Distributed Dictionary Learning via Projections onto Convex Sets

D. Ampeliotis, C. Mavrokefalidis, K. Berberidis
Department of Computer Engineering and Informatics, University of Patras
& C.T.I RU-2, 26500, Rio - Patras, Greece
E-mails: {ampeliot, maurokef, berberid}@ceid.upatras.gr

***Abstract*—We study a problem in which the nodes of a network, each with different data, are interested in computing a common dictionary that is suitable for the efficient sparse coding of all their data. To this end, distributed processing is employed, that is, the nodes merge local and neighboring information. We formulate this as a convex feasibility problem, and propose a suitable distributed algorithm for obtaining a solution that employs projections onto convex sets. A fast method for computing the involved projection operations is also given. The proposed approach allows the associated convex sets to be updated at every iteration of the algorithm, thus resulting into a faster agreement of the nodes in a common dictionary. Simulation results are provided that demonstrate the effectiveness of the proposed scheme in computing a common dictionary, in a scenario where the data of the nodes are significantly different and a second scenario, in which the nodes have the same data.**

I. Introduction

The description of signals using sparse representations is an active area of research [1], [2] and many applications employ this type of modeling, such as medical imaging, audio / video denoising and compression [3]. A notably interesting approach that is often adopted, so-called “dictionary learning”, leads to over-determined linear models for the available data [4]. In particular, these approaches aim to determine a set of representative signals (or, else, atoms) that constitutes a dictionary. Based on this dictionary, each desired signal is modeled as a linear combination of only a small number of atoms leading to a corresponding sparse representation.

In recent years, due to the development of small devices, either mobile or static, with communication and computing capabilities, new data processing possibilities have been created. Being able to set-up networks and carry out tasks of common interest, these devices cooperate and process data in a distributed fashion [5], instead of a centralized one which relies on a central node. This way, increased robustness and scalability are achieved as, on one hand, there is no single point of failure and, on the other hand, energy and communication resources are only required for local interactions and processing. Although distributed algorithms were initially developed for estimation problems (e.g., in power-grids [6]), nowadays, such algorithms are also available for the dictionary learning problem.

This work was supported in part by the University of Patras, and the General Secretariat for Research and Technology, Greece, under the project FP7 ERANET COM-MED.

Generally speaking, the distributed algorithms for dictionary learning, are devised for a network of nodes (e.g., microphones, cameras, etc.) that acquire observations originating from the same phenomenon. Furthermore, it is assumed that these observations can be sparsely represented based on a common dictionary. These algorithms usually have two main parts. The first one is focused on the processing of local observations while, in the second part, the devices exchange relevant information (e.g., the actual local dictionaries) and attempt to agree upon a common, global dictionary.

In the following, some representative works will be briefly discussed. In particular, [7] proposes a diffusion-based, adaptive dictionary learning approach in which the devices adapt their local copy of the dictionary using the so-called adapt-then-combine strategy. In [8], [9], the desired dictionary is assumed to be known only partially by each device and a representation error is exchanged, minimizing the communication overhead while maintaining the privacy of the local data of the devices. In [10] and [11], two consensus-based, distributed algorithms are proposed. In the first one, the alternate direction method of multipliers is utilized among the devices for acquiring the common dictionary. In the second one (and also in [12] where a more extended version can be found), a distributed version of the centralized K-SVD algorithm is proposed, while convergence analysis of the algorithm is presented in [13]. Additionally, in [14], a consensus-based, distributed algorithm for general inference / learning problems is proposed which can also be applied for the problem of dictionary learning. An online algorithm has appeared in [15], where the recursive least squares algorithm is employed. Other online algorithms can be found in [16] and [17], which are specifically tailored for classification tasks, concerning object recognition in natural images and fossil pollen grains in microscopy images, respectively. In [18], a distributed algorithm is described by proposing an adapt-align-combine strategy that takes into account an inherent permutation ambiguity concerning the atoms of the dictionary.

In this paper, a novel distributed, dictionary learning approach is proposed for a network of devices that have, in general, different observations (e.g., a network of cameras with different points of view). In order to agree upon a common, global dictionary, instead of using gradient-based techniques such as the ones utilized in the aforementioned works, the devices are associated with individual convex sets and they project on those sets the local dictionaries of neighboring devices in an incremental (i.e., sequential) manner. A fast technique is also determined for the Projection Onto Convex Sets (POCS) operation. Moreover, the approach updates the associated convex sets per device in every iteration for increased performance. For evaluation purposes, simulation results are presented for a network of five nodes that utilize either different or the same images. The results demonstrate the effectiveness of the proposed approach.

II. Problem Formulation

We consider a network of N nodes, where each node $n \in \mathcal{N} = \{1, 2, \ldots, N\}$ has obtained some data that we represent by the matrix $\boldsymbol{Y}_n \in \mathbb{R}^{p \times q_n}, \quad n \in \mathcal{N}$, where p is the dimension of the data samples and q_n is the number of observed samples at node n. For example, each column of $\boldsymbol{Y}_n$ could represent a small portion of an image (known as an image patch), in which case p is the number of pixels of the patch and q_n is the number of patches at node n. In general, the nodes are interconnected as described by a connected graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{N}$ is the set of vertices and $\mathcal{E}$ is the set of edges.

We are interested in the computation of a so-called dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{p \times K}, \ p < K$, common for all nodes, so that

$$\boldsymbol{Y}_n \approx \boldsymbol{D}\boldsymbol{A}_n, \quad \forall \ n \in \mathcal{N}, \tag{1}$$

with the additional requirement that the respective matrices $\boldsymbol{A}_n \in \mathbb{R}^{K \times q_n}$ are sparse, in the sense that the number of non-zero elements on each column is small. Furthermore, we require that each column $\boldsymbol{d}_k, k = 1, 2, \ldots, K$ of $\boldsymbol{D}$ has unit length. The scope of distributed dictionary learning is to compute the required matrix $\boldsymbol{D}$ in a distributed fashion, where the nodes exchange local messages with respect to the graph $G(\mathcal{V}, \mathcal{E})$ and, thus, there is no need for a so-called *fusion-center* that gathers all the required data and executes a centralized algorithm.

In this work, we consider that each node n at discrete time instant t has computed the estimates $\boldsymbol{D}_n^{(t)}$ and $\boldsymbol{A}_n^{(t)}$. For example, such estimates could be the result of some centralized dictionary learning and sparse approximation algorithm [4] that was applied locally at node n, using local data $\boldsymbol{Y}_n$. In the following, we develop a proper distributed algorithm to make the nodes agree on a common $\boldsymbol{D}$.

III. POCS-Based Distributed Dictionary Learning

Let us consider that, at some discrete time instant t, the representation error $v_n^{(t)}$ for the data of node n is given by the following equation, based upon the Frobenius norm

$$v_n^{(t)} = \left\| \boldsymbol{Y}_n - \boldsymbol{D}_n^{(t)} \boldsymbol{A}_n^{(t)} \right\|_{\mathrm{F}}^2 \ . \tag{2}$$

Furthermore, for each node n, we define the set

$$\mathcal{S}_n^{(t)} = \{ \boldsymbol{D} \in \mathbb{R}^{p \times K} : \left\| \boldsymbol{Y}_n - \boldsymbol{D}\boldsymbol{A}_n^{(t)} \right\|_{\mathrm{F}}^2 \leq v_n^{(t)} \} \ , \tag{3}$$

that is, the set of all dictionary matrices for which the representation error (given $\boldsymbol{A}_n^{(t)}$) is not increased. This set is convex and the following lemma provides a proof.

Lemma 1: The set $\mathcal{S}_n^{(t)}$ is convex.
The set $\mathcal{S}_n^{(t)}$ is convex if the condition

$$\forall \boldsymbol{D}_1, \ \boldsymbol{D}_2 \in \mathcal{S}_n^{(t)} \Rightarrow \left(\zeta \boldsymbol{D}_1 + (1 - \zeta) \boldsymbol{D}_2 \right) \in \mathcal{S}_n^{(t)} \tag{4}$$

is valid for any $\zeta \in [0, 1]$.
Proof: For any dictionaries $\boldsymbol{D}_1, \ \boldsymbol{D}_2 \in \mathcal{S}_n^{(t)}$, the inequalities

$$\|\boldsymbol{Y}_n - \boldsymbol{D}_1 \boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 \quad \leq \quad v_n^{(t)}, \tag{5}$$

$$\|\boldsymbol{Y}_n - \boldsymbol{D}_2 \boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 \quad \leq \quad v_n^{(t)}, \tag{6}$$

hold according to (3). Multiplying (5), (6) with ζ^2 and $(1-\zeta)^2$, respectively, the following inequalities are also true.

$$\zeta^2 \|\boldsymbol{Y}_n - \boldsymbol{D}_1 \boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 \leq \zeta^2 v_n^{(t)} \leq \zeta v_n^{(t)}, \tag{7}$$

$$(1 - \zeta)^2 \|\boldsymbol{Y}_n - \boldsymbol{D}_2 \boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 \leq (1 - \zeta)^2 v_n^{(t)} \leq (1 - \zeta) v_n^{(t)}. \tag{8}$$

Finally, by adding up (7), (8), it can be shown that

$$\|\boldsymbol{Y}_n - \left(\zeta \boldsymbol{D}_1 + (1 - \zeta) \boldsymbol{D}_2 \right) \boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 \leq v_n^{(t)}, \tag{9}$$

which proves the lemma. ■

Based on the above, we define the convex set $\mathcal{C}^{(t)}$ as the intersection of the individual convex sets $\mathcal{S}_n^{(t)}$, given by

$$\mathcal{C}^{(t)} = \bigcap_{n \in \mathcal{N}} \mathcal{S}_n^{(t)} \subset \mathbb{R}^{p \times K} \ . \tag{10}$$

In the case where the set $\mathcal{C}^{(t)}$ is not empty, any matrix $\boldsymbol{D}^{(t)} \in \mathcal{C}^{(t)}$ will satisfy the requirements of all nodes. Thus, we can let the nodes compute such a matrix to enforce consensus among them. However, in order to also cover the case in which the set $\mathcal{C}^{(t)}$ might be empty, we propose to enforce consensus by setting

$$\boldsymbol{D}^{(t)} = \arg\min_{\boldsymbol{D}} \sum_{n \in \mathcal{N}} \left\| \boldsymbol{D} - \mathcal{P}_{\mathcal{S}_n^{(t)}}(\boldsymbol{D}) \right\|_{\mathrm{F}}^2 \ , \tag{11}$$

where $\mathcal{P}_{\mathcal{S}_n^{(t)}}(\boldsymbol{D})$ is the orthogonal projection of $\boldsymbol{D}$ onto $\mathcal{S}_n^{(t)}$. It can be easily verified that (11) includes also the case of non-empty set $\mathcal{C}^{(t)}$, since for any $\boldsymbol{D}^{(t)} \in \mathcal{C}^{(t)}$, equation (11) attains zero cost. The optimization problem in (11) can be solved by adopting an approach that is inspired by the POCS method [19], [20]. Furthermore, the adopted approach can be implemented in a distributed fashion, similarly to the method shown in [21], [22], for other estimation problems.

After organizing the nodes of the network into a circle, each node n participates in an iterative update algorithm by computing, at iteration i, an estimate $\boldsymbol{D}_n^{(t,i)}$ of the global dictionary. In particular,

$$\boldsymbol{D}_n^{(t,i)} \leftarrow \hat{\boldsymbol{D}}_n^{(t,i)} + \lambda \left(\mathcal{P}_{\mathcal{S}_n^{(t)}} \left(\hat{\boldsymbol{D}}_n^{(t,i)} \right) - \hat{\boldsymbol{D}}_n^{(t,i)} \right) \ , \tag{12}$$

where $\hat{\boldsymbol{D}}_n^{(t,i)}$ is an estimate computed by the previous node in the circle and received by node n, and λ is a properly selected constant (in more detail, a properly selected relaxation sequence can be used [19], [20]).

The most demanding operation in (12) is the involved projection operation. In the following subsection, we derive an algorithm that computes the required orthogonal projection efficiently.

A. Efficient computation of the projection operator

For the computation of the projection $\mathcal{P}_{\mathcal{S}_n^{(t)}}\left(\hat{\boldsymbol{D}}_n^{(t,i)}\right)$ in (12), the following optimization problem must be solved

$$\mathcal{P}_{\mathcal{S}_n^{(t)}}\left(\hat{\boldsymbol{D}}_n^{(t,i)}\right) = \arg\min_{\boldsymbol{D}} \left\|\boldsymbol{D} - \hat{\boldsymbol{D}}_n^{(t,i)}\right\|_{\mathrm{F}}^2 \quad \text{s.t. } \|\boldsymbol{Y}_n - \boldsymbol{D}\boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 - v_n^{(t)} \leq 0 \; . \tag{13}$$

We can use the method of Lagrange multipliers to transform this problem into an unconstrained one. In particular, the Lagrangian for our problem is given by

$$\mathcal{L}(\boldsymbol{D},\mu) = \left\|\boldsymbol{D} - \hat{\boldsymbol{D}}_n^{(t,i)}\right\|_{\mathrm{F}}^2 - \mu\left(\|\boldsymbol{Y}_n - \boldsymbol{D}\boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 - v_n^{(t)}\right) \; , \tag{14}$$

where μ is our Lagrange multiplier. Setting the partial derivative of the Lagrangian function with respect to $\boldsymbol{D}$ equal to zero, yields

$$\boldsymbol{D} = \left(\hat{\boldsymbol{D}}_n^{(t,i)} - \mu\boldsymbol{Y}_n(\boldsymbol{A}_n^{(t)})^{\mathrm{T}}\right)\left(\boldsymbol{I} - \mu\boldsymbol{A}_n^{(t)}(\boldsymbol{A}_n^{(t)})^{\mathrm{T}}\right)^{-1} \; , \tag{15}$$

where $\boldsymbol{I}$ denotes the identity matrix and $()^{\mathrm{T}}$ is the matrix transpose operator. Of course, setting the partial derivative with respect to μ equal to zero, yields the equation of the constraint. Unfortunately, substituting (15) into the equation of the constraint does not lead to a closed formula for μ. However, taking into account the fact that the projection of any point that lies outside a convex set onto that convex set is unique, the equation can be solved for μ as follows,

$$\|\boldsymbol{Y}_n - \left(\hat{\boldsymbol{D}}_n^{(t,i)} - \mu\boldsymbol{Y}_n(\boldsymbol{A}_n^{(t)})^{\mathrm{T}}\right)\cdots \cdots\left(\boldsymbol{I} - \mu\boldsymbol{A}_n^{(t)}(\boldsymbol{A}_n^{(t)})^{\mathrm{T}}\right)^{-1}\boldsymbol{A}_n^{(t)}\|_{\mathrm{F}}^2 - v_n^{(t)} = 0 \; , \tag{16}$$

using (for example) the bisection method. The resulting unique μ_o can then be inserted into equation (15) to compute the projection. Simulation results demonstrated that this procedure is much faster than solving the original problem (13) using the standard CVX solver. It can easily be verified that the involved projection operations might violate the requirement for having dictionaries with unit length atoms. However, we can easily normalize the resulting dictionaries so as to have unit length atoms, by also scaling the respective rows of the associated sparse approximation matrices.

B. A heuristic modification for accelerating POCS

In this paragraph, departing from the standard POCS literature, we propose a modification in which the involved sets are updated from iteration to iteration. In more detail, after node n has applied (12) to update its dictionary estimate, it normalizes this estimate to have unit length. We denote the normalized dictionary as $\bar{\boldsymbol{D}}_n^{(t,i)}$. Then, it also updates its local sparse

Input: Data matrix $\boldsymbol{Y}_n$, next node index n' , relaxation parameter λ , total POCS iterations I , knowledge if node is leader or not , sparse approximation algorithm $\mathcal{F}_1(\cdot)$, dictionary update algorithm $\mathcal{F}_2(\cdot)$

Initialize $\bar{\boldsymbol{D}}_n^{(0,I)}$ with unit length atoms
Initialize $\boldsymbol{A}_n^{(0,I)}$
for $t = 1$ **to** ∞ **do**
  $\boldsymbol{A}_n^{(t,0)} \leftarrow \mathcal{F}_1\left(\bar{\boldsymbol{D}}_n^{(t-1,I)}, \boldsymbol{A}_n^{(t-1,I)}, \boldsymbol{Y}_n\right)$
  $\boldsymbol{D}_n^{(t,0)} \leftarrow \mathcal{F}_2\left(\bar{\boldsymbol{D}}_n^{(t-1,I)}, \boldsymbol{A}_n^{(t,0)}, \boldsymbol{Y}_n\right)$
  if Node n is the leader **then**
    Send $\boldsymbol{D}_n^{(t,0)}$ to the next node n'
  end if
  for $i = 1$ **to** I **do**
    Listen until $\hat{\boldsymbol{D}}_n^{(t,i)}$ is received from previous node
    $\boldsymbol{D}_n^{(t,i)} \leftarrow \hat{\boldsymbol{D}}_n^{(t,i)} + \lambda\left(\mathcal{P}_{\mathcal{S}_n^{(t,i)}}\left(\hat{\boldsymbol{D}}_n^{(t,i)}\right) - \hat{\boldsymbol{D}}_n^{(t,i)}\right)$
    Set $\bar{\boldsymbol{D}}_n^{(t,i)}$ as the atom-normalized $\boldsymbol{D}_n^{(t,i)}$
    Send $\bar{\boldsymbol{D}}_n^{(t,i)}$ to the next node n'
    $\boldsymbol{A}_n^{(t,i)} \leftarrow \mathcal{F}_1\left(\bar{\boldsymbol{D}}_n^{(t,i)}, \boldsymbol{A}_n^{(t,i-1)}, \boldsymbol{Y}_n\right)$
    $v_n^{(t,i)} \leftarrow \left\|\boldsymbol{Y}_n - \bar{\boldsymbol{D}}_n^{(t,i)}\boldsymbol{A}_n^{(t,i)}\right\|_{\mathrm{F}}^2$
  end for
end for

TABLE I
THE PROPOSED DISTRIBUTED DICTIONARY LEARNING ALGORITHM AT NODE n

approximation matrix and the associated error. In particular, the following equations are used

$$\boldsymbol{A}_n^{(t,i)} \leftarrow \mathcal{F}_1\left(\bar{\boldsymbol{D}}_n^{(t,i)}, \boldsymbol{A}_n^{(t,i-1)}, \boldsymbol{Y}_n\right) \tag{17}$$

and

$$v_n^{(t,i)} \leftarrow \left\|\boldsymbol{Y}_n - \bar{\boldsymbol{D}}_n^{(t,i)}\boldsymbol{A}_n^{(t,i)}\right\|_{\mathrm{F}}^2 \; , \tag{18}$$

where $\boldsymbol{A}_n^{(t,i)}$ and $v_n^{(t,i)}$ are the iteration dependent version of $\boldsymbol{A}_n^{(t)}$ and $v_n^{(t)}$, that define the iteration dependent convex set $\mathcal{S}_n^{(t,i)}$. Finally, as mentioned also in the following subsection, $\mathcal{F}_1(\cdot)$ is any proper sparse approximation algorithm (e.g., the Matching Pursuit - MP - [23], the Orthogonal Matching Pursuit - OMP - [24], the Focal Underdetermined System Solver - FOCUSS - [25]).

Of course, the modified POCS - inspired algorithm loses the convergence guarantees of the original scheme. However, it was experimentally observed that the proposed heuristic algorithm reaches a steady state significantly faster than the original scheme. Furthermore, in all the experiments conducted (with $0 < \lambda < 1$) we observed that the algorithm always converges, although an actual analytical proof is missing.

C. The proposed algorithm

In this paragraph, we summarize the proposed distributed dictionary learning algorithm, outlined in Table I. We assume that the nodes of the network are equipped with a sparse approximation algorithm $\mathcal{F}_1(\cdot)$ (as the ones mentioned in the previous paragraph) and a dictionary update algorithm $\mathcal{F}_2(\cdot)$ (e.g., the Method of Optimal Directions - MOD - [26], the K-SVD [27]) to process their local data. After the execution of one step of $\mathcal{F}_1(\cdot)$ and $\mathcal{F}_2(\cdot)$, the nodes use the previous POCS - inspired iterative approach to compute a common dictionary. In particular, we assume that, the nodes of the

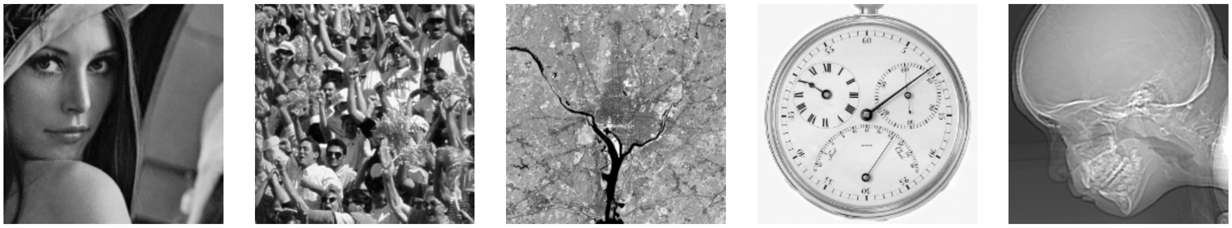

Fig. 1. The images used for generating the input data at the five nodes considered in the simulations

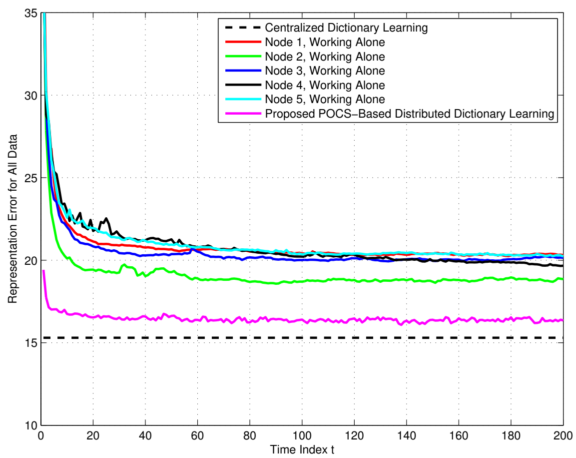

Fig. 2. Simulation results when the nodes have different data

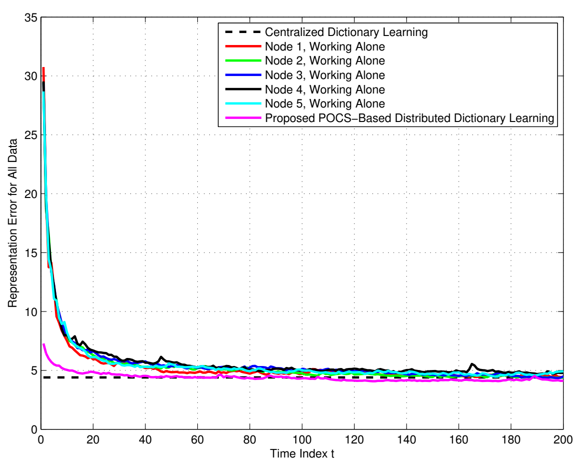

Fig. 3. Simulation results when the nodes have the same data

network have been organized into a circle, and one of them has been labeled as the leader node. The leader node starts the POCS iterations. A number of I such iterations are performed, where at each iteration the projection of the dictionary received from the previous node into the local convex set is computed. Thus, our proposed distributed dictionary learning algorithm employs projections onto convex sets, and allows for these convex sets to be time-varying.

IV. Simulations

We consider a network with 5 nodes, where the data at each node is created using some real world image. Each image is decomposed into non-overlapping patches of 8×8 pixels. The images used by the nodes can be seen in Fig. 1, and their common size is 192×192 pixels. All nodes use the OMP algorithm, in place of algorithm $\mathcal{F}_1(\cdot)$ (with a sparsity level equal to 10) for performing the sparse approximation steps, and the K-SVD algorithm, in place of algorithm $\mathcal{F}_2(\cdot)$ for performing the dictionary update steps. We consider dictionaries with $K = 128$ atoms, thus the size of the dictionaries is 64×128.

In the following we demonstrate the performance of the proposed algorithm against a centralized scheme that uses the K-SVD / OMP algorithms and the performance achieved when each node works alone, i.e., when no consensus is enforced. In all cases, the representation error for all the data of the nodes is the adopted performance metric (global error), which is plotted against the time index t of the algorithms. In more detail, Figures 2 and 3 show

$$e_n^{(t)} = \left\| \boldsymbol{Y} - \boldsymbol{D}_n^{(t)} \boldsymbol{A}_{n,g}^{(t)} \right\|_{\mathrm{F}}^2 ,$$

where $\boldsymbol{Y} = [\boldsymbol{Y}_1 \ \boldsymbol{Y}_2 \ \cdots \boldsymbol{Y}_5]$ is the matrix of all data, $\boldsymbol{D}_n^{(t)}$ is $\boldsymbol{D}_n^{(t,I)}$ for the cases where the nodes cooperate and $\boldsymbol{D}_n^{(t,0)}$ for the cases where the nodes work alone, and finally $\boldsymbol{A}_{n,g}^{(t)}$ is the sparse approximation matrix computed for all data $\boldsymbol{Y}$ using dictionary $\boldsymbol{D}_n^{(t)}$. Note that the proposed algorithm also utilizes sub-iterations (for the consensus step) which are not shown here. Thus, as the other schemes do not utilize such sub-iterations, the obtained results are better suited for steady-state comparisons rather than convergence speed comparisons. Also, for the centralized scheme, only the error obtained at the steady state is shown.

Note that the proposed distributed dictionary learning method comprises two phases, namely a phase in which all nodes update their dictionaries using local data and a phase in which the nodes execute a distributed algorithm to reach an agreement about a common dictionary. These two phases are repeated in time. Also, at the first phase, any dictionary learning algorithm can be utilized to make the local dictionaries more suitable for local data. Since this philosophy is somewhat different from what has been reported in literature, here we focus on the comparison of the proposed approach against the centralized algorithm. Due to the previous considerations, a fair comparison to other distributed methods is not immediate and will be the subject of future work.

Two scenarios are examined, namely, (a) the case where each node has different data (particularly the images shown in Fig. 1) and (b) the case where all nodes have the same image (Lena). 200 iterations were performed in both of the examined cases. We have used $\lambda = 0.9$, and the number of POCS iterations I was determined using a termination criterion that required for the maximum Frobenious norm between the dictionaries of consecutive nodes in a POCS cycle to be less that 0.0001.

In Fig. 2 we demonstrate the results for scenario (a), i.e., when the nodes have different data. It is evident that, since the nodes have different data, the dictionaries computed using only local data cannot describe the data of all nodes as efficient as the dictionary computed in a centralized fashion with all the data available. Node 2, utilizing the "crowd" image from Fig. 1, computes a "richer" dictionary, able to represent all data better than those computed by other nodes. In the same figure we show the performance of the proposed distributed dictionary learning algorithm. All nodes achieve exactly the same performance (i.e., consensus is achieved), thus, we plot only one curve for the proposed algorithm. We note that this performance is significantly better than what the nodes achieve if they work alone. However, the performance of the distributed algorithm is not that close to that of the centralized algorithm, in this setting. Finally, in Fig. 3 we demonstrate the results for scenario (b), i.e., when the nodes have the same data. It is easy to note that all schemes achieve the performance of the centralized algorithm in this setting.

V. Conclusions

In this work, a novel distributed algorithm for dictionary learning was presented. The problem of enforcing consensus among the nodes was formulated as a convex feasibility problem, and an algorithm that utilizes projections onto convex sets was proposed to yield a solution. An efficient method to compute the projection operations was also derived. Furthermore, a heuristic modification of the method was proposed that significantly accelerates the reach of a steady state. In particular, we proposed the utilization of time varying convex sets that are updated iteratively. Simulation results were conducted to demonstrate the effectiveness of the proposed algorithm, in two scenarios where the involved nodes seek for a common dictionary having (a) completely different data and (b) the same data.

References

[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[2] S. Theodoridis, Y. Kopsinis, and K. Slavakis, "Sparsity-aware learning and compressed sensing: An overview," in *Academic Press Library in Signal Processing: Singal Processing Theory and Machine Learning*, R. Chellappa and S. Theodoridis, Eds. Academic Press, 2013, ch. 23.

[3] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, 2015.

[4] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, March 2011.

[5] A. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[6] V. Kekatos, E. Vlahos, D. Ampeliotis, G. Giannakis, and K. Berberidis, "A decentralized approach to generalized power system state estimation," in *IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2013, pp. 77–80.

[7] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2013, pp. 133–136.

[8] J. Chen, Z. Towfic, and A. Sayed, "Online dictionary learning over distributed models," in *EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3874–3878.

[9] ——, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, Feb 2015.

[10] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2528–2541, June 2014.

[11] H. Raja and W. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2013, pp. 1474–1481.

[12] ——, "Cloud K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 173–188, Jan 2015.

[13] H. Raja and W. U. Bajwa, "A convergence analysis of distributed dictionary learning based on the k-svd algorithm," in *IEEE Int. Symposium on Information Theory (ISIT)*, June 2015, pp. 2186–2190.

[14] H. T. Wai, T. H. Chang, and A. Scaglione, "A consensus-based decentralized algorithm for non-convex optimization with application to dictionary learning," in *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 3546–3550.

[15] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, "An online algorithm for distributed dictionary learning," in *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 3292–3296.

[16] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "D4L: Decentralized dynamic discriminative dictionary learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 2966–2973.

[17] S. Kong, S. Punyasena, and C. C. Fowlkes, "Spatially aware dictionary learning and coding for fossil pollen identification," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1605.00775

[18] D. Ampeliotis, C. Mavrokefalidis, K. Berberidis, and S. Theodoridis, "Adapt-align-combine for diffusion-based distributed dictionary learning," in *European Signal Processing Conference (EUSIPCO)*, 2016.

[19] L. Gubin, B. Polyak, and E. Raik, "The method of projections for finding the common point of convex sets," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 6, pp. 1–24, 1967.

[20] Y. Censor and S. A. Zenios, *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press on Demand, 1997.

[21] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, Sept 2006.

[22] D. Ampeliotis and K. Berberidis, "Sorted order-k voronoi diagrams for model-independent source localization in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 426–437, Jan 2010.

[23] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[24] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[25] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *IEEE Transactions on signal processing*, vol. 45, no. 3, pp. 600–616, 1997.

[26] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.

[27] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.