Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-15 (2018) 622–627

Identification of nonlinear dynamical system with synthetic data: a preliminary investigation

M. Mazzoleni *, **M. Scandella** *, **S. Formentin** **, **F. Previdi** *

* *Department of Management, Information and Production engineering University of Bergamo, via Galvani 2, 24044 Dalmine (BG), Italy (e-mail: mirko.mazzoleni@unibg.it).*

** *Department of Electronics, Information and Bioengineering, Politecnico di Milano, via G. Ponzio 34/5, 20133 Milano, Italy (e-mail: simone.formentin@polimi.it).*

Abstract:
This paper introduces a new rationale for learning nonlinear dynamical systems. The method makes use of an additional identification dataset, obtained without performing a new experiment on the system under study. The data are generated in an automatical manner, starting from a set of experimentally acquired measurements. In order to leverage the additional generated information, fundamental techniques from the machine learning field known as Semi-Supervised Learning (SSL) are employed and adapted. The problem is then cast as a regularized parametric learning problem. The effectiveness of the proposed approach is assessed on various nonlinear benchmark systems via repeated simulations, comparing the obtained results with a standard regularization method for learning parametric models.

Keywords: System Identification; Semi-Supervised Learning; Regularization

1. INTRODUCTION

System identification has seen, in recent years, a flourishing development in both theories and applications. One of the reasons that most contributed to this growth was the introduction of statistical learning techniques into the system identification framework. Some of the newest approaches include: i) the use of *Sequential Monte Carlo* (SMC) techniques for state-space models, Svensson and Schön (2017), implemented by means of probabilistic programming languages, Vajda (2014); ii) the employment of *kernel methods* to impose smoothness constraints on the learned function via a regularization network, Pillonetto et al. (2014); Evgeniou et al. (2000). Regularization techniques, initially developed for static systems, Friedman et al. (2001), consisted into constraining the loss provided by the Empirical Risk Minimization (ERM) principle, Vapnik (1998). It is interesting to notice that, however, all of the aforementioned methods can be cast into a Bayesian framework. This means that prior information on parameters or on the entire function can be leveraged to guide the learning procedure by employing, as an example, Gaussian Processes (GP), Rasmussen and Williams (2006).

Another possible way to induce regularization is to use artificially generated data. Ridge regression estimates can be obtained by ordinary least squares regression on an augmented dataset, where generated response values are set to zero. In this way, the fitting procedure is forced to shrink also the coefficients toward smaller values, Friedman et al. (2001). Authors in Chapelle et al. (2001) formalized the Vicinal Risk Minimization (VRM) principle. Here, additional virtual examples can be drawn from a defined vicinity distribution of the training examples, to enlarge the support of the training distribution. In this work the authors showed how, using the VRM approach, one can obtain the regularized Ridge regression and Support Vector Machine (SVM), Friedman et al. (2001), solutions. This method is currently often applied for training deep neural networks, in particular when performing image classification. In fact, it is common to define the vicinity of one image as the set of its horizontal reflections, slight rotations, and mild scalings, see Krizhevsky et al. (2012). In this setting, a recent data augmentation technique has been introduced to alleviate overfitting problems and sensitivity to adversarial examples, Zhang et al. (2017). A related idea has been presented by Abu-Mostafa (1994), where model constraints are implemented by adding artificial data examples that satisfy them. Differently from previously cited methods, here the learning "hints" can be designed by relying only on the independent variables.

Following the previous line of reasoning, we wondered if synthetic data generation can be helpful for system identification. In particular, we focus on the case where additional input data (which, in dynamic systems, may contain also past output samples) can be devised. Artificial data can be used to regularize the estimation, especially in a small-data regime. In this case, even a simple model is hard to compute due to a very limited amount of samples in the dataset. Examples of this kind can be found among, e.g., biomedical applications like glucose dynamics, Cobelli

10.1016/j.ifacol.2018.09.227

et al. (2009), or biochemical/biotechnology processes, where experiments can be of great cost, see Yang et al. (2012). Another similar scenario occurs when performing a measurement is costly and/or destructive, Chapelle et al. (2010).

Suppose now to have at disposal such additional regressors' dataset. The problem translates into how to effectively leverage such information. One choice that naturally arises is the framework of Semi-Supervised Learning (SSL), Zhu (2011). Differently from the pure supervised and unsupervised cases, in the SSL paradigm both supervised and unsupervised data are supposed to be available. The synthetic regressors' dataset plays therefore the role of the *unsupervised dataset*. Acquisitions measured through a real experiment on the system, comprising both input and ouput values, constitute the *supervised dataset*. In this paper, we propose the Semi-Supervised Identification (SSI) approach. Here, prior information embedded in the distribution of the unsupervised dataset is employed in the form of an additional regularization term, called *manifold regularization*. The *contribution* of this paper is to apply this rationale to autoregressive models. The idea that we adopted here is to generate the additional regressors in the neighborhood of the existing ones, obtained from measured data.

The remainder of the paper is organized as follows. Section 2 defines the problem formulation. Section 3 describes the use of the semi-supervised techniques for making use of synthetic data. In Section 5, the semi-supervised parametric approach for learning autoregressive models is highlighted. Section 6 compares simulated results of the proposed approach with Ridge regression. Lastly, Section 7 is devoted to concluding remarks and future developments.

2. PROBLEM STATEMENT

2.1 System and data definitions

Let the NARX 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, $e(t) \in \mathbb{R}$ is an additive white noise, and $\varphi(t) \in \mathbb{R}^{(r+q)\times 1}$ is the regression vector such that:

$$\varphi(t) = \big[y(t), \cdots, y(t-r+1), \; u(t), \cdots, u(t-q+1)\big]^T, \tag{2}$$

where r and q denotes the orders of the autogressive and exogenous part, respectively, with $k = r + q$. We suppose that the orders r and q are known, and postpone their estimate to future research.

In this paper, in order to test the idea firstly on simpler problems and to guarantee that a global optimum model can be achieved, we restrict ourselves to NARX models which are linear in the parameters. Specifically, let the function g in (1) be:

$$g\big(\varphi(t)\big|\vartheta\big) = \vartheta^T \cdot \gamma\big(\varphi(t)\big) = \sum_{i=1}^{m} \vartheta_i \cdot \gamma_i\big(\varphi(t)\big), \tag{3}$$

where $\gamma : \mathbb{R}^{k\times 1} \to \mathbb{R}^{m\times 1}$ is a nonlinear mapping vector function, and γ_i represents the i-th component of γ. Thus, $\gamma_i\big(\varphi(t)\big)$ is the i-th feature. The variable $\vartheta \in \mathbb{R}^{m\times 1}$ represents the vector of the m parameters that have to be estimated from data.

The aim now is to identify systems of type (1) in the form (3), with the help of two different datasets: a *supervised* data set $\mathcal{D}_S$, obtained from real experiments, and an *unsupervised* one $\mathcal{D}_U$, synthetically generated.

The supervised dataset is such that:

$$\mathcal{D}_S = \big\{\big(u_S(t), y(t)\big) \quad | \quad 1 \le t \le N_S\big\}, \tag{4}$$

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:

$$\mathcal{D}_U = \big\{\big(u_U(t), y_U(t)\big) \quad | \quad 1 \le t \le N_U\big\}, \tag{5}$$

where $u_U(t)$ and $y_U(t)$ are, respectively, an input and an output artificially generated, and N_U is the number of these unsupervised data.

Notice that both the datasets $\mathcal{D}_S$ and $\mathcal{D}_U$ contain input and output samples, although the latter consists only of synthetic data.

2.2 Creating the datasets

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 $Y \in \mathbb{R}^{N\times 1}$:

$$Y = \big[\, y(\tau+1) \; \cdots \; y(N_S) \,\big]^T, \tag{6}$$

which contains the observations $y(t)$ stacked in row, $N = N_S - \tau$ is the number of outputs that it is possible to employ for the identification stage, given $\tau = \max(r, q)$.

In the same way, it is possible to construct the N *supervised* regressors $\varphi_S(t) \in \mathbb{R}^{k\times 1}$, for $\tau \le t \le N_S - 1$, as:

$$\varphi_S(t) = \big[y(t), \cdots, y(t-r+1), \; u_S(t), \cdots, u_S(t-q+1)\big]^T. \tag{7}$$

The features' matrix $G \in \mathbb{R}^{N\times m}$ can be defined by stacking all the features computed from the supervised regressors (7), leading to:

$$G = \begin{bmatrix} \gamma^T\Big(\varphi_S(\tau)\Big) \\ \vdots \\ \gamma^T\Big(\varphi_S(N_S-1)\Big) \end{bmatrix}. \tag{8}$$

It is possible to construct the model's regressors as in (7), by leveraging also on $\mathcal{D}_U$, the dataset that contains artificial inputs and outputs. There are, therefore, $N_{rU} = N_U - \tau$ available *unsupervised* regressors $\varphi_U(t) \in \mathbb{R}^{k\times 1}$, each one of them defined, for $\tau \le t \le N_U - 1$, as:

$$\varphi_U(t) = \big[y_U(t), \cdots, y_U(t-r+1), \; u_U(t), \cdots, u_U(t-q+1)\big]^T. \tag{9}$$

The (unsupervised) features' matrix $G_U \in \mathbb{R}^{N_{rU}\times m}$ groups all these unsupervised regressors, as:

$$G_U = \begin{bmatrix} \gamma^T\Big(\varphi_U(\tau)\Big) \\ \vdots \\ \gamma^T\Big(\varphi_U(N_U-1)\Big) \end{bmatrix}. \tag{10}$$

Combining the available information, we can define the joint matrix, containing both supervised (7) and unsupervised (9) regressors, as:

$$\widetilde{G} = \begin{bmatrix} G \\ G_U \end{bmatrix}, \tag{11}$$

where $\widetilde{G} \in \mathbb{R}^{N_r \times m}$ 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 Y will be denoted as $y(i)$.

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

3. MANIFOLD REGULARIZATION

The dataset $\mathcal{D}_U$ can be useful for discovering the relation between inputs and outputs if the marginal probability density $p(\varphi)$, which generated the inputs, happens to be informative about the conditional distribution $p(y|\varphi)$, see Belkin et al. (2006). We can make this possible by stating a specific assumption about the connection between the marginal and the conditional distributions, Chapelle et al. (2010):

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 behaviour 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. A possible choice for the manifold regularization term, which enforces Assumption 1, has been advocated in Belkin et al. (2006) as:

$$S_g = \int_{\mathcal{G}} \|\nabla \cdot g\|^2 \, dp(\varphi) = \int_{\mathcal{G}} g \cdot \Delta \cdot g \, dp(\varphi), \tag{12}$$

where ∇ and Δ are the gradient and the Laplace-Beltrami operators along the manifold $\mathcal{G}$, respectively, 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 the gradient of g, and so S_g, must be small when Assumption 1 holds. Then, minimizing S_g with respect to model's parameters is a way to enforce Assumption 1.

The information about the input distribution $p(\varphi)$ is rarely used because, most of the times, $p(\varphi)$, and $\mathcal{G}$, are unknown. It turns out that S_g can be approximated using the *regressor graph*, Belkin et al. (2005, 2006). This is a weighted complete graph with the (supervised and unsupervised) regressors as its vertexes, and the weight of each edge defined as:

$$w_{i,j} = e^{-\frac{\|\varphi(i)-\varphi(j)\|^2}{2{\sigma_e}^2}}, \tag{13}$$

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 (13) indicates that two regressors are similar. This rationale derives from a manifold learning algorithm called Laplacian Eigenmaps, Belkin and Niyogi (2003). It can be shown that, Belkin et al. (2006):

$$S_g \simeq \widetilde{Y}^T \cdot L \cdot \widetilde{Y}, \tag{14}$$

where $\widetilde{Y} = \Big[g\big(\varphi(1)\big), \cdots, g\big(\varphi(N_r)\big)\Big]^T \in \mathbb{R}^{N_r \times 1}$ contains the *noiseless* outputs, corresponding to *both supervised and unsupervised input regressors*. In order to obtain the approximation of S_g (14), *only the 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.

4. SYNTHETIC DATASET GENERATION

In this work, we adopted the following idea to generate the artificial data. The idea is to generate the synthetic regressors *in the neighbourhood* of the existing supervised ones. The choice is motivated by the fact that, following Assumption 1, if the system is smooth enough, similar regressors should have a similar corresponding output. We generate in this way p new datasets. Thus, the number of unsupervised regressors will be $N_{rU} = p \cdot N$. The total number of regressors data is $N_r = N + N_{rU} = (p+1) \cdot N$. The method is formalized as follows.

Let $\mathcal{D}_U$ be composed of p unsupervised datasets $\mathcal{D}_U^i$, $i = 1, \ldots, p$ as:

$$\mathcal{D}_U^i = \Big\{ \big(u_U^i(t), y_U^i(t)\big) \quad | \quad 1 \le t \le N_S \Big\}. \tag{15}$$

We define the following relations:

$$\begin{cases} u_U^i(t) = u_S(t) + v^i(t) \\ y_U^i(t) = y(t) + v^i(t), \end{cases} \tag{16}$$

where $v^i(t)$ is a random variable, that determines the distance of the p unsupervised points with respect to the supervised one. Thus, the synthetic data $\Big\{\big(u_U^i(t), y_U^i(t)\big)\Big\}$ form a "region" in the neighbourhood of a supervised regressor $\Big\{\big(u_S(t), y(t)\big)\Big\}$. A 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. We introduce a tuning parameter $\alpha \in \mathbb{R}$, allowing to regulate the regions' maximum area. In particular $\alpha = 1$ will 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:

$$v^i(t) \sim \mathrm{U}(-h, h) \quad 1 \le t \le N_S, \;\; i = 1, \ldots, p \tag{17}$$

where $h > 0$ determines the area of the unsupervised points regions. To impose distinct regions, the following inequalities must hold, for $\tau \le t \le N_S - 1$, $i = 1, \ldots, p$:

$$\big\|\varphi_U^i(t) - \varphi_S(t)\big\| \le \frac{d}{2}, \tag{18}$$

where d is the distance between the two closest supervised regressors. It can be shown that (18) can be written as:

$$\sum_{j=1}^{k} \left(v^i\left(t-j+1\right)\right)^2 \leq \left(\frac{d}{2}\right)^2 \quad \tau \leq t \leq N_S - 1, \tag{19}$$

Since $\left|v^i\left(t-j+1\right)\right| \leq h$ (it is generated from the random variable (17)), the inequalities (19) hold if:

$$\sum_{j=1}^{k} h^2 \leq \left(\frac{d}{2}\right)^2. \tag{20}$$

Recalling that $h \geq 0$, we have that (20) corresponds to:

$$h \leq \frac{d}{2\sqrt{k}}. \tag{21}$$

The condition described in (21) 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:

$$h = \frac{d}{2\alpha\sqrt{k}} = \frac{d}{2\alpha\sqrt{r+q}}. \tag{22}$$

Having defined the random variable $v^i(t)$, it is possible to define the N_{rU} unsupervised regressors $\varphi_U^i(t) \in \mathbb{R}^{m\times 1}$, $i = 1, \ldots, p$ according to (9), for $\tau \leq t \leq N_S - 1$:

$$\begin{aligned}\varphi_U^i\left(t\right) = \big[&y_U^i\left(t\right), \cdots, y_U^i\left(t-r+1\right), \\ &u_U^i\left(t\right), \cdots, u_U^i\left(t-q+1\right)\big]^T.\end{aligned} \tag{23}$$

Using the unsupervised regressors computed in (23) using the i-th dataset, it is possible to form the i-th unsupervised features' matrix $G_U^i \in \mathbb{R}^{N\times m}$ as in (10):

$$G_U^i = \begin{bmatrix} \gamma^T\Big(\varphi_U^i\left(\tau\right)\Big) \\ \vdots \\ \gamma^T\Big(\varphi_U^i\left(N_S-1\right)\Big) \end{bmatrix}. \tag{24}$$

The global (unsupervised) features' matrix $G_U \in \mathbb{R}^{N_{rU}\times m}$ can be therefore composed by stacking the matrices (24), $i = 1, \ldots, p$:

$$G_U = \begin{bmatrix} G_U^i \\ \vdots \\ G_U^p \end{bmatrix}. \tag{25}$$

5. PARAMETRIC APPROACH

5.1 Parameters identification

In order to identify the true NARX system $\mathcal{S}$ in (1), the classical Prediction Error Method (PEM) is used, Ljung (1998). The predictor for (1) can be simply found as:

$$\widehat{y}\left(t|t-1;\vartheta\right) = \vartheta^T \cdot \gamma\big(\varphi\left(t\right)\big) \tag{26}$$

The value of the parameters' is then estimated by minimizing the variance of the prediction error by:

$$J\left(\vartheta\right) = \sum_{t=1}^{N} \Big(y\left(t\right) - \vartheta^T \cdot \gamma\big(\varphi\left(t\right)\Big)^2. \tag{27}$$

Since (27) could be ill-conditioned, a regularization term is usally employed, leading as an example to the Ridge regression problem, Friedman et al. (2001):

$$J\left(\vartheta\right) = \sum_{t=1}^{N} \Big(y\left(t\right) - \vartheta^T \cdot \gamma\big(\varphi\left(t\right)\Big)^2 + \lambda \cdot \vartheta^T\vartheta, \tag{28}$$

where $\lambda > 0$ controls the regularization strength. By expressing (28) in matrix notation, we can write the cost function as:

$$J\left(\vartheta\right) = \left\|Y - G\cdot\vartheta\right\|_2^2 + \lambda \cdot \left\|\vartheta\right\|_2^2. \tag{29}$$

The semi-supervised approach pursued in this paper consists into employing the manifold regularization term (14), in order to include information about the local smoothness of the function (leveraging on the unsupervised data points). This leads to the following cost function:

$$J_M\left(\vartheta\right) = \left\|Y - G\cdot\vartheta\right\|_2^2 + \lambda_M \cdot \widetilde{Y}^T \cdot L \cdot \widetilde{Y}, \tag{30}$$

where $\lambda_M > 0$ has the same meaning of λ.

Since $\widetilde{Y} = \widetilde{G}\cdot\vartheta$, it holds that (14) can be rewritten as $\widetilde{Y}^T \cdot L \cdot \widetilde{Y} = \vartheta^T\widetilde{G}^T \cdot L \cdot \widetilde{G}\vartheta$. Thus, (30) assumes the form:

$$J_M\left(\vartheta\right) = \left\|Y - G\cdot\vartheta\right\|_2^2 + \lambda_M \cdot \vartheta^T\widetilde{G}^T \cdot L \cdot \widetilde{G}\vartheta. \tag{31}$$

Since (31) is the sum of two quadratic terms, it admits a closed-form solution, that is:

$$\widehat{\vartheta} = \left[G^TG + \lambda_M \cdot \widetilde{G}^T \cdot L \cdot \widetilde{G}\right]^{-1} \cdot G^TY, \tag{32}$$

where $\widehat{\vartheta} \in \mathbb{R}^{m\times 1}$ represents the parameters' estimate.

5.2 Effective degrees of freedom

With the parameter's estimate $\widehat{\vartheta}$, it is trivial to compute the model's predictions as $\widehat{Y} = G\cdot\widehat{\vartheta}$, with $\widehat{Y} \in \mathbb{R}^{N\times 1}$. By expanding the definition of $\widehat{\vartheta}$, we can write:

$$\widehat{Y} = G \cdot \left[G^TG + \lambda_M \cdot \widetilde{G}^T \cdot L \cdot \widetilde{G}\right]^{-1} \cdot G^TY = H \cdot Y, \tag{33}$$

with $H \in \mathbb{R}^{N\times N}$. Following Friedman et al. (2001), the number of effective degrees of freedom of a linear model, such that $\widehat{Y} = H \cdot Y$, can be found as:

$$\nu = \mathrm{Tr}\left(H\right). \tag{34}$$

A similar expression can be obtained for Ridge regression. When $\lambda_M = 0$ or $\lambda = 0$, ν is equal to the number of parameters of the linear model. The quantity in (34) will be used to compare the application of the Ridge and the manifold regularization terms. It is interesting to notice that ν can be used to efficiently compute an estimate of the Leave One Out Cross-Validation (LOOCV) validation score, see Friedman et al. (2001).

6. RESULTS

This section provides numerical examples in order to compare the SSI method with Ridge regression. We tested the proposed method on two nonlinear system, following the results in Pillonetto et al. (2014):

1) $y(t) = 0.5y(t-1) - 0.05y(t-2)^2 + u(t-1)^2 + 0.8u(t-2) + e(t)$
   $e(t) \sim \mathrm{WN}(0,1)$, $u(t) \sim \mathrm{WN}(0, 0.25)$

$$2)\ \ y(t) = 0.8y(t-1) + u(t-1) - 0.3t(t-1)^3 + 0.25u(t-1)u(t-2) - 0.3u(t-2) + 0.24u(t-2)^3 - 0.2u(t-2)u(t-3) - 0.4u(t-3) + e(t)$$
$$e(t) \sim \text{WN}(0,1),\ u(t) \sim \text{WN}(0,1)$$

The system 1) has 4 parameters, while the system 2) presents 8 coefficients. The supervised dataset $\mathcal{D}_S$, generated from each systems, consists of $N_S = 20$ measures. Thus, the problem is ill-conditioned and suited to test regularization approaches. The unsupervised dataset $\mathcal{D}_U$ is generated according to Section 4. The hyperparameter of the Ridge regression method (29) is λ, whereas the hyperparameters of the manifold regression method (30) are λ_M, σ_e (the shape parameter of the Laplacian Eigenmaps), α (the parameter controlling the area of the generated unsupervised points) and p, i.e., the number of additional unsupervised datasets. In our experiments, we fixed $p = 3$.

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 $\mathcal{D}_S$. This has been done in order to assess the method capability and value. 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 NMAE (Normalized Mean Absolute Error) metric:

$$NMAE = \frac{\sum_{t=1}^{N_T} \left| \widehat{y}(t) - y_T(t) \right|}{\sum_{t=1}^{N_T} \left| y_T(t) - \bar{y}_T \right|}, \qquad (35)$$

where $\widehat{y}(t)$ is the predicted test output in correspondence of a test regressor, $y_T(t)$ is the true test output, and $\bar{y}_T$ is the mean value of the test outputs. A Monte Carlo simulation has been carried out to show the statistical significance of the proposed methodology, using $N_M = 1000$ runs. At each run, a different generation of the random noise was considered. Table 1 reports the search space of the hyperparameters.

Table 1. Values of the tuning parameters for the parametric approach

Ridge regression	
λ	10000 log-spaced values in $[10^{-4}, 10^{6}]$
Semi-Supervised Identification	
λ_M	100 log-spaced values in $[10^{-2}, 10^{4}]$
σ_e	100 log-spaced values in $[10^{-2}, 10^{2}]$
α	10 evenly spaced values in $[1, 10]$

Results are depicted in Figure 1 and 2 for the numerical examples 1) and 2) respectively. The plots depict the comparison of Ridge and manifold regularization, in term of the error defined in (35). The manifold regularization shows some improvement on standard Ridge regression.

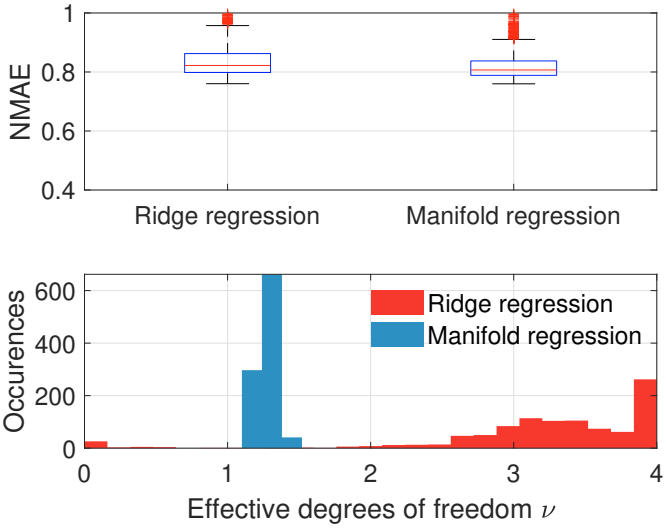

Fig. 1. Methods comparison for the system 1). Top) Plot of the NMAE for the Ridge and manifold regularization methods over 1000 runs. Bottom) Histogram of the effective degrees of freedom

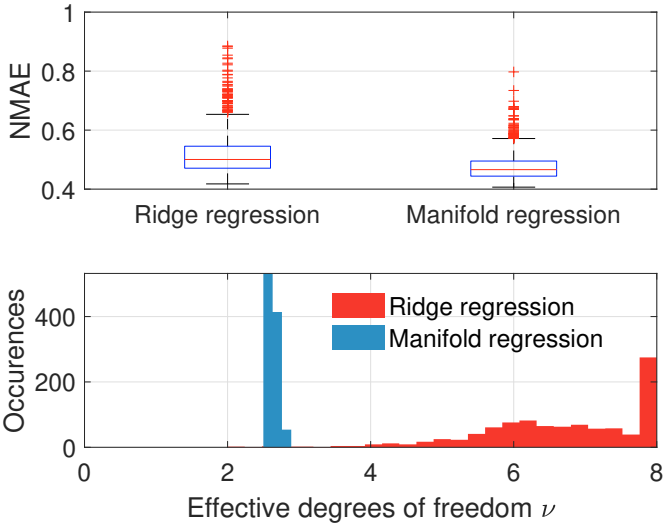

Fig. 2. Methods comparison for the system 2). Top) Plot of the NMAE for the Ridge and manifold regularization methods over 1000 runs. Bottom) Histogram of the effective degrees of freedom

This behaviour can be better understood by looking at the degrees of freedom ν. As it is possible to observe, the manifold regression achieves overall a lower complexity, compared to Ridge regression. This latter method shows also a larger spread over the 1000 simulation runs. The manifold regression, instead, estimates models with almost the same complexity. Notice that this complexity is coherent with the heuristic of having a model complexity such as $N > 10 \cdot \nu$, condition which is almost always respected when the manifold regularization is employed, see Abu-Mostafa et al. (2012). Figures from 3 to 5 show the variation of the degrees of freedom with λ_M, for the different hyperparameters values, considering the numerical example 2). It can be seen how, as p gets larger, less regularization is needed: this is because the unsupervised data acts as a regularizer. When α increases, the unsupervised points are closer to the supervised ones:

this makes the region for which we apply the smoothness assumption smaller, and therefore more regularization is needed. The effect of increasing σ_e consists into considering all points as "close together", thereby leading to a fitted function which is equal to the mean of the measured outputs. Thus, less regularization is required.

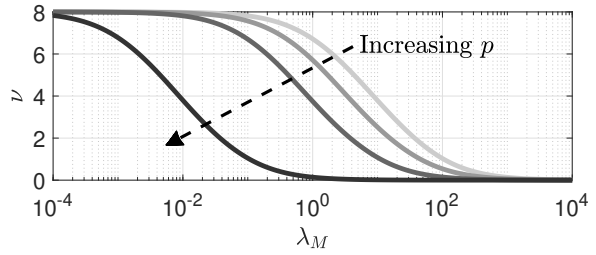

Fig. 3. Sensitivity analysis of the hyperparameters, $p = \{3, 5, 10, 100\}$, $\sigma_e = 0.5$, $\alpha = 1$

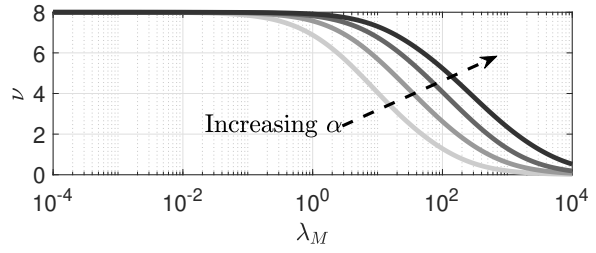

Fig. 4. Sensitivity analysis of the hyperparameters, $p = 3$, $\sigma_e = 0.5$, $\alpha = \{1, 2, 5, 10\}$

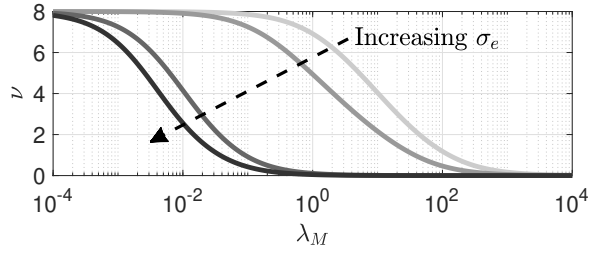

Fig. 5. Sensitivity analysis of the hyperparameters, $p = 3$, $\sigma_e = \{0.5, 1, 5, 50\}$, $\alpha = 1$

7. CONCLUSIONS AND FUTURE DEVELOPMENTS

In this paper, we presented a method for learning nonlinear dynamical system by employing additional synthetic data. The additional dataset is generated by perturbing the measured regressors. In order to leverage such information, the framework of Semi-Supervised Learning is used. Following this idea, we derived a new regularization term, that makes use of how the regressors are distributed in the regressor space. We tested the approach in a parametric setting on examples found in the related literature, comparing the proposed regularization with the Ridge one. Results shown that the method achieved better results on numerical experiments. Future research is devoted to define a hyperparameters' selection method, evaluating the effect of the number of additional datasets and of the hyperparameters on the model performance, and a comparison with more advanced parametric regularization techniques.

REFERENCES

Abu-Mostafa, Y.S. (1994). Learning from hints. *Journal of Complexity*, 10(1), 165–178.

Abu-Mostafa, Y.S., Magdon-Ismail, M., and Lin, H.T. (2012). *Learning from data*, volume 4. AMLBook New York, NY, USA:.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 1373–1396.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov), 2399–2434.

Belkin, M., Niyogi, P., and Sindhwani, V. (2005). On manifold regularization. In *AISTATS*, 1.

Chapelle, O., Schlkopf, B., and Zien, A. (2010). *Semi-Supervised Learning*. The MIT Press, 1st edition.

Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. (2001). Vicinal risk minimization. In *Advances in neural information processing systems*, 416–422.

Cobelli, C., Dalla Man, C., Sparacino, G., Magni, L., De Nicolao, G., and Kovatchev, B.P. (2009). Diabetes: models, signals, and control. *IEEE reviews in biomedical engineering*, 2, 54–96.

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1), 1–50.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25*, 1097–1105. Curran Associates, Inc.

Ljung, L. (1998). System identification. In *Signal analysis and prediction*, 163–173. Springer.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3), 657–682.

Rasmussen, C.E. and Williams, C.K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Svensson, A. and Schön, T.B. (2017). A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80, 189–199.

Vajda, S. (2014). *Probabilistic programming*. Academic Press.

Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Yang, J., Wei, H.L., Kadirkamanathan, V., and Lin, X. (2012). System identification from small data sets using an output jittering method with application to model estimation of bioethanol production. In *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on*, volume 3, 949–955. IEEE.

Zhang, H., Cisse, M., Dauphin, Y.N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhu, X. (2011). Semi-supervised learning. In *Encyclopedia of machine learning*, 892–897. Springer.