DIFFUSION ADAPTATION OVER NETWORKS WITH KERNEL LEAST-MEAN-SQUARE

Wei Gao [†‡] *Jie Chen* [‡] *Cédric Richard* [†] *Jianguo Huang* [‡]

[†]Université de Nice Sophia-Antipolis, Nice, France
wei.gao.uns@gmail.com dr.jie.chen@ieee.org cedric.richard@unice.fr

[‡]Northwestern Polytechnical University, Xi'an, China
jghuang@nwpu.edu.cn

ABSTRACT

Distributed learning over networks has become an active topic of research in the last decade. Adaptive networks are suitable for decentralized inference tasks, e.g., to monitor complex natural phenomena or infrastructure. Most of works focus on distributed estimation methods of linear regression models. However, there are many important applications that deal with nonlinear parametric models to be fitted, in a collaborative manner. In this paper, we derive functional diffusion strategies in reproducing kernel Hilbert spaces.

1. INTRODUCTION

Distributed learning over networks allows a set of interconnected agents to perform preassigned tasks such as detection and estimation from streaming data. Potential applications include, for instance, natural phenomena and infrastructure monitoring. Due to energy constraints, limited communication capabilities and large scale networks, signal processing strategies have moved from centralized solutions with a fusion center [1] to decentralized cooperative solutions with in-network sensor data processing. For online parameter estimation, a variety of distributed strategies have been proposed. These include incremental strategies [2], consensus strategies [3] and diffusion strategies [4]. With diffusion modes of cooperation, the agents cooperate with each other through local interactions that consist of exchanging raw data and local estimates. Diffusion strategies are attractive because they are scalable, robust, and enable continuous adaptation and learning.

Decentralized detection and estimation have often been considered with parametric models, in which the statistics of observations are assumed known. Such assumptions are usually motivated by prior application-specific domain knowledge. Robust nonparametric methods are however desirable when few prior information is available. To address such situations, nonparametric methods based on kernel functions were primarily considered for decentralized detection and estimation over networks [1, 5]. The successive orthogonal projection (SOP) algorithm was derived to address distributed learning problems over networks with kernel-based models [6, 7]. An incremental kernel-based strategy was introduced in [8, 9]. A linear combination of Gaussian functions was considered in [10] for estimating scalar fields with diffusion networks.

In this paper, we introduce functional diffusion strategies in reproducing kernel Hilbert spaces with distributed KLMS algorithm. This paper is organized as follows. In Section 2, we introduce some basic principles on online learning with KLMS. In Section 3, we derive a functional framework for diffusion adaption over networks. In Section 4, we present some illustrative simulation results.

This work was supported by ANR/DGA grant ANR-13-ASTR-0030, and by National Natural Science Foundation of China grant 61271415.

2. THE KERNEL LEAST-MEAN-SQUARE ALGORITHM

Let $\mathcal{H}$ be a Hilbert space of functions ψ from a subspace $\mathcal{U}$ of $\mathbb{R}^L$ to $\mathbb{R}$. Assume that $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS), that is, there exists a map $\kappa : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ such that:

$$\forall \boldsymbol{u} \in \mathcal{U}, \quad \kappa(\boldsymbol{u}, \cdot) \in \mathcal{H} \tag{1a}$$

$$\forall \psi \in \mathcal{H}, \quad \psi(\boldsymbol{u}) = \langle \psi, \kappa(\boldsymbol{u}, \cdot) \rangle_{\mathcal{H}} \tag{1b}$$

Property (1b) is called the reproducing property of the RKHS. Replacing ψ by $\kappa(\boldsymbol{u}_j, \cdot)$ in this property yields

$$\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \langle \kappa(\boldsymbol{u}_i, \cdot), \kappa(\boldsymbol{u}_j, \cdot) \rangle_{\mathcal{H}} \tag{2}$$

for all $\boldsymbol{u}_i, \boldsymbol{u}_j \in \mathcal{U}$. Every RKHS is characterized by a unique reproducing kernel κ. This kernel is positive definite, that is, κ is a symmetric function and satisfies

$$\sum_{i,j=1}^{n} q_i q_j \kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) \geq 0 \tag{3}$$

for all $n \geq 0$, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in \mathcal{U}$, and $q_1, \ldots, q_n \in \mathbb{R}$. In the sequel, we make the weak assumption that κ is bounded.

Consider the kernel least-squares problem. Given pairs of input vectors and desired outputs $\{(\boldsymbol{u}_i, d(i))\}_i$, which satisfy the model

$$d(i) = \psi^o(\boldsymbol{u}_i) + v(i) \tag{4}$$

where $v(i)$ is a zero-mean white noise with power σ_v^2, the problem is to estimate ψ^o such that:

$$\psi^o = \arg\min_{\psi \in \mathcal{H}} J(\psi) \quad \text{with} \quad J(\psi) = \mathbb{E}\{|d(i) - \psi(\boldsymbol{u}_i)|^2\} \tag{5}$$

Calculating the Fréchet derivative of $J(\psi)$ with respect to ψ we find:

$$\nabla J(\psi) = -2\,\mathbb{E}\{[d(i) - \psi(\boldsymbol{u}_i)]\,\kappa(\cdot, \boldsymbol{u}_i)\} \tag{6}$$

The desired function ψ^o satisfies the normal equation:

$$\mathbb{E}\{\psi^o(\boldsymbol{u}_i)\,\kappa(\cdot, \boldsymbol{u}_i)\} = \mathbb{E}\{d(i)\,\kappa(\cdot, \boldsymbol{u}_i)\} \tag{7}$$

It is seen from (5) that:

$$\begin{aligned} J(\psi) &= \sigma_d^2 - 2\langle \mathbb{E}\{d(i)\,\kappa(\cdot, \boldsymbol{u}_i)\}, \psi \rangle_{\mathcal{H}} + \mathbb{E}\{\psi^2(\boldsymbol{u}_i)\} \\ &= \sigma_d^2 - 2\langle \mathbb{E}\{\psi^o(\boldsymbol{u}_i)\,\kappa(\cdot, \boldsymbol{u}_i)\}, \psi \rangle_{\mathcal{H}} + \mathbb{E}\{\psi^2(\boldsymbol{u}_i)\} \\ &= \sigma_d^2 - 2\mathbb{E}\{\langle \psi^o(\boldsymbol{u}_i), \psi(\boldsymbol{u}_i) \rangle_{\mathcal{H}}\} + \mathbb{E}\{\psi^2(\boldsymbol{u}_i)\} \end{aligned} \tag{8}$$

where σ_d^2 denotes $\mathbb{E}\{d^2(i)\}$. The first and the third equalities follow from (1b). The second equality follows from (7). We see that $J(\psi^o) = \sigma_d^2 - \mathbb{E}\{[\psi^o(\boldsymbol{u}_i)]^2\}$. A completion-of-squares argument finally shows that $J(\psi)$ can be expressed as

$$J(\psi) = J(\psi^o) + \mathbb{E}\{[\psi(\boldsymbol{u}_i) - \psi^o(\boldsymbol{u}_i)]^2\} \tag{9}$$

We shall use this expression in the sequel, where $J(\psi^o)$ and $J_{\min}$ will interchangeably denote the minimum cost value of $J(\psi)$.

The optimal implementation (7) for determining ψ^o requires knowledge of the data moments. This information is usually unavailable. Stochastic-gradient methods are popular adaptive learning algorithms obtained from gradient-descent implementations by replacing the required derivatives by some suitable approximations. One of the simplest approximations for $\nabla J(\psi)$ consists of replacing the random variables in (5) by the observations at iteration i, namely,

$$-\nabla J(\psi) \approx [d(i) - \psi(\boldsymbol{u}_i)]\,\kappa(\cdot, \boldsymbol{u}_i) \tag{10}$$

The corresponding steepest-descent recursion is widely known as the kernel least-mean-squares (KLMS) algorithm [11]:

$$\psi_i = \psi_{i-1} + \mu\,[d(i) - \psi_{i-1}(\boldsymbol{u}_i)]\,\kappa(\cdot, \boldsymbol{u}_i), \tag{11}$$

where μ is a small positive step-size. Despite its computational simplicity, the main drawback of KLMS is that an increasing number of kernel functions $\kappa(\cdot, \boldsymbol{u}_i)$ is involved in the estimation process as new data $\boldsymbol{u}_i$ are collected. To overcome this limitation, finite-size models of the form

$$\psi = \sum_{j=1}^{M} \alpha_j\,\kappa(\,\cdot\,, \boldsymbol{u}_{\omega_j}) \tag{12}$$

and sparsity-promoting strategies are usually considered in the literature [11], where $\mathcal{D} = \{\kappa(\cdot, \boldsymbol{u}_{\omega_j})\}_j$ is a dictionary learnt from the input data $\{\kappa(\cdot, \boldsymbol{u}_i)\}_i$. Then, KLMS reduces to a two-alternative choice procedure at each instant i: (Case 1) a dictionary learning stage that inserts $\kappa(\cdot, \boldsymbol{u}_i)$ into $\mathcal{D}_{i-1}$ if some given sparsification rule, such as the coherence rule below, is satisfied; (Case 2) Otherwise, an adaptation step to update the vector $\boldsymbol{\alpha}$ of parameters α_j.

- Case 1: $\max_{j=1,\ldots,\text{card}(\mathcal{D}_{i-1})} |\kappa(\boldsymbol{u}_i, \boldsymbol{u}_{\omega_j})| \leq \delta_0$

$$\begin{aligned} \boldsymbol{\alpha}_i &= \begin{pmatrix} \boldsymbol{\alpha}_{i-1} \\ 0 \end{pmatrix} + \mu\, e_i\, \boldsymbol{\kappa}_i \\ \mathcal{D}_i &= \mathcal{D}_{i-1} \cup \{\kappa(\cdot, \boldsymbol{u}_i)\} \end{aligned} \tag{13}$$

- Case 2: $\max_{j=1,\ldots,\text{card}(\mathcal{D}_{i-1})} |\kappa(\boldsymbol{u}_i, \boldsymbol{u}_{\omega_j})| > \delta_0$

$$\begin{aligned} \boldsymbol{\alpha}_i &= \boldsymbol{\alpha}_{i-1} + \mu\, e_i\, \boldsymbol{\kappa}_i \\ \mathcal{D}_i &= \mathcal{D}_{i-1} \end{aligned} \tag{14}$$

where $\boldsymbol{\kappa}_i = [\kappa(\boldsymbol{u}_i, \boldsymbol{u}_{\omega_1}), \ldots, \kappa(\boldsymbol{u}_i, \boldsymbol{u}_{\omega_{\text{card}(\mathcal{D}_{i-1})}})]^\top$, and δ_0 is a parameter in $[0, 1)$ determining both the level of sparsity and the coherence of the dictionary. In [11], it is shown that the dictionary learning step converges to a dictionary $\mathcal{D}$ of finite size, say M, and the algorithm above reduces to (14) after a finite number of iterations. An analysis of this algorithm is proposed in [12], and a sparse dictionary learning strategy is introduced in [13].

3. DIFFUSION ADAPTATION WITH KLMS

Consider a collection of N agents interested in estimating the same function ψ^o of $\mathcal{H}$ from data realizations $(\boldsymbol{u}_{k,i}, d_k(i))$, which satisfy a model of the form

$$d_k(i) = \psi^o(\boldsymbol{u}_{k,i}) + v_k(i) \tag{15}$$

where $v_k(i)$ is a zero-mean white noise with power $\sigma_{v,k}^2$. To recover this unknown function ψ^o, our strategy is to optimize the following global cost function in a distributed manner:

$$J(\psi) = \sum_{k=1}^{N} \mathbb{E}\{|d_k(i) - \psi(\boldsymbol{u}_{k,i})|^2\} \tag{16}$$

Assume that the set of neighbors connected with the ℓ-th agent is fixed and denoted by $\mathcal{N}_\ell$. We can express $J(\psi)$ as follows:

$$\begin{aligned} J(\psi) &= \sum_{\ell=1}^{N} J_\ell^{\text{loc}}(\psi) \\ \text{with} \quad J_\ell^{\text{loc}}(\psi) &= \sum_{k\in\mathcal{N}_\ell} c_{k\ell}\, \mathbb{E}\{|d_k(i) - \psi(\boldsymbol{u}_{k,i})|^2\} \end{aligned} \tag{17}$$

where $\{c_{k\ell}\}$ is a set of nonnegative coefficients, freely chosen by the designer, that satisfy:

$$c_{k\ell} = 0 \text{ if } k \notin \mathcal{N}_\ell \quad \text{and} \quad \sum_{\ell=1}^{N} c_{k\ell} = 1 \tag{18}$$

We collect the coefficients $\{c_{k\ell}\}$ into an $N \times N$ matrix $\boldsymbol{C}$, which is right stochastic since each row of $\boldsymbol{C}$ adds up to one.

Consider the local cost function $J_\ell^{\text{loc}}(\psi)$ at each node ℓ. It follows from (9) that:

$$J_\ell^{\text{loc}}(\psi) = J_{\ell,\min}^{\text{loc}} + \sum_{k\in\mathcal{N}_\ell} c_{k\ell}\, \mathbb{E}\{|\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\} \tag{19}$$

where $J_{\ell,\min}^{\text{loc}} = J_\ell^{\text{loc}}(\psi^o)$. Substituting (19) into (17), and dropping the term that does not depend on ψ, we obtain the following alternative global cost function:

$$J(\psi) = J_n^{\text{loc}}(\psi) + \sum_{\ell\neq n} \sum_{k\in\mathcal{N}_\ell} c_{k\ell}\, \mathbb{E}\{|\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\} \tag{20}$$

In this expression, the minimizer ψ^o in the correction term that relates the global cost function to the local cost function at every node n is not known since the nodes wish to estimate it. This issue is addressed in the sequel. Likewise, not all information needed to compute the expected value are available to node n since it can only have access to information from its neighbors. We thus introduce the modified local cost function at node n:

$$\begin{aligned} J_n(\psi) = J_n^{\text{loc}}(\psi) &+ \\ \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} \sum_{k\in\mathcal{N}_\ell} & c_{k\ell}\, \mathbb{E}\{|\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\} \end{aligned} \tag{21}$$

The probability density functions required to calculate the expected values may not be available because often nodes can only observe realizations $\boldsymbol{u}_{k,i}$. To address this issue, note that:

$$\begin{aligned} &\mathbb{E}\{|\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\} \\ &\quad = \int |\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\, dP(\boldsymbol{u}_{k,i}) \\ &\quad \leq \|\psi - \psi^o\|_{\mathcal{H}}^2 \int \kappa(\boldsymbol{u}_{k,i}, \boldsymbol{u}_{k,i})\, dP(\boldsymbol{u}_{k,i}) \\ &\quad \leq M\|\psi - \psi^o\|_{\mathcal{H}}^2 \end{aligned} \tag{22}$$

where P is a probability measure. The first and the second inequality follow from the Cauchy-Schwarz inequality and the boundedness of

the kernel κ, respectively. We suggest to replace the second term on the right-hand side of (21) by the following upper-bound:

$$\sum_{k\in\mathcal{N}_\ell} c_{k\ell}\,\mathbb{E}\{|\psi(\boldsymbol{u}_{k,i}) - \psi^o(\boldsymbol{u}_{k,i})|^2\} \le b_{\ell n}\|\psi-\psi^o\|_{\mathcal{H}}^2 \tag{23}$$

where $b_{\ell n}$ is some nonnegative coefficient. The modified cost function (21) is then relaxed as follows:

$$J_n'(\psi) = J_n^{\text{loc}}(\psi) + \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} b_{\ell n}\|\psi-\psi^o\|_{\mathcal{H}}^2 \tag{24}$$

With the exception of ψ^o, the cost (24) at node n relies solely on information available to this node from its neighborhood.

Node n can compute successive steepest-descent iterations to minimize $J_n'(\psi)$. Let $\psi_{n,i-1}$ be the estimate for ψ^o by node n at time $i-1$. The update from $\psi_{n,i-1}$ to $\psi_{n,i}$ can be performed as:

$$\psi_{n,i} = \psi_{n,i-1} - \mu_n \nabla J_n'(\psi_{n,i-1}), \quad \psi_{n,-1} = \text{initial guess} \tag{25}$$

where μ_n is a small positive step size at node n. Computing the Fréchet derivative of (24), and dropping the expectation operator from the definition of $J_n^{\text{loc}}(\psi)$ to use instantaneous approximations instead, we get:

$$\begin{aligned}\psi_{n,i} &= \psi_{n,i-1} \\ &\quad + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \psi_{n,i-1}(\boldsymbol{u}_{\ell,i})]\,\kappa(\cdot,\boldsymbol{u}_{\ell,i}) \\ &\quad + \mu_n \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} b_{\ell n}(\psi^o - \psi_{\ell,i-1})\end{aligned} \tag{26}$$

Among other possible forms, we can implement (26) in two successive steps involving each one a correction term as follows:

$$\begin{aligned}\varphi_{n,i} &= \psi_{n,i-1} \\ &\quad + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \psi_{n,i-1}(\boldsymbol{u}_{\ell,i})]\,\kappa(\cdot,\boldsymbol{u}_{\ell,i-1})\end{aligned} \tag{27a}$$

$$\psi_{n,i} = \varphi_{n,i} + \mu_n \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} b_{\ell n}(\psi^o - \psi_{n,i-1}) \tag{27b}$$

First, in (27b), neither node n nor its neighbors know the optimum function ψ^o. Each node ℓ can however use its local intermediate estimate $\varphi_{\ell,i}$ as an approximation. Second, $\psi_{n,i-1}$ in (27b) can be advantageously replaced by $\varphi_{n,i}$ since it is obtained by incorporating information from the neighbors in (27a). Step (27b) then becomes:

$$\psi_{n,i} = \left(1 - \mu_n \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} b_{\ell n}\right)\varphi_{n,i} + \mu_n \sum_{\ell\in\mathcal{N}_n} b_{\ell n}\,\varphi_{\ell,i} \tag{28}$$

We introduce the following weighting coefficients:

$$\begin{aligned}a_{nn} &= 1 - \mu_n \sum_{\ell\in\mathcal{N}_n\setminus\{n\}} b_{\ell n} \\ a_{\ell n} &= \mu_n\, b_{\ell n}, \quad \ell\in\mathcal{N}_n\setminus\{n\} \\ a_{\ell n} &= 0, \quad \ell\notin\mathcal{N}_n\end{aligned} \tag{29}$$

and collect these coefficients into an $N\times N$ matrix $\boldsymbol{A}$. For sufficiently small step-sizes μ_n, observe that the coefficients $\{a_{\ell n}\}$ are nonnegative and each column of $\boldsymbol{A}$ adds up to one. Just like the coefficients $\{c_{\ell n}\}$, the coefficients $\{a_{\ell n}\}$ can be freely chosen by the designer provided that $\boldsymbol{A}$ is left stochastic.

3.1. Functional Adapt-then-Combine (FATC) diffusion strategy

Substituting the so-called coefficients $\{a_{\ell n}\}$ into (28), we arrive at the following diffusion strategy:

$$\boxed{\begin{aligned}\varphi_{n,i} &= \psi_{n,i-1} \\ &\quad + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \psi_{n,i-1}(\boldsymbol{u}_{\ell,i})]\,\kappa(\cdot,\boldsymbol{u}_{\ell,i}) \\ \psi_{n,i} &= \sum_{\ell\in\mathcal{N}_n} a_{\ell n}\,\varphi_{\ell,i}\end{aligned}} \tag{30}$$

3.2. Functional Combine-then-Adapt (FCTA) diffusion strategy

Similarly, returning to (26) and considering the second correction first, we get the alternative diffusion strategy:

$$\boxed{\begin{aligned}\varphi_{n,i-1} &= \sum_{\ell\in\mathcal{N}_n} a_{\ell n}\,\psi_{\ell,i-1} \\ \psi_{n,i} &= \varphi_{n,i-1} \\ &\quad + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \varphi_{n,i-1}(\boldsymbol{u}_{\ell,i})]\,\kappa(\cdot,\boldsymbol{u}_{\ell,i})\end{aligned}} \tag{31}$$

3.3. Implementation

Online processing of time series data raises the question of how to process an increasing amount of observations $\boldsymbol{u}_{\ell,i}$ as new data is collected at each node. Indeed, as the KLMS algorithm (11), an undesirable characteristic of FATC and FCTA algorithms (30)-(31) is that the number of terms in $\varphi_{n,i}$ and $\psi_{n,i}$ grows linearly with the number of input data. This dramatically increases the computational burden and memory requirement. To overcome this barrier, in this paper, we shall consider as a prior that nodes share a dictionary $\mathcal{D}$ of finite size M. We leave this sharing processing, which should be based on the coherence rule, for future work.

Then, we can write $\varphi_{n,i}$ and $\psi_{n,i}$ in (30)-(31) as:

$$\begin{aligned}\varphi_{n,i} &= \boldsymbol{\beta}_{n,i}^\top\,\boldsymbol{\kappa}_{n,i} \\ \psi_{n,i} &= \boldsymbol{\alpha}_{n,i}^\top\,\boldsymbol{\kappa}_{n,i}\end{aligned} \tag{32}$$

The FATC strategy can be expressed as follows:

$$\boxed{\begin{aligned}\boldsymbol{\beta}_{n,i} &= \boldsymbol{\alpha}_{n,i-1} + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \boldsymbol{\alpha}_{n,i-1}^\top\boldsymbol{\kappa}_{\ell,i}]\,\boldsymbol{\kappa}_{\ell,i} \\ \boldsymbol{\alpha}_{n,i} &= \sum_{\ell\in\mathcal{N}_n} a_{\ell n}\,\boldsymbol{\beta}_{\ell,i}\end{aligned}} \tag{33}$$

Similarly, the FCTA strategy is given by:

$$\boxed{\begin{aligned}\boldsymbol{\beta}_{n,i-1} &= \sum_{\ell\in\mathcal{N}_n} a_{\ell n}\,\boldsymbol{\alpha}_{\ell,i-1} \\ \boldsymbol{\alpha}_{n,i} &= \boldsymbol{\beta}_{n,i-1} + \mu_n \sum_{\ell\in\mathcal{N}_n} c_{\ell n}\,[d_\ell(i) - \boldsymbol{\beta}_{n,i-1}^\top\boldsymbol{\kappa}_{\ell,i}]\,\boldsymbol{\kappa}_{\ell,i}\end{aligned}} \tag{34}$$

4. EXPERIMENTS

Consider the network with $N = 10$ nodes given in Fig. 1. The input signal at each node k and time instant i was a sequence of statistically independently vector defined as:

$$\boldsymbol{u}_{k,i} = \begin{bmatrix}u_{k,i}(1) & u_{k,i}(2)\end{bmatrix}^\top \tag{35}$$

with correlated samples satisfying $u_{k,i}(1) = 0.5\, u_{k,i}(2) + v_{k,i}$. The second entry of $\boldsymbol{u}_{k,i}$ and $v_{k,i}$ were both i.i.d. Gaussian samples with variance equal to 0.035. We considered the linear system with memory defined by

$$\begin{aligned} y_{k,i} &= \boldsymbol{a}^\top \boldsymbol{u}_{k,i} - 0.2\, y_{k,i-1} + 0.35\, y_{k,i-2} \\ y_{k,0} &= 0, y_{k,-1} = 0 \end{aligned} \tag{36}$$

where $\boldsymbol{a} = [1\ 0.5]^\top$ and the nonlinear Wiener function

$$\begin{aligned} \varphi(y_{k,i}) &= \begin{cases} \dfrac{y_{k,i}}{3[0.1 + 0.9\, y_{k,i}^2]^{1/2}} & \text{for } y_{k,i} \geq 0 \\ \dfrac{-y_{k,i}^2[1 - \exp(0.7 y_{k,i})]}{3} & \text{for } y_{k,i} < 0, \end{cases} \\ d_k(i) &= \varphi(y_{k,i}) + z_{k,i}. \end{aligned} \tag{37}$$

Noise $z_{k,i}$ was zero-mean Gaussian i.i.d. with variance $\sigma_{z_k}^2 = 0.09$. The Gaussian kernel

$$\kappa(\boldsymbol{u}_i, \boldsymbol{u}_j) = \exp(-\|\boldsymbol{u}_i - \boldsymbol{u}_j\|_2^2 / 2\xi^2) \tag{38}$$

was considered with $\xi = 0.15$, and the step-sizes μ_k were all set to 0.05. The entries $c_{\ell n}$ of $\boldsymbol{C}$ were set to $|\mathcal{N}_n|^{-1}$ for all $n \in \mathcal{N}_\ell$. The combination matrix $\boldsymbol{A}$ simply averaged the estimates from the neighbors, namely, $a_{\ell n} = |\mathcal{N}_n|^{-1}$ for $\ell \in \mathcal{N}_n$. The coherence rule with threshold $\delta_0 = 0.3$ was used to set the same dictionary $\mathcal{D}$ for all nodes beforehand. This led to a 17-elements dictionary that remains unchanged throughout the experiment.

The MSE learning curves in Fig. 2 were obtained by averaging over 200 Monte Carlo runs. They show that the performance of the cooperative FCTA and FATC strategies were almost identical. For comparison, we implemented the non-cooperative KLMS strategy (14) at each node with the same dictionary. It can be observed that it performed poorly compared to FCTA and FATC strategies.

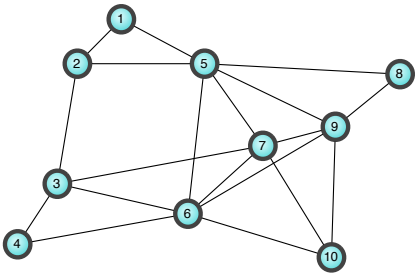

Fig. 1. Network topology.

5. CONCLUSION

In this paper, we derived functional counterparts of the adapt-then-combine and combine-then-adapt diffusion strategies. These algorithms allow to perform online learning of nonlinear fitting models. Their efficiency was illustrated with simulation results. In future works, we will analyze their convergence in the mean and mean-square-error sense. We will also address the problem of dictionary learning at each node to circumvent the drawbacks of the KLMS.

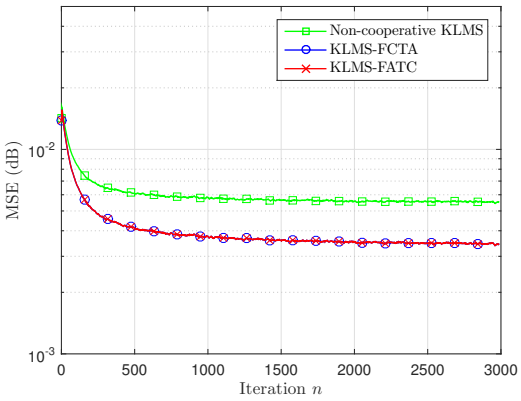

Fig. 2. MSE learning curves

REFERENCES

[1] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Nonparametric decentralized detection using kernel methods," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4053–4066, 2005.

[2] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, 2007.

[3] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, 2009.

[4] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE ICASSP*, 2007, vol. 3, pp. 917–920.

[5] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proc. IPSN'04*, 2004, pp. 1–10.

[6] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Regression in sensor networks: Training distributively with alternating projections," in *Proc. SPIE Conf. Advanced Signal Processing Algorithm, Architectures, and Implementations XV*, 2005.

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

[8] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–5.

[9] P. Honeine, C. Richard, J.-C. M. Bermudez, H. Snoussi, M. Essoloh, and F. Vincent, "Functional estimation in Hilbert space for distributed learning in wireless sensor networks," in *Proc. IEEE ICASSP*, 2009, pp. 2861–2864.

[10] Y. P. Bergamo and C. G. Lopes, "Scalar field estimation using adaptive networks," in *Proc. IEEE ICASSP*, 2012, pp. 3565–3568.

[11] C. Richard, J.-C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, 2009.

[12] W. D. Parreira, J.-C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the Gaussian kernel-least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2208–2222, 2012.

[13] W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2765–2777, 2014.