Distributed Information-Based Clustering of Heterogeneous Sensor Data

Jia Chen and Ioannis D. Schizas¹

*University of Texas at Arlington, Department of Electrical Engineering
416 Yates Street, Arlington, TX 76010, USA*

Abstract

Heterogeneous sensing systems, consisting of sensors with different sensing capabilities, offer flexibility and provide multiple views of the sensed field by acquiring different types of measurements. The acquired sensor measurements are affected by different and unknown in number phenomena/sources of interest. To this end, a novel canonical correlation analysis (CCA) framework equipped with norm-one and norm-two regularization terms is designed to cluster the sensor data based on their information content. Block coordinate descent (BCD) is combined with the alternating direction method of multipliers (ADMM) framework to derive a centralized algorithm tackling the novel regularized CCA framework. Further, splitting of the regularized CCA into localized minimization subtasks across sensors enables distributed clustering of heterogeneous data based on their information content. Numerical tests demonstrate that the novel framework can achieve higher probability of correct clustering than existing alternatives.

Keywords: Canonical correlation analysis, heterogeneous sensor network, clustering, regularization

Email address: jia.chen81@mavs.uta.edu, schizas@uta.edu (Jia Chen and Ioannis D. Schizas)

¹Work in this paper is supported by the NSF Grant 1509780 and University of Texas at Arlington.

1. Introduction

The utilization of sensors with different types of sensing capabilities in heterogeneous sensor networks provide different ‘views’ of the sensed field by acquiring different types of measurements. However, the sensed data oftentimes are collected in challenging environments, whose statistical structure is not known and maybe dynamically changing with time [29]. The acquired sensor measurements are affected by different and unknown in number phenomena of interest such as thermal and/or pollution sources [30, 23, 26], while other sensor data may just contain noise, e.g., a vehicle could be sensed by both a thermal sensor due to the heat it produces, as well as a carbon monoxide sensor due to fuel emissions. Thus, sensor measurements can be clustered into different groups each of which will mainly contain information about the present field sources. Before applying any data processing task such as estimation and detection [20] it is essential to develop techniques to identify and separate these unknown groups of sensor measurements that have the same information content. Such schemes will prevent mixing sensor measurements with irrelevant information content before applying any statistical inference task, while further it will identify regions of interest in the sensed field, such as pollution sources [23, 26], by interpreting the acquired data.

The focus is to match different types of sensor measurement groups based on their information source content. However, this is challenging due to the absence of sensor localization information due to the cost and energy considerations imposed by GPS equipment, see e.g., [12]. To this end, we develop here an algorithmic framework that has the ability to both identify, cluster and match different types of sensor data based on their information content. Clustering and matching of different types of sensor measurements is extremely useful to learn the sensed field and categorize the data based on the information they contain.

Field sources in practice are localized and affect a small portion of the acquired sensor measurements. Further, the sensor measurements containing information about the same underlying source are statistically correlated irrespective of the data model and the type of sensor they are acquired from. We exploit such a property by utilizing canonical correlation analysis (CCA) [5, 16], combined alongside with sparsity-

inducing regularization techniques, see e.g., [34, 40] to obtain a framework that can extract correlated sensor measurements, group them and match them based on their information content.

CCA is widely used in data analysis to extract the correlated components among two distinct data sets [5, 16]. CCA-based methods provide attractive solutions to blind source separation [10], detection of diseases[14], and genomic data integration [24] to name a few. A number of related ℓ_1 -norm regularized CCA methods have been proposed in [17, 28, 38, 37] which seek to maximize the correlation between two data sets, while performing variable selection. These schemes are challenged by the facts that either i) no optimization criterion is specified; or ii) prior information is required on the sparsity structure of the canonical loadings; or iii) only one pair of canonical variables is searched for. Besides the ℓ_1 -norm regularized CCA, mixed ℓ_1/ℓ_2 norm penalty sparse CCA or weighted fusion penalized CCA have been studied in [37, 21].

Inspired by related sparse decomposition approaches [31, 41], along with the aforementioned existing sparse CCA alternatives, we put forth a novel ℓ_1 -norm and ℓ_2 -norm regularized CCA to cluster sensor measurements into different informative groups and find the unknown number of sources. Related work relying on ℓ_1 -norm regularized CCA for homogeneous sensor networks can be found in [11]. Different from the ℓ_1/ℓ_2 mixed norm CCA formulation in [37] in which each canonical loading is either penalized by ℓ_2 or ℓ_1 -norm, here the proposed regularized CCA framework imposes both group sparsity and entry-wise sparsity into each canonical loading by proper ℓ_2 and ℓ_1 -norm regularization terms. Compared to the aforementioned sparse CCA approaches, our proposed regularized CCA has the following advantages: a) treats both the centralized and distributed cases, so it can be implemented in networks of spatially scattered sensors as well as networks with a fusion center; b) the related sparsity-controlling coefficients are selected through a computationally efficient heuristic way; c) a specific minimization criterion is derived in our regularized CCA; and d) heterogeneous data sets from two different modalities are considered. Centralized CCA approaches that work with biomedical data of different modalities can be found in [14, 33].

Various approaches have been put forth to solve the problem of clustering data into different groups which share similar properties. One of the most common algorithms

for data clustering is K-means [22]. The task for K-means is to find the optimal centroid points representing the clusters, and the idea is to allocate every data vector to the cluster which has the most similar centroid with respect to a predefined distance metric. The limitation in K-means exists in the fact that the number of clusters should be given, aiming at which a novel intelligent K-means is proposed in [13], which extracts *anomalous patterns* from the data one-by-one to estimate the number of clusters. The existing state-of-the art clustering algorithms [39, 3, 18, 19] often implicitly assume knowledge of the cluster shapes or the multiple cluster configurations which are based on the availability of proper similarity measures. Thus, the challenge of applying these clustering approaches in our setting lies on the fact that the similarity between data entries containing the information of the same source is unknown due to the different modalities of data sets and the unavailability of the underlying data models.

This paper is organized as follows. The heterogeneous sensor network and corresponding bimodal data setting is introduced in Sec. 2. In Sec. 3, the task of grouping sensor measurements according to their source information content for an unknown number of sources, is transformed into a pertinent regularized CCA (RCCA) framework. The associated centralized RCCA (Sec. 4) and distributed RCCA (Sec. 5) are derived relying on block coordinate descent techniques [6] and the alternating direction method of multipliers, see e.g., [7, 9] which results a distributed algorithm. A novel scheme to select the sparsity-controlling coefficients is put forth in Sec. 6. Finally, extensive numerical tests demonstrate the performance of our proposed RCCA algorithm while they corroborate the better performance of the novel scheme than existing alternatives.

2. Heterogeneous Sensor Setting

We consider a heterogeneous network of sensors, where two types of sensing nodes are deployed in an area of interest. For instance, in environmental monitoring, or pollution detection, the first type of sensors could sense temperature across time, while the second type of sensors could be sensing CO (or CO₂) levels [23, 26]. In the monitored field there are p_1 sensors of the first type, [see red dots on Fig.1 (bottom)], and p_2 sensors of the second type [see blue boxes in Fig. 1 (bottom)]. In the field there are M spatially uncorrelated sources, where the number of sources M is *unknown*. The m th source whose intensity is denoted by random variable $s_m(t)$, for $m = 1, \dots, M$, is

located at position $\mathbf{l}_m \in \mathbb{R}^2$, while t denotes the time index and $t = 0, 1, \dots, N - 1$.

Let $\mathfrak{A}$ and $\mathfrak{B}$ denote the sets of the first and second type of sensors, with cardinality

95 $|\mathfrak{A}| = p_1$ and $|\mathfrak{B}| = p_2$. Further, it is assumed that the sources are not overlapping in

the sense that each sensor is observing at most one field source. Further, let $\mathbf{l}_j^{\mathfrak{A}} \in \mathbb{R}^2$

denote the location for sensor $j \in \mathfrak{A}$, and $\mathbf{l}_i^{\mathfrak{B}} \in \mathbb{R}^2$ the position of sensor $i \in \mathfrak{B}$.

Diffusion fields are considered here, which will be modeled using the Green's function

[4]. The field $f_1(\mathbf{r}, t)$ denotes the strength value of the field measured by sensors in $\mathfrak{A}$,

100 and $f_2(\mathbf{r}, t)$ the strength of the field measured by sensors in $\mathfrak{B}$, at position $\mathbf{r} \in \mathbb{R}^2$ and

time t . Diffusion fields are pertinent for modeling how heat or chemical substances are

diffusing in space and time [4]. From the theory of Green's function, see e.g., [4], the

two diffusion fields are modeled as

$$f_1(\mathbf{r}, t) = (g_1 * S_{d,1})(\mathbf{r}, t), \quad f_2(\mathbf{r}, t) = (g_2 * S_{d,2})(\mathbf{r}, t) \quad (1)$$

where $*$ is the convolution operator, while the functions

$$g_1(\mathbf{r}, t) = \frac{1}{4\pi\gamma_1 t} e^{-\frac{\|\mathbf{r}\|_2^2}{4t\gamma_1}} H(t), \quad g_2(\mathbf{r}, t) = \frac{1}{4\pi\gamma_2 t} e^{-\frac{\|\mathbf{r}\|_2^2}{4t\gamma_2}} H(t) \quad (2)$$

105 indicate the Green's functions of the two-dimensional diffusion field, and $H(t)$ is the

unit step function which is equal to unity for $t \geq 0$ and zero otherwise. Moreover γ_1

and γ_2 are the diffusivities of the medium through which the field propagates, see e.g.,

[4]. Further, $S_{d,1} \in \mathbb{R}^2$ and $S_{d,2} \in \mathbb{R}^2$ represent the sources' distribution at position $\mathbf{r}$

and time t , and are given as:

$$S_{d,1}(\mathbf{r}, t) = \sum_{m=1}^M \chi_{1,m}(s_m(t)) \cdot \delta((\mathbf{r} - \mathbf{l}_m), t) \quad (3)$$

$$S_{d,2}(\mathbf{r}, t) = \sum_{m=1}^M \chi_{2,m}(s_m(t)) \cdot \delta((\mathbf{r} - \mathbf{l}_m), t) \quad (4)$$

110 where $\delta(\cdot)$ is the Dirac delta function [with a spike at $\mathbf{r} = \mathbf{l}_m$ and $t = 0$], and $\chi_{1,m}$ and

$\chi_{2,m}$ are arbitrary scalar nonlinear mappings controlling the intensity of the different

quantities being measured.

Sensor $j \in \mathfrak{A}$ and $i \in \mathfrak{B}$, located at *unknown* positions $\mathbf{l}_j^{\mathfrak{A}}$ and $\mathbf{l}_i^{\mathfrak{B}}$, respectively,

acquire at time instant t scalar measurements $x_j(t)$ and $y_i(t)$, which constitute noisy

115 measurements of the sensed fields, i.e.,

$$x_j(t) = f_1(\mathbf{l}_j^{\mathfrak{A}}, t) + w_j(t), \quad \text{for } j = 1, 2, \dots, p_1 \quad (5)$$

$$y_i(t) = f_2(\mathbf{l}_i^{\mathfrak{B}}, t) + w_i(t), \quad \text{for } i = 1, 2, \dots, p_2 \quad (6)$$

where $w_j(t)$ and $w_i(t)$ correspond to zero-mean white sensing noise which is independent

of the source intensity signals $\chi_{1,m}(s_m(t))$, $\chi_{2,m}(s_m(t))$ for $m = 1, \dots, M$. Also,

$w_j(t)$ and $w_i(t)$ are independent for $i \neq j$. Further, let

$$\mathbf{x}_t := [x_1(t), x_2(t), \dots, x_{p_1}(t)]^T, \mathbf{y}_t := [y_1(t), y_2(t), \dots, y_{p_2}(t)]^T \quad (7)$$

denote the $p_1 \times 1$ and $p_2 \times 1$ vectors that contain the measurements acquired by sensors $\mathfrak{A}$ and $\mathfrak{B}$, respectively, at time instant t . Vectors $\mathbf{x}_t$, and $\mathbf{y}_t$ essentially entail spatial samples of the different views of the monitored area, e.g., a temperature field and a CO field in Fig. 1 (top) and Fig. 1 (center). Due to cost considerations and limited power budget, sensors may not have GPS localization capabilities. As a result the data in $\mathbf{x}_t$ and $\mathbf{y}_t$ do not have a location signature. Thus, when gathering and processing the data, it is not known which sensor measurements in $\mathbf{x}_t$ and $\mathbf{y}_t$ are affected by the same field sources since no location information is available.

Note that the source signals decay exponentially with the propagation distance. A sensor is considered to observe a source if it is located within the area where 90% of the source's signal energy is contained, otherwise it does assumed not to contain any information about that source. Thus, a threshold corresponding to 90% of a source's signal energy is used to determine which source every sensor observes. The field sources [see e.g., Fig. 1 (top, center)] are quite localized affecting a limited number of sensors in the monitored field. This further implies that different entries of $\mathbf{x}_t$ and $\mathbf{y}_t$ are mainly affected by different field sources. Denote $\mathcal{S}_x^m$ and $\mathcal{S}_y^m$ as the set of indices of entries in $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively, that are mainly affected by the dominant source, saying the m th source for $m = 1, \dots, M$, while $\mathcal{S}^0$ is the set of indices in $\mathbf{x}_t$ and $\mathbf{y}_t$ that sense just noise. For example, in Fig. 1 (bottom), there are $p_1 = 30$ sensors of the first type in $\mathfrak{A}$ (red dots), and $p_2 = 30$ sensors of the second type in $\mathfrak{B}$ (blue squares). $M = 2$ sources are present forming the two diffusion fields in Fig. 1 (top), and Fig. 1 (center). In Fig. 1 (bottom) the blue and red ellipsoids surround the area in which 90% of the energy of the source field sources is contained. Thus, the red dots surrounded by the red ellipsoid correspond to sensors in $\mathfrak{A}$ acquiring information-bearing measurements about the two sources in the field in Fig. 1 (top), whereas the blue squares surrounded by the blue ellipsoids correspond to sensors in $\mathfrak{B}$ acquiring information about the two sources in the sensed field in Fig. 1 (center). Specifically, sensors $\mathcal{S}_x^1 = \{4, 23\}$ in $\mathfrak{A}$, and $\mathcal{S}_y^1 = \{2, 8, 10, 18, 24, 30\}$ in $\mathfrak{B}$ sense source $s_1(t)$, while sensors $\mathcal{S}_x^2 = \{2\}$ in $\mathfrak{A}$, and $\mathcal{S}_y^2 = \{5, 29\}$ in $\mathfrak{B}$ sense source $s_2(t)$, while the rest of sensors just acquire

noise. It is of interest to associate measurements acquired from the two different types of sensors, which have the entries in $\mathbf{x}_t$ and $\mathbf{y}_t$, with the present field sources. This task is essential to avoid mixing measurements that correspond to different sensed sources that may confuse subsequent estimation/detection procedures. Thus, a framework for matching different types of measurements in $\mathbf{x}_t$ and $\mathbf{y}_t$, and clustering them into groups based on their information content is developed here.

A fundamental property that can be utilized here is the fact that the entries in $\mathbf{x}_t$ and $\mathbf{y}_t$ tend to be correlated when they sense the same source irrespective of the fact that they measure different quantities. Canonical Correlation Analysis (CCA) is a widely used method to extract shared features present in two data sequences, see e.g., [5]. Given the data pairs $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=0}^{N-1}$, traditional CCA is used to *linearly* extract the correlated common features hidden in them, see e.g., [5, 17], [16]. Given the number of source M , the task can be reformulated as finding matrices $\mathbf{D} \in \mathbb{R}^{M \times p_1}$ and $\mathbf{E} \in \mathbb{R}^{M \times p_2}$ with $M \leq \min(p_1, p_2)$ that minimize

$$\begin{aligned} (\hat{\mathbf{D}}, \hat{\mathbf{E}}) = \arg \min_{\mathbf{D}, \mathbf{E}} N^{-1} \sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}_t - \mathbf{D}\mathbf{x}_t - \hat{\boldsymbol{\mu}}\|_2^2 \\ \text{s.to } \mathbf{D}\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T = \mathbf{I}, \mathbf{E}\hat{\boldsymbol{\Sigma}}_y\mathbf{E}^T = \mathbf{I} \end{aligned} \quad (8)$$

where $\|\cdot\|_2$ is the Euclidean norm and $\hat{\boldsymbol{\mu}} := \mathbf{E}\hat{\boldsymbol{\mu}}_y - \mathbf{D}\hat{\boldsymbol{\mu}}_x$ in which $\hat{\boldsymbol{\mu}}_x := N^{-1} \sum_{t=0}^{N-1} \mathbf{x}_t$ (similarly for $\hat{\boldsymbol{\mu}}_y$) denote the sample-average estimates for the expectation of $\mathbf{x}_t$ (or $\mathbf{y}_t$), respectively. Matrices $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ denote the sample-average estimates for covariance matrices of measurements $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively, found as $\hat{\boldsymbol{\Sigma}}_x := N^{-1} \sum_{t=0}^{N-1} (\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x)(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x)^T$ and $\hat{\boldsymbol{\Sigma}}_y := N^{-1} \sum_{t=0}^{N-1} (\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y)(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y)^T$. A closed-form solution for $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ can be found in e.g., [5].

The reason for imposing *canonical variates* $\hat{\mathbf{D}}\mathbf{x}_t$ and $\hat{\mathbf{E}}\mathbf{y}_t$ to be as close as possible is the fact that each entry of them is trying to uncover the shared sources sensed by $\mathbf{x}_t$ and $\mathbf{y}_t$. Given the source number M , traditional CCA is capable of estimating the sources that are present in both $\mathbf{x}_t$ and $\mathbf{y}_t$, however it is not able to identify which entries of $\mathbf{x}_t$ and $\mathbf{y}_t$ are affected by the same dominant source. A necessary ingredient is the proper introduction of zeros (sparsity) in the CCA matrices $\mathbf{D}$ and $\mathbf{E}$, such that the nonzero entries in each row of $\mathbf{D}$ and $\mathbf{E}$ will point to these entries in $\mathbf{x}_t$ and $\mathbf{y}_t$ (as well as the corresponding sensors) that are effected by the same source.

For instance with reference to Fig. 1 (bottom), there should be one row of $\mathbf{D}$ and $\mathbf{E}$ with nonzero values at entries with indices $\{4, 23\}$ and $\{2, 8, 10, 18, 24, 30\}$, respectively, corresponding to sensors that sense source $s_1(t)$; while the other row of $\mathbf{D}$ and $\mathbf{E}$ should have nonzero values at entries with indices $\{2\}$ and $\{5, 29\}$ corresponding to sensors that sense source $s_2(t)$, while the rest of the entries should be zero. To properly induce sparsity in $\mathbf{D}$ and $\mathbf{E}$, while coping with an unknown number of sources M , and motivated by the sparse techniques in [34, 36, 41, 31, 40], the standard CCA formulation will be enhanced with norm-one, and norm-two regularization. Further, a proper centralized and distributed algorithmic framework will be put forth to facilitate the enhanced regularized CCA formulation.

Notation: $\mathbf{A}_{\rho\cdot}$ denotes the ρ th row of matrix $\mathbf{A}$; $\mathbf{A}(\alpha, \beta)$ corresponds to the (α, β) entry of matrix $\mathbf{A}$; $\mathbf{I}$ is the identity matrix of size $q \times q$; $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_F$ denote norm-one, norm-two and the Frobenius norm operators.

3. Regularized Canonical Correlation

In order to induce sparsity in matrices $\mathbf{D}$ and $\mathbf{E}$, and subsequently identify different subsets of entries in $\mathbf{x}_t$ and $\mathbf{y}_t$ that are mainly affected by the same dominant field source, norm-one regularization will be induced in the standard CCA formulation, motivated by the work in [34, 36, 41, 31]. Further, sensors are unaware of the number of sources M that may be present. To cope with this limitation, not handled in standard CCA, the idea is to set the number of rows q in $\mathbf{D}$ and $\mathbf{E}$ such that it is larger than the actual number of sources M . Such an upper bound on M can be easily set by selecting q sufficiently high. Then, norm-two regularization can be incorporated in (8), e.g., see [2, 40] to zero-out the extra rows by inducing ‘group’ sparsity. In detail, the proposed regularized CCA framework equipped with ℓ_1 - and ℓ_2 -regularization takes the form:

$$\begin{aligned} (\hat{\mathbf{D}}, \hat{\mathbf{E}}) = \arg \min_{\mathbf{D}, \mathbf{E}} & N^{-1} \sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}_t - \mathbf{D}\mathbf{x}_t - \hat{\boldsymbol{\mu}}\|_2^2 + v \|\mathbf{E}\hat{\boldsymbol{\Sigma}}_y \mathbf{E}^T - \mathbf{I}\|_F^2 \\ & + \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x \mathbf{D}^T - \mathbf{I}\|_F^2 + \sum_{\rho=1}^q \lambda_{E,\rho} \|\mathbf{E}_{\rho\cdot}\|_1 + \sum_{\rho=1}^q \lambda_{D,\rho} \|\mathbf{D}_{\rho\cdot}\|_1 \\ & + \phi_D \sum_{\rho=1}^q \|\mathbf{D}_{\rho\cdot}\|_2 + \phi_E \sum_{\rho=1}^q \|\mathbf{E}_{\rho\cdot}\|_2 \end{aligned} \quad (9)$$

where the second and third terms account for the constraints in (8), and ε and v are positive constants. Further, the positive scalars $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ are controlling the sparsity (number of zeroes) in the ρ th row of $\mathbf{D}$ and $\mathbf{E}$, respectively. Further, the nonnegative

coefficients ϕ_D and ϕ_E multiplying the last norm-two terms in (9) control the number of nonzero rows in $\mathbf{D}$ and $\mathbf{E}$. The last two terms in (9) introduce group sparsity, e.g., [8][40], and can zero out an entire rows in $\mathbf{D}$ and $\mathbf{E}$ that are not necessary, especially if $q > M$. Thus, proper selection of ϕ_D and ϕ_E can zero out the $(q - M)$ redundant rows in $\mathbf{D}$ and $\mathbf{E}$, while estimating the ground truth number of sources.

The cost function in (9) will be split into $2q$ subproblems each of which subproblems involves minimization w.r.t. a single row of $\mathbf{D}$, say $\mathbf{D}_{\rho:}$ (or $\mathbf{E}$, say $\mathbf{E}_{\rho:}$) while fixing the remaining rows of $\mathbf{D}$ and $\mathbf{E}$ to their most up-to-date values. To tackle the subtasks, the alternating direction method of multipliers (ADMM), see e.g., [9][1] will be applied. ADMM is an efficient method for sparsity-inducing minimization problems, e.g., see [11]. Both centralized and distributed algorithms will be derived to tackle the problem in (9), and perform the desired clustering tasks in the heterogeneous sensor measurements. [The novel algorithms are able to work for more than one sources in the field as it will be demonstrated in Sec. 7.](#)

4. Centralized Optimization of RCCA

A centralized setting is considered first where a fusion center (FC) collects the measurements acquired across all the sensors. Notice that the cost in (9) is a nonconvex function w.r.t. $\mathbf{D}$ and $\mathbf{E}$. Block coordinate descent [6][35] can surpass this challenge by iteratively solving the cost in (9) w.r.t. one matrix, say $\mathbf{D}$ (or $\mathbf{E}$) while making the other matrix $\mathbf{E}$ (or $\mathbf{D}$) fixed. Specifically, the minimization task wrt $\mathbf{D}$ can be rewritten as

$$\hat{\mathbf{D}} = \arg \min_{\mathbf{D}} N^{-1} \sum_{t=0}^{N-1} \|\hat{\mathbf{E}}\mathbf{y}_t - \mathbf{D}\mathbf{x}_t - \hat{\boldsymbol{\mu}}\|_2^2 \quad (10)$$

$$+ \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T - \mathbf{I}\|_F^2 + \sum_{\rho=1}^q \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 + \phi_D \sum_{\rho=1}^q \|\mathbf{D}_{\rho:}\|_2.$$

In order to solve the problem in (10), the proposed ADMM solver will focus on one row of $\mathbf{D}$, e.g., $\mathbf{D}_{\rho:}$, while fixing the rest of the rows. Thus, $\mathbf{D}_{\rho:}$ is obtained as

$$\begin{aligned} \hat{\mathbf{D}}_{\rho:} = \arg \min_{\mathbf{D}_{\rho:}} & N^{-1} \|\hat{\mathbf{E}}_{\rho:} \mathbf{Y} - \mathbf{D}_{\rho:} \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 \\ & + \phi_D \|\mathbf{D}_{\rho:}\|_2 + \varepsilon \|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:} \hat{\boldsymbol{\Sigma}}_x \mathbf{D}_{\rho:}^T\|_2^2 \end{aligned} \quad (11)$$

in which, $\mathbf{X} := [\mathbf{x}_0 - \hat{\boldsymbol{\mu}}_x, \mathbf{x}_1 - \hat{\boldsymbol{\mu}}_x, \dots, \mathbf{x}_{N-1} - \hat{\boldsymbol{\mu}}_x] \in \mathbb{R}^{p_1 \times N}$ and $\mathbf{Y} := [\mathbf{y}_0 - \hat{\boldsymbol{\mu}}_y, \mathbf{y}_1 - \hat{\boldsymbol{\mu}}_y, \dots, \mathbf{y}_{N-1} - \hat{\boldsymbol{\mu}}_y] \in \mathbb{R}^{p_2 \times N}$. Let τ denote the block coordinate cycle index. Note that the last term in (11) produces fourth-order polynomials in the cost function. To simplify the process of solving (11) w.r.t. $\mathbf{D}_{\rho:}$, we fix the second $\mathbf{D}$ in the last term of the cost in (11) to the most recent update $\hat{\mathbf{D}}^{\tau-1}$ during the τ th coordinate descent cycle

(similar when updating $\mathbf{E}$). Given the estimates $\hat{\mathbf{D}}^{\tau-1}$ and $\hat{\mathbf{E}}^{\tau-1}$ in the beginning of coordinate cycle τ , the minimization problem in (11) can be rewritten as

$$\begin{aligned} \hat{\mathbf{D}}_{\rho:}^{\tau} = \arg \min_{\mathbf{D}_{\rho:}} & N^{-1} \|\hat{\mathbf{E}}_{\rho:}^{\tau-1} \mathbf{Y} - \mathbf{D}_{\rho:} \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 \\ & + \phi_D \|\mathbf{D}_{\rho:}\|_2 + \varepsilon \|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:} \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 \end{aligned} \quad (12)$$

To tackle the minimization problem in (12) we will employ ADMM. To this end, we introduce an auxiliary vector $\mathbf{b}_{\rho} \in \mathbb{R}^{1 \times p_1}$ and reformulate (12) as the following equivalent constrained minimization problem

$$\begin{aligned} (\hat{\mathbf{D}}_{\rho:}^{\tau}, \hat{\mathbf{b}}_{\rho}^{\tau}) = \arg \min_{\mathbf{D}_{\rho:}, \mathbf{b}_{\rho}} & N^{-1} \|\hat{\mathbf{E}}_{\rho:}^{\tau-1} \mathbf{Y} - \mathbf{D}_{\rho:} \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 + \phi_D \|\mathbf{b}_{\rho}\|_2 \\ & + \varepsilon \|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:} \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T\|_2^2, \text{ subject to, } \mathbf{b}_{\rho} = \mathbf{D}_{\rho:} \end{aligned} \quad (13)$$

Notice that (13) and (12) are equivalent in the sense that they have the same optimal solution. A necessary step in applying ADMM in (13) is the formation of augmented

Lagrangian function

$$\begin{aligned} \mathcal{L}^{\tau}(\mathbf{D}_{\rho:}, \mathbf{b}_{\rho}, \mathbf{p}_{\rho}) = & N^{-1} \|\hat{\mathbf{E}}_{\rho:}^{\tau-1} \mathbf{Y} - \mathbf{D}_{\rho:} \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 + \phi_D \|\mathbf{b}_{\rho}\|_2 \\ & + \varepsilon \|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:} \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 + (\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}) \mathbf{p}_{\rho} + \frac{c}{2} \|\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}\|_2^2 \end{aligned} \quad (14)$$

where $\mathbf{p}_{\rho} \in \mathbb{R}^{p_1 \times 1}$ denotes the Lagrange multiplier accounting for the constraint $\mathbf{b}_{\rho} = \mathbf{D}_{\rho:}$, while $c > 0$ is a penalty coefficient ensuring (14) is strictly convex w.r.t. $\mathbf{D}_{\rho:}$ and $\mathbf{b}_{\rho}$. ADMM is an iterative method, see e.g., [9], which involves the following three steps (details follow later):

Step 1) Minimize the augmented Lagrangian in (14) w.r.t. $\mathbf{D}_{\rho:}$, while fixing $\mathbf{b}_{\rho}$, and $\mathbf{p}_{\rho}$ to their most recent updates, to obtain update $\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa}$ which denotes the ADMM update for $\mathbf{D}_{\rho:}$ during coordinate cycle τ , while the index κ denotes the ADMM iteration index nested inside coordinate cycle τ . Here K ADMM iterations will be applied per coordinate cycle τ , where K is a user-defined parameter.

Step 2) Minimize the augmented Lagrangian in (14) w.r.t. $\mathbf{b}_{\rho}$, while fixing $\mathbf{D}_{\rho:}$, $\mathbf{p}_{\rho}$ to the ADMM updates $\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa}$, $\mathbf{p}_{\rho}^{\tau,\kappa-1}$ to obtain update $\hat{\mathbf{b}}_{\rho}^{\tau,\kappa}$.

Step 3) Update the Lagrange multiplier using gradient ascent iterations. Let $\mathbf{p}_{\rho}^{\tau,\kappa}$ denote the ADMM update for $\mathbf{p}_{\rho}$ during cycle τ , and the κ th ADMM iteration. These three steps are applied for K ADMM iterations, i.e., $\kappa = 1 \dots K$, nested inside cycle τ .

The first step in ADMM, during coordinate descent cycle τ , involves minimization of (14) w.r.t. $\mathbf{D}_{\rho:}$, while fixing the remaining optimization variables to their most up-

to-date values $\hat{\mathbf{E}}^{\tau-1}$, $\mathbf{b}_\rho^{\tau,\kappa-1}$, and $\mathbf{p}_\rho^{\tau,\kappa-1}$. Specifically, $\hat{\mathbf{D}}_\rho^{\tau,\kappa}$ can be obtained as

$$\begin{aligned} \hat{\mathbf{D}}_\rho^{\tau,\kappa} = \arg \min_{\mathbf{D}_\rho} & N^{-1} \|\hat{\mathbf{E}}_\rho^{\tau-1} \mathbf{Y} - \mathbf{D}_\rho \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_\rho\|_1 \\ & + \varepsilon \|\mathbf{I}_\rho - \mathbf{D}_\rho \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 + \frac{c}{2} \|\mathbf{D}_\rho - \mathbf{b}_\rho^{\tau,\kappa-1}\|_2^2 + \mathbf{D}_\rho \mathbf{p}_\rho^{\tau,\kappa-1}. \end{aligned} \quad (15)$$

The second step in ADMM involves minimizing (14) w.r.t. $\mathbf{b}_\rho$, while using the most recent updates $\hat{\mathbf{D}}_\rho^{\tau,\kappa}$ and $\mathbf{p}_\rho^{\tau,\kappa-1}$ to obtain

$$\mathbf{b}_\rho^{\tau,\kappa} = \arg \min_{\mathbf{b}_\rho} \phi_D \|\mathbf{b}_\rho\|_2 + \frac{c}{2} \|\hat{\mathbf{D}}_\rho^{\tau,\kappa} - \mathbf{b}_\rho\|_2^2 - \mathbf{b}_\rho \mathbf{p}_\rho^{\tau,\kappa-1} \quad (16)$$

The third step involves updating the Lagrange multiplier vector

$$\mathbf{p}_\rho^{\tau,\kappa} = \mathbf{p}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{D}}_\rho^{\tau,\kappa} - \mathbf{b}_\rho^{\tau,\kappa})^T. \quad (17)$$

Using the convergence claims for ADMM in [9], it can be readily shown that the iterates $\hat{\mathbf{D}}_\rho^{\tau,k}$ converge to $\lim_{k \rightarrow \infty} \hat{\mathbf{D}}_\rho^{\tau,k} = \hat{\mathbf{D}}_\rho^\tau$, where $\hat{\mathbf{D}}_\rho^\tau$ for $\rho = 1, \dots, q$ is the minimizer of (12) as the number of ADMM iterations goes to ∞ .

To tackle the minimization problem in (15), we resort to a coordinate descent technique, where we minimize w.r.t. an entry of $\mathbf{D}_\rho$, say $\mathbf{D}(\rho, \beta)$, while fixing the rest of the entries in $\mathbf{D}_\rho$ to their most up-to-date values. Thus, p_1 subproblems are obtained from (15) each giving the update

$$\begin{aligned} \hat{\mathbf{D}}^{\tau,\kappa}(\rho, \beta) = \arg \min_d & \|\zeta_{\rho,\beta}^\tau - d \cdot \mathbf{h}_{\rho,\beta}^\tau\|_2^2 + \lambda_{D,\rho} |d| \\ & + 0.5c(d - \mathbf{b}_\rho^{\tau,\kappa-1}(\beta))^2 + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(\beta) \end{aligned} \quad (18)$$

for $\beta = 1, 2, \dots, p_1$, where $\zeta_{\rho,\beta}^\tau := [\zeta_{\rho,\beta}^{1,\tau}, \zeta_{\rho,\beta}^{2,\tau}]^T$ and $\mathbf{h}_{\rho,\beta}^\tau := [N^{-0.5} \mathbf{X}_{\beta,:}, (\hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T)_{\beta,:}]^T$,

in which

$$\zeta_{\rho,\beta}^{1,\tau} := N^{-\frac{1}{2}} [\hat{\mathbf{E}}_\rho^{\tau-1} \mathbf{Y} - \sum_{\ell=1, \ell \neq \beta}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, \ell) \mathbf{X}_{\ell,:}]$$

$$\zeta_{\rho,\beta}^{2,\tau} := \varepsilon^{0.5} [\mathbf{I}_\rho - \sum_{\ell=1, \ell \neq \beta}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, \ell) [\hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T]_{\ell,:}]$$

As it has been shown in [31, Apdx. A], the closed form solution to (18) can be expressed as

$$\hat{\mathbf{D}}^{\tau,\kappa}(\rho, \beta) = \mathbb{F}(\zeta_{\rho,\beta}^\tau, \mathbf{h}_{\rho,\beta}^\tau, \frac{c}{2}(\mathbf{b}_\rho^{\tau,\kappa-1}(\beta) - \frac{\mathbf{p}_\rho^{\tau,\kappa-1}(\beta)}{c}), \frac{c}{2}, \lambda_{D,\rho}) \quad (19)$$

$$\text{where } \mathbb{F}(\mathbf{p}_1, \mathbf{p}_2, p_3, p_4, \lambda) := \text{sgn}(\mathbf{p}_1^T \mathbf{p}_2 + p_3) \quad (20)$$

$$\times \left(\max \left(0, \left(\left| \frac{\mathbf{p}_1^T \mathbf{p}_2 + p_3}{\|\mathbf{p}_2\|_2^2 + p_4} \right| - \left(\frac{\lambda}{2(\|\mathbf{p}_2\|_2^2 + p_4)} \right) \right) \right) \right).$$

It is established in Apdx. A that the minimizer of (16) results the following update

$$\mathbf{b}_\rho^{\tau,\kappa} = c^{-1} \mathcal{S}_v(\mathbf{p}_\rho^{\tau,\kappa-1} + c \cdot \hat{\mathbf{D}}_\rho^{\tau,\kappa}, \phi_D), \text{ where } \mathcal{S}_v(\mathbf{v}, \phi) = [1 - \frac{\phi}{\|\mathbf{v}\|_2}]_+ \mathbf{v} \quad (21)$$

Next, starting from (9), we focus on updating each row of $\mathbf{E}$, e.g., $\mathbf{E}_{\rho:}$, while fixing the rest of the rows of $\mathbf{E}$ and $\mathbf{D}$. During τ th coordinate cycle, the estimate $\mathbf{E}_{\rho:}^\tau$ is obtained

$$\begin{aligned} \text{as: } \hat{\mathbf{E}}_{\rho:}^\tau &= \arg \min_{\mathbf{E}_{\rho:}} N^{-1} \|\hat{\mathbf{D}}_{\rho:}^{\tau-1} \mathbf{X} - \mathbf{E}_{\rho:} \mathbf{Y}\|_2^2 + \lambda_{E,\rho} \|\mathbf{E}_{\rho:}\|_1 \\ &\quad + \phi_E \|\mathbf{E}_{\rho:}^T\|_2 + v \|\mathbf{I}_{\rho:} - \mathbf{E}_{\rho:} \hat{\Sigma}_y (\hat{\mathbf{E}}^{\tau-1})^T\|_2^2. \end{aligned} \quad (22)$$

In order to solve the problem in (22), we follow a similar process as the one used to update $\hat{\mathbf{D}}_{\rho:}^\tau$ in (12). Then, we can obtain the updating recursions for $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j)$, $\check{\mathbf{b}}_{\rho}^{\tau,\kappa} \in \mathbb{R}^{1 \times p_2}$, and $\check{\mathbf{p}}_{\rho}^{\tau,\kappa} \in \mathbb{R}^{p_2 \times 1}$, in which the later two quantities are similar to $\mathbf{b}_{\rho}^{\tau,\kappa}$, and $\mathbf{p}_{\rho}^{\tau,\kappa}$, respectively. The update $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j)$ can be obtained as

$$\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j) = \mathbb{F}([\check{\zeta}_{\rho,j}^{1,\tau}, \check{\zeta}_{\rho,j}^{2,\tau}]^T, \check{\mathbf{h}}_{\rho}^{\tau}, \frac{c}{2}(\check{\mathbf{b}}_{\rho}^{\tau,\kappa-1}(j) - \frac{\check{\mathbf{p}}_{\rho}^{\tau,\kappa-1}(j)}{c}), \frac{c}{2}, \lambda_{E,\rho}) \quad (23)$$

for $\rho = 1, \dots, q$, and $j = 1, \dots, p_2$, while $\check{\zeta}_{\rho,j}^{1,\tau}$, $\check{\zeta}_{\rho,j}^{2,\tau}$, and $\check{\mathbf{h}}_{\rho}^{\tau}$ are obtained similarly to $\zeta_{\rho,\beta}^{1,\tau}$, $\zeta_{\rho,\beta}^{2,\tau}$, and $\mathbf{h}_{\rho}^{\tau}$, respectively, after substituting $\mathbf{X}$, $\mathbf{Y}$, $\hat{\Sigma}_x$, $\hat{\mathbf{D}}^{\tau}$, $\hat{\mathbf{D}}^{\tau-1}$, $\hat{\mathbf{E}}^{\tau-1}$, β , p_1 , and ε by $\mathbf{Y}$, $\mathbf{X}$, $\hat{\Sigma}_y$, $\hat{\mathbf{E}}^{\tau}$, $\hat{\mathbf{E}}^{\tau-1}$, $\hat{\mathbf{D}}^{\tau-1}$, j , p_2 , and v , respectively. Then, the auxiliary vector $\check{\mathbf{b}}_{\rho}^{\tau,\kappa}$ can be obtained by

$$\check{\mathbf{b}}_{\rho}^{\tau,\kappa} = c^{-1} \mathcal{S}_v(\check{\mathbf{p}}_{\rho}^{\tau,\kappa-1} + c \cdot \hat{\mathbf{E}}_{\rho:}^{\tau,\kappa}, \phi_E). \quad (24)$$

Finally, the Lagrangian multiplier $\check{\mathbf{p}}_{\rho}^{\tau,\kappa}$ can be updated as

$$\check{\mathbf{p}}_{\rho}^{\tau,\kappa} = \check{\mathbf{p}}_{\rho}^{\tau,\kappa-1} + c(\hat{\mathbf{E}}_{\rho:}^{\tau,\kappa} - \check{\mathbf{b}}_{\rho}^{\tau,\kappa})^T \quad (25)$$

The entire updating process for implementing the centralized regularized CCA (CR-CCA) algorithm is tabulated as Algorithm 1.

Notice that the updates in (19) and (23) for $\beta = 1, \dots, p_1$ and $j = 1, \dots, p_2$ can be implemented for multiple nested coordinate cycles, within coordinate cycle τ and ADMM iteration k , indicated by iteration index τ_2 , i.e., the updates would be written as $\hat{\mathbf{D}}^{\tau,k,\tau_2}(\rho, \beta)$ and $\hat{\mathbf{E}}^{\tau,k,\tau_2}(\rho, \beta)$. In Alg. 1 it was assumed that $\tau_2 = 1$, which implies that the quantities in (19) and (23) were updated for one cycle, and thus the τ_2 superscript was not included. However, notice that to solve (15) (as well as the corresponding minimization task when updating $\hat{\mathbf{E}}_{\rho:}^{\tau,k}$) the updates (19) [and correspondingly (23)] should be applied for an increasing number of nested coordinate iterations, i.e., $\tau_2 \rightarrow \infty$. The following convergence claim is shown in Apdx. B.

Proposition 1 : *Let $\mathbf{E}^*$ and $\mathbf{D}^*$ denote a stationary point of (9) for $\varepsilon = v$. When the coordinate iterations τ , ADMM iterations κ and nested coordinate iterations τ_2 go to*

infinity, the updates $\hat{\mathbf{E}}^{\tau, \kappa, \tau_2}$ and $\hat{\mathbf{D}}^{\tau, \kappa, \tau_2}$ obtained from (23) and (19) in R-CCA will satisfy that $\|\hat{\mathbf{E}}^{\tau, \kappa, \tau_2} - \mathbf{E}^*\|_F \leq \delta(\varepsilon)$ and $\|\hat{\mathbf{D}}^{\tau, \kappa, \tau_2} - \mathbf{D}^*\|_F \leq \delta(\varepsilon)$, where $\lim_{\varepsilon \rightarrow 0} \delta(\varepsilon) = 0$ and ε is the parameter used in the third term in (9).

Proposition 1 indicates that the iterative updates $\hat{\mathbf{E}}^{\tau, \kappa}(\alpha, \beta)$ and $\hat{\mathbf{D}}^{\tau, \kappa}(\alpha, \beta)$ can be arbitrarily close to a stationary point in (9) by selecting a sufficiently small ε , since $\lim_{\varepsilon \rightarrow 0} \delta(\varepsilon) = 0$.

5. Distributed Implementation

The proposed regularized (R-)CCA framework in (9) will be tackled here in a distributed fashion in a setting where sensors can only communicate with neighboring sensors located within sufficient communication range (single-hop neighbors). No central fusion center exists, and sensors need to carry out the data processing and perform the clustering tasks in the network (in-network processing). The communication graph of the sensor network is formed by all sensors in $\mathfrak{A} \cup \mathfrak{B}$, which correspond to the nodes of the graph. Two nodes are connected if and only if they are within the communication range, in which case an edge connects them in the graph. To develop a distributed algorithm, we will impose the following assumptions in the heterogeneous network topology: A1) The communication graph $\mathfrak{A}$ is connected; A2) The communication graph $\mathfrak{B}$ is connected; A3) Every sensor in $\mathfrak{A} \cup \mathfrak{B}$ has at least two neighboring sensors where one is in $\mathfrak{A}$ and the other in $\mathfrak{B}$, i.e., if $\mathcal{N}_j$ denotes the single-hop neighborhood of sensor $j \in \mathfrak{A} \cup \mathfrak{B}$, then $\exists j$ and j' , with $j \in \mathfrak{A}$ and $j' \in \mathfrak{B}$, where $\{j, j'\} \subseteq \mathcal{N}_j$. Each sensor $j \in \mathfrak{A}$ (first type of sensing sensor), is responsible for updating the j th column of $\mathbf{D}$, namely $\mathbf{d}_j \in \mathbb{R}^{q \times 1}$, while sensor $i \in \mathfrak{B}$ is responsible for updating the i th column in $\mathbf{E}$, namely $\mathbf{e}_i \in \mathbb{R}^{q \times 1}$. Note that assumptions A1), A2) and A3) imply that the communication graph $\mathfrak{A} \cup \mathfrak{B}$ is connected. The information acquired across sensors is spatially scattered, i.e., sensor $j \in \mathfrak{A}$ has available only measurements $\mathbf{x}_t(j)$ (j th entry in $\mathbf{x}_t$), while sensor $i \in \mathfrak{B}$ has available only the scalar measurement $\mathbf{y}_t(i)$ (i th entry in $\mathbf{y}_t$). Further, let $\bar{\mathbf{x}}_t(j)$ and $\bar{\mathbf{y}}_t(i)$ denote the j th and i th entry, respectively of the zero-mean translated vectors $\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x$, and $\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y$, respectively.

In order to obtain a distributed algorithm for tackling the R-CCA framework, the three ADMM steps outlined in Sec. 4 will be carried out in an in-network fashion

here. To this end, a framework to estimate the global quantities $\mathbf{D}(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x)$ and $\mathbf{E}(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y)$, which contain information from all sensors, needs to be obtained. Note that these global quantities can be written as an average sum of local terms available across sensors, i.e.,

$$\mathbf{D}(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x) = p_1 \left[\frac{1}{p_1} \sum_{j=1}^{p_1} \mathbf{d}_j \bar{\mathbf{x}}_t(j) \right], \mathbf{E}(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y) = p_2 \left[\frac{1}{p_2} \sum_{i=1}^{p_2} \mathbf{e}_i \bar{\mathbf{y}}_t(i) \right] \quad (26)$$

Substituting the previous quantities in the cost in (15), we obtain the following minimization problem

$$\begin{aligned} \hat{\mathbf{D}}_{\rho:}^{\tau, \kappa} = \arg \min_{\mathbf{D}_{\rho:}} & N^{-1} \sum_{t=0}^{N-1} \left\| \sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i) - \sum_{j=1}^{p_1} \mathbf{D}(\rho, j) \bar{\mathbf{x}}_t(j) \right\|_2^2 \\ & + \lambda_{D, \rho} \|\mathbf{D}_{\rho:}\|_1 + \mathbf{D}_{\rho:}^{\tau, \kappa-1} + \frac{c}{2} \|\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}^{\tau, \kappa-1}\|_2^2 \\ & + \varepsilon \|\mathbf{I}_{\rho:} - \frac{1}{N} \sum_{t=0}^{N-1} [(\sum_{j=1}^{p_1} \mathbf{D}(\rho, j) \bar{\mathbf{x}}_t(j)) (\sum_{j=1}^{p_1} \hat{\mathbf{d}}_j^{\tau-1} \bar{\mathbf{x}}_t(j))^T]\|_2^2 \end{aligned} \quad (27)$$

Sensor $j \in \mathfrak{A}$ is responsible for updating the entries $\{\mathbf{D}(\rho, j)\}_{\rho=1}^q$ in $\mathbf{D}$, which further implies that sensor j will keep tracking of the j th entry in vectors $\mathbf{b}_{\rho}$ and $\mathbf{p}_{\rho}$, namely $\mathbf{b}_{\rho}(j)$ and $\mathbf{p}_{\rho}(j)$, respectively, for $\rho = 1, \dots, q$. Toward this end, the minimization problem in (27) is split into p_1 subtasks each one of which focuses on updating one entry of $\mathbf{D}_{\rho:}$ while fixing the remaining entries. Then, the optimization problem for

updating $\mathbf{D}(\rho, j)$ at sensor $j \in \mathfrak{A}$ can be written as (d denotes the optimization variable)

$$\begin{aligned} \hat{\mathbf{D}}^{\tau, \kappa}(\rho, j) = \arg \min_d & N^{-1} \sum_{t=0}^{N-1} \left\| \sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i) - \sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i) \right. \\ & + \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, j) \bar{\mathbf{x}}_t(j) - d \cdot \bar{\mathbf{x}}_t(j) \left. \right\|_2^2 + \lambda_{D, \rho} |d| + d \mathbf{p}_{\rho}^{\tau, \kappa-1}(j) + \frac{c}{2} [d - \mathbf{b}_{\rho}^{\tau, \kappa-1}(j)]^2 \\ & + \varepsilon \|\mathbf{I}_{\rho:} - N^{-1} \sum_{t=0}^{N-1} [(\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i) - \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, j) \bar{\mathbf{x}}_t(j) \\ & + d \cdot \bar{\mathbf{x}}_t(j)) (\sum_{i=1}^{p_1} \hat{\mathbf{d}}_i^{\tau-1} \bar{\mathbf{x}}_t(i))^T]\|_2^2, \text{ for } j = 1, 2, \dots, p_1, \end{aligned} \quad (28)$$

where $\hat{\mathbf{E}}^{\tau-1} := \hat{\mathbf{E}}^{\tau-1, K}$, and $\hat{\mathbf{D}}^{\tau-1} := \hat{\mathbf{D}}^{\tau-1, K}$ (the updates after K ADMM iterations) while $\hat{\mathbf{d}}_i^{\tau-1}$ is the i th column of $\hat{\mathbf{D}}^{\tau-1}$. Further, ADMM will be employed to allow sensors estimate in a distributed fashion the global quantities $\sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i)$, $\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$, and $\sum_{i=1}^{p_1} \hat{\mathbf{d}}_i^{\tau-1} \bar{\mathbf{x}}_t(i)$. Sensor $j \in \mathfrak{A}$ has available measurements $\{\bar{\mathbf{x}}_t(j)\}_{t=0}^{N-1}$, and is responsible for updating $\mathbf{b}_{\rho}(j)$, $\mathbf{p}_{\rho}(j)$, and $\{\mathbf{D}(\rho, j)\}_{\rho=1}^q$. To this end, ADMM will be utilized to express the quantities $\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$, in (29), for $t = 0, \dots, N-1$ as the solution of a separable convex minimization problem that can be solved in a distributed fashion and enable each sensor j to estimate

these global quantities. These local estimates will be used to replace the corresponding global quantities in (28) and enable sensor j to update $\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)$.

ADMM based estimation of global quantities:

Sensor j can obtain estimates for $\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$ for $t = 0, \dots, N-1$, by solving the separable constrained minimization problem:

$$\min_{\eta_{i,t,\rho}^{\tau,\kappa-1}} \sum_{i=1}^{p_1} \|\eta_{i,t,\rho}^{\tau,\kappa-1} - p_1 \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)\|_2^2 \quad (29)$$

$$\text{s. to } \eta_{i,t,\rho}^{\tau,\kappa-1} = \eta_{i',t,\rho}^{\tau,\kappa-1}, i' \in \mathcal{N}_i^{\mathfrak{A}} \text{ for } t = 0, 1, \dots, N-1$$

whose optimal solution is $\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$, and $\eta_{i,t,\rho}^{\tau,\kappa-1} \in \mathbb{R}^1$ corresponds to the local estimate of $\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$ for sensor $i \in \mathfrak{A}$, and $\mathcal{N}_i^{\mathfrak{A}}$ are the neighboring sensors of i in sensor set $\mathfrak{A}$. Note that the constraints $\eta_{i,t,\rho}^{\tau-1} = \eta_{i',t,\rho}^{\tau-1}, i' \in \mathcal{N}_i^{\mathfrak{A}}$ for $i = 1, \dots, p_1$ guarantee that the local estimates $\eta_{i,t,\rho}^{\tau,\kappa-1}$ will be equal across all the p_1 sensors. Employing ADMM across the network, see details in Apdx. C, the minimization task in (29) can be tackled by sensor j through updating the sensor j 's local estimate $\eta_{j,t,\rho}^{\tau,\kappa-1}$, along with the Lagrange multipliers $\{v_{i',t,\rho}^{j',\tau,\kappa-1}\}_{i' \in \mathcal{N}_i^{\mathfrak{A}}}$, that correspond to the quality constraints $\eta_{i,t,\rho}^{\tau,\kappa-1} = \eta_{i',t,\rho}^{\tau,\kappa-1}$ for $i' \in \mathcal{N}_i^{\mathfrak{A}}$.

The corresponding ADMM updating recursions are given as

$$v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota) = v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota-1) + 0.5c_2[\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota) - \eta_{j',t,\rho}^{\tau,\kappa-1}(\iota)] \quad (30)$$

$$\begin{aligned} \eta_{j,t,\rho}^{\tau,\kappa-1}(\iota+1) &= [(2 + 2c_2|\mathcal{N}_j^{\mathfrak{A}}|)\mathbf{I}_q]^{-1} \times [2p_1 \hat{\mathbf{D}}^{\tau-1}(\rho, j) \bar{\mathbf{x}}_t(j) \\ &\quad - \sum_{j' \in \mathcal{N}_j^{\mathfrak{A}}} ((v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota) - v_{j',t,\rho}^{j,\tau,\kappa-1}(\iota)) + c_2(\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota) + \eta_{j',t,\rho}^{\tau,\kappa-1}(\iota)))] \end{aligned} \quad (31)$$

in which, ι is the network ADMM iteration index, and c_2 represents a positive step-size. The convergence results in [32] reveal that if $\iota \rightarrow \infty$, $\lim_{\iota \rightarrow \infty} \eta_{j,t,\rho}^{\tau,\kappa-1}(\iota) = \sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)$, no matter how $v_{j,t,\rho}^{j',\tau,\kappa-1}(0)$ and $\eta_{j,t,\rho}^{\tau,\kappa-1}(0)$ are initialized. In the practice, as well as the numerical tests a finite number of K_2 ADMM iterations are performed and let $\hat{\eta}_{j,t,\rho}^{\tau,\kappa-1} := \eta_{j,t,\rho}^{\tau,\kappa-1}(K_2)$. Let's define $\hat{\boldsymbol{\eta}}_{j,t}^{\tau-1} := [\hat{\eta}_{j,t,1}^{\tau-1,K}, \dots, \hat{\eta}_{j,t,q}^{\tau-1,K}]^T \in \mathbb{R}^{q \times 1}$ denote the corresponding estimates, which can be obtained in the beginning of τ th coordinate cycle at sensor j .

A similar procedure (30)-(31) is applied across sensors in $\mathfrak{B}$ for estimating $\sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i)$ after implementing K_2 network ADMM iterations to obtain estimates $\hat{\nu}_{f,t,\rho}^{\tau-1}$ where sensor $f \in \mathfrak{B}$ is in the neighborhood of sensor $j \in \mathfrak{A}$ considered earlier. Thus, after using the local estimates the cost function wrt $\mathbf{D}(\rho, j)$ in (28) can be replaced with

$$\begin{aligned}
\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) &= \arg \min_d 0.5c[d - \mathbf{b}_\rho^{\tau,\kappa-1}(j)]^2 + \lambda_{D,\rho}|d| + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(j) \\
&+ \varepsilon \|\mathbf{I}_\rho - \frac{1}{N} \sum_{t=0}^{N-1} [(\hat{\eta}_{j,t,\rho}^{\tau,\kappa-1} - \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j) \bar{\mathbf{x}}_t(j) + d \cdot \bar{\mathbf{x}}_t(j)) \times (\hat{\eta}_{j,t}^{\tau-1})^T]\|_2^2 \\
&+ N^{-1} \sum_{t=0}^{N-1} \|\hat{\nu}_{f,t,\rho}^{\tau-1} - \hat{\eta}_{j,t,\rho}^{\tau,\kappa-1} - d \cdot \bar{\mathbf{x}}_t(j) + \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j) \bar{\mathbf{x}}_t(j)\|_2^2 \text{ for } j = 1, \dots, p_1
\end{aligned} \tag{32}$$

For notational simplicity let

$$\begin{aligned}
\hat{\nu}_{f,\rho}^{\tau-1} &:= [\hat{\nu}_{f,0,\rho}^{\tau-1}, \hat{\nu}_{f,1,\rho}^{\tau-1} \dots \hat{\nu}_{f,N-1,\rho}^{\tau-1}] \in \mathbb{R}^{1 \times N}, \hat{\eta}_j^{\tau-1} := [\hat{\eta}_{j,0}^{\tau-1}, \hat{\eta}_{j,1}^{\tau-1} \dots \hat{\eta}_{j,N-1}^{\tau-1}] \in \mathbb{R}^{q \times N}, \\
\hat{\eta}_{j,\rho}^{\tau,\kappa-1} &:= [\hat{\eta}_{j,0,\rho}^{\tau,\kappa-1}, \hat{\eta}_{j,1,\rho}^{\tau,\kappa-1} \dots \hat{\eta}_{j,N-1,\rho}^{\tau,\kappa-1}] \in \mathbb{R}^{1 \times N}, \bar{\mathbf{X}}_j := [\bar{\mathbf{x}}_0(j), \bar{\mathbf{x}}_1(j) \dots \bar{\mathbf{x}}_{N-1}(j)] \in \mathbb{R}^{1 \times N}
\end{aligned}$$

Moreover, the following notation is used

$$\begin{aligned}
\mathbf{M}_{j,\rho}^{\tau,\kappa} &:= \hat{\nu}_{f,\rho}^{\tau-1} - \hat{\eta}_{j,\rho}^{\tau,\kappa-1} + \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j) \bar{\mathbf{X}}_j, \mathbf{Q}_j^{\tau,\kappa} := \varepsilon^{0.5} N^{-1} \bar{\mathbf{X}}_j (\hat{\eta}_j^{\tau-1})^T \\
\mathbf{P}_{j,\rho}^{\tau,\kappa} &:= \varepsilon^{0.5} \mathbf{I}_\rho - \varepsilon^{0.5} N^{-1} (\hat{\eta}_{j,\rho}^{\tau,\kappa-1} - \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j) \bar{\mathbf{X}}_j) (\hat{\eta}_j^{\tau-1})^T
\end{aligned} \tag{33}$$

then the cost in (32) can be rewritten as

$$\begin{aligned}
\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) &= \arg \min_d N^{-1} \|\mathbf{M}_{j,\rho}^{\tau,\kappa} - d \cdot \bar{\mathbf{X}}_j\|_2^2 + \lambda_{D,\rho}|d| + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(j) \\
&+ \|\mathbf{P}_{j,\rho}^{\tau,\kappa} - d \cdot \mathbf{Q}_j^{\tau,\kappa}\|_2^2 + \frac{c}{2} (d - \mathbf{b}_\rho^{\tau,\kappa-1}(j))^2 \text{ for } j = 1, 2, \dots, p_1.
\end{aligned} \tag{34}$$

Using a similar method to solve (12) in Centralized R-CCA framework, the solution to (34) can be expressed as

$$\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) = \mathbb{F}(\mathbf{\Gamma}_{\rho,j}^{\tau,\kappa}, \mathbf{\Lambda}_{\rho,j}^{\tau,\kappa}, \frac{c}{2} (\mathbf{b}_\rho^{\tau,\kappa-1}(j) - \frac{\mathbf{p}_\rho^{\tau,\kappa-1}(j)}{c}), \frac{c}{2}) \tag{35}$$

where $\mathbf{\Gamma}_{\rho,j}^{\tau,\kappa} = [N^{-0.5} \mathbf{M}_{j,\rho}^{\tau,\kappa}, \mathbf{P}_{j,\rho}^{\tau,\kappa}]^T$, and $\mathbf{\Lambda}_{\rho,j}^{\tau,\kappa} = [N^{-0.5} \bar{\mathbf{X}}_j, \mathbf{Q}_j^{\tau,\kappa}]^T$.

After finishing the first ADMM step for (15), our focus is shifted to implementing the second step in (16) in a distributed fashion. Similar to the optimal solution for $\mathbf{b}_\rho^{\tau,\kappa-1}$ given by (21) in the Centralized R-CCA scheme, sensor $j \in \mathfrak{A}$ is responsible for updating the j th entry of $\mathbf{b}_\rho^{\tau,\kappa}$ in (21) which is given by

$$\frac{1}{c} [1 - \frac{\phi_D}{\|\mathbf{p}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{D}}_{\rho,:}^{\tau,\kappa})^T\|_2}] + (\mathbf{p}_\rho^{\tau,\kappa-1}(j) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)) \tag{36}$$

where $\mathbf{p}_\rho^{\tau,\kappa-1}(j)$ and $\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)$ are available for sensor $j \in \mathfrak{A}$. The challenge in calculating (36) is finding the global value $\|\mathbf{p}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{D}}_{\rho,:}^{\tau,\kappa})^T\|_2$ in a distributed fashion. Let $m_\rho^{\tau,\kappa} := \|\mathbf{p}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{D}}_{\rho,:}^{\tau,\kappa})^T\|_2^2$, then

$$m_\rho^{\tau,\kappa} = \sum_{\ell=1}^{p_1} (\mathbf{p}_\rho^{\tau,\kappa-1}(\ell) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho, \ell))^2. \tag{37}$$

Notice that, $m_\rho^{\tau,\kappa}$ is an average-like quantity, and ADMM can be employed as in (35)-(36) to estimate $m_\rho^{\tau,\kappa}$ across sensors.

Let $\hat{u}_{j,\rho}^{\tau,\kappa}$ denote sensor j 's local estimate of the global quantity $m_\rho^{\tau,\kappa}$ after taking K_2 network ADMM iterations, say $u_{j,\rho}^{\tau,\kappa}(K_2)$. Thus, $\mathbf{b}_\rho^{\tau,\kappa}(j)$ can be obtained as

$$\mathbf{b}_\rho^{\tau,\kappa}(j) = \frac{1}{c} \left[1 - \frac{\phi_{D,\rho}}{\hat{u}_{j,\rho}^{\tau,\kappa}} \right]_+ (\mathbf{p}_\rho^{\tau,\kappa-1}(j) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)). \quad (38)$$

Finally, the last task is to complete the third ADMM step in (17) in a distributed way. Sensor $j \in \mathfrak{A}$ needs to update $\mathbf{p}_\rho^{\tau,\kappa}(j)$ based on the most recent updated $\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)$ and $\mathbf{b}_\rho^{\tau,\kappa}(j)$, which can be directly obtained by

$$\mathbf{p}_\rho^{\tau,\kappa}(j) = \mathbf{p}_\rho^{\tau,\kappa-1}(j) + c(\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) - \mathbf{b}_\rho^{\tau,\kappa}(j)). \quad (39)$$

Starting from (9), a similar process will be followed to obtain updating recursions for $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)$, which involves iteratively updating the corresponding vector $\check{\mathbf{b}}_\rho^{\tau,\kappa}(i)$ and Lagrangian multiplier $\check{\mathbf{p}}_\rho^{\tau,\kappa}(i)$. Specifically, let's denote $\hat{\nu}_{i,t,\rho}^{\tau,\kappa-1}$ as the local estimate of $\sum_{\ell=1}^{p_2} \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, \ell) \bar{\mathbf{y}}_\ell(\ell)$ after applying K_2 ADMM iterations across sensors in $\mathfrak{B}$. Define $\hat{\nu}_{i,t}^{\tau-1} := [\hat{\nu}_{i,t,1}^{\tau-1,K}, \dots, \hat{\nu}_{i,t,q}^{\tau-1,K}]^T \in \mathbb{R}^{q \times 1}$ and let

$$\hat{\nu}_{i,\rho}^{\tau,\kappa-1} := [\hat{\nu}_{i,0,\rho}^{\tau,\kappa-1}, \dots, \hat{\nu}_{i,N-1,\rho}^{\tau,\kappa-1}]^T \in \mathbb{R}^{q \times N} \quad (40)$$

$$\hat{\boldsymbol{\eta}}_{h,\rho}^{\tau-1} := [\hat{\boldsymbol{\eta}}_h^{\tau-1}]_\rho: \in \mathbb{R}^{1 \times N}, \bar{\mathbf{Y}}_i := [\bar{\mathbf{y}}_0(i), \dots, \bar{\mathbf{y}}_{N-1}(i)]^T \in \mathbb{R}^{q \times N} \quad (41)$$

where $h \in \mathfrak{A}$ corresponds to the neighboring sensor of $i \in \mathfrak{B}$. Using similar notation as in (33) let us define for sensor $i \in \mathfrak{B}$

$$\begin{aligned} \check{\mathbf{M}}_{i,\rho}^{\tau,\kappa} &:= \hat{\boldsymbol{\eta}}_{h,\rho}^{\tau-1} - \hat{\nu}_{i,\rho}^{\tau,\kappa-1} + \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{Y}}_i, \check{\mathbf{Q}}_i^{\tau,\kappa} := v^{0.5} N^{-1} \bar{\mathbf{Y}}_i (\hat{\nu}_i^{\tau-1})^T \\ \check{\mathbf{P}}_{i,\rho}^{\tau,\kappa} &:= v^{0.5} \mathbf{I}_\rho - v^{0.5} N^{-1} (\hat{\nu}_{i,\rho}^{\tau,\kappa-1} - \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, i) \bar{\mathbf{Y}}_i) (\hat{\nu}_i^{\tau-1})^T \end{aligned} \quad (42)$$

Thus, the recursions for updating $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)$, $\check{\mathbf{b}}_\rho^{\tau,\kappa}(i)$, and $\check{\mathbf{p}}_\rho^{\tau,\kappa}(i)$ can be obtained as

$$\begin{aligned} \hat{\mathbf{E}}^{\tau,\kappa}(\rho, i) &= \mathbb{F}([N^{-0.5} \check{\mathbf{M}}_{i,\rho}^{\tau,\kappa}, \check{\mathbf{P}}_{i,\rho}^{\tau,\kappa}]^T, [N^{-0.5} \bar{\mathbf{Y}}_i, \check{\mathbf{Q}}_i^{\tau,\kappa}]^T, \\ &\quad \frac{c}{2} (\check{\mathbf{b}}_\rho^{\tau,\kappa-1}(i) - \frac{\check{\mathbf{p}}_\rho^{\tau,\kappa-1}(i)}{c}), \frac{c}{2}) \end{aligned} \quad (43)$$

$$\check{\mathbf{b}}_\rho^{\tau,\kappa}(i) = \frac{1}{c} \left(1 - \frac{\phi_E}{\hat{u}_i^{\tau,\kappa}} \right) (\check{\mathbf{p}}_\rho^{\tau,\kappa-1}(i) + c\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)) \quad (44)$$

$$\check{\mathbf{p}}_\rho^{\tau,\kappa}(i) = \mathbf{p}_\rho^{\tau,\kappa-1}(j) + c(\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i) - \check{\mathbf{b}}_\rho^{\tau,\kappa}(i)) \quad (45)$$

where $\check{u}_i^{\tau,\kappa}$ denotes the local estimate at sensor $i \in \mathfrak{B}$ local estimate of $\|\check{\mathbf{p}}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{E}}_{\rho}^{\tau,\kappa})^T\|_2^2$ after applying K_2 ADMM iterations. To summarize, the distributed regularized CCA(DR-CCA) algorithm is tabulated as Algorithm 2.

From the convergence claims in [32] it follows that, as $K_2 \rightarrow \infty$, $\hat{\eta}_{j,\rho}^{\tau,\kappa-1} \rightarrow \hat{\mathbf{D}}_{\rho:}^{\tau,\kappa-1} \mathbf{X}$, $\hat{\eta}_j^{\tau-1} \rightarrow \hat{\mathbf{D}}^{\tau-1} \mathbf{X}$, $\hat{\nu}_{i,\rho}^{\tau,\kappa-1} \rightarrow \hat{\mathbf{E}}_{\rho:}^{\tau,\kappa-1} \mathbf{Y}$, and $\hat{\nu}_i^{\tau-1} \rightarrow \hat{\mathbf{E}}^{\tau-1} \mathbf{X}$. Notice that, we implement ADMM iterations for $\hat{\eta}_{j,\rho}^{\tau,\kappa-1}$ and $\hat{\nu}_{j,\rho}^{\tau,\kappa-1}$ separately, each of which entails three updating variables as in standard ADMM. Furthermore, $\hat{\eta}_j^{\tau-1}$ and $\hat{\nu}_j^{\tau-1}$ are obtained through $\hat{\eta}_{j,\rho}^{\tau-1,K_2}$ and $\hat{\nu}_{j,\rho}^{\tau-1,K_2}$. Specifically, $\hat{\eta}_j^{\tau-1} := [\hat{\eta}_{j,1}^{\tau-1,K_2}; \dots; \hat{\eta}_{j,q}^{\tau-1,K_2}]$ and $\hat{\nu}_j^{\tau-1} := [\hat{\nu}_{j,1}^{\tau-1,K_2}; \dots; \hat{\nu}_{j,q}^{\tau-1,K_2}]$. Similary, as $K \rightarrow \infty$, the $\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa}$ and $\hat{\mathbf{E}}_{\rho:}^{\tau,\kappa}$ obtained from (35) and (43) go to the optimizer of $\hat{\mathbf{D}}_{\rho:}^{\tau}$ and $\hat{\mathbf{E}}_{\rho:}^{\tau}$ of (12) and (22), respectively. Further, as the block coordinate cycle $\tau \rightarrow \infty$ the updates $\hat{\mathbf{D}}^{\tau}$ and $\hat{\mathbf{E}}^{\tau}$ will approach $\delta(\epsilon)$ -close to a stationary point of the cost in (9), where $\lim_{\epsilon \rightarrow 0} \delta(\epsilon) = 0$ [similar arguments as in the proof of Prop. 1 can be used here].

6. Parameter Selection

The CR-CCA and DR-CCA schemes utilize two different kinds of regularization coefficients. The ℓ_2 -regularization coefficients ϕ_D and ϕ_E control the number of zero-rows in $\mathbf{D}$ and $\mathbf{E}$. The ℓ_1 -regularization coefficients $\{\lambda_{D,\rho}$ and $\lambda_{E,\rho}\}_{\rho=1}^q$ are used to control the number of zeros in the ρ th row of matrices $\mathbf{D}$ and $\mathbf{E}$, respectively. Thus, it is critical to choose the proper coefficients, which ensure CR-CCA and DR-CCA algorithms can accurately and efficiently identify and match groups of sensors in $\mathfrak{A}$ and $\mathfrak{B}$, whose measurements are affected by the same source. Further, proper selection of ϕ_D and ϕ_E can facilitate estimation of the number of field sources via the number of nonzero rows in the estimated $\mathbf{D}$ and $\mathbf{E}$ matrices.

First, a range of the number of field sources is going to be estimated. Recall that, in standard CCA, $\hat{\mathbf{D}}\mathbf{x}_t$ is an estimate of the underlying source signals. The ensemble covariance of $\hat{\mathbf{D}}\mathbf{x}_t$ is $\hat{\mathbf{D}}\hat{\Sigma}_x\hat{\mathbf{D}}^T$ and given that there are M sources, it should ideally have rank equal to M . Thus, in an ideal noiseless setting $\hat{\mathbf{D}}\hat{\Sigma}_x\hat{\mathbf{D}}^T$ should have M nonzero eigenvalues corresponding the energy contributed by the M field sources. In practice, the ensemble $\hat{\mathbf{D}}\hat{\Sigma}_x\hat{\mathbf{D}}^T$ is estimated by the sample-average based estimator $\hat{\mathbf{D}}\hat{\hat{\Sigma}}_x\hat{\mathbf{D}}^T$, which will be affected by noise and the usage of a finite number of samples estimating $\hat{\hat{\Sigma}}_x$. In general, $\text{rank}(\hat{\mathbf{D}}\hat{\hat{\Sigma}}_x\hat{\mathbf{D}}^T) = q$ in the presence of noise. The challenge is to decide how many eigenvalues to keep and interpret them as source components.

After applying CR-CCA (or DR-CCA) for $\phi_D = \phi_E = 0$ and $\lambda_{D,\rho} = \lambda_{E,\rho} = 0$ for $\rho = 1, \dots, q$, we estimate the number of ‘source-related’ eigenvalues in $\hat{\mathbf{D}}\hat{\hat{\Sigma}}_x\hat{\mathbf{D}}^T$

using the cumulative percent variance (CPV) approach, see. e.g. [42]. Specifically, the percentage of total variance captured by the first A largest eigenvalues of $\hat{\mathbf{D}}\hat{\Sigma}_x\hat{\mathbf{D}}^T$, namely $\lambda_{x,1} \geq \lambda_{x,2} \geq \dots \geq \lambda_{x,A}$ is quantified as

$$CPV_A(\%) = \frac{\sum_{i=1}^A \lambda_{x,i}}{\sum_{i=1}^q \lambda_{x,i}} \times 100, \quad (46)$$

where q corresponds also to the total number of eigenvalues of $\hat{\mathbf{D}}\hat{\Sigma}_x\hat{\mathbf{D}}^T$. A relatively large and small value, namely $\mathfrak{R}_u$ and $\mathfrak{R}_l$, are used for $CPV_A(\%)$ to estimate an upper and lower bound on the number of field sources. Values $\mathfrak{R}_u = 95\%$ and $\mathfrak{R}_l = 75\%$, will be chosen here to estimate the upper and lower bounds $\hat{M}_u$ and $\hat{M}_l$, respectively, on the number of sources. The percentages can be set based on prior information we may have on how strong the sources are relative to the sensing noise.

Secondly, the zero column index sets of $\mathbf{D}$ and $\mathbf{E}$, denoted as $\mathcal{C}_D$ and $\mathcal{C}_E$, respectively, are estimated. At this stage the number of nonzero rows in $\mathbf{D}$ and $\mathbf{E}$ are irrelevant, thus $\phi_D = \phi_E = 0$, while $\lambda_{D,1} = \lambda_{D,2} = \dots = \lambda_{D,q}$ and $\lambda_{E,1} = \lambda_{E,2} = \dots = \lambda_{E,q}$ since we are looking for column-wise sparse structures. The estimates of sparse sets $\mathcal{C}_D$ and $\mathcal{C}_E$ can be obtained through the following three steps: Step 1) Find the smallest values of $\lambda_{D,\rho}$ s and $\lambda_{E,\rho}$ s, denoted here as λ_D^{max} and λ_E^{max} , that result $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ in CR-CCA (or DR-CCA) to be equal to zero; Step 2) Multiply λ_D^{max} and λ_E^{max} with a sufficiently small coefficient ω_1 , and apply the scaled sparsity-controlling in CR-CCA (or DR-CCA) to get $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$; Step 3) Use the zero column support of $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ to estimate sets $\mathcal{C}_D$ and $\mathcal{C}_E$. This process can be used to distinguish the columns in $\mathbf{D}$ and $\mathbf{E}$ (and subsequently the entries in $\mathbf{x}_t$ and $\mathbf{y}_t$) that sense a source (corresponding to a nonzero column), or just contain sensing noise (corresponding to zero-columns). To estimate the unknown λ_D^{max} and λ_E^{max} , via estimates $\hat{\lambda}_D^m$ and $\hat{\lambda}_E^m$ someone can start from $\lambda_{D,\rho} = \lambda_{E,\rho} = 0$ for $\rho = 1, \dots, q$, and gradually increase $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ by a small step size Δ_λ and apply CR-CCA (or DR-CCA) to obtain $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ until the matrices $\hat{\mathbf{D}} = \mathbf{0}$ and $\hat{\mathbf{E}} = \mathbf{0}$. Then, the largest set of values $\lambda_{D,\rho}$, $\lambda_{E,\rho}$ that gave nonzero estimates $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ is retained as $\hat{\lambda}_D^m$ and $\hat{\lambda}_E^m$.

Third, proper ϕ_D , ϕ_E , $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^q$ are selected to implement the clustering task via CR-CCA (or DR-CCA). Coefficients ϕ_D , ϕ_E , $\lambda_{D,\rho}$, and $\lambda_{E,\rho}$ are initialized to zero. Then, $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ are increased gradually until the zero-column set of estimates $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$ match the sets $\mathcal{C}_D$, $\mathcal{C}_E$ obtained earlier. Let the sparsity-controlling

coefficients achieving that be denoted as $\lambda_{D,0}$ and $\lambda_{E,0}$. For simplicity in notation let $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ denote the estimates for $\mathbf{D}$ and $\mathbf{E}$ after using the most recently updated values for $\phi_D, \phi_E, \{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^q$ in CR-CCA (or DR-CCA).

460 Fixing the coefficients $\lambda_{D,\rho} = \lambda_{D,0}$ and $\lambda_{E,\rho} = \lambda_{E,0}$ for $\rho = 1, \dots, q$, the coefficients $\phi_D = \phi_E$ are gradually increased (starting from 0) until the number of non-zero rows in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ lies between $\hat{M}_l$ and $\hat{M}_u$.

As ϕ_D and ϕ_E are increasing, more rows in $\mathbf{D}$ and $\mathbf{E}$ will be zeroed out. The challenge is when to stop increasing ϕ_D and ϕ_E such that the number of nonzero rows in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ does not drop below the number of field sources. The goal is to obtain estimates $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ that share the same nonzero row index set and they are robust in the sense that when slightly changing ϕ_D or ϕ_E does not change the number of nonzero rows in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$. Let $\tilde{\mathcal{R}}_D$ and $\tilde{\mathcal{R}}_E$ denote the nonzero row index set for matrices $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$, respectively. The coefficients ϕ_D and ϕ_E will be increased up to the point where 470 $\tilde{\mathcal{R}}_D = \tilde{\mathcal{R}}_E$. Since there is a possibility that a row of $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$, say $\hat{\mathbf{D}}_{\rho_i}$ and $\hat{\mathbf{E}}_{\rho_i}$, may not be zeroed out simultaneously by adjusting ϕ_D and ϕ_E , we force concurrent zeroing out by increasing correspondingly $\lambda_{D,\rho}$ or $\lambda_{E,\rho}$. For fixed ϕ_D and ϕ_E parameters, the sparsity-controlling coefficients are readjusted until $\tilde{\mathcal{R}}_D = \tilde{\mathcal{R}}_E$, and $\hat{\mathbf{D}}, \hat{\mathbf{E}}$ are *robust* (the set of nonzero rows does not change with slight changes in ϕ_D or ϕ_E). If the latter 475 requirements can not be met, ϕ_D and ϕ_E are increased to generate new zero rows in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$, and then $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ can be readjusted again.

The parameter selection scheme is summarized below as Algorithm 3. The zero-column index sets of the estimates $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ are denoted as $\mathcal{C}_{D,c}$ and $\mathcal{C}_{E,c}$, respectively; while the number of the common non-zero rows in $\hat{\mathbf{D}}, \hat{\mathbf{E}}$ is denoted as M_c .

480 The **Break while** in Alg. 3 terminates the execution of the inner while loop, and the **STOP** exits from Alg. 3. In the numerical tests, the parameters involved in Alg.3 are set as follows: $\mathfrak{R}_l = 75\%$, $\mathfrak{R}_u = 95\%$, $\omega_1 = 0.01$, $\Delta_\lambda = 0.005$, $\Delta_\phi = 0.005$, $\varepsilon = 0.5$, and $\nu = 0.5$.

7. Numerical Tests

485 The performance of CR-CCA and DR-CCA schemes is tested and compared with the existing methods in terms of the probability of correctly clustering sensors according to their source content, which is equal to the probability of correctly assigning zero and

non-zero entries when estimating $\mathbf{D}$ and $\mathbf{E}$. The proposed schemes are compared with *i*) the sparsity inducing CCA algorithm in [36] abbreviated as PMD; and *ii*) intelligent K-Means (iK-Means) clustering approach in [13], [25] which estimates the number of clusters based on sequential extraction of anomalous patterns in the observation data, and then applying estimated cluster number in the traditional K-Means [22].

Consider a setting in which $M = 3$ uncorrelated sources are present, with p_1 sensors in $\mathfrak{A}$ and p_2 sensors in set $\mathfrak{B}$. The diffusivities in the Green's function, namely γ_1 and γ_2 , are set as $\gamma_1 = 1$ and $\gamma_2 = 2$. Three scenarios with low, medium, and high number of sensors (denoted as *low*, *medium*, and *high* setting, respectively) are considered in the testing. Specifically, the *low* setting consists of $p_1 = 10$ sensors in $\mathfrak{A}$, and $p_2 = 10$ sensors in set $\mathfrak{B}$, while sensors in sets $\{\mathfrak{A}_1, \mathfrak{B}_2\}$, $\{\mathfrak{A}_3, \mathfrak{B}_4\}$, and $\{\mathfrak{A}_5, \mathfrak{B}_6\}$ observe sources $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$, and $\mathbf{s}_3(t)$, respectively, and the remaining 14 sensors sense noise. In the *medium* setting, $p_1 = 15$ and $p_2 = 15$ sensors are deployed, where sensors in $\{\mathfrak{A}_1, \mathfrak{A}_5, \mathfrak{B}_2, \mathfrak{B}_3\}$, $\{\mathfrak{A}_6, \mathfrak{B}_7\}$, and $\{\mathfrak{A}_2, \mathfrak{A}_7, \mathfrak{B}_8\}$ sense sources $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$, and $\mathbf{s}_3(t)$, respectively, and the remaining 21 sensors only sense noise. The *high* setting consists of $p_1 = 30$ sensors in $\mathfrak{A}$ and $p_2 = 30$ sensors in set $\mathfrak{B}$. Sensors $\{\mathfrak{A}_4, \mathfrak{A}_{23}, \mathfrak{B}_2, \mathfrak{B}_8, \mathfrak{B}_{10}, \mathfrak{B}_{18}, \mathfrak{B}_{24}, \mathfrak{B}_{30}\}$ observe source $\mathbf{s}_1(t)$, while source $\mathbf{s}_2(t)$ affects sensors $\{\mathfrak{A}_2, \mathfrak{B}_5, \mathfrak{B}_{29}\}$. Further, sensors $\{\mathfrak{A}_7, \mathfrak{A}_{13}, \mathfrak{A}_{21}, \mathfrak{A}_{24}, \mathfrak{B}_1, \mathfrak{B}_4, \mathfrak{B}_{23}\}$ acquire measurements from source $\mathbf{s}_3(t)$, and the remaining sensors observe just sensing noise. The *low* number of sensors setting simulation results are depicted in Fig. 3, while Fig. 4 shows the performance for the *medium* setting, and Fig. 2, Fig. 5, Fig. 6 correspond to the *high* number of sensors setting. In Fig. 2, Fig. 3, Fig. 4, Fig. 5, and Fig. 7, the sensing signal-to-noise ratio (SNR) in the acquired measurements is set to be 10dB. The number of ADMM iterations, namely K , is set equal to $K = 20$. Note that, all the simulation results are produced after 150 independent Monte Carlo runs.

In Fig. 2 we compare the probability of correctly clustering sensors among CR-CCA for different q (upper bounding the actual number of sources), DR-CCA for different q ($q = 5, 6, 7$), PMD and iK-Means. The number of the network ADMM iterations, say K_2 , is set to 20. The corresponding regularization coefficients, ϕ_D , ϕ_E , $\lambda_{D,pS}$ and $\lambda_{E,pS}$ in CR-CCA and DR-CCA are selected using the Alg. 3 in Sec. 6. The sparsity-controlling coefficients in PMD are selected through cross-validation, whose

details can be found in [36]. Fig. 2 depicts that CR-CCA yields the best performance,
 and as the number of data samples increases, the probability of correct sensor clustering
 will increase. It is also interesting that DR-CCA achieves better performance than other
 centralized alternatives, i.e., PMD and iK-Means. Thus, Fig. 2 clearly demonstrates
 the capability of CR-CCA and DR-CCA to correctly cluster sensors in heterogeneous
 networks based on their information content. Note also that selecting q does not af-
 fect performance as long as $q > M$. Notice that for a fixed number of data samples,
 the probability of correctly clustering sensors based on their source content is not af-
 fected significantly by the value of q . Notice that, in the iK-Means, the selection of the
 number of clusters and the task of clustering the sensors are quite associated with the
 magnitude of the sensor data, resulting errors when estimating the number of groups
 and improper clustering, as sensors in the same source-group do not necessary acquire
 measurements of similar magnitude. From testing iK-Means, the estimated number of
 clusters was always equal to 1 irrespective of the number of measurements used, mak-
 ing the probability curve to be straight. The same conclusions with Fig. 2 can also be
 obtained from Fig. 3 and Fig. 4 under the *low* and *medium* number of sensors setting,
 respectively. The results in Fig. 2, Fig. 3 and Fig. 4 corroborate the capability of our
 algorithm in sensor clustering no matter how many sensors are deployed as long as a
 sufficiently large number of consensus iterations K_2 is run.

Fig. 5 depicts the performance of DR-CCA for different network ADMM itera-
 tions, i.e., $K_2 = 10$, $K_2 = 20$, and $K_2 = 30$ for the $p_1 + p_2 = 60$ sensors. The
 test results show that as K_2 increases, DR-CCA can achieve better performance. An-
 other interesting property is that CR-CCA outperforms DR-CCA, which is expected.
 Note that as $K_2 \rightarrow \infty$, the involved global quantities can be precisely estimated by
 the ADMM technique applied in $\mathfrak{A}$ and $\mathfrak{B}$, resulting DR-CCA to behave similarly to
 CR-CCA.

Here we also test the performance of CR-CCA and DR-CCA along with sensing SNR,
 i.e., SNR=5dB, 10dB, 15dB, 20dB and 25dB. The network ADMM iteration is set as
 $K_2 = 2$ for DR-CCA, the sparsity-controlling coefficients are chosen through the Alg.
 3 in Sec. 6, and the number of data vector is set as $N = 500$. Fig. 6 depicts that CR-
 CCA and DR-CCA with larger SNR exhibit better behavior than those with smaller

SNR, irrespective of the values of q used in the simulation for $p_1 + p_2 = 60$ sensors.

Finally, we examine the performance of DR-CCA for $q = 5, 6, 7$ and $N = 500, 1000$, versus the average number of scalars communicated per sensor in the network. Recall that, in τ th block coordinate cycle, in the beginning of updating the ρ th row of $\mathbf{D}$, sensor $j \in \mathfrak{A}$ receives scalars $\{\hat{\nu}_{f,t,\rho}^{\tau-1}\}_{t=0}^{N-1}$ from its neighboring sensor $f \in \mathfrak{B}$, then during the κ th ADMM iteration, sensor j needs to communicate with its $|\mathcal{N}_j^{\mathfrak{A}}|$ neighboring sensors in $\mathfrak{A}$, which includes transmitting scalars $\{\{v_{j',t,\rho}^{j,\tau,\kappa-1}(\ell)\}_{j' \in \mathcal{N}_j^{\mathfrak{A}}}, \eta_{j,t,\rho}^{\tau,\kappa-1}(\ell), w_{j',\rho}^{j,\tau,\kappa}(\ell), u_{j,\rho}^{\tau,\kappa}(\ell)\}_{t=0,\ell=1}^{N-1,K_2}$, and receiving scalars $\{v_{j,t,\rho}^{j',\tau,\kappa-1}(\ell), \eta_{j',t,\rho}^{\tau,\kappa-1}(\ell), w_{j,\rho}^{j',\tau,\kappa}(\ell), u_{j',\rho}^{\tau,\kappa}(\ell)\}_{t=0,\ell=1,j' \in \mathcal{N}_j^{\mathfrak{A}}}$. Here the number of network ADMM iterations is set as $K_2 = 20$, and the values of $\{|\mathcal{N}_j^{\mathfrak{A}}|\}_{j=1}^{p_1}$ and $\{|\mathcal{N}_i^{\mathfrak{B}}|\}_{i=1}^{p_2}$ are summarized in Table. 1 (the single-hop communication distance is set to be 0.4 with sensors lying in the area $[0, 1] \times [0, 1]$). In the simulation, we fix the number of the block coordinate cycle, i.e., $\tau = 0, 1, \dots, 99$, for different testing cases, and also apply the parameter selection algorithm in Alg. 3 to DR-CCA. The simulation results are shown in Fig. 7 for $p_1 + p_2 = 60$ sensor. It can be seen that, in the beginning as the average number of scalars communicated among neighboring (n for short) increases, the probability of correctly clustering sensors also increases. Within each coordinate cycle, a finite number of ADMM iterations, say K and a finite number of network ADMM iterations, namely K_2 , are used to estimate the global quantities in a distributed fashion for DR-CCA. Note that the probability for $N = 1000$ converges to a higher value than that for $N = 500$, under the same q values for $q = 5, 6, 7$.

8. Concluding Remarks

A norm-one and norm-two regularized CCA framework (R-CCA) was put forth and applied to clustering sensor measurements based on their source content in heterogeneous sensor networks. Norm-two regularization was utilized to estimate the unknown number of field sources, while norm-one terms were employed to recover different clusters of information within the sensor data. Relying on block coordinate descent techniques equipped with alternating direction method of multipliers, a novel centralized R-CCA (CR-CCA) was developed to minimize the associated cost problem, which was solved in a recursive way, to perform the heterogeneous data clustering. Building

on CR-CCA and further employing a network ADMM approach, a distributed iterative scheme, namely DR-CCA, was derived to carry out the clustering task in ad hoc sensor network where only neighboring sensors collaborate. The potential of the proposed CR-CCA and DR-CCA in correctly recovering clusters for heterogeneous sensors, was demonstrated via extensive numerical tests.

Appendix A. Derivation for (21): After some matrix manipulations, the minimization problem in (16) is equal to

$$\mathbf{b}_\rho^{\tau+1, \kappa+1} = \arg \min_{\mathbf{b}_\rho} (-(\mathbf{p}_\rho^{\tau+1, \kappa})^T - c\hat{\mathbf{D}}_{\rho:}^{\tau+1, \kappa+1})\mathbf{b}_\rho^T \frac{c}{2} \|\mathbf{b}_\rho\|_2^2 + \phi_D \|\mathbf{b}_\rho\|_2 \quad (47)$$

Using the result in [27], the solution of

$$\min_{\mathbf{x}} 0.5 \cdot \mathbf{x}^T k \mathbf{x} + \mathbf{g}^T \mathbf{x} + \lambda \|\mathbf{x}\|_2 \text{ where } \lambda > 0, k > 0 \quad (48)$$

is $\hat{\mathbf{x}} = -\frac{1}{k}(\|\mathbf{g}\|_2 - \lambda)_+ \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$. After comparing (47) with (48), (21) is obtained.

Appendix B. Proof for Proposition 1:

(I) First it is shown that that as the nested coordinate descent iterations $\tau_2 \rightarrow \infty$, the update $\hat{\mathbf{D}}^{\tau, \kappa, \tau_2}(\rho, \beta)$ using (19) [similarly for $\hat{\mathbf{E}}^{\tau, \kappa, \tau_2}(\rho, \beta)$] converge to a stationary point of the cost in (15).

Let $h(\{\mathbf{D}(\rho, \beta)\}_{\beta=1}^{p_1})$ denote the cost given in (15), which is defined over $\mathbb{R}^{p_1 \times 1}$ and

$$h_0(\{\mathbf{D}(\rho, \beta)\}_{\beta=1}^{p_1}) := N^{-1} \|\hat{\mathbf{E}}_{\rho:}^{\tau-1} \mathbf{Y} - \mathbf{D}_{\rho:} \mathbf{X}\|_2^2 + \varepsilon \|\mathbf{I}_q(\rho, :) - \mathbf{D}_{\rho:} \Sigma_x (\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 + \frac{c}{2} \|\mathbf{D}_{\rho:} - \mathbf{b}_\rho^{\tau, \kappa-1}\|_2^2 + \mathbf{D}_{\rho:} \mathbf{p}_\rho^{\tau, \kappa-1} \quad (49)$$

Consider further the level set $\mathcal{C}^0 := \{\{\mathbf{D}(\rho, \beta)\}_{\beta=1}^{p_1} : h(\{\mathbf{D}(\rho, \beta)\}_{\beta=1}^{p_1}) \leq h(\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, 0})\}$

where $\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, 0}$ is a $1 \times p_1$ vector used to initialize the recursive updates in (19) and selected such that $\|\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, 0}\|_1 < \infty$, from which it follows that $h(\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, 0}) < \infty$. Further, $h(\cdot)$ ensures that any $\mathbf{D}_{\rho:}$ in $\mathcal{C}^0$ satisfy $\lambda_{D, \rho} \|\mathbf{D}_{\rho:}\|_1 \leq h(\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, 0}) < \infty$.

Thus, the set $\mathcal{C}^0$ is compact (closed and bounded), and $h(\cdot)$ is continuous on $\mathcal{C}^0$. Note that the cost in (15) is strictly convex ensuring that the minimizer $\hat{\mathbf{D}}^{\tau, \kappa}(\rho, \cdot)$ is unique.

Also, function $h(\cdot)$ is regular at the unique minimizer, which is outlined in [35, (A1)]. Specifically, the domain of function of $h_0(\cdot)$ is formed by vectors which satisfy that $\mathbf{D}(\rho, \beta) \in (-\infty, +\infty)$. Then, the domain(h_0) = $(-\infty, +\infty)^{p_1 \times 1}$ is an open set. Moreover, $h_0(\cdot)$ is Gâteaux-differential over domain(h_0). Thus, as the iterate τ_2 goes to ∞ , $\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, \tau_2}$ will converge to the minimizer of $h(\cdot)$ for [35, Thm. 4.1 (c)].

(II) According to the convergence claims of ADMM in [9], as the ADMM iterates $\kappa \rightarrow \infty$ and nested coordinate iterations $\tau_2 \rightarrow \infty$, the updates $\hat{\mathbf{D}}^{\tau, \kappa, \tau_2}(\rho, \beta)$ converge to the optimizer $\hat{\mathbf{D}}_{\rho:}^\tau$ in (12) [similarly for update $\{\hat{\mathbf{E}}^{\tau, \kappa, \tau_2}(\rho, j)\}_{j=1}^{p_2}$ in (23)].

In order to minimize the cost w.r.t. one row of $\mathbf{D}$ (or $\mathbf{E}$), namely $\mathbf{D}_{\rho:}$ (or $\mathbf{E}_{\rho:}$), in (9), let us use block coordinate descent (BCD) here without the approximation introduced in (12), and let $\check{\mathbf{D}}^\tau$ and $\check{\mathbf{E}}^\tau$ denote the corresponding updates. During the τ th BCD cycle, the minimization of (9) w.r.t. only $\mathbf{d} \equiv \mathbf{D}_{\rho:}$ results the cost $J_\rho^\tau(\mathbf{d}) =$

$$N^{-1} \|\hat{\mathbf{E}}_{\rho:}^{\tau-1} \mathbf{Y} - \mathbf{d} \cdot \mathbf{X}\|_2^2 + \lambda_{D,\rho} \|\mathbf{d}\|_1 + \phi_D \|\mathbf{d}\|_2 + \varepsilon \|\mathbf{d} \cdot \hat{\Sigma}_x \mathbf{D}^T - \mathbf{I}_{\rho:}\|_F^2 \quad (50)$$

The problem in (50) can be solved through BCD, splitting the cost into p_1 subproblems, in which each subproblem concerns one entry in $\mathbf{d}$, i.e., $\mathbf{D}(\rho, \beta)$ while fixing the rest entries to their most up-to-date values. Let τ_3 denote the nested BCD cycle index, and denote $\check{\mathbf{D}}^{\tau, \tau_3}(\rho, \beta)$ as the update for $\mathbf{D}(\rho, \beta)$ at τ_3 th nested BCD cycle within cycle τ .

Following similar steps as in (I), it can be proved that as τ_3 goes to infinity, the updates $\check{\mathbf{D}}^{\tau, \tau_3}(\rho, \beta)$ will converge to a stationary point of the cost in (50) (Similar claims can be established for the $\mathbf{E}$ matrix). Now, let $\check{\mathbf{D}}_{\rho:}^\tau$ denote the minimizer of (50) (Similar for $\check{\mathbf{E}}_{\rho:}^\tau$). Note that in Alg. 1 proposed here, instead of considering (50), the updates in (12) [and (22)] are obtained after making the approximation $\varepsilon \|\mathbf{D} \hat{\Sigma}_x \mathbf{D}^T - \mathbf{I}\|_F^2$ with $\varepsilon \|\mathbf{D} \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T - \mathbf{I}\|_F^2$ in (12) [similar approximations are applied in (22)].

Notice that, the optimizer $\hat{\mathbf{D}}_{\rho:}^\tau$ in (12) is obtained from the update of $\hat{\mathbf{D}}_{\rho:}^{\tau, \kappa, \tau_2}$ when $\kappa \rightarrow \infty$ and $\tau_2 \rightarrow \infty$ [similarly for $\hat{\mathbf{E}}_{\rho:}^\tau$]. Further, the minimizer of (50) is obtained from the update of $\check{\mathbf{D}}_{\rho:}^{\tau, \tau_3}$ as τ_3 goes to infinity [similarly for the notation $\check{\mathbf{E}}_{\rho:}^\tau$].

(III) Next, it is demonstrated that the updates obtained here from (50), say $\check{\mathbf{D}}_{\rho:}^\tau$, can be arbitrarily close to the optimizer of (12), say $\hat{\mathbf{D}}_{\rho:}^\tau$, i.e., $\|\hat{\mathbf{D}}_{\rho:}^\tau - \check{\mathbf{D}}_{\rho:}^\tau\|_2 \leq \delta(\varepsilon)$, where $\delta(\varepsilon)$ is a nonnegative quantity for which $\lim_{\varepsilon \rightarrow 0} \delta(\varepsilon) = 0$.

Assume that at BCD iteration $\tau - 1$ it holds that $\|\hat{\mathbf{D}}_{\rho:}^{\tau-1} - \check{\mathbf{D}}_{\rho:}^{\tau-1}\|_2 \leq \delta'(\varepsilon)$ and $\|\hat{\mathbf{E}}_{\rho:}^{\tau-1} - \check{\mathbf{E}}_{\rho:}^{\tau-1}\|_2 \leq \delta'(\varepsilon)$, where $\lim_{\varepsilon \rightarrow 0} \delta'(\varepsilon) = 0$. Then, using the cost in (50), the

$$\text{cost in (12) can be written as } J_\rho^\tau(\mathbf{d}) + \phi(\mathbf{d}, \varepsilon) \quad (51)$$

where $\phi(\mathbf{d}, \varepsilon) := \varepsilon (\|\mathbf{d} \cdot \hat{\Sigma}_x (\hat{\mathbf{D}}^{\tau-1})^T - \mathbf{I}_{\rho:}\|_F^2 - \|\mathbf{d} \cdot \hat{\Sigma}_x \mathbf{D}^T - \mathbf{I}_{\rho:}\|_F^2) + \delta'(\varepsilon) \cdot \Delta_\tau$, where Δ_τ is a finite coefficient. The continuity argument in [15, page 15] and (51) implies that, for any $\delta(\varepsilon)$, we can select ε sufficiently small such that $\|\hat{\mathbf{D}}_{\rho:}^\tau - \check{\mathbf{D}}_{\rho:}^\tau\|_2 \leq \delta(\varepsilon)$, where $\lim_{\varepsilon \rightarrow 0} \delta(\varepsilon) = 0$. Thus, by induction it follows that $\|\hat{\mathbf{D}}_{\rho:}^\tau - \check{\mathbf{D}}_{\rho:}^\tau\| \leq \delta(\varepsilon)$ and $\|\hat{\mathbf{E}}_{\rho:}^\tau - \check{\mathbf{E}}_{\rho:}^\tau\| \leq \delta(\varepsilon)$ for any τ .

(IV) Then, we will show that the updates $\check{\mathbf{D}}^\tau(\rho, \beta)$ for $\beta = 1, 2, \dots, p_1$, using (50) (similarly the updates $\{\check{\mathbf{E}}^\tau(\rho, j)\}_{j=1}^{p_2}$) converge to a stationary point of the cost in (9).

Then, the outcomes in (I), (II), (III), and (IV) will be used here to establish Prop.1.

Let $g(\{\mathbf{D}_{\rho}, \mathbf{E}_{\rho}\}_{\rho=1}^q)$ denote the R-CCA cost given in (9), which is defined over $\mathbb{R}^{(qp_1+qp_2) \times 1}$ and

$$g_0(\{\mathbf{D}_{\rho}, \mathbf{E}_{\rho}\}_{\rho=1}^q) := v \|\mathbf{E} \hat{\Sigma}_y \mathbf{E}^T - \mathbf{I}\|_F^2 + \varepsilon \|\mathbf{D} \hat{\Sigma}_x \mathbf{D}^T - \mathbf{I}\|_F^2 \quad (52)$$

$$+ N^{-1} \sum_{t=0}^{N-1} \|\mathbf{E} \mathbf{y}(t) - \mathbf{D} \mathbf{x}(t) - \hat{\mu}\|_2^2 + \phi_D \sum_{\rho=1}^q \|\mathbf{D}_{\rho}\|_2 + \phi_E \sum_{\rho=1}^q \|\mathbf{E}_{\rho}\|_2$$

Furthermore, let's consider the level set

$$\mathfrak{C}^0 := \{ \{\mathbf{D}_{\rho}, \mathbf{E}_{\rho}\}_{\rho=1}^q : g(\{\mathbf{D}_{\rho}, \mathbf{E}_{\rho}\}_{\rho=1}^q) \leq g(\check{\mathbf{D}}^0, \check{\mathbf{E}}^0) \}$$

where $\check{\mathbf{D}}^0$ and $\check{\mathbf{E}}^0$ are used to initialize the updates obtained from (50), and selected such that $\|\check{\mathbf{D}}^0\|_1 < \infty$, $\|\check{\mathbf{E}}^0\|_1 < \infty$, from which it follows that $g(\check{\mathbf{D}}^0, \check{\mathbf{E}}^0) < \infty$. Further, from the form of $g(\cdot)$ it follows that any $\mathbf{D}, \mathbf{E}$ in $\mathfrak{C}^0$ satisfy $\sum_{\rho=1}^q \lambda_{E,\rho} \|\mathbf{E}_{\rho}\|_1 + \sum_{\rho=1}^q \lambda_{D,\rho} \|\mathbf{D}_{\rho}\|_1 \leq g(\hat{\mathbf{D}}^0, \hat{\mathbf{E}}^0) < \infty$. Thus, (p1) the set $\mathfrak{C}^0$ is compact (closed and bounded). Moreover, (p2) note that $g(\cdot)$ is continuous on $\mathfrak{C}^0$.

As mentioned earlier, when the nested BCD iterations τ_3 go to infinity, the iterates $\check{\mathbf{D}}^{\tau, \tau_3}(\rho, \beta)$ will converge to a stationary point of the cost in (50). Further, (p3) we can always ensure the uniqueness of minimizer of the cost in (50) per iteration.

Also, (p4) function $g(\cdot)$ is regular at the unique minimizer of the cost in (50), which is outlined in [35, (A1)]. Specifically, the domain of function $g_0(\cdot)$ is formed by matrices which satisfy that $\mathbf{D}(\alpha, \beta) \in (-\infty, +\infty)$ and $\mathbf{E}(\alpha, \beta) \in (-\infty, +\infty)$. Then, the domain(g_0) = $(-\infty, +\infty)^{(qp_1+qp_2) \times 1}$ is an open set. Moreover, $g_0(\cdot)$ is Gâteaux-differential over domain(g_0). The properties (p1), (p2), (p3) and (p4) ensure the iterates $\check{\mathbf{D}}^{\tau}(\rho, \beta)$ and $\check{\mathbf{E}}^{\tau}(\rho, j)$ will converge to a stationary point of $g(\cdot)$ [35, Thm. 4.1 (c)]. Thus the updates $\hat{\mathbf{D}}^{\tau, \kappa, \tau_2}(\rho, \beta)$ (and $\hat{\mathbf{E}}^{\tau, \kappa, \tau_2}(\rho, \beta)$) in (19) and (23) will be $\delta(\varepsilon)$ -close to a stationary point of the cost in (9) as τ, κ , and τ_2 go to infinity using (II), (III), (IV).

Appendix C. Derivation of ADMM recursions in (30): The auxiliary variables $z_{i,t,\rho}^{i', \tau, \kappa-1}$ for $i' \in \mathcal{N}_i^{\mathfrak{A}}$ and $t = 0, 1, \dots, N-1$ are introduced to substitute the equality constraints in (29) with $\eta_{i,t,\rho}^{\tau, \kappa-1} = z_{i,t,\rho}^{i', \tau, \kappa-1}$, and $\eta_{i',t,\rho}^{\tau, \kappa-1} = z_{i,t,\rho}^{i', \tau, \kappa-1}$ for $i' \in \mathcal{N}_i^{\mathfrak{A}}$ and $i \neq i'$. Then, the augmented Lagrangian function can be formed as

$$\begin{aligned} \mathcal{L}[\{\eta_{i,t,\rho}^{\tau, \kappa-1}\}_{i=1}^{p_1}, v, w, z] &= \sum_{i=1}^{p_1} \|\eta_{i,t,\rho}^{\tau, \kappa-1} - p_1 \hat{\mathbf{D}}^{\tau, \kappa-1}(\rho, i) \bar{\mathbf{x}}_t(i)\|_2^2 \\ &+ \sum_{i=1}^{p_1} \sum_{i' \in \mathcal{N}_i^{\mathfrak{A}}} [v_{i,t,\rho}^{i', \tau, \kappa-1} (\eta_{i,t,\rho}^{\tau, \kappa-1} - z_{i,t,\rho}^{i', \tau, \kappa-1}) + w_{i,t,\rho}^{i', \tau, \kappa-1} (\eta_{i,t,\rho}^{\tau, \kappa-1} - z_{i',t,\rho}^{i', \tau, \kappa-1})] \\ &+ 0.5c_2 \sum_{i=1}^{p_1} \sum_{i' \in \mathcal{N}_i^{\mathfrak{A}}} [\|\eta_{i,t,\rho}^{\tau, \kappa-1} - z_{i,t,\rho}^{i', \tau, \kappa-1}\|_2^2 + \|\eta_{i',t,\rho}^{\tau, \kappa-1} - z_{i',t,\rho}^{i', \tau, \kappa-1}\|_2^2] \end{aligned} \quad (53)$$

where v and w contain the Lagrange multipliers $v_{i,t,\rho}^{i',\tau,\kappa-1}$ and $w_{i,t,\rho}^{i',\tau,\kappa-1}$ corresponding to the constraints $\eta_{i,t,\rho}^{\tau,\kappa-1} = z_{i,t,\rho}^{i',\tau,\kappa-1}$ and $\eta_{i,t,\rho}^{\tau,\kappa-1} = z_{i',t,\rho}^{i,\tau,\kappa-1}$, respectively. Solving (29) involves three steps: Step 1 uses gradient ascent iterations to update the Lagrange multipliers; Step 2 updates $\eta_{i,t,\rho}^{\tau,\kappa-1}$ by minimizing (29) w.r.t. $\eta_{i,t,\rho}^{\tau,\kappa-1}$ while treating the rest variables as fixed; and Step 3 minimizes (29) w.r.t. $z_{i,t,\rho}^{i',\tau,\kappa-1}$ while fixing the other variables. After eliminating the redundant variables, the subproblems (30) for $t = 0, \dots, N-1$ will be tackled through updating the sensor j 's local estimate, $\eta_{j,t,\rho}^{\tau,\kappa-1}$, along with the Lagrange multipliers $\{v_{j,t,\rho}^{j',\tau,\kappa-1}\}_{j' \in \mathcal{N}_j^a}$. Thus, sensor j is responsible for carrying out the updating recursions in (30)-(31).

9. References

References

- [1] A. Aduroja, I. D. Schizas, and V. Maroulas, "Distributed Principal Component Analysis in Sensor Networks," in *Proc. of the Intl. Conf. on Acoust., Speech and Sig. Proc.*, Vancouver, BC, pp. 5850–5854, May 2013.
- [2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Convex Optimization with Sparsity-Inducing Norms," *Optimization for Machine Learning*, MIT Press, 2011.
- [3] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Generative Model-Based Clustering of Directional Data," *Proc. of ACM SIGKDD Intl. Conf. on Knowledge Disc. and Data Mining*, Washington, DC, pp. 19–28, 2003.
- [4] J. M. Bruce and P. L. Dragotti, "Reconstructing Diffusion Fields Sampled with a Network of Arbitrarily Distributed Sensor," *Proc. of 22nd European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, Sep. 2014.
- [5] D. R. Brillinger, *Time Series: Data Analysis and Theory*. , Holden Day, 1981.
- [6] D. P. Bertsekas, *Nonlinear Programming*. 2nd Edition, Athena Scientific, Massachusetts, 1999.
- [7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends, in Machine Learning*, 3(1), pp. 1–122, 2010.
- [8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. "Convex Optimization with Sparsity-inducing Norms," *Optimization for Machine Learning*, MIT press, 2011.

- [9] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. 2nd Edition, Athena Scientific, Massachussets 1997.
- [10] M. Borga and H. Knutsson, "A Canonical Correlation Approach to Blind Source Separation", *Department of Biomedical Engineering, Linkping University, Tech. Rep. LiU-IMT-EX-0062*, 2001.
- [11] J. Chen and I. D. Schizas, "Distributed Sparse Canonical Correlation Analysis in Clustering Sensor Data," *Proc. of the Asilomar Conf. on Signals, Systems and Comp.*, Pacific Grove, CA , Nov. 2013.
- [12] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li and C. Maple, "A Survey of Localization in Wireless Sensor Network," *Intl. Journal of Distributed Sensor Networks*, vol. 2012, doi:10.1155/2012/962523, pp. 1–12, 2012.
- [13] M. M. Chiang and B. Mirkin, "Intelligent Choice of the Number of Clusters in k-Means Clustering: An Experimental Study with Different Cluster Spreads," *Journal of Classification*, vol. 27, no. 3, pp. 3–40, 2010.
- [14] N. M. Correa, Y. O. Li, and T. Adali, "Canonical Correlation Analysis for Feature-based Fusion of Biomedical Imaging Modalities and its Application to Detection of Associative Networks in Schizophrenia", *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 6, pp. 998-1007, 2008.
- [15] A. V. Fiacco, "Introduction to Sensitivity and Stability Analysis in Nonlinear Programming", Academic Press, 1983.
- [16] H. Hotelling, "Relations Between Two Sets of Variates," *Biometrika*, vol. 28, pp. 321–377, 1936.
- [17] D. R. Hardoon and J. Taylor, "The Double-Barrelled Lasso," in *Learning from Multiple Sources Workshop, Advances on Neural Information Processing Systems*, Vancouver, Canada, 2008.
- [18] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning", Springer, 2009.
- [19] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, 31(3), pp. 264–323, 1999.
- [20] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

- [21] S. Kim, K. A. Sohn, and E. P. Xing, "A Multivariate Regression Approach to Association Analysis of a Quantitative Trait Network," *Bioinformatics*, 25(12), pp. 204–212, 2009.
- [22] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
- [23] Y. Liu, Y. He, M. Le, J. Wang, K. Liu, L. Mo, W. Doing, Z. Yang, M. Xi, J. Zhao, and X. Y. Li, "Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs," *Proc. of the IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)*, Shanghai, China, pp. 873–881, 2011.
- [24] D. Lin, J. Zhang, J. Li, V. D. Calhoun, H.-W. Deng and Y.-P. Wang, "Group Sparse Canonical Correlation Analysis for Genomic Data Integration," *BMC bioinformatics*, vol. 14, no. 1, pp. 245, 2013.
- [25] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Boca Raton, FL: Chapman and Hall/CRC, 2005.
- [26] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Y. Li, and Y. Liu, "City-See: Urban CO2 Monitoring with Sensors," *Proc. IEEE Int. Conf. Comput. Commun.*, pp. 1611–1619, 2012.
- [27] A. T. Puig, A. Wiesel, and A. O. Hero, "Multidimensional Shrinkage-thresholding Operator and Group LASSO Penalties," *IEEE Signal Process. Lett.*, vol. 18, pp. 363–366, Jun. 2011.
- [28] E. Parkhomenko, D. Tritchler, and J. Beyene, "Sparse Canonical Correlation Analysis with Application to Genomic Data Integration," *Statistical Applications in Genetics and Molecular Biology*, 8, pp. 1–34, 2009.
- [29] G. Ren, I. D. Schizas and V. Maroulas, "Joint Sensors-Sources Association and Tracking," *Proc. of the IEEE Sensor Array and Multichannel Signal Processing Workshop*, A Coruna, Spain, June 22-25, 2014.
- [30] S. Simiè and S. Sastry, "Distributed Environmental Monitoring Using Random Sensor Networks," *Proc. of the 2nd Intl. Workshop on Info. Proc. in Sensor Nets.*, Palo Alto, CA, USA, pp. 582–592, 2003.
- [31] I. D. Schizas and G. B. Giannakis, "Covariance Eigenvector Sparsity for Data Compression and Denoising," *IEEE Trans. on Signal Processing*, vol. 60, no. 5,

pp. 2408–2421, May 2012.

- [32] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in Ad Hoc WSNs with Noisy Links - Part I: Distributed Estimation of Deterministic Signals,” *IEEE Trans. on Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.
- [33] J. Sui, H. He, G. D. Pearlson, T. Adali, K. A. Kiehl, Q. Yu, V. P. Clark, E. Castro, T. White, B. A. Mueller, B. C. Ho, N. C. Andreasen, V. D. Calhoun, “Three-way (N-way) Fusion of Brain Imaging Data based on mCCA+ jICA and its Application to Discriminating Schizophrenia”, *Neuroimage*, vol. 66, pp. 119–132, 2013.
- [34] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [35] P. Tseng, “Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization,” *Journal of Opt. Theory and Applications*, vol. 109, no. 3, pp. 475–494, Jun. 2001.
- [36] D. M. Witten, R. Tibshirani, and T. Hastie, “A Penalized Matrix Decomposition, with Applications to Sparse Principal Components and Canonical Correlation Analysis,” *Biostatistics*, vol. 10, no. 3, pp. 515–534, 2009.
- [37] A. Wiesel, M. Kliger, and A. Hero, “A Greedy Approach to Sparse Canonical Correlation Analysis,” Technical Report, University of Michigan, available in arXiv:0802.2748, 2008.
- [38] S. Waaijenborg, P. C. V. de. W. Hamer, and A. H. Zwinderma, “Quantifying the Association between Gene Expressions and DNA-markers by Penalized Canonical Correlation Analysis,” *Statistical Applications in Genetics and Molecular Biology*, vol. 7, no. 1, Article 3, 2008.
- [39] R. Xu and D. Wunsch, II, “Survey of Clustering Algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
- [40] M. Yuan and Y. Lin, “Model Selection and Estimation in Regression with Grouped Variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [41] H. Zou, T. Hastie, and R. Tibshirani, “Sparse Principal Component Analysis,” *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, 2006.
- [42] D. Zumoffen and M. Basualdo, “From Large Chemical Plant Data to Fault Diag-

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

785

nosis Integrated to Decentralized Fault Tolerant Control: Pulp Mill Process Appli-
cation,” *Industrial and Engineering Chemistry Research*, vol. 47, 2007.

Algorithm 1: CR-CCA

```

790 Initialize  $\mathbf{D}^{(0)}$  and  $\mathbf{E}^{(0)}$  randomly. Initialize  $\mathbf{b}_\rho^0, \check{\mathbf{b}}_\rho^0, \mathbf{p}_\rho^0, \check{\mathbf{p}}_\rho^0$  to  $\mathbf{0}$  for  $\rho = 1, 2, \dots, q$ .
  for  $\tau = 1, 2, \dots$ , do
    for  $\rho = 1, 2, \dots, q$  do
      for  $\kappa = 1, 2, \dots, K$  do
        Update  $\hat{\mathbf{D}}^{\tau, \kappa}(\rho, \beta)$ ,  $\mathbf{b}_\rho^{\tau, \kappa}$ , and  $\mathbf{p}_\rho^{\tau, \kappa}$  via (19), (21), and (17) for  $\beta = 1, \dots, p_1$ .
        795 Update  $\hat{\mathbf{E}}^{\tau, \kappa}(\rho, j)$ ,  $\check{\mathbf{b}}_\rho^{\tau, \kappa}$ , and  $\check{\mathbf{p}}_\rho^{\tau, \kappa}$  via (23), (24), and (25) for  $j = 1, \dots, p_2$ .
      end for
    end for
    Set  $\hat{\mathbf{D}}_{\rho:}^{\tau+1} = \hat{\mathbf{D}}_{\rho:}^{\tau, K}$  and  $\hat{\mathbf{E}}_{\rho:}^{\tau+1} = \hat{\mathbf{E}}_{\rho:}^{\tau, K}$  for  $\rho = 1, \dots, q$ .
    If  $\|\hat{\mathbf{D}}^{\tau+1} - \hat{\mathbf{D}}^\tau\|_F + \|\hat{\mathbf{E}}^{\tau+1} - \hat{\mathbf{E}}^\tau\|_F < \epsilon$  for a prescribed tolerance  $\epsilon$ , then break.
    800 end for

```

Algorithm 2: DR-CCA

805 Initialize $\{\mathbf{d}_j^0 \in \mathbb{R}^{q \times 1}$ for $j \in \mathfrak{A}$ and $\mathbf{e}_i \in \mathbb{R}^{q \times 1}$ for $i \in \mathfrak{B}$ randomly. Initialize
 scalars $\mathbf{b}_\rho^0(j) = \check{\mathbf{b}}_\rho^0(i) = \mathbf{p}_\rho^0(j) = \check{\mathbf{p}}_\rho^0(i) = 0$ for $\rho = 1, 2, \dots, q$, $j = 1, 2, \dots, p_1$ and
 $i = 1, 2, \dots, p_2$.
for $\tau = 1, 2, \dots$, **do**
 for $\rho = 1, 2, \dots, q$ **do**
 810 **for** $\kappa = 1, 2, \dots, K$ **do**
 Sensor $j \in \mathfrak{A}$ and $i \in \mathfrak{B}$ form estimates $\{\hat{\eta}_{j,t,\rho}^{\tau,\kappa-1}\}_{t=0}^{N-1}$ and $\{\hat{\nu}_{i,t,\rho}^{\tau,\kappa-1}\}_{t=0}^{N-1}$,
 via K_2 network ADMM iterations.
 Update $\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j)$, $\mathbf{b}_{\rho}^{\tau,\kappa}(j)$ and $\mathbf{p}_{\rho}^{\tau,\kappa}(j)$ via (35), (38) and (39) respectively
 for $j = 1, \dots, p_1$ and $j \in \mathfrak{A}$.
 815 Update $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)$, $\check{\mathbf{b}}_{\rho}^{\tau,\kappa}(i)$ and $\check{\mathbf{p}}_{\rho}^{\tau,\kappa}(i)$ via (43), (44) and (45) respectively
 for $i = 1, \dots, p_2$ and $i \in \mathfrak{B}$.
 end for
 end for
 If $\max_{j \in \mathfrak{A}} \|\hat{\mathbf{d}}_j^\tau - \hat{\mathbf{d}}_j^{\tau-1}\|_2 < \epsilon$ and $\max_{i \in \mathfrak{B}} \|\hat{\mathbf{e}}_i^\tau - \hat{\mathbf{e}}_i^{\tau-1}\|_2 < \epsilon$ for a prescribed
 820 tolerance ϵ , **then break**.
end for

Algorithm 3: Sparsity-Controlling Coefficient Selection

825

- Find the upper and lower bounds $\hat{M}_u$ and $\hat{M}_l$.
 Use $CPV_{\hat{M}_l} = 0.75$, $CPV_{\hat{M}_u} = 0.95$.
- Estimate the zero-column index sets of $\mathbf{D}$, $\mathbf{E}$: $\mathcal{C}_D$ and $\mathcal{C}_E$.
 Initialize: $\phi_D = \phi_E = 0$, $\{\lambda_{D,\rho} = \lambda_{E,\rho} = 0\}_{\rho=1}^q$.
 830 Step 2.1) Find estimates λ_D^{max} , λ_E^{max} , namely $\hat{\lambda}_D^m$, $\hat{\lambda}_E^m$.
 Step 2.2) Use $\omega_1 \hat{\lambda}_D^m$, $\omega_1 \hat{\lambda}_E^m$ into CR-CCA (or DR-CCA).
 Determine $\mathcal{C}_D$ and $\mathcal{C}_E$ of $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ obtained at Step 2.2.
- Adjust ϕ_D , ϕ_E , and $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^q$ alternatively.
 Initialize $\phi_D = \phi_E = \lambda_{D,\rho} = \lambda_{E,\rho} = 0$ for $\rho = 1, \dots, q$.
 835 Find $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$ using CR-CCA (or DR-CCA) and determine sets $\mathcal{C}_{D,c}$ and $\mathcal{C}_{E,c}$.
 3.1) Find proper $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^q$ resulting sets $\mathcal{C}_D$ and $\mathcal{C}_E$.
while ($\mathcal{C}_{D,c} \subset \mathcal{C}_D$ or $\mathcal{C}_{E,c} \subset \mathcal{C}_E$)
 If $\mathcal{C}_{D,c} \subset \mathcal{C}_D$: $\{\lambda_{D,\rho} = \lambda_{D,\rho} + \Delta_\lambda\}_{\rho=1}^q$;
 If $\mathcal{C}_{E,c} \subset \mathcal{C}_E$: $\{\lambda_{E,\rho} = \lambda_{E,\rho} + \Delta_\lambda\}_{\rho=1}^q$;
 840 Run CR-CCA (or DR-CCA) and update $\mathcal{C}_{D,c}$ and $\mathcal{C}_{E,c}$.
end
 3.2) Increase ϕ_D and ϕ_E gradually (using step size Δ_ϕ) and run CR-CCA (or DR-CCA) to update $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$, and M_c , until $\hat{M}_l \leq M_c \leq \hat{M}_u$. Initialize $\mathcal{R}_c$ as the set of the common nonzero row indices in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$.
 845 3.3) Find pertinent $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$.
while ($\hat{M}_l \leq M_c \leq \hat{M}_u$)
 while (true)
 (i) **If** $\rho \in \mathcal{R}_c$ and $\hat{\mathbf{D}}_\rho \neq \mathbf{0}$, **then** $\lambda_{D,\rho} = \lambda_{D,\rho} + \Delta_\lambda$, for $\rho = 1, \dots, q$.
 (ii) **If** $\rho \in \mathcal{R}_c$ and $\hat{\mathbf{E}}_\rho \neq \mathbf{0}$, **then** $\lambda_{E,\rho} = \lambda_{E,\rho} + \Delta_\lambda$, for $\rho = 1, \dots, q$.
 850 (iii) Run CR-CCA (or DR-CCA), and update $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$, $\tilde{\mathcal{R}}_D$, $\tilde{\mathcal{R}}_E$.
 (iv) **If** $\tilde{\mathcal{R}}_D = \tilde{\mathcal{R}}_E = \mathcal{R}_c$
 If $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$ are *robust*, **then STOP**.
 else, **ExitFlag** = 1, and **Break while**.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

855

860

```

    end
end
(v) If the number of the common non-zero rows in  $\hat{\mathbf{D}}$  and  $\hat{\mathbf{E}} < |\mathcal{R}_c|$ , ExitFlag = 2
    and Break while.
end
If ExitFlag = 2, then update  $M_c = |\mathcal{R}_c|$ .
If ExitFlag = 1, then increase  $\phi_D, \phi_E$  gradually, i.e.,  $\phi_D = \phi_D + n \cdot \Delta_\phi$ ,
 $\phi_E = \phi_E + n \cdot \Delta_\phi$  for  $n=1,2,\dots$ , and use them in CR-CCA (or DR-CCA), until
the number of the common non-zero rows in  $\hat{\mathbf{D}}, \hat{\mathbf{E}} < |\mathcal{R}_c|$ . Update  $M_c = |\mathcal{R}_c|$ .
end
```

$ \mathcal{N}_j^{\mathfrak{A}} $	$j = 1, 2, \dots, 30$														
	7	7	11	10	9	10	11	12	10	10	13	14	14	6	10
	7	13	6	9	16	9	5	5	13	10	8	9	8	9	11
$ \mathcal{N}_i^{\mathfrak{B}} $	$i = 1, 2, \dots, 30$														
	7	7	16	11	11	11	10	13	8	6	7	13	5	8	8
	4	8	12	14	11	16	5	14	11	13	11	9	12	17	16

Table 1: The cardinality of $\mathcal{N}_j^{\mathfrak{A}}$ and $\mathcal{N}_i^{\mathfrak{B}}$ for $i = 1, \dots, 30$ and $j = 1, \dots, 30$.

Figure 1: Diffusion field examples (top, center) sensed by a heterogeneous sensor network with two types of sensing units (bottom).

Figure 2: Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 60$ sensors.

Figure 3: Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 20$ sensors.

Figure 4: Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 30$ sensors.

Figure 5: Probability of correctly clustering sensors vs. number of data vectors for different number of ADMM iterations.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Figure 6: Probability of correctly clustering sensors vs. sensing SNR.

Figure 7: Probability of correctly clustering sensors vs. the average number of scalars communicated per sensor for $N = 500$ measurements (left) and $N = 1000$ measurements (right).