Distributed Gaussian Process Regression for Mobile Sensor Networks Under Localization Uncertainty

Sungjoon Choi, Mahdi Jadaliha, Jongeun Choi, and Songhwai Oh

***Abstract*— In this paper, we propose distributed Gaussian process regression for resource-constrained mobile sensor networks under localization uncertainty. The proposed distributed algorithm, which combines Jacobi over-relaxation (JOR) and discrete-time average consensus (DAC), can effectively handle localization error as well as limited communication range and computation capabilities of mobile sensor networks. The performance of the proposed method is verified in numerical simulations against the centralized maximum a posteriori solution and the quick-and-dirty solution. We show that the proposed method outperforms the quick-and-dirty solution and achieves the accuracy which is close to the centralized solution.**

I. Introduction

Recent dramatic climatic changes, due to increasing greenhouse gas concentrations, stratospheric ozone depletion, and tropical deforestation, are great threats to the environment and our society [1]. The advances in embedded processors and mobile sensor networks (MSN) technologies allow a number of important and successful applications in environmental monitoring and prediction such as monitoring complex interactions in wildlife habitats and disaster management of harmful algal blooms in water bodies [2], [3].

Recently, it has been demonstrated that a stochastic process, such as a Gaussian process, is very efficient for modeling spatio-temporal phenomena of changing environments. Mysorewala et al. [4] combined a neural network and an extended Kalman filter (EKF) with greedy search heuristics for developing a distributed multi-scale sampling strategy for environmental monitoring. The use of EKF allowed the method to handle localization and measurement uncertainties. However, the proposed method is based on a parametric model and it is difficult to apply the method directly to highly complex time-varying real-world situations.

In this paper, we develop a distributed estimation method which is robust against localization and measurement errors using Gaussian process regression (GPR) [5], a nonparametric Bayesian regression method. GPR can provide both an estimate for an unseen input and the uncertainty (variance) about the estimate. A capability of assessing the uncertainty about its prediction is an attractive feature of GPR. Hence, GPR has been adapted to make statistical inferences on geostatistical and environmental data and applied to robot navigation [6]–[8] and sensor placement [9], [10]. However, GPR suffers from two major drawbacks: heavy computational cost and the difficulty of handling localization errors.

The computational complexity of the original form of GPR is $O(n^3)$, where n is the number of measurements, for the computation of an inverse of the covariance matrix. A number of approximation schemes have been proposed to reduce the computational complexity of GPR, including [11]–[15]. In particular, Xu et al. [16] proposed Gaussian process regression based on truncated observations for mobile sensor networks with limited memory and computational power.

For environmental monitoring, better mapping of the environment is possible when accurate sensing locations are available. However, there can be many situations where sensing locations cannot be measured accurately, e.g., GPS denied areas. In addition, there could be significant localization errors from inexpensive GPS receivers. A number of localization algorithms have been proposed to address this issue, including [17]–[20]. Oguz-Ekim et al. [19] iteratively maximized likelihoods of position estimates given measurements and Karlsson et al. [20] applied particle filtering for surface and underwater navigation as a supplement to the GPS.

A mobile sensor network consists of a number of resource-constrained agents with limited processing power, communication bandwidth, and battery capacity, to name a few. These limitations play an important role in designing an application using mobile sensor networks [2]. In order to handle these physical constraints and take a full capability as a team, it is important to process information in a distributed manner [2], [21]–[23]. For example, in [23], a distributed learning and control algorithm is developed for mobile sensor networks for estimating an unknown field of interest from noisy measurements and coordinating multiple agents to discover peaks of the unknown field. In [24], a distributed GPR algorithm is proposed using a compactly supported covariance functions. However, in both cases, it is assumed that there is no localization error.

Jadaliha et al. [25] incorporated the localization uncertainty into the Gaussian process regression framework. Since the proposed predictive mean and variance estimators have no closed-form solutions, they suggested two approximation schemes, Laplace approximation and Monte Carlo importance sampling. They also proposed a simple Laplace approximation method which uses the maximum a posteriori (MAP) estimator of noisy position reports. In this paper, we extend

S. Choi and S. Oh are with the School of Electrical Engineering and Computer Science and ASRI, Seoul National University, Seoul 151-744, Korea (e-mail: {sungjoon.choi, songhwai.oh}@cpslab.snu.ac.kr).

M. Jadaliha is with the Department of Mechanical Engineering, Michigan State University, MI 48824, USA (e-mail: jadaliha@egr.msu.edu).

J. Choi is with the Department of Mechanical Engineering and Electrical and Computer Engineering, Michigan State University, MI 48824, USA (e-mail: jchoi@egr.msu.edu).

Preprint submitted to 52nd IEEE Conference on Decision and Control .
Received March 6, 2013.

[25] by developing a distributed version of GPR, which can handle both localization and measurement uncertainties, such that it can be suitable for resource-constrained mobile sensor networks.

The remainder of this paper is organized as follows. In Section II, Gaussian processes are briefly described. In Section III and Section IV, we provide a mathematical formulation which incorporates both localization and measurement errors into a single Bayesian framework. We also introduce an approximation algorithm for computing the predictive mean estimator (PME) and predictive variance estimator (PVE). In Section V, we propose a distributed Gaussian process algorithm for computing PME and PVE. The performance of the proposed prediction algorithm is extensively evaluated in Section VI.

II. Gaussian Process Regression

A Gaussian process (GP) is completely specified by its mean function and covariance function and it is formally defined as a collection of random variables, any finite number of which have a joint Gaussian distribution [5]. Let us denote the mean function by $m(x)$ and the covariance function by $k(x, x')$ for a Gaussian process $f(x)$ describing an environmental parameter. $f(x)$ can be represented as:

$$f(x) \sim GP(m(x),\ k(x, x')). \tag{1}$$

Suppose that $x \in \mathbb{R}^{n_x}$ is an input and $y \in \mathbb{R}$ is an output (or a target), such that $y = f(x) + w$, where w is a white Gaussian noise. When the target y is continuous (respectively, discrete), we have a regression (respectively, classification) problem. In this paper, we assume y takes a continuous value. Suppose that there are n observations, $\{(x_i, y_i)|i = 1, \cdots, n\}$. Given observations, Gaussian process regression (GPR) can predict an output $y_\star$ for a new input vector $x_\star$.

For notational simplicity, we assume a zero-mean Gaussian process. A Gaussian process with a nonzero mean can be treated by a change of variables. Even without a change of variables, this is not a drastic limitation, since the mean of the posterior process is not confined to zero [5]. There are a number of choices for the covariance function and the widely used covariance function is the squared exponential (SE) kernel:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{||x - x'||^2}{2\sigma_x^2}\right), \tag{2}$$

where σ_f^2 and σ_x^2 are hyperparameters which can be estimated by maximizing the log-likelihood. The log-likelihood can be computed as follows.

$$\begin{aligned}\log P(y|x, \sigma_w) &= \frac{1}{2} y^T (K + \sigma_w^2 I)^{-1} y \\ &- \frac{1}{2}\log|K + \sigma_w^2 I| - \frac{n}{2}\log 2\pi.\end{aligned} \tag{3}$$

Assuming that $y_i = f(x_i) + w_i$ and $w_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_w^2)$, the covariance between y_i and y_j can be computed as

$$\mathbf{cov}(y^{(p)}, y^{(q)}) = k(x^{(p)}, x^{(q)}) + \sigma_w^2 \delta_{pq} \tag{4}$$

and we represent the covariance in the following matrix form

$$\mathbf{cov}(\mathbf{y}) = K(\mathbf{x}, \mathbf{x}) + \sigma_w^2 I, \tag{5}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathbf{x} = (x_1^T, \ldots, x_n^T)^T$, and $K(\mathbf{x}, \mathbf{x})$ is the covariance matrix computed from n data points.

Let $D = \{(x_i, y_i) \mid i = 1, \cdots, n\}$ be a set of input-output pairs. The conditional distribution of $y_\star$ at a new input $x_\star$ given data becomes

$$y_\star|D \sim N(\mu_\star(x_\star|D),\ \sigma_\star^2(x_\star|D)) \tag{6}$$

where

$$\mu_\star(x_\star|D) = k(x_\star, \mathbf{x})^T (K(\mathbf{x}, \mathbf{x}) + \sigma_w^2 I)^{-1} \mathbf{y} \tag{7}$$

and

$$\sigma_\star^2(x_\star|D) = \sigma_f^2 - k(x_\star, \mathbf{x})^T (K(\mathbf{x}, \mathbf{x}) + \sigma_w^2 I)^{-1} k(x_\star, \mathbf{x}). \tag{8}$$

Here, $k(x_\star, \mathbf{x}) \in \mathbb{R}^n$ is a covariance vector between $\mathbf{y}$ and $y_\star$.

Note that (7) and (8) require an inversion of a covariance matrix, which has complexity of $O(n^3)$. Considering limited capabilities of mobile sensor networks, the computation can be prohibitive when the number of measurements becomes large. A number of approximation methods have been proposed to address this issue [11]–[15]. A distributed algorithm, such as the one described in Section V, can be considered as another solution to reduce the computational demand.

III. GPR Under Localization Uncertainty

In a realistic situation, acquiring samples $\{x_i, y_i\}$ with exact localization for x_i is often impossible. As Gaussian processes incorporate a measurement noise naturally, it is desirable to incorporate a localization noise into Gaussian processes. In [25], Gaussian process regression was reformulated to incorporate noisy input data $\bar{\mathbf{x}}$, where $\bar{x}_i = x_i + v_i$, $\bar{\mathbf{x}} = \{\bar{x}_1, \ldots, \bar{x}_n\}$, and v_i is a zero-mean white Gaussian noise with variance σ_v^2. Let $D' = \{(\bar{x}_i, \bar{y}_i) \mid i = 1, \ldots, n\}$, where $\bar{y}_i = f(\bar{x}_i) + w_i$. Under this new formulation, the following predictive mean estimator (PME) and predictive variance estimator (PVE) are derived in [25].

$$\mathbb{E}(y_\star|D') = \frac{\int_X \mu_\star(x_\star|D') p(\bar{\mathbf{y}}|\mathbf{x}) p(\mathbf{x}|\bar{\mathbf{y}}) d\mathbf{x}}{\int_X p(\bar{\mathbf{y}}|\mathbf{x}) p(\mathbf{x}|\bar{\mathbf{y}}) d\mathbf{x}} \tag{9}$$

and

$$\begin{aligned}\mathbf{var}(y_\star|D') = &\frac{\int_X (\sigma_\star^2(x_\star|D') + \mu_\star^2(x_\star|D')) p(\bar{\mathbf{y}}|\mathbf{x}) p(\mathbf{x}|\bar{\mathbf{y}}) d\mathbf{x}}{\int_X p(\bar{\mathbf{y}}|\mathbf{x}) p(\mathbf{x}|\bar{\mathbf{y}}) d\mathbf{x}} \\ &- \mathbb{E}^2(y_\star|D'),\end{aligned} \tag{10}$$

where $\mu_\star(x_\star|D')$ and $\sigma_\star^2(x_\star|D')$ are given by (7) and (8).

This new formulation of Gaussian process regression incorporates both localization and observation noises in a Bayesian framework. However, there are no closed-form solutions for the PME and PVE given in (9) and (10), respectively. In [25], three approaches are proposed to approximate the PME and PVE and they are the fully-exponential Laplace approximation method [26]–[29], Monte Carlo importance

Preprint submitted to 52nd IEEE Conference on Decision and Control .
Received March 6, 2013.

sampling, and a simple Laplace approximation method. The simple Laplace approximation makes predictions based on the mode of the posterior distribution of the position of an input, i.e., the maximum a posteriori (MAP) estimator of the input.

IV. Simple Laplace Approximation Method

The simple Laplace approximation method for estimating (9) and (10) is based on the maximum a posteriori (MAP) estimation and requires less computation compared to the fully-exponential Laplace approximation method and Monte Carlo importance sampling [25]. The simple Laplace approximation method will be called MAP-GP in the remainder of this paper. The following proposition from [25] demonstrates that MAP-GP can provide good estimates.

Proposition 1: Let $\hat{\mathbf{x}}$ be an asymptotic mode of order $O(n^{-1})$ for $-h(x)$, such that

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in X} p(\bar{\mathbf{y}}|\mathbf{x})p(\mathbf{x}|\bar{\mathbf{x}}) \quad (11)$$

$$h(\mathbf{x}) = -\frac{1}{n}\log(p(\bar{\mathbf{y}}|\mathbf{x})p(\mathbf{x}|\bar{\mathbf{x}})). \quad (12)$$

Assume that $\{h, \widehat{\mathbf{x}}\}$ satisfies the regularity conditions described in [25]. Consider the following approximations for $\mathbb{E}(y_\star|D)$ and $\mathbf{var}(y_\star|D)$

$$\hat{\mathbb{E}}(y_\star|D) = k^T(x_\star, \widehat{\mathbf{x}})(K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}) + \sigma_w^2 I)^{-1}\bar{\mathbf{y}}, \quad (13)$$

$$\widehat{\mathbf{var}}(y_\star|D) = \sigma_f^2 - k^T(x_\star, \widehat{\mathbf{x}})(K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}) + \sigma_w^2 I)^{-1}k(x_\star, \widehat{\mathbf{x}}), \quad (14)$$

where $K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}) \in \mathbb{R}^{n\times n}$ and $k(x_\star, \widehat{\mathbf{x}}) \in \mathbb{R}^n$ are covariance matrices obtained using $\widehat{\mathbf{x}}$. Then, we have the following orders of errors.

$$\hat{\mathbb{E}}(y_\star|D) = \mathbb{E}(y_\star|D) + O(n^{-1})$$
$$\widehat{\mathbf{var}}(y_\star|D) = \mathbf{var}(y_\star|D) + O(n^{-1}).$$

Remark 1: Note that the MAP-GP predictive mean and variance in (13) and (14) take the same form as the original predictive mean and variance, cf. (7) and (8), but the MAP estimator $\widehat{\mathbf{x}}$ from (11) is used.

V. Distributed GPR Algorithm

In this section, we introduce a distributed algorithm for an individual agent (sensor) to estimate an environmental parameter of the surveillance region S only by exchanging local information within r-disk neighbors. Consider a mobile sensor network consisting of q mobile sensing agents distributed in S. This distributed approach can be implemented for a class of kernel functions $K(\cdot,\cdot)$ that have compact supports. The following kernel function is considered in the development of a distributed algorithm.

$$k(x, x') = \sigma_f^2 \lambda\left(\frac{||x - x'||}{r}\right), \quad (15)$$

where

$$\lambda(h) = \begin{cases} (1-h)\cos(ph) + \frac{1}{p}\sin(ph), & \text{if } h \le 1, \\ 0, & \text{otherwise.} \end{cases}$$

Notice that the kernel function K in (15) has a compact support, i.e., $K_{ij} = K(x^{(i)}, x^{(j)})$ is non-zero if and only if $r_{ij} = ||x^{(i)} - x^{(j)}||$ is less than the support r and, similarly, $k_i = K(x^{(i)}, x_\star)$ is non-zero if and only if $d_{i\star} = ||x_i - x_\star||$ is less than the support r. Consider a case in which each agent in a sensor network can only communicate with other agents within a limited communication range of R. In addition, we assume that there exists no central station.

The index of the robotic sensors is denoted by $I = \{1, \cdots, q\}$. The position of agent i is denoted by $x^{(i)}$. Agent i can only communicate with its neighbors in a limited range of R. The adjacency matrix Q indicates whether two agents are neighbors or not. If the element in the i-th row and j-th column of Q, i.e., $Q_{ij} = 1$, then agent i and agent j are neighbors and they have a communication link, and if $Q_{ij} = 0$, then they are not neighbors.

$$Q_{ij} = \begin{cases} 1, & ||x^{(i)} - x^{(j)}|| \le R \text{ and } i \ne j \\ 0, & \text{otherwise,} \end{cases}$$

where R is the communication range between neighbors.

The assumptions made for the resource-constrained mobile sensor networks are listed as follows.

Assumption 1: The radius r of the support of the kernel function (15) satisfies that $0 < r < R$.

Assumption 2: Agent i can only communicate with neighbors in $N(i) = \{j \in I \mid Q_{ij} = 1\}$.

These assumptions indicate that the communication range must be longer than the measurement radius of each agent. If these assumptions are not held, the multi-agent system cannot reach a consensus due to an inability to make communication with each other.

As a result of the specified covariance matrix in (15) and Assumption 1, agent i knows the i-th row of K, i.e., $[K]_{(i)}$, where $K_{ij} \ne 0$ if and only if $j \in N(i)$.

A. Jacobi Over-Relaxation Method

Jacobi over-relaxation (JOR) is a method for solving $Ax = b$, where $A \in \mathbb{R}^{q\times q}$, and $x, b \in \mathbb{R}^q$ [30]. This method assumes that agent i knows the $[A]_{(i)} \in \mathbb{R}^N$ and b_i, and $a_{ij} = (A)_{ij} = 0$ if agent i and agent j are not neighbors which goes along with Assumption 2. The i-th element of solution $x = A^{-1}b$ can be obtained by the following iterative steps:

$$x_i^{(k+1)} = (1-h)x_i^{(k)} + \frac{h}{a_{ii}}\left(b_i - \sum_{j\in N_i} a_{ij}x_j^{(k)}\right), \quad (16)$$

where h is a constant controlling the speed of convergence.

The convergence properties of the iterative JOR algorithm with respect to eigenvalues of the matrix A is specified in [30]. Moreover, Udwadia et al. [31] proved that if $h < \frac{2}{q}$, the convergence of the JOR algorithm to the solution $x = A^{-1}b$ is guaranteed from any initial point x_{init}, where A is a symmetric, positive-definite matrix.

Remark 2: Note that, the solution of the JOR method approaches to the solution $x = A^{-1}b$ without requiring the

Preprint submitted to 52nd IEEE Conference on Decision and Control .
Received March 6, 2013.

computation of an inverse of A. As shown in Section II, a drawback of Gaussian process regression is $O(N^3)$ complexity of calculating $(K(\mathbf{x},\mathbf{x})+\sigma_w^2 I)^{-1}$. This complexity can be relieved if the JOR method is applied in a distributed manner.

B. Discrete-Time Average Consensus Method

The discrete-time average consensus (DAC) method is used to compute the arithmetic mean of elements of a vector [32]. Let

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_q \end{bmatrix} \in \mathbb{R}^q. \tag{17}$$

If the graph is connected, and agent i knows the i-th element of vector $\mathbf{c}$, the arithmetic mean of $\mathbf{c}$ can be computed by iterating

$$x_i^{(k+1)} = x_i^{(k)} + \epsilon \sum_{j\in N_i} a_{ij}(x_j^{(k)} - x_i^{(k)}), \tag{18}$$

with an initial condition $x_i^0 = c_i$, where $a_{ij}=1$ if and only if agent i and agent j are connected. It is proven that if ϵ satisfies

$$0 < \epsilon < \frac{1}{\max_i(\sum_{j\neq i} a_{ij})}, \tag{19}$$

then the DAC algorithm converges to the solution [33].

After the convergence, every node in the network converges to the arithmetic mean of vector $\mathbf{c}$, i.e. $\frac{1}{q}\sum_{i=1}^q c_i$.

C. Distributed Maximum a Posteriori Mode Estimator

In order to find the asymptotic mode $\widehat{\mathbf{x}}$ in a distributed manner, the Jacobi over-relaxation method introduced in Section V-A is used. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 Distributed Algorithm for Finding $\hat{\mathbf{X}}$

Given Initial position $\bar{\mathbf{x}}$, corrupted measurement $\bar{y}$ and ϵ satisfying (19).
begin $\forall_i\ \hat{x}_i^{(0)} = \bar{\mathbf{x}}_i,\ \Gamma_i^{(0)} = 0,\ B_i^{(0)} = \mathbf{0} \in \mathbb{R}^{n\times n}$
repeat
  repeat
    1. Update Γ_i using (22).
    2. Update B_i using (23).
  until Γ_i and B_i converges.
  1. Compute $\frac{\partial h}{\partial x^{(i)}}$ using (25).
  2. Update $\hat{x}_i$ using (26).
until $\hat{x}_i$ converges.

Agent i only knows the i-th row and column of K and $\partial K/\partial x^{(i)}$. In addition, note that for $\partial K/\partial x^{(i)}$, the elements out of i-th row and column are zeros. Define Γ and B_l as follows:

$$\Gamma = (K(\mathbf{x},\mathbf{x})+\sigma_w^2 I)^{-1}\bar{\mathbf{y}} \tag{20}$$

$$B_l = \frac{1}{2}(K(\mathbf{x},\mathbf{x})+\sigma_w^2 I)^{-1}\frac{\partial K(\mathbf{x},\mathbf{x})}{\partial x^{(l)}}. \tag{21}$$

Agent i knows i-th row of $(K(\mathbf{x},\mathbf{x})+\sigma_w^2 I)$ and i-th element of $\bar{\mathbf{y}}$. So $\Gamma^{(i)}$, the i-th element of Γ, can be computed by applying the JOR method over $(K(\mathbf{x},\mathbf{x})+\sigma_w^2 I)^{-1}\bar{\mathbf{y}}$ based on the following recursion:

$$\Gamma^{(i)}(k+1) = (1-\alpha)\Gamma^{(i)}(k) + \frac{\alpha}{\sigma_f^2+\sigma_w^2} \times \left(\bar{y}^{(i)} - \sum_{j\in N(i)} k(\hat{x}^{(i)},\hat{x}^{(j)})\Gamma^{(j)}(k)\right) \tag{22}$$

Similarly, agent i knows $[(K+\sigma_w^2 I)]_{(i)}$, the i-th row of $(K+\sigma_w^2 I)$, and the i-th element of $\bar{\mathbf{y}}$. So $\Gamma^{(i)}$, the i-th element of Γ, can be computed by applying the JOR method. However, agent i needs to receive $\frac{\partial K}{\partial x^{(l)}}$ from agent l if they are connected or use zero instead if they are not connected. In other words, agent i will

$$\begin{cases} \text{get } \left[\frac{\partial K}{\partial x^{(l)}}\right]_{(i)} \text{ from agent } l & \text{if } i\in N(l) \\ \left[\frac{\partial K}{\partial x^{(l)}}\right]_{(i)} = 0 & l\neq i \text{ and } i\notin N(l) \end{cases}$$

and

$$[B_l(k+1)]_{(i)} = (1-\alpha)[B_l(k)]_{(i)} + \frac{\alpha}{\sigma_f^2+\sigma_w^2}\times \left(\left[\frac{\partial K}{\partial x^{(l)}}\right]_{(i)}\Bigg|_{\mathbf{x}=\widehat{\mathbf{x}}} - \sum_{j\in N(i)} k(\widehat{\mathbf{x}}^{(i)},\widehat{\mathbf{x}}^{(j)})[B_l(k)]_{(j)}\right) \tag{23}$$

for $k\in\mathbb{Z}_{\geq 0}$ and $i\in 1,2,\ldots,q$, where $\alpha\in(0, 2\lambda_{min}(K+\sigma_w^2 I))$. At the end of JOR iterations, agent i knows $\Gamma^{(i)}$ and $[B_l]_{(i)}$.

Proposition 2: Given $B_i^{(j)}$, the j-th diagonal element of B_i, $\frac{\partial h}{\partial x^{(i)}}$ calculated for agent i is

$$\frac{\partial h}{\partial x^{(i)}} = \frac{x^{(i)}-\bar{x}^{(i)}}{n\sigma_v^2} - \frac{1}{n}\sum_{j\in N(i)}\left(\Gamma^{(j)}\frac{\partial k(x^{(j)},x^{(i)})}{\partial x^{(i)}}\Gamma^{(i)} - B_i^{(j)}\right). \tag{24}$$

Proof: h given by (12) can be expressed as

$$h = \frac{d}{2}\log(2p\sigma_v^2) + \frac{\sum_{i=1}^n (x^{(i)}-\bar{x}^{(i)})^2}{2m\sigma_v^2} + \frac{1}{2}\log(2p) + \frac{1}{2n}\log|K+\sigma_w^2 I| + \frac{1}{2m}\bar{\mathbf{y}}^T(K+\sigma_w^2 I)\bar{\mathbf{y}}.$$

Now take a derivative respect to $x^{(i)}$ to get

$$\frac{\partial h}{\partial x^{(i)}} = \frac{x^{(i)}-\bar{x}^{(i)}}{n\sigma_v^2} + \frac{1}{2n}tr\left(\frac{\partial K}{\partial x^{(i)}}(K+\sigma_w^2 I)^{-1}\right) + \frac{1}{2n}\bar{\mathbf{y}}^T(K+\sigma_w^2 I)^{-1}\frac{\partial K}{\partial x^{(i)}}(K+\sigma_w^2 I)^{-1}\bar{\mathbf{y}}.$$

Using (20) and (21), we get

$$\frac{\partial h}{\partial x^{(i)}} = \frac{x^{(i)}-\bar{x}^{(i)}}{n\sigma_v^2} + \frac{1}{n}tr(B_i^T) - \frac{1}{2n}\Gamma^T\frac{\partial K}{\partial x^{(i)}}\Gamma. \tag{25}$$

It is known that $\frac{\partial K}{\partial x^{(i)}}$ is a sparse matrix with only non-zero elements on the i-th row and column for the neighbors

Preprint submitted to 52nd IEEE Conference on Decision and Control .
Received March 6, 2013.

of i. Rewriting the above equation in the summation form leads to (24). ■

Finally using the recursive gradient method given by (26), agents can update their modes in a distributed manner, where $\bar{x}^{(i)}$ can be used as an initial condition.

$$\hat{x}^{(i)}(t+1) = \hat{x}^{(i)}(t) + \gamma \frac{\partial h}{\partial x^{((i)}}\bigg|_{x^{(j)}=\hat{x}^{(j)}(t), j\in N(i)}. \quad (26)$$

Algorithm 2 Distributed Algorithm for Sequential Field Prediction

Given Mode $\widehat{\mathbf{x}}$, corrupted measurement $\bar{y}$, unmeasured position $x_\star$ and converged Γ.
begin $\theta_i^{(0)} = k(x_\star, \hat{x}_i)\Gamma^{(i)}$
repeat
1. Update θ_i using (27).
until θ_i converges.
output $\hat{\mathbb{E}}[y_\star|D] = q \times \theta_i$

D. Distributed posterior mean and variance estimators

Let us consider the first-order approximation of the PME estimator in (13). The elements of the covariance matrices $k(\hat{x}^{(i)}, x_\star)$ and $k(\hat{x}^{(i)}, \hat{x}^{(j)})$ can be calculated using (15) and the converged asymptotic mode $\widehat{\mathbf{x}}$ of $-h$ can be found.

Proposition 3: If a sensor network is connected, every agent can compute $\hat{\mathbb{E}}(y_\star|D)$ with the DAC method by exchanging only local information using:

$$\theta_i(k+1) = \theta_i(k) + \epsilon \sum_{j\in N(i)} (\theta_j(k) - \theta_i(k)), \quad (27)$$

where $\theta_i(0) = k^{(i)}\Gamma^{(i)}$.

Proof: By Proposition 1 and (20), PME in (13) can be represented as follows:

$$\begin{aligned} \hat{\mathbb{E}}(y_\star|D) &= k^T(x_\star, \widehat{\mathbf{x}})(K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}) + \sigma_w^2 I)^{-1}\bar{\mathbf{y}} \\ &= k^T(x_\star, \widehat{\mathbf{x}})\Gamma. \end{aligned} \quad (28)$$

The solution of (28) can be solved using DAC since (28) can be matched to (17). Recall that (28) is a scalar value and can be represented as $\sum_{i=1}^{q} k^T(x_\star, \widehat{\mathbf{x}}^{(i)})\Gamma^{(i)}$ and the i-th agent knows both $k^T(x_\star, \widehat{\mathbf{x}}^{(i)})$ and $\Gamma^{(i)}$.

From Section V-B, we know that θ_i converges to $\hat{\mathbb{E}}[y_\star|D]$ when $0 < \epsilon < \frac{1}{\max_i \sum_{i\neq j} Q_{ij}}$ and each agent can easily compute $\hat{\mathbb{E}}[y_\star|D]$ by $q \times \theta_i$. ■

The predictive variance can be approximated similarly with

$$\widehat{\mathbf{var}}(y_\star|D) = \sigma_f^2 - k^T(K + \sigma_w^2 I)^{-1}k. \quad (29)$$

First, suppose that

$$\Phi = (K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}) + \sigma_w^2 I)^{-1}k(x_\star, \widehat{\mathbf{x}}). \quad (30)$$

Agent i can calculate the i-th element of (30) by the following recursion (31).

$$\begin{aligned} \Phi^{(i)}(k+1) &= (1-\alpha)\Phi^{(i)}(k) + \frac{\alpha}{\sigma_f^2 + \sigma_w^2} \\ &\times \left(\bar{y}^{(i)} - \sum_{j\in N(i)} k(\hat{x}^{(i)}, \hat{x}^{(j)})\Phi^{(j)}(k)\right). \end{aligned} \quad (31)$$

Once JOR converges, the error variance can be computed using following recursion based on the DAC method

$$\theta_i(k+1) = \theta_i(k) + \epsilon \sum_{j\in N(i)} (\theta_j(k) - \theta_i(k)), \theta_i(0) = k^{(i)}\Gamma^{(i)}, \quad (32)$$

where

$$\theta_i(0) = k^T(x_\star, \widehat{\mathbf{x}}^{(i)})\Phi^{(i)}. \quad (33)$$

Without loss of generality, the iterative DAC solution in (27) and (32) can be extended to vector formulation, where $\theta_i \in \mathbb{R}^k$.

Remark 3: The proposed distributed posterior mean and variance estimators given by (28) and (29) are different with the naive approach given by (7) and (8). In the naive approach we completely neglect the effect of localization uncertainty. In the proposed distributed approach, we first find a better estimate $\widehat{\mathbf{x}}$ for the uncertain location $\mathbf{x}$ and then use instead of $\mathbf{x}$ for estimating the field.

VI. Simulation Results

In this section, we perform a number of numerical simulations to validate the prediction performance of the proposed distributed GPR algorithm. For simulation, we have randomly generated ten reference fields from a Gaussian process with a covariance function given in (15). We then compare the predicted fields using three different algorithms against the reference field. Three algorithms used in simulations are the quick and dirty solution (QDS) given in (7), the centralized solution using the simple Laplace approximation using the MAP estimator (MAP-GP) in (13), and the proposed distributed Gaussian process regression (D-GP).

We assume that there are ten sensing agents ($q = 10$). For each reference field, sensing agents are randomly located and each agent only knows a noisy position of itself. The variance of the position uncertainty is set to $\sigma_v^2 = 1$. Each agent then makes a noisy measurement from the reference field. The collection of all measurements by all agents is denoted by $D' = \{\bar{\mathbf{x}}, \bar{\mathbf{y}}\}$. The hyper-parameter σ_f^2 of the kernel function in (15) and the variance of the measurement noise σ_w^2 are estimated by maximizing the likelihood (3), i.e.,

$$\{\widehat{\sigma_f^2}, \widehat{\sigma_w^2}\} = \arg\max \log P(\bar{\mathbf{y}}|\bar{\mathbf{x}}, \sigma_f^2, \sigma_w^2),$$

given $D' = \{\bar{\mathbf{x}}, \bar{\mathbf{y}}\}$. Then we have performed the reconstruction of the entire field from D' using three algorithms. For more accurate validation, we have repeated the above procedure ten times for each reference field, resulting ten independent runs for each reference field. For each reference field, we have computed the mean and variance of the reconstruction error from ten independent runs.

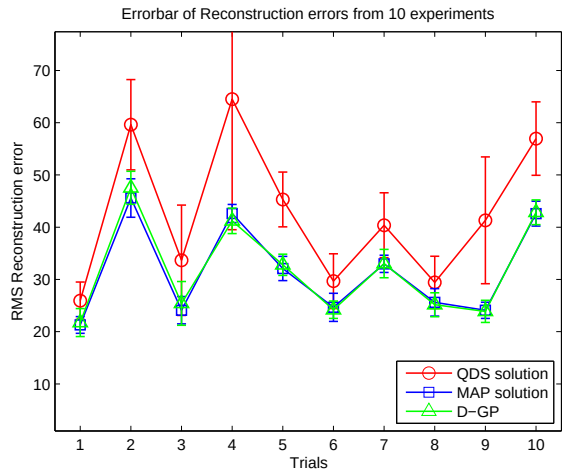

Fig. 1. Reconstruction errors of three algorithms (QDS, MAP-GP, and D-GP) for ten different scenarios. Error bars indicate one standard deviation from ten independent runs for each scenario.

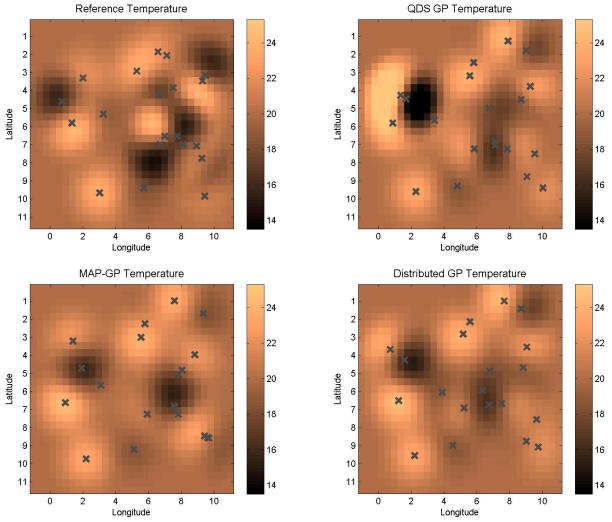

Fig. 2. An example of a reference field and fields reconstructed by three algorithms (QDS, MAP-GP, and D-GP). The reference field is shown in the upper left corner and, clockwise from the top, fields reconstructed using QDS, D-GP, and MAP-GP. The MAP estimates of sensor positions are marked by gray cross marks.

The mean and variance of the root mean squared (RMS) errors between the predicted field and the reference field are shown in Figure 1. As expected, the centralized algorithm MAP-GP shows the best performance in terms of the reconstruction error, followed by D-GP and QDS. However, the mean of the reconstruction error of D-GP is comparable to that of MAP-GP, demonstrating the performance of the proposed distributed algorithm. One example of the reference field and the predicted fields using three algorithms are shown in Figure 2. Figure 3 and Figure 4 demonstrate the convergence of JOR and DAC used in our algorithm.

Since our objective function (12) is non-convex, we used a gradient-based nonlinear optimization method for both MAP-GP and D-GP and this can explain the difference

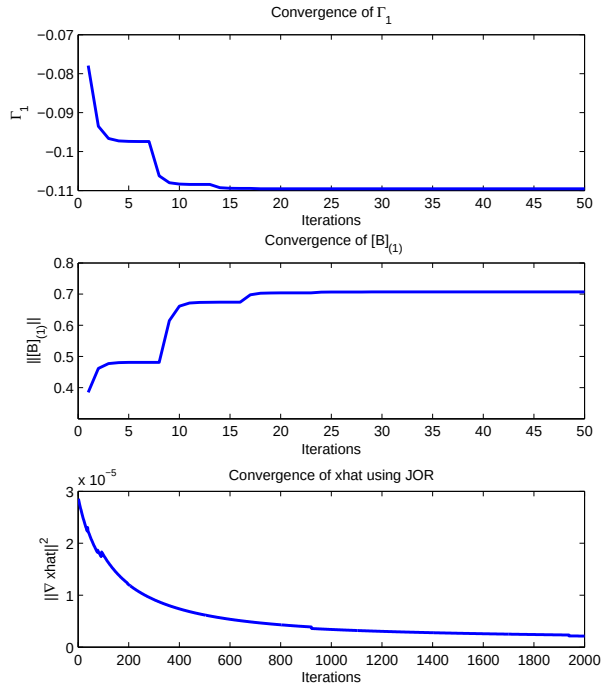

Fig. 3. Convergence of the JOR method. The upper and middle figure indicate Γ_1 and $[B]_{(1)}$ of agent 1, respectively. The bottom figure shows the norm of the gradient of $\widehat{\mathbf{x}}$, i.e., the solution of the MAP estimator in (11).

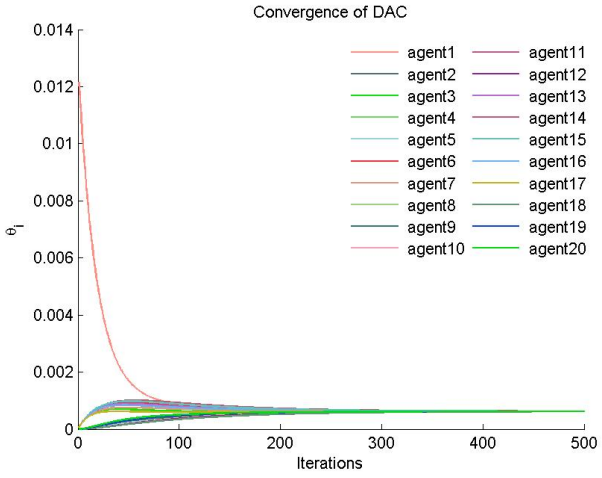

Fig. 4. Convergence of the DAC method. With an increasing number of iterations, θ_i from Algorithm 2 of all agents converges.

between the reconstruction errors of these two algorithms as shown in Figure 1. However, as shown in Table I, D-GP has outperformed QDS in terms of the reconstruction error and its performance is comparable to MAP-GP in most cases. Additionally, it is worth noting that the variance of MAP-GP and D-GP is clearly smaller than that of QDS. This indicates that the prediction performance of both solutions are more robust against diverse situations.

Preprint submitted to 52nd IEEE Conference on Decision and Control .
Received March 6, 2013.

Method						
	QDS		MAP-GP		D-GP	
trial	mean	variance	mean	variance	mean	variance
1	25.93	3.57	**21.28**	1.60	21.73	2.67
2	59.63	8.64	**45.59**	3.68	47.54	3.16
3	33.67	10.56	**24.11**	2.62	25.41	4.18
4	64.52	25.00	42.63	1.74	**41.21**	2.44
5	45.31	5.25	**32.09**	2.32	32.80	4.03
6	29.67	5.24	24.65	2.68	**24.18**	1.66
7	40.37	6.20	**32.98**	1.65	33.03	2.72
8	29.44	5.00	25.62	2.63	**25.15**	2.30
9	41.31	12.14	24.09	1.55	**23.88**	2.12
10	56.96	7.03	**42.60**	2.38	42.85	2.36

TABLE I

MEAN AND VARIANCE OF RECONSTRUCTION ERRORS OF THREE ALGORITHMS: QDS: QUICK AND DIRTY SOLUTION, MAP-GP: CENTRALIZED SOLUTION, AND D-GP: PROPOSED APPROACH.

VII. CONCLUSION

We have presented a distributed Gaussian process regression algorithm that can incorporate both localization and measurement noises in a Bayesian framework. The proposed distributed algorithm is constructed by combining Jacobi over-relaxation (JOR) and discrete time average consensus (DAC) in a distributed manner and suitable mobile sensor networks with the limited communication range. At the same time, the proposed algorithm relieves the heavy computation burden of computing the inverse of a covariance matrix required for Gaussian process regression. The performance of our proposed algorithm is evaluated extensively using a number of simulations. It has been demonstrated that the proposed distributed Gaussian process regression (D-GP) outperforms the quick-and-dirty solution (QDS) in most cases and its performance is comparable to the centralized solution.

REFERENCES

[1] D. Ehrlich, J. Estes, and A. Singh, "Applications of NOAA-AVHRR 1 km data for environmental monitoring," *Remote Sensing*, vol. 15, no. 1, pp. 145–161, 1994.

[2] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: overview of sensor networks," *Computer*, pp. 41–49, 2004.

[3] L. Backer and D. McGillicuddy, "Harmful algal blooms," *Oceanography*, vol. 19, no. 2, p. 94, 2006.

[4] M. Mysorewala, D. Popa, and F. Lewis, "Multi-scale adaptive sampling with mobile agents for mapping of forest fires," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 4, pp. 535–565, 2009.

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

[6] N. Cressie, "Kriging nonstationary data," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 625–634, 1986.

[7] J. Choi, J. Lee, and S. Oh, "Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes," in *Proc. of the American Control Conference*, 2008, pp. 135–140.

[8] ——, "Biologically-inspired navigation strategies for swarm intelligence using spatial Gaussian processes," in *Proc. of the IFAC World Congress*, 2008.

[9] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proc. of the International Conference on Information Processing in Sensor Networks*, 2006.

[10] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[11] A. Smola and P. Bartlett, "Sparse greedy Gaussian process regression," in *Proc. of the Advances in Neural Information Processing Systems*, 2001.

[12] C. Williams and M. Seeger, "Using the Nystrom method to speed up kernel machines," in *Proc. of the Advances in Neural Information Processing Systems*, 2001.

[13] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Proc. of the Advances in Neural Information Processing Systems*, 2002.

[14] M. Seeger, "Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations," Ph.D. dissertation, University of Edinburgh, 2003.

[15] V. Tresp, "A Bayesian committee machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.

[16] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using Gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118–1131, 2011.

[17] L. Yang and K. Ho, "Alleviating sensor position error in source localization using calibration emitters at inaccurate locations," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 67–83, 2010.

[18] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of the International Conference on Mobile Computing and Networking*, 2004.

[19] P. Oguz-Ekim, J. Gomes, J. Xavier, and P. Oliveira, "Robust localization of nodes and time-recursive tracking in sensor networks using noisy range measurements," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3930–3942, 2011.

[20] R. Karlsson and F. Gustafsson, "Bayesian surface and underwater navigation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4204–4213, 2006.

[21] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[22] K. Lynch, I. Schwartz, P. Yang, and R. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.

[23] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.

[24] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1279–1290, 2012.

[25] M. Jadaliha, Y. Xu, J. Choi, N. Johnson, and W. Li, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2013.

[26] L. Tierney and J. Kadane, "Accurate approximations for posterior moments and marginal densities," *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 82–86, 1986.

[27] L. Tierney, R. Kass, and J. B. Kadane, "Fully exponential Laplace approximations to expectations and variances of nonpositive functions," *Journal of the American Statistical Association*, vol. 84, no. 407, 1989.

[28] Y. Miyata, "Fully exponential Laplace approximations using asymptotic modes," *Journal of the American Statistical Association*, vol. 99, no. 468, pp. 1037–1049, 2004.

[29] ——, "Laplace approximations to means and variances with asymptotic modes," *Journal of Statistical Planning and Inference*, vol. 140, no. 2, pp. 382–392, 2010.

[30] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[31] F. Udwadia, "Some convergence results related to the JOR iterative method for symmetric, positive-definite matrices," *Applied Mathematics and Computation*, vol. 47, no. 1, pp. 37–45, 1992.

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

[33] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proc. of the IEEE*, vol. 95, no. 1, 2007, pp. 215–233.