-error approach [6] switched the focus by investigating the estimation of the number of modes for a given error tolerance. Each one of these offers a practical solution to deal with a specific case: noiseless for [2], with few data for [4], with prior knowledge on parameters for [5] or on the noise level for [6].

But despite this activity, most of the proposed approaches have strong limitations on the dimension of the data they can handle and are mostly applicable to small data sets with less than a thousand points and ten regressors. The algebraic approach [2, 7] provides a closed form solution, which can be very efficiently computed from large data sets, but which is only valid in the noise-free setting and rather sensitive to noise otherwise. Robust formulations of this approach exist [8, 9], but these still suffer from a major limitation inherent in the algebraic approach: the complexity grows exponentially with respect to the dimension of the data and the number of modes. This issue is also critical for the approach of [10] which, for small data dimensions, efficiently deals with noise in large data sets through a global optimization strategy applied to a continuous cost function. The recent method of [11], based on sparse optimization, circumvents the issue of the number of modes by iteratively estimating each parameter vector independently, in the spirit of the bounded-error approach [6]. However, the method relies on an ℓ_1-norm relaxation of a sparse optimization problem, which requires restrictive conditions on the fraction of data generated by each mode to apply. In particular, when the number of modes increases, the assumption on the fraction of data generated by a single mode becomes less obvious. Other works on convex relaxation of sparse optimization formulations include [12, 13], but the number of variables and of constraints in these convex optimization problems quickly grow with the number of data.

In a realistic scenario, say with noise, more than a thousand data points and more than two modes, globally optimal solutions (such as those obtained by [4]) cannot be computed in reasonable time and little can be stated on the quality of the models trained by approximate or local methods. Even for convex optimization based methods [11, 9, 13], the conditions under which the global minimizer of the convex relaxation coincides with the one of the original problem can be unclear or violated in practice. In this context, experimentally asserted efficiency and performance of an algorithm are of primary interest. In this respect, the paper shows empirical evidence that a rather straightforward approach to the problem can outperform other approaches from the recent literature in low dimensional problems while increasing the range of tractable problems towards larger dimensions.

Methods and contribution. This paper considers one of the most straightforward and easy to implement switched regression method and analyzes the conditions under which it offers an accurate solution to the problem with high probability. The algorithm discussed here is inspired by classical approaches to clustering and the k-means algorithm [14], hence its name: k-LinReg. As such, it is a local optimization method based on alternatively solving the problem (to be specified later) with respect to integer and real variables. The key issue in such an approach is therefore how to obtain a solution sufficiently close to the global one, which is typically tackled by multiple initializations and runs of the algorithm, but without guarantees on the quality of the result. In this paper, we focus on random initializations of the k-LinReg algorithm and the estimation of the probability of drawing an initialization leading to a satisfactory solution. In particular, we show how this probability is related to the number of data and how the number of random initializations can be chosen *a priori* to yield good results. This analysis is based on a random sampling of both the problem and initialization spaces to compute the estimates of the probability of success. Inspired by works on probabilistic methods [15], probabilistic bounds on the accuracy of these estimates are derived. These bounds provide the ground to consider the estimates as data for the subsequent estimation of a predictive model of the probability of success, from which the number of restarts from different initializations for a particular task can be inferred.

This study also reveals a major difference with the classical k-means problem, namely, that high dimensional problems can be solved efficiently and globally if the number of data is large enough. Compared with other approaches to switched linear regression, the computing time of the proposed method can even decrease when the number of data increases for high dimensional problems due to a reduced number of required restarts. Note that the approach developed in [16] has some similarities with the k-LinReg algorithm discussed here, but also some differences, for instance with respect to working in a recursive or batch

manner. In addition, the issue of the convergence towards a global solution was not clearly addressed in [16], whereas it is the central subject of the proposed analysis.

Beside these aspects, the paper also provides new insights into the inherent difficulty of hybrid system identification problems measured through the probability of success for the proposed baseline method. In particular, numerical examples show that test problems typically considered in the literature can be solved with few randomly initialized runs of the k-LinReg algorithm.

Paper organization. The paper starts by formally stating the problem in Section 2. The k-LinReg algorithm is presented in Section 3 and Section 4 is devoted to the study of its ability to find a solution close enough to the global one. Then, these results are used to build a model of its probability of success in Sect. 5, on the basis of which the number of restarts is computed in Sect. 5.2. Finally, Section 6 provides numerical experiments to test the proposed model and compares the final algorithm with some of the most recent approaches for hybrid system identification.

2. Problem formulation

Consider switched linear regression as the problem of learning a collection of n linear models $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T \boldsymbol{x},\ j = 1, \ldots, n$, with parameter vectors $\boldsymbol{w}_j \in \mathbb{R}^p$ from a training set of N pairs $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$, where $\boldsymbol{x}_i$ is the regression vector and y_i the observed output. In particular, we focus on least squares estimates of $\{\boldsymbol{w}_j\}_{j=1}^n$, i.e., parameter vector estimates minimizing the squared error, $(y_i - f(\boldsymbol{x}_i))^2$, over the training set. The model f is a switching model in the sense that the output $f(\boldsymbol{x})$ for a given input $\boldsymbol{x}$ is computed as the output of a single submodel f_j, i.e., $f(\boldsymbol{x}) = f_\lambda(\boldsymbol{x})$, where λ is the index of the particular submodel being used, which we refer to as the *mode* of $\boldsymbol{x}$.

When i is the time index and the regression vector is built from lagged outputs y_{i-k} and inputs u_{i-k} of a system as $\boldsymbol{x}_i = [y_{i-1}, \ldots, y_{i-n_y}, u_{i-n_a}, \ldots, u_{i-n_b}]^T$, we obtain a class of hybrid dynamical systems known as the Switched Auto-Regressive eXogenous (SARX) model, compactly written as

$$y_i = \boldsymbol{w}_{\lambda_i}^T \boldsymbol{x}_i + e_i,$$

with an additive noise term e_i. In this setting, the identification of the SARX model can be posed as a switched linear regression problem.

In this paper, we are particularly interested in instances of this switched linear regression problem with large data sets and focus on the case where the number of linear models n is fixed *a priori*[1]. Note that, even with this assumption, the problem is nontrivial and highly nonconvex. Indeed, given a data set, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, the least squares estimates of the parameters of a switched linear model with n modes, $\{\boldsymbol{w}_j\}_{j=1}^n$, are obtained by solving the following mixed integer optimization problem.

Problem 1 (Least squares switched linear regression)**.**

$$\begin{aligned} \min_{\{\boldsymbol{w}_j\},\{\beta_{ij}\}} \quad & \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \beta_{ij}\ (y_i - \boldsymbol{w}_j^T \boldsymbol{x}_i)^2 \\ \text{s.t.} \quad & \beta_{ij} \in \{0, 1\},\ i = 1, \ldots, N,\ j = 1, \ldots, n \\ & \sum_{j=1}^{n} \beta_{ij} = 1,\ i = 1, \ldots, N, \end{aligned}$$

where the β_{ij} are binary variables coding the assignment of point i to submodel j.

[1] Any data set can be arbitrarily well approximated by an unbounded set of linear models, e.g., by considering a linear model for each data point.

Problem 1 belongs to one of the most difficult class of optimization problems, namely, mixed-integer nonlinear programs, and is known to be NP-hard [1]. For a particular choice of model structure (hinging hyperplanes, that are only suitable for piecewise affine system identification), it can be transformed into a mixed-integer quadratic program, as detailed in [4]. However, even in this "simplified" form, it remains NP-hard and its global solution cannot be obtained except for very small instances with few data points [4].

In order to deal with numerous data while maintaining the dimension of the problem as low as possible, the following minimum-of-error reformulation of Problem 1 can be considered [10].

Problem 2 (Minimum-of-error formulation)**.**

$$\min_{\boldsymbol{w}} \; F(\boldsymbol{w}),$$

where

$$F(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \min_{j\in\{1,\dots,n\}} (y_i - \boldsymbol{w}_j^T \boldsymbol{x}_i)^2, \tag{1}$$

and where $\boldsymbol{w} = [\boldsymbol{w}_1^T \;\dots\; \boldsymbol{w}_n^T]^T$ *is the concatenation of all model parameter vectors* $\boldsymbol{w}_j$*.*

The equivalence between Problem 1 and 2 is given by the following proposition (see the proof in Appendix A). In particular, Proposition 1 shows that an optimal solution[2] to Problem 1 can be directly obtained from a global solution of Problem 2.

Proposition 1. *Problems 1 and 2 are equivalent under the simple mapping*

$$\beta_{ij} = \begin{cases} 1, & \text{if } j = \arg\min_{k=1,\dots,n} (y_i - \boldsymbol{w}_k^T \boldsymbol{x}_i)^2, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1,\dots,N,\; j = 1,\dots,n. \tag{2}$$

Moreover, this mapping is optimal in the sense that no other choice of $\{\beta_{ij}\}$ *leads to a lower value of the cost function in Problem 1.*

Though being equivalent to Problem 1, Problem 2 has the advantage of being a continuous optimization problem (with a continuous objective function and without integer variables) involving only a small number of variables equal to $n \times p$. To see this, note that the cost function (1) can be rewritten as

$$F(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{x}_i, y_i, \boldsymbol{w}),$$

where the loss function ℓ is defined as the point-wise minimum of a set of n loss functions $\ell_j(\boldsymbol{x}_i, y_i, \boldsymbol{w}) = (y_i - \boldsymbol{w}_j^T \boldsymbol{x}_i)^2$, $j = 1,\dots,n$. Since the point-wise minimum of a set of continuous functions is continuous, F is continuous (though non-differentiable) with respect to the variables $\boldsymbol{w}$.

Nonetheless, Problem 2 is a nonconvex problem which may have many local minima. In particular, switching from Problem 1 to 2 does not alleviate the NP-hardness, but simply changes the focus in terms of complexity from the number of data to the number of modes, n, and the dimension, p. This paper will therefore concentrate on the main issue of finding a solution to Problem 2 that is sufficiently close to a global one.

Remark 1. *The paper focuses on the error measured by the cost function* F*, and more particularly on the gap between* $F(\boldsymbol{w})$ *obtained with estimates* $\boldsymbol{w}$ *and* $F(\boldsymbol{\theta})$ *obtained with the true parameters* $\boldsymbol{\theta}$*, rather than on classification errors. While* $|F(\boldsymbol{w}) - F(\boldsymbol{\theta})|$ *can be zero, classification errors are unavoidable in hybrid system identification due to so-called undecidable points that lie at the intersection between submodels and for which*

[2]Multiple solutions with the same cost function value exist due to the symmetry of Problems 1 and 2. These can all be recovered from a single solution by swapping parameter vectors as $\boldsymbol{w}_j \leftrightarrow \boldsymbol{w}_k$.

one cannot determine the true mode solely on the basis of $(\boldsymbol{x}_i, y_i)$*. Though all switched system identification methods are subject to such misclassifications, these errors have a limited influence on parameter estimates, since they correspond by definition to data points that are consistent with more than one submodel. We refer to [6] for a more in-depth discussion on this issue and some corrective measures that can be employed in piecewise affine system identification.*

3. The *k*-LinReg algorithm

The k-LinReg algorithm builds on the relationship between Problem 1 and the classical unsupervised classification (or clustering) problem. These problems share the common difficulty of simultaneously computing a classification of the data points (through the binary variables β_{ij}) and a model of each group of points. In the clustering literature, the baseline method typically used to solve such problems is the k-means algorithm [14], which alternates between assignments of data points to groups and updates of the model parameters. Applying a similar strategy in the context of switched regression leads to the k-LinReg algorithm, which is depicted in Algorithm 1, where we let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$ be the regression matrix and $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ be the target output vector.

Algorithm 1 k-LinReg

Require: the data set $(\boldsymbol{X}, \boldsymbol{y}) \in \mathbb{R}^{N \times p} \times \mathbb{R}^N$, the number of modes n and an initial vector $\boldsymbol{w}^0 = [{\boldsymbol{w}_1^0}^T, \ldots, {\boldsymbol{w}_n^0}^T]^T$.
Initialize $t \leftarrow 0$.
repeat
Classify the data points according to

$$\lambda_i^t = \arg \min_{j \in \{1,\ldots,n\}} (y_i - {\boldsymbol{w}_j^t}^T \boldsymbol{x}_i)^2, \qquad i = 1, \ldots, N. \tag{3}$$

for $j = 1$ **to** n **do**
if $|\{i \ : \ \lambda_i^t = j\}| < p$ **then**
return $\boldsymbol{w}^* = \boldsymbol{w}^t$ and $F(\boldsymbol{w}^*)$.
end if
Build the matrix $\boldsymbol{X}_j^t$, containing all the ith rows of $\boldsymbol{X}$ for which $\lambda_i^t = j$, and the target vector $\boldsymbol{y}_j^t$ with the corresponding components from $\boldsymbol{y}$.
Update the model parameters for mode j with

$$\boldsymbol{w}_j^{t+1} = \arg \min_{\boldsymbol{w}_j \in \mathbb{R}^p} \ \|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t \boldsymbol{w}_j\|_2^2. \tag{4}$$

The solution to this least-squares problem is obtained via the pseudo-inverse of $\boldsymbol{X}_j^t$ and can be given in closed form if $\boldsymbol{X}_j^t$ is full rank by

$$\boldsymbol{w}_j^{t+1} = ({\boldsymbol{X}_j^t}^T \boldsymbol{X}_j^t)^{-1} {\boldsymbol{X}_j^t}^T \boldsymbol{y}_j^t,$$

or through the singular value decomposition of $\boldsymbol{X}_j^t$.
end for
Increase the counter $t \leftarrow t + 1$.
until convergence, e.g., until

$$||\boldsymbol{w}^{t+1} - \boldsymbol{w}^t||_2 \leq \epsilon,$$

or no more changes occur in the classification.
return $\boldsymbol{w}^* = \boldsymbol{w}^{t+1}$ and $F(\boldsymbol{w}^*)$.

The IF statement in the FOR loop of Algorithm 1 simply ensures that the subsequent update of $\boldsymbol{w}_j^{t+1}$ can proceed with sufficient data. If not, then the algorithm returns the current solution before updating (which will usually correspond to a failure in the terms of Section 4). Such situations are also typical in the k-means algorithm, where a group of points can become empty. In this case, possible refinements include resetting the corresponding parameter vector to a random value or merely dropping it to obtain a final model with fewer modes. However, since the aim of the paper is to analyze the most simple version of the k-LinReg algorithm, these strategies will not be considered.

Algorithm 1 can be interpreted as a block-coordinate descent algorithm, where the cost function in Problem 1 is alternatively optimized over two sets of variables: the discrete and the real ones. Convergence towards a local minimum is guaranteed by the following proposition that considers the equivalent form of Problem 2 (see the proof in Appendix B).

Proposition 2. *Algorithm 1 monotonically decreases the cost function $F(\boldsymbol{w}^t)$ in (1), in the sense that $\forall t \geq 0,\ F(\boldsymbol{w}^{t+1}) \leq F(\boldsymbol{w}^t)$.*

The k-LinReg algorithm is guaranteed to converge to a local minimum of the cost function (1). However, such cost functions may exhibit a number of local minima, many of which are not good solutions to the original problem of switched linear modeling. In order to obtain a good local minimum, if not a global one, the procedure must be restarted from different initializations. The following sections will provide an indication on the number of restarts required to obtain a good solution with high probability.

Remark 2. *In [3], a clustering-based technique was proposed for hybrid system identification. It is "clustering-based" in the sense that a particular inner step of this method uses the k-means algorithm to classify the data points mapped in some feature space. Here, the approach is different in that the entire problem is solved by an adaptation of k-means to switched regression, where the parameter vectors are estimated together with the classification which takes place in the original regression space of the data. Also, note that the method of [3] only applies to piecewise affine systems, whereas the k-LinReg algorithm is designed for arbitrarily switched systems.*

4. Estimating the probability of success

One important issue when using local algorithms such as k-LinReg is the evaluation of their performance, i.e., their ability to reach global optimality. Here, we aim at a probabilistic evaluation under a random sampling of both the initialization and the data. More precisely, we will measure the probability of success, defined for given n, p, N and $\varepsilon \geq 0$ as

$$P_{success}(n, p, N, \varepsilon) = 1 - P_{fail}(n, p, N, \varepsilon) \tag{5}$$

with

$$P_{fail}(n, p, N, \varepsilon) = P_{\boldsymbol{w},\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}\{F(\boldsymbol{w}^*; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) - F(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) \geq \varepsilon\},$$

where $F(\boldsymbol{w}^*; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) = F(\boldsymbol{w}^*)$ emphasizes the dependency of the cost function (1) on the data and where all variables are now random variables. In particular, $\boldsymbol{w}^*$ is the random vector defined as the output of the k-LinReg algorithm initialized at $\boldsymbol{w}$ and applied to a data set generated by the random vector $\boldsymbol{\theta}$ with $y_i = \boldsymbol{\theta}_{\lambda_i}^T \boldsymbol{x}_i + e_i$, $i = 1, \ldots, N$, $\boldsymbol{\lambda}$ is a random switching sequence, $\boldsymbol{e}$ is a random noise sequence and $\boldsymbol{X}$ a random regression matrix with rows $\boldsymbol{x}_i^T$. Note that the randomness of $\boldsymbol{w}^*$ is only due to the randomness of the initialization $\boldsymbol{w}$ and the data, while the k-LinReg algorithm implementing the mapping $\boldsymbol{w} \mapsto \boldsymbol{w}^*$ is deterministic.

The probability P_{fail} can also be defined as the expected value of the loss

$$L = \mathbb{I}(F(\boldsymbol{w}^*; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) - F(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) \geq \varepsilon),$$

i.e., $P_{fail}(n, p, N, \varepsilon) = \mathbb{E}L$, where $\mathbb{I}$ is the indicator function and $\mathbb{E}$ is the expectation over fixed and predefined distributions of the various random quantities $\boldsymbol{w}^*, \boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}$.

Since P_{fail} involves the solution $\boldsymbol{w}^*$ of the iterative Algorithm 1, its direct calculation is intractable and we instead rely on the estimate given by the empirical average as

$$P_{fail}^{emp}(n,p,N,\varepsilon) = \frac{1}{m}\sum_{k=1}^{m} L_k, \tag{6}$$

where the L_k are independent copies of L, i.e., $L_k = \mathbb{I}(F(\boldsymbol{w}^{k*}; \boldsymbol{X}^k, \boldsymbol{\lambda}^k, \boldsymbol{e}^k) - F(\boldsymbol{\theta}^k; \boldsymbol{X}^k, \boldsymbol{\lambda}^k, \boldsymbol{e}^k) \geq \varepsilon)$, where the superscript k denotes the kth copy of a random variable. The estimate of the probability of success is finally given by

$$P_{success}^{emp}(n,p,N,\varepsilon) = 1 - P_{fail}^{emp}(n,p,N,\varepsilon). \tag{7}$$

Thanks to the law of large numbers, we know that the empirical average, $P_{fail}^{emp}(n,p,N,\varepsilon)$, asymptotically converges (in m) towards the true probability $P_{fail}(n,p,N,\varepsilon)$. In addition, classical concentration inequalities provide a quantitative version of the law of large numbers, with finite-sample bounds on the error between the empirical average and the expected value. More precisely, Hoeffding inequality [17] yields

$$P^m\left\{\left|\frac{1}{m}\sum_{k=1}^{m} L_k - \mathbb{E}L\right| \geq t\right\} \leq 2\exp(-2mt^2),$$

where P^m stands for the product probability over an m-sample of iid. L_k. By setting $\delta = 2\exp(-2mt^2)$, we obtain the following error bound, which holds with probability at least $1-\delta$,

$$\left|P_{fail}^{emp}(n,p,N,\varepsilon) - P_{fail}(n,p,N,\varepsilon)\right| \leq \sqrt{\frac{\log\frac{2}{\delta}}{2m}}. \tag{8}$$

As an example, by choosing $\delta = 0.01$ and $m = 10000$ samples, the right-hand side of (8) is 0.0163, and there is a probability less than 1% of drawing 10000 samples which lead to an estimate $P_{fail}^{emp}(n,p,N,\varepsilon)$ not within $[P_{fail}(n,p,N,\varepsilon) - 0.0163, P_{fail}(n,p,N,\varepsilon) + 0.0163]$.

Remark 3. *The analysis above can be interpreted in the framework of probabilistic and randomized methods reviewed in [15]. In the terms defined in [15], the probability of failure is the* probability of violation *and the probability of success* (5) *is the* reliability *of the specification $F(\boldsymbol{w}^*; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) \leq F(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) + \varepsilon$, which is estimated by the* empirical reliability (7). *Under this framework, a* probabilistic design method *would search for a realization of the control variable, here, e.g., the initialization $\boldsymbol{w}$, leading to a predefined level of reliability. However, this approach is not applicable to our problem, since the randomness of the system embedded in $\boldsymbol{\theta}$ is non-informative. Indeed, the distribution of $\boldsymbol{\theta}$ does not represent some uncertainty interval around a mean value of the system parameters, but rather allows for a whole range of different systems to be identified, as discussed below.*

Choice of distribution. The bound (8) applies for any distribution of the random variables $\boldsymbol{w}, \boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}$. The only requirement is that the distribution remains fixed throughout the estimation procedure and that the samples are drawn independently and identically. Thus, when estimating the empirical mean (6), we have to choose a particular distribution to draw the samples from. In the absence of strong assumptions on the distribution of the random variables beyond simple limiting intervals, the uniform distribution is the choice typically considered in randomized methods [15]. Indeed, with few assumptions on the class of possible distributions and the probability of interest, this choice was rigorously justified in [18] in the following sense. The uniform distribution is a minimizer of the probability of interest over the whole class of distributions, and thus leads to worst-case estimates of the probability when used for sampling. Therefore, in all the following experiments, we sample the variables according to the distributions as detailed below.

In order to sample as much diverse switched regression problems as possible, while not favoring particular instances, the regression vectors, $\boldsymbol{x}_i$, and the parameter vectors, $\boldsymbol{\theta}_j$, $j = 1, \ldots, n$, are uniformly distributed in $[-5, 5]^p$. The data sets on which Algorithm 1 is applied are further generated with a uniformly distributed

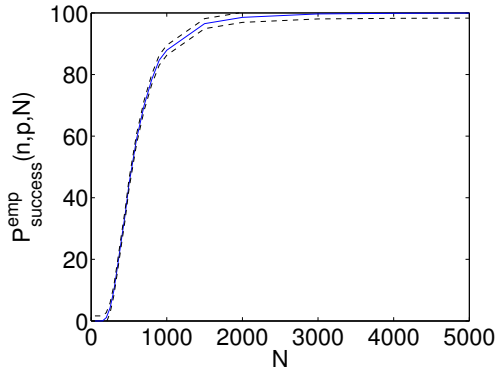

Figure 1: Probability of success (in %) estimated for $n = 4$ and $p = 10$ by (7) over $m = 10000$ samples (plain line) with the error bounds given by (8) at $\delta = 1\%$ (dashed lines).

mode λ_i in $\{1, \ldots, n\}$ and with a zero-mean Gaussian noise e_i of standard deviation $\sigma_e = 0.2$. The initialization $\boldsymbol{w}$ of Algorithm 1 follows a uniform distribution in the rather large interval $[-100, 100]^{np}$, assuming no prior knowledge on the true parameter values.

Section 4.1 below analyzes the influence of the size of the data sets, N, on the probability of success under these conditions, while Section 4.2 further studies the validity of the results for other noise levels. Finally, Section 4.3 will emphasize the effect of the constraints on the regressors implied by a SARX system.

4.1. Influence of the number of data

Consider the following set of experiments which aim at highlighting the relationship between the number of data and the difficulty of the problem measured through the probability of success of the simple k-LinReg algorithm. For each setting, defined by the triplet of problem dimensions (n, p, N), we generate $m = 10000$ training sets of N points, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, with the equation $y_i = \boldsymbol{\theta}_{\lambda_i}^T \boldsymbol{x}_i + e_i$, $i = 1, \ldots, N$, where $\boldsymbol{\theta}$, $\boldsymbol{x}_i$, λ_i, e_i are realizations of the corresponding random variables drawn according to the distributions defined above.

For each data set, we apply Algorithm 1 with a random initialization $\boldsymbol{w}^k$ and compute the estimate (7) of the probability of success. More precisely, we set $\epsilon = 10^{-9}$, which means that an initialization must lead to a value of the cost function (1) very close to the global optimum in order to be considered as successful.

In fact, the analysis considers a slight modification to Algorithm 1, in which the returned value of the objective function, $F(\boldsymbol{w}^{k*})$, is arbitrarily set to $F(\boldsymbol{w}^{k*}) = +\infty$ for problematic cases discussed in Sect. 3 and where, at some iteration t and for some mode j, $|\{i \ : \ \lambda_i^t = j\}| < p$. More generally, note that, in $\mathbb{R}^p$, any set of $p - 1$ points can be fitted by a single linear model. Therefore, for $N \leq n(p - 1)$, there are trivial solutions that minimize the cost $F(\boldsymbol{w})$, but do not correspond to solutions of the modeling problem with n submodels. The modification to Algorithm 1 ensures that such cases are considered as failures throughout the analysis.

Figure 1 shows a typical curve of the influence of the number of data on the quality of the solution returned by k-LinReg in terms of the quantity (7). The probabilistic bounds within which the true probability of success lies are also plotted and we see that they are very tight. The curves obtained in various settings are reported in Figure 2. These plots of (7) indicate that using more data increases the probability of success, so that the cost function (1) seems to have fewer local minima for large N than for small N.

Remark 4. *Most approaches that aim at globally solving Problem 1 have difficulties to handle large data sets. This is often related to an increase of the number of optimization variables, which may lead to a prohibitive computational burden. However, this does not necessarily means that the corresponding nonconvex optimization problem is more difficult to solve in terms of the number of local minima or the relative size of their basin of attraction compared with the one of the global optimum. Indeed, the results above provide evidence in favor of the opposite statement: the probability of success of the local optimization Algorithm 1 monotonically increases with the number of data.*

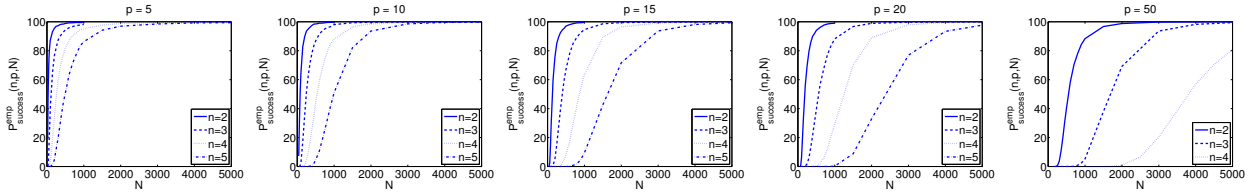

Figure 2: Estimates of the probability of success, $P^{emp}_{success}(n, p, N)$, versus the number of data N. Each plot corresponds to a different dimension p and each curve corresponds to a different number of modes n.

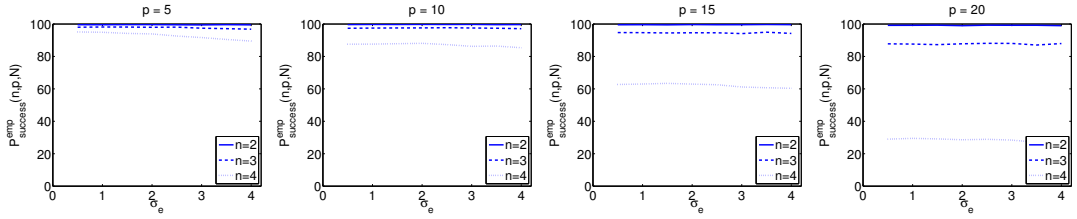

Figure 3: Estimates of the probability of success (in %) versus the noise level σ_e for different settings (n, p).

4.2. Influence of the noise level

The analysis in the previous section holds for fixed distributions of the random variables $\boldsymbol{w}, \boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}$, and in particular for the chosen noise level implemented by the standard deviation, σ_e, of the Gaussian distribution of $\boldsymbol{e}$. Here, we evaluate the influence of this noise level on the estimates of the probability of success.

Experiments are conducted as in the previous subsection, except for the training set size, N, which is fixed to 1000 and the noise level, σ_e, which now varies over a grid of the interval $[0.5, 4.0]$ (corresponding to a range of about 20 dB in terms of signal-to-noise ratio and reasonable values within which the submodels can be distinguished). This means that for each value of σ_e on this grid, 10000 samples of the random variables are drawn to compute the estimate of the probability of success via the application of Algorithm 1 and (6)–(7). These experiments are repeated for different problem dimensions (n, p) and the results are plotted in Figure 3. Note that, here, the aim is not to evaluate the range of noise levels that the k-LinReg algorithm can deal with, but instead to determine whether the curves of Figure 2 are sensitive to the noise level.

These results show that the noise level has little influence on the performance of the algorithm. Recall that here, the performance of the algorithm is evaluated as its ability to reach a value of the cost function sufficiently close to the global minimum. Of course, once a solution close to the global one has been found, the noise level influences the accuracy of the parameter estimation, as for any regression algorithm. However, these experiments support the idea that the noise level does not directly determines the level of difficulty of a switched regression problem, which rather lies in the ability of an algorithm to discriminate between the modes. It will therefore be omitted from the predictive model discussed in Section 5 which will focus on the influential parameters n and p.

4.3. Influence of the SARX system structure

This section reproduces the analysis of Sect. 4.1 for SARX system identification instead of switched regression, i.e., for a regression vector $\boldsymbol{x}_i = [y_{i-1}, \ldots, y_{i-n_y}, u_{i-n_a}, \ldots, u_{i-n_b}]^T$, constrained to a manifold of $\mathbb{R}^p$. More precisely, $\boldsymbol{X}$ is now a deterministic function of the random variables $\boldsymbol{\theta}$, $\boldsymbol{x}_0$ (the initial conditions), $\boldsymbol{u}$ (the input sequence), $\boldsymbol{\lambda}$ and $\boldsymbol{e}$. The probability is now defined over $\boldsymbol{x}_0, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{e}$ instead of $\boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}$ and with uniform distributions for $\boldsymbol{x}_0$ and $\boldsymbol{u}$. In addition, for a given p, we uniformly draw n_y in $\{1, \ldots, p-1\}$ and set $n_a = 0$, $n_b = p - n_y$. Thus, we uniformly sample identification problems with various system orders. We use a simple rejection method to discard unbounded trajectories without particular assumptions on the

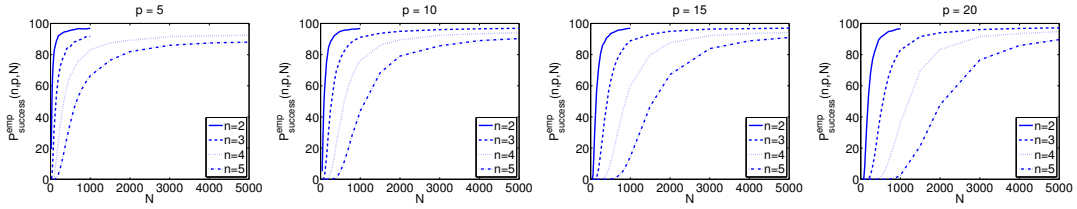

Figure 4: Same curves as in Figure 2, but for SARX system identification.

system whose parameters $\boldsymbol{\theta}$ remain uniformly distributed in $[-1,1]^{np}$ (depending on the switching sequence, unstable subsystems can lead to stable trajectories and vice versa).

The results are plotted in Fig. 4 and confirm the general tendency of the probability of success with respect to N. However, these results also show that the constraints on the regression vector increase the difficulty of the problem. In particular, both the rate of increase of the probability of success and its maximal value obtained for large N are smaller than in the unconstrained case reported in Fig. 2.

5. Modeling the probability of success

The following presents the estimation of a model of the probability of success based on the measurements of this quantity derived in the previous section. The aim of this model is to provide a simple means to set the number of restarts for k-LinReg (Section 5.2) and an indication on the number of data with which we can expect k-LinReg to be successful on a particular task (Section 5.3).

More precisely, we are interested in obtaining these numbers from general parameters of a given problem instance summarized by the triplet (n, p, N). In order to fulfill this goal, we require an estimate of the probability of success of k-LinReg for any given triplet of problem dimensions (n, p, N). Since estimating this probability as in Section 4 for all possible triplets would clearly be intractable, we resort to a modeling approach. Thus, the aim is to estimate a model that can predict with sufficient accuracy the probability $P_{success}(n,p,N,\varepsilon)$. Irrespective of the quantity of interest, the classical approach to modeling involves several steps. First, we need to choose a particular model structure. This structure can either be parametric and based on general assumptions or nonparametric as in black-box models. Here, we consider the parametric approach and choose a particular model structure from a set of assumptions. Then, the parameters of this model are estimated from data. Here, the data consist in the estimates of the probability of success obtained in Section 4 for a finite set of triplets (n,p,N). Therefore, in the following, these estimates, $P^{emp}_{success}(n,p,N,\varepsilon)$, are interpreted as noisy measurements of the quantity of interest, $P_{success}$, from which the model can be determined.

This section and the followings are based on the measurements obtained in Sect. 4 with a fixed and small value of the threshold $\varepsilon = 10^{-9}$. Therefore, in the remaining of the paper we will simply use the notation $P_{success}(n,p,N)$ to refer to $P_{success}(n,p,N,\varepsilon)$ for this specific value of ε.

The choice of structure for the proposed model relies on the following assumptions.

- The probability of success should be zero when N is too small. Obviously, without sufficient data, accurate estimation of the parameters $\boldsymbol{w}$ is hopeless. In particular, we aim at the estimation of all parameter vectors, $\{\boldsymbol{w}_j\}_{j=1}^n$, and therefore consider that solutions leading to a low error with fewer submodels as inaccurate. More precisely, this is implemented in the data by the modification of Algorithm 1 discussed in Sect. 3 and which returns an infinite error value in such cases.

- The probability of success monotonically increases with N. This assumption is clearly supported by the observations of Section 4. It is also in line with common results in learning and estimation theory, where better estimates are obtained with more data. However, the validity of such assumptions in the context of switched regression was not obvious, since these results usually apply to local algorithms under the condition that the initialization is in the vicinity of the global minimizer (or that the problem is convex).

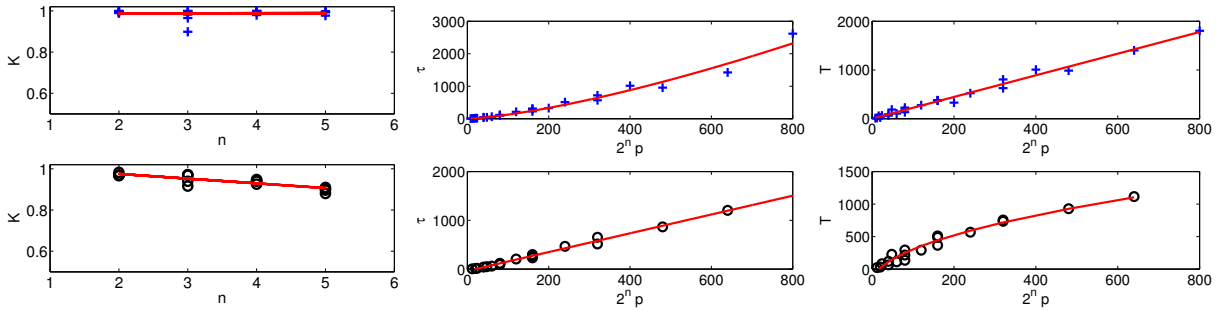

Figure 5: Parameters of the model (9) as obtained from the average curves in Fig. 2 (blue '+'), the ones in Fig. 4 (black 'o') and from the proposed formulas (13)–(18) in a switched regression (top row) or SARX system identification (bottom row) setting.

- The probability of success converges, when N goes to infinity, to a value smaller than one. This assumption reflects the fact that the size of the data set cannot alleviate all difficulties, in particular regarding the issue of local minima. Note that this is an issue specific to the switched regression setting, whereas classical regression algorithms can compute consistent estimates. In switched regression, consistency of an estimator formulated as the solution to a nonconvex optimization problem does not imply the computability of consistent estimates.
- The probability of success should be a decreasing function of both n and p. This assumption reflects the increasing difficulty of the problem with respect to its dimensions. In particular, with more submodels (larger n), more local minima can be generated, while increasing the dimension p increases the range of possible initializations and thus decreases the probability of drawing an initialization in the vicinity of the global optimum.

Following these assumptions, we model the probability of success, $P_{success}(n,p,N)$, by the unitary step response of a first order dynamical system with delay, where the number of data N plays the role of the time variable, i.e.,

$$\widehat{P}_{success}(n,p,N) = \begin{cases} K(n,p)\left(1-\exp\left(\dfrac{-(N-\tau(n,p))}{T(n,p)}\right)\right), & \text{if } N > \tau(n,p) \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

and where the dimensions n and p influence the parameters. In particular, $K(n,p)$ is the "gain", $\tau(n,p)$ is the "time-delay" and $T(n,p)$ is the "time-constant". Such models are used in Broïda's identification method [19], in which the constants K, τ and T are estimated on the basis of curves as the ones in Figures 2 and 4. This estimation relies on the following formulas:

$$K(n,p) = \lim_{N\to+\infty} P_{success}(n,p,N) = \sup_N \ P_{success}(n,p,N), \tag{10}$$

$$\tau(n,p) = 2.8N_1 - 1.8N_2, \tag{11}$$

$$T(n,p) = 5.5(N_2 - N_1), \tag{12}$$

where N_1 and N_2 are the numbers of data (originally, the "times") at $P_{success}(n,p,N) = 0.28K(n,p)$ and $P_{success}(n,p,N) = 0.40K(n,p)$, respectively. We use linear interpolation to determine N_1 and N_2 more precisely from the curves in Fig. 2 and 4 and obtain estimates of the constants K, τ and T for different numbers of modes n and dimensions p.

As shown by Figure 5, these constants can be approximated by the following functions of n and p:

$$K(n,p) = 0.99, \tag{13}$$

$$\tau(n,p) = 0.2\left(2^n p\right)^{1.4}, \tag{14}$$

$$T(n,p) = 2.22 \times 2^n p, \tag{15}$$

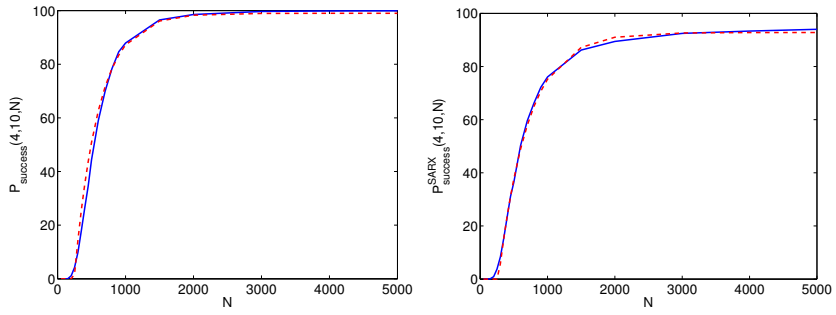

Figure 6: Empirical estimates of the probability of success (solid line) and values predicted by the model (7) (dash line) for switched regression (left) and SARX system identification (right). These curves are obtained with $n = 4$ and $p = 10$.

for switched regression, while the following approximations are suitable for SARX system identification:

$$K^{SARX}(n,p) = 1.02 - 0.023n, \tag{16}$$

$$\tau^{SARX}(n,p) = 1.93 \times 2^n p - 37, \tag{17}$$

$$T^{SARX}(n,p) = 52\sqrt{2^n p} - 220. \tag{18}$$

The coefficients in these equations are the least squares estimates of the generic parameters a and b in the linear regressions $K(n,p) = an + b$, $\tilde{\tau} = a\tilde{z} + b$ and $T(n,p) = az + b$, where $z = 2^n p$, $\tilde{\tau} = \log \tau(n,p)$ and $\tilde{z} = \log z$. For SARX systems, the relationships giving τ^{SARX} and T^{SARX} are modified as $\tau^{SARX}(n,p) = az + b$ and $T^{SARX}(n,p) = a\sqrt{z} + b$.

Finally, the model of the probability of success is given either by

$$\widehat{P}_{success}(n,p,N) = \begin{cases} 0.99\left(1 - \exp\left(\frac{-\left(N - 0.2(2^n p)^{1.4}\right)}{2.22 \times 2^n p}\right)\right), & \text{if } N > 0.2\,(2^n p)^{1.4} \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

in a regression setting or by,

$$\widehat{P}^{SARX}_{success}(n,p,N) = \begin{cases} (1.02 - 0.023n)\left(1 - \exp\left(\frac{-(N - 1.93 \times 2^n p - 37)}{52\sqrt{2^n p} - 220}\right)\right), & \text{if } N > 1.93 \times 2^n p - 37 \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

for SARX system identification. The output of these models is very close to the empirical average of $P_{success}(n,p,N)$ as illustrated for $n = 4$ and $p = 10$ by Figure 6. The mean absolute error, $1/|\mathcal{T}| \sum_{(n,p,N)\in\mathcal{T}} |P_{success}(n,p,N) - \widehat{P}_{success}(n,p,N)|$, over the set of triplets, $\mathcal{T}$, used for their estimation, is 0.048 for $\widehat{P}_{success}(n,p,N)$ and 0.102 for $\widehat{P}^{SARX}_{success}(n,p,N)$.

5.1. Estimation of the generalization ability of the model via cross-validation

The generalization error of the models (19)-(20) reflects their ability to accurately predict the probability of success for triplets (n,p,N) not in $\mathcal{T}$, i.e., for previously unseen test cases. This generalization error can be estimated from the data through a classical cross-validation procedure (see, e.g., [20]), which iterates over K_{CV} subsets of the data. At each iteration, the model is estimated from $K_{CV} - 1$ subsets and tested on the remaining subset. The average error thus obtained yields the cross-validation estimate of the generalization error.

Since the computation of each value of the constants K, τ, T (each point in Fig. 5) rely on an entire curve with respect to N, we split the data with respect to n and p: all data in a subset correspond to a curve in Fig. 2 or 4 for fixed (n,p). Then, at iteration k, the models (19)-(20) are estimated without the data for the

setting (n_k, p_k) and the sum of absolute errors, $|P_{success}(n_k, p_k, N) - \widehat{P}_{success}(n_k, p_k, N)|$, is computed over all N.

Applying this procedure leads to a cross-validation estimate of the mean absolute error of 0.050 for switched regression and 0.105 for SARX system identification. Though the model is less accurate for SARX systems, its error remains reasonable for practical purposes such as the automatic tuning of the number of restarts for k-LinReg, as proposed next.

5.2. Computing the number of restarts for k-LinReg

Consider now the algorithm restarted r times with initializations that are independently and identically drawn from a uniform distribution. The probability of drawing a successful initialization for a particular problem instance, i.e., for given $\boldsymbol{\theta}$, $\boldsymbol{X}$, $\boldsymbol{\lambda}$ and $\boldsymbol{e}$, is the conditional probability of success $P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}) = P_{\boldsymbol{w}|\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}(F(\boldsymbol{w}^*) - F(\boldsymbol{\theta}) \leq 10^{-9})$ and the probability of not drawing a successful initialization in any of the restarts is

$$P_{fail}(r) = \prod_{k=1}^{r} (1 - P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e})) = (1 - P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}))^r. \tag{21}$$

To set the number of restarts, we consider the average probability of drawing a successful initialization, where the average is taken over all problem instances, i.e., we consider the expected value of the conditional probability of success $P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e})$. This expectation corresponds to the joint probability of success, $P_{success}(n, p, N) = P_{\boldsymbol{w},\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}(F(\boldsymbol{w}^*) - F(\boldsymbol{\theta}) \leq 10^{-9})$, since

$$\begin{aligned} P_{success}(n, p, N) &= \mathbb{E}_{\boldsymbol{w},\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}[\mathbb{I}(F(\boldsymbol{w}^*) - F(\boldsymbol{\theta}) \leq 10^{-9})] \\ &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}} \mathbb{E}_{\boldsymbol{w}|\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}[\mathbb{I}(F(\boldsymbol{w}^*) - F(\boldsymbol{\theta}) \leq 10^{-9}) | \boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e}] \\ &= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{\lambda},\boldsymbol{e}}[P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e})]. \end{aligned}$$

Considering the mean conditional probability, i.e., replacing $P_{cond}(\boldsymbol{\theta}, \boldsymbol{X}, \boldsymbol{\lambda}, \boldsymbol{e})$ by $P_{success}(n, p, N)$ in (21), leads to a trade-off between optimistic estimates for difficult problems and pessimistic ones for easier problems. This trade-off allows us to build an algorithm that is both computationally efficient and sufficiently successful on average. This algorithm relies on the following estimate of the probability of failure:

$$\widehat{P}_{fail}(r) = (1 - \widehat{P}_{success}(n, p, N))^r.$$

Then, the number of restarts r^* required to obtain a given maximal probability of failure P_f^* such that $\widehat{P}_{fail}(r) \leq P_f^*$, is computed as

$$r^* = \min_{r \in \mathbb{N}^*} r, \quad \text{s.t.} \quad r \geq \frac{\log P_f^*}{\log\left(1 - \widehat{P}_{success}(n, p, N)\right)}, \tag{22}$$

where $\widehat{P}_{success}(n, p, N)$ is given by (19) or (20).

In practice, the bound in (22) can be used as a stopping criterion on the restarts of the algorithm once the value of the hyperparameter P_f^* has been set. This leads to Algorithm 2, which can easily be cast into a parallel form by executing the line in italic multiple times in separate working threads.

The internal box bounds on the initialization vectors, $[\underline{\boldsymbol{w}}, \overline{\boldsymbol{w}}]$, can be used to include prior knowledge on the values of the parameters, but they can also be set to quite large intervals in practice. Note that these bounds do not constrain the solution of the algorithm beyond the initialization, so that $\boldsymbol{w}_{best} \notin [\underline{\boldsymbol{w}}, \overline{\boldsymbol{w}}]$ can be observed.

5.3. Estimating the sample complexity

We define the sample complexity of k-LinReg at level γ as the minimal size of the training sample, $N(\gamma)$, leading to

$$P_{success}(n, p, N) \geq \gamma.$$

Algorithm 2 k-LinReg with multiple restarts

Require: the data set $(\boldsymbol{X}, \boldsymbol{y}) \in \mathbb{R}^{N\times p} \times \mathbb{R}^N$, the number of modes n and the probability of failure P_f^*.
  Initialize $F_{best} = +\infty$ and $k = 0$.
  Compute K, τ, T as in (13)-(15) or (16)-(18).
  Set r^* according to (22).
  repeat
    $k \leftarrow k + 1$.
    Draw a random vector $\boldsymbol{w}$ uniformly in $[\underline{\boldsymbol{w}}, \overline{\boldsymbol{w}}] \subset \mathbb{R}^{np}$.
    Apply Algorithm 1 initialized at $\boldsymbol{w}$ *to get the pair* $(F(\boldsymbol{w}^*), \boldsymbol{w}^*)$.
    if $F(\boldsymbol{w}^*) < F_{best}$ **then**
      Update the best local minimum $F_{best} \leftarrow F(\boldsymbol{w}^*)$ and minimizer $\boldsymbol{w}_{best} \leftarrow \boldsymbol{w}^*$.
    end if
  until $k \geq r^*$.
  return F_{best} and $\boldsymbol{w}_{best}$.

After substituting the model $\widehat{P}_{success}(n, p, N)$ for $P_{success}(n, p, N)$ in the above, this sample complexity can be estimated by using (9) as

$$\begin{cases} \widehat{N}(\gamma) \geq \tau(n,p) - T(n,p) \log\left(1 - \dfrac{\gamma}{K(n,p)}\right), & \text{if } \gamma < K(n,p), \\ \widehat{N}(\gamma) = +\infty, & \text{otherwise.} \end{cases} \tag{23}$$

In a given setting with n and p fixed, this can provide an indicative lower bound on the number of data required to solve the problem with probability γ by a single run of k-LinReg. The case $\widehat{N}(\gamma) = +\infty$ reflects the impossibility to reach the probability γ, which may occur for unreasonable values of $\gamma > 0.99$ or for SARX systems with a very large number of modes n.

However, note that (23) is mostly of practical use, since its analytical form depends on the particular choice of structure for (9).

6. Numerical experiments

This section is dedicated to the assessment of the model of the probability of success (19) as a means to tune the number of restarts on one hand and of the k-LinReg algorithm as an efficient method for switched linear regression on the other hand. Its time efficiency is studied in Sec. 6.1 and 6.2 with respect to the number of data and the problem dimensions, respectively. Section 6.3 analyses the ability of the method to reach global optimality, while Sec. 6.4 is dedicated to its application to hybrid dynamical systems.

The k-LinReg method is compared with the approach of [10], which directly attempts to globally optimize (1) with the Multilevel Coordinate Search (MCS) algorithm [21] and which we label ME-MCS. The second version of this approach, based on a smooth product of error terms, is also included in the comparison and labeled PE-MCS. We also consider the algebraic approach [2], which can deal efficiently with noiseless data, and the recent sparse optimization-based method [11]. The application of the latter is slightly different as it requires to set a threshold on the modeling error instead of the number of modes. The comparison with this approach will therefore only be conducted on the example taken from [11].

Three evaluation criteria are considered: the computing time, the mean squared error (MSE) computed by the cost function (1), and the normalized mean squared error on the parameters (NMSE) evaluated against the true parameter values $\{\boldsymbol{\theta}_j\}$ as

$$\text{NMSE} = \frac{1}{n} \sum_{j=1}^{n} \frac{\|\boldsymbol{\theta}_j - \boldsymbol{w}_j\|_2^2}{\|\boldsymbol{\theta}_j\|_2^2}, \tag{24}$$

for the set of estimated parameter vectors $\{\boldsymbol{w}_j\}$ ordered such that the NMSE is minimum. All numbers reported in the Tables and points in the plots are averages computed over 100 random problems generated

with a different set of parameters $\{\boldsymbol{\theta}_j\}$, except for the examples in Sect. 6.4 where the parameters are fixed and the averages are taken over 100 trials with different noise sequences.

For indicative purposes, a *reference model* is also included in the experiments and corresponds to the model obtained by applying n independent least-squares estimators to the data classified on the basis of the true mode (which is unknown to the other methods).

Except when mentioned otherwise, k-LinReg refers to Algorithm 2, which is stopped after completing the number of restarts r^* given by (22) with a maximal probability of failure $P_f^* = 0.1\%$ and with the model (19), or (20) for the examples of Sect. 6.4. The bounds on the initialization are always set to $\underline{\boldsymbol{w}} = -100 \cdot \mathbf{1}_{np}$ and $\overline{\boldsymbol{w}} = 100 \cdot \mathbf{1}_{np}$. The web page at `http://www.loria.fr/~lauer/klinreg/` provides open source code for k-LinReg in addition to all the scripts used in the following experiments.

6.1. Large data sets in low dimension

The performance of the methods on large data sets is evaluated through a set of low-dimensional problems with $n = 2$ and $p = 3$, in a similar setting as in [10]. The N data are generated by $y_i = \boldsymbol{\theta}_{\lambda_i}^T \boldsymbol{x}_i + v_i$, with uniformly distributed random regression vectors $\boldsymbol{x}_i \in [-5, 5]^p$, a random switching sequence $\{\lambda_i\}$ and additive zero-mean Gaussian noise v_i of standard deviation $\sigma_v = 0.2$. The goal is to recover the set of true parameters $\{\boldsymbol{\theta}_j\}$ randomly drawn from a uniform distribution in the interval $[-2, 2]^p$ in each experiment.

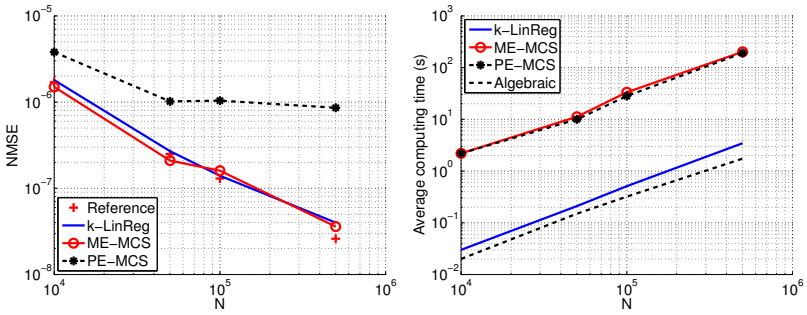

Figure 7: Average NMSE (left) and computing time (right) over 100 trials versus the number of data N (plots in log-log scale).

Figure 7 shows the resulting NMSE and computing times of k-LinReg, ME-MCS and PE-MCS, as averaged over 100 experiments for different numbers of data N. The computing time of the algebraic method, considered as the fastest method for such problems, is also shown for indicative purposes though this method cannot handle such a noise level (times are obtained by applying it to noiseless data). All computing times are given with respect to Matlab implementations of the methods running on a laptop with 2 CPU cores at 2.4 Ghz. The k-LinReg algorithm is much faster than the general purpose MCS algorithm and almost as fast as the algebraic method, while still leading to very accurate estimates of the parameters. In particular, the average NMSE of k-LinReg is similar to the ones obtained by the reference model and the ME-MCS. However, the ME-MCS algorithm fails to find a relevant model (with NMSE < 1) in a number of cases which are not taken into account in its average NMSE. Note that the k-LinReg estimates are found with only two random initializations since (22) leads to $r^* = 2$ in this setting.

6.2. High dimensions

Figure 8 shows the average computing times over 100 random experiments of the k-LinReg, the PE-MCS and the algebraic methods versus the dimension p. Data sets of $N = 10\,000$ points are generated as in the previous subsection with a noise standard deviation of $\sigma_v = 0.1$, except for the algebraic method which is applied to noiseless data in order to produce consistent results. The reported numbers reflect the computing times of Matlab implementations of the three methods as obtained on a computer with 8 cores running at 2.53 GHz. For low dimensions, i.e., $p \leq 5$, the k-LinReg algorithm is slightly slower than the algebraic method, with however computing times that remain below 1 second. For high dimensions, $p \geq 50$, the

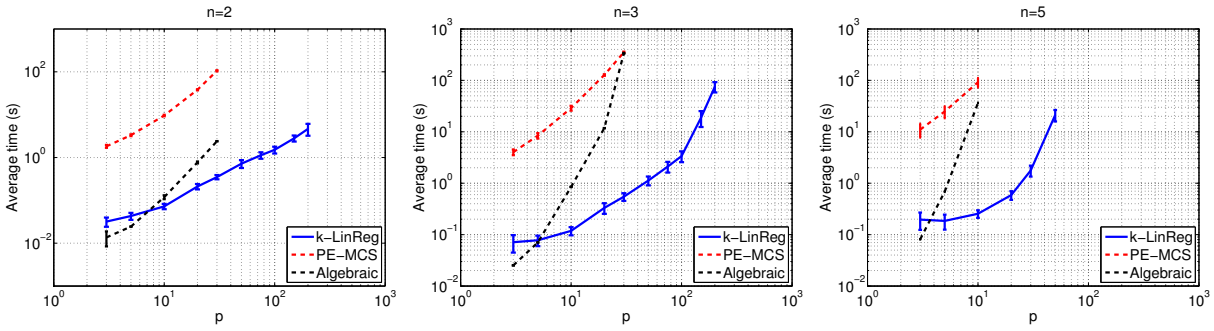

Figure 8: Average computing time (over 100 trials) of the methods versus the dimension p for different numbers of modes n and $N = 10000$ data points (log-log scale). The Figure is best viewed in color.

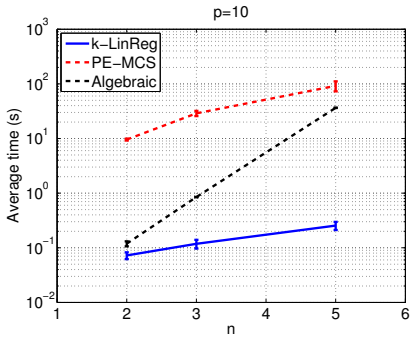

Figure 9: Average computing time (over 100 trials) of the methods versus the number of modes n for $p = 10$ and $N = 10000$ (log scale). The Figure is best viewed in color.

PE-MCS and the algebraic methods cannot yield a satisfactory result in a reasonable time (or run out of memory for the latter) even for the smallest number of modes $n = 2$. On the other hand, the k-LinReg algorithm remains efficient and can handle a dimension of $p = 200$ in few seconds for $n = 2$, despite an exponential complexity with respect to p. The knees in the curves of the k-LinReg computing time, observed for instance for $n = 3$ at $p = 100$, reflect the increase in the number of restarts given by (22). Figure 9 shows that all methods also have an exponential computing time with respect to the number of modes n, and that the algebraic approach suffers from the largest rate of increase.

6.3. Global optimality

For noiseless data, the global optimum of the cost function (1) is known to be zero. In such a setting, we can measure the running time and the number of restarts required by the algorithm to reach the global optimum. Table 1 shows the results of such experiments for problems of different sizes. Algorithm 1 is restarted until the criterion $F(\boldsymbol{w}^*) < 10^{-6}$ is satisfied or the number of restarts exceeds 100, in which case this is interpreted as a failure. The experiments are repeated for 100 random problems for all sizes and if a failure is detected, only lower bounds on the number of restarts and the computing time are given in Table 1 (even if the global optimum was found in another experiment of the same set). The results indicate that the k-LinReg algorithm always finds the global optimum in less than 100 restarts (often just one) and few seconds, except when the numbers n and p lead to $\tau(n, p) > N$, in which case the model (19) predicts the failure. Table 1 additionally shows how the dimension p can be increased without affecting the performance of k-LinReg when the number of data N is also sufficiently increased. Though the algebraic approach [2] also leads to the global optimum in noiseless cases, it is not suitable for such high-dimensional problems, as previously emphasized by Fig. 8.

Table 1: Number of restarts and computing times (on a laptop with 2 CPUs at 2.4 GHz) required to reach global optimality. Numbers are given in the format *average* ± *standard deviation* ≤ *maximum*, or > *lower bound* for failures.

$\mathbf{N = 1\,000}$						$\mathbf{N = 10\,000}$					
n	p	τ	# restarts	Time (s)		n	p	τ	# restarts	Time (s)	
2	50	333	$1 \pm 1 \leq 3$	$0.1 \pm 0.1 < 0.5$		2	100	879	$1 \pm 0 \leq 1$	$4.2 \pm 0.7 < 7.8$	
	100	879	$3 \pm 3 \leq 17$	$1.4 \pm 1.0 < 5.6$			500	8365	$1 \pm 1 \leq 3$	$70 \pm 43 < 269$	
	200	*2320*	> 100	> 98			1000	*22076*	> 100	> 10800	
3	10	92	$1 \pm 1 \leq 2$	$0.0 \pm 0.0 < 0.1$		3	100	2320	$1 \pm 0 \leq 1$	$6.8 \pm 1.9 < 16.4$	
	30	430	$1 \pm 1 \leq 5$	$0.1 \pm 0.1 < 0.5$			200	6121	$1 \pm 1 \leq 6$	$53 \pm 45 < 290$	
	50	879	> 100	> 11			400	*16153*	> 100	> 5712	
4	10	244	$1 \pm 1 \leq 2$	$0.0 \pm 0.0 < 0.1$		4	50	2320	$1 \pm 0 \leq 1$	$1.7 \pm 0.8 < 4.8$	
	20	643	$3 \pm 3 \leq 19$	$0.3 \pm 0.2 < 1.8$			100	6121	$1 \pm 1 \leq 4$	$24 \pm 16 < 94$	
	30	*1134*	> 100	> 12			200	*16153*	> 100	> 2748	
5	10	643	$2 \pm 2 \leq 10$	$0.1 \pm 0.1 < 0.4$		5	10	643	$1 \pm 0 \leq 1$	$0.1 \pm 0.1 < 0.4$	
	20	*1697*	> 100	> 11			50	6121	$1 \pm 1 \leq 3$	$4.7 \pm 2.8 < 15.1$	
							100	*16153*	> 100	> 1506	

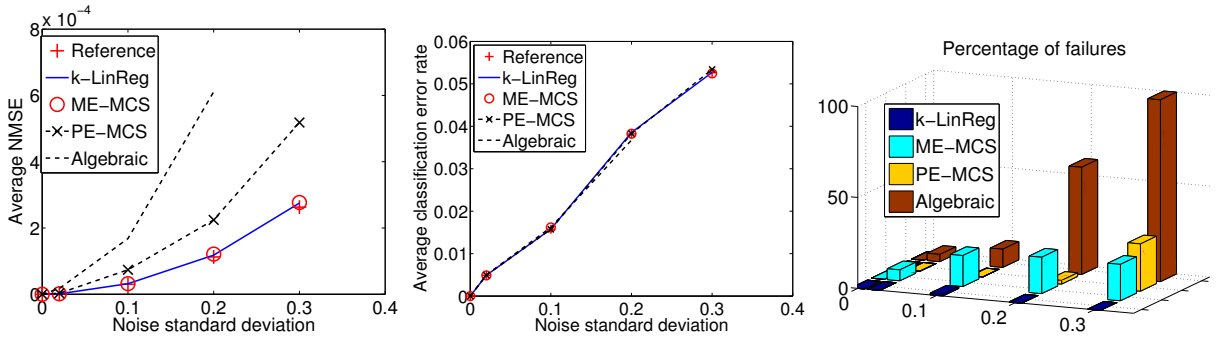

Figure 10: Influence of the noise level on the average NMSE and classification error rate (the standard deviations follow similar patterns) and on the percentage of failures of the methods. The average NMSEs for k-LinReg, ME-MCS and the reference model are hardly distinguishable and so are the classification error rates for all methods.

6.4. Application to hybrid dynamical system identification

The following studies two hybrid system identification examples from the recent literature and will show that these benchmark problems can be solved with few restarts of k-LinReg.

6.4.1. Robustness to noise

Consider the example taken from [7] and also considered in [10]. The aim is to recover, from $N = 1000$ samples, the parameters $\boldsymbol{\theta}_1 = [0.9, 1]^T$ and $\boldsymbol{\theta}_2 = [1, -1]^T$ of a dynamical system, arbitrarily switching between $n = 2$ modes, with $\boldsymbol{x}_i = [y_{i-1}, u_{i-1}]^T$ and a zero-mean Gaussian input with unit variance u_i. Experiments with additive and centered Gaussian noise are conducted for different values of the noise standard deviation ranging from 0 to 0.3, whereas the standard deviation of the output is $\sigma_y \approx 2$.

Figure 10 shows the parametric error (NMSE) and the classification error rate of the k-LinReg, the ME-MCS, the PE-MCS and the algebraic methods as a function of the noise level and averaged over 100 experiments. All methods recover the parameters without error from the noiseless data. But only the k-LinReg and the ME-MCS methods achieve errors similar to the ones of the reference model for all noise levels. In addition, Figure 10 shows that the ME-MCS method failed in about 15% of the experiments. These failures represent the cases for which the NMSE is larger than 10^{-3} and which are not taken into account in the averaged NMSE and classification error rate. The PE-MCS method benefits from much less failures, but leads to a larger average NMSE. Finally, the error of the algebraic method quickly increases with

Table 2: Average parameter estimates over 100 trials for the example taken from [11] and discussed in Sect. 6.4.2.

$\boldsymbol{\theta}_1$	-0.4	0.25	-0.15	0.08
Reference	-0.3998 ± 0.0051	0.2499 ± 0.0051	-0.1516 ± 0.0083	0.0806 ± 0.0095
k-LinReg ($r = 5$)	-0.3999 ± 0.0053	0.2499 ± 0.0053	-0.1514 ± 0.0085	0.0801 ± 0.0099
k-LinReg ($r^* = 3$)	-0.3964 ± 0.0405	0.2448 ± 0.0537	-0.1504 ± 0.0105	0.0860 ± 0.0689
Sparse optim. [11]	-0.3914 ± 0.0115	0.2452 ± 0.0106	-0.1666 ± 0.0201	0.0875 ± 0.0200
$\boldsymbol{\theta}_2$	1.55	-0.58	-2.1	0.96
Reference	1.5495 ± 0.0048	-0.5751 ± 0.0047	-2.1013 ± 0.0085	0.9594 ± 0.0114
k-LinReg ($r = 5$)	1.5428 ± 0.0667	-0.5751 ± 0.0437	-2.1017 ± 0.0096	0.9551 ± 0.0443
k-LinReg ($r^* = 3$)	1.5328 ± 0.0740	-0.5626 ± 0.0793	-2.0886 ± 0.1127	0.9422 ± 0.1069
Sparse optim. [11]	1.5360 ± 0.0549	-0.5706 ± 0.0337	-2.0680 ± 0.1421	0.9434 ± 0.0728
$\boldsymbol{\theta}_3$	1.0	-0.24	-0.65	0.30
Reference	1.0002 ± 0.0043	-0.2407 ± 0.0042	-0.6502 ± 0.0074	0.2989 ± 0.0085
k-LinReg ($r = 5$)	1.0033 ± 0.0296	-0.2439 ± 0.0310	-0.6504 ± 0.0100	0.3016 ± 0.0283
k-LinReg ($r^* = 3$)	1.0111 ± 0.0677	-0.2535 ± 0.0736	-0.6573 ± 0.0434	0.3226 ± 0.1323
Sparse optim. [11]	0.9909 ± 0.0128	-0.2365 ± 0.0124	-0.6727 ± 0.0263	0.3102 ± 0.0271

the noise level, which leads to many failures in the high-noise regime. Note that, with a single initialization ($r^* = 1$), the k-LinReg algorithm showed only one failure over the hundreds of trials of these experiments.

Regarding the classification errors, all methods lead to similar rates, including the reference model, which shows that these errors cannot be avoided. As discussed in Remark 1, this is due to data points that are consistent with multiple submodels and which are in a number increasing with the noise level. However, these misclassifications have a limited influence on the estimation of the parameters by k-LinReg as they correspond to small values of $(y_i - \boldsymbol{w}_j^T \boldsymbol{x}_i)^2$ in (4).

6.4.2. A slightly more difficult example

The next hybrid system identification example is taken from [11] with $n = 3$ modes and $N = 300$ data points in dimension $p = 4$. The signal-to-noise ratio (SNR) in this data is 30 dB. For these problem dimensions, the model (20) leads to set the number of restarts to $r^* = 3$ in accordance with (22).

We perform 100 random experiments as described in [11] and Table 2 shows the average value of the estimated parameters over these 100 trials. The estimates obtained by k-LinReg in less than 0.1 second are comparable with the results of the sparse optimization method reported in [11]. However, in order to cancel the difference between the k-LinReg average estimates and the ones of the reference model, the number of restarts needs to be slightly increased to $r = 5$. For few instances (out of 100 trials) of this particular example, the estimate r^* in (22) is too optimistic and the global solution is not found with r^* restarts.

7. Conclusions

We analyzed a k-means-like algorithm for switched linear regression and estimated a model of its expected performance. This model can be used in practice to set the main parameter of the algorithm, that is, the number of restarts or initializations. The resulting k-LinReg algorithm is very simple and can quickly identify switched linear systems from large data sets. In addition, experiments showed that the algorithm is able to yield a global solution when the number of data is sufficiently large, which corroborates the predictions of the model. This also indicates that switched linear regression problems with large data sets are not as difficult to solve as one would expect. In this respect, the k-LinReg algorithm and its model of performance offer a simple means to evaluate the difficulty of a particular problem.

While the paper focused on a simple model of the probability of success designed to provide the number of restarts, future work will further study the relationship between the sample size and the probability of success in order to produce more accurate models. Another issue concerns the determination of the validity range of the model: are predictions still accurate for much larger values of n or p? At the practical level,

various enhancements of the k-LinReg algorithm can be investigated, such as the strategy to adopt when too few data points are assigned to a mode and how this could be used to estimate the number of submodels when n is overestimated. Finally, application of k-LinReg to switched *nonlinear* system identification with kernel submodels as proposed in [22] could also be investigated.

Acknowledgements

The author thanks the anonymous reviewers whose comments and suggestions helped to improve the paper.

References

[1] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems: a tutorial, European Journal of Control 13 (2-3) (2007) 242–262.

[2] R. Vidal, S. Soatto, Y. Ma, S. Sastry, An algebraic geometric approach to the identification of a class of linear hybrid systems, in: Proc. of the 42nd IEEE Conf. on Decision and Control (CDC), Maui, Hawaï, USA, 2003, pp. 167–172.

[3] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, A clustering technique for the identification of piecewise affine systems, Automatica 39 (2) (2003) 205–217.

[4] J. Roll, A. Bemporad, L. Ljung, Identification of piecewise affine systems via mixed-integer programming, Automatica 40 (1) (2004) 37–50.

[5] A. L. Juloski, S. Weiland, W. Heemels, A Bayesian approach to identification of hybrid systems, IEEE Transactions on Automatic Control 50 (10) (2005) 1520–1533.

[6] A. Bemporad, A. Garulli, S. Paoletti, A. Vicino, A bounded-error approach to piecewise affine system identification, IEEE Transactions on Automatic Control 50 (10) (2005) 1567–1580.

[7] R. Vidal, Recursive identification of switched ARX systems, Automatica 44 (9) (2008) 2274–2287.

[8] Y. Ma, R. Vidal, Identification of deterministic switched ARX systems via identification of algebraic varieties, in: Proc. of the 8th Int. Conf. on Hybrid Systems: Computation and Control (HSCC), Zürich, Switzerland, Vol. 3414 of LNCS, 2005, pp. 449–465.

[9] N. Ozay, C. Lagoa, M. Sznaier, Robust identification of switched affine systems via moments-based convex optimization, in: Proc. of the 48th IEEE Conf. on Decision and Control (CDC), Shanghai, China, 2009, pp. 4686–4691.

[10] F. Lauer, G. Bloch, R. Vidal, A continuous optimization framework for hybrid system identification, Automatica 47 (3) (2011) 608–613.

[11] L. Bako, Identification of switched linear systems via sparse optimization, Automatica 47 (4) (2011) 668–677.

[12] N. Ozay, M. Sznaier, C. Lagoa, O. Camps, A sparsification approach to set membership identification of a class of affine hybrid systems, in: Proc. of the 47th IEEE Conf. on Decision and Control (CDC), Cancun, Mexico, 2008, pp. 123–130.

[13] N. Ozay, M. Sznaier, C. Lagoa, O. Camps, A sparsification approach to set membership identification of switched affine systems, IEEE Transactions on Automatic Control 57 (3) (2012) 634–648.

[14] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability, 1967, pp. 281–297.

[15] G. Calafiore, F. Dabbene, R. Tempo, Research on probabilistic methods for control system design, Automatica 47 (7) (2011) 1279–1293.

[16] L. Bako, K. Boukharouba, E. Duviella, S. Lecoeuche, A recursive identification algorithm for switched linear/affine models, Nonlinear Analysis: Hybrid Systems 5 (2) (2011) 242–253.

[17] W. Hoeffding, Probability inequalities for sums of bounded random variables, Journal of the American Statistical Association (1963) 13–30.

[18] B. Barmish, C. Lagoa, The uniform distribution: A rigorous justification for its use in robustness analysis, Mathematics of Control, Signals, and Systems 10 (3) (1997) 203–222.

[19] P. Borne, G. Dauphin-Tanguy, J.-P. Richard, F. Rotella, I. Zambettakis, Analyse et régulation des processus industriels: Régulation continue (*in french*), Vol. 1, Editions Technip, Paris, France, 1993.

[20] M. Stone, Cross-validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society. Series B (Methodological) 36 (2) (1974) 111–147.

[21] W. Huyer, A. Neumaier, Global optimization by multilevel coordinate search, Journal of Global Optimization 14 (4) (1999) 331–355.

[22] V. Le, G. Bloch, F. Lauer, Reduced-size kernel models for nonlinear hybrid system identification, IEEE Transactions on Neural Networks 22 (12) (2011) 2398–2405.

Appendix A. Proof of Proposition 1 (equivalence of Problems 1 and 2)

Proof. We first prove by contradiction that, for all $\boldsymbol{w} \in \mathbb{R}^{np}$, the minimum of the cost function of Problem 1 is obtained with β_{ij} set as in (2). Consider a set of optimal variables $\{\beta^*_{ij}\}$ and assume without loss of

generality that it differs from β_{ij} in (2) only for the set of points with indexes in I and that $\forall i \in I,\ \lambda_i^* \neq \lambda_i$, $\beta^*_{i\lambda_i^*} = 1$ and $\beta_{i\lambda_i} = 1$, while the constraints of Problem 1 are satisfied by both $\{\beta^*_{ij}\}$ and $\{\beta_{ij}\}$. Then, the cost in Problem 1 with $\{\beta^*_{ij}\}$ is bounded from below as follows:

$$\begin{aligned}\frac{1}{N}\sum_{i\in I}(y_i - \boldsymbol{w}_{\lambda_i^*}^T\boldsymbol{x}_i)^2 + \frac{1}{N}\sum_{i\notin I}\sum_{j=1}^{n}\beta^*_{ij}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2 &= \frac{1}{N}\sum_{i\in I}(y_i - \boldsymbol{w}_{\lambda_i^*}^T\boldsymbol{x}_i)^2 + \frac{1}{N}\sum_{i\notin I}\sum_{j=1}^{n}\beta_{ij}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2 \\ &\geq \frac{1}{N}\sum_{i\in I}(y_i - \boldsymbol{w}_{\lambda_i}^T\boldsymbol{x}_i)^2 + \frac{1}{N}\sum_{i\notin I}\sum_{j=1}^{n}\beta_{ij}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2 \\ &\geq \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n}\beta_{ij}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2,\end{aligned} \tag{A.1}$$

since, by the definition of β_{ij} in (2), $\forall i \in I$, $(y_i - \boldsymbol{w}_{\lambda_i}^T\boldsymbol{x}_i)^2 \leq (y_i - \boldsymbol{w}_{\lambda_i^*}^T\boldsymbol{x}_i)^2$.

Thus, the values $\{\beta^*_{ij}\}$ cannot be optimal and the minimum of the cost function is obtained for $\{\beta_{ij}\}$ set as in (2), except in cases where equality holds in (A.1) and in which both choices lead to similar costs.

Secondly, since the variables β_{ij} are entirely determined by $\boldsymbol{w}$ through (2), we can rewrite Problem 1 in terms of $\boldsymbol{w}$ only. This leads to

$$\begin{aligned}&\min_{\boldsymbol{w}}\ \frac{1}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{w}_{\lambda_i}^T\boldsymbol{x}_i)^2 \\ &\text{s.t.}\quad \lambda_i = \arg\min_{j=1,\dots,n}(y_i - \boldsymbol{w}_j^T\boldsymbol{x}_i)^2, \quad i = 1,\dots,N,\end{aligned}$$

where the constraints of Problem 1 are automatically satisfied by (2). Further simplifying the formulation finally yields Problem 2. □

Appendix B. Proof of Proposition 2 (convergence of Algorithm 1)

Proof. At each iteration t, the classification (3) leads to

$$F(\boldsymbol{w}^t) = \frac{1}{N}\sum_{i=1}^{N}\min_{j\in\{1,\dots,n\}}(y_i - {\boldsymbol{w}_j^t}^T\boldsymbol{x}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - {\boldsymbol{w}_{\lambda_i^t}^t}^T\boldsymbol{x}_i)^2.$$

Or equivalently, by letting $\mathcal{I}_j = \{i \in \{1,\dots,N\} : \lambda_i^t = j\}$,

$$F(\boldsymbol{w}^t) = \frac{1}{N}\sum_{j=1}^{n}\sum_{i\in\mathcal{I}_j}(y_i - {\boldsymbol{w}_j^t}^T\boldsymbol{x}_i)^2 = \frac{1}{N}\sum_{j=1}^{n}\|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t\boldsymbol{w}_j^t\|_2^2.$$

On the other hand, the update (4) ensures that $\|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t\boldsymbol{w}_j^{t+1}\|_2^2$ is minimium for all $j \in \{1,\dots,n\}$ and thus that

$$\frac{1}{N}\sum_{j=1}^{n}\|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t\boldsymbol{w}_j^{t+1}\|_2^2 \leq \frac{1}{N}\sum_{j=1}^{n}\|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t\boldsymbol{w}_j^t\|_2^2 = F(\boldsymbol{w}^t).$$

Since the inequality

$$\frac{1}{N}\sum_{i=1}^{N}\min_{j\in\{1,\dots,n\}}(y_i - {\boldsymbol{w}_j^{t+1}}^T\boldsymbol{x}_i)^2 \leq \frac{1}{N}\sum_{i=1}^{N}(y_i - {\boldsymbol{w}_{\lambda_i^t}^{t+1}}^T\boldsymbol{x}_i)^2,$$

holds for all sequences $\{\lambda_i^t\}_{i=1}^N \in \{1,\dots,n\}^N$, we have

$$F(\boldsymbol{w}^{t+1}) \leq \frac{1}{N}\sum_{j=1}^{n}\|\boldsymbol{y}_j^t - \boldsymbol{X}_j^t\boldsymbol{w}_j^{t+1}\|_2^2 \leq F(\boldsymbol{w}^t).$$

This completes the proof. □