Semi-supervised learning of dynamical systems: a preliminary study

Mirko Mazzoleni, Simone Formentin, Matteo Scandella and Fabio Previdi

***Abstract*— System identification has, in recent years, drawn insightful inspirations from techniques and concepts of the statistical learning research area. Examples of this consist in the widely adoption of regularization and kernels methods, in order to better condition the identification problem. By pursuing the same purpose, we introduce the concept of semi-supervised learning to tackle the system identification challenge. The problem, casted into the framework of the Reproducing Kernel Hilbert Spaces, leads to a new regularization technique, called manifold regularization. An application to the identification of a NFIR model is carried out, and a comparison with the standard Tikhonov regularization technique is shown.**

I. Introduction

The majority of the recent breakthroughs in system identification have their foundations built upon statistical learning methods. Whether the innovative approaches are built upon kernel methods [1], leverage the potential of Sequential Monte Carlo (SMC) filtering [2], or are casted into a Bayesian framework [3], they all share, as a common denominator, their statistical roots. In particular, kernel methods have found the interest not only in the aforementioned time-domain approaches but, recently, also of the frequency domain system identification community [4]. This motivates further research on these non-parametric approaches.

With the same spirit of the previous works, one may wonder what statistical learning still has to offer to the system identification cause. Broadly speaking, statistical learning deals with four types of problems: *Reinforcement learning* [5], *Supervised learning* [6], *Unsupervised learning* [6] and *Semi-supervised learning* [7]. In the latter class of problems, both supervised and unsupervised data are supposed to be available. Such a scenario may occur when performing a measurement is costly or it is a destructive experiment [8]. A variety of techniques have been developed to make use of the additional (unsupervised) data, to represent a function that (statically) maps inputs to outputs, both in classification [8] and regression [9] problems.

When moving to *dynamical* system identification, we can legitimately wonder *if* and *how* the presence of *inputs with no corresponding outputs* (i.e. unsupervised data) can be useful to the system identification purpose. This would translate the system identification problem from *supervised* to *semi-supervised*, opening for the inclusion of related techniques.

S. Formentin is with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, via G. Ponzio 34/5, 20133 Milano, Italy. M. Mazzoleni, M. Scandella and F. Previdi are with the Department of Management, Information and Production Engineering, University of Bergamo, Via G. Marconi 5, 24044 Dalmine (BG), Italy. Email to: mirko.mazzoleni@unibg.it.

In general, in regression problems like system identification ones the aim is to learn the function that generates the supervised data. These are data that are at our disposal, and are such that both inputs and outputs are known. When, in addition to the supervised data, other inputs are available (but without the corresponding output), their disposition in the regressor space gives additional information about how the unknown outputs could be.

An algorithm to exploit such additional information has been presented in [9]: however, the described method is a *transductive learning* one. This means that its purpose is to correcly predict only *specific* test data. This is in contrast with the *inductive learning* reasoning, that tries to generalize from specific training examples to general rules (which are then applied to specific test cases).

The method presented in this paper, named *Semi-supervised System Identification* (SSI), gives a solution for the *inductive semi-supervised learning* problem, by leveraging on the framework of the Reproducing kernel Hilbert Spaces (RKHS) [10]. The aim is to identify the unknown true system $\mathcal{S}$ given a set of input-output data (supervised dataset), leveraging also on the information contained in a set of input-only measurements (unsupervised dataset). The prior information embedded in the distribution of the unsupervised dataset are employed in the form of an additional regularization term, called *manifold regularization*. Classical Tikhonov regularization impacts the *global smoothness* of the learned function, while we will show that the manifold regularization term implies, instead, the concept of *local smoothness*.

Since, in dynamic systems, the regressors may also contain the past output samples (not available in correspondence of the unlabeled input points), we will restrict our analysis to the case of Nonlinear Finite Impulse Response (NFIR) systems, leaving this extension to future research. We should here acknowledge that the application of a manifold regularization term in (inductive) static fuction learning has already been faced in [11], [12]. We built on these results, first by applying the methodology to the system identification case and, afterwards, by introducing a new method for generating the unsupervised input data (without the need to perform an additional experiment).

II. Problem statement

Let the NFIR Single-Input Single-Output (SISO) model be defined as:

$$\mathcal{S}: \; y(t+1) = g\big(\varphi(t)\big) + e(t), \tag{1}$$

where $y(t) \in \mathbb{R}$ denotes the system output, g is a nonlinear function, $\varphi(t) \in \mathbb{R}^{m\times 1}$ is the regression vector such that $\varphi(t) = [u(t), \cdots, u(t-m+1)]^T$, and $e(t) \in \mathbb{R}$ is an additive white noise. From now on, m will be referred to as the *model order*. The objective of this work is to identify a system of type (1), assuming that m is known[1]. We suppose furthermore that two different datasets are available: a *supervised* data set $\mathcal{D}_S$ and an *unsupervised* one $\mathcal{D}_U$. The supervised dataset is such that:

$$\mathcal{D}_S = \left\{ \left(u_S(t), y(t) \right) \quad | \quad 1 \leq t \leq N_S \right\}, \tag{2}$$

where $u_S(t)$ is the input at time t, $y(t)$ is the output associated with the input $u_S(t)$, and N_S is the number of supervised data. The unsupervised dataset $\mathcal{D}_U$ is defined as:

$$\mathcal{D}_U = \left\{ \left(u_U(t) \right) \quad | \quad 1 \leq t \leq N_U \right\}, \tag{3}$$

where $u_U(t)$ is an input for which the associated output has not been measured, and N_U is the number of these unsupervised input measurements. Notice that the dataset $\mathcal{D}_S$ contains both input and output samples, while the dataset $\mathcal{D}_U$ consists of only input measurements. In order to obtain a more compact representation, it is useful to represent the observations and the regressors in matrix form. By using the supervised dataset $\mathcal{D}_S$, we obtain the output vector $Y \in \mathbb{R}^{N\times 1}$:

$$Y = \begin{bmatrix} y(m+1) & \cdots & y(N_S) \end{bmatrix}^T, \tag{4}$$

which contains the observations $y(t)$ stacked in row, and $N = N_S - m$ is the number of outputs that it is possible to employ for the identification stage, given the model order m. In the same way, it is possible to construct the N *supervised* model's regressors, for $m \leq t \leq N_S - 1$, as:

$$\varphi_S(t) = \begin{bmatrix} u_S(t) & \cdots & u_S(t-m+1) \end{bmatrix}^T \tag{5}$$

where $\varphi_S(t) \in \mathbb{R}^{m\times 1}$. The regressors' matrix $\Phi \in \mathbb{R}^{N\times m}$ can be defined by stacking all supervised regressors $\varphi_S(t)$, leading to:

$$\Phi = \begin{bmatrix} \varphi_S(m) & \cdots & \varphi_S(N_S-1) \end{bmatrix}^T. \tag{6}$$

Remember now that, in addition to the supervised dataset $\mathcal{D}_S$, we have at our disposal also the unsupervised dataset $\mathcal{D}_U$, containing only input samples. It is therefore possible to construct the model's regressors as in (5), by leveraging on $\mathcal{D}_U$. There are therefore $N_{rU} = N_U - m + 1$ available *unsupervised* model's regressors, each one of them defined, for $m \leq t \leq N_S - 1$, as:

$$\varphi_U(t) = \begin{bmatrix} u_U(t) & \cdots & u_U(t-m+1) \end{bmatrix}^T \tag{7}$$

where $\varphi_U(t) \in \mathbb{R}^{m\times 1}$. It is then possible to group all of these unsupervised regressors into an (unsupervised) regressors' matrix $\Phi_U \in \mathbb{R}^{N_{rU}\times m}$, as:

$$\Phi_U = \begin{bmatrix} \varphi_U(m) & \cdots & \varphi_U(N_U) \end{bmatrix}^T, \tag{8}$$

[1]The knowledge of m is generally not available and an estimate is usually derived from the data. Since the issue is not trivial, this is postponed to future research.

Combining the input datasets, we can define the joint matrix, containing both supervised (5) and unsupervised (7) regressors, as:

$$\widetilde{\Phi} = \begin{bmatrix} \Phi^T & \Phi_U^T \end{bmatrix}^T, \tag{9}$$

where $\widetilde{\Phi} \in \mathbb{R}^{N_r\times m}$ and $N_r = N + N_{rU}$ is the total number of regressors, both supervised and unsupervised. From now on, for simplicity, the i-th row of $\widetilde{\Phi}$ and Y will be denoted as $\varphi(i)$ and $y(i)$, respectively.

The aim now is to *identify the system $\mathcal{S}$ by employing the information contained in $\mathcal{D}_S$ and $\mathcal{D}_U$.*

III. Manifold regularization

When can $\mathcal{D}_U$ be of some use into discovering the relation between inputs and outputs? This is the case if the marginal probability density $p(\varphi)$ which, we suppose, generates the inputs, happens to be informative about the conditional distribution $p(y|\varphi)$, describing the possible outputs values in correspondence of the input regressor φ [12]. We can make this possible by stating a specific assumption about the connection between the marginal and the conditional distributions [8]:

Assumption 1: **Semi-supervised smoothness**
If two regressors $\varphi(i)$ and $\varphi(j)$ in a high-density region are close, then so should be their corresponding outputs $y(i)$ and $y(j)$.

Assumption 1, therefore, constrains the solution to be smooth with respect to the *manifold* onto which the regressors lie. This can be enforced by a *proper regularization term*, that should reflect the intrinsic structure of $p(\varphi)$. This term has a different taste with respect to the standard Tikhonov one, that instead, enforces a *global smooth behaviour* to the unknown function. One of the first attempts to formalize Assumption 1 has been taken in [13]. Here, we take the opposite path: our aim is to define a regularization term which, in turn, enforces Assumption 1, without a specific set of choices and ad-hoc definitions. In this view, a possible choice for the manifold regularization term has been advocated in [12] as:

$$S_g = \int_{\mathcal{G}} \|\nabla g(\varphi)\|^2 \cdot p(\varphi)\, d\varphi, \tag{10}$$

where $\mathcal{G} \subseteq \mathbb{R}^{m\times 1}$ is the regressor space and $p(\varphi)$ denotes the probability density function of the regressors defined over $\mathcal{G}$. The main idea behind the manifold regularization rationale considered here is that, if Assumption 1 holds, the gradient of g, and so S_g, must be small. Then, minimizing S_g with respect to model parameters or missing outputs values is a way to enforce Assumption 1.

In the standard supervised learning approach, the information about the input distribution $p(\varphi)$ is rarely used. This is the case because, most of the times, $p(\varphi)$ is unknown and the smoothness index S_g cannot be computed exactly. It turns out that S_g can be approximated using the *regressor graph* [11], [12]. This is a weighted complete graph with the (supervised and unsupervised) regressors as its vertexes, and the weight of each edge defined as $w_{i,j} = e^{-\frac{\|\varphi(i)-\varphi(j)\|^2}{2\sigma_e{}^2}}$,

where $\sigma_e \in \mathbb{R}$ is a tuning parameter. Now, let's consider the Laplacian matrix $L = D - W$, where $D \in \mathbb{R}^{N_r \times N_r}$ is the diagonal matrix with elements $D_{ii} = \sum_{j=1}^{N_r} w_{i,j}$, and $W \in \mathbb{R}^{N_r \times N_r}$ is the matrix composed by the weights $w_{i,j}$. A higher value of $w_{i,j}$ indicates that two regressors are similar. This rationale derives from a manifold learning algorithm called Laplacian Eigenmaps [14]. It can be shown that [12]:

$$S_g \simeq \widetilde{Y}^T \cdot L \cdot \widetilde{Y}, \tag{11}$$

where $\widetilde{Y} = \left[g\big(\varphi(1)\big), \cdots, g\big(\varphi(N_r)\big)\right]^T \in \mathbb{R}^{N_r \times 1}$ contains the *noiseless* outputs, corresponding to *both supervised and unsupervised input regressors*[2]. In order to obtain the approximation of S_g (11), *only the input regressors are needed*. Thus, both supervised and unsupervised regressors can be employed for this purpose. Notice that Y differs from $\widetilde{Y}$, since the former is a *noisy* vector of N observations, while the latter is a *noiseless* vector of N_r observations.

IV. The semi-supervised approach

In this work, we will consider the realistic case where g is unknown and therefore no prior parameterization is available. A powerful tool for dealing with such challenges is the framework of the RKHS. Therefore, a kernel-based nonparametric approach, based on RKHS, is proposed. The method embodies the notion of manifold regularization, in order to take advantage of the presence of unsupervised data.

Suppose now that g belongs to a RKHS $\mathcal{H}$ defined using the kernel K. The tipical variational formulation consists into finding the best function $\hat{g}$ according to the criterion [16]:

$$\hat{g} = \underset{g \in \mathcal{H}}{\arg\min} \sum_{t=1}^{N} \Big(y\,(t) - g\big(\varphi\,(t)\,\big)\Big)^2 + \lambda \cdot \|g\|_{\mathcal{H}}^2, \tag{12}$$

where $\|g\|_{\mathcal{H}}^2$ is the Tikhonov regularization term and $\lambda \in \mathbb{R}$ controls the regularization strength. The solution to (12) is given by the *representer theorem* [17]:

$$\hat{g}\big(\varphi(t)\big) = \sum_{s=1}^{N} c_s K\big(\varphi\,(t)\,, \varphi\,(s)\,\big) = \sum_{s=1}^{N} c_s r_{\varphi(s)}\big(\varphi\,(t)\,\big), \tag{13}$$

for some N-tuple $c = [c_1, c_2, \ldots, c_N]^T \in \mathbb{R}^{N \times 1}$. The functions $r_{\varphi(s)}(\cdot)$ are called *representers* of the point $\varphi(s)$.

In order to include information about the local smoothness of the function (leveraging on the unsupervised data points), it is meaningful to add the *manifold regularization term* (11) to (12), leading to [12]:

$$\hat{g} = \underset{g \in \mathcal{H}}{\arg\min} \sum_{t=1}^{N} \Big(y\,(t) - g\big(\varphi\,(t)\,\big)\Big)^2 + \lambda \|g\|_{\mathcal{H}}^2 + \lambda_M \widetilde{Y}^T L \widetilde{Y}, \tag{14}$$

where $\lambda_M \in \mathbb{R}$ has the same role as λ.

It is possible to show that the representer theorem still holds for the cost function (14), and the solution is given by considering all N_r regressors, both the N supervised and the N_{rU} unsupervised ones [12]:

$$\hat{g}\big(\varphi(t)\big) = \sum_{s=1}^{N_r} \tilde{c}_s K\big(\varphi\,(t)\,, \varphi\,(s)\,\big) = \sum_{s=1}^{N_r} \tilde{c}_s r_{\varphi(s)}\big(\varphi\,(t)\,\big), \tag{15}$$

for some N_r-tuple $\tilde{c} = [\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_{N_r}]^T \in \mathbb{R}^{N_r \times 1}$.

In order to proper evaluate the effect of the new introduced regularization term (11), we will suppose from now on that the Tikhonov term in the cost function (14) is set to zero, leading to the following *purely semi-supervised formulation*:

$$\hat{g} = \underset{g \in \mathcal{H}}{\arg\min} \sum_{t=1}^{N} \Big(y\,(t) - g\big(\varphi\,(t)\,\big)\Big)^2 + \lambda_M \cdot \widetilde{Y}^T \cdot L \cdot \widetilde{Y}. \tag{16}$$

The vector $\widetilde{Y}$ introduced in (11) can the be rewritten as:

$$\widetilde{Y} = \widetilde{\mathcal{K}} \tilde{c}, \tag{17}$$

where $\widetilde{\mathcal{K}} \in \mathbb{R}^{N_r \times N_r}$ is a semidefinite positive and symmetric matrix (also called Gram matrix or kernel matrix) such that $\widetilde{\mathcal{K}}_{ij} = K\big(\varphi(i), \varphi(j)\big)$.

Now, by using results (15) and (17) it is possible to write the minimization problem (16) in such a way that it depends only on the unknown vector $\tilde{c} \in \mathbb{R}^{N_r \times 1}$:

$$\widehat{c} = \underset{\tilde{c} \in \mathbb{R}^{N_r}}{\arg\min} \left\| \begin{bmatrix} Y \\ 0_{N_{rU}} \end{bmatrix} - P \cdot \widetilde{\mathcal{K}} \tilde{c} \right\|_2^2 + \lambda_M \cdot \tilde{c}^T \cdot \widetilde{\mathcal{K}} \cdot L \cdot \widetilde{\mathcal{K}} \cdot \tilde{c}, \tag{18}$$

where $0_{N_{rU}} \in \mathbb{R}^{N_{rU} \times 1}$ is column vector of all zeros. The matrix $P \in \mathbb{R}^{N_r \times N_r}$ permits to select only the elements of $\widetilde{\mathcal{K}}$ that contribute to explain the N supervised data points, such that:

$$P = \begin{bmatrix} I_N & 0 \\ 0 & 0 \end{bmatrix}. \tag{19}$$

Since (18) is now quadratic in $\tilde{c}$, its minimization can be carried out analytically. The minimizer of (18), therefore, can be found by solving the linear system:

$$\left[P \cdot \widetilde{\mathcal{K}} + \lambda_M \cdot L \cdot \widetilde{\mathcal{K}}\right] \cdot \widehat{c} = \begin{bmatrix} Y \\ 0_{N_{rU}} \end{bmatrix}. \tag{20}$$

The unsupervised points contribute to the overall estimated function via the matrix $\widetilde{\mathcal{K}}$.

It is now interesting to show a comparison between searching the unknown function following formulation (14) or (16). Consider a static unknown function $g(x)$ that presents a discontinuity point at $x = 0$. The Tikhonov term enforces a global smooth behaviour, while the manifold term strives for local smoothness. The employed kernel is the Gaussian kernel $K\big(\varphi(t), \varphi(s)\big) = e^{-\frac{\|\varphi(t) - \varphi(s)\|^2}{2\sigma^2}}$, where σ regulates the Gaussian dispersion. Figure 1 shows the results of a regularization network that employs only the Tikhonov regularization for different values of λ and σ. In this case, the unsupervised points are of no use, and therefore are not depicted. When $\lambda = 0$, also the Tikhonov term is absent, and the estimated function interpolates each one of the supervised points. Choosing a low value of σ, we are defining a function space that admits also non-smooth functions [18]. Because

[2]It is interesting to observe that the same problem structure (11) is shared by other manifold learning methods, although they do not use L, but a different symmetric matrix [15].

of this, the learned function is composed by a series of sharply peaked Gaussians, centered at the observed points. This is in line with the definition given by the representer theorem, given that, in the case of a Gaussian kernel, the representers are still Gaussian functions. As σ grows, the estimated function gets smoother, fitting worse and worse the high variation regions of the true underlying function. The effect of the regularization hyperparameter λ is that of weighting the regularization and the error cost function. With high values of λ, the estimated function tends to the zero one: this is in line with the parametric approach, where a high λ value makes all parameters' estimates null. In all of these cases, given the global nature of the imposed regularization, the estimated function fails to approximate well the discountinuity region.

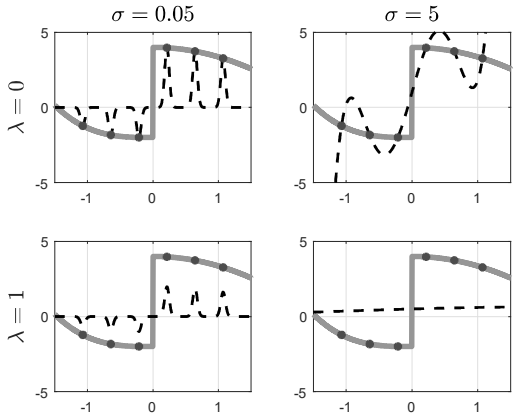

Fig. 1: Example of hyperparameters' sensitivity when employing only the Tikhonov regularization term. The plots depict the true unknown function (solid green line), the supervised data (red dots), and the estimated function (dotted black line)

The function's estimation example using only the manifold regularization term is depicted in Figure 2. Here, we suppose that unsupervised points are available in a neighbourhood of the discontinuity point. The method should *not* regularize in this region, in order to allow non-smooth (rapid) variation of the estimated function, and should enforce, instead, smoothness elsewhere. By choosing an appropriate low value of σ, it is possible to fit the function even in the discountinuity region, being not sensible to the variation of λ_M. High values of σ makes the estimate smoother, just like as λ controlling the Tikhonov regularization. Increasing λ_M make the function as smooth as possible: in this case, this means that the manifold regularization term is weighted much. This, in turns, translates into making each domain point similar to the other, and the estimated function reduces to the mean of the supervised points.

V. Unsupervised inputs selection

In real-world identification of dynamic systems, contrary to the standard semi-supervised problems encountered in statistical learning, the unsupervised data set $\mathcal{D}_U$ may not be a problem input as in Figure 2, but, instead, a *design*

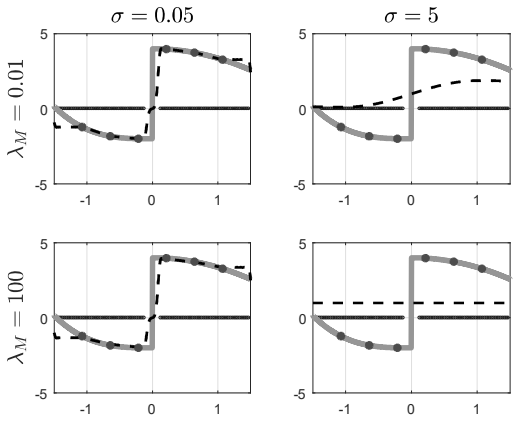

Fig. 2: Example of hyperparameters' sensitivity when employing only the manifold regularization term. The plots depict the true unknown function (solid green line), the supervised data (red dots), the unsupervised data (blue dots), and the estimated function (dotted black line). The hyperparameter σ_e is fixed to 0.01

parameter. In some cases, $\mathcal{D}_U$ may contain some input time series which are likely to excite the system dynamics in future operating conditions (when the model will be used). More often, since this additional data set affects the model quality, $\mathcal{D}_U$ could be chosen to enforce Assumption 1 to be true. Notice that to obtain such an additional data set, *it is not required to run a new experiment on the plant*. Following Figure 2, if the discontinuity region would be known, a possible unsupervised points generation method could be to generate the additional inputs as in the example. If the discontinuity region is not known, then, a more general method has to be devised for the generation of $\mathcal{D}_U$.

Before discussing the choice of $\mathcal{D}_U$, notice that Assumption 1 requires only that, inside the same high density region, the regressors have a similar corresponding output, namely that their difference is "small". For this reason, the proposed method will generate the unsupervised regressors in the neighborhood of the supervised ones, where, if the system is smooth enough, they should have a similar corresponding output. This approach will generate a regressors set similar to the one shown in Figure 3, where it is possible to notice the presence of N_S regions, containing a supervised regressor and some unsupervised ones. The algorithm used to select $\mathcal{D}_U$ is indicated next. Let $\mathcal{D}_U$ be composed of p unsupervised datasets $\mathcal{D}_U^i$, $i = 1, \ldots, p$ as:

$$\mathcal{D}_U^i = \left\{ \left(u_U^i \left(t \right) \right) \quad | \quad 1 \leq t \leq N_S \right\} \tag{21}$$

where $u_U^i \left(t \right) = u_S \left(t \right) + v^i(t)$, $v^i(t)$ is a random variable and p is a tuning knob of the method. Each one of the p new (unsupervised) datasets contain therefore exactly N_S unsupervised input regressors, see Figure 3.

The value of $v^i(t)$ determines the distance of the p unsupervised points with respect to the supervised one (proportional to the area of the regressors' regions): therefore, it has to be small enough to guarantee that the system output does not vary significantly inside these

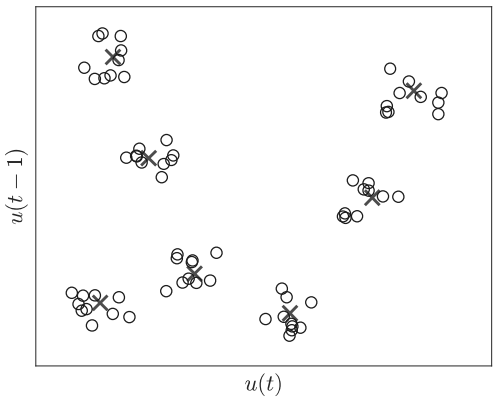

Fig. 3: An example of unsupervised regressors' selection, for a system with $m = 2$ using $p = 10$. The plot represents the supervised regressors (red crosses) and the unsupervised regressors (blue circles)

regions. The choice of $v^i(t)$ will be discussed later.

From such p datasets, it is possible to determine the quantities defined in Section II. Since the unsupervised points are generated in correspondence of the supervised ones, we have N employable unsupervised regressors for each one of the p datasets. This leads to a total of $N_{rU} = p \cdot N$ unsupervised regressors $\varphi_U^i(t) \in \mathbb{R}^{m\times 1}$, $i = 1,\dots,p$. Each one of them is such that, according to (7), for $m \le t \le N_S - 1$:

$$\varphi_U^i(t) = \begin{bmatrix} u_U^i(t) & \cdots & u_U^i(t-m+1) \end{bmatrix}^T. \tag{22}$$

From the unsupervised regressors computed in (22) using the i-th dataset, it is possible to form the i-th unsupervised regressors' matrix $\Phi_U^i \in \mathbb{R}^{N\times m}$ as in (8):

$$\Phi_U^i = \begin{bmatrix} \varphi_U^i(m) & \cdots & \varphi_U^i(N_S-1) \end{bmatrix}^T. \tag{23}$$

The complete (unsupervised) regressors' matrix $\Phi_U \in \mathbb{R}^{N_{rU}\times m}$ can there be therefore composed by stacking the matrices (23), $i = 1,\dots,p$:

$$\Phi_U = \begin{bmatrix} \left(\Phi_U^1\right)^T & \cdots & \left(\Phi_U^p\right)^T \end{bmatrix}^T. \tag{24}$$

A reasonable criterion for the selection of the random variable $v^i(t)$ is to consider that the regions should not mix with each other, since this would lead to non-smooth functions (e.g., with jumps in certain points). It is then useful to introduce a tuning parameter $\alpha \in \mathbb{R}$, allowing to regulate the regions' maximum area, and that highlights if the regions mix or not. In particular, in the method indicated next, $\alpha = 1$ corresponds to the threshold between mixed regions (achieved using $\alpha < 1$) and completely distinct regions ($\alpha > 1$). In order to use α, it is necessary to define a distribution of $v^i(t)$ that depends on α and guarantees the aforementioned properties. A possible way is to use a uniform distribution:

$$v^i(t) \sim \mathrm{U}(-h,h) \quad 1 \le t \le N_S,\ \ i = 1,\dots,p \tag{25}$$

where $h > 0$ determines the area of the unsupervised points regions. To impose distinct regions, the following inequalities must hold:

$$\left\|\varphi_U^i(t) - \varphi_S(t)\right\| \le \frac{d}{2} \quad m \le t \le N_S - 1,\ \ i = 1,\dots,p \tag{26}$$

where d is the distance between the two closest supervised regressors. After some computations, it can be shown that (26) can be written as:

$$\sum_{j=1}^{m} \left(v^i(t-j+1)\right)^2 \le \left(\frac{d}{2}\right)^2 \quad m \le t \le N_S - 1, \tag{27}$$

Since $\left|v^i(t-j+1)\right| \le h$ (it is generated from the random variable (25)), the inequalities (27) hold if $\sum_{i=1}^{m} h^2 \le \left(\frac{d}{2}\right)^2$. Recalling that $h \ge 0$, this corresponds to impose that $h \le \frac{d}{2\sqrt{m}}$. This condition imposes a constraint for h to maintain N_S distinct regions. To make such a constraint more or less conservative, it is possible to use α, for examples, as follows:

$$h = \frac{d}{2\alpha\sqrt{m}} \tag{28}$$

VI. Results and Discussion

In this section, a numerical example is provided to show the effectiveness of the Semi-Supervised Identification algorithm, presented in the previous sections, that employs the manifold regularization term as in (16). The approach is compared with the standard formulation (12), where only the Tikhonov regularization is considered. We employed the Gaussian kernel to estimate the second order ($m = 2$) NFIR system:

$$\begin{aligned} y(t) =& 1.432 \cdot u(t)^2 + +1.034 \cdot u(t-1)^3\, 1.564 \cdot u(t-1)^2 + \\ &+ 3.234 \cdot u(t)\, u(t-1) + 2.145 \cdot u(t)^3 + \\ &+ 3.432 \cdot u(t)^2\, u(t-1) + 2.745 \cdot u(t)\, u(t-1)^2. \end{aligned} \tag{29}$$

The supervised dataset $\mathcal{D}_S$ generated from (29) is composed by very few points, namely $N_S = 15$ measures, corrupted by a Gaussian white noise input of zero mean, unitary variance and signal to noise ratio of $15\,\mathrm{dB}$. The problem is then badly conditioned and is well suited for testing the proposed methodology. The unsupervised input dataset $\mathcal{D}_U$ has been generated according to Section V.

The hyperparameters of the Tikhonov regression method (12) are the regularization coefficient λ (chosen from 100 evenly spaced values in $\left[0, 10^{-3}\right]$) and the shape parameter of the Gaussian kernel σ (chosen from the values $[3, 6, 10, 11]$). The hyperparameters of the manifold regression method (16) are instead not only the regularization coefficient λ_M (chosen from 100 evenly spaced values in $\left[10^{-6}, 10^{-1}\right]$) and σ, but also the shape parameter of the Laplacian Eigenmaps σ_e (chosen from the values $[0.1, 1, 10, 100]$), the parameter controlling the area of the generated unsupervised points α (chosen from 100 evenly spaced values in $[1, 17]$) and the number of additional unsupervised datasets p (chosen from the values $[2, 3, 4]$).

In order to tune the respective hyperparameters of the

methods, an additional supervised dataset $\mathcal{D}_V$ of $N_V = 1000$ points has been generated in the same way as $\mathcal{D}_S$. For obvious reasons, $\mathcal{D}_V$ should not be available. This problem is postponed to future research.

In order to assess the overall performance of the estimation methods, a supervised testing dataset $\mathcal{D}_T$ of $N_T = 10000$ points is employed, generated analogously to $\mathcal{D}_S$. Using $\mathcal{D}_T$ it is possible to evaluate the NRMSE (Normalized Root-Mean Square Error) fitness metric:

$$\text{Fit} = 100 \cdot \left(1 - \frac{\left\|Y_T - \widehat{Y}\right\|}{\|Y_T - \overline{y}_T \cdot \mathbf{1}\|}\right), \tag{30}$$

where $\widehat{Y}$ is the vector of the estimated test outputs using the test inputs, Y_T is the true test outputs vector, $\overline{y}_T$ is the mean of Y_T and $\mathbf{1}$ denotes a vector of ones of suitable dimension.

In particular, a Montecarlo test using $N_M = 100$ sets of measures (with different random initializations) is proposed, to show the statistical significance of the method. The notched boxplots in Figure 4 depicts that SSI significantly outperform the Tikhonov regularization method, showing a significant difference in the medians. The SSI method, in addition, exhibits a lower estimation variance.

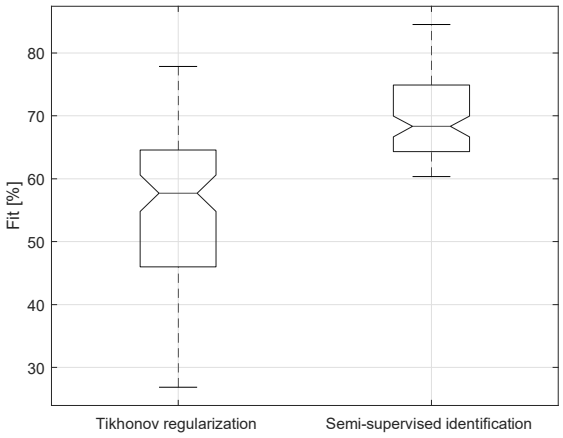

Fig. 4: Comparison between Tikhonov regression and semi-supervised identification in terms of the NRMSE measure of fitness. The boxplots represent a total of 100 different simulation and estimation trials

Given the numerical result, the belief is that they clearly show the potential of the semi-supervised approach for nonlinear system identification with respect to state of the art techniques. The price to pay is the fact that, e.g., compared to Tikhonov regression, three additional knobs need to be tuned, namely σ_e, α, p. However, notice that they are characterized by a clear physical interpretation: σ_e controls how much far two points can be considered similar or "connected" (if $\sigma_e \to \infty$ the result is an adiacency graph); p (that is, N_{rU}/N) indicates the relevance of the prior smoothness assumption over the measured data, while α represents a degree of smoothness. Therefore, they can be reasonably tuned with some (mild) prior information on the system dynamics. Moreover, simulations showed that, at least for the considered example, the performance are not sensitive to a fine tuning of such parameters.

VII. Conclusions and future developments

In this work, a semi-supervised learning approach suited to nonlinear dynamical system identification has been developed. The method applies to NFIR models and turns out to be equivalent to a weighted regularization network. The different smothness behaviours induced on the learned function by Tikhonov and the manifold regularization term have been pointed out. The approach has been shown to outperform the statistical performance of Tikhonov regularization, when the additional unsupervised dataset is selected as indicated by the method.

Future research work will be dedicated to the extension of the semi-supervised paradigm to auto-regressive models and to a comparison with more complex regularization techniques. Another challenging open problem concerns the estimation of the hyperparameters.

References

[1] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[2] T. B. Schön, A. Wills, and B. Ninness, "System identification of nonlinear state-space models," *Automatica*, vol. 47, no. 1, pp. 39–49, 2011.

[3] A. Svensson and T. B. Schön, "A flexible state-space model for learning nonlinear dynamical systems," *Automatica*, vol. 80, pp. 189–199, 2017.

[4] M. A. H. Darwish, J. Lataire, and R. Toth, "Bayesian frequency domain identification of LTI systems with OBFs kernels," in *20th World Congress*. IFAC, 2017.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[6] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[7] X. Zhu, "Semi-supervised learning," in *Encyclopedia of machine learning*. Springer, 2011, pp. 892–897.

[8] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*, 1st ed. The MIT Press, 2010.

[9] H. Ohlsson and L. Ljung, "Semi-supervised regression and system identification," in *Three Decades of Progress in Control Sciences*. Springer, 2010, pp. 343–360.

[10] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[11] M. Belkin, P. Niyogi, and V. Sindhwani, "On manifold regularization." in *AISTATS*, 2005, p. 1.

[12] ——, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[13] V. Castelli and T. M. Cover, "The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter," *IEEE Transactions on information theory*, vol. 42, no. 6, pp. 2102–2117, 1996.

[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[15] L. Cayton, "Algorithms for manifold learning," *Univ. of California at San Diego Tech. Rep*, vol. 12, pp. 1–17, 2005.

[16] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[17] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.

[18] R. Vert and J.-P. Vert, "Consistency and convergence rates of one-class svms and related algorithms," *Journal of Machine Learning Research*, vol. 7, no. May, pp. 817–854, 2006.