ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

An Empirical Bayes Approach for Distributed Estimation of Spatial Fields

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
An Empirical Bayes Approach for Distributed Estimation of Spatial Fields / Sasso, Francesco; Coluccia, Angelo; Notarstefano, Giuseppe. - ELETTRONICO. - (2018), pp. 8550231.2206-8550231.2211. (Intervento presentato al convegno 16th European Control Conference, ECC 2018 tenutosi a Limassol, CYPRUS nel JUN 12-15, 2018) [10.23919/ECC.2018.8550231].

Availability:
This version is available at: https://hdl.handle.net/11585/678743 since: 2019-02-28

Published:
DOI: http://doi.org/10.23919/ECC.2018.8550231

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

28 April 2024

This is the final peer-reviewed accepted manuscript of:

F. Sasso, A. Coluccia and G. Notarstefano, "An Empirical Bayes Approach for Distributed Estimation of Spatial Fields," 2018 European Control Conference (ECC), Limassol, 2018, pp. 2206-2211

The final published version is available online at: http://dx.doi.org/10.23919/ECC.2018.8550231

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

An Empirical Bayes Approach for Distributed Estimation of Spatial Fields

Francesco Sasso, Angelo Coluccia, *Senior Member, IEEE* and Giuseppe Notarstefano, *Member, IEEE*

***Abstract*— In this paper we consider a network of spatially distributed sensors which collect measurement samples of a spatial field, and aim at estimating in a distributed way (without any central coordinator) the entire field by suitably fusing all network data. We propose a general probabilistic model that can handle both partial knowledge of the physics generating the spatial field as well as a purely data-driven inference. Specifically, we adopt an Empirical Bayes approach in which the spatial field is modeled as a Gaussian Process, whose mean function is described by means of parametrized equations. We characterize the Empirical Bayes estimator when nodes are heterogeneous, i.e., perform a different number of measurements. Moreover, by exploiting the sparsity of both the covariance and the (parametrized) mean function of the Gaussian Process, we are able to design a distributed spatial field estimator. We corroborate the theoretical results with two numerical simulations: a stationary temperature field estimation in which the field is described by a partial differential (heat) equation, and a data driven inference in which the mean is parametrized by a cubic spline.**

I. INTRODUCTION

The presence of ubiquitous portable devices makes available a massive amount of spatially distributed measurements of several quantities, which can be used to estimate spatial fields of interest. While single measurements at each node can be inaccurate, fusion of data from multiple nodes gives the possibility to have a much more reliable estimate of the field. In order to avoid collecting and processing all data in a single computing unit, distributed estimation methods play an important role. Indeed, through this new computation paradigm, devices can improve their local measurement of the field and predict nearby values.

There are two main batches of literature related to the set-up investigated in this paper, namely Gaussian Process Regression and Kriging. Gaussian Process Regression has received significant attention in Artificial Intelligence and Machine Learning [1]. In a distributed context, efficient Gaussian Process Regression is discussed in [3], [4], while [2] presents a Bayesian approach for (distributed) spatio-temporal regression. Kriging interpolation techniques have been widely investigated in geostatistics [5], [6]. In this context, one of the first references on distributed estimation of a spatial field is [7], which formulates the estimation of a spatio-temporal field via Bayesian Universal Kriging. Reference [8] presents adaptive interpolation schemes for time-varying field estimation. In [9] distributed estimators are proposed for regression problems in sensor networks. In [10] a receding horizon approach is considered for estimation of spatially distributed systems modeled through an advection-diffusion Partial Differential Equation (PDE). A similar PDE is considered in [11] where the optimal sensor location problem is investigated for pollution monitoring. In [12] an environmental estimation method is developed. Finally, distributed estimation of a spatial field has application in cooperative control as highlighted in [13]–[15].

The contribution of the paper is twofold. First, we propose a probabilistic model for a network estimation set-up in which sensors take (possibly multiple) measurements of a spatial field and aim at cooperatively estimating the overall field by optimally fusing all data. Specifically, we propose an Empirical Bayesian framework in which the spatial field is modeled as a Gaussian Process having known covariance and, differently from existing works, non-zero mean depending on some hyperparameters that are estimated from the all measurements. In particular, the mean of the process is a parametric function that might encode some (deterministic) physical law modeling the expected behavior of the spatial field. This probabilistic model extends the one introduced in [16], [17] where the prior distribution did not depend on the spatial displacement of the nodes. Second, we design a distributed estimator of the spatial field relying on a Maximum Likelihood (ML) estimator of the hyperparameters and a local Maximum A Posteriori (MAP) estimator. We show how the ML estimator can be obtained by solving a structured optimization problem amenable to distributed computation. In particular, the optimization problem has a partitioned structure which calls for more efficient distributed optimization algorithms.

The paper is organized as follows. In Section II we present the spatial field estimation set-up. In Section III we introduce the proposed probabilistic model and give two example scenarios. In Section IV we derive the distributed estimator and, finally, in Section V we provide numerical simulations to show the performance of the distributed estimator applied to the two examples in Section III. .

II. SPATIAL FIELD ESTIMATION SET-UP

We consider a network of N spatially distributed sensors with local computation and communication capabilities. Nodes communicate according to a *communication graph* $\mathcal{G}^{\mathrm{cmm}} = (\{1, \ldots, N\}, \mathcal{E}^{\mathrm{cmm}})$. Each sensor has the possibility to perform multiple measurements of an external quantity at the point where it is located as represented in Fig. 1. The goal for the network is to suitably fuse the spatially distributed

---

Francesco Sasso, Angelo Coluccia and Giuseppe Notarstefano are with the Department of Engineering, Università del Salento, via Monteroni, 73100, Lecce, Italy, {name.lastname}@unisalento.it.

This result is part of a project that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

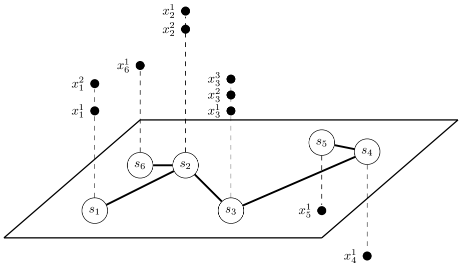

Fig. 1. Example of graph $\mathcal{G}^{\mathrm{cmm}}$ and its related observations $\mathbf{x}$.

measurements in order to estimate the spatial field. Formally, each sensor $i = 1, \dots, N$:

- is at a *measurement* (or *training*) point $s_i \in P \subset \mathbb{R}^d$;
- makes L_i *observations* $x_i^\ell \in X \subset \mathbb{R}$, $\ell = 1, \dots, L_i$.

Each observation x_i^ℓ is obtained as:

$$x_i^\ell = \theta(s_i) + \epsilon_i^\ell, \tag{1}$$

where all ϵ_i^ℓ are i.i.d. (noise) random variables with Gaussian distribution $\mathcal{N}(0, \sigma^2)$, while $\theta : P \mapsto X$ is an unknown vector valued function.

The unknown function $\theta(\cdot)$ represents a *spatial field* that sensors sample through noisy observations at the measurement points s_i, with ϵ_i^ℓ being the measurement noise.

For notational purpose, we introduce the following shorthands:

$$\begin{aligned} \boldsymbol{x}_i &= (x_i^1, \dots, x_i^{L_i})^\top, & \mathbf{x} &= (\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_N^\top)^\top, \\ \theta_i &= \theta(s_i), & \boldsymbol{\theta} &= (\theta_1, \dots, \theta_N)^\top. \end{aligned}$$

From the observation model (1) and the assumption that noises are i.i.d. Gaussian variables, it follows that

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\theta_i) = \prod_{i=1}^{N} \prod_{\ell=1}^{L_i} p(x_i^\ell|\theta_i), \tag{2.1}$$

$$p(x_i^\ell|\theta_i) \sim \mathcal{N}(\theta_i, \sigma^2). \tag{2.2}$$

Given the observations $\mathbf{x}$ and the graph $\mathcal{G}^{\mathrm{cmm}}$, our goal is to design a distributed algorithm allowing nodes to estimate the value of the spatial field $\theta(\cdot)$ in a set of *regression* (or *testing*) points $s_1^R, \dots, s_M^R \in P$.

It is worth noticing that the regression points can either coincide or not with the measurement points. In the first case, a node i in the network is trying to improve its local estimate of $\theta(s_i)$, while in the second one it wants to interpolate the field in a neighboring point where measurements are missing. In the following we will show how a single node can take advantage of all other measurements in the network by exchanging information only with neighboring nodes.

III. EMPIRICAL BAYES FRAMEWORK

In this section we introduce a probabilistic model allowing nodes, especially the ones with few measurements, to have a better estimate of the spatial field at their measurement points or in neighboring areas.

A. Probabilistic Model

We adopt a Bayesian model in which the spatial field $\theta(\cdot)$ is a *Gaussian Process* with mean function $\mu(\cdot)$ and covariance function $K(\cdot, \cdot)$, denoted by

$$\theta(\cdot) \sim \mathcal{GP}(\mu(\cdot), K(\cdot, \cdot)). \tag{3}$$

For the covariance function $K(\cdot, \cdot)$, known *kernel* functions can be used as customary. As for the mean function $\mu(\cdot)$, different from existing works where it is assumed to be null, we consider a more general parametric model which is meant to capture some prior knowledge. Since it is not realistic to assume the model (hyper)parameters to be known by nodes in the network, we propose an approach, based on the Empirical Bayes paradigm, in which hyperparameters will be estimated from the data through Maximum Likelihood (ML). As we will show through two example scenarios, this framework can model both a set-up in which prior knowledge encodes some (deterministic) physical law (but with unknown values of the physical parameters), and a data-driven inference scenario in which an interpolating function is used.

Formally, we introduce an interaction graph, $\mathcal{G}^{\mathtt{int}} = (\{1, \dots, N\}, \mathcal{E}^{\mathtt{int}})$, induced by the Gaussian Process. We denote $\mathcal{N}_i^{\mathtt{int}} = \{j : (i,j) \in \mathcal{E}^{\mathtt{int}}\}$ the set of neighbors of node i in the interaction graph. Consistently, we let $\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}} = \left[s_j\right]_{j \in \mathcal{N}_i^{\mathtt{int}}}$ be the vector of measurement points which are neighbors of point s_i, and $\boldsymbol{\mu}_{\mathcal{N}_i^{\mathtt{int}}} = \left[\mu(s_j)\right]_{j \in \mathcal{N}_i^{\mathtt{int}}}$ the vector of values of $\mu(\cdot)$ at those points.

We, thus, assume that

$$K(s_i, s_j) \neq 0 \iff (i,j) \in \mathcal{E}^{\mathtt{int}}, \tag{4}$$

and that $\mu(\cdot)$ is the (unique) solution of the following parametrized system of equations, termed *spatial dynamics*:

$$F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, \boldsymbol{\mu}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = 0, \qquad i = 1, \dots, N, \tag{5}$$

where the functions $F_i(\cdot, \cdot; \boldsymbol{\gamma})$ are known, while $\boldsymbol{\gamma} \in \Gamma$ is the unknown *hyperparameter* vector.

We point out that $\boldsymbol{\theta}$ is the vector of parameters of $\mathbf{x}$, and $\boldsymbol{\gamma}$ is the parameter of $\boldsymbol{\theta}$, and thus the name hyperparameter.

B. Examples of application scenarios

Here, we present two scenarios that highlight the flexibility of the proposed probabilistic framework. In the first one, we consider a set-up in which physical knowledge of the spatial field is available through a partial differential equation that can be discretized at the measurement points. In the second one, we show how interpolating curves can be used when there is no physical knowledge of the phenomenon.

1) Temperature Dynamics: We consider a scenario in which sensors want to estimate the temperature in a given environment. In this case, the mean function $\mu(\cdot)$ of the spatial field $\theta(\cdot)$ is expected to obey the following Heat Equation parametrized by $\boldsymbol{\gamma}$:

$$\sum_{\ell=1}^{d} \partial_{\ell\ell}^2 \mu(s,t) - \partial_t \mu(s,t) = w(s,t;\boldsymbol{\gamma}),$$

where $(s,t) \in P \times [0,T)$, ∂_ℓ is the partial derivative with respect to the ℓ-th component of s, and ∂_t is the partial derivative with respect to time t. The function $w(\cdot;\boldsymbol{\gamma})$ is also known as the heat source.

If one considers a thermostatic (equilibrium) condition, the Heat Equation turns out to be the so called Poisson Equation:

$$\sum_{\ell=1}^{d} \partial^2_{\ell\ell}\mu(s) = w(s;\boldsymbol{\gamma}), \quad s \in P \subset \mathbb{R}^d.$$

Now, in order to guarantee the uniqueness of $\mu(\cdot)$, we specify the boundary condition. For the sake of clarity, we consider the 1-dimensional case in which we want to monitor the temperature in a bar $P = [s_1, s_N]$. As a consequence, $\mu(\cdot)$ is uniquely determined by the following Cauchy Problem

$$\begin{cases} \mu''(s) = w(s;\boldsymbol{\gamma}), & s \in (s_1, s_N), \\ \mu(s_1) = w_1, \\ \mu(s_N) = w_N. \end{cases}$$

Then, the Poisson Equation can be discretized at the measurement points $s_1, \dots, s_N$, by using a suitable approximation rule as, e.g.,

$$\mu(s_i)'' \approx \frac{\mu(s_{i+1}) - 2\mu(s_i) + \mu(s_{i-1})}{\epsilon^2}, \quad i = 2, \dots, N-1,$$

where we have assumed, for simplicity, that the measurement points are uniformly spaced, i.e., $\epsilon = s_{i+1} - s_i$ for all $i = 1, \dots, N-1$. By plugging in the boundary conditions, the discretized system turns out to be

$$\begin{bmatrix} 1 & & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \mu(s_1) \\ \mu(s_2) \\ \vdots \\ \mu(s_{N-1}) \\ \mu(s_N) \end{bmatrix} = \begin{bmatrix} w_1 \\ \epsilon^2 w(s_2;\boldsymbol{\gamma}) \\ \vdots \\ \epsilon^2 w(s_{N-1};\boldsymbol{\gamma}) \\ w_N \end{bmatrix},$$

which has the sparsity of (5) once we set

$$\mathcal{N}_i^{\texttt{int}} = \begin{cases} \{i\}, & \text{for } i \in \{1, N\}, \\ \{i-1, i, i+1\} & \text{otherwise}, \end{cases}$$

and

$$F_i(\boldsymbol{s}_{\mathcal{N}_i^{\texttt{int}}}, \boldsymbol{\mu}_{\mathcal{N}_i^{\texttt{int}}}; \boldsymbol{\gamma}) = \begin{cases} \mu(s_i) - u_i, & \text{for } i \in \{1, N\}, \\ \frac{\mu(s_{i+1}) - 2\mu(s_i) + \mu(s_{i-1})}{\epsilon^2} - u(s_i;\boldsymbol{\gamma}) & \text{otherwise}. \end{cases}$$

We point out that, as depicted in Fig. 2, the discretization process may induce a set of neighbors, call it $\mathcal{N}_i^\mu$, while the covariance function may correlate with a broader set of agents, call it $\mathcal{N}_i^K \supset \mathcal{N}_i^\mu$. Clearly, we can set $\mathcal{N}_i^{\texttt{int}} = \mathcal{N}_i^K$.

2) Interpolating Dynamics: Here we consider a data-driven inference scenario in which the shape of the mean $\mu(\cdot)$ is described by

$$\mu(s) = \mathcal{I}(s;\boldsymbol{\gamma}), \quad s \in P,$$

where $\mathcal{I}(s;\boldsymbol{\gamma})$ is an interpolation function (e.g., a cubic spline) of the points $\{(s_i, \gamma_i)\}_i$, parametrized by $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)^\top \in \mathbb{R}^N$. This gives the possibility to fit the observations even if there is no information about the physics of the problem.

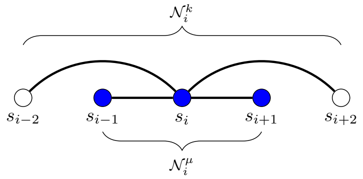

Fig. 2. Neighborhoods of agent i induced by the covariance function $K(\cdot,\cdot)$ and by the mean function $\mu(\cdot)$, denoted by $\mathcal{N}_i^k$ and $\mathcal{N}_i^\mu$ respectively.

IV. DISTRIBUTED EMPIRICAL BAYES ESTIMATOR

In this section we develop our distributed estimation approach following the Empirical Bayes framework introduced in the previous section. Specifically, we derive the Maximum Likelihood (ML) estimator of the hyperparameter and the Maximum A Posteriori (MAP) estimator of the spatial field. Then, we show how to recast the ML problem into an equivalent optimization problem with a partitioned structure that is amenable to distributed computation. Finally, we provide a local procedure allowing each node to compute the MAP estimator of the field at its location or to perform a regression at nearby points.

We start by introducing some useful notation:

$$\mu_i = \mu(s_i), \qquad \boldsymbol{\mu} = (\mu_1, \dots, \mu_N)^\top,$$
$$\mu_m^R = \mu(s_m^R), \qquad \boldsymbol{\mu}^R = (\mu_1^R, \dots, \mu_M^R)^\top.$$

When needed, we highlight the dependence of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}^R$ on $\boldsymbol{\gamma}$ (from (5)) by writing $\boldsymbol{\mu}_{\boldsymbol{\gamma}}$ and $\boldsymbol{\mu}_{\boldsymbol{\gamma}}^R$ respectively. Moreover, we define the matrices $K_{ss} \in \mathbb{R}^{N \times N}$ with entries $[K_{ss}]_{ij} = K(s_i, s_j)$, $K_{s^R s} \in \mathbb{R}^{M \times N}$ with entries $[K_{s^R s}]_{mi} = K(s_m^R, s_i)$, and consistently K_{ss^R} and $K_{s^R s^R}$. We also introduce

$$\overline{x}_i = \frac{1}{L_i} \sum_{\ell=1}^{L_i} x_i^\ell, \qquad \overline{\mathbf{x}} = (\overline{x}_1, \dots, \overline{x}_N)^\top.$$

Finally,

$$[D]_{ij} = \delta_{ij} \frac{\sigma^2}{L_i} \in \mathbb{R}^{N \times N},$$

where δ_{ij} is the Kronecker delta (i.e., $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise).

A. Distributed ML Estimator

The ML hyperparameter estimator is given by

$$\boldsymbol{\gamma}_{\text{ML}} = \operatorname*{argmax}_{\boldsymbol{\gamma} \in \Gamma} p(\mathbf{x};\boldsymbol{\gamma}). \tag{6}$$

We are now ready to state the first result characterizing the structure of optimization problem (6).

Theorem 4.1: The ML hyperparameter estimator $\boldsymbol{\gamma}_{\text{ML}}$ in (6) can be equivalently obtained by solving the following minimization problem

$$\min_{\boldsymbol{\gamma} \in \Gamma} (\boldsymbol{\mu}_{\boldsymbol{\gamma}} - \overline{\mathbf{x}})^\top (K_{ss} + D)^{-1} (\boldsymbol{\mu}_{\boldsymbol{\gamma}} - \overline{\mathbf{x}}). \tag{7}$$

□

The proof will be provided in a forthcoming document.

Next, we introduce the distributed formulation of the optimization problem (6). We start by observing that the most important consequence of (4) is that matrix K_{ss} has the same sparsity as graph $\mathcal{G}^{\mathtt{int}}$ and so does $K_{ss} + D$. However, looking at equation (7), we notice that $(K_{ss}+D)^{-1}$ is not necessarily a sparse matrix. In order to preserve the sparsity for a distributed computation, we then introduce a new optimization variable $\boldsymbol{z}$, so that $\boldsymbol{\gamma}_{\text{ML}}$ can be computed by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Gamma,\boldsymbol{z}\in\mathbb{R}^N} \quad & \boldsymbol{z}^\top (K_{ss}+D)\boldsymbol{z}, \\ \text{subj. to} \quad & (K_{ss}+D)\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\gamma}} - \overline{\mathbf{x}}. \end{aligned} \tag{8}$$

At this point, making explicit the relationship between $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$ due to (5), we can write (8) as

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Gamma,\boldsymbol{z},\boldsymbol{\mu}\in\mathbb{R}^N} \quad & \boldsymbol{z}^\top (K_{ss}+D)\boldsymbol{z}, \\ \text{subj. to} \quad & (K_{ss}+D)\boldsymbol{z} = \boldsymbol{\mu} - \overline{\mathbf{x}}, \\ & F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, \boldsymbol{\mu}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = 0, \quad i = 1,\dots,N. \end{aligned} \tag{9}$$

Defining the functions

$$f_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}) = \frac{\sigma^2}{L_i} z_i^2 + \sum_{j\in\mathcal{N}_i^{\mathtt{int}}} z_i K(s_i, s_j) z_j, \qquad i = 1,\dots,N,$$

$$g_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}) = \frac{\sigma^2}{L_i} z_i + \overline{x}_i + \sum_{j\in\mathcal{N}_i^{\mathtt{int}}} K(s_i, s_j) z_j, \quad i = 1,\dots,N.$$

we can rewrite (9) as

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Gamma,\boldsymbol{z},\boldsymbol{\mu}\in\mathbb{R}^N} \quad & \sum_{i=1}^N f_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}), \\ \text{subj. to} \quad & g_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}) = \mu_i, \\ & F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, \boldsymbol{\mu}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = 0, \ i = 1,\dots,N. \end{aligned} \tag{10}$$

which is a formulation of problem (7) amenable to distributed solution.

Due to this formulation, the solution $(\boldsymbol{\gamma}_{\text{ML}}, \boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}, \boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}})$ of (9) can be computed by solving an optimization problem that has a separable cost (i.e., the sum of N local costs). Available distributed optimization algorithms can be adopted to this aim, e.g., [18], [19], [20]. The special sparsity of cost function and constraints could be exploited by using techniques as the one proposed in [21].

Remark 4.2: A special case is the one in which we have

$$F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, \boldsymbol{\mu}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, \mu_i; \boldsymbol{\gamma}), \quad i = 1,\dots,N.$$

In this case, if we define the functions

$$G_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = F_i(\boldsymbol{s}_{\mathcal{N}_i^{\mathtt{int}}}, g_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}); \boldsymbol{\gamma}), \qquad i = 1,\dots,N,$$

$\boldsymbol{\gamma}_{\text{ML}}$ can be computed by solving the optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\Gamma,\boldsymbol{z}\in\mathbb{R}^N} \quad & \sum_{i=1}^N f_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}), \\ \text{subj. to} \quad & G_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = 0, \quad i = 1,\dots,N, \end{aligned} \tag{11}$$

which has a simpler structure that can be exploited to speed up the calculation. Moreover, when $\mu(\cdot)$ is an explicit function of $\boldsymbol{\gamma}$, i.e., $\mu(\cdot) = \mu(\cdot; \boldsymbol{\gamma})$, the constraints in (11) simplify as $G_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}; \boldsymbol{\gamma}) = g_i(\boldsymbol{z}_{\mathcal{N}_i^{\mathtt{int}}}) - \mu(s_i; \boldsymbol{\gamma})$. □

The vector $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}$ gives the evaluation of the Gaussian Process mean $\mu(\cdot)$ in the measurement points. Once a value of $\boldsymbol{\gamma}_{\text{ML}}$ is available, computing nodes located at the measurement points may interpolate $\mu(\cdot)$ at regression points s_m^R in their spatial proximity, i.e., $\left[\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}\right]_m$, as follows.

- If $\mu(\cdot)$ satisfies a (discretized) partial differential equation parametrized by $\boldsymbol{\gamma}$, then $\left[\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}\right]_m$ can be obtained by integrating the differential equation with hyperparameter $\boldsymbol{\gamma}_{\text{ML}}$ and suitable boundary conditions based on neighboring points in the interaction graph.
- If $\mu(\cdot)$ is an explicit function of $\boldsymbol{\gamma}$, then $\left[\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}\right]_m$ can be obtained by $\left[\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}\right]_m = \mu(s_m^R; \boldsymbol{\gamma}_{\text{ML}})$.
- Alternatively, an interpolation can be performed by using the set of neighboring points.

B. Local MAP Estimator

In this subsection we show how a node can locally compute the MAP estimator $\boldsymbol{\theta}^R_{\text{MAP}}$. Defining $\theta_i^R = \theta(s_i^R)$ and $\boldsymbol{\theta}^R = (\theta_1^R, \dots, \theta_M^R)^\top$, the MAP estimator in the regression points is given by

$$\boldsymbol{\theta}^R_{\text{MAP}} = \mathop{\text{argmax}}_{\boldsymbol{\theta}^R\in X^M\subset\mathbb{R}^M} p(\boldsymbol{\theta}^R | \mathbf{x}; \boldsymbol{\gamma}_{\text{ML}}). \tag{12}$$

Notice that, once a value for $\boldsymbol{\gamma}_{\text{ML}}$ is available, the estimation proceeds similarly as in a purely Bayesian set-up, where the prior distribution is *fully specified* and given by $\mathcal{N}(\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}, K_{s^R s^R})$. Specifically, the case in which the prior is known and each sensor i performs only one observation ($L_i = 1$) has been widely investigated in the literature [1].

In our work we generalize this classical framework in two ways. First, we consider a heterogeneous network set-up in which nodes perform a different number of observations, so that even those with few observations take advantage of nodes with more observations (especially neighboring ones). Second, this MAP estimator exploits the ML estimation of the hyperparameters, thus "optimally" adapting the prior to all network data. In the next theorem we provide a closed form expression of the posterior distribution, and, based on that, we derive a formulation of the MAP estimator which can be computed by each node in the network in a decentralized way (i.e., by collecting data from neighboring nodes in the interaction graph).

Theorem 4.3: The posterior distribution $p(\boldsymbol{\theta}^R | \mathbf{x}; \boldsymbol{\gamma}_{\text{ML}})$ is a multivariate Gaussian with mean $\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}} - K_{s^R s}\boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}$, where $\boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}$ is a solution of problem (10), and covariance matrix $K_{s^R s^R} - K_{s^R s}(K_{ss} + D)^{-1} K_{s s^R}$. Moreover, by defining $\mathcal{N}_m^E := \{j : K(s_m^R, s_j) \neq 0\}$, the MAP estimator at s_m^R is given by

$$\left[\boldsymbol{\theta}^R_{\text{MAP}}\right]_m = \left[\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\text{ML}}}\right]_m - \sum_{j\in\mathcal{N}_m^E} K(s_m^R, s_j) \left[\boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}\right]_j. \qquad \square$$

The proof will be provided in a forthcoming document.

It is worth noting that if a regression point coincides with a measurement point, i.e., $s_m^R = s_m$, then $\mathcal{N}_m^E = \mathcal{N}_i^{\texttt{int}}$ and

$$\left[\boldsymbol{\theta}_{\text{MAP}}^R\right]_i = \left[\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}\right]_i - \sum_{j \in \mathcal{N}_i^{\texttt{int}}} \left[K_{ss}\right]_{ij} \left[\boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}\right]_j .$$

For a regression point s_m^R that does not coincide with a measurement point, a computing node located at s_m^R, or equivalently one of the network nodes in the proximity, can obtain $[\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R]_m$ through the interpolation procedure described in the previous subsection, and collect $\left[\boldsymbol{z}_{\boldsymbol{\gamma}_{\text{ML}}}\right]_j$ from $j \in \mathcal{N}_m^E$. Notice that the sparsity of the kernel with respect to the entire space P implies that $\mathcal{N}_m^E$ typically contains a limited number of nodes.

Generally speaking, Theorem 4.3 provides an efficient way to compute the MAP estimator at each point. In fact, if one used the standard formula of the posterior (see [1]) the MAP estimator would be $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R - K_{s^R q}(K_{qq} + \sigma^2 I_L)^{-1}(\boldsymbol{\mu}_L - \mathbf{x})$ where $L = \sum_{i=1}^N L_i$, $q = (\underbrace{s_1, \dots, s_1}_{L_1}, \dots, \underbrace{s_N, \dots, s_N}_{L_N})^\top$ and $\boldsymbol{\mu}_L = (\underbrace{\mu(s_1), \dots, \mu(s_1)}_{L_1}, \dots, \underbrace{\mu(s_N), \dots, \mu(s_N)}_{L_N})^\top$, thus involving matrices of much larger size.

V. NUMERICAL SIMULATIONS

In this section we analyze the numerical results related to the two example scenarios introduced in Section III-B. We have used, in both cases, a kernel function known as Mahalanobis function (see [22]), which allows us to make K_{ss} sparse. Moreover, we took as regression points a fine grained discretization of the (one dimensional) set P to better visualize the regression in the entire spatial domain.

A. Temperature Dynamics

For this numerical simulation, we have considered $N = 10$ sensors uniformly spaced between $s_1 = 0$ and $s_N = 2\pi/3$. The vector $\mathbf{x}$ of observations has been generated according to the following Temperature dynamics,

$$\begin{cases} \mu''(s) = -A\omega^2 \sin(\omega s + \phi), & s \in (s_1, s_N), \\ \mu(s_1) = w_1, \\ \mu(s_N) = w_N. \end{cases}$$

where the right hand side of the first equation represents the heat source with $A = 6$, $\omega = 3$ and $\phi = 3$, and the boundary conditions are $w_1 = 3$ and $w_N = 0$.

This differential equation has a closed form solution $\theta(s) = A\sin(\omega s + \phi) + C_1 s + C_0$, with

$$C_1 := \frac{w_N + A\sin(\omega s_N + \phi) - w_1 - A\sin(\omega s_1 + \phi)}{s_N - s_1},$$
$$C_0 := w_1 - A\sin(\omega s_1 + \phi) - C_1 s_1,$$

so that the ℓ-th observation at s_i is given by $x_i^\ell = \theta(s_i) + \epsilon_i$.

In the estimation process we assume that the prior available information is the above family of Temperature dynamics with parameters $\boldsymbol{\gamma} = (A, \omega, \phi)^\top$ being the hyperparameters of our framework. Once $\boldsymbol{\gamma}_{\text{ML}}$ has been obtained, $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R$ can be computed by integrating the Cauchy problem.

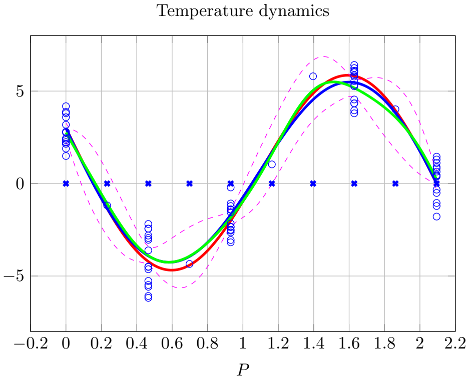

Fig. 3. Temperature dynamics scenario: the measurement points s_i are marked with a cross, while the observations $\mathbf{x}$ with circles. The red line is the true mean function $\mu(\cdot)$, the blue line is $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R$ and the green line is the posterior mean $\boldsymbol{\theta}_{\text{MAP}}^R$. In magenta we represent the 95% confidence bound for the posterior mean.

The result of the simulation is depicted in Fig. 3, where the estimated (process) prior mean in the regression points, $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R$, is depicted in blue, the posterior mean $\boldsymbol{\theta}_{\text{MAP}}^R$ is shown in green and the true field $\theta(\cdot)$ in red. Remarkably, both the $\boldsymbol{\mu}_{\boldsymbol{\gamma}_{\text{ML}}}^R$ and $\boldsymbol{\theta}_{\text{MAP}}^R$ provide an accurate estimate of the spatial field $\theta(\cdot)$. As one would expect, at measurement points with a greater number of observations the local estimation is more accurate with a tighter confidence bound.

B. Interpolating Dynamics

In this case we have considered $N = 12$ sensors spaced non-uniformly between $s_1 = -15$ and $s_N = 14$. The vector $\mathbf{x}$ of observations has been generated according to a spatial field $\theta(\cdot)$ being a quadratic polynomial, i.e.,

$$\theta(s) = as^2 + bs + c, \tag{13}$$

where we have chosen $a = 0.1$, $b = 0.1$ and $c = 10$.

Differently from the previous numerical example, during the estimation process we have used as parametrized mean function (prior) an interpolating curve that does not match (13). In fact we have used the family of *natural cubic splines*. Specifically, we have taken $N = 12$ measurement points $s_1, \dots, s_{12}$ clustered in five subregions. Then we have chosen as control points c_j of the spline one point for each cluster, namely $c_1 = s_1$, $c_2 = s_3$, $c_3 = s_5$, $c_4 = s_7$ and $c_5 = s_{12}$.

Specifically, the spline function is defined as

$$\mu(s) = \begin{cases} p_1(s), & \text{if } s < c_1, \\ p_j(s), & \text{if } s \in [c_j, c_{j+1}), \\ p_{N-1}(s), & \text{if } s \geq c_5, \end{cases}$$

where each cubic polynomial $p_j(\cdot)$ satisfies the following conditions:

$$\begin{cases} p_j(c_j) = \gamma_j, \quad p_j(c_{j+1}) = \gamma_{j+1}, & j = 1, \dots, 4, \\ p'_{j-1}(c_j) = p'_j(c_j), & j = 2, \dots, 4, \\ p''_{j-1}(c_j) = p''_j(c_j), & j = 2, \dots, 4 \\ p''_1(c_1) = 0, \quad p''_4(c_5) = 0, \end{cases}$$

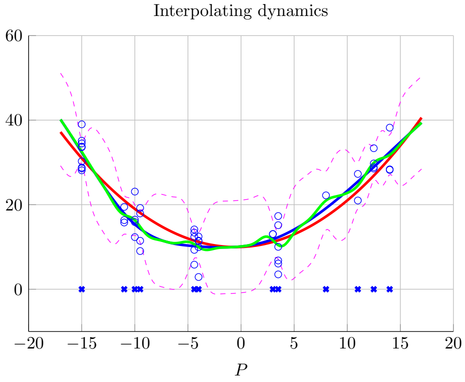

Fig. 4. Interpolating dynamics scenario: the measurement points s_i are labeled with a cross, while the observations $\mathbf{x}$ are marked with circles. The red line is the true mean function $\mu(\cdot)$, the blue line is $\boldsymbol{\mu}^R_{\boldsymbol{\gamma}_{\mathrm{ML}}}$ and the green line is the posterior mean $\boldsymbol{\theta}^R_{\mathrm{MAP}}$. In magenta we represent the 95% confidence interval relative to the posterior mean.

with $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_5)^T$.

The result of the simulation is depicted in Fig. 4. As in the previous example, the estimate is more accurate in those points with an higher number of observations. Moreover, we can appreciate the capability of the Interpolating dynamics of fitting the spatial field $\theta(\cdot)$ near the measurement points.

VI. CONCLUSIONS

In this paper we have proposed a novel probabilistic framework for distributed estimation of a spatial field in large-scale sensor networks, wherein each node performs multiple, local measurements. The proposed set-up is based on an Empirical Bayes approach in which the spatial field is modeled as a Gaussian Process with known covariance and unknown, but parametrized, mean. Specifically, we suppose that the mean function at the measurement points satisfies a set of equations (encoding some prior knowledge of the mean shape) parametrized by the unknown hyperparameters. The Empirical Bayes estimation procedure consists of two parts: the computation of the ML estimate of the hyperparameters, and the computation of the MAP estimate of the field into regression points. In particular, we have shown that multiple observations improve the estimation accuracy by reducing the posterior variance, and we have also provided a sparse formulation of the ML optimization problem, which is amenable to distributed solution. Thus, each sensor optimally fuses information from the network in a distributed way thus improving its own MAP estimate of the field. Numerical simulations for the proposed scheme have been presented for two example scenarios. In the first one, the temperature field in a thermostatic bar is estimated by using the Heat Equation as a prior knowledge for the mean. In the second one, interpolating curves have been used in a data-driven inference scenario with no physical knowledge of the field.

REFERENCES

[1] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[2] Y. Xu, J. Choi, S. Dass, and T. Maiti, “Bayesian prediction and adaptive sampling algorithms for mobile sensor networks,” in *IEEE American Control Conference (ACC)*, 2011, pp. 4195–4200.

[3] M. Todescato, A. Dalla Libera, R. Carli, G. Pillonetto, and L. Schenato, “Distributed Kalman filtering for time-space Gaussian processes,” *IFAC Proceedings Volumes*, 2017.

[4] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, “Machine learning meets Kalman filtering,” in *IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4594–4599.

[5] N. Cressie, *Statistics for spatial data*. John Wiley & Sons, 2015.

[6] M. L. Stein, *Interpolation of spatial data: some theory for Kriging*. Springer Science & Business Media, 2012.

[7] J. Cortés, “Distributed kriged Kalman filter for spatial estimation,” *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, 2009.

[8] S. Martinez, “Distributed interpolation schemes for field estimation by mobile sensor networks,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 491–500, 2010.

[9] 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.

[10] A. Kozma, J. Andersson, C. Savorgnan, and M. Diehl, “Distributed multiple shooting for optimal control of large interconnected systems,” *IFAC Proceedings Volumes*, vol. 45, no. 15, pp. 143–147, 2012.

[11] D. Georges, “Optimal sensor location and mobile sensor crowd modeling for environmental monitoring,” *IFAC Proceedings Volumes*, 2017.

[12] M. L. Elwin, R. A. Freeman, and K. M. Lynch, “Environmental estimation with distributed finite element agents,” in *IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 5918–5924.

[13] R. Graham and J. Cortés, “Spatial statistics and distributed estimation by robotic sensor networks,” in *IEEE American Control Conference (ACC)*, 2010, pp. 2422–2427.

[14] J. Le Ny and G. J. Pappas, “On trajectory optimization for active sensing in Gaussian process models,” in *48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference. CDC/CCC*, 2009, pp. 6286–6292.

[15] B. M. Bell and G. Pillonetto, “A distributed Kalman smoother,” *IFAC Proceedings Volumes*, vol. 42, no. 20, pp. 132–137, 2009.

[16] A. Coluccia and G. Notarstefano, “Distributed estimation of binary event probabilities via hierarchical Bayes and dual decomposition,” in *IEEE 52nd Conference on Decision and Control (CDC)*, 2013, pp. 6753–6758.

[17] A. Coluccia and G. Notarstefano, “A Bayesian framework for distributed estimation of arrival rates in asynchronous networks,” *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3984–3996, 2016.

[18] R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, “Analysis of Newton-Raphson consensus for multi-agent convex optimization under asynchronous and lossy communications,” in *54th IEEE Conference on Decision and Control*, 2015, pp. 418–424.

[19] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[20] Y. Sun, G. Scutari, and D. Palomar, “Distributed nonconvex multiagent optimization over time-varying networks,” in *IEEE 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 788–794.

[21] R. Carli and G. Notarstefano, “Distributed partition-based optimization via dual decomposition,” in *IEEE 52nd Conference on Decision and Control (CDC)*, 2013, pp. 2979–2984.

[22] A. Melkumyan and F. Ramos, “A sparse covariance function for exact Gaussian process inference in large datasets,” in *21st international jont conference on Artifical intelligence*. Morgan Kaufmann Publishers Inc., 2009, pp. 1936–1942.