# SCALAR FIELD ESTIMATION USING ADAPTIVE NETWORKS

*Yannick P. Bergamo and Cassio G. Lopes*

Electrical Engineering, University of Sao Paulo
Escola Politecnica, Sao Paulo, Brazil, 05508-970
yannick@usp.br, cassio@lps.usp.br

**ABSTRACT**

A new method for estimating scalar fields is proposed employing adaptive networks. The theoretical problem of function approximation is posed and the networked solution via distributed adaptive algorithms is introduced. Two approximate solutions employing adaptive networks are considered, and it is shown that both cases approach the theoretical solution in the limit case. In the first setup, nodes remain anchored during the network operation. The second version involves node relocation, either entirely at random over the region of interest or drifting according to a random walk model. Simulations illustrate the method for estimating a 2D scalar field.

***Index Terms*—** Scalar fields, adaptive networks, distributed estimation

## 1. INTRODUCTION

Scalar fields arise naturally in a myriad of scenarios, such as precision agriculture, meteorological measures, disaster relief, pollution control, etc. Often they need to be properly modelled, in order to study a variable of interest covering a (geographic) area of interest, or even take action as consequence of the estimated field. A typical approach is to combine a pre-selected set of continuous basis functions, so that the field of interest is modelled. The set of combiners is chosen so that the error between the model and the actual field over the region of interest is kept within a desired accuracy. For linear combiners, a set of coefficients may be calculated optimally in the least-squares sense, for instance.

This paper shows how the scalar field estimation problem may be posed in the adaptive network (AN) framework [1, 2]. By collecting a set of field measurements at the nodes location, a network of nodes running a distributed adaptive algorithm is able to cooperatively learn the set of combiners that, together with the basis functions, model the scalar field of interest. This is achieved in a fully distributed manner.

A brief performance study is carried out in two scenarios. In the first, nodes are placed randomly over the region of interest and remain stationary; in agreement with intuition, it is shown that as the number of nodes increases, the AN solution approaches the theoretical solution. In the second scenario, nodes may move freely, and it is shown that if they cover properly the region of interest, the AN may in the long run achieve the theoretical solution on the mean. Simulations illustrate the results obtained by the proposed method.

## 2. PROBLEM FORMULATION: LEAST SQUARES SOLUTION

Let us consider a (geographical) region $X$ and a scalar field $F$ defined over this region. Assume further that $X$ is a compact subset of $\mathbb{R}^p$ and that $F \in \mathcal{C}(X)$, where $\mathcal{C}(X)$ is the linear space of all continuous functions from $X$ to $\mathbb{R}$. Given a (linearly independent) set of basis functions $B = \{h_l \in \mathcal{C}(X) : l = 1, \ldots, M\}$, the best approximation for $F$, say in the least-squares sense, may be found via a linear combination of the functions in $B$.

Specifically, denote by $\mathcal{H}$ the subspace spanned by the functions in $B$, and let $H(x) = [h_1(x) \ldots h_M(x)]$ be the row vector resulting from evaluating each function $h_l \in B$ at the point $x \in X$. We are then interested in finding the column vector $\omega^o \in \mathbb{R}^M$ that solves

$$\min_{\omega} J_D(\omega), \quad J_D(\omega) \triangleq \int_X [F(x) - H(x)\omega]^2 \, dx \tag{1}$$

If we endow $\mathcal{C}(X)$ with the canonical inner product

$$\langle f, g \rangle \triangleq \int_X f(x)g(x) \, dx \tag{2}$$

and since $\mathcal{H}$ is finite dimensional, the problem is readily solved by the orthogonal projection of $F$ over $\mathcal{H}$ [3], which can be done by solving the normal equations

$$G\omega = y \tag{3}$$

where $[G]_{i,j} = \langle h_i, h_j \rangle$ and $y_l = \langle h_l, F \rangle$. The solution so obtained, which we denote by $\hat{F}$, is then of the form

$$\hat{F}(x) = H(x)\omega^o = \sum_{l=1}^{M} h_l(x)[\omega^o]_l \tag{4}$$

The resulting error $\varepsilon(x) \triangleq F(x) - \hat{F}(x)$ is orthogonal to $\mathcal{H}$, i.e., $\langle \varepsilon, h_l \rangle = 0, \quad l = 1, \ldots, M$.

Figure 1 depicts an example where $X = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ and the scalar field is given by

$$F(x_1, x_2) = \frac{0.7}{\sqrt{1 + 20 \cdot (x_1 - 0.15)^2 + 20 \cdot (x_2 - 0.15)^2}} + \frac{1}{\sqrt{1 + 15 \cdot (x_1 - 0.7)^2 + 15 \cdot (x_2 - 0.7)^2}} \tag{5}$$

This example is used throughout the paper. For illustration purposes, we select $M = 36$ Gaussian basis functions

$$h_l(x) = \exp(-a_l \|x - c_l\|^2) \tag{6}$$

where the $\{c_l\}$ are evenly distributed over $X$ (black dots in Fig. 1(d)) and $a_l = a$ was chosen so that $h_l(x) = 0.15$ on a neighboring centroid, making each individual function reasonably relevant within the scope of its nearby functions. The solution and the corresponding

residual error were obtained, respectively, from (3) and (1) by numerical integration, since, in general, a closed form solution cannot be obtained; the resulting residual error was $J_D(\omega^o) = 2.11 \cdot 10^{-4}$. The residual error may be arbitrarily decreased by, for example, increasing the number of basis functions in the region, since any function in $\mathcal{C}(X)$ can be approximated arbitrarily well by a linear combination of Gaussian functions [4].

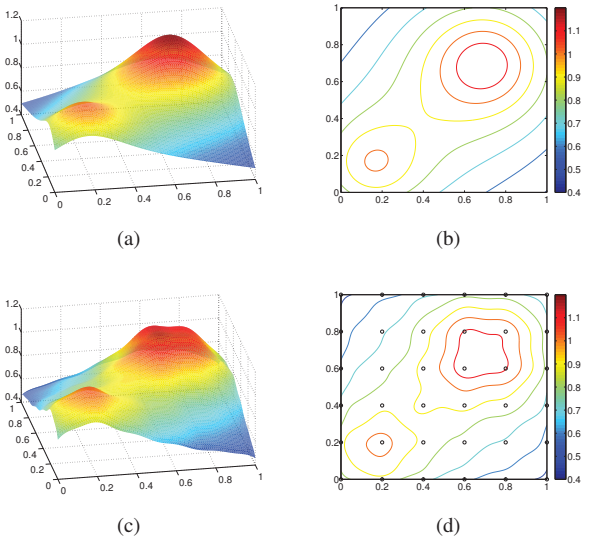

**Fig. 1**: (a), (b): surface and contour plot for the scalar field $F$ defined by (5). (c), (d): same plots obtained with the least squares estimate as in (1). The black dots on Fig. 1(d) are the Gaussian functions' centroids $\{c_l\}$.

## 3. THE ADAPTIVE NETWORK SOLUTION

In this section, we proceed to show how the theoretical solution from the previous section can be solved in a practical scenario, with arbitrary accuracy, by an adaptive network. For that matter, at time $i$ a set of $N$ nodes is placed in the field $X$. We assume that each node knows its own current position $p_{k,i}$. The regressor has samples of the basis functions at the node location, i.e., $u_{k,i} = H(p_{k,i})$ and a noisy version of the scalar field is captured by the desired signal $d_k(i) = F(p_{k,i}) + v_k(i)$. Note that $\{d_k(i), u_{k,i}\}$ are realizations of the random quantities[1]

$$\mathbf{d}_k = F(\mathbf{p_k}) + \mathbf{v}_k \tag{7a}$$

$$\mathbf{u}_k = H(\mathbf{p}_k) \tag{7b}$$

where $\mathbf{v}_k$ is a zero mean white Gaussian process with variance $\sigma_v^2$ independent of each $\mathbf{p}_k$. The randomness in the regressor comes from the fact that the nodes, in the general case, are randomly located[2]. We can now pose the problem that the network will try to solve (see Figure 2). Collecting the data into

$$\mathbf{U}_c \triangleq \text{col}\{\mathbf{u}_1, \ldots, \mathbf{u}_N\} \ (N \times M) \tag{8a}$$

$$\mathbf{d} \triangleq \text{col}\{\mathbf{d}_1, \ldots, \mathbf{d}_N\} \ (N \times 1) \tag{8b}$$

[1] In this work, random quantities are boldfaced and their realizations are normal font.

[2] Node's $k$ current position $p_{k,i}$ is a realization of $\mathbf{p}_k$

then the task of the adaptive network is to find $w^o$ that solves [1]

$$\min_w \mathsf{E}\,\|\mathbf{d} - \mathbf{U}_c w\|^2 \tag{9}$$

We would like to have $w^o$ from (9) approaching $\omega^o$ from (1).

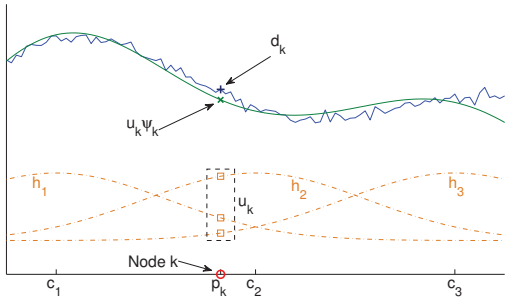


**Fig. 2**: Method setup.

Although several distributed algorithms may be employed to solve (9) adaptively [2, 5, 6, 7, 8, 9, 10], the setup will be illustrated via the standard diffusion LMS algorithm (dLMS) [1]. For the dLMS, at every time step $i$, each node $k$ has access to a set of estimates $\{\psi_k^{(i-1)}\}_{k\in\mathcal{N}_{k,i-1}}$ from its neighborhood $\mathcal{N}_{k,i-1}$, which is the subset of nodes currently connected to node $k$, including itself. Each node then fuses its estimate with those from its neighbors into $\phi_k^{(i-1)}$, and then updates its own, according to the following update rule

$$\phi_k^{(i-1)} = \sum_{l\in\mathcal{N}_{k,i-1}} a_{kl}\psi_l^{(i-1)}, \qquad \phi_k^{(-1)} = 0 \tag{10a}$$

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right) \tag{10b}$$

where $\{a_{kl} \geq 0\}$ are convex combiner coefficients, and $\mu_k$ is the step size for node $k$. The spatial fusion step represented by equation (10a) is mandatory for proper scalar field interpolation, not necessarily for mean square performance improvement as in the original data model formulation for ANs [1, 2].

The network topology is defined assuming that each node is able to communicate with its neighboring nodes within a predetermined range. Two nodes are connected when they are less than $s$ units of distance apart, that is

$$\mathcal{N}_{k,i} = \{l \ : \ \|p_{k,i} - p_{l,i}\| < s\} \tag{11}$$

## 4. ASSESSING NETWORK PERFORMANCE

We would like to assess the impact of the adaptive network solution when compared with the theoretical solution from Section 2. In other words we would like to asses under which conditions $w^o \to \omega^o$. As a measure of performance we adopt the global square deviation (MSD)

$$\eta(i) = \frac{1}{N}\sum_{k=1}^{N} \mathsf{E}\,\|\omega^o - \psi_k^{(i-1)}\|^2 \tag{12}$$

that measures how far is, on average, the network solution from the optimal obtained by solving the classical least squares problem (1).

Let the network cost function be

$$J(w) \triangleq \frac{|X|}{N}\mathsf{E}\,\|\mathbf{d} - \mathbf{U}_c w\|^2 \tag{13}$$

where $|X| = \int_X dx$, so that minimizing $J$ is equivalent to solving the problem (9). Expanding (13), together with the data model (7), yields

$$\|\mathbf{d} - \mathbf{U}_c w\|^2 = \sum_{l=1}^{N} [\mathbf{d}_l - \mathbf{u}_l w]^2 = \sum_{l=1}^{N} [F(\mathbf{p}_l) - H(\mathbf{p}_l)w]^2 - 2\sum_{l=1}^{N} [F(\mathbf{p}_l) - H(\mathbf{p}_l)w]\mathbf{v}_l + \sum_{l=1}^{N} \mathbf{v}_l^2 \quad (14)$$

Taking expectations (recall that $\mathbf{v}_l$ is zero mean and independent of $\mathbf{p}_l$) and multiplying by $|X|/N$ we have

$$J(w) = \frac{|X|}{N} \sum_{l=1}^{N} \mathsf{E}\,[F(\mathbf{p}_l) - H(\mathbf{p}_l)w]^2 + |X|\sigma_v^2 \quad (15)$$

Two scenarios of interest are considered from (15) in the sequel, both converging in the limit to the theoretical solution (1).[3]

### 4.1. Anchored nodes

In the first scenario we assume the nodes are deployed randomly at start and remain fixed in their locations, so that, as far as the AN operation is concerned, the deterministic quantities $\{p_k\}$ are realizations of the uniformly distributed random variables $\{\mathbf{p}_k\}$.

Recall that for some $f \in \mathcal{C}(X)$, if the points $\{x_1, \ldots x_N\}$ are i.i.d. according to a uniform distribution, then

$$\frac{|X|}{N} \sum_{l=1}^{N} f(x_l) \approx \int_X f(x)\, dx \quad (16)$$

with the corresponding error variance approaching zero at a rate proportional to $1/N$. Thus, the first term in (15) can be treated as an approximation $\hat{J}_D$ for $J_D$:

$$\hat{J}_D(w) = \frac{|X|}{N} \sum_{l=1}^{N} [F(p_l) - H(p_l)w]^2 \quad (17)$$

For fixed nodes, we can rewrite the network cost function as

$$J(w) = \hat{J}_D(w) + |X|\sigma_v^2 \quad (18)$$

Thus, as long as the number of nodes is sufficiently high, minimizing $J$ leads to a good approximation to the problem in (1). In other words, as N tends to infinity, (17) tends to an integral and equation (1) is recovered.

### 4.2. Relocating the nodes

An alternative for increasing the number of nodes is to reposition them regularly, which enhances the coverage of the scalar field. As a limit scenario, we consider the case where the nodes are randomly relocated at each iteration: the random variables $\{\mathbf{p}_k\}$ are uniformly distributed over $X$ during the operation of the network. Regarding the summand in the first term of (15) we have

$$\mathsf{E}\,[F(\mathbf{p}_l) - H(\mathbf{p}_l)w]^2 = \int_X [F(x) - H(x)w]^2 f_{p_l}(x)\, dx = \frac{1}{|X|} \int_X [F(x) - H(x)w]^2\, dx \quad (19)$$

[3] Adaptive networks run adaptive algorithms that are subject to gradient noise, also introducing deviations from the optimal solution [1, 11].

Thus, the sum in (15) results exactly in $J_D$, and we can write the network cost function as

$$J(w) = J_D(w) + |X|\sigma_v^2 \quad (20)$$

Hence, minimizing $J$ is equivalent to minimizing $J_D$.

### 4.3. Random walk

A more practical solution to achieve the optimal solution, instead of increasing ad infinitum the number of nodes, or random relocation, is to make the nodes sample properly the region of interest. In fact, there are several strategies to allocate the sensors [12]. We test here a simple strategy based on a random walk model, where the nodes drift around according to

$$\mathbf{p}_{k,i} = \mathbf{p}_{k,i-1} + \mathbf{q}_{k,i} \quad (21)$$

where $\mathbf{q}_{k,i}$ is a zero mean white Gaussian process, with variance $\sigma_q^2 = 1/N^2$. In our simulations, in order to constrain $\mathbf{p}_{k,i}$ within $X$, we included a hard limiter on each coordinate, restricting its value to the interval $[0, 1]$. Intuitively, in the long run, given a reasonable number of nodes, the whole region will be visited, therefore the solution should tend to the optimal. This is corroborated by simulations. Formal proof will be provided in future work, along with a suitable redirection strategy.

## 5. SIMULATIONS

We tested our method with the scalar field defined in (5), and considering the three scenarios described on section 4: anchored nodes, random relocation and random walk. In each case we present the ensemble average MSD, averaged over 100 experiments. The noise variance $\sigma_v^2$ was set to 0.07, corresponding to an average SNR of 10 dB across $X$. The diffusion LMS employs $\mu_k = \mu = 1$, and the combiner coefficients were selected according to the nearest neighbor rule:

$$a_{kl} = \begin{cases} \frac{1}{|\mathcal{N}_k|}, & l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

where the neighborhood set is determined according to (11) with $s = \sqrt{0.1}$.

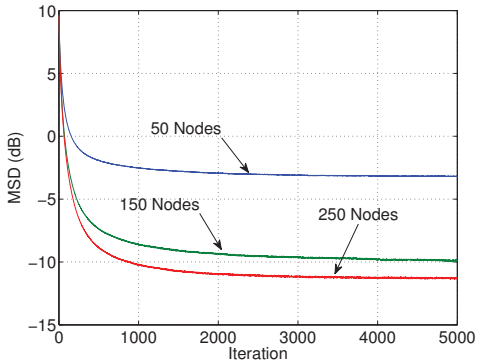


**Fig. 3**: Ensamble average of the global MSD for the anchored nodes scenario

Figure 3 shows the ensemble average for the anchored nodes case. As expected, increasing the number of nodes $N$ resultes in

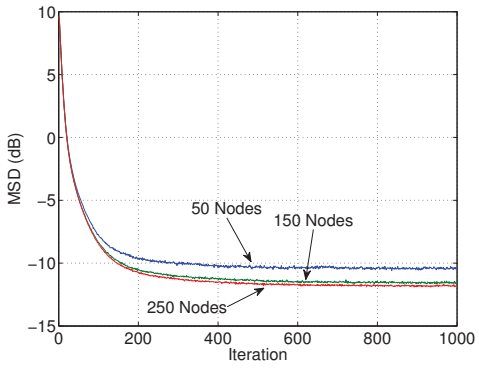


**Fig. 4**: Ensamble average of the global MSD for scenario where the nodes are randomly relocated at each iteration.

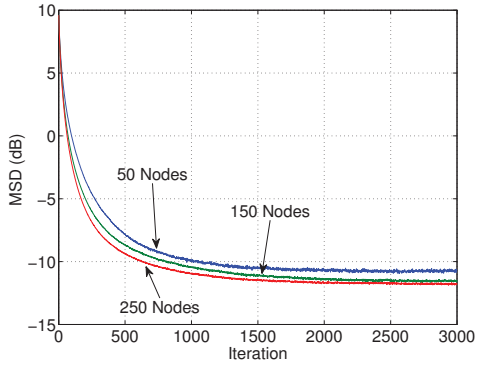


**Fig. 5**: Ensamble average of the global MSD for the random walk scenario.

better performance. For this scenario we also show, in Figure 6, the surface and contour plot of the average weight [4]

$$\frac{1}{N}\sum_{k=1}^{N}\psi_k^{(\infty)} \tag{23}$$

obtained by the network for $N = 50$ nodes; in all other cases (Figure (3) for $N = 150$ and $N = 250$ and Figure 4 and 5) the analogous plots are visually indistinguishable from the ones in Figure 1(c) and 1(d).

Figures 4 and 5 show the ensemble average MSD curves for, respectively, the random relocation and random walk model. Again the results corroborates the study from section 4 reaching, in all cases, practically the same error in steady state. Note that when nodes are allowed to move, even for small $N$, the solution gets considerably better as compared to the anchored nodes' case (compare Figures 4 and 5 with Figure 3).

## 6. CONCLUSION AND FUTURE WORK

A new method for estimating scalar fields has been introduced employing the concept of adaptive networks. Simulations show that for a sufficient number of nodes, the agreement between the network solution and the actual scalar field is as accurate as desired. Repositioning the nodes regularly may be an alternative to increasing the number of nodes, provided that the relocation strategy covers reasonably well the field of interest.

Future work involves deriving relocation strategies that further improve network performance, including time varying scalar fields. Different distributed adaptive algorithms [10, 9] will be considered, as well as analytical models for the network error evolution.

[4]For illustration purposes only: no global information is required in our method.

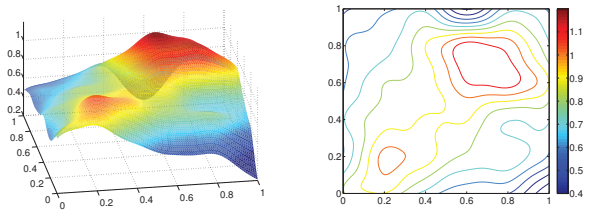


**Fig. 6**: (a), (b): surface and contour plot for the average network weight in the anchored node scenario, for $N = 50$.

## 7. REFERENCES

[1] C.G. Lopes and A.H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122 –3136, july 2008.

[2] F.S. Cattivelli and A.H. Sayed, "Diffusion lms strategies for distributed estimation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035 –1048, march 2010.

[3] Sheldon Axler, *Linear Algebra Done Right*, Springer, 1996.

[4] Federico Girosi and Tomaso Poggio, "Networks and the best approximation property," Tech. Rep., Cambridge, MA, USA, 1989.

[5] N. Takahashi, I. Yamada, and A.H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 58, no. 9, pp. 4795 –4810, sept. 2010.

[6] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *Signal Processing, IEEE Transactions on*, vol. 59, no. 10, pp. 4692 –4707, oct. 2011.

[7] Leilei Li, J.A. Chambers, C.G. Lopes, and A.H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *Signal Processing, IEEE Transactions on*, vol. 58, no. 1, pp. 151 –164, jan. 2010.

[8] I.D. Schizas, G. Mateos, and G.B. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2365 –2382, june 2009.

[9] C.G. Lopes and A.H. Sayed, "Randomized incremental protocols over adaptive networks," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 3514 –3517.

[10] C.G. Lopes and A.H. Sayed, "Diffusion adaptive networks with changing topologies," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 3285 –3288.

[11] Ali H. Sayed, *Adaptive Filters*, Wiley-IEEE Press, 2008.

[12] Huiyu Luo, Xiangming Kong, and G. Pottie, "An adaptive algorithm for sampling two-dimensional fields using mobile sensors," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, april 2007, vol. 1, pp. I–857 –I–860.