HAL
open science

Non-negative Distributed Regression for Data Inference in Wireless Sensor Networks

Jie Chen, Cédric Richard, Paul Honeine, José C. M. Bermudez

To cite this version:

Jie Chen, Cédric Richard, Paul Honeine, José C. M. Bermudez. Non-negative Distributed Regression for Data Inference in Wireless Sensor Networks. Proc. 44th Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2010, Pacific Grove (CA), USA, United States. pp.451-455, 10.1109/ACSSC.2010.5757599 . hal-01966045

HAL Id: hal-01966045

https://hal.science/hal-01966045v1

Submitted on 27 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Non-negative Distributed Regression for Data Inference in Wireless Sensor Networks

Jie Chen[*,†], Cédric Richard[†], Paul Honeine[*], José Carlos M. Bermudez[‡]

[*] Institut Charles Delaunay(FRE CNRS 2848), Université de Technologie de Troyes,10010 Troyes, France
[†]Laboratoire Fizeau(UMR 6525 CNRS), Université de Nice Sophia-Antipolis, 06108 Nice, France
[‡] Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, SC - Brazil

***Abstract*—Wireless sensor networks are designed to perform on inferences about the environment they are sensing. Due to the inherent physical characteristics of systems under investigation, non-negativity is a desired constraint that can be imposed on the system parameters in some real-life phenomena sensing tasks. In this paper, we propose a kernel-based machine learning strategy to deal with regression problems. Multiplicative update rules are derived in this context to ensure the non-negativity constraints to be satisfied. A distributed algorithm that requires only communication between neighbors is proposed to cope with typical limited energy and bandwidth resources. Synthetic data managed by heat diffusion equations are used to compare the proposed and known algorithms and to illustrate their tracking capabilities.**

I. Introduction

Wireless sensor networks (WSNs) rely on sensor devices deployed in an environment to provide an inexpensive way to monitor physical phenomena. In many real-life phenomena, including biological and physical ones, physical characteristics inherent to the system under investigation require the imposition of non-negativity constraints on the parameters to estimate. For instance, observations in studies of concentration fields or thermal radiation fields are always described with non-negative values (in ppm or in Kelvin). Non-negativity as a physical constraint has received growing attention from the signal processing community during the last decade[1].

Non-parametric approaches based on reproducing kernel methods have recently been successfully applied to distributed regression with collaborative networks. In [2], the authors present a general framework for distributed linear regression motivated by WSNs. In [3], a learning algorithm based on successive orthogonal projections is derived to solve the regularized kernel least-squares problem for regression in sensor networks. In [4], the authors present a projection based kernel distributed learning strategy with reduced order models by using a sparsification criterion. These works provide comprehensive studies in the functional regression and estimation for distributed learning in WSNs. In [5], the authors explore online algorithm for linear system identification. However, none of these algorithms can be used directly to solve the estimation problems in sensor networks under non-negativity constraints.

In this paper, we concentrate on the problem of modeling physical phenomena under non-negativity constraints, and of tracking its evolution. Firstly we formulate the non-negative regression with kernels in a centralized context. A simple multiplicative algorithm is derived to solve this problem. Then we show how the optimization problem can be relaxed to a problem of distributed regression in which nodes only need to communicate with neighbors.

II. Non-negative regression for inference

Within the context of learning in a wireless sensor network of N sensors, we often model a physical phenomenon as a function of the location. Consider a relationship $\psi(.)$ between the sensor's measurement and its position $\boldsymbol{x}_n$. We seek to estimate the function $\psi(\cdot)$ based on newly available position-measurement data y_n to minimize the summed square error

$$\min_{\psi\in\mathcal{H}} \sum_{n=1}^{N} E\left(\psi(\boldsymbol{x}_n) - y_n\right)^2 . \tag{1}$$

By virtue of the representer theorem, the function $\psi(\cdot)$ of reproducing Hilbert kernel space $\mathcal{H}$ can be written with a kernel expansion $\psi(\cdot) = \sum_{n=j}^{N} \alpha_j \kappa(\cdot, \boldsymbol{x}_n)$. Doing that, the cost function can be written as

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \sum_{n=1}^{N} E(\sum_{j=1}^{N} \alpha_j \kappa(\boldsymbol{x}_n, \boldsymbol{x}_j) - y_n)^2 \\ &= \sum_{n=1}^{N} E\left(\boldsymbol{\alpha}^\top \boldsymbol{\kappa}_{\boldsymbol{x}_n} - y_n\right)^2 . \end{aligned} \tag{2}$$

After determining the weight vector $\boldsymbol{\alpha}$, the field can be inferred at any points $\boldsymbol{x}$. One of the most widely used kernels is the Gaussian kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{e}^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2}$ When a non-negative field is to be estimated, and considering that the Gaussian kernel is always positive, each component of the coefficient vector $\boldsymbol{\alpha}$ should be constrained to be non-negative to ensure a non-negative inference function $\psi(\boldsymbol{x})$ at any given position $\boldsymbol{x}$. The constrained optimization problem can be formalized as

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \tag{3}$$

$$\text{subject to } \boldsymbol{\alpha} \geq \mathbf{0} \tag{4}$$

The gradient of $J(\boldsymbol{\alpha})$ is easily computed as follows

$$\nabla J(\boldsymbol{\alpha}) = \sum_{n=1}^{N} E\left(\boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^\top \boldsymbol{\alpha} - y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n}\right) \tag{5}$$

As the evaluation of the gradient usually cannot be achieved in many real-life applications, we use the instantaneous estimator

$$\widetilde{\nabla} J(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \left(\boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\alpha} - y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} \right) \tag{6}$$

Note that $\psi(.)$ is linear with respect to the kernel functions $\kappa(., \boldsymbol{x}_n)$, although it is nonlinear with respect to $\boldsymbol{x}_n$. Therefore, we use an algorithm similar to that proposed in our previous work [5] to solve the problem. We decompose the gradient $-\widetilde{\nabla} J(\boldsymbol{\alpha})$ as follows,

$$[-\nabla \widetilde{J}(\boldsymbol{\alpha}(k))]_i = [\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i - [\boldsymbol{V}(\boldsymbol{\alpha}(k))]_i \tag{7}$$

where $[\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i$ and $[\boldsymbol{V}(\boldsymbol{\alpha}(k))]_i$ are strictly positive components. The update equation for the i-th component can then be expressed as

$$\alpha_i(k+1) = \alpha_i(k) \frac{[\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i}{[\boldsymbol{V}(\boldsymbol{\alpha}(k))]_i} \tag{8}$$

This expression is referred to as the multiplicative weight update algorithm. If we initialize the weight vector with a positive vector, the constraints will be always satisfied due to the non-negativity of $[\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i$ and $[\boldsymbol{V}(\boldsymbol{\alpha}(k))]_i$. The gradient defined by (6) can be decomposed as in (7) by setting $\boldsymbol{U}(\boldsymbol{\alpha}(k)) = \sum_{n=1}^{N} y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} + \xi$ and $\boldsymbol{V}(\boldsymbol{\alpha}(k)) = \sum_{n=1}^{N} \boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\alpha}(k) + \xi$, with ξ positive to avoid $[\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i$ to become negative due to the noise. The centralized algorithm vector weight update is then

$$\boldsymbol{\alpha}(k+1) = \boldsymbol{\alpha}(k) \text{diag} \left(\frac{[\sum_{n=1}^{N} y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} + \xi]_i}{[\sum_{n=1}^{N} \boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\alpha}(k) + \xi]_i} \right) \tag{9}$$

III. Distributed regression with diffusion strategy in WSNs

In what follows, we show how the optimization problem in (2) can be relaxed for distributed inference. Let $\mathcal{N}$ denote the set of neighbors for sensor k. Consider an $N \times N$ matrix $\mathbf{B}$ with entries $\{b_{n,k}\}$ defined by $b_{n,k} = 0 \quad \text{if } n \notin \mathcal{N}_k$ and $\mathbf{B}\mathbb{1} = \mathbb{1}$, $\mathbf{B} = \mathbb{1}^{\top}$. With the constraint of communication range, the cost function of (2) is rewritten as follows

$$J(\boldsymbol{\alpha}) = J_k(\boldsymbol{\alpha}) + \sum_{n=1, n \neq k}^{N} J_n(\boldsymbol{\alpha}) \tag{10}$$

We define diagonal matrices $\mathbf{C_k}$ for each node k with elements $c_{k,i,i} = 1$ if $i \in \mathcal{N}_k$, and $c_{k,i,i} = 0$ otherwise. The local cost function $J_k(\boldsymbol{\alpha})$ is defined as

$$J_k(\boldsymbol{\alpha}) = \sum_{n=1}^{N} b_{n,k} E \left(\boldsymbol{\alpha}^{\top} \mathbf{C}_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} - y_n \right)^2 \tag{11}$$

which is actually equivalent to

$$J_k(\boldsymbol{\alpha}_k) = \sum_{n \in \mathcal{N}_k} b_{n,k} E \left(\boldsymbol{\alpha}_k^{\top} \boldsymbol{\kappa}_{\boldsymbol{x}_n} - y_n \right)^2 \tag{12}$$

Each node could only communicate with the nodes within the range of a neighborhood of node k. The instananeous gradient of the cost function (10) at each node k is

$$[\nabla J(\boldsymbol{\alpha}_k)]_i = \left[\sum_{n \in \mathcal{N}_k} b_{n,k} \left(\boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\alpha}_k - y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} \right) \right]_i + \left[\sum_{n=1, n \neq k}^{N} \mathbf{C}_n \nabla J_n(\boldsymbol{\alpha}) \right]_i \tag{13}$$

where $i \in \mathcal{N}_k$. To relax the problem so that sensors only need to get information from its neighbors, we use

$$[\nabla J(\boldsymbol{\alpha}_k)]_i = \left[\sum_{n \in \mathcal{N}_k} b_{n,k} \left(\boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\alpha}_k - y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n} \right) \right]_i + \left[\sum_{n \in \mathcal{N}_k, n \neq k}^{N} \mathbf{C}_n \nabla J_n(\boldsymbol{\alpha}) \right]_i \tag{14}$$

The first part of (14) can be viewed as the gradient of local cost function $\nabla J_k(\boldsymbol{\alpha}_k)$. Using the proposed multiplicative algorithm, $[-\nabla J_k(\boldsymbol{\alpha}_k)]_i$ is decomposed into two positive components $[\boldsymbol{U_k}(\boldsymbol{\alpha}(k))]_i = \left[\sum_{n \in \mathcal{N}_k} b_{n,k} y_n \boldsymbol{\kappa}_{\boldsymbol{x}_n}\right]_i + \xi$ and $[\boldsymbol{V_k}(\boldsymbol{\alpha}(k))]_i = \left[\sum_{n \in \mathcal{N}_k} b_{n,k} \boldsymbol{\kappa}_{\boldsymbol{x}_n} \boldsymbol{\kappa}_{\boldsymbol{x}_n}^{\top} \boldsymbol{\alpha}_k(k)\right]_i + \xi$. The second part of (14) can be viewed as a regularization item which is decomposed using $\left[\widetilde{\boldsymbol{U}}_k(\boldsymbol{\alpha}(k))\right]_i = \sum_{n \in \mathcal{N}_k, n \neq k} [\boldsymbol{U}_n(\boldsymbol{\alpha}(k))]_i$ and $\left[\widetilde{\boldsymbol{V}}_k(\boldsymbol{\alpha}(k))\right]_i = \sum_{n \in \mathcal{N}_k, n \neq k} [\boldsymbol{V}_n(\boldsymbol{\alpha}(k))]_i$, where $[\boldsymbol{U}(\boldsymbol{\alpha}(k))]_i$ and $[\widetilde{\boldsymbol{V}}(\boldsymbol{\alpha}(k))]_i$ are transferred from its neighbors. Finally, the coefficients update rule for node k is written in multiplicative form as

$$\boldsymbol{\alpha}_i(k+1) = \boldsymbol{\alpha}_i(k) \frac{[\boldsymbol{U_k}(\boldsymbol{\alpha}(k))]_i + \left[\widetilde{\boldsymbol{U}}_k(\boldsymbol{\alpha}(k))\right]_i}{[\boldsymbol{V_k}(\boldsymbol{\alpha}(k))]_i + \left[\widetilde{\boldsymbol{V}}_k(\boldsymbol{\alpha}(k))\right]_i} \tag{15}$$

IV. Simulation experiments

We consider a classical application of estimating a heat diffusion field governed by the partial differential equation

$$\frac{\partial T(\boldsymbol{x}, t)}{\partial t} - c \nabla_{\boldsymbol{x}}^2 T(\boldsymbol{x}, t) = Q(\boldsymbol{x}, t).$$

We studied the problem of monitoring the evolution of the heat using $N = 100$ random sensors. One heat source was activated from $t = 1$ to $t = 100$, and the second one from $t = 100$ to $t = 200$. The bandwidth of Gaussian function $\sigma = 0.1826$. Two random scenarios, respectively depicted in Figure 1 and Figure 2, are taken into simulations. In the 1st scenario, two sources are located at positions relatively "poor" of sensors; otherwise, in the 2nd scenario, the two sources are located at "better" positions. The algorithms compared are: 1) Centralized multiplicative algorithm; 2) Proposed distributed multiplicative; 3) Centralized Barzilai-Borwein gradient projection; 4) Distributed gradient projection. The learning curves are shown in figures 3 and 4 respectively. The abrupt change in heat sources at $t = 100$ is clearly visible, and the proposed algorithm has good performance with little computational burden.

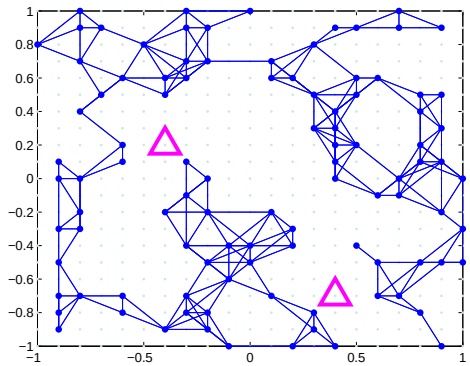

Fig. 1. 1st simulation scenario.The edges between two nodes show the neighborhood relation. Two magenta △ represent the positions of two sources.

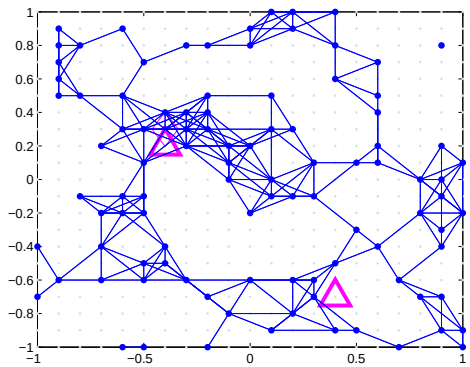

Fig. 2. 2nd simulation scenario. In this scenario, the positions of two sources are more close to sensors than that in scenario1.

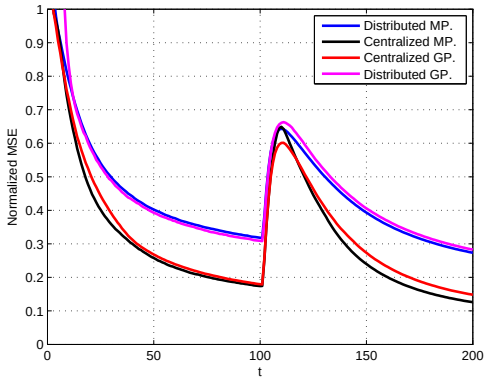

Fig. 3. Convergence comparison of the 1st scenario.

V. Conclusion

In many real-life phenomena non-negativity is a desired constraint that must be imposed on the parameters to estimate due to the inherent physical characteristics of systems In this paper, we proposed a multiplicative method for data inference under non-negativity constraints. Under the context of wireless sensor networks, we developed a distributed learning algorithm to enable each sensor to estimate the non-negative field with the help of neighbor information. The proposed algorithm also shows a good performance in its tracking capacity.

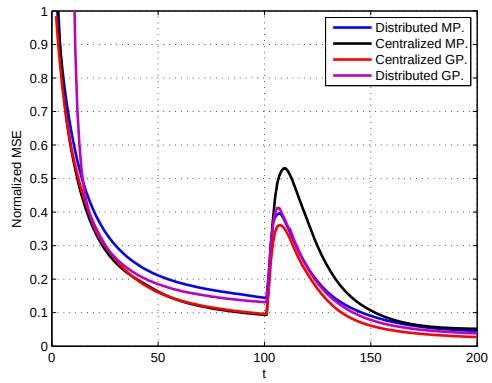

Fig. 4. Convergence comparison of the 2nd scenario.

References

[1] L. Saul, F. Sha, and D. Lee, "Statistical signal processing with nonnegativity constraints," in *Eighth European Conference on Speech Communication and Technology*, vol. 2. Citeseer, 2003, pp. 1001–1004.

[2] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM New York, NY, USA, 2004, pp. 1–10.

[3] J. Predd, S. Kulkarni, and H. Poor, "Regression in sensor networks: Training distributively with alternating projections," in *Proc. SPIE*, vol. 5910. Citeseer, 2005, pp. 42–56.

[4] 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 *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing-Volume 00*. IEEE Computer Society, 2009, pp. 2861–2864.

[5] J. Chen, C. Richard, P. Honeine, H. Lantéri, and C. Theys, "System identification under non-negativity constraints," in *Proc. of European Conference on Signal Processing (EUSIPCO)*, Aalborg, Denmark, 2010.