Gossip algorithms for principal component analysis in networks

Nisrine Ghadban, Paul Honeine, Farah Mourad-Chehade, Joumana Farah, Clovis Francis

GOSSIP ALGORITHMS FOR PRINCIPAL COMPONENT ANALYSIS IN NETWORKS

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

[1] Institut Charles Delaunay (CNRS), Université de technologie de Troyes, France
[2] Faculté de Génie, Université Libanaise, Lebanon

ABSTRACT

This paper deals with the issues of the dimensionality reduction and the extraction of the structure of data using principal component analysis for the multivariable data in large-scale networks. In order to overcome the high computational complexity of this technique, we derive several in-network strategies to estimate the principal axes without the need for computing the sample covariance matrix. To this aim, we propose to combine Oja's iterative rule with average gossiping algorithms. Gossiping is used as a solution for communication between asynchronous nodes. The performance of the proposed approach is illustrated on time series acquisition in wireless sensor networks.

Index Terms— Gossip averaging, principal component analysis, in-network processing, adaptive learning, distributed processing

1. INTRODUCTION

Extracting information from multivariable data is a difficult task. For a better understanding of data compression and denoising, one may identify relevant patterns. Among many existing techniques, Principal Component Analysis (PCA) [1,2] is probably the most widely used. It is also called the discrete Karhunen-Loève transform in the signal processing literature and the Hotelling transform in multivariate analysis. While the scope of the PCA is very wide, it provides a powerful tool in the context of sensor networks. It is investigated to extract features from noisy samples [3], compress and denoise time series measurements [4], as well as for the detection of intrusion [5] or anomaly [6].

The PCA consists in determining a subspace that retains the largest variance of the data. This subspace is spanned by the most relevant principal axes. The conventional PCA algorithm requires the eigen-decomposition of the sample covariance matrix. In network context, sending all time series to a fusion center is not scalable. Moreover, the computational complexity is cubic with the size of the dataset. Several techniques have been proposed to overcome these drawbacks, such as [7, 8], however these techniques still present a high computational cost. In [9], the authors used the power iteration method to estimate the most relevant principal axis. This method requires the computation of the sample covariance matrix, which is inappropriate for networks.

On the other hand, synchronization in a decentralized network is a very crucial problem, more specifically for the above algorithms. Indeed, nodes communicate by exchanging each other's information. Therefore time synchronization is required to insure the same time scale for the nodes' local clocks. It demands oscillators to emit signals at the same frequency. Designing synchronization algorithms has been investigated in many research works [10, 11]. However, these techniques necessitate a significant consumption of resources in the network which are very often limited. To overcome this problem, distributed and asynchronous algorithms have been investigated. In such algorithms, each node exchanges information with only one neighbor in a time slot. The gossip algorithms, also called epidemic algorithms, are based on an asynchronous information exchange [12, 13]. They allow a distribution of the computational burden since at each time instance, a node communicates with only one randomly chosen neighbor. Gossip algorithms are simple, scalable, and robust to node failures, message loss, and transient network disruptions.

In this paper, we propose to estimate the principal axis using the gossip approach, without the need of computing the sample covariance matrix. To this end, we revisit Oja's rule, initially described in [14] and studied more recently in [15] for nonlinear PCA with kernel-based machines. Within the network settings, we combine this rule with the averaging gossip where sensors cooperate between each other to estimate the principal axis. Communication between nodes is governed by the gossip concept in order to relax constraints on the reception synchronism. The relevance of the proposed algorithms is shown in the context of Wireless Sensor Networks without restricting the large spectrum of applications that can take advantage of the presented study in this paper.

The rest of this paper is organized as follows. Next section describes the gossip algorithm for averaging applications. In Section 3, we present strategies for PCA. We describe the gossip algorithm for PCA in Section 4. Section 5 provides experimental results and discussions, whereas Section 6 concludes the paper.

This work is supported by Université de Technologie de Troyes, Université Libanaise, and Région Champagne-Ardenne (grant "WiDiD").

2. AVERAGING GOSSIP

Consider a network of N nodes. In a centralized network, the nodes communicate with a fusion center (FC); in a decentralized network, the nodes provide in-network processing, either in a noncooperative way by a given routing path, or in a cooperative way where each node communicates with its neighboring nodes. Let $\mathcal{V}_k$ denote the set of indices of the neighboring nodes to agent k, *i.e.*, the nodes that are directly connected to k. We consider that k is not adjacent to itself, that is to say $k \notin \mathcal{V}_k$. Let $\boldsymbol{x}_k$ be the $(p \times 1)$ vector of measures collected by the node k, 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 $\boldsymbol{x}_k, \boldsymbol{x}_l \in \mathbb{X}$. Let $\boldsymbol{w}_k$ be some feature extracted from the measurements and corresponding to the node k. We are interested, in this section, in the case where the nodes estimate the average of $\boldsymbol{w}_k$, $k = 1, \cdots, N$.

In a centralized network, the nodes send their features/states $\boldsymbol{w}_k$, $k = 1, \cdots, N$ to a FC where the average is computed via the formula $\frac{1}{N}\sum_{k=1}^{N} \boldsymbol{w}_k$. In a noncooperative strategy, nodes communicate according to a routing process. Each node k receives an estimate from the previous node, adds $\frac{1}{N}\boldsymbol{w}_k$ to it, and transmits it to the following node. This technique requires a well-defined routing scheme and a synchronous transmission/reception of the information exchange. Gossip algorithms provide an elegant asynchronous solution to overcome these issues, without the need for a FC.

Gossip algorithms use an asynchronous protocol with low computational complexity, described as follows. At iteration t, an arbitrarily chosen node k picks out randomly one of its neighbors, designed by node l. Node l sends its state to node k which produces a new state according to

$$\boldsymbol{w}_{k,t} = (1-\epsilon)\boldsymbol{w}_{k,t-1} + \epsilon \boldsymbol{w}_{l,t-1}, \tag{1}$$

for a mixing parameter $\epsilon \in [0, 1]$. The states of the other nodes remain unchanged, namely $\boldsymbol{w}_{j,t} = \boldsymbol{w}_{j,t-1}$ for all $j \neq k$. This is the asymmetric gossip, described in details in [16].

Unlike the asymmetric gossip, the symmetric gossip described in [12] uses a symmetric communication protocol between nodes. In such case, both nodes k and l exchange their states to compute the update according to the mixing parameter $\epsilon \in [0, 1]$, as follows:

$$\boldsymbol{w}_{k,t} = (1-\epsilon)\boldsymbol{w}_{k,t-1} + \epsilon \boldsymbol{w}_{l,t-1}, \tag{2}$$

$$\boldsymbol{w}_{l,t} = \epsilon \boldsymbol{w}_{k,t-1} + (1-\epsilon)\boldsymbol{w}_{l,t-1}. \tag{3}$$

The states of the other nodes remain unchanged. We note that this update preserves the total sum, and hence the mean of the nodes' estimates [17], since we have $\sum_{k=1}^{N} \boldsymbol{w}_{k,t} = \sum_{k=1}^{N} \boldsymbol{w}_{k,t-1}$. Note that this characteristic does not apply to the asymmetric gossip. This is the main cause why symmetric gossip outperforms asymmetric gossip, but with the cost of a much higher energy consumption for communication and computing.

3. PRINCIPAL COMPONENT ANALYSIS

In this section, we review previous works for PCA in networks. 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 $y_{\boldsymbol{w},k}$. The ultimate goal would be to compute $\boldsymbol{w}$, optimally for some given cost function and according to the network topology, and as a consequence to keep afterwards only $y_{\boldsymbol{w},k}$ (and $\boldsymbol{w}$), for $k = 1, ..., N$, which allows to represent efficiently the data $\boldsymbol{x}_k$.

3.1. Centralized strategy

Here, the network is assumed to be centralized, with nodes connected to the FC, to which they send their data without any in-networks processing. In order to extract the first principal axis, the FC 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 the input data. By taking the empirical estimation of the variance of the projected data 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. The problem takes the form $\boldsymbol{C}\boldsymbol{w} = \lambda \boldsymbol{w}$. This is the well-known eigen-decomposition problem where $\boldsymbol{w}$ is the eigenvector associated with the eigenvalue λ of $\boldsymbol{C}$.

Having the data, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the FC solves the eigen-decomposition problem and associates the eigenvector with the largest eigenvalue to the principal axis, denoted $\boldsymbol{w}_*$ in the following. Resolving of the eigen-decomposition problem has a computational complexity $\mathcal{O}(p^3)$. There is also the communication complexity, which is $\mathcal{O}(Np)$ over a distance $\mathcal{O}(1)$. Such configuration is not scalable and inappropriate for network applications. In the following, we propose several strategies that allow reducing the computational complexity by avoiding the covariance matrix computation and the eigen-decomposition.

3.2. PCA-based distributed approach

The PCA-based distributed approach (PCADID) proposed in [18] is outlined as follows. The N nodes are divided into d clusters of N_i nodes, for $i = 1, \ldots, d$, where $\sum_{i=1}^{d} N_i = N$. Each cluster i deals the p_i features (*i.e.*, dimensions) where $\sum_{i=1}^{d} p_i = p$. In this setting, the PCADID algorithm operates in several steps, as follows. First, for each cluster i, data associated to the cluster are normalized to the range $[0\ , 1]$ and then one computes the column-centered matrix $\boldsymbol{X}_i^c$. The principal axes of the data in the matrix $\boldsymbol{X}_i^c$ are extracted using a singular value decomposition $\boldsymbol{X}_i^c = \boldsymbol{U}_i \boldsymbol{\Sigma}_i \boldsymbol{V}_i^\top$, or equivalently the eigen-decomposition of the corresponding covariance matrix. Each cluster sends its principal axis to its neighbor cluster according to a routing process. Finally, the principal axes of all the data are given by the union of the principal axes of all the clusters.

The PCADID operates through a "divide-to-conquer" scheme in order to reduce the computational complexity of the eigen-decomposition problem. However, it still requires the eigen-decomposition of several matrices, making it inappropriate for large-scale networks.

3.3. Noncooperative strategy

We propose to adaptively learn the first principal axis, drawing inspiration from Oja's rule [14] which is a single-neuron special case of the generalized Hebbian learning [19]. Each node k receives, according to a routing process, an estimate $\boldsymbol{w}_{t-1}$ from another node, and adjusts it using its own data $\boldsymbol{x}_k$ by maximizing $y^2_{\boldsymbol{w},k}$. The quadratic reconstruction error is minimized, namely

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

Applying the gradient descent technique on $J(\boldsymbol{w})$ gives the adaptive update rule:

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \eta_t\,(\boldsymbol{x}_k\, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k}\,\boldsymbol{w}_{t-1}), \tag{5}$$

where η_t is the learning rate.

The noncooperative strategy requires an "instantaneous" estimation of the principal axis at each node and necessitates a synchronous communication protocol. In our previous work, we proposed two cooperative strategies, adapt-then-combine and combine-then-adapt [20, 21]. Although these strategies do not require a routing process, they use a synchronous communication protocol with a broadcast scheme to diffuse information. Next, we propose to solve this issue by using the gossip for PCA.

4. GOSSIP FOR PCA

In this cooperative strategy, each node k has access to the information of its neighborhood $\mathcal{V}_k$, which is the subset of nodes currently connected to node k. We aim to minimize 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 $J_k(\boldsymbol{w}) = \frac{1}{4}\|\boldsymbol{x}_k - y_{\boldsymbol{w},k}\boldsymbol{w}\|^2$. The global cost function of node k can be decomposed as

$$\sum_{l=1}^{N} J_l(\boldsymbol{w}) = J_k(\boldsymbol{w}) + \sum_{\substack{l=1\\ l\neq k}}^{N} J_l(\boldsymbol{w}). \tag{6}$$

The first term in the right-hand-side is known at the node under scrutiny, while 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:

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

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 $J_l(\boldsymbol{w})$, as described in [22]. Since gossiping is used between nodes, at each iteration, node k communicates with only one of its neighbors l. Therefore, $b_{lk} = \nu$ where ν is a constant and $b_{jk} = 0$ for all $j \in \mathcal{V}_k\backslash\{l\}$. Thus, we have:

$$\sum_{\substack{l=1\\ l\neq k}}^{N} J_l(\boldsymbol{w}) \approx \nu\,\|\boldsymbol{w} - \boldsymbol{w}_*\|^2.$$

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. Injecting this approximation into (6) allows us to rewrite the global cost function at node k as follows

$$J_k^{\text{glob}}(\boldsymbol{w}) = J_k(\boldsymbol{w}) + \nu\|\boldsymbol{w} - \boldsymbol{w}_*\|^2. \tag{7}$$

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

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} - \eta_{k,t}\,\nabla_{\boldsymbol{w}} J_k^{\text{glob}}(\boldsymbol{w}_{k,t-1}). \tag{8}$$

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 (7), we get:

$$\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} - \eta_{k,t}\nabla_{\boldsymbol{w}} J_k(\boldsymbol{w}_{t-1}) + \eta_{k,t}\nu(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}).$$

In this expression, $\nabla_{\boldsymbol{w}} J_k(\boldsymbol{w}_{t-1})$ is the gradient of the PCA-based cost function. In this case, we get

$$\begin{aligned}\boldsymbol{w}_{k,t} = \boldsymbol{w}_{k,t-1} &+ \eta_{k,t}\Big(\boldsymbol{x}_k\, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k}\,\boldsymbol{w}_{k,t-1}\Big)\\ &+ \eta_{k,t}\nu(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}).\end{aligned} \tag{9}$$

This update from $\boldsymbol{w}_{k,t-1}$ to $\boldsymbol{w}_{k,t}$ involves adding two correction terms to $\boldsymbol{w}_{k,t-1}$. We decompose this update rule in two successive steps by including one correction term at a time. In the following, we show two possibilities of the update.

First possibility

We express the update rule (9) as follows:

$$\begin{cases} \boldsymbol{\phi}_{k,t} &= \boldsymbol{w}_{k,t-1} + \eta_{k,t}\left(\boldsymbol{x}_k y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k}\boldsymbol{w}_{k,t-1}\right),\\ \boldsymbol{w}_{k,t} &= \boldsymbol{\phi}_{k,t} + \eta_{k,t}\,\nu(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}).\end{cases}$$

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 $\boldsymbol{\phi}_{k,t}$. The second step updates $\boldsymbol{\phi}_{k,t}$ to $\boldsymbol{w}_{k,t}$. This second step is not realizable since the nodes do not know $\boldsymbol{w}_*$. However, each node l has its own approximation for $\boldsymbol{w}_*$, which is its local intermediate estimate $\boldsymbol{\phi}_{l,t}$, and since the parameter ν refers to the neighboring node l, we replace $\boldsymbol{w}_*$ by $\boldsymbol{\phi}_{l,t}$. On the other hand, the intermediate value $\boldsymbol{\phi}_{k,t}$ at

node k is obtained by adapting the information as given by the first step. Thus, it is generally a better estimate for $\boldsymbol{w}_*$ than $\boldsymbol{w}_{k,t-1}$. Therefore, we further replace $\boldsymbol{w}_{k,t-1}$ by $\boldsymbol{\phi}_{k,t}$ in the second step. Hence, the second step is replaced by:

$$\begin{aligned}\boldsymbol{w}_{k,t} &= \boldsymbol{\phi}_{k,t} + \eta_{k,t} \sum_{l \in \mathcal{V}_k \setminus \{k\}} b_{lk} \left(\boldsymbol{\phi}_{l,t} - \boldsymbol{\phi}_{k,t}\right) \\ &= \Big(1 - \eta_{k,t}\nu\Big)\, \boldsymbol{\phi}_{k,t} \, + \eta_{k,t}\nu \, \phi_{l,t} \\ &= \Big(1 - \epsilon\Big)\, \boldsymbol{\phi}_{k,t} \, + \epsilon \, \phi_{l,t}.\end{aligned}$$

Finally, the update rule for the adapt-then-asymmetric-gossip (ATAG) strategy is:

$$\left\{\begin{array}{ll} \boldsymbol{\phi}_{k,t} & = \boldsymbol{w}_{k,t-1} + \eta_{k,t} \Big(\boldsymbol{x}_k y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k} \boldsymbol{w}_{k,t-1}\Big), \\ \boldsymbol{w}_{k,t} & = (1-\epsilon)\boldsymbol{\phi}_{k,t} + \epsilon \boldsymbol{\phi}_{l,t}. \end{array}\right.$$

Motivated by the symmetric averaging gossip where the update is applied for the node k and its neighbor l, we apply the equations of ATAG on the two nodes. Therefore, we obtain the adapt-then-symmetric-gossip (ATSG)technique with the following update rule:

$$\left\{\begin{array}{ll} \boldsymbol{\phi}_{k,t} & = \boldsymbol{w}_{k,t-1} + \eta_{k,t} \Big(\boldsymbol{x}_k y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k} \boldsymbol{w}_{k,t-1}\Big), \\ \boldsymbol{\phi}_{l,t} & = \boldsymbol{w}_{l,t-1} + \eta_{l,t} \Big(\boldsymbol{x}_l y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l} \boldsymbol{w}_{l,t-1}\Big), \\ \boldsymbol{w}_{k,t} & = (1-\epsilon)\boldsymbol{\phi}_{k,t} + \epsilon \boldsymbol{\phi}_{l,t}, \\ \boldsymbol{w}_{l,t} & = \epsilon\boldsymbol{\phi}_{k,t} + (1-\epsilon) \boldsymbol{\phi}_{l,t}. \end{array}\right.$$

Second possibility

In this strategy, we express the update rule (9) as follows:

$$\left\{\begin{array}{ll} \boldsymbol{\phi}_{k,t} & = \boldsymbol{w}_{k,t-1} + \eta_{k,t}\, \nu(\boldsymbol{w}_* - \boldsymbol{w}_{k,t-1}) \\ \boldsymbol{w}_{k,t} & = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \Big(\boldsymbol{x}_k \, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k} \, \boldsymbol{w}_{k,t-1}\Big). \end{array}\right.$$

For the same reasons shown in the first possibility, we replace in the first step $\boldsymbol{w}_*$ by $\boldsymbol{w}_{l,t-1}$ and in the second step we replace $\boldsymbol{w}_{l,t-1}$ by $\boldsymbol{\phi}_{l,t}$. We obtain the update rule for the asymmetric-gossip-then-adapt (AGTA) strategy:

$$\left\{\begin{array}{ll} \boldsymbol{\phi}_{k,t} & = (1-\epsilon)\boldsymbol{w}_{k,t-1} + \epsilon \boldsymbol{w}_{l,t-1}, \\ \boldsymbol{w}_{k,t} & = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \Big(\boldsymbol{x}_k \, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k} \, \boldsymbol{\phi}_{k,t}\Big). \end{array}\right. \tag{10}$$

and the symmetric-gossip-then-adapt (SGTA) strategy:

$$\left\{\begin{array}{ll} \boldsymbol{\phi}_{k,t} & = (1-\epsilon)\boldsymbol{w}_{k,t-1} + \epsilon \boldsymbol{w}_{l,t-1}, \\ \boldsymbol{\phi}_{l,t} & = \epsilon\boldsymbol{w}_{k,t-1} + (1-\epsilon) \boldsymbol{w}_{l,t-1}, \\ \boldsymbol{w}_{k,t} & = \boldsymbol{\phi}_{k,t} + \eta_{k,t} \Big(\boldsymbol{x}_k \, y_{\boldsymbol{w}_{t-1},k} - y^2_{\boldsymbol{w}_{t-1},k} \, \boldsymbol{\phi}_{k,t}\Big), \\ \boldsymbol{w}_{l,t} & = \boldsymbol{\phi}_{l,t} + \eta_{l,t} \Big(\boldsymbol{x}_l \, y_{\boldsymbol{w}_{t-1},l} - y^2_{\boldsymbol{w}_{t-1},l} \, \boldsymbol{\phi}_{l,t}\Big). \end{array}\right.$$

Convergence

All proposed learning rules converge to the first principal axis $\boldsymbol{w}_*$. This can be verified by considering, for instance, the AGTA strategy given in (10). When $\boldsymbol{w}_{k,t}$ for $k = 1, \cdots, N$ converges to some state $\boldsymbol{w}$, in the first equation, we have for $k = 1, \cdots, N$:

$$\boldsymbol{\phi}_{k,t} = (1-\epsilon)\boldsymbol{w} + \epsilon \boldsymbol{w} = \boldsymbol{w}.$$

As for the second equation, it comes down to

$$\boldsymbol{x}_k y_{\boldsymbol{w},k} = y^2_{\boldsymbol{w},k} \boldsymbol{w},$$

namely

$$\boldsymbol{x}_k \boldsymbol{x}_k^\top \boldsymbol{w} = y^2_{\boldsymbol{w},k} \boldsymbol{w}.$$

Averaging over the whole data, we get the well-known eigen-decomposition problem of the covariance matrix $\boldsymbol{C}\boldsymbol{w} = \boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w}\, \boldsymbol{w}$,where the eigenvalue $\boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w}$ corresponds to the squared output y_i that one wishes to maximize. Therefore, the update rule of AGTA converges to the largest eigenvector of the covariance matrix, without the need to compute $\boldsymbol{C}$. The same demonstration can be applied to the other strategies.

5. EXPERIMENTATIONS

In this section, we illustrate the performance of the proposed approach and compare it to the algorithm presented in [23] as well as the aforementioned PCADID algorithm [18]. In order to provide a fair comparative study, we use the same initial random estimate for all strategies.

As an application to the proposed study, we consider the problem of tracking a gas spread using a wireless sensor network (WSN) [24]. The sensor k at a position denoted by $\boldsymbol{z}_k \in \mathbb{Z}$ measures, at time t, a gas quantity denoted by $x_{k,t}$. The region under scrutiny $\mathbb{Z} = [-0.5, 0.5] \times [-0.5, 0.5]$ is a two-dimensional unit-area. We aim to reduce the time series order 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 at 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 uniformly deployed 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: two nodes are considered as being connected when they are less than s units of distance apart, that is $\mathcal{V}_k = \{l \, : \, \|\boldsymbol{z}_k - \boldsymbol{z}_l\| < s\}$. In our experiments, we set this threshold to $s = 0.38$. For the stepsize parameters, we consider $\eta = 0.0025$ and $\epsilon = 0.5$. 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 $\Theta_i = \arccos \frac{\boldsymbol{w}_{*,l}^\top \boldsymbol{w}_*}{\|\boldsymbol{w}_{*,l}\| \|\boldsymbol{w}_*\|}$. Results are shown in terms of the angle averaged over all the nodes. Figure 1 shows the convergence of the proposed strategies for WSN. It shows that all the proposed strategies outperform the PCADID algorithm [18]. These learning curves show the benefits of implementing the gossip. The symmetric gossip outperform the asymmetric gossip as expected. Moreover, we note that the order of gossip (AGTA/ATAG or SGTA/ATSG) has almost the same performance. Note that the noncooperative strategy presents fluctuations mainly because of the routing system and it requires a synchronous protocol of communication. Figure 1 shows the stability of the gossip strategies. Although using gossip makes the convergence slow, but it solves the problem of synchronization and communication between nodes.

The algorithm Korada *et al.* [23]is a centralized algorithm based on gossiping the PCA. By using a sparsification[1] of $\boldsymbol{C}$, it is multiplied at each iteration by the last estimate $\boldsymbol{w}_t$, then normalized. The final result is the gossip average of all estimates. Figure 1 shows that the proposed approach outperforms Korada *et al.* algorithm.

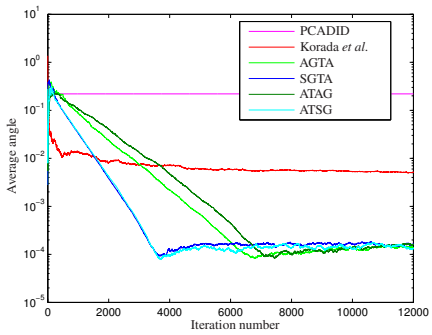

Fig. 1. Convergence of the proposed strategies for WSN, measured by the average angle.

6. CONCLUSION

In this paper, we studied the issue of estimating the principal axis from PCA in networks. In order to relax constraints on reception synchronism, we proposed to apply gossip in PCA. Experiments are done taking into account the constraints imposed in WSNs. The results showed the relevance of the proposed strategies. Future works will include the extraction of multiple axes.

REFERENCES

[1] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] P. E. Gill, G. H. Golub, W. A. Murray, and M. A. Saunders, "Methods for modifying matrix factorizations." Stanford, CA, USA, Tech. Rep., 1972.

[1] a sparsification of a given matrix is obtained by setting to 0 some of its entries and rescaling the non-zero entries.

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

[9] 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.

[10] J. Lu and G. Chen, "A time-varying complex dynamical network model and its controlled synchronization criteria," *Automatic Control, IEEE Transactions on*, vol. 50, no. 6, pp. 841–846, June 2005.

[11] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *Network, IEEE*, vol. 18, no. 4, pp. 45–50, July 2004.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, June 2006.

[13] C. Asensio-Marco and B. Beferull-Lozano, "Fast average gossiping under asymmetric links in wsns," in *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, Sept 2014, pp. 131–135.

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

[15] 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.

[16] F. Fagnani and S. Zampieri, "Asymmetric randomized gossip algorithms for consensus," in *IFAC World Conference*, 2008, pp. 9052–9056.

[17] ——, "Randomized consensus algorithms over large scale networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 634–649, May 2008.

[18] M. A. Livani and M. Abadi, "A PCA-based distributed approach for intrusion detection in wireless sensor networks," in *International Symposium on Computer Networks and Distributed Systems*, 2011.

[19] R. Dony and S. Haykin, "Neural network approaches to image compression," *Proceedings of the IEEE*, vol. 83, no. 2, pp. 288–303, Feb 1995.

[20] N. Ghadban, P. Honeine, F. Mourad-Chehade, C. Francis, and J. Farah, "Diffusion strategies for in-network principal component analysis," in *Proc. 24th IEEE workshop on Machine Learning for Signal Processing*, Reims, France, 21–24 September 2014.

[21] N. Ghadban, P. Honeine, C. Francis, F. Mourad-Chehade, and J. Farah, "Strategies for principal component analysis in wireless sensor networks," in *Proc. eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, A Coruña, Spain, 22–25 June 2014.

[22] 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.

[23] S. B. Korada, A. Montanari, and S. Oh, "Gossip pca," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '11. New York, NY, USA: ACM, 2011, pp. 209–220. [Online]. Available: http://doi.acm.org/10.1145/1993744.1993764

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