Auto-tuning procedures for distributed nonparametric regression algorithms

Damiano Varagnolo, Gianluigi Pillonetto, Luca Schenato

Abstract—We propose a distributed regression algorithm with the capability of automatically calibrating its parameters during its on-line functioning. The estimation procedure corresponds to a Regularization Network, i.e., the structural form of the estimator is a linear combination of basis functions whose coefficients are computed by solving a linear system. The automatic tuning strategy instead constructs and then exploits opportune bounds on the distance between the distributed estimation results and the unknown centralized optimal estimate that would be computed processing the whole dataset at once. By numerical simulations we show how the proposed procedure allows the sensor networks to effectively self-tune the parameters of the distributed regression scheme by simple consensus strategies.

Index Terms—distributed regression, distributed calibration, self-organizing sensor networks, regularization networks, nonparametric estimation

I. INTRODUCTION

Applications like surveillance, monitoring, tracking and sensing, benefit of the distributed paradigm, where unmanned agents perform auxiliary and automatic operations. But to broaden the applicability of distributed paradigms, and to increase their robustness with respect to human error, algorithms should be self-configuring and self-tuning; these are indeed intermediate steps for implementing self-organizing and truly smart sensors and actuators networks.

Towards this vision we consider a specific class of distributed estimation strategies, more specifically nonparametric regression algorithms. Our interests in contributing to this field is indeed driven by some practical considerations, that make us believe in their technological possibilities: i) nonparametric strategies may be statistically more effective than parametric ones (e.g., identification of linear systems using Akaike Information Criterion plus Prediction Error Methods [1]); ii) nonparametric approaches may be consistent where parametric approaches fail to be [2], [3]; iii) nonparametric methods usually require the tuning of very few parameters, and this allows the implementation of fast calibration strategies [4]. We moreover specifically consider scenarios where agents have limited communication bandwidth, so that representations of the estimated quantities must be kept small.

Literature review: endowing nonparametric distributed estimators with self- and online-calibration capabilities is complicated by the fact that the regularization parameters \(\gamma\) in the following Equation (5), typical of nonparametric strategies, combine with global quantities that are generally unknown to the single agents, such as the total number of measurements available in the whole network.

Up to now, and to the best of our knowledge, the problem of how to address this lack of information, and thus of how to tune regularization parameters of distributed nonparametric estimators in an online fashion, has not been treated. We recognize several implementations of ad-hoc distributed self-calibration / self-diagnosis strategies, e.g., [5], [6], [7], [8], [9], and literature on the calibration of centralized nonparametric estimators, e.g., [10, Chap. 5], [11, Chap. 7], but for distributed settings the usual approach is to assume the regularization parameter (or the parameters governing the sparsification rules) to be fixed and computed off-line [12], [13], [14], [15].

Statement of contributions: there are then two ways to overcome the lack of information on global quantities like the number of measurements in the network: either distribute this information, or bypass it and exploit some other structural property of the distributed nonparametric regression framework.

Here we consider the second approach, and devise online tuning procedures that are based on opportune Euclidean distances concepts. More specifically, we consider opportune a-posteriori probabilistic bounds on the distance between the outputs of the distributed regression strategy and the centralized optimal one. We notice that the proposed strategies do not follow iterative minimization procedures, but rather compare in parallel a set of different parameters and then choose the optimal one.

Organization of the manuscript: Sec. II describes the considered regression framework, while Secs. III and IV describe respectively a centralized nonparametric estimator and its distributed version. Secs. V-A and V-B introduce then the distributed procedures for the calibration of the parameters of the regression strategy. We conclude with numerical examples in Sec. VI and with some conclusions and indications of future works in Sec. VII. To improve the readability of the paper, the proofs have been collected in the appendix.

Notice that, to the best of our knowledge, strategies for the automatic tuning of the parameters of distributed nonparametric regression algorithms have never presented before. We are thus not able to offer comparative results with some other literature works.

II. REGRESSION FRAMEWORK

Let \(f_0 : \mathcal{X} \to \mathbb{R}\) denote an unknown function defined on the compact \(\mathcal{X} \subset \mathbb{R}^d\). For brevity, and w.l.o.g. (the same
derivations could be made by letting the sensors collect more information, assume that there are $S$ sensors, each collecting a single noisy measurement $y_i$, i.e.,

$$y_i = f_\mu (x_i) + \nu_i, \quad i = 1, \ldots, S$$

(1)

with $\nu_i$ white noise and $i$ the sensor index. We assume that each input location $x_i$ is known only to the $i$-th sensor and that it is independently drawn from a probability measure $\mu$ known to all the sensors.

Notice that hereafter we will use the following notation:

- $f_\mu$ is the unknown function that has to be estimated;
- $f$ is a generic function;
- $f_\nu$ is a centralized estimate of $f_\mu$;
- $f_d$ is a distributed estimate of $f_\mu$.

### III. Centralized Regression

Given the data set $\{(x_i, y_i)\}_{i=1}^S$, one of the most used approaches to estimate $f_\mu$ relies upon the Tikhonov regularization theory [16], [17]. The hypothesis space is typically given by a reproducing kernel Hilbert space (RKHS) defined by a Mercer Kernel $K : X \times X \rightarrow \mathbb{R}$ [18], [19], [20] that is spanned by the eigenfunctions\(^\dagger\) $\phi_e$ of the positive integral operator

$$\int X K (x, x') g (x') d\mu (x')$$

(2)

where the corresponding eigenvalues $\lambda_e$ are s.t. $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$. Under mild assumptions (see, e.g., [21]), the hypothesis space is given by the Hilbert space

$$\mathcal{H}_K := \left\{ g \in L^2 (\mu) \text{ s.t. } g = \sum_{e=1}^{\infty} \alpha_e \phi_e \right\}$$

(3)

with $\{\alpha_e\}$ s.t. $\sum_{e=1}^{\infty} \frac{\alpha_e^2}{\lambda_e} < +\infty$.

Letting $g_1 = \sum_{e=1}^{\infty} \alpha_e \phi_e$ and $g_2 = \sum_{e=1}^{\infty} \beta_e \phi_e$, this implies that the inner product in $\mathcal{H}_K$ is

$$\langle g_1, g_2 \rangle_K := \sum_{e=1}^{\infty} \frac{\alpha_e \beta_e}{\lambda_e}$$

(4)

with the $\lambda_e$'s the eigenvalues of the kernel $K$.

To define the estimator of $f_\mu$ given the dataset $\{(x_i, y_i)\}_{i=1, \ldots, S}$, a commonly used cost function is

$$Q (f) := \sum_{i=1}^{S} (y_i - f (x_i))^2 + \gamma \| f \|_K^2$$

(5)

where $\gamma$ is the so-called regularization parameter that trades off empirical evidence and smoothness information on $f_\mu$. Assume w.l.o.g. $\gamma$ to be known (cf. the discussion at the beginning of Sec. V). It is known that the optimal estimate

$$f_\nu := \arg \min_{f \in \mathcal{H}_K} Q (f)$$

(6)

admits the structure of a Regularization Network, see [19], being the sum of $S$ basis functions with expansion coefficients obtainable by inverting a system of linear equations.

\(^\dagger\)For numerical computation of eigenvalues and eigenfunctions see for example [10, Chap. 4.3.2].

### IV. Distributed Regression

A potential strategy for computing $f_\nu$ over networks is to route all the information to a specific unit, and let that unit perform the computations. Since this requires the processing unit to perform $O (S^3)$ operations and to store all the $x_i$'s, generally this strategy is impractical in distributed scenarios, where agents may have both limited computational and communication resources.

We thus aim at deriving an alternative approach, more suitable for distributed settings. To this aim we consider the following roadmap:

- rewrite the optimization problem (6) in an alternative but equivalent way, by exploiting the structure of $\mathcal{H}_K$;
- change, thanks to Principal Components Analysis-like concepts, the hypothesis space from $\mathcal{H}_K$ to an approximated one;
- derive the distributed estimator as an approximated version of the centralized one.

#### A. Rewriting optimization problem (6)

Let $\mathbb{R}^\infty$ be the space of vectors with an infinite number of real scalar components. Introducing the map $T : \mathcal{H}_K \rightarrow \mathbb{R}^\infty$ associating to a function $f (\cdot) = \sum_{e=1}^{\infty} \alpha_e \phi_e (\cdot)$ in $\mathcal{H}_K$ the sequence $[\alpha_1, \alpha_2, \ldots]$ of its eigenfunctions weights, it is possible to rewrite the estimand $f_\mu$ as the novel estimand

$$b_\mu = T [f_\mu].$$

(7)

Of course $b_\mu$ and $f_\mu$ are equivalent.

Letting moreover

$$C_i := [\phi_1 (x_i) \; \phi_2 (x_i) \ldots],$$

(8)

it is possible to rewrite the measurement model (1) as

$$y_i = C_i b_\mu + \nu_i, \quad i = 1, \ldots, S,$$

(9)

and the cost function (5) as

$$Q (b) := \sum_{i=1}^{S} (y_i - C_i b)^2 + \gamma \| b \|_{K}^2.$$  

(10)

The optimal estimate $b_\nu := \arg \min_{b \in \mathbb{R}^\infty} Q (b)$ of the estimand $b_\mu$ is thus (see also [4])

$$b_\nu = \left( \text{diag} \left( \frac{\gamma}{\lambda_e} \right) + \sum_{i=1}^{S} C_i^T C_i \right)^{-1} \left( \sum_{i=1}^{S} C_i^T y_i \right)$$

(11)

with diag($\alpha_e$) indicating the matrix with diagonal elements given by $\alpha_1, \alpha_2, \ldots$.

#### B. Changing the hypothesis space

The optimal estimate $b_\nu$ in (10) is infinite dimensional, and thus numerically intractable. To obtain a numerically tractable estimator, we consider the most natural finite-dimensional alternative of $\mathcal{H}_K$, i.e., the subspace $\mathcal{H}_K^E$ generated by the first $E$ eigenfunctions $\phi_e$, i.e.,

$$\mathcal{H}_K^E := \left\{ g \in L^2 (\mu) \text{ s.t. } g = \sum_{e=1}^{E} \alpha_e \phi_e \right\}$$

(12)

with $[\alpha_1, \ldots, \alpha_E]^T \in \mathbb{R}^E$.

Substituting $\mathcal{H}_K$ with $\mathcal{H}_K^E$ is then motivated by the presence of the penalty term $\| b \|_{K}^2$ in (5): from Bayesian viewpoints,
In this case the optimal estimate of $b$ and from the cost function (9) to algorithms: one on computational costs, i.e., to exchange an amount of information and the regularization parameter $\gamma$, and the total number of measurements in the network $S$. $E$ drives the computational and communication requirements of the distributed strategy, but also the accuracy of the final estimate (as noticed in Sec. IV-B). The ratio $\gamma/S$, instead, dictates how much the empirical evidence of the final solution should be traded off with its smoothness.

In practical situations, both $E$ and $\gamma/S$ should be chosen a-posteriori, i.e., after that sensors have collected their $y_i$. The aim of this paper is then the following: considering $S$ and $E$ as unknown (even if $\gamma$ can instead w.l.o.g. be considered known, or arbitrarily be set to 1), develop in-line strategies so that sensors will find a guess $S_g$ for $S$ and for $E$ maximizing in some sense the performance of $b_d$.

In other words we highlight this parametric dependency of $b_d$ on $S_g$ and $E$ by writing

$$b_d = b_d(S_g, E),$$

and thus propose a distributed in-line self-calibration technique that allows the sensors to opportunely select $E$ and $S_g$ assuming that the $y_i$’s are locally available. The details of this strategy are offered in the following sections, and are based on the following mild assumption:

**Assumption 1** $S \in [S_{\min}, S_{\max}]$ and sensors have knowledge about $S_{\min}$ and $S_{\max}$.

**Remark 2** Even if $\gamma$ and $S$ are known, because of the additional noise coming from the approximation $I \approx \frac{1}{S} \sum_{i=1}^{S} (C_{i}^{E})^{T} C_{i}^{E}$, it can be shown that in general, for any fixed $E$, implementing $b_d$ with the exact $S$ does not maximize the predictive capabilities of $b_d$. So, even if $S$ is actually known, one may want to find on-line that $S_g$ that maximizes the statistical performance of $b_d$.

### A. Calibration of the Regularization Parameter

Assume for now $E$ to be fixed, and write $b_d(S_g)$ instead of $b_d(S_g, E)$. Despite the fact that, for any finite number of measurements, it may happen that an opportunely tuned $b_d(S_g)$ has better predictive capabilities of the centralized optimal estimate $b_c$, usually $b_c$ has bigger generalization capabilities of $b_d(S_g)$ for any $S_g \in \mathbb{R}_+$. It is then meaningful to consider $\|b_d(S_g) - b_c\|_2$ as a performance indicator, and try to tune $S_g$ seeking to minimize this distance.

Importantly, in actual distributed estimation scenarios it is impossible to compute

$$S_g^* := \arg\min_{S_g \in \mathbb{R}_+} \|b_d(S_g) - b_c\|_2.$$
since $b_r$ is unknown. It is thus necessary to proceed finding appropriate bounds for $\|b_d (S_g) - b_r\|_2$ that depend on $S_g$, and then find $S_g^*$ minimizing these bounds. The first step is given by the following proposition, that bounds $\|b_d (S_g) - b_r\|_2$ with terms that can then be computed by agents independently. (The numerical validity of these bounds is analyzed in Sec. VI.)

**Proposition 3**

Let

\[
C_i^E := [0, \ldots, 0, \phi_{E+1} (x_i), \phi_{E+2} (x_i), \ldots] \tag{19}
\]

\[
\gamma_a := \sup_{x \in \mathcal{X}} \left\| \text{diag} \left( \frac{\lambda_e}{\gamma} \right) \left( C^E (x) \right)^T \right\|_2 \tag{20}
\]

\[
\gamma_b := \sup_{x \in \mathcal{X}} \left\| \text{diag} \left( \frac{\lambda_e}{\gamma} \right) \left( C^E (x) \right)^T C^E \right\|_2 \tag{21}
\]

\[
V_r := \left( \frac{1}{S} \text{diag} \left( \frac{\gamma}{\lambda_e} \right) + \frac{1}{S} \sum_{i=1}^{S} (C_i^E)^T C_i^E \right)^{-1} \tag{22}
\]

\[
V_d (S_g) := \left( \frac{1}{S} \text{diag} \left( \frac{\gamma}{\lambda_e} \right) + I \right)^{-1} \tag{23}
\]

\[
U_C := I - \frac{1}{S} \sum_{i=1}^{S} (C_i^E)^T C_i^E \tag{24}
\]

\[
U_S (S_g) := \left( \frac{1}{S} - \frac{1}{S} \right) \text{diag} \left( \frac{\gamma}{\lambda_e} \right) . \tag{25}
\]

Then

\[
\|b_d (S_g) - b_r\|_2 \leq \| V_r U_S (S_g) b_d (S_g) \|_2 + \| V_r U_C b_d (S_g) \|_2 \tag{26}
\]

and

\[
\|b_d (S_g) - b_r\|_2 \leq (\gamma_b S_{\text{max}} + 1) \|b_d (S_g) - b_r\|_2 + \sum_{i=1}^{S} \gamma_a \|y_i - C_i^E b_d (S_g)\|_2 \tag{27}
\]

The terms involved in Prop. 3 have the following interpretations:

- $C_i^E$ is the part of the transformation expressed in (8) corresponding to the discarded eigenfunctions;
- $\gamma_a$ and $\gamma_b$ respectively bound how much the residuals $y_i - C_i^E b_d (S_g)$ and $b_d (S_g) - b_r$ will influence the overall approximation error $b_d (S_g) - b_r$;
- $V_r$ is s.t. $\frac{1}{S} V_r^{-1}$ is an approximation of the true covariance of the set of measurements $\{y_i\}$. More precisely, $\frac{1}{S} V_r^{-1}$ would be the actual covariance if $\lambda_{E+1} = \lambda_{E+2} = \ldots = 0$. The smaller these eigenvalues are, the better $\frac{1}{S} V_r^{-1}$ is an approximation of the actual covariance;
- $V_d (S_g)$ corresponds to an opportune approximation of $V_r$;
- $U_C$ corresponds to the approximation error encountered replacing $\frac{1}{S} \sum_{i=1}^{S} (C_i^E)^T C_i^E$ with $E_{\mu} \left[ (C_i^E)^T C_i^E \right]$;
- $U_S (S_g)$ modulates how the error on the regularization parameter affect the regularization properties of the proposed distributed estimator.

The usefulness of Prop. 3 is that it is possible to build on top of it to construct the following bound for $\|b_d (S_g) - b_r\|_2$:

\[
\mathcal{B} (S_g) := (\gamma_b S_{\text{max}} + 1) \left( \| V_r U_S (S_g) b_d (S_g) \|_2 + \| V_r U_C b_d (S_g) \|_2 \right) + \sum_{i=1}^{S} \gamma_a \|y_i - C_i^E b_d (S_g)\|_2 . \tag{28}
\]

One would then want to optimize on-line the unknown parameter $S_g$ through

\[
S_{g}^* := \arg \min_{S_g \in \mathbb{R}_+} \mathcal{B} (S_g) ; \tag{29}
\]

nonetheless $\mathcal{B} (S_g)$ cannot be directly used for computing $S_g$ since the quantities $V_r, U_S (S_g), U_C$ and $S$ are unknown to the various sensors.

To cope with this lack of information we propose thus to:

1) majorize $U_S^* (S_g)$ with $U_S^* (S_g)$, defined as

\[
U_S^* (S_g) := \max \left( \frac{1}{S} - \frac{1}{S_{\text{max}}} \right) \text{diag} \left( \frac{\gamma}{\lambda_e} \right) \tag{30}
\]

and exploiting Assumption 1. Indeed it is immediate to check that

\[
U_S^* (S_g) \geq U_S (S_g) \quad \forall S_g \in \mathbb{R}_+ \tag{31}
\]

where the inequality is in a matricial positive definite sense.

2) majorize $V_r$ and $U_C$ with quantities that are generated locally by each sensor $i$ as follows: $a)$ locally simulate a particular scenario of the network by locally generating $S_{\text{min}}$ independent virtual input locations $x_{i,j}$ by means of density $\mu$, i.e., each $i$ generates

\[
x_{i,j} \sim \mu \quad \text{where} \quad j = 1, \ldots, S_{\text{min}}. \tag{32}
\]

b) then each $i$ locally computes

\[
C_{i,j}^E := \left[ \phi_1 (x_{i,j}), \ldots, \phi_E (x_{i,j}) \right] ; \tag{33}
\]

\[
V_{r,i}^* := \left( \frac{1}{S_{\text{max}}} \text{diag} \left( \frac{\gamma}{\lambda_e} \right) + \frac{1}{S_{\text{max}}} \sum_{j=1}^{S_{\text{min}}} (C_{i,j}^E)^T C_{i,j}^E \right)^{-1} \tag{34}
\]

\[
U_{C,i}^* := \left( I - \frac{1}{S_{\text{min}}} \sum_{j=1}^{S_{\text{min}}} (C_{i,j}^E)^T C_{i,j}^E \right) , \tag{35}
\]

i.e., from probabilistic viewpoints, generate $V_{r,i}^*$ and $U_{C,i}^*$ as pessimistic but informative versions of the true and unknown $V_r$ and $U_C$.

By means of the previous scheme, optimization of $S_g$ is now then possible through solving

\[
S_{g}^* := \arg \min_{S_g \in \mathbb{R}_+} \mathcal{B}^* (S_g) \tag{36}
\]
where

\[ B^*(S_g) := (\gamma_bS_{\text{max}} + 1) \cdot \frac{1}{S} \sum_{i=1}^{S} \left( \| V_{r_i}^* U_{S_i}^* S_g b_d (S_g) \|_2 \right) + (\gamma_a S_{\text{max}}) \cdot \frac{1}{S} \sum_{i=1}^{S} \| y_i - C_i^b b_d (S_g) \|_2 \]  

(35)

Intuitively, thus, agents try to minimize a pessimistic estimate \( B^*(S_g) \) of \( B(S_g) \) instead of \( B(S_g) \) itself. The complete algorithm is then reported in Alg. 1, solving problem (35) by gridding, i.e., selecting the best \( S_g \) from a set of candidates \( S_g^{(1)}, \ldots, S_g^{(P)} \).

**Algorithm 1 Distributed calibration of the regularization parameter**

**Off-line work:** Sensors are given \( S_{\text{min}}, S_{\text{max}}, \mu, E, \gamma_a, \gamma_b \), a set of \( R \) different candidates \( S_g^{(1)}, \ldots, S_g^{(P)} \) and relative matrices \( U_{S}^* S_g^{(1)} \), \ldots, \( U_{S}^* S_g^{(P)} \). In addition, each sensor \( i \) locally generates \( S_{\text{min}} \) independent virtual input locations \( x_{i,j}, j = 1, \ldots, S_{\text{min}} \) by means of density means of average consensus protocols, the \( E \)-dimensional vector

\[ Z := \frac{1}{S} \sum_{i=1}^{S} (C_i^E)^T y_i \]  

(36)

**On-line and distributed work:**

1: (distributed step) sensors distributively compute, by means of average consensus protocols, the \( E \)-dimensional vector

2: (local step) each sensor \( i \) computes the \( P \) versions of the estimator (17), namely \( b_d (S_g^{(p)}) = V_d (S_g^{(p)}) Z \), for \( p = 1, \ldots, P \).

3: (local step) each sensor \( i \) computes the local \( P \) auxiliary scalars, for \( p = 1, \ldots, P \)

\[ B_i^* (S_g^{(p)}) := (\gamma_b S_{\text{max}} + 1) \| V_{r_i}^* U_{S_i}^* S_g^{(p)} b_d (S_g^{(p)}) \|_2 + (\gamma_a S_{\text{max}}) \cdot \| y_i - C_i^b b_d (S_g^{(p)}) \|_2 \]  

(37)

4: (distributed step) sensors distributively compute, by means of average consensus protocols, the \( P \) scalars, for \( p = 1, \ldots, P \)

\[ B^* (S_g^{(p)}) := \frac{1}{S} \sum_{i=1}^{S} B_i^* (S_g^{(p)}) \]  

(38)

5: (local step) each sensor \( i \) computes \( S_g^* = S_g^{(p^*)} \) where

\[ (p^*) = \arg \min_{(p)} B^* (S_g^{(p)}) \]  

(39)

B. Calibration of the Number of Eigenfunctions

The maximum admissible value for \( E \) is upper bounded by computational complexity and transmission capability constraints. Assuming \( E \) to be this maximum value, the usage of a naive strategy like \( E = E \) could lead to resource wasting. In the following algorithm 2 we offer a practical and general guideline for the choice of \( E \) exploiting pessimistic bounds on the approximation error \( \| b_c - b_r \|_2 \).

**Algorithm 2 Calibration of the number of eigenfunctions**

1: assume the knowledge of a lower bound on the energy of the unknown signal \( f_\mu \), indicated with \( \min \| f_\mu \|_2 \)

2: choose a threshold \( \delta \) for the maximal tolerable error \( \| b_c - b_r \|_2 \)

3: compute the minimal value of \( E \) s.t.

\[ 3\sigma S_{\text{max}} \gamma_a (E) \cdot \min \| f_\mu \|_2 \leq \delta \]  

(39)

where we highlighted the dependence of \( \gamma_a \) on \( E \).

From a practical point of view, the algorithm returns a number \( E \) assuring the operator that the normalized approximation error \( \| b_c - b_r \|_2 \) is smaller than a certain threshold. The algorithm is derived from the consideration that inequality (55) in the proof of Prop. 3 implies

\[ \| b_c - b_r \|_2 \leq \gamma_a \sum_{i=1}^{S} \| y_i - C_i b_r \|_2 \]  

(40)

and the consideration that, in general, residuals \( \| y_i - C_i b_r \|_2 \) are far smaller than 3 times the standard deviation of the measurement noise. We notice that this choice is arbitrary and relies on the assumption that the estimation result will have a certain minimum level of generalization capabilities. Pessimistic considerations can lead to increase the number of standard deviations, with the limit case of no approximation capabilities of \( b_r \) corresponding to set \( b_r = 0 \) in (40) and to substitute \( 3\sigma \) with \( \max_i \| y_i \|_2 \) in (39).

We notice that, substituting \( \min \| f_\mu \|_2 \) with \( \max_i \| y_i \|_2 \) in (39), algorithm 2 can be used in a-posteriori scenarios, where sensors decide \( E \) by means of a max consensus on \( \| y_i \|_2 \) before computing (36). We also notice that high uncertainties on \( S \) lead to overestimations of \( E \) because of the approximation \( S_{\text{max}} \).

VI. NUMERICAL EXAMPLES

In this section we show the effectiveness of the proposed strategies through some numerical examples. We consider \( f_\mu : \mathcal{X} = [0, 1] \rightarrow \mathbb{R} \) to be given by

\[ f_\mu (x) = \sum_{n=1}^{100} \alpha_n \sin (\omega_n x) \]  

(41)

with \( \alpha_n \sim \mathcal{N} (0, 0.01) \) i.i.d., \( \omega_n \sim \mathcal{U} [0, 25] \) i.i.d., \( \mu \sim \mathcal{U} [0, 1] \) and a measurement noise standard deviation \( \sigma = \).
0.75 s.t., on average, \( \text{SNR} := \frac{\text{var}(f_\mu)}{\sigma^2} \approx 2.5. \) Moreover we consider the Gaussian kernel

\[
K(x, x') = \exp\left(-\frac{(x - x')^2}{0.02}\right)
\]

with the estimators (10) and (17) defined by \( \gamma = 0.3. \)

To show the effectiveness of the estimation strategy (17), a randomly generated realization of \( f_\mu \) is sampled by \( S = 100 \) sensors and estimated using \( E = 20 \) eigenfunctions\(^2\) under two different uncertainty levels on \( S \), namely case (a), where \( S_{\text{min}} = 90 \) and \( S_{\text{max}} = 110 \), and case (b), where \( S_{\text{min}} = 20 \) and \( S_{\text{max}} = 2000 \). In Fig. 1 we plot then the actual realization (solid line), its estimates reconstructed from \( b_c \) (dashed line) and \( b_c^{(S)}(S^*_g) \) with \( S^*_g \) chosen by Algorithm 1 among 20 candidates logarithmically spaced inside \([1, S_{\text{max}}]\), and \( (\cdot) = (a) \) or (b) accordingly to the level of uncertainty on \( S \) (dotted and dashed-dotted lines, respectively). We claim an overall insensitivity of \( b_d \) on the uncertainty on \( S \) considering that both \( T^{-1}[b_d^{(a)}] \) and \( T^{-1}[b_d^{(b)}] \) are close to the centralized estimate \( T^{-1}[b_c] \).

Despite this valuable property, bounds \( B^* \) are good indicators about the actual distance \( \|b_d(S^*_g) - b_c\|_2 \) only for the case (a) (low uncertainty on \( S \)), as Fig. 2 indicates. In this figure we generate 200 independent realizations of \( f_\mu \), then estimate each \( f_\mu \) as before, and finally plot the actual distances \( \|b_d(S^*_g) - b_c\|_2 \) versus the obtained bounds \( B^* \). It is important to see that the bound provides, for the case (b), meaningless information on the actual distance. This lack of meaningfulness is caused by the presence in the bound of the multiplicative factor \( S_{\text{max}} \). This implies that in general the accuracy of the bound is tightly connected with the accuracy of the knowledge on \( S \).

For sake of completeness, we show in Fig. 3 the values of the bounds \( B^*(S^*_g) \) defined in (37) associated to the experiment of Fig. 1, and the relative distances \( \|b_d(S^*_g) - b_c\|_2 \). It is possible to see how the qualitative behavior of curve \( B^*(S^*_g) \) is similar to the one of curve \( \|b_d(S^*_g) - b_c\|_2 \).

\(^2\)This particular choice will be motivated later.

---

(a) case (a): \( S_{\text{min}} = 90, S_{\text{max}} = 110 \)

(b) case (b): \( S_{\text{min}} = 20, S_{\text{max}} = 2000 \)

---

We then aim to check if it is better to use Alg. 1 or to try directly to try to estimate \( S \). We thus compare the estimation performance obtainable with three different naïve strategies for the choice of \( S_g \), namely \( S^*_g = S_{\text{min}} \), \( S^*_g = S_{\text{max}} \), \( S^*_g = S_{\text{ave}} := \frac{S_{\text{min}} + S_{\text{max}}}{2} \). Considering panels (a) of Figs. 2 and 3 it is possible to infer that:

- in case of low uncertainty levels, Alg. 1 will not lead to big improvements w.r.t. naïve strategies, but will give accurate descriptions of the actual distance with the centralized estimate;
- in case of high uncertainty levels, Alg. 1 will not give accurate descriptions of the actual distance with the centralized estimate but its usage will lead to improvements w.r.t. naïve strategies.
To numerically prove the last statement, we consider the previously generated 200 independent realizations of \( f_\mu \) and the case \( S_{\text{min}} = 20 \) and \( S_{\text{max}} = 2000 \). We then plot in Fig. 4 the 100 points

\[
\left( \|b_d(S_{\text{min}}) - b_c\|_2, \|b_d(S_g^*) - b_c\|_2 \right) \quad (43)
\]

\[
\left( \|b_d(S_{\text{ave}}) - b_c\|_2, \|b_d(S_g^*) - b_c\|_2 \right) \quad (44)
\]

\[
\left( \|b_d(S_{\text{max}}) - b_c\|_2, \|b_d(S_g^*) - b_c\|_2 \right) \quad (45)
\]

in panels (a), (b) and (c) respectively. Since these points generally lie below the bisector of the first quadrant, the distributed estimators \( b_d \) with \( S_g \) chosen with Alg. 1 are generally closer to the centralized estimates \( b_c \) than the ones with naïvely chosen \( S_g \)'s. Finally, to check the level of suboptimality of the results of Alg. 1, in panel (d) of the same figure we plot also the points

\[
\left( \|b_d(S_g^{\text{ora}}) - b_c\|_2, \|b_d(S_g^*) - b_c\|_2 \right) \quad (46)
\]

where \( S_g^{\text{ora}} \) are the optimal \( S_g \)'s obtained exactly solving problem (18). Since the distance of these points from the bisector is small, we can conclude that the level of suboptimality of Alg. 1 is also small.

![Fig. 4. Scatter plots to test the effectiveness of Alg. 1. Left-up panel: scatter plots of the points defined in (43). Right-up panel: points defined in (44). Left-down panel: points defined in (45). Right-down panel: points defined in (46). \( S_{\text{min}} = 20 \) and \( S_{\text{max}} = 2000 \).](image)

To test the effectiveness of Alg. 2 and motivate the previous choice \( E = 20 \), we plot in Fig. 5 the values of \( E \) returned by the on-line version of Alg. 2, applied to the experiment of Fig. 1 with \( S_{\text{min}} = 20 \) and \( S_{\text{max}} = 2000 \), and fed with various values for the threshold \( \delta \). We notice that the exponential decay of the bound is inherited by the exponential decay of eigenvalues \( \lambda_e \) associated to the Gaussian kernel. Different kernels would lead to different outputs. Notice that if we let \( \delta = 10^{-3} \) we obtain \( E = 20 \) and thus motivate the previous choice.

![Fig. 5. Values of \( E \) returned by the on-line version of Alg. 2 fed with various choices of the threshold \( \delta \) and applied to the experiment of Fig. 1 with \( S_{\text{min}} = 20 \) and \( S_{\text{max}} = 2000 \).](image)

VII. CONCLUSIONS

In this paper we analyze how to endow distributed non-parametric regression strategies with self-tuning capabilities. The considered estimator is characterized by two parameters: the first one, the regularization parameter, that trades off the empirical evidence and the smoothness information on the true function. The second one, the number of eigenfunctions to be used, determines the size of the hypothesis space. Here we constructed a novel distributed and on-line parameters self-calibration strategy exploiting opportune a-posteriori probabilistic bounds on the distance between the parametrized distributed estimator and the unknown estimate that would be computed in a centralized scenario.

We also analyzed the performances of this distributed parameters calibration strategy through numerical experiments, and shown that under highly uncertain topological knowledge, the strategy leads to improvements with respect to naïve calibration strategies. On the contrary, in case of accurate knowledge on the number of sensors in the network, the computed probabilistic bounds constitute an accurate description of the distance between the distributed regression strategy and an optimal centralized one.

As examples of future works, we notice that the proposed strategy can be ameliorated exploiting statistical knowledge about the number of sensors in the network. Moreover, the strategy can be extended in order to compute on the fly the minimal number of eigenfunctions guaranteeing a certain regression quality.

APPENDIX

Proof (Proof of Prop. 3) We rewrite (14) as

\[
V_r^{-1}b_r = \mathcal{Z} \quad (47)
\]

and (17) as

\[
(V_r^{-1} + V_d^{-1}(S_g) - V_r^{-1}) b_d(S_g) = \mathcal{Z} . \quad (48)
\]

Subtracting (48) to (47) we then obtain

\[
b_r - b_d(S_g) = V_r(V_d^{-1}(S_g) - V_r^{-1}) b_d(S_g) \quad (49)
\]

from which it immediately follows that

\[
\|b_d - b_r(S_g)\|_2 = \|V_r(V_d^{-1}(S_g) - V_r^{-1}) b_d(S_g)\|_2 . \quad (50)
\]
Defining then $U_C$ and $U_2$ by means of (24) and (25), it is immediate to check that $V_d^{-1}(S_g) - V_r^{-1} = U_S(S_g) + U_C$ from which inequality (26) immediately follows.

To prove (27), we rewrite (14) as

$$
\left( \text{diag} \left( \frac{\gamma}{\lambda_c} \right) + \sum_{i=1}^{S} C_i^T C_i \right) b_r + \left( \sum_{i=1}^{S} (C_i^E)^T \right) y_i = \sum_{i=1}^{S} C_i^T y_i - \sum_{i=1}^{S} \left( C_i^{\lambda E} \right)^T y_i.
$$

(51)

and (10) as

$$
\left( \text{diag} \left( \frac{\gamma}{\lambda_c} \right) + \sum_{i=1}^{S} C_i^T C_i \right) b_c = \sum_{i=1}^{S} C_i^T y_i.
$$

(52)

After subtracting (52) to (51), we obtain

$$
\left( \text{diag} \left( \frac{\gamma}{\lambda_c} \right) + \sum_{i=1}^{S} C_i^T C_i \right) (b_c - b_r) = \left( \sum_{i=1}^{S} (C_i^E)^T \right) y_i.
$$

(53)

Substituting now each $C_i$ in the right side of (53) with $C_i^E + C_i^{\lambda E}$, exploiting the fact that $C_i^{\lambda E} b_r = 0$ (where $0$ is an infinite dimensional vector of zeros), and properly collecting the various terms, we obtain

$$
b_c - b_r = \left( \text{diag} \left( \frac{\gamma}{\lambda_c} \right) + \sum_{i=1}^{S} C_i^T C_i \right) \left( \sum_{i=1}^{S} (C_i^E)^T \right) (y_i - C_i b_r).
$$

(54)

Since $\text{diag} \left( \frac{\gamma}{\lambda_c} \right) + \sum_{i=1}^{S} C_i^T C_i \succeq \text{diag} \left( \frac{\lambda_c}{\gamma} \right)$ (in a matricial positive definite sense), we obtain

$$
\|b_c - b_r\|_2 \leq \gamma \sum_{i=1}^{S} \left( \text{diag} \left( \frac{\lambda_c}{\gamma} \right) \right) \left( C_i^E \right)^T (y_i - C_i b_r).
$$

(55)

Rewriting $y_i - C_i b_r$ as $y_i - C_i^{\lambda E} b_d(S_g) + C_i^{\lambda E} b_d(S_d) - C_i^{\lambda E} b_r$, and using definitions (20) and (21), it follows immediately that

$$
\|b_c - b_r\|_2 \leq \gamma \sum_{i=1}^{S} \left( y_i - C_i b_d(S_g) \right) + \sum_{i=1}^{S} \left( \|b_r - b_d(S_g)\|_2 \right)
$$

$$
\leq \gamma \sum_{i=1}^{S} \left( y_i - C_i b_d(S_d) \right) + \sum_{i=1}^{S} \left( \|b_r - b_d(S_d)\|_2 \right)
$$

(56)

Notice that $\gamma$ is finite since for every $x \in X$ it holds that

$$
\left( \text{diag} \left( \frac{\lambda_c}{\gamma} \right) C_i^{\lambda E} (x) \right)_2 \leq \sup_{x \in X, \epsilon \in \mathbb{N}_+} \phi_\epsilon(x) \sum_{e=E+1}^{\infty} \frac{\lambda_c}{\gamma}
$$

(57)

with $\sup_{x \in X, \epsilon \in \mathbb{N}_+} \phi_\epsilon(x) < +\infty$ because eigenfunctions are continuous on a compact, and also with $\sum_{e=E+1}^{\infty} \frac{\lambda_c}{\gamma} < +\infty$ since $K$ is Mercer. In the same way it is possible to show that also $\gamma$ is finite.

(27) can then be proved substituting (56) in

$$
\|b_c - b_d(S_g)\|_2 \leq \|b_c - b_r\|_2 + \|b_r - b_d(S_g)\|_2.
$$

(58)