HAL
open science

Distributed regression in sensor networks with a reduced-order kernel model

Paul Honeine, Mehdi Essoloh, Cédric Richard, Hichem Snoussi

To cite this version:

Paul Honeine, Mehdi Essoloh, Cédric Richard, Hichem Snoussi. Distributed regression in sensor networks with a reduced-order kernel model. Proc. 51st IEEE GLOBECOM Global Communications Conference, 2008, New Orleans, LA, USA, United States. pp.1-5, 10.1109/GLOCOM.2008.ECP.29 . hal-01966053

HAL Id: hal-01966053

https://hal.science/hal-01966053v1

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.

Distributed regression in sensor networks with a reduced-order kernel model

Paul Honeine, Mehdi Essoloh, Cédric Richard and Hichem Snoussi
Institut Charles Delaunay (FRE CNRS 2848) – LM2S
Université de technologie de Troyes
10010 Troyes, France

***Abstract*—Over the past few years, wireless sensor networks received tremendous attention for monitoring physical phenomena, such as the temperature field in a given region. Applying conventional kernel regression methods for functional learning such as support vector machines is inappropriate for sensor networks, since the order of the resulting model and its computational complexity scales badly with the number of available sensors, which tends to be large. In order to circumvent this drawback, we propose in this paper a reduced-order model approach. To this end, we take advantage of recent developments in sparse representation literature, and show the natural link between reducing the model order and the topology of the deployed sensors. To learn this model, we derive a gradient descent scheme and show its efficiency for wireless sensor networks. We illustrate the proposed approach through simulations involving the estimation of a spatial temperature distribution.**

I. Introduction

Wireless sensor networks involve large numbers of deployed tiny radio-equipped sensors, and provide an inexpensive way to monitor physical phenomena, such as a temperature field. With relatively inexpensive wireless devices, each device has a limited amount of memory, reduced processing capabilities, limited power resources, and low communication capacities. In order to carry out good coverage of the region under scrutiny, sensors must be deployed densely, resulting in highly redundant spatial information. Many researchers in the sensor network community exploit this redundancy in order to reduce the complexity of the resulting model, see for instance [1] and references therein. Guestrin *et al.* consider in [2] the spatial and temporal redundancy, where the latter is handled by fitting a three-degree polynomial to the sensed data. While this is a model-based technique, model-independent approaches received considerable attention recently. Rabbat *et al.* applied in [3] an incremental subgradient optimization technique for parameter estimation, yielding an efficient scheme for sensor network in terms of the energy-accuracy tradeoff. As pointed out by Predd *et al.* in [4], such a technique is however inappropriate for functional estimation.

Recently, increasing research attention has been directed towards kernel methods for pattern recognition, in both unsupervised and supervised learning for classification and regression problems. Based on the concept of reproducing kernels initially introduced by Aronszajn in the 50's, these methods have gained popularity with the prelude of support vector machines and the statistical learning theory [5]. The literature on wireless sensor networks does not escape from the proliferation of kernel machines and its attractiveness, as studied for problems such as localization [6], detection [7], estimation [8], and regression [9]. Predd *et al.* propose in [9] to assess distributively the regression problem by solving it locally on each sensor device, which gets information from its neighbors. While this technique is computationally efficient, it has some drawbacks. On the one hand, sensor devices must be set in broadcast mode in order to communicate with their neighbors, and thus require more power for lateral communication as compared to a sensor-to-sensor communication. On the other hand, each sensor needs to solve the optimization problem, resulting in a *on-sensor* matrix inversion that leads to considerable computational burden. This technique suffers from these disadvantages as long as sensors are densely deployed.

Classical kernel machines are inappropriate for regression in the context of sensor network, mainly for one major drawback, illustrated by the Representer Theorem [10], [11]: The order of the resulting model is equal to the number of available observations. Therefore, the model order scales badly with the number of deployed sensors. In order to overcome this difficulty, we consider in this paper a reduced-order approach. Controlling the complexity of the model is widely used not only in kernel machines [12], but also in sparse representation literature [13], [14]. It turns out that this is equivalent to selecting a small set of sensor devices that are well spread in the network. In the proposed scheme, each device updates the model from its measurement, and increments its order *if necessary*, then transmits the model parameters to the next device, and so on. Both this sensor-to-sensor communication scheme which reduces the overall energy consumption, and the gradient-based learning technique that we derive in this paper, allow us to overcome the disadvantages of the method proposed by Predd *et al.*

This paper is organized as follows. In the next section, functional learning with classical kernel machines is concisely introduced, and we illustrate via the Representer Theorem their inappropriateness for learning in wireless sensor network. A solution to this drawback is proposed by controlling the model order, as derived in Section III. Section IV is devoted to the proposed algorithm based on a gradient descent scheme, and implementation issues regarding wireless sensor networks are studied. The efficiency of the proposed approach is illustrated in Section V, with an application to estimating a spatial temperature distribution.

II. Functional learning in sensor networks

In a conventional regression problem, one seeks a function that links the best the input space to the observation domain. By designating the former by $\mathcal{X}$ and the latter by $\mathcal{D}$, we learn this function $\psi^*(\cdot)$ from available data, $(\boldsymbol{x}_i, d_i) \in \mathcal{X} \times \mathcal{D}$ for $i = 1, \ldots, n$. The optimization problem is given by minimizing the mean-square-error between the model output $\psi(\boldsymbol{x}_i)$ and the desired output d_i, with

$$\psi^*(\cdot) = \arg\min_{\psi} \frac{1}{n} \sum_{i=1}^{n} |d_i - \psi(\boldsymbol{x}_i)|^2.$$

It is well-known that one needs to constrain this optimization problem into a functional space of smooth functions. This can be done by combining the reproducing kernel Hilbert space (RKHS) with Tikhonov regularization, yielding the optimization problem

$$\psi^*(\cdot) = \arg\min_{\psi \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} |d_i - \psi(\boldsymbol{x}_i)|^2 + \eta \, \|\psi\|_{\mathcal{H}}^2, \qquad (1)$$

where the parameter η controls the tradeoff between smoothness and fitting the data. In this expression, $\mathcal{H}$ is the RKHS of a given reproducing kernel $\kappa(\cdot, \cdot)$. This means that each function of $\mathcal{H}$ is evaluated at any $\boldsymbol{x} \in \mathcal{X}$ with $\psi(\boldsymbol{x}) = \langle \psi(\cdot), \kappa(\cdot, \boldsymbol{x}) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product in $\mathcal{H}$.

Initially introduced in [10] and generalized more recently in [11] for the wide class of kernel machines, the Representer Theorem states that the solution of the regularized optimization problem (1) is of the form

$$\psi^*(\cdot) = \sum_{k=1}^{n} \alpha_k \kappa(\boldsymbol{x}_k, \cdot). \qquad (2)$$

In order to determine the model, completely identified by its coefficients α_k, we inject this expression back into (1), yielding the so-called dual optimization problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{d} - \boldsymbol{K}\boldsymbol{\alpha}\|^2 + \eta \, \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha}, \qquad (3)$$

where $\boldsymbol{d}$ and $\boldsymbol{\alpha}$ are n-by-1 column vectors whose i-th entries are d_i and α_i, respectively, and $\boldsymbol{K}$ is the n-by-n matrix whose (i, j)-th entry is $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This is a linear optimization problem, and its solution is given by

$$\boldsymbol{\alpha}^* = \left(\boldsymbol{K}^\top \boldsymbol{K} + \eta \boldsymbol{K}\right)^{-1} \boldsymbol{K}^\top \boldsymbol{d}. \qquad (4)$$

Consider the regression problem in sensor networks where, for instance, we seek to estimate a temperature field. The input space corresponds to the location in the region under scrutiny, $\mathcal{X} \subset \mathbb{R}^2$ in a plane, and the output is the temperature measure $\mathcal{D} \subset \mathbb{R}$. Sensors measure temperature at their respective locations, and therefore construct the training set. In a distributive scheme, each sensor updates the coefficient vector with available information and transmits it to the next sensor.

However, solving the optimization problem described above can become cumbersome as illustrated by the matrix inversion in (4), whose computational complexity is $\mathcal{O}(n^3)$, and therefore scales badly with the number of sensors in the network. While we derive this result for the particular case of the mean-square-error as a cost functional, such drawback is common to all classical kernel machines as illustrated by the Representer Theorem, with a model completely determined by the n coefficients, α_k, as well as the n coordinates $\boldsymbol{x}_k$ representing sensor locations. In order to propose a distributive technique to solve this problem, Predd *et al.* propose in [9] to solve it locally by every sensor, with location and measurement of each neighboring sensor taking part in the calculations. This requires to convey this information and to invert a matrix similar to the one in (4) whose dimensions correspond to the number of available neighbors. Such technique is however unadapted in practice as sensors are densely deployed, with heavier communication burden and higher neighborhood concentration.

III. A reduced-order model for sensor networks

In order to overcome the difficulties invoked in the previous section, we propose a reduced-order model constructed from the optimal model (2) by considering only a small number of kernel functions in the expansion. Let m be that number, the reduced-order model is defined by

$$\psi(\cdot) = \sum_{k=1}^{m} \alpha_k \kappa(\boldsymbol{x}_{\omega_k}, \cdot), \qquad (5)$$

where the m coordinates $\boldsymbol{x}_{\omega_k}$ are selected from the locations of all the sensors, and thus $\omega_k \in \{1, \ldots, n\}$. Without going into details, we note that by injecting (5) back into the cost functional (1) of the regularized optimization problem, we obtain a solution similar to (4), where one needs to inverse an m-by-m matrix with entries of the form $\kappa(\boldsymbol{x}_{\omega_i}, \boldsymbol{x}_{\omega_j})$. In Section IV, we derive a receive-update-transmit scheme for each sensor in order to learn the model by minimizing the error on its sensed information. But before, we study the problem of selecting these kernel functions, or equivalently selecting the m sensors with locations given by $\boldsymbol{x}_{\omega_1}, \ldots, \boldsymbol{x}_{\omega_m}$.

The problem of sparse representations spans many scientific and research area, and more specifically in signal processing [13], [14] and pattern recognition with kernel machines [15], [16]. Sparse techniques with ℓ_1-based penalization are too computational expensive for wireless sensor networks. A more appropriate approach would be the one proposed in [17], [16], which translates in sensor networks as follows: the model is constructed in a simple walk through the network, where each sensor discards its kernel function from the model, and thus leaves its order unchanged, if the kernel function can be well approximated by the model; otherwise the order is incremented by adding the kernel function to the model. In other words, at sensor i, if the quantity

$$\min_{\beta_1, \ldots, \beta_m} \|\kappa(\boldsymbol{x}_i, \cdot) - \sum_{k=1}^{m} \beta_k \kappa(\boldsymbol{x}_{\omega_k}, \cdot)\|_{\mathcal{H}}^2$$

is smaller than a given threshold, then the kernel function $\kappa(\boldsymbol{x}_i, \cdot)$ is not included into the model; otherwise, it is added to the model. This criterion however requires the inversion of an m-by-m matrix, and therefore demands high precision and computational resources from the on-sensor microprocessor for each sensor.

We propose in this article to reduce further the computational burden of evaluating the criterion for determining the relevance of a kernel function with respect to the model. For this, we take advantage of recent developments investigated by two of the authors, on the coherence criterion defined as follows: The kernel function $\kappa(\boldsymbol{x}_i, \cdot)$ is included to the m-order model if

$$\max_{k=1,\ldots,m} |\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{\omega_k})| \leq \nu, \tag{6}$$

where ν is a given threshold. This means that the kernel functions of the resulting model have a bounded cross-correlation, since from (6) we have $\max_{k,l=1,\ldots,m} |\langle \kappa(\boldsymbol{x}_{\omega_k}, \cdot), \kappa(\boldsymbol{x}_{\omega_l}, \cdot)\rangle| \leq \nu$. Without going into details, properties of this criterion and connections to other sparsification criteria are studied in our recent papers [18], [19] for prediction in time series data. From this criterion, we derive a natural one for learning in wireless sensor networks.

In the particular case of the widely used radial kernels which can be expressed in terms of $\kappa(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$, the criterion (6) may be simplified further. This is because $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{\omega_k})$ can be written in terms of $\|\boldsymbol{x}_i - \boldsymbol{x}_{\omega_k}\|$. To illustrate this property without loss of generality, we consider the Gaussian kernel defined by $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{e}^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\beta_0^2}$ where β_0 is a tunable parameter. By substituting this definition into the criterion expression above, we obtain the expression

$$\min_{k=1,\ldots,m} \|\boldsymbol{x}_i - \boldsymbol{x}_{\omega_k}\|^2 > 2\beta_0^2 \ln(1/\nu).$$

In other words, sensor i increments the model order by including its kernel function to the model, if it does not belong to the neighborhood of sensors previously in the model. The notion of neighborhood here is defined by the right-hand-side of the above expression. By setting $\nu_0^2 = 2\beta_0^2 \ln(1/\nu)$, we get the so-called neighborhood criterion

$$\min_{k=1,\ldots,m} \|\boldsymbol{x}_i - \boldsymbol{x}_{\omega_k}\| > \nu_0. \tag{7}$$

Thus, it turns out that the criterion (6) based on functional approximation in the RKHS is natural for the topology of sensor networks, as it is equivalent to the neighborhood criterion defined in (7). Next we derive an algorithm to learn such reduced-order model, by considering the criterion (7).

IV. Distributed reduced-order model algorithm

A large class of algorithms can be proposed for solving the optimization problem, mainly with techniques studied in [18], [19] and adapted here to the neighborhood criterion. In what follows, we derive a low-computational demanding algorithm, based on a simple gradient descent approach, and study implementation issues in wireless sensor networks.

A. The algorithm

For this purpose, each sensor i updates the coefficient vector $\boldsymbol{\alpha}_i$ to be as close as possible to the previous one $\boldsymbol{\alpha}_{i-1}$ by imposing a null error on approximating the measurement. This is equivalent to solving the following constraint optimization problem :

$$\min_{\boldsymbol{\alpha}_i} \quad \|\boldsymbol{\alpha}_{i-1} - \boldsymbol{\alpha}_i\|^2 \tag{8}$$

$$\text{subject to} \quad \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_i = d_i, \tag{9}$$

with $\boldsymbol{\kappa}_i$ an m-by-1 column vector whose k-th entry is $\kappa(\boldsymbol{x}_{\omega_k}, \boldsymbol{x}_i)$, where the constraint is obtained by applying (5) to information relative to sensor i. Solving this problem at each sensor depends on the neighborhood criterion, and thus two cases are possible, depending if (7) is satisfied or not.

Case 1. $\min_{k=1,\ldots,m} \|\boldsymbol{x}_i - \boldsymbol{x}_{\omega_k}\| < \nu_0$:

This is the case when sensor i is in the neighborhood of any of the m previously selected sensors $\omega_1, \ldots, \omega_m$. Therefore its kernel function may be well approximated by a linear combination of the model kernel functions. To solve the constraint optimization problem (8)-(9), we consider minimizing the corresponding Lagrangian given by

$$\|\boldsymbol{\alpha}_{i-1} - \boldsymbol{\alpha}\|^2 + \lambda(d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}),$$

where λ is the Lagrangian multiplier. By setting to zero the derivatives of the above cost function with respect to $\boldsymbol{\alpha}$ and λ, we get the following conditions on $\boldsymbol{\alpha}_i$ to verify:

$$2\,(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1})^\top = \lambda\, \boldsymbol{\kappa}_i^\top \quad \text{and} \quad \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_i = d_i.$$

Since $\boldsymbol{\kappa}_i^\top \boldsymbol{\kappa}_i$ is nonzero, these equations lead to $\lambda = 2\,(\boldsymbol{\kappa}_i^\top \boldsymbol{\kappa}_i)^{-1}(d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_{i-1})$, where $d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_{i-1}$ is the *a priori* error at sensor i. By substituting into the above expression, we obtain the update equation

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{i-1} + \frac{\rho}{\|\boldsymbol{\kappa}_i\|^2}\, \boldsymbol{\kappa}_i \left(d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_{i-1}\right). \tag{10}$$

where we have introduced the step-size control parameter ρ, as preconised in conventional adaptive filtering techniques.

Case 2. $\min_{k=1,\ldots,m} \|\boldsymbol{x}_i - \boldsymbol{x}_{\omega_k}\| > \nu_0$:

In this case, the sensor i is not covered by the m sensors defining the kernel functions of the model. Therefore, we increment the model order by including the corresponding kernel function to it. To accommodate the new element in $m+1$ vector $\boldsymbol{\alpha}_i$, we modify problem (8)-(9) as

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}_{[1,\ldots,m]} - \boldsymbol{\alpha}_{i-1}\|^2 + \alpha_{m+1}^2 \quad \text{subject to} \quad d_i = \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha},$$

where $\boldsymbol{\alpha}_{[1,\ldots,m]}$ denotes the first m elements of the vector, and $\boldsymbol{\kappa}_i$ has been increased by one entry. Considerations similar to those made in Case 1 lead to the updating rule

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \boldsymbol{\alpha}_{i-1} \\ 0 \end{bmatrix} + \frac{\rho}{\|\boldsymbol{\kappa}_i\|^2}\, \boldsymbol{\kappa}_i \left(d_i - \boldsymbol{\kappa}_i^\top \begin{bmatrix} \boldsymbol{\alpha}_{i-1} \\ 0 \end{bmatrix}\right), \tag{11}$$

with $\boldsymbol{\kappa}_i$ the $(m+1)$-by-1 column vector whose k-th entry is $\kappa(\boldsymbol{x}_{\omega_k}, \boldsymbol{x}_i)$, and $\omega_{m+1} = i$.

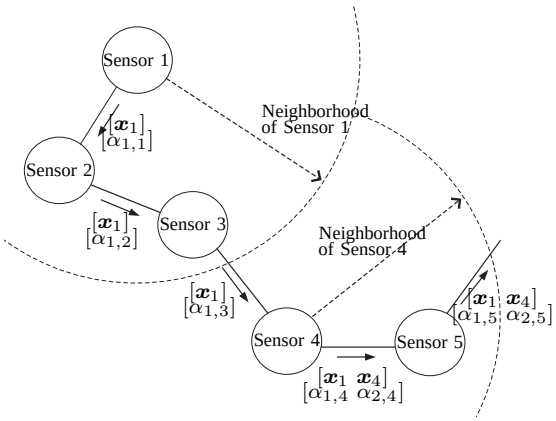

Fig. 1. A schematic representation illustrating the proposed approach.

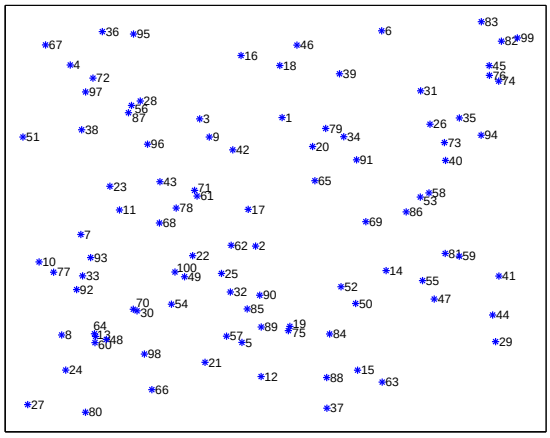

Fig. 2. Map of the sensors, randomly (and uniformly) distributed over the region under scrutiny.

B. Analysis of the algorithm

We illustrate the proposed approach in Fig. 1, where we present five sensors with links representing the walk of the information throughout the network. Next we denote by $\alpha_{k,i}$ the k-th entry of the coefficient vector $\boldsymbol{\alpha}_i$ updated by sensor i. Initializing at Sensor 1 yields $\omega_1 = 1$ and $m = 1$. Its location $\boldsymbol{x}_1$ and the estimated coefficient $\alpha_{1,1}$ are transmitted to the next sensor in the network. Sensor 2 belongs to the neighborhood of Sensor 1, since $\|\boldsymbol{x}_2 - \boldsymbol{x}_{\omega_1}\|$ is less than the given neighborhood threshold. Thus it updates the coefficient into $\alpha_{1,2}$ with the rule (10) by leaving the model order unchanged. It transmits $\alpha_{1,2}$ and $\boldsymbol{x}_1$ to the next sensor, Sensor 3, with updating similar to Sensor 2 since it also belongs to the neighborhood of Sensor 1. Not in that neighborhood, Sensor 4 increments the model order by adding its kernel function to the model and a new weighting coefficient. Updating with (11), the resulting coefficients as well as the locations $[\boldsymbol{x}_1\ \boldsymbol{x}_4]$ are transmitted to the next sensor, and so on.

In designing algorithms for wireless sensor network applications, several implementation issues must be taken into consideration, mainly algorithmic complexity and energy consumption, the latter being proportional to the size of transmitted messages. The proposed approach answers this question, on both aspects. On the one hand, the gradient technique has low computational complexity and minor memory requirements, with no matrix inversion as opposed to Predd's technique [4], and thus can be efficiently implemented on tiny low-cost processors. On the other hand, the messages communicated between sensors are constituted only of the m coordinates and m coefficients taking part in the reduced-order model, as opposed to the full-order model where all the coordinates and coefficient values must be conveyed between sensors.

It is worth noting that for small values of the neighborhood threshold, the influence region of each sensor is reduced, resulting into a model with a larger order, therefore closer to the optimal full-order model given in (2). However, the price to pay is an increasing energy consumption, mainly due to the communication for conveying the parameters of the model. This is the classical accuracy-energy dilemma in wireless sensor networks, with the neighborhood threshold determines the tradeoff between the optimal approximation and the reduction in the model order.

V. Simulations

In order to illustrate the proposed approach, we consider estimating the spatial temperature distribution, with sensors densely and randomly spread over the region under scrutiny. Due to lack of experimental data from densely deployed sensor networks, we consider simulation results[1] obtained from classical partial differential equations. We consider three heat sources randomly set inside a square region, and estimate the temperature distribution from a single measurement available on each of the $n = 100$ sensors randomly spread.

As the configuration settings determines the neighborhood threshold to be used, we set in what follows $\nu_0 = 0.3$, which yields a 15-order model, corresponding to a 85% of reduction in the model order compared to the full-order one. In what follows, the Gaussian kernel is used. To determine the influence of both its bandwidth and the step-size parameter,

[1]A detailed description on the experimental configuration and the dataset is available in http://www.ulb.ac.be/di/mlg/sensorNet/modeling.html.

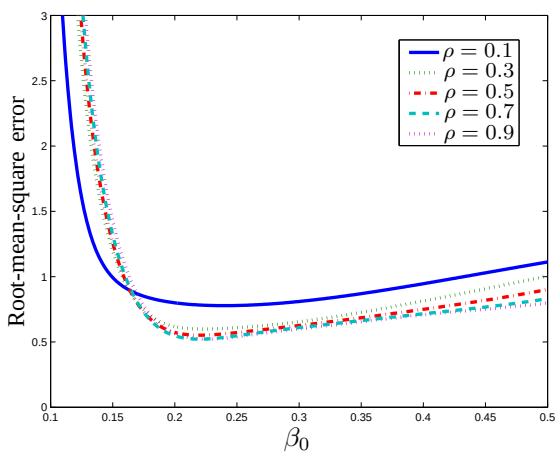

Fig. 3. Evolution of the error as a function of the bandwidth of the Gaussian kernel, for different values of the step-size parameter.

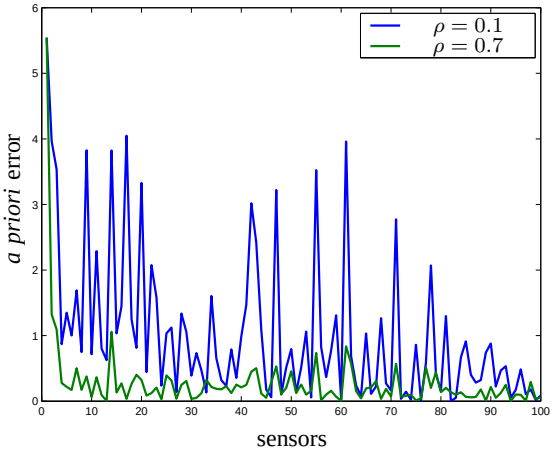

Fig. 4. Convergence of the algorithm as we walk through the 100 sensors of the network.

we define the root-mean-square (rms) error by the square root of $\frac{1}{n}\sum_{i=1}^{n}(d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_{i-1})^2$, where $d_i - \boldsymbol{\kappa}_i^\top \boldsymbol{\alpha}_{i-1} = \epsilon_i$ is the *a priori* error of the sensor i. These results are sketched in Fig. 3 for different values of the step-size parameter. As shown, the optimal bandwidth is given by $\beta_0 = 0.2$, independently of the chosen step-size parameter and with almost the same rms error, except for $\rho = 0.1$. Such exception is mainly due to low convergence resulting from small values of ρ. In order to circumvent such problem, one may increase the number of sensors in the network, or more likely consider several cycles through the network, as proposed in [3]. However, this case is beyond the scope of this work, as we stick to the convergence case given by any of the other studied values of ρ.

To illustrate this convergence behavior, we plot in Fig. 4 the evolution of the (absolute) values of the *a priori* error as we visit each of the sensors, for cases $\rho = 0.1$ and $\rho = 0.7$. For the latter case, the convergence occurs within the first couple of sensors, as shown in the figure. The resulting temperature distribution is illustrated in Fig. 5, where we distinguish sensors contributing to the incrementation of model order. As expected, these sensors span the region under scrutiny.

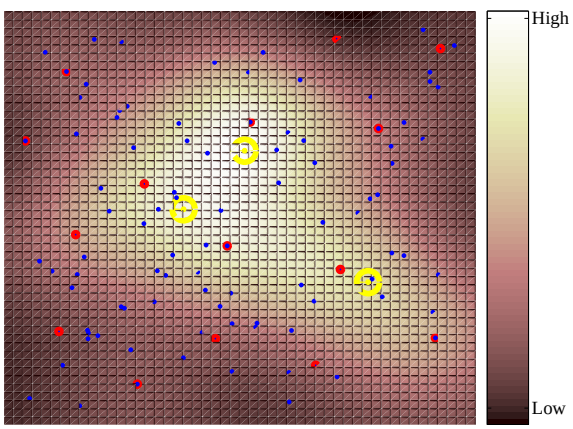

Fig. 5. Resulting temperature map obtained from the Gaussian kernel with $(\rho = 0.7, \beta_0 = 0.2)$. The three heat sources are represented by big yellow-discs ●, sensors by blue-dots ·, and incrementing-order ones by red-discs ●.

VI. Conclusion

In this work, we showed that regression in wireless sensor networks can take advantage of recent developments in sparse representations with kernel machines. In order to derive a reduced-order model, we developed a criterion natural to the topology of the deployed sensors. A gradient-based algorithm is proposed for this purpose, and implementation issues are discussed. Simulation results show good convergence of the proposed approach.

Current and future works concentrate on expanding this approach for monitoring the evolution of the temperature with time. One way to achieve this is to use a reproducing kernel that takes into account both the spatial and the temporal information, for instance by combining kernels defined on each of those domains.

References

[1] M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 316–329, 2006.

[2] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression: an efficient framework for modeling sensor network data," in *Proc. third international symposium on information processing in sensor networks (IPSN)*. New York, NY, USA: ACM, 2004, pp. 1–10.

[3] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. third international symposium on Information Processing in Sensor Networks (IPSN)*. New York, USA: ACM, 2004, pp. 20–27.

[4] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.

[5] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[6] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 134–152, 2005.

[7] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralized detection using kernel methods," *IEEE Trans. Signal Processing*, vol. 53, pp. 4053–4066, 2005.

[8] H. Snoussi and C. Richard, "Distributed bayesian fault diagnosis in collaborative wireless sensor networks," in *Proc. IEEE Globecom*, San Francisco, USA, November 2006, in press.

[9] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed kernel regression: An algorithm for training collaboratively," in *IEEE Proc. Information Theory Workshop*, 2006.

[10] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82–95, 1971.

[11] B. Schölkopf, R. Herbrich, and R. Williamson, "A generalized representer theorem," Royal Holloway College, Univ. of London, UK, Tech. Rep. NC2-TR-2000-81, 2000.

[12] C. Burges, "Simplified support vector decision rules," in *International Conference on Machine Learning*, 1996, pp. 71–77.

[13] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Information Theory*, vol. 50, pp. 2231–2242, 2004.

[14] D. L. Donoho, "Compressed sensing," *IEEE Trans. Info. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[15] L. Hoegaerts, "Eigenspace methods and subset selection in kernel based learning," PhD thesis, Faculty of Engineering, K.U.Leuven, Leuven, Belgium, Jun. 2005.

[16] L. Csató and M. Opper, "Sparse representation for gaussian process models," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 444–450.

[17] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," in *In International Joint Conference on Neural Networks (IJCNN)*, vol. 5, Washington, DC, USA, July 2001, pp. 1244–1249.

[18] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007.

[19] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *submitted to IEEE Trans. Signal Processing*, 2008.