2014 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2014, REIMS, FRANCE

DIFFUSION STRATEGIES FOR IN-NETWORK PRINCIPAL COMPONENT ANALYSIS

Nisrine Ghadban[1,2]*, Paul Honeine*[1]*, Farah Mourad-Chehade*[1]*, Clovis Francis*[2]*, Joumana Farah*[3]

[1] Institut Charles Delaunay, Université de technologie de Troyes, CNRS, France
[2] Faculté de Génie, Université Libanaise, Lebanon
[3] Telecommunications department, Faculty of Engineering, Holy-Spirit University of Kaslik, Lebanon

ABSTRACT

This paper deals with the principal component analysis in networks, where it is improper to compute the sample covariance matrix. To this end, we derive several in-network strategies to estimate the principal axes, including noncooperative and cooperative (diffusion-based) strategies. The performance of the proposed strategies is illustrated on diverse applications, including image processing and dimensionality reduction of time series in wireless sensor networks.

***Index Terms*—** Principal component analysis, network, adaptive learning, distributed processing

1. INTRODUCTION

The principal component analysis (PCA) is one of the most popular unsupervised learning techniques, with applications in statistical analysis, data compression, and feature extraction [1]. It defines a set of principal axes that transforms a number of correlated variables into uncorrelated ones, the so-called *principal components*. The most relevant principal axes retain the largest variance of the data. The PCA technique has been investigated in many applications involving multivariate analysis, e.g., data validation and fault detection [2] and quality control [3]. It is very useful in a sensor network, where it helps in extracting features from noisy samples [4], compressing and denoising times series measurements [5, 6], as well as for intrusion detection [7] and anomaly detection [8].

The conventional PCA requires the eigen-decomposition of the sample covariance matrix. Such calculation is inappropriate for networks since sending all data to a fusion center (FC) is not scalable. Moreover, the computational complexity is cubic with the size of the dataset. Several attempts have been made to alleviate this problem, such as in [9, 10]; however, these techniques remain computationally inefficient. More recently, in [11], only the principal components have been transmitted to the FC, instead of the whole time series. In [12], the power iteration method is investigated to estimate the most relevant principal axis. This method requires the computation of the sample covariance matrix, making it inappropriate for in-network processing.

In this paper, we propose to estimate the principal axis without computing the sample covariance matrix. To this end, we investigate two principles, the Oja's neural-based rule [13], which has been recently investigated in [14] for nonlinear PCA with kernel-based machines, and the information theoretical criterion derived in [15]. Within a network setting, we derive several in-network algorithms, including noncooperative and cooperative strategies. In the latter, sensors cooperate by information diffusion in order to estimate the principal axis. Two diffusion strategies are derived, the so-called combine-then-adapt and adapt-then-combine strategies, which have been recently investigated in linear adaptive filtering literature by A. Sayed *et al.* in [16]. To the best of our knowledge, the present study is the first work that investigates these adaptation strategies for unsupervised learning, as given here with the PCA.

The rest of this paper is organized as follows. Section 2 presents the network topology and Section 3 describes the proposed strategies for estimating the first principal axis. The strategies are extended in Section 4 for multiple axes extraction. Section 5 provides experimentation results and discussions.

2. NETWORK MODEL

We consider a connected network of N nodes (agents), where any two nodes are either directly linked if they are neighbors, or through other intermediate nodes. We differentiate between a centralized network where the nodes are connected to a FC, and a decentralized network where the nodes are connected in a noncooperative way by a routing system or in a cooperative way where each node communicates with its neighbors. Figure 1 illustrates these network topologies. Let $\mathcal{V}_k$ denote the indices set of the neighboring nodes to node k, with $k = 1, ..., N$, *i.e.,* the nodes that are directly connected to it. We consider that the node k is adjacent to itself, that is to say $k \in \mathcal{V}_k$. The notation $\mathcal{V}_k$ would be useful in the following for cooperative networks.

This work is supported by the Région Champagne-Ardenne (grant "WiDiD") and the Lebanese University.

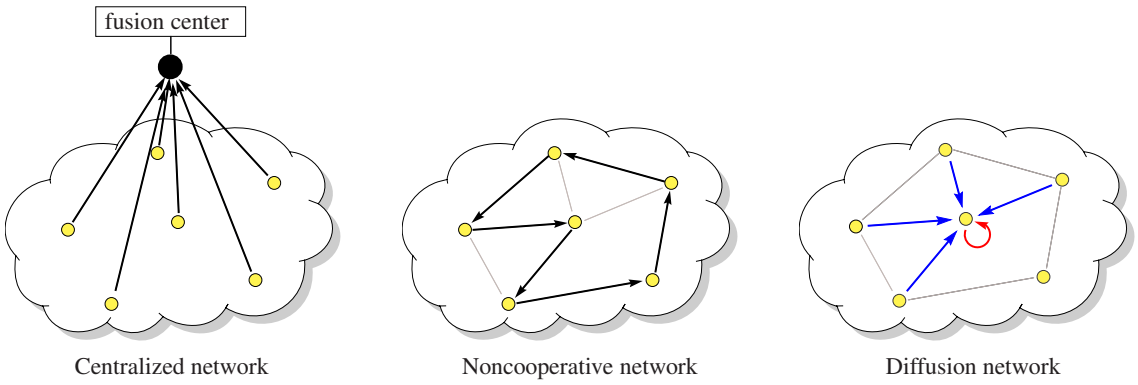

Fig. 1. Illustration of information processing within the proposed networks.

In this paper, the nodes estimate the same principal axis $\boldsymbol{w}$ based on some measured information corresponding to a common phenomenon. Let $\boldsymbol{x}_k$ be the $(p \times 1)$ vector collected by the node k, for $k = 1 \cdots N$, p being the measurements dimension, and let $\mathbb{X} \subset \mathbb{R}^p$ be the space of these collected data (assumed to be zero-mean), with the conventional inner product $\boldsymbol{x}_k^\top \boldsymbol{x}_l$ for any $\boldsymbol{x}_k, \boldsymbol{x}_l \in \mathbb{X}$. We denote by the scalar value $y_{\boldsymbol{w},k} = \boldsymbol{w}^\top \boldsymbol{x}_k$ the inner product associated with the orthogonal projection of any $\boldsymbol{x}_k \in \mathbb{X}$ onto the vector $\boldsymbol{w} \in \mathbb{X}$ and by $y_{\boldsymbol{w}}$ the scalar random variable taking the values $y_{\boldsymbol{w},k}$, with $k = 1, ..., N$. The ultimate goal would be to compute $\boldsymbol{w}$, according to the networks topologies, to keep afterwards only $y_{\boldsymbol{w},k}$ and $\boldsymbol{w}$, $k = 1, ..., N$, allowing us to rebuild optimally the data $\boldsymbol{x}_k$ when needed.

3. IN-NETWORK PCA

In this section, we first expose the conventional strategy for PCA where the network is assumed to be centralized. We derive then our new strategies to extract the first principal axis.

3.1. Centralized strategy

Here, the nodes are assumed to be connected to a FC, to which they send their collected data without any local processing. The FC extracts the first principal axis that maximizes the variance of the projected data, namely $\max_{\boldsymbol{w}} \mathbb{E}(y_{\boldsymbol{w}}^2)$, where $\mathbb{E}(\cdot)$ is the expectation over the density of input data. By taking the empirical estimation of the latter with respect to the available samples, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, we get $\max_{\boldsymbol{w}} \boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w}$, where $\boldsymbol{C} = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{x}_k \boldsymbol{x}_k^\top$ is the covariance of the measured data. It is well known that the problem takes the form $\boldsymbol{C}\boldsymbol{w} = \lambda \boldsymbol{w}$. This is an eigen-decomposition problem where $\boldsymbol{w}$ is the eigenvector associated with the eigenvalue λ of $\boldsymbol{C}$. One can easily see that the latter is the corresponding variance, since

$$\boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w} = \boldsymbol{w}^\top \lambda \boldsymbol{w} = \lambda \boldsymbol{w}^\top \boldsymbol{w} = \lambda.$$

Therefore, to maximize the variance of the projected data, the eigenvector associated with the largest eigenvalue must be considered. This eigenvector, denoted by $\boldsymbol{w}_*$, corresponds to the first principal axis and is determined by the FC using the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. Since the covariance matrix $\boldsymbol{C}$ is a p-by-p matrix, the computational complexity[1] of such operation is $\mathcal{O}(p^3)$. In the following, we propose strategies that allow reducing the computational complexity by avoiding the calculation of the covariance matrix.

3.2. Noncooperative strategy

Instead of explicitly investigating the covariance matrix, we propose to adaptively learn the first principal axis, by considering an in-network scheme. To this end, instead of directly maximizing the overall projected variance $\mathbb{E}(y_{\boldsymbol{w}}^2)$, we consider an "instantaneous" estimation at each node. According to a routing process, each node k receives an estimate $\boldsymbol{w}_{t-1}$ from another node, and adjusts it using its own data $\boldsymbol{x}_k$ by maximizing $y_{\boldsymbol{w},k}^2$. In the following, two different cost functions are explored.

- In the first alternative, the "instantaneous" quadratic reconstruction error is minimized, namely

$$J_k(\boldsymbol{w}) = \tfrac{1}{4} \|\boldsymbol{x}_k - y_{\boldsymbol{w},k} \boldsymbol{w}\|^2. \quad (1)$$

The gradient of this cost function with respect to $\boldsymbol{w}$ is

$$\nabla_{\boldsymbol{w}} J_k(\boldsymbol{w}) = y_{\boldsymbol{w},k} (y_{\boldsymbol{w},k} \boldsymbol{w} - \boldsymbol{x}_k), \quad (2)$$

which leads to the gradient descent formulation

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \eta_t \left(\boldsymbol{x}_k \, y_{\boldsymbol{w}_{t-1},k} - y_{\boldsymbol{w}_{t-1},k}^2 \, \boldsymbol{w}_{t-1} \right), \quad (3)$$

where η_t is the learning rate. This update rule is essentially Oja's rule [13], which is a single-neuron special case of the generalized Hebbian learning, applied here within the in-network setting described above.

[1] One may also include the communication complexity, which is $\mathcal{O}(Np)$ over a distance $\mathcal{O}(1)$.

- Drawing inspiration from the information theoretical framework studied in [15, 17], we minimize the following cost function associated with node k:

$$J_k(\boldsymbol{w}) = \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \tfrac{1}{2}\ln(y^2_{\boldsymbol{w},k}). \tag{4}$$

whose gradient with respect to $\boldsymbol{w}$ is

$$\nabla_{\boldsymbol{w}} J_k(\boldsymbol{w}) = \boldsymbol{w} - \frac{\boldsymbol{x}_k}{y_{\boldsymbol{w},k}}. \tag{5}$$

By applying the gradient descent technique on this cost function, this leads to the following update rule:

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \eta_t \left(\frac{\boldsymbol{x}_k}{y_{\boldsymbol{w}_{t-1},k}} - \boldsymbol{w}_{t-1}\right). \tag{6}$$

The update rules (3) and (6) are intimately connected, since the latter can be derived from the former by using the normalized learning rate $\eta_t / y^2_{\boldsymbol{w}_{t-1},k}$. For this reason, we will provide throughout this paper the general expressions, while we study the effect of such normalization in experiments.

These learning rules converge to the same equilibrium state, which is the first principal axis $\boldsymbol{w}_*$. In fact, when $\boldsymbol{w}_t$ converges to some state $\boldsymbol{w}$, we have $\boldsymbol{x}_k y_{\boldsymbol{w},k} = y^2_{\boldsymbol{w},k}\boldsymbol{w}$ (from (3)) or $\boldsymbol{x}_k / y_{\boldsymbol{w},k} = \boldsymbol{w}$ (from (6)), namely $\boldsymbol{x}_k \boldsymbol{x}_k^\top \boldsymbol{w} = y^2_{\boldsymbol{w},k}\boldsymbol{w}$. Averaging the latter expression over the whole data, we get the well-known eigen-decomposition problem of the covariance matrix $\boldsymbol{C}\boldsymbol{w} = \mathbb{E}(y^2_{\boldsymbol{w}})\,\boldsymbol{w}$, where the eigenvalue is the squared output $y_{\boldsymbol{w},k}$ to be maximized. Therefore, the two update rules (3) and (6) converge to the eigenvector associated with the largest eigenvalue of the covariance matrix.

3.3. Cooperative strategies

Now, each node k has access to the information of its neighborhood $\mathcal{V}_k$. The optimization problem can be presented as a minimization of the cost function $\sum_{k=1}^N J_k(\boldsymbol{w})$, where $J_k(\boldsymbol{w})$ is the cost function at node k, for instance either (1) or (4). Since each node k communicates only with its neighbors, we introduce the nonnegative coefficients c_{kl} that relate the node k to its neighbors. They satisfy the following conditions:

$$c_{kl} \geq 0, \qquad \sum_{l\in\mathcal{V}_k} c_{kl} = 1, \quad \text{and} \quad c_{kl} = 0 \text{ if } l \notin \mathcal{V}_k. \tag{7}$$

It is worth noting that if $l \notin \mathcal{V}_k$, $k \notin \mathcal{V}_l$ and thus c_{lk} is also null. These coefficients allow defining, for each node l, a local cost that consists of a weighted combination of the individual costs of the neighbors of node l (including l itself):

$$J_l^{\text{loc}}(\boldsymbol{w}) = \sum_{k\in\mathcal{V}_l} c_{kl}\, J_k(\boldsymbol{w}) = \sum_{k=1}^N c_{kl}\, J_k(\boldsymbol{w}). \tag{8}$$

It turns out that the cumulative sum of these local cost functions is equal to the global cost function since we have

$$\sum_{k=1}^N J_k(\boldsymbol{w}) = \sum_{k=1}^N \Big(\sum_{l=1}^N c_{kl}\Big) J_k(\boldsymbol{w}) = \sum_{l=1}^N \sum_{k=1}^N c_{kl} J_k(\boldsymbol{w}) = \sum_{l=1}^N J_l^{\text{loc}}(\boldsymbol{w}).$$

For node k, the global cost function can be decomposed as

$$\sum_{l=1}^N J_l(\boldsymbol{w}) = J_k^{\text{loc}}(\boldsymbol{w}) + \sum_{\substack{l=1\\ l\neq k}}^N J_l^{\text{loc}}(\boldsymbol{w}). \tag{9}$$

While the first term in the right-hand-side is known at the node under scrutiny, the second one should be estimated using information from the neighbors of node k; thus, the above summation is restricted to its neighborhood. Moreover, we relax it by constraining the norm between the estimated vector $\boldsymbol{w}$ and the optimal (global) principal axis $\boldsymbol{w}_*$ within the neighborhood of node k, with the following regularization:

$$\sum_{\substack{l=1\\ l\neq k}}^N J_l^{\text{loc}}(\boldsymbol{w}) \approx \sum_{l\in\mathcal{V}_k\setminus\{k\}} b_{lk}\, \|\boldsymbol{w} - \boldsymbol{w}_*\|^2,$$

where the parameters b_{lk} control the tradeoff between the accuracy and the smoothness of the solution. Such regularization is also motivated by investigating the second-order Taylor expansion of $J_l^{\text{loc}}(\boldsymbol{w})$, as described in [18]. Note that we use $\boldsymbol{w}_*$ knowing that we do not have access to its value. We will show in the following how to overcome this problem.

By injecting this approximation into (9), and using (8), the minimization of the latter is equivalent to minimizing

$$J_k^{\text{glob}}(\boldsymbol{w}) = \sum_{l\in\mathcal{V}_k} c_{lk} J_l(\boldsymbol{w}) + \sum_{l\in\mathcal{V}_k\setminus\{k\}} b_{lk}\|\boldsymbol{w} - \boldsymbol{w}_*\|^2. \tag{10}$$

In order to minimize the above cost function, the node k applies the gradient descent on $J_k^{\text{glob}}(\boldsymbol{w})$ with:

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} - \eta_{k,t}\, \nabla_{\boldsymbol{w}} J_k^{\text{glob}}(\boldsymbol{w}_{k,t-1}). \tag{11}$$

Here, $\eta_{k,t}$ is the learning rate for node k at iteration t. Replacing $J_k^{\text{glob}}(\boldsymbol{w})$ by its expression in (10), we get:

$$\begin{aligned}\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} &- \eta_{k,t} \sum_{l\in\mathcal{V}_k} c_{lk} \nabla_{\boldsymbol{w}} J_l(\boldsymbol{w}_{t-1}) \\ &+ \eta_{k,t} \sum_{l\in\mathcal{V}_k\setminus\{k\}} b_{lk}(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}).\end{aligned}$$

In this expression, $\nabla_{\boldsymbol{w}} J_l(\boldsymbol{w}_{t-1})$ is the gradient of the PCA-based cost function, presented earlier in (2) and (5). In the former case, we get

$$\begin{aligned}\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} &+ \eta_{k,t} \sum_{l\in\mathcal{V}_k} c_{lk}\big(\boldsymbol{x}_l\, y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l}\, \boldsymbol{w}_{l,t-1}\big) \\ &+ \eta_{k,t} \sum_{l\in\mathcal{V}_k\setminus\{k\}} b_{lk}(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}),\end{aligned} \tag{12}$$

while the latter corresponds to substituting the above learning rate by $\eta_{k,t} / y^2_{\boldsymbol{w}_{t-1},k}$. Without loss of generality, we will consider the update rule (12) in the following. This update rule

from $\boldsymbol{w}_{k,t-1}$ to $\boldsymbol{w}_{k,t}$ involves adding two correction terms: the first one operates in the maximum variance direction and the second one associates neighborhood regularization. By decomposing this expression into two successive steps, , we get two possible strategies, depending on the order of adding the correction terms, and that differ essentially in the approximation of the unknown $\boldsymbol{w}_*$ in (12), as shown next.

Adapt-then-combine strategy

We express the update rule (12) as follows:

$$\boldsymbol{\phi}_{k,t} = \boldsymbol{w}_{k,t-1} + \eta_{k,t} \sum_{l\in\mathcal{V}_k} c_{lk}\Big(\boldsymbol{x}_l y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l}\boldsymbol{w}_{l,t-1}\Big),$$

$$\boldsymbol{w}_{k,t} = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \sum_{l\in\mathcal{V}_k\backslash\{k\}} b_{lk}(\boldsymbol{w}_* - \boldsymbol{\phi}_{k,t}).$$

The first step uses local gradient vectors from the neighborhood of the node k in order to update $\boldsymbol{w}_{k,t-1}$ to the intermediate estimate $\boldsymbol{\phi}_{k,t}$. By incorporating information from the neighbors, this intermediate estimate is generally a better estimate for $\boldsymbol{w}_*$ than $\boldsymbol{w}_{k,t-1}$, as used in the second step, which becomes

$$\boldsymbol{w}_{k,t} = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \sum_{l\in\mathcal{V}_k\backslash\{k\}} b_{lk}\,(\boldsymbol{\phi}_{l,t} - \boldsymbol{\phi}_{k,t}),$$

or, equivalently $\boldsymbol{w}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\,\boldsymbol{\phi}_{l,t}$ where we have used the following nonnegative weighting coefficients:

$$a_{kl} = \begin{cases} 1 - \eta_{k,t}\sum_{i\in\mathcal{V}_k\backslash\{k\}} b_{ik}, & \text{if } l = k; \\ \eta_{k,t}\, b_{lk}, & \text{if } l \in \mathcal{V}_k\backslash\{k\}\ ; \\ 0, & \text{otherwise.} \end{cases}$$

Combine-then-adapt strategy

In this strategy, we express the update rule (12) as follows:

$$\boldsymbol{\phi}_{k,t} = \boldsymbol{w}_{k,t-1} + \eta_{k,t} \sum_{l\in\mathcal{V}_k\backslash\{k\}} b_{lk}(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1})$$

$$\boldsymbol{w}_{k,t} = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \sum_{l\in\mathcal{V}_k} c_{lk}\Big(\boldsymbol{x}_l\, y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l}\,\boldsymbol{\phi}_{l,t}\Big).$$

By approximating $\boldsymbol{w}_*$ by $\boldsymbol{w}_{l,t-1}$ and introducing the same coefficients a_{kl}, we obtain the update rule

$$\boldsymbol{\phi}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\,\boldsymbol{w}_{l,t-1},$$

$$\boldsymbol{w}_{k,t} = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \sum_{l\in\mathcal{V}_k} c_{lk}\Big(\boldsymbol{x}_l\, y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l}\,\boldsymbol{\phi}_{l,t}\Big).$$

3.4. Connections to the consensus strategies

The consensus-type implementation is a class of distributed strategies that takes the following form:

$$\boldsymbol{w}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\boldsymbol{w}_{k,t-1} + \eta_{k,t}\Big(\boldsymbol{x}_k\, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k}\boldsymbol{w}_{k,t-1}\Big).$$

We notice that the consensus strategy has the same computational complexity as for the adapt-then-combine and the combine-then-adapt diffusion strategies. However, these diffusion strategies outperform the consensus implementation. Indeed, we can express the adapt-then-combine and the combine-then-adapt diffusion strategies respectively by:

$$\boldsymbol{w}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\Big(\boldsymbol{w}_{k,t-1} + \eta_{k,t}\,\big(\boldsymbol{x}_k\, y_{\boldsymbol{w},k} - y^2_{\boldsymbol{w},k}\boldsymbol{w}_{k,t-1}\big)\Big),$$

$$\boldsymbol{w}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\boldsymbol{w}_{k,t-1} + \eta_{k,t}\,\Big(\boldsymbol{x}_k\, y_{\boldsymbol{w},k} - y^2_{\boldsymbol{w},k}\sum_{l\in\mathcal{V}_k} a_{kl}\boldsymbol{w}_{k,t-1}\Big).$$

Considering for instance the combine-then-adapt strategy, we note that the combination coefficients a_{kl} appear in the correction term, while the consensus uses only $\boldsymbol{w}_{k,t-1}$ to correct the error, which is less effective, as will be shown in the experimental results.

4. MULTIPLE PRINCIPAL AXES

To extend the derivation to multiple principal axes, we denote by $\boldsymbol{W}_t = [\boldsymbol{w}_{1,t} \quad \boldsymbol{w}_{2,t} \quad \cdots \quad \boldsymbol{w}_{r,t}]^\top$, the r-by-p matrix of the first r principal axes estimated at iteration t, listed in descending order of their eigenvalues. Let $\boldsymbol{y}_{\boldsymbol{W},k} = [y_{\boldsymbol{w}_1,k} \quad y_{\boldsymbol{w}_2,k} \quad \cdots \quad y_{\boldsymbol{w}_r,k}]^\top$, where $y_{\boldsymbol{w}_j,k} = \boldsymbol{w}_{j,t}^\top\boldsymbol{x}_k$. Next, we combine the update rules given in Sections 3.2 and 3.3 with the Gram-Schmidt orthogonalization process.[2]

In the noncooperative strategy, we get the update rule of the j-th principal axis as following:

$$\boldsymbol{w}_{j,t} = \boldsymbol{w}_{j,t-1} + \eta_t\,\Big(\boldsymbol{x}_k\, y_{\boldsymbol{w}_j,k} - y_{\boldsymbol{w}_j,k}\sum_{l=1}^{j} y_{\boldsymbol{w}_j,l}\boldsymbol{w}_{l,t-1}\Big), \tag{13}$$

which is essentially similar to a Sanger's generalized Hebbian algorithm [19]. In matrix form, we obtain

$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} + \eta_t\,\big(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\,\boldsymbol{x}_k^\top - \mathrm{LT}(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\,\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}^\top)\,\boldsymbol{W}_{t-1}\big), \tag{14}$$

where $\mathrm{LT}(\cdot)$ makes its argument lower triangular by setting to zero the entries above its diagonal. Note that the learning rate does not needs to be the same for all the principal axes.

In the cooperative strategies, we denote by $\boldsymbol{\Phi}_{k,t} = [\boldsymbol{\phi}_{1,k,t} \quad \boldsymbol{\phi}_{2,k,t} \quad \cdots \quad \boldsymbol{\phi}_{r,k,t}]^\top$ the r-by-p matrix of the r intermediate estimates. Written in a matrix form, the update rule associated with the adapt-then-combine strategy becomes

$$\boldsymbol{\Phi}_{k,t} = \boldsymbol{W}_{k,t-1} + \eta_{k,t}\Big(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\boldsymbol{x}_k^\top - \mathrm{LT}(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}^\top)\boldsymbol{W}_{k,t-1}\Big),$$

$$\boldsymbol{W}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\,\boldsymbol{\Phi}_{l,t}.$$

For the combine-then-adapt strategy, we get the steps

$$\boldsymbol{\Phi}_{k,t} = \sum_{l\in\mathcal{V}_k} a_{kl}\,\boldsymbol{W}_{l,t-1},$$

$$\boldsymbol{W}_{k,kt} = \boldsymbol{\Phi}_{k,t} + \eta_{k,t}\Big(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\boldsymbol{x}_k^\top - \mathrm{LT}(\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}\boldsymbol{y}_{\boldsymbol{W}_{t-1},k}^\top)\boldsymbol{\Phi}_{k,t}\Big).$$

[2]Several orthogonalization techniques can be investigated within the proposed framework, including the Gram-Schmidt process, the deflation, and the symmetric orthogonalization. in order to extract multiple principal axes.

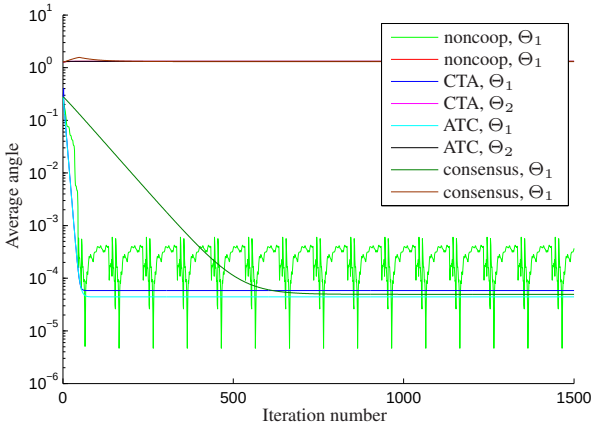

Fig. 2. Convergence analysis in the WSN settings.

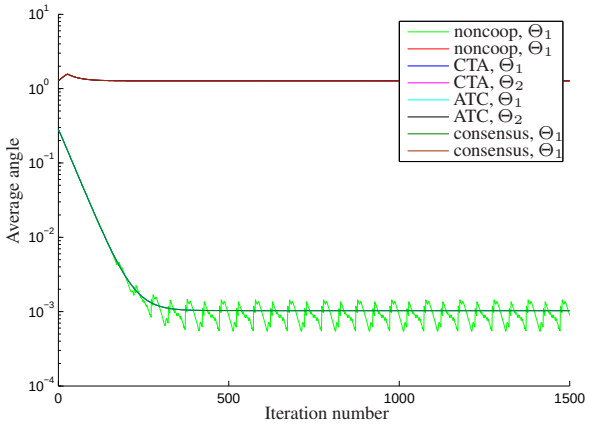

Fig. 3. Convergence analysis of the strategies based on information theory, in the WSN settings.

5. EXPERIMENTATIONS

In this section, we illustrate the performance of the proposed approach. In order to provide a fair comparative study, we use the same initial random estimate for all strategies. The performance is measured in terms of the angle between the principal axis $\boldsymbol{w}_*$, obtained from the centralized strategy with the eigen-decomposition of the covariance matrix, and the estimate $\boldsymbol{w}_{*,l}$ at node l, namely $\arccos \frac{\boldsymbol{w}_{*,l}^\top \boldsymbol{w}_*}{\|\boldsymbol{w}_{*,l}\| \|\boldsymbol{w}_*\|}$.

5.1. Time series measurements in a WSN

In this section, we consider the problem of tracking a gas spread using a wireless sensor network (WSN) [20]. Here, $x_{k,t}$ denotes the gas measurement of the k-th sensor at time t. The position of a sensor k is denoted by $\boldsymbol{z}_k \in \mathbb{Z}$. The region under scrutiny $\mathbb{Z} = [-0.5, 0.5] \times [-0.5, 0.5]$ is a two-dimensional unit-area. Our goal is to reduce the order of the time series of the measurements. The gas diffusion within this region is governed by the following differential equation:

$$\frac{\partial G(\boldsymbol{z}, \theta)}{\partial \theta} - c\, \nabla^2_{\boldsymbol{z}} G(\boldsymbol{z}, \theta) = Q(\boldsymbol{z}, \theta),$$

where $G(\boldsymbol{z}, \theta)$ is the density of gas depending on the position $\boldsymbol{z}$ and time θ, $\nabla^2_{\boldsymbol{z}}$ is the Laplace operator, c the conductivity of the medium, and $Q(\boldsymbol{z}, \theta)$ corresponds to the added quantity of gas. A gas source placed at the origin is activated from $\theta = 1$ to $\theta = 15$. We use $N = 100$ sensors deployed uniformly in the region $\mathbb{Z}$, each acquiring a time series of 15 measurements, between $\theta = 1$ and $\theta = 15$.

We consider a predetermined range of communication in the WSN. In this case, two nodes are connected when their distance is less than 0.38. For the stepsize parameters, we consider the stepsize $\eta_1 = 0.0025$ for the first principal axis and $\eta_2 = 0.0005$ for the second principal axis. We choose another stepsize for the second axis for a better convergence.

As for the information theory, we consider $\eta_1 = 0.025$ and $\eta_2 = 0.05$ for both cooperative and noncooperative strategies.

Figure 2 and Figure 3 show the convergence of each strategy for the general expressions and for the information theory respectively, with the angles Θ_1 and Θ_2, for the first and the second axes respectively, averaged over the N nodes. The confronted strategies are the noncooperative strategy, the combine-then-adapt (CTA) and adapt-then-combine (ATC) diffusion strategies, and the consensus implementation. For the diffusion strategies, we assumed that $c_{kk} = 1$, and $c_{lk} = 0$ for each $k \neq l$, with $l, k = 1, \cdots, 100$. Results are shown in terms of the angle averaged over all the nodes. These learning curves show that the noncooperative strategy is outperformed by any diffusion strategy, independently of the combination rule. The analysis of the diffusion strategies shows that the adapt-then-combine strategy performs slightly better that the combine-then-adapt strategy. The overall efficiency of the diffusion strategies is shown in terms of stability.

5.2. Image processing application

We consider an image processing application, with the handwritten digit "1" given in 90 images of 28-by-28 pixels, each treated as a 784-dimensional vector. This time, the connectivity between nodes is tested for $s = 2240$. For the stepsize parameters, we consider the stepsize $\eta_1 = 5.10^{-8}$ for the first principal axis and $\eta_2 = 1.10^{-7}$ for the second principal axis. The same stepsize values are taken for the cooperative strategies. As for the information theory, we take $\eta_1 = 0.05$ and $\eta_2 = 0.05$ for both cooperative and noncooperative strategies. Figures 4 and 5 show the convergence of each strategy for the general form and for the information theory respectively, with the angles averaged over the N nodes. Again, the noncooperative strategy is outperformed by all diffusion strategies, independently of the combination rule.

6. CONCLUSION

In this paper, we studied the issue of estimating the principal axes from PCA in networks. While taking into account the constraints imposed in WSNs and image processing applications, we proposed several strategies including noncooperative and diffusion strategies. Experimental results showed the relevance of these strategies. As a future work, we will also include the use of the spatial information within the study.

7. REFERENCES

[1] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[2] B. M. Wise, N. L. Ricker, and D. J. Veltkamp, "Upset and sensor failure detection in multivariable processes," AlChE Meeting, San Francisco, 1989.

[3] J. MacGregor, "Multivariate statistical methods for monitoring large data sets from chemical processes," AlChE Meeting, San Francisco, 1989.

[4] N. Chitradevi, K. Baskaran, V. Palanisamy, and D. Aswini, "Designing an efficient PCA based data model for wireless sensor networks," in *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief*, ser. ACWR '11. New York, NY, USA: ACM, 2011, pp. 147–154.

[5] F. Chen, F. Wen, and H. Jia, "Algorithm of data compression based on multiple principal component analysis over the wsn," in *Wireless Communications Networking and Mobile Computing (WiCOM), 6th International Conference on*, Sept 2010, pp. 1–4.

[6] A. Rooshenas, H. Rabiee, A. Movaghar, and M. Naderi, "Reducing the data transmission in wireless sensor networks using the principal component analysis," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Sixth International Conference on*, Dec 2010, pp. 133–138.

[7] M. Ahmadi Livani and M. Abadi, "A PCA-based distributed approach for intrusion detection in wireless sensor networks," in *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*, Feb 2011, pp. 55–60.

[8] L. Huang, M. I. Jordan, A. Joseph, M. Garofalakis, and N. Taft, "In-network pca and anomaly detection," in *NIPS*. MIT Press, 2006, pp. 617–624.

[9] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, pp. 111–129, 1978.

[10] P. M. Hall, A. D. Marshall, and R. R. Martin, "Merging and splitting eigenspace models." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 1042–1049, 2000.

[11] H. Kargupta, W. Huang, K. Sivakumar, B.-H. Park, and S. Wang, "Collective principal component analysis from distributed, heterogeneous data," in *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*. London, UK, UK: Springer-Verlag, 2000, pp. 452–457.

[12] Y.-A. Le Borgne, S. Raybaud, and G. Bontempi, "Distributed principal component analysis for wireless sensor networks," *Sensors*, vol. 8, no. 8, pp. 4821–4850, 2008.

[13] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, no. 3, pp. 267–273, November 1982.

[14] P. Honeine, "Online kernel principal component analysis: a reduced-order model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1814–1826, September 2012.

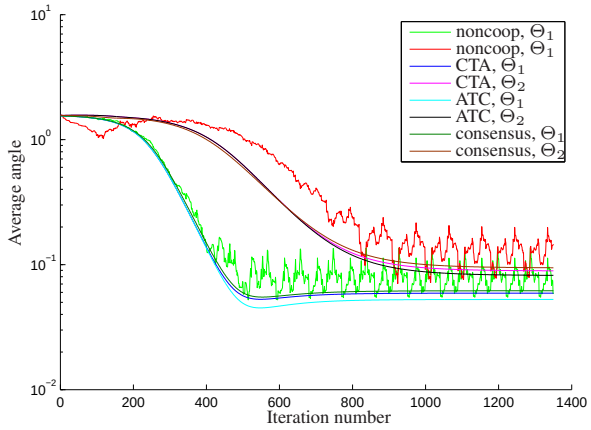

Fig. 4. Convergence analysis for the image dataset.

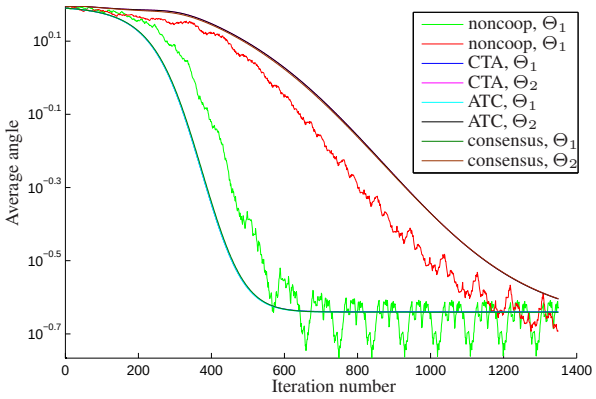

Fig. 5. Convergence analysis of the strategies based on information theory, for the image dataset.

[15] M. D. Plumbley, "Lyapunov functions for convergence of principal component algorithms," *Neural Networks*, vol. 8, pp. 11–23, 1995.

[16] A. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 155–171, May 2013.

[17] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion." *IEEE Transactions on Signal Processing*, vol. 46, no. 7, pp. 1967–1979, 1998.

[18] J. Chen and A. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 4289–4305, Aug 2012.

[19] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network." *Neural Networks*, vol. 2, pp. 459–473, 1989.

[20] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.