

A comparison of manifold regularization approaches for kernel-based system identification

M. Mazzoleni*, M. Scandella*, 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).*

Abstract: In this paper, we present a simulation study to investigate the role of manifold regularization in kernel-based approaches for nonparametric nonlinear SISO (Single-Input Single-Output) system identification. This problem is tackled as the estimation of a static nonlinear function that maps regressors (that contain past values of both input and output of the dynamic system) to the system outputs. Manifold regularization, as opposite to the Tikhonov one, enforces a local smoothing constraint on the estimated function. It is based on the assumption that the regressors lie on a manifold in the regressors space. This manifold is usually approximated with a weighted graph that connects the regressors. The present work analyzes the performance of kernel-based methods estimates when different choices are made for the graph connections and their respective weights. The approach is tested on benchmark nonlinear systems models, for different connections and weights strategies. Results give an intuition about the most promising choices in order to adopt manifold regularization for system identification.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Kernel methods; System Identification; Manifold regularization

1. INTRODUCTION

Kernel methods and regularized approaches are nowadays predominant approaches in both linear and nonlinear black-box system identification. Due to their ability to trade bias and variance of the estimated models, they quickly showed improved performance with respect to traditional Prediction Error Methods (PEM), see Pillonetto et al. (2014). In kernel methods, the estimation problem is casted in a nonparametric way: the aim is to find directly the (possibly nonlinear) function that best matches input/output data, as opposite to identify an a-priori fixed number of model parameters. The estimated function is searched within a Reproducing Kernel Hilbert Space (RKHS), see Aronszajn (1950). The *kernel function* (or simply kernel) is in a 1:1 relation with the RKHS, and determines the properties of the functions inside the functional space.

Kernel methods have been applied mostly to linear discrete-time systems, see Ljung et al. (2019); Chen et al. (2012), where the developed methods can be translated similarly to the continuous-time setup. In this linear case, the unknown function to be estimated is the impulse response of the system, as reviewed in Pillonetto et al. (2014). In the nonlinear case, the aim is to learn the map from the regressors vector (with predefined exogenous and autoregressive orders) to the system output, as done in Pillonetto et al. (2011).

The main reason for such popularity of kernel methods lies in their regularized nature. In their most common formulation, this equals to a *Tikhonov-like* regularization

term that penalizes too complex functions. By trading data fit and solution complexity in a *continuous way*, better results can be achieved than employing complexity criteria such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) for model orders selection, see Pillonetto et al. (2011, 2014).

Manifold regularization is a different type of regularization term that, as opposite to the Tikhonov one that strives for *global smoothness*, enforces *local smoothness* on the estimated function. Manifold regularization is usually employed in nonlinear estimation problems. In this setting, this regularization strategy assumes that the regressors lie in a manifold of the regressor space. The rationale is based on the *smoothness assumption* that nearby regressors should have a similar corresponding output. Then, learning the manifold (i.e. the distribution of the regressors) can give additional information about the behaviour of the unknown function that has to be estimated. In dynamical models, due to the intrinsic correlation among the regressors components, their distribution is not uniform in the regressors space. Instead, the regressors are likely to lie on a certain manifold. Recently, manifold regularization has been employed for nonlinear system identification with kernel methods in Mazzoleni et al. (2018a,b); Formentin et al. (2019).

Since the regressors manifold, if present, is usually unknown, it is approximated via a graph that connects the regressors (the nodes of the graph) through weighted connections (the edges of the graph). In order to build the regressors graph, it is necessary to define its topology (i.e. the connections between its nodes) and the weights on

the edges. Furthermore, based on these information, the manifold regularization term can be chosen with different rationales. In the literature, there are two main approaches to graph selection for manifold reconstruction:

- (1) employ a completely connected graph with different edges weights;
- (2) use a non-completely connected graph with the same weight on all the edges.

The first category includes simple Gaussian weights, see Belkin and Niyogi (2003), the diffusion maps rationale of Coifman and Lafon (2006) and their generalizations developed in Berry and Harlim (2016); Berry and Sauer (2016). The main rationales of the second category are the fixed ε -ball and the K -NN connections, see Berry and Sauer (2019), where each regressor is connected to all the regressors inside a sphere with radius ε or to their K Nearest Neighbour regressors, respectively. In this case, no specific weights are given and they are all set to one. Furthermore, the authors in Formentin et al. (2019) introduced a connection scheme called *dynamic connections*, that is tailored to dynamical systems. It is interesting to remark that regressors graphs can be also learned directly from data, see Mateos et al. (2019); Dong et al. (2018).

As far as the authors are aware, there is still no clear indication or intuition about what choices of regressors graph are best for nonlinear kernel-based system identification with manifold regularization. The *aim of this paper is to move a first step in this direction*. In the detail, we evaluate and compare:

- Two types of weighting schemes: (i) Gaussian weights, Belkin and Niyogi (2003); Belkin et al. (2006); (ii) weights based on the employed kernel function.
- Three types of connection strategies, i.e. (i) fully-connected; (ii) dynamic connections of Formentin et al. (2019); (iii) the ε -ball scheme of Berry and Sauer (2019).
- Two types of manifold regularization terms: (i) the Laplacian EigenMaps (LEM) of Belkin and Niyogi (2003) and (ii) the Local Linear Embedding (LLE) scheme, see Roweis and Saul (2000).

The different settings are compared on four different nonlinear dynamical systems taken from the literature. Simulation results give a first indication of the best choices of the manifold regularization tuning knobs when applied to nonlinear system identification with kernel methods.

The remainder of the paper is organized as follows. Section 2 reviews the formulation of kernel-based nonlinear system identification problems with manifold regularization. Section 3 describes the choices considered for the construction of the regressor graph. Section 4 shows simulation results on benchmark nonlinear system identification problems. Section 5 is then devoted to some concluding remarks.

Notation

Lower case bold letters \mathbf{x} indicate vectors. Bold upper case letters \mathbf{X} indicate matrices. The identity matrix is indicated with $I_n \in \mathbb{R}^{n \times n}$. The covariance between two random variables \mathbf{x}_r and \mathbf{x}_s is $\text{cov}\{\mathbf{x}_r, \mathbf{x}_s\}$, with $\mathbf{x}_r, \mathbf{x}_s \in$

$\mathcal{X} \subset \mathbb{R}^{d \times 1}$. A white noise signal e_t with zero mean and variance β^2 is indicated with $e_t \sim \text{WN}(0, \beta^2)$.

2. KERNEL-BASED SYSTEM IDENTIFICATION WITH MANIFOLD REGULARIZATION

2.1 Problem statement

Let f be a generic, possibly nonlinear, mapping $f: \mathcal{X} \rightarrow \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^{d \times 1}$, such that

$$y_t = f(\mathbf{x}_t) + e_t, \quad (1)$$

where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \mathbb{R}$ are, respectively, the system input regressor and output at time $t \in \mathbb{Z}_{\geq 0}$, and $e_t \sim \text{WN}(0, \beta^2)$ is an additive white noise. The regressor \mathbf{x}_t could contain past samples of both input u_t and output y_t of the SISO dynamical system that generates the data. In this context, $f(\mathbf{x}_t)$ is the one-step ahead predictor $\hat{y}_{t|t-1}$ and e_t is the one-step ahead prediction error. Suppose that we have n observations of regressor-output data $\mathcal{D} = \{\mathbf{x}_t, y_t\}_{t=1}^n$. The aim is to obtain an estimate \hat{f} of the unknown regressor-output mapping f using \mathcal{D} .

Kernel methods with Tikhonov regularization look for the estimate \hat{f} by solving the variational problem

$$\hat{f} = \arg \min_{f \in \mathcal{H}} \sum_{t=1}^n (y_t - f(\mathbf{x}_t))^2 + \tau \cdot \|f\|_{\mathcal{H}}^2 \quad (2)$$

where $\tau \in \mathbb{R}_{>0}$ and $\mu \in \mathbb{R}_{>0}$ are constant values (called *hyperparameters*) and \mathcal{H} is a Reproducing Kernel Hilbert Space (RKHS) characterized by the *kernel function* $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, that depends on some hyperparameters $\psi \in \mathbb{R}^{q \times 1}$. The second term in (2) is the Tikhonov regularization term, that penalizes too complex functions.

The Tikhonov regularization term in (2) imposes a global smoothness behaviour on the estimated function. It is possible to define other penalty terms based on some *different undesired behaviors* of the estimated function. As motivation for this, consider again the formulation (2). Here, the kernel k defines both \mathcal{H} and the Tikhonov regularizer $\|f\|_{\mathcal{H}}^2$. However, sometimes it could be desirable to use the space \mathcal{H} , *but not the norm* $\|\cdot\|_{\mathcal{H}}$ as complexity penalty term. In this view, *manifold regularization* can be employed to enforce different regularity properties on the estimate (e.g. local instead of global smoothness).

2.2 Manifold regularization

The aim of manifold regularization is to exploit the geometry of the regressors in order to provide prior information on f . In the standard learning paradigm, the outputs y_t are supposed to be generated, given a regressor \mathbf{x}_t , as $y_t \sim p_{y|\mathbf{x}}(y|\mathbf{x} = \mathbf{x}_t)$ with \mathbf{x} denoting a generic regressor variable. In this setting, the aim is to find a good approximation of the conditional distribution $p_{y|\mathbf{x}}$. For this reason, usually, the marginal distribution $p_{\mathbf{x}}$, that determines how the regressors are sampled, is ignored because it does not contain any useful information about $p_{y|\mathbf{x}}$. The knowledge of $p_{\mathbf{x}}$ can be useful if a specific assumption is made about the connection between the marginal and the conditional distributions. A typical assumption is the following, see Belkin et al. (2006):

Assumption 1. (Smoothness assumption). The distribution $p_{y|\mathbf{x}}(y|\mathbf{x})$ varies smoothly alongside the intrinsic geometry of $p_{\mathbf{x}}$.

Assumption 1 basically tells that, if two regressors $\mathbf{x}_r, \mathbf{x}_s$ are close in space, than so should be their corresponding outputs $f(\mathbf{x}_r), f(\mathbf{x}_s)$. Now, given a function f , we can define a new term $\|f\|_{\mathcal{I}}^2$, called *intrinsic regularizer*, that is higher when f is not smooth alongside $p_{\mathbf{x}}$. In the literature, there are valid choices for the definition of $\|f\|_{\mathcal{I}}^2$, see Belkin et al. (2006). However, one of the most used definitions assumes that the support \mathcal{X} of $p_{\mathbf{x}}$ is a compact manifold. In this case, one of the possible intrinsic regularizers, based on the concept of manifold, is

$$\|f\|_{\mathcal{I}}^2 = \int_{\mathcal{X}} \|\nabla f(\mathbf{x})\|_2^2 p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{X}} f(\mathbf{x}) \Delta f(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (3)$$

where ∇ and Δ are the gradient and the Laplace-Beltrami operators along the manifold \mathcal{X} , respectively. Minimizing (3) is a way to leverage Assumption 1. This term penalizes functions that exhibits a high gradient with a weight that depends on $p_{\mathbf{x}}$. Therefore, it penalizes functions that have high variation in the high probability regions, but it allows large swings of the function in the other regions. In other words, this intrinsic regularizer promotes functions that are *smooth locally* in the high density regions of the regressors space, even if they are not smooth globally in all the domain \mathcal{X} .

Remark 1. Since this type of intrinsic regularizer is defined on a manifold, this method is often called *manifold regularization*.

Unfortunately, the term in (3) can not be computed since the manifold structure and $p_{\mathbf{x}}$ are usually unknown. A common approach to approximate the manifold is to use a *regressors graph* \mathcal{G} , i.e. a graph with the following properties: (i) each vertex of the graph is associated with a regressor \mathbf{x}_r , $r = 1, \dots, n$; (ii) the weight ω_{rs} of the edge between the vertices r and s represents the degree of proximity between \mathbf{x}_r and \mathbf{x}_s in the intrinsic geometry of $p_{\mathbf{x}}$; (iii) if the edge between the vertices r and s is missing, then \mathbf{x}_r and \mathbf{x}_s are not considered neighbours.

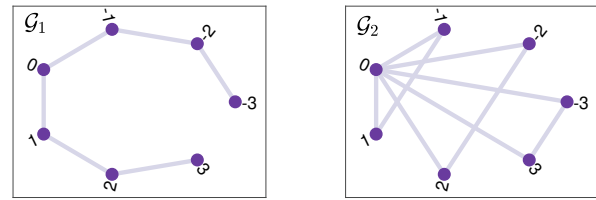
It can be shown, see Hein et al. (2005), that the regularization term (3) can be approximated as

$$\|f\|_{\mathcal{I}}^2 \approx \sum_{r=1}^n \sum_{s=1}^n \omega_{rs} \cdot [f(\mathbf{x}_r) - f(\mathbf{x}_s)]^2 = \mathbf{f}^{\top} \mathbf{L} \mathbf{f}, \quad (4)$$

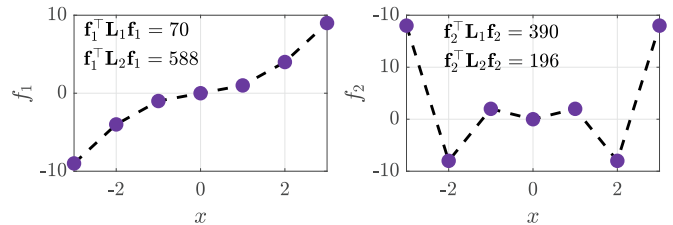
where $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^{\top} \in \mathbb{R}^{n \times 1}$ is the vector of noiseless function evaluations at measured regressors points $\mathbf{x}_t, t = 1, \dots, n$, and $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the graph Laplacian. The Laplacian of the regressors graph can be computed as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, with $\mathbf{W} \in \mathbb{R}^{n \times n}$ the weighted adjacency matrix of the regressors graph, i.e. the element (r, s) of \mathbf{W} is the weight ω_{rs} , and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose r -th diagonal element is $d_{rr} = \sum_{s=1}^n \omega_{rs}$.

The term in (4) is lower when the function f is *smooth along the graph*. In order to better understand the role of (4), consider the following example.

Example 1. Consider the two regressors graphs in Fig. 1a, and the regressors set $\mathcal{D}_{\mathbf{x}} = \{-3, -2, -1, 0, 1, 2, 3\} \subseteq \mathbb{R}$. The graph \mathcal{G}_1 connects the regressors with their respective



(a) The regressors graphs \mathcal{G}_1 and \mathcal{G}_2 used in Example 1. If the edge is not drawn, then its weight is 0, otherwise it is 1.



(b) The functions f_1 and f_2 used in Example 1.

Fig. 1. Illustration of the manifold regularization term behaviour, on different graphs topologies and functions defined over them.

neighbors, while the graph \mathcal{G}_2 connects each regressor with 0 and with the one with the opposite sign. The Laplacian matrix defined by the first graph is $\mathbf{L}_1 \in \mathbb{R}^{7 \times 7}$ and the one defined using the second graph is $\mathbf{L}_2 \in \mathbb{R}^{7 \times 7}$. Now, consider the two functions shown in Fig. 1b. The first function f_1 is smooth along the real axis, while the second one f_2 has an erratic behavior. The value of the intrinsic regularizer of each function evaluated on each graph is

$$\text{Graph } \mathcal{G}_1: \quad \mathbf{f}_1^{\top} \mathbf{L}_1 \mathbf{f}_1 = 70 \quad \mathbf{f}_2^{\top} \mathbf{L}_1 \mathbf{f}_2 = 390 \quad (5)$$

$$\text{Graph } \mathcal{G}_2: \quad \mathbf{f}_1^{\top} \mathbf{L}_2 \mathbf{f}_1 = 588 \quad \mathbf{f}_2^{\top} \mathbf{L}_2 \mathbf{f}_2 = 196 \quad (6)$$

where $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{R}^{7 \times 1}$ contain the evaluations at regressors points of the functions f_1 and f_2 , respectively. According to the graph \mathcal{G}_1 , the smoothest function is f_1 , while, according to \mathcal{G}_2 , f_2 is smoother. Thus, *the graph topology determines which regressor is considered close to another one*, and thus which function is considered smooth. So, the definition of the graph properties is an important factor for the meaning of the manifold regularization term. \square

Example 1 showed how the regressors graphs connections and weights depend on user choices, that determine what is considered “smooth” and in which manner. In this work, *we will compare different choices of graph definitions* and apply them to nonparametric nonlinear system identification. We indicate with $\gamma \in \mathbb{R}^{q_m \times 1}$ the hyper-parameters needed for the construction of the graph.

2.3 Kernel methods with manifold regularization

Following the previous section, it is possible to add the manifold regularization term to the cost function (2), obtaining

$$\hat{f} = \arg \min_{f \in \mathcal{H}} \sum_{t=1}^n (y_t - f(\mathbf{x}_t))^2 + \tau \cdot \|f\|_{\mathcal{H}}^2 + \mu \cdot \mathbf{f}^{\top} \mathbf{L} \mathbf{f}, \quad (7)$$

where $\mu \in \mathbb{R}_{>0}$ determines the manifold regularization strength. The solution to (7) is given by the *representer*

theorem, see Kimeldorf and Wahba (1971); Belkin et al. (2006), and it reads as

$$\hat{f}(\mathbf{x}) = \sum_{t=1}^n \hat{c}_t k(\mathbf{x}_t, \mathbf{x}), \quad (8)$$

with $\hat{\mathbf{c}} = [\hat{c}_1, \dots, \hat{c}_n]^\top \in \mathbb{R}^{n \times 1}$ given by

$$\hat{\mathbf{c}} = (\mathbf{K} + \tau I_n + \mu \mathbf{L} \mathbf{K})^{-1} \mathbf{y}. \quad (9)$$

where $\mathbf{y} \in \mathbb{R}^{n \times 1}$ contains the available output measurements $y_t, t = 1, \dots, n$, and $\mathbf{K} \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix (called kernel or Gram matrix) such that $\mathbf{K}_{rs} = k(\mathbf{x}_r, \mathbf{x}_s)$.

In the following, we will indicate the hyperparameters vector of the method with $\boldsymbol{\theta} = [\boldsymbol{\psi}^\top \ \tau \ \mu \ \boldsymbol{\gamma}^\top]^\top \in \mathbb{R}^{q \times 1}$.

3. DEFINITION OF THE REGRESSORS GRAPH

In this section, we introduce the various choices for the construction of the regressors graph. We will compare different: (i) weights definitions; (ii) connections schemes and (iii) manifold regularization terms.

3.1 Choice of the graph weights

We will consider two formulations of the graph weights: (i) Gaussian weights and (ii) weights defined by the kernel function.

Gaussian weights The use of Gaussian weights is advocated in Belkin and Niyogi (2003); Belkin et al. (2006) as a meaningful choice for the approximation of (3). This choice weights more regressors closer *in space*, with respect to those that are far apart in the regressors domain. Thus, the weight of the (r, s) edge, with $\boldsymbol{\gamma} = \sigma_m \in \mathbb{R}_{>0}$, is

$$\omega_{rs}^g = \exp \left[- \left(\frac{\|\mathbf{x}_r - \mathbf{x}_s\|_2^2}{\sigma_m} \right) \right]. \quad (10)$$

Kernel weights The rationale of this choice is to weight each connection (r, s) with the covariance between the regressors \mathbf{x}_r and \mathbf{x}_s , i.e. $\text{cov}\{\mathbf{x}_r, \mathbf{x}_s\} = k(\mathbf{x}_r, \mathbf{x}_s)$. This choice weights more correlated regressors, and gives low weight to uncorrelated ones. Notice that $k(\cdot, \cdot)$ is the same kernel used to define the RKHS \mathcal{H} , where \hat{f} lies. Thus, the weight of the (r, s) edge, with $\boldsymbol{\gamma} = \boldsymbol{\psi}$, is

$$\omega_{rs}^k = k(\mathbf{x}_r, \mathbf{x}_s). \quad (11)$$

3.2 Choice of the graph connections

We consider the following connections schemes: (i) fully-connected graph; (ii) dynamic connections; (iii) ε -ball connections.

Fully-connected graph In this straightforward case, all possible connections between the graph nodes, i.e. the regressors, are imposed on the graph. In this case, $\boldsymbol{\gamma} = \emptyset$.

Dynamic connections In this rationale, see Formentin et al. (2019), the regressors are connected based on their *temporal order*. Therefore, a regressor \mathbf{x}_r can be connected to its $\tilde{p} \in \mathbb{N}$ previous or subsequent regressors *in time*. Practically, we connect \mathbf{x}_r with $\mathbf{x}_{r-1}, \dots, \mathbf{x}_{r-\tilde{p}}$ and with

$\mathbf{x}_{r+1}, \dots, \mathbf{x}_{r+\tilde{p}}$. The rationale of the method is to not to constrain the outputs of the model to be similar when two regressors are not correlated, i.e. when the memory of the dynamical system has already faded. In this case, $\boldsymbol{\gamma} = \tilde{p}$.

ε -ball connections In this method, a regressor \mathbf{x}_r is connected to all other regressors such that $\text{dist}(\mathbf{x}_r, \mathbf{x}_s)^2 \leq \varepsilon$, where $\varepsilon \in \mathbb{R}_{>0}$ and dist is a distance measure such as the Euclidean norm. We focus on the algorithm recently developed in Berry and Sauer (2019), where ε is not fixed but depends on the regressors, i.e. $\varepsilon_r = \delta \sqrt{\text{dist}(\mathbf{x}_r, \mathbf{x}_r^K), \text{dist}(\mathbf{x}_r, \mathbf{x}_s^K)}$, where $\delta \in \mathbb{R}_{>0}$, \mathbf{x}_r^K , \mathbf{x}_s^K , $K \in \mathbb{N}$, are the K -th neighbours (in space) of \mathbf{x}_r and \mathbf{x}_s , respectively. In this case, $\boldsymbol{\gamma} = \{\delta, K\}$.

3.3 Choice of the manifold regularization terms

The structure of the regularization term in (4) is shared by many manifold learning methods, where \mathbf{L} is substituted by other symmetric matrices, see Cayton (2005). Such algorithms are still based on Assumption 1, but they formalize it from different perspectives. Here, we compare: (i) the LEM algorithm and (ii) the LLE algorithm.

Laplacian EigenMaps The LEM formulation, see Belkin and Niyogi (2003), is a manifold learning algorithm that leads to the manifold regularization term already introduced in (4), with \mathbf{L} being the Laplacian of the graph. In this case, $\boldsymbol{\gamma} = \sigma_m$.

Locally Linear Embedding The LLE formulation, see Roweis and Saul (2000), is based on the assumption that each data point and its neighbors lie on, or are close to, a locally linear patch of the function f . LLE tries to characterize the geometry of the local patches by finding the linear coefficients that reconstruct each point from its neighbors. In particular, these coefficients \tilde{w}_{rs} are computed minimizing the reconstruction error:

$$\begin{aligned} J(\tilde{\mathbf{W}}) &= \sum_{r=1}^n \left\| \mathbf{x}_r - \sum_{s=1}^n \tilde{w}_{rs} \cdot \mathbf{x}_s \right\|_2^2 \\ \text{s.t. } \sum_{s=1}^n \tilde{w}_{rs} &= 1; \quad t = 1 \dots n \\ \tilde{w}_{rs} &= 0; \quad r = 1 \dots n, \mathbf{x}_s \notin \mathcal{N}_r^K \end{aligned} \quad (12)$$

where \tilde{w}_{rs} is the weight of \mathbf{x}_s in reconstructing \mathbf{x}_r , $\tilde{\mathbf{W}} \in \mathbb{R}^{n \times n}$ is the matrix composed by all the weighting coefficients \tilde{w}_{rs} , and \mathcal{N}_r^K is the set composed by the K neighbours of \mathbf{x}_r . For this reason, the LLE algorithm is defined for K -NN connections type, where each regressor is connected to its K Nearest Neighbours in space. Each connection (r, s) is weighted by \tilde{w}_{rs} . Using $\tilde{\mathbf{W}}$, it is possible to compute an intrinsic regularization term, as conceived by the LLE manifold learning algorithm, as

$$\|f\|_{\text{LLE}}^2 \approx \sum_{r=1}^n \left(f(\mathbf{x}_r) - \sum_{s=1}^n \tilde{w}_{rs} \cdot f(\mathbf{x}_s) \right)^2 = \mathbf{f}^\top \mathbf{H} \mathbf{f}, \quad (13)$$

where $\mathbf{H} = I_n - \tilde{\mathbf{W}} - \tilde{\mathbf{W}}^\top + \tilde{\mathbf{W}}^\top \tilde{\mathbf{W}} \in \mathbb{R}^{n \times n}$. In this case, $\boldsymbol{\gamma} = K$.

4. SIMULATION RESULTS

4.1 Simulations setup

The different settings in Section 3 are applied to the identification of four nonlinear systems taken from Pillonetto et al. (2011), see Table 1 where GWN stands for Gaussian White Noise.

Table 1. Nonlinear dynamic models benchmark

- (1) $y_t = 2e^{-0.1y_{t-1}^2}y_{t-1} - e^{-0.1y_{t-2}^2}y_{t-2} + e_t$
 $e_t = \text{GWN}(0, 1)$
- (2) $y_t = e^{-0.1y_{t-1}^2}(2y_{t-1} - y_{t-2}) + e_t$
 $e_t = \text{GWN}(0, 1)$
- (3) $y_t = 0.5y_{t-1} - 0.05y_{t-2}^2 + u_{t-1}^2 + 0.8u_{t-2} + e_t$
 $e_t = \text{GWN}(0, 0.22^2)$
- (4) $y_t = 0.8y_{t-1} + u_{t-1} - 0.3u_{t-1}^3 + 0.25u_{t-1}u_{t-2} - 0.3u_{t-2} + 0.24u_{t-2}^3 - 0.2u_{t-2}u_{t-3} - 0.4u_{t-3} + e_t$
 $e_t = \text{GWN}(0, 0.14^2)$

For each system, we performed 100 Monte Carlo simulations varying the noise realization. We tested the approaches with $n = 100$ and $n = 200$ data, generated from zero initial conditions. The input is $u_t \sim \text{WN}(0, 1)$. The kernel employed is the one developed in Pillonetto et al. (2011) for nonlinear system identification:

$$k(\mathbf{x}_r, \mathbf{x}_s) = \lambda_1 \cdot \sum_{t=1}^{m-p+1} \exp[-\lambda_2 t] \cdot \exp\left[-\frac{d_t(r, s)}{\sigma}\right],$$

$$d_t(r, s) = \sum_{i=0}^{p-1} (u_{r-t-i} - u_{s-t-i})^2 + (y_{r-t-i} - y_{s-t-i})^2,$$

where $m \in \mathbb{N}$ is the order of both the autoregressive and exogenous part of model, such that the regressor is built as $\mathbf{x}_t = [u_{t-m} \cdots u_{t-1} \ y_{t-1} \cdots y_{t-m}]^\top$. The hyperparameters $\lambda_1, \lambda_2, \sigma \in \mathbb{R}_{>0}$ and $1 < p < m$ need to be tuned from data, where p determines the degree of interaction from different time instants. The hyperparameters $\boldsymbol{\theta}$, with $\boldsymbol{\psi} = [\lambda_1, \lambda_2, \sigma]^\top$, were tuned by Generalized Cross Validation (GCV), see Pillonetto et al. (2014). The employed kernel weights more recent observations that data farther away in the past. The value $p = 2$ was used for all the simulations, since in the considered system the maximum degree of interaction is one time stamp, while we fixed $m = 10$ for practical purposes. For dynamic connections, we fixed $\tilde{p} = p$. For ε -connections, we used the Euclidean norm distance, fixed $K = n/5$ and optimized only δ , (K is not a critical parameter when δ is tuned accordingly). The identification performance were tested on $n_T = 500$ data generated from the systems in the same way as training data.

Summarizing, the following approaches are compared:

- (1) Full connections + Gaussian weights + LEM regularizer
- (2) Dynamic connections + Gaussian weights + LEM regularizer
- (3) Full connections + kernel weights + LEM regularizer
- (4) Dynamic connections + kernel weights + LEM regularizer
- (5) ε -ball connections + Gaussian weights + LEM regularizer
- (6) ε -ball connections + kernel weights + LEM regularizer
- (7) K -NN connections + LLE weights + LLE regularizer

4.2 Results and discussion

Results are reported in Fig. 2 - Fig. 5. The estimation performance is evaluated using the Normalized Root Mean Square Error (NRMSE). A value of NRMSE greater than 1 means that the estimated model perform worse than the naive one (that predicts always the average value of the output data). It is possible to look for the effect of the different choices as: (i) comparison between choice of the weights; (ii) comparison between choice of the connections; (iii) comparison between choice of the manifold regularization term.

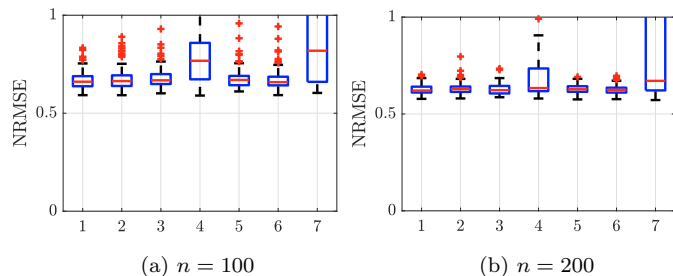


Fig. 2. Estimates comparison on benchmark system 1.

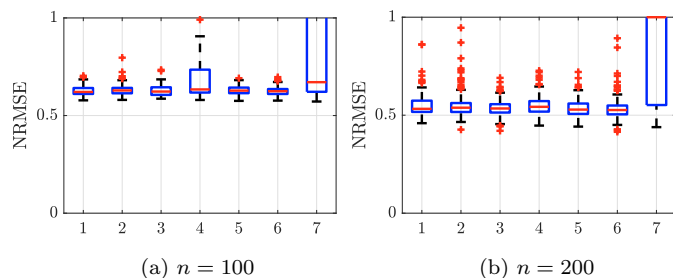


Fig. 3. Estimates comparison on benchmark system 2.

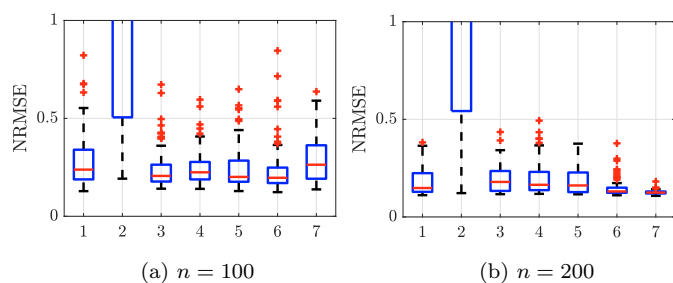


Fig. 4. Estimates comparison on benchmark system 3.

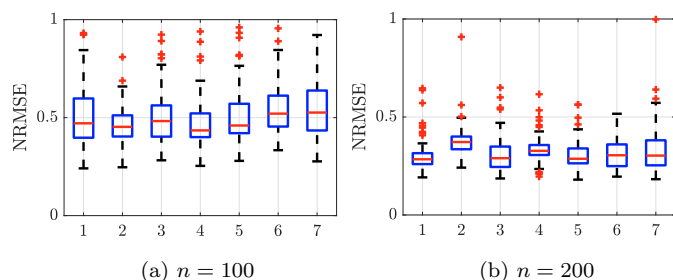


Fig. 5. Estimates comparison on benchmark system 4.

4.3 Comparison between weights

Suppose now to fix the connections and the regularizer type to LEM. We can therefore compare the approaches (1)-(2) with full connections, (3)-(4) with dynamic connections and (5)-(6) with ε -ball connections, by looking at changes in the choice of the weights. We notice that: (i) with *full connection or dynamic ones*, *Gaussian weights* perform better than kernel weights. With *ε -ball connections*, often *kernel weights* are better in terms of performance.

4.4 Comparison between connections

We now fix how to compute the weights and the regularizer to LEM, and make a comparison between different connections types. We can compare the approaches (1)-(3)-(5) that have Gaussian weights, with the approaches (2)-(4)-(6) that have kernel weights. It can be noticed that the connections type has little influence when used with *Gaussian weights*. The same is true also for *kernel weights*, but only when NAR systems are considered, i.e. Fig. 2 and Fig. 5. When NARX systems are considered, and *kernel weights* are used, the effect of the connections type is present: in particular, *ε -ball connections perform better than dynamic and full ones*.

4.5 Comparison between manifold regularization terms

It is also possible to compare the two different manifold regularization terms, i.e. the ones given by the LEM and LLE algorithms. We can compare the approaches (1),(2) with approach (7) and methods (3),(4) with method (7). In both cases, the LEM is almost always superior to LLE (that is defined only for K -NN connections).

This preliminary results show that: (i) the LEM method is to preferred to the LLE algorithm; (ii) **approach (1)**, i.e. full connections work well with Gaussian weights; (ii) **approach (6)** ε -ball connections perform well with kernel weights, especially for the ARX systems. In particular, approach (1) requires in one hyperparameter less to tune than approach (6), that, however, has a sparser Laplacian matrix and can be therefore optimized for faster computation and memory saving.

5. CONCLUSIONS

In this work, we performed a simulation study to gain intuition about the role of the regressors graph topology and weighting scheme for nonparametric nonlinear system identification with manifold regularization. Even if results are shown to be quite robust to the different strategies, these are additional degrees of freedom that can be leveraged for the specific system identification problem at hand. Future research is devoted to directly learning the regressors graph from data, and apply it in the context of system identification.

REFERENCES

- Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3), 337–404.
- Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6), 1373–1396.
- 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 Sindhvani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7, 2399–2434.
- Berry, T. and Harlim, J. (2016). Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 40(1), 68–96.
- Berry, T. and Sauer, T. (2016). Local kernels and the geometric structure of data. *Applied and Computational Harmonic Analysis*, 40(3), 439–469.
- Berry, T. and Sauer, T. (2019). Consistent manifold representation for topological data analysis. *Foundations of Data Science*, 1.
- Cayton, L. (2005). Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17), 1.
- Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and gaussian processes-revisited. *Automatica*, 48(8), 1525 – 1535.
- Coifman, R.R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1), 5–30.
- Dong, X., Thanou, D., Rabbat, M.G., and Frossard, P. (2018). Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36, 44–63.
- Formentin, S., Mazzoleni, M., Scandella, M., and Previdi, F. (2019). Nonlinear system identification via data augmentation. *Systems & Control Letters*, 128, 56 – 63.
- Hein, M., Audibert, J.Y., and Von Luxburg, U. (2005). From graphs to manifolds—weak and strong pointwise consistency of graph laplacians. In *International Conference on Computational Learning Theory*, 470–485. Springer.
- Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1), 82 – 95.
- Ljung, L., Chen, T., and Mu, B. (2019). A shift in paradigm for system identification. *International Journal of Control*, 0(0), 1–8.
- Mateos, G., Segarra, S., Marques, A.G., and Ribeiro, A. (2019). Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3), 16–43.
- Mazzoleni, M., Formentin, S., Scandella, M., and Previdi, F. (2018a). Semi-supervised learning of dynamical systems: a preliminary study. In *2018 European Control Conference (ECC)*, 2824–2829.
- Mazzoleni, M., Scandella, M., Formentin, S., and Previdi, F. (2018b). Identification of nonlinear dynamical system with synthetic data: a preliminary investigation. *IFAC-PapersOnLine*, 51(15), 622 – 627. 18th IFAC Symposium on System Identification SYSID 2018.
- Pillonetto, G., Dinuzzo, F., Chen, T., Nicolao, G.D., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3), 657 – 682.
- Pillonetto, G., Quang, M.H., and Chiuso, A. (2011). A new kernel-based approach for nonlinear system identification. *IEEE Transactions on Automatic Control*, 56(12), 2825–2840.
- Roweis, S.T. and Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500), 2323–2326.