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 which 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 (γ 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 a 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 distributedly estimate 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 on-line 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.

This work is supported by the Framework Programme for Research and Innovation Horizon 2020 under grant agreement n. 636834 (DISIRE), and the Swedish research council Norrbottens Forskningsråd.

D. Varagnolo is with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden. Email: damiano.varagnolo@ltu.se. G. Pillonetto and L. Schenato are with the Department of Information Engineering, University of Padova, Italy. Email: { giapi | schenato }@dei.unipd.it.

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 $f_\mu : \mathcal{X} \to \mathbb{R}$ 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.,

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

with ν_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 μ known to all the sensors.

Notice that hereafter we will use the following notation: • f_μ is the unknown function that has to be estimated; • f is a generic function; • f_c is a centralized estimate of f_μ; • f_d is a distributed estimate of f_μ.

III. Centralized Regression

Given the data set $\{x_i, y_i\}_{i=1}^{S}$, one of the most used approaches to estimate f_μ 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 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ [18], [19], [20] that is spanned by the eigenfunctions[1] ϕ_e of the positive integral operator

$$\int_{\mathcal{X}} K(x, x')\, g(x')\, d\mu(x') \tag{2}$$

where the corresponding eigenvalues λ_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

$$\begin{aligned} \mathcal{H}_K \quad := \quad & \Big\{ g \in \mathcal{L}^2(\mu) \ \text{ s.t. } \ g = \textstyle\sum_{e=1}^{\infty} \alpha_e \phi_e \\ & \ \text{with } \{\alpha_e\} \text{ s.t. } \textstyle\sum_{e=1}^{\infty} \frac{\alpha_e^2}{\lambda_e} < +\infty \Big\}. \end{aligned} \tag{3}$$

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

$$\langle g_1, g_2 \rangle_K := \sum_{e=1}^{+\infty} \frac{\alpha_e \beta_e}{\lambda_e} \tag{4}$$

with the λ_e's the eigenvalues of the kernel K.

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

$$Q(f) := \sum_{i=1}^{S} \big(y_i - f(x_i)\big)^2 + \gamma \|f\|_K^2 \tag{5}$$

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

$$f_c := \arg \min_{f \in \mathcal{H}_K} Q(f) \tag{6}$$

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.

IV. Distributed Regression

A potential strategy for computing f_c 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 $O\left(S^3\right)$ 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.

[1] For numerical computation of eigenvalues and eigenfunctions see for example [10, Chap. 4.3.2].

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

$$T : \mathcal{H}_K \to \mathbb{R}^\infty \qquad T\left[\sum_{e=1}^{+\infty} a_e \phi_e(\cdot)\right] = [a_1, a_2 \ldots] \tag{7}$$

i.e., the map associating to a generic function $f(\cdot) = \sum_{e=1}^{+\infty} a_e \phi_e(\cdot)$ in $\mathcal{H}_K$ the sequence $[a_1, a_2 \ldots]$ of its eigenfunctions weights, it is possible to rewrite the estimand f_μ as the novel estimand $b_\mu = T[f_\mu]$. Of course b_μ and f_μ are equivalent.

Letting moreover

$$C_i := [\phi_1(x_i) \;\; \phi_2(x_i) \;\; \ldots], \tag{8}$$

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

$$y_i = C_i b_\mu + \nu_i, \qquad\qquad i = 1, \ldots, S, \tag{9}$$

and the cost function (5) as

$$Q(b) := \sum_{i=1}^{S} (y_i - C_i b)^2 + \gamma \|b\|_K^2 . \tag{10}$$

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

$$b_c = \left(\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) + \sum_{i=1}^{S} C_i^T C_i\right)^{-1} \left(\sum_{i=1}^{S} C_i^T y_i\right) \tag{11}$$

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

B. Changing the hypothesis space

The optimal estimate b_c in (11) 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 ϕ_e, i.e.,

$$\begin{array}{rcl} \mathcal{H}_K^E & := & \Big\{ g \in \mathcal{L}^2(\mu) \;\; \text{s.t.} \;\; g = \sum_{e=1}^{E} \alpha_e \phi_e \\ & & \quad \text{with} \;\; [\alpha_1, \ldots, \alpha_E]^T \in \mathbb{R}^E \quad \Big\} . \end{array} \tag{12}$$

Substituting $\mathcal{H}_K$ with $\mathcal{H}_K^E$ is then motivated by the presence of the penalty term $\|\cdot\|_K^2$ in (5): from Bayesian viewpoints, $\mathcal{H}_K^E$ represents the subspace that, before seeing the data, captures the biggest part of the signal variance among all the subspaces of dimension E [22], [10], in accordance with the Rayleigh's principle which underlies Principal Component Analysis [23].

C. Deriving the distributed estimator

Given the change from the hypothesis space $\mathcal{H}_K$ to $\mathcal{H}_K^E$, consider also the change from C_i in (8) to

$$C_i^E = C^E(x_i) := [\phi_1(x_i), \cdots, \phi_E(x_i), 0, 0, \ldots], \tag{13}$$

and from the cost function (10) to

$$Q^E(b) := \sum_{i=1}^{S} \left(y_i - C_i^E b\right)^2 + \gamma \|b\|_K^2 \;\; . \tag{14}$$

In this case the optimal estimate of b_μ using $\mathcal{H}_K^E$ as hypothesis space is then given by (see also [4])

$$\begin{aligned} b_r &:= \arg\min_{b \in \mathcal{H}_K^E} Q(b) = \arg\min_{b \in \mathcal{H}_K^E} Q^E(b) \\ &= \left(\frac{1}{S}\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{S}\sum_{i=1}^{S} \left(C_i^E\right)^T C_i^E\right)^{-1} \left(\frac{1}{S}\sum_{i=1}^{S} \left(C_i^E\right)^T y_i\right) \end{aligned} \tag{15}$$

Thus, if sensors know the number of measurements S and the regularization parameter γ, then b_r can be distributedly computed through two parallel average consensus algorithms: one on $\left(C_i^E\right)^T C_i^E$ and one on $\left(C_i^E\right)^T y_i$, plus multiplications and inversions of $E \times E$ matrices and E-dimensional vectors.

But even if sensors know the number of measurements S and the regularization parameter γ, as noticed in [24], the distributed implementation of (15) may still be problematic since it requires $O\left(E^2\right)$-communication and $O\left(E^3\right)$-computational costs, i.e., to exchange an amount of information that scales with the square of E, potentially too high. To this aim it is possible to consider that

$$\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E \approx \mathbb{E}_\mu\left[\left(C_i^E\right)^T C_i^E\right] = \mathrm{diag}\left(I, 0\right) \tag{16}$$

where I is $E\times E$-dimensional, and 0 is infinite dimensional. This equivalence is guaranteed by the fact that for $1 \leq m, n \leq E$

$$\left[\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E\right]_{mn} = \frac{1}{S}\sum_{i=1}^{S}\phi_m\left(x_i\right)\phi_n\left(x_i\right) \tag{17}$$

and, that, due to the orthogonality of the eigenfunctions of the kernel K in $\mathcal{L}^2\left(\mu\right)$ and the fact that the x_i's are i.i.d. and extracted from μ,

$$\frac{1}{S}\sum_{i=1}^{S}\phi_m\left(x_i\right)\phi_n\left(x_i\right) \xrightarrow{S\rightarrow+\infty} \int_{\mathcal{X}}\phi_i\left(x\right)\phi_j\left(x\right)d\mu\left(x\right) = \delta_{ij}\ .$$

This means that b_r can be approximated with

$$b_d := \mathrm{diag}\left(\frac{\lambda_e}{\gamma/S+\lambda_e}\right)\left(\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T y_i\right), \tag{18}$$

an estimator that is particularly suitable for distributed estimation purposes since it does neither require sensors to exchange information on their input locations x_i (i.e., the C_i^E) nor to compute matrix inversions; it only requires an average consensus on the E-dimensional vectors $\left(C_i^E\right)^T y_i$.

V. Autotuning procedures

Consider estimator b_d in (18). This estimator is parametrized in the number of eigenfunctions E, the regularization parameter γ, 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 γ/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 γ/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 (γ 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

$$b_d = b_d\left(S_g, E\right),$$

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

Remark 2 Even if γ and S are known, because of the additional noise coming from the approximation $I \approx \frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E$, 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 $b_d\left(S_g\right)$ instead of $b_d\left(S_g, E\right)$. Despite the fact that, for any finite number of measurements, it may happen that an opportunely tuned $b_d\left(S_g\right)$ has better predictive capabilities of the centralized optimal estimate b_c, usually b_c has bigger generalization capabilities of $b_d\left(S_g\right)$ for any $S_g \in \mathbb{R}_+$. It is then meaningful to consider $\left\|b_d\left(S_g\right) - b_c\right\|_2$ 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

$$S_g^* := \arg\min_{S_g \in \mathbb{R}_+} \left\| b_d\left(S_g\right) - b_c \right\|_2 \; . \tag{19}$$

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

Proposition 3 Let

$$C_i^{\setminus E} := \left[0, \ldots, 0, \phi_{E+1}\left(x_i\right), \phi_{E+2}\left(x_i\right), \ldots\right] \tag{20}$$

$$\gamma_a := \sup_{x \in \mathcal{X}} \left\| \operatorname{diag}\left(\frac{\lambda_e}{\gamma}\right) \left(C^{\setminus E}\left(x\right)\right)^T \right\|_2 \tag{21}$$

$$\gamma_b := \sup_{x \in \mathcal{X}} \left\| \operatorname{diag}\left(\frac{\lambda_e}{\gamma}\right) \left(C^{\setminus E}\left(x\right)\right)^T C^E\left(x\right) \right\|_2 \tag{22}$$

$$V_r := \left(\frac{1}{S}\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E\right)^{-1} \tag{23}$$

$$V_d\left(S_g\right) := \left(\frac{1}{S_g}\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) + I\right)^{-1} \tag{24}$$

$$U_C := I - \frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E \tag{25}$$

$$U_S\left(S_g\right) := \left(\frac{1}{S_g} - \frac{1}{S}\right)\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) \; . \tag{26}$$

Then

$$\left\| b_d\left(S_g\right) - b_r \right\|_2 \leq \left\| V_r U_S\left(S_g\right) b_d\left(S_g\right) \right\|_2 + \left\| V_r U_C b_d\left(S_g\right) \right\|_2 \tag{27}$$

and

$$\begin{aligned} \left\| b_d\left(S_g\right) - b_c \right\|_2 \quad \leq & \quad \left(\gamma_b S_{\text{max}} + 1\right) \left\| b_d\left(S_g\right) - b_r \right\|_2 \\ & + \textstyle\sum_{i=1}^{S} \gamma_a \left\| y_i - C_i^E b_d\left(S_g\right) \right\|_2 \end{aligned} \tag{28}$$

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

- $C_i^{\setminus E}$ is the part of the transformation expressed in (9) corresponding to the discarded eigenfunctions;
- γ_a and γ_b respectively bound how much the residuals $y_i - C_i^E b_d\left(S_g\right)$ and $b_d\left(S_g\right) - b_r$ will influence the overall approximation error $b_d\left(S_g\right) - b_c$;
- 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;
- $V_d\left(S_g\right)$ corresponds to an opportune approximation of V_r;
- U_C corresponds to the approximation error encountered replacing $\frac{1}{S}\sum_{i=1}^{S}\left(C_i^E\right)^T C_i^E$ with $\mathbb{E}_\mu\left[\left(C_i^E\right)^T C_i^E\right]$;
- $U_S\left(S_g\right)$ 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 $\left\| b_d\left(S_g\right) - b_c \right\|_2$:

$$\begin{aligned} \mathcal{B}\left(S_g\right) \;\; := \;\; & \left(\gamma_b S_{\text{max}} + 1\right)\big(\left\| V_r U_S\left(S_g\right) b_d\left(S_g\right) \right\|_2 + \\ & \qquad + \left\| V_r U_C b_d\left(S_g\right) \right\|_2\big) + \\ & + \textstyle\sum_{i=1}^{S} \gamma_a \left\| y_i - C_i^E b_d\left(S_g\right) \right\|_2 \; . \end{aligned} \tag{29}$$

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

$$S_g^* := \arg\min_{S_g \in \mathbb{R}_+} \mathcal{B}\left(S_g\right); \tag{30}$$

nonetheless $\mathcal{B}\left(S_g\right)$ cannot be directly used for computing S_g since the quantities V_r, $U_S\left(S_g\right)$, 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
$$U_S^*(S_g) := \max\left(\left|\frac{1}{S_g} - \frac{1}{S_{\text{max}}}\right|, \left|\frac{1}{S_g} - \frac{1}{S_{\text{min}}}\right|\right) \cdot \operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) \tag{31}$$
and exploiting Assumption 1. Indeed it is immediate to check that
$$U_S^*(S_g) \geq U_S(S_g) \qquad \forall S_g \in \mathbb{R}_+$$
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_{min} independent virtual input locations $x_{i,j}$ by means of density μ, i.e., each i generates
$$x_{i,j} \sim \mu \qquad \text{where} \qquad j = 1, \ldots, S_{\text{min}} \, . \tag{32}$$
b) then each i locally computes
$$C_{i,j}^E := \left[\phi_1(x_{i,j}), \ldots, \phi_E(x_{i,j})\right],$$
$$V_{r,i}^* := \left(\frac{1}{S_{\text{max}}} \operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right) + \frac{1}{S_{\text{max}}} \sum_{j=1}^{S_{\text{min}}} \left(C_{i,j}^E\right)^T C_{i,j}^E\right)^{-1} \tag{33}$$
$$U_{C,i}^* := \left(I - \frac{1}{S_{\text{min}}} \sum_{j=1}^{S_{\text{min}}} \left(C_{i,j}^E\right)^T C_{i,j}^E\right) \, , \tag{34}$$
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
$$S_g^* := \arg \min_{S_g \in \mathbb{R}_+} \mathcal{B}^*(S_g) \tag{35}$$
where
$$\begin{aligned} \mathcal{B}^*(S_g) \, := \, & (\gamma_b S_{\text{max}} + 1) \cdot \frac{1}{S} \sum_{i=1}^{S} \Big(\left\| V_{r,i}^* U_S^*(S_g) \, b_d(S_g) \right\|_2 \\ & \qquad + \left\| V_{r,i}^* U_{C,i}^* b_d(S_g) \right\|_2 \Big) \\ & + (\gamma_a S_{\text{max}}) \cdot \frac{1}{S} \sum_{i=1}^{S} \left\| y_i - C_i^E b_d(S_g) \right\|_2 \, . \end{aligned} \tag{36}$$

Intuitively, thus, agents try to minimize a pessimistic estimate $\mathcal{B}^*(S_g)$ of $\mathcal{B}(S_g)$ instead of $\mathcal{B}(S_g)$ itself. The complete algorithm is then reported in Alg. 1, solving problem (36) by gridding, i.e., selecting the best S_g from a set of candidates $S_g^{(1)}, \ldots, S_g^{(P)}$.

B. Calibration of the Number of Eigenfunctions

The maximum admissible value for E is upper bounded by computational complexity and transmission capability constraints. Assuming $\overline{E}$ to be this maximum value, the usage of a naïve strategy like $E = \overline{E}$ could lead to communicate more than necessary. In the following Alg. 2 we offer a practical and general guideline for the choice of E exploiting pessimistic bounds on the approximation error $\|b_c - b_r\|_1$.

From a practical point of view, Alg. 2 returns a number E assuring the operator that the normalized approximation error $\frac{\|b_c - b_r\|_2}{\|f_\mu\|_2}$ is smaller than a certain threshold. The algorithm is derived from the consideration that inequality (56) in the proof of Prop. 3 implies
$$\|b_c - b_r\|_2 \leq \gamma_a \sum_{i=1}^{S} \|y_i - C_i b_r\|_2 \tag{41}$$
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 (41) and to substitute 3σ with $\max_i \|y_i\|_2$ in (40).

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 $S_g^{(1)}, \dots, S_g^{(P)}$ and relative matrices $U_S^*\left(S_g^{(1)}\right), \dots, U_S^*\left(S_g^{(P)}\right)$. In addition, each sensor $i$ locally generates $S_{\text{min}}$ independent virtual input locations $x_{i,j}$, $j = 1, \dots, S_{\text{min}}$ by means of density $\mu$, from which it computes $C_{i,j}^E$, $V_{r,i}^*$ and $U_{C,i}^*$.
On-line and distributed work:
(distributed step) sensors distributedly compute, by means of average consensus protocols, the $E$-dimensional vector
    $\mathcal{Z} := \frac{1}{S} \sum_{i=1}^{S} \left(C_i^E\right)^T y_i \qquad (37)$
(local step) each sensor $i$ computes the $P$ versions of the estimator (18), namely $b_d\left(S_g^{(p)}\right) = V_d\left(S_g^{(p)}\right)\mathcal{Z}$, for $p = 1, \dots, P$.
(local step) each sensor $i$ computes the local $P$ auxiliary scalars, for $p = 1, \dots, P$
    $\mathcal{B}_i^*\left(S_g^{(p)}\right) := (\gamma_b S_{\text{max}} + 1)\left\| V_{r,i}^* U_S^*\left(S_g^{(p)}\right) b_d\left(S_g^{(p)}\right)\right\|_2 + (\gamma_b S_{\text{max}} + 1)\left\| V_{r,i}^* U_{C,i}^* b_d\left(S_g^{(p)}\right)\right\|_2 + (\gamma_a S_{\text{max}}) \cdot \left\| y_i - C_i^E b_d\left(S_g^{(p)}\right)\right\|_2$
(distributed step) sensors distributedly compute, by means of average consensus protocols, the $P$ scalars, for $p = 1, \dots, P$
    $\mathcal{B}^*\left(S_g^{(p)}\right) := \frac{1}{S} \sum_{i=1}^{S} \mathcal{B}_i^*\left(S_g^{(p)}\right) \qquad (38)$
(local step) each sensor $i$ computes $S_g^* = S_g^{(p^*)}$ where
    $(p^*) = \arg\min_{(p)} \mathcal{B}^*\left(S_g^{(p)}\right) \qquad (39)$
```

Algorithm 2 Calibration of the number of eigenfunctions

```
assume the knowledge of a lower bound on the energy of the unknown signal $f_\mu$, indicated with $\min \|f_\mu\|_2$
choose a threshold $\delta$ for the maximal tolerable error $\frac{\|b_c - b_r\|_2}{\|f_\mu\|_2}$
compute the minimal value of $E$ s.t.
    $\frac{3\sigma S_{\text{max}} \gamma_a(E)}{\min \|f_\mu\|_2} \leq \delta \qquad (40)$
    where we highlighted the dependence of $\gamma_a$ on $E$.
```

We notice that, substituting $\min \|f_\mu\|_2$ with $\max_i \|y_i\|_2$ in (40), 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 (37). We also notice that high uncertainties on S lead to overestimations of E because of the approximation S_{max}.

VI. Numerical examples

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

$$f_\mu(x) = \sum_{n=1}^{100} \alpha_n \sin(\omega_n x) \tag{42}$$

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, $\text{SNR} := \dfrac{\text{var}(f_\mu)}{\sigma^2} \approx 2.5$. Moreover we consider the Gaussian kernel

$$K(x, x') = \exp\left(-\frac{(x - x')^2}{0.02}\right) \tag{43}$$

with the estimators (11) and (18) defined by $\gamma = 0.3$.

To show the effectiveness of the estimation strategy (18), a randomly generated realization of f_μ 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, its estimates reconstructed from b_c, and $b_d^{(\cdot)}\left(S_g^*\right)$ with S_g^* chosen by Alg. 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 $T^{-1}\left[b_d^{(a)}\right]$ and $T^{-1}\left[b_d^{(b)}\right]$ are close to the centralized estimate $T^{-1}[b_c]$ (where T^{-1}, given (7), corresponds to the map from a sequence of eigenfunctions weights to the corresponding function in $\mathcal{H}_K$).

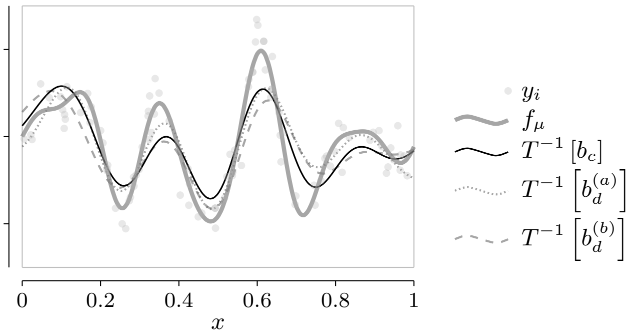
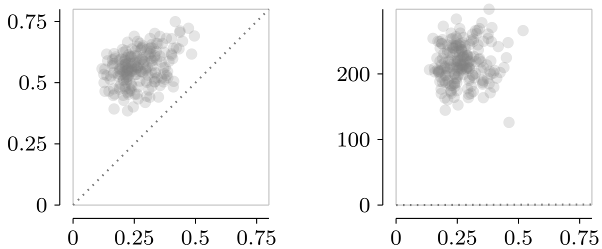


Fig. 1. Effectiveness of the estimation strategy (18) on a randomly generated f_μ, for various levels of uncertainty on S.

Despite this valuable property, bounds $\mathcal{B}^*$ are good indicators about the actual distance $\left\|b_d\left(S_g^*\right) - b_c\right\|_2$ only for the case (a) (low uncertainty on S), as Fig. 2 indicates. In this figure we generate 200 independent realizations of f_μ, then estimate each f_μ as before, and finally plot the actual distances $\left\|b_d^{(\cdot)} - b_c\right\|_2$ versus the obtained bounds $\mathcal{B}^*$. It is immediate 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_{max}. This implies that in general the accuracy of the bound is tightly connected with the accuracy of the knowledge on S.



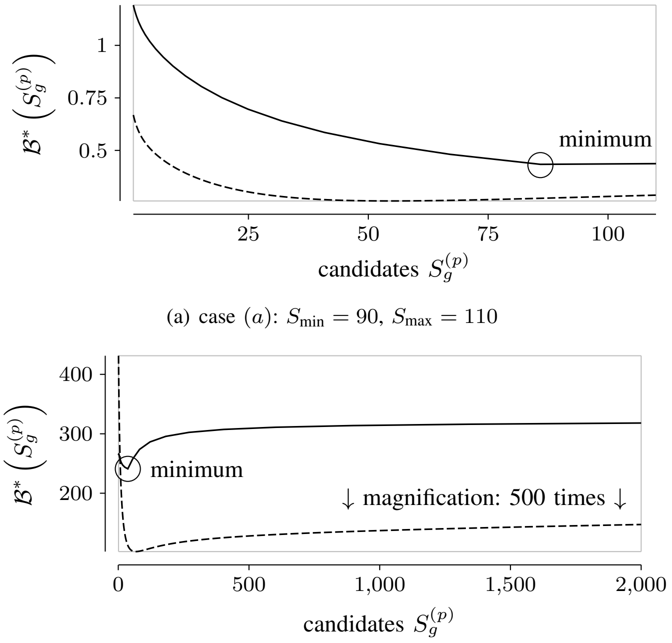
(a) case (a): $S_{\text{min}} = 90$, $S_{\text{max}} = 110$

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

Fig. 2. Actual distances $\left\|b_d\left(S_g^*\right) - b_c\right\|_2$ vs. bounds values $\mathcal{B}^*$ for different levels of uncertainty on S.

[2] This particular choice will be motivated later.

For sake of completeness, we show in Fig. 3 the values of the bounds $\mathcal{B}^*\left(S_g^{(p)}\right)$ defined in (38) associated to the experiment of Fig. 1, and the relative distances $\left\|b_d\left(S_g^{(p)}\right) - b_c\right\|_2$. It is possible to see how the qualitative behavior of curve $\mathcal{B}^*\left(S_g^{(p)}\right)$ is similar to the one of curve $\left\|b_d\left(S_g^{(p)}\right) - b_c\right\|_2$.



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

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

Fig. 3. Values of the bounds $\mathcal{B}^*\left(S_g^{(p)}\right)$ under different uncertainty levels on S for the experiment of Fig. (1) (solid lines), and relative values of the distances $\left\|b_d\left(S_g^{(p)}\right) - b_c\right\|_2$ (dashed lines). Circles on the solid lines indicate the optimal values $\mathcal{B}^*$. The dashed line in panel (b) has been magnified 300 times.

We then aim to check if it is better to use Alg. 1 or to try to directly try to estimate S. We thus compare the estimation performance obtainable with three different naïve strategies for the choice of S_g, namely $S_g^* = S_{\text{min}}$, $S_g^* = S_{\text{max}}$, $S_g^* = S_{\text{ave}} := \dfrac{S_{\text{min}} + S_{\text{max}}}{2}$. 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. to 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. to naïve strategies.

To numerically prove the last statement, we consider the previously generated 200 independent realizations of f_μ and the case $S_{\text{min}} = 20$ and $S_{\text{max}} = 2000$. We then plot in Fig. 4 the 100 points

$$\left(\left\|b_d\left(S_{\text{min}}\right) - b_c\right\|_2, \left\|b_d\left(S_g^*\right) - b_c\right\|_2\right) \tag{44}$$

$$\left(\left\|b_d\left(S_{\text{ave}}\right) - b_c\right\|_2, \left\|b_d\left(S_g^*\right) - b_c\right\|_2\right) \tag{45}$$

$$\left(\left\|b_d\left(S_{\text{max}}\right) - b_c\right\|_2, \left\|b_d\left(S_g^*\right) - b_c\right\|_2\right) . \tag{46}$$

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_gs. 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

$$\left(\left\|b_d\left(S_g^{\text{ora}}\right) - b_c\right\|_2, \left\|b_d\left(S_g^*\right) - b_c\right\|_2\right) \tag{47}$$

where S_g^{ora} are the optimal S_gs obtained exactly solving problem (19) (i.e., by using an oracle). 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.

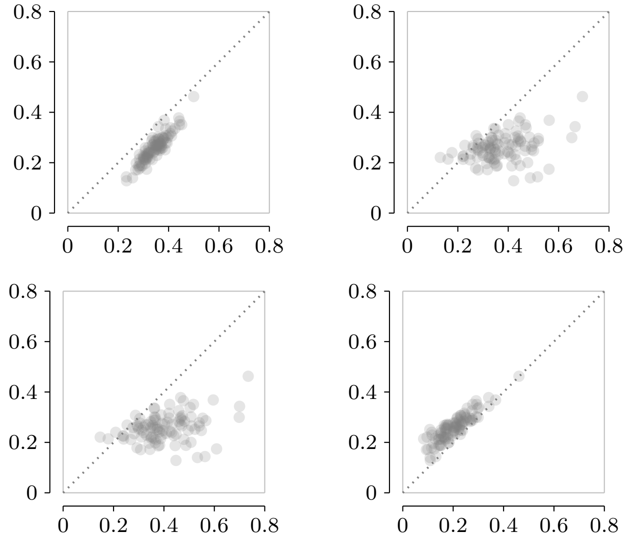


Fig. 4. Scatter plots to test the effectiveness of Alg. 1. Left-up panel: scatter plots of the points defined in (44). Right-up panel: points defined in (45). Left-down panel: points defined in (46). Right-down panel: points defined in (47). $S_{\text{min}} = 20$ and $S_{\text{max}} = 2000$.

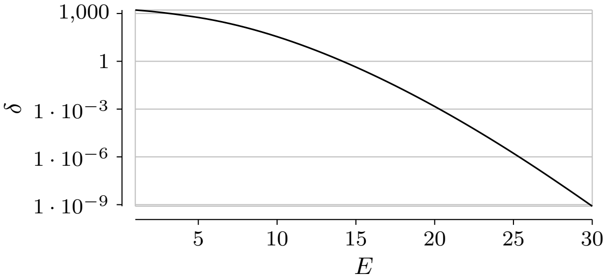


Fig. 5. Values of E returned by the on-line version of Alg. 2 fed with various choices of the threshold δ and applied to the experiment of Fig. 1 with $S_{\text{min}} = 20$ and $S_{\text{max}} = 2000$.

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 δ. We notice that the exponential decay of the bound is inherited by the exponential decay of eigenvalues λ_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.

VII. Conclusions

In this paper we analyze how to endow distributed nonparametric 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 have 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 (of Prop. 3) We rewrite (15) as

$$V_r^{-1} b_r = \mathcal{Z} \tag{48}$$

and (18) as

$$\left(V_r^{-1}+V_d^{-1}\left(S_g\right)-V_r^{-1}\right)b_d\left(S_g\right)=\mathcal{Z}\ . \tag{49}$$

Subtracting (49) to (48) we then obtain

$$b_r-b_d\left(S_g\right)=V_r\left(V_d^{-1}\left(S_g\right)-V_r^{-1}\right)b_d\left(S_g\right) \tag{50}$$

from which it immediately follows that

$$\left\|b_d-b_r\left(S_g\right)\right\|_2=\left\|V_r\left(V_d^{-1}\left(S_g\right)-V_r^{-1}\right)b_d\left(S_g\right)\right\|_2\ . \tag{51}$$

Defining then U_C and U_S by means of (25) and (26), it is immediate to check that $V_d^{-1}\left(S_g\right)-V_r^{-1}=U_S\left(S_g\right)+U_C$from which inequality (27) immediately follows.

To prove (28), we rewrite (15) as

$$\begin{aligned}\left(\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)+\sum_{i=1}^{S}C_i^TC_i\right)b_r+\left(\sum_{i=1}^{S}\left(C_i^E\right)^TC_i^E-\sum_{i=1}^{S}C_i^TC_i\right)b_r\\ =\sum_{i=1}^{S}C_i^Ty_i-\sum_{i=1}^{S}\left(C_i^{\backslash E}\right)^Ty_i\end{aligned} \tag{52}$$

and (11) as

$$\left(\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)+\sum_{i=1}^{S}C_i^TC_i\right)b_c=\sum_{i=1}^{S}C_i^Ty_i\ . \tag{53}$$

After subtracting (53) to (52), we obtain

$$\begin{aligned}\left(\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)+\sum_{i=1}^{S}C_i^TC_i\right)\left(b_c-b_r\right)=\\ =\left(\sum_{i=1}^{S}\left(C_i^E\right)^TC_i^E-\sum_{i=1}^{S}C_i^TC_i\right)b_r+\sum_{i=1}^{S}\left(C_i^{\backslash E}\right)^Ty_i\ .\end{aligned} \tag{54}$$

Substituting now each C_i in the right side of (54) with $C_i^E+C_i^{\backslash E}$, exploiting the fact that $C_i^{\backslash E}b_r=0$ (where 0 is an infinite dimensional vector of zeros), and properly collecting the various terms, we obtain

$$\begin{aligned}&b_c-b_r=\\ &\left(\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)+\sum_{i=1}^{S}C_i^TC_i\right)^{-1}\sum_{i=1}^{S}\left(C_i^{\backslash E}\right)^T\left(y_i-C_ib_r\right)\ .\end{aligned} \tag{55}$$

Since $\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)+\sum_{i=1}^{S}C_i^TC_i\geq\operatorname{diag}\left(\frac{\gamma}{\lambda_e}\right)$ (in a matricial positive definite sense), we obtain

$$\left\|b_c-b_r\right\|_2\leq\sum_{i=1}^{S}\left\|\operatorname{diag}\left(\frac{\lambda_e}{\gamma}\right)\left(C_i^{\backslash E}\right)^T\left(y_i-C_ib_r\right)\right\|_2\ . \tag{56}$$

Rewriting $y_i-C_ib_r$ as $y_i-C_i^Eb_d\left(S_g\right)+C_i^Eb_d\left(S_g\right)-C_i^Eb_r$ and using definitions (21) and (22), it follows immediately that

$$\begin{aligned}\left\|b_c-b_r\right\|_2&\leq\gamma_a\sum_{i=1}^{S}\left\|y_i-C_ib_d\left(S_g\right)\right\|_2+\gamma_b\sum_{i=1}^{S}\left\|b_r-b_d\left(S_g\right)\right\|_2\\ &\leq\gamma_a\sum_{i=1}^{S}\left\|y_i-C_ib_d\left(S_g\right)\right\|_2+\gamma_bS_{\max}\left\|b_r-b_d\left(S_g\right)\right\|_2.\end{aligned} \tag{57}$$

Notice that γ_a is finite since for every $x\in\mathcal{X}$ it holds that

$$\left\|\operatorname{diag}\left(\frac{\lambda_e}{\gamma}\right)C^{\backslash E}\left(x\right)\right\|_2^2\leq\sup_{x\in\mathcal{X},e\in\mathbb{N}_+}\phi_e\left(x\right)\cdot\sum_{e=E+1}^{+\infty}\frac{\lambda_e}{\gamma} \tag{58}$$

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

(28) can then be proved substituting (57) in

$$\left\|b_c-b_d\left(S_g\right)\right\|_2\leq\left\|b_c-b_r\right\|_2+\left\|b_r-b_d\left(S_g\right)\right\|_2\ . \tag{59}$$

References

[1] G. Pillonetto, A. Chiuso, and G. De Nicolao, "Prediction error identification of linear systems: A nonparametric gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291 – 305, February 2011.

[2] S. Smale and D.-X. Zhou, "Learning theory estimates via integral operators and their approximations," *Constructive approximation*, vol. 26, pp. 153–172, 2007.

[3] G. De Nicolao and G. Ferrari-Trecate, "Consistent identification of NARX models via Regularization Networks," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2045 – 2049, November 1999.

[4] G. Pillonetto and B. M. Bell, "Bayes and empirical Bayes semi-blind deconvolution using eigenfunctions of a prior covariance," *Automatica*, vol. 43, no. 10, pp. 1698–1712, October 2007.

[5] S. M. S. Rezeki, W. Chan, M. R. Haskard, D. E. Mulcahy, and D. E. Davey, "Realization of self-diagnosis and self-calibration strategies using conventional signal processing and fuzzy approach for distributed intelligent sensor systems," in *SPIE conference on Smart Structures and Materials 1999: Smart Electronics and MEMS*, 1999.

[6] M. Gopinathan, G. A. Pajunen, P. S. Neelakanta, , and M. Arockiasamy, "Linear quadratic distributed self-tuning control of vibration in a cantilever beam," in *SPIE conference on Smart Structures and Materials 1995: Smart Structures and Integrated Systems*, 1995.

[7] A. Karnik, A. Kumar, and V. Borkar, "Distributed self-tuning of sensor networks," *Wireless Networks*, vol. 12, pp. 531 – 544, 2006.

[8] Y. Li, J. Yu, M. Zhao, and K. Han, "Self-tuning distributed measurement fusion kalman filter," in *IEEE International Conference on Information and Automation*, 2010.

[9] G.-L. Tao, W. Wei, and Z.-L. Deng, "The self-tuning distributed information fusion wiener filter for the ARMA signals," in *8th World Congress on Intelligent Control and Automation*, 2010.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.

[12] F. Pérez-Cruz and S. R. Kulkarni, "Robust and low complexity distributed kernel least squares learning in sensor networks," *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 355 – 358, April 2010.

[13] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1856 – 1871, April 2009.

[14] ——, "Distributed kernel regression: An algorithm for training collaboratively," in *Proceedings of the IEEE Information Theory Workshop*, March 2006, pp. 332 – 336.

[15] P. Honeine, C. Richard, J. Bermudez, H. Snoussi, M. Essoloh, and F. Vincent, "Functional estimation in Hilbert space for distributed learning in wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 2861–2864.

[16] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-posed Problems*. Wiston, 1977.

[17] B. Schölkopf and A. J. Smola, *Learning with Kernels*. The MIT Press, 2002.

[18] K. Yosida, *Functional Analysis*. Springer-Verlag, 1965, vol. 123.

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

[20] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[21] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American Mathematical Society*, vol. 39, pp. 1 – 49, 2002.

[22] H. Zhu, C. K. I. Williams, R. Rohwer, and M. Morciniec, "Gaussian regression and optimal finite dimensional linear models," in *Neural Networks and Machine Learning*. Springer-Verlag, 1998.

[23] W. Nef, *Linear Algebra*. McGraw-Hill, 1967.

[24] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468 – 2481, 2012.