REGULARIZED CANONICAL CORRELATIONS
IN CLUSTERING SENSOR DATA

by

JIA CHEN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2016

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Ioannis D. Schizas for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. I would also like to extend my appreciation to Dr. Ioannis D. Schizas for providing financial support for my doctoral studies, and taking the time to critically evaluate this dissertation. His enthusiasm and attitude toward research have deeply influenced me and helped me to do better research. I wish to thank my academic advisors Dr. Jonathan Bredow, Dr. Michael T. Manry, Dr. Qilian Liang, Dr. Kamisetty R. Rao and Dr. Saibun Tjuatja for their interest in my research and for taking time to serve in my comprehensive exam and thesis defense committee.

I am grateful to all the teachers who taught me during the years I spent in school, first in China, then in the Unites States. I would like to thank Prof. Wei Guo and Prof. Li Hao for encouraging me and inspiring me to pursue graduate studies.

I am also grateful to my labmates, Guohua, Akshay and Kazi for their interest in my research and for the helpful discussions and invaluable comments.

Finally, I would like to express my deep gratitude to my parents and Weize Yu who have encouraged me, inspired me and supported my studies unconditionally. I also thank several of my friends who have helped me throughout my career.

April 15, 2016

ABSTRACT

REGULARIZED CANONICAL CORRELATIONS
IN CLUSTERING SENSOR DATA

JIA CHEN, Ph.D.

The University of Texas at Arlington, 2016

Supervising Professor: Ioannis D. Schizas

In many data acquisition applications such as in sensor networks, the acquired sensor measurements contain information about multiple sources placed at different spatial locations. Such sources could correspond to different e.g., thermal sources or transmitters placed at different locations inside the sensed field. Before applying any statistical inference task, it is essential to identify which groups of sensors acquire observations that contain information about the same sources. This is essential to avoid 'mixing' observations that contain information about uncorrelated sources. In this thesis, the goal is to cluster sensors into different groups based on their source information content about the field sources and isolate sensors acquiring only noise. Two scenarios are considered in this thesis, in one of which the number of sources is given to the sensors and in the other scenario, the number of sources is unknown.

Toward this end, for the first scenario, a novel canonical correlation analysis (CCA) framework equipped with sparsity-inducing norm-one regularization is introduced to identify correlated sensor measurements and identify informative groups of sensors. It is established that the novel framework is capable to cluster sensors, based on their source content, correctly (with probability one) even in nonlinear settings and when sources do not overlap. Block coordinate techniques (BCD) are employed to derive a centralized algorithm that minimizes the sparsity-aware CCA framework. The latter framework is reformulated as a separable optimization program which is tackled in a distributed fashion via the alternating direction method of multipliers (ADMM). A computationally efficient online distributed algorithm is further derived that is capable to process sensor data online. Extensive numerical tests corroborate that the novel techniques outperform existing alternatives

Furthermore, in the second scenario where the number of the sources is not available, two strategies are provided. One strategy is that the traditional canonical correlation analysis (CCA) framework is equipped with norm-one and norm-two regularization terms in order to cluster the sensor data while determining the number of field sources. ADMM and BCD techniques are utilized to derive centralized and distributed algorithms tackling the proposed regularized CCA framework. The capability of correct clustering of sensors in the novel regularized CCA algorithm is verified in heterogeneous sensing systems, consisting of sensors with different sensing capabilities, offering flexibility and providing multiple views of the sensed field by acquiring different types of measurements. The other strategy is that principal component analysis (PCA) combined with moving-average (MA) filtering is utilized to eliminate sensing noise variance and extract the number of principal components in the sensor data covariance corresponding to the uncorrelated sources. Given the estimated number of sources, two applications are considered. In the first application, a novel communication efficient scheme for reconstructing a field sensed by spatially scattered sensors is put forth, which relies on norm-one regularized CCA, PCA, as well as normalized least mean-square adaptive filtering. In the second application, a multiset CCA (M-CCA) framework is proposed to uncover information in multiple heterogeneous sensor

data sets and cluster sensors according to their source content.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

Figure Page

CHAPTER 1

INTRODUCTION

Data analysis has become ubiquitous in the sense that our living environment involves devices that have the ability of sensing and processing data. Fields such as machine learning, data mining, statistics and so on, rely on the analysis of sensed data and extraction of information from them. To analyze data sets, it is common to employ clustering, a task which has not received much attention especially in distributed sensor networks.

1.1 Goals of the Thesis

1.1.1 Homogeneous Sensor Data Clustering

Grouping sensors based on their source information content has been considered for linear data models and memoryless sources in see e.g., [65] and references therein. In this dissertation a generalized framework for grouping sensors based on their information content is put forth which is able to deal with nonlinear settings. Interestingly, sensor measurements containing information about the same sources are statistically correlated irrespective of the underlying data model. To exploit such spatial correlations, canonical correlation analysis (CCA), see e.g., [6,33], is combined with sparsity-inducing regularization techniques [70, 87] to obtain a framework that can extract correlated sensor data and cluster them into groups. The principle of CCA that involves linear extraction of common features from two data sets, will be applied in time-shifted data sequences to cluster sensors with similar information content.

The sparsity-aware CCA framework is derived by extending the standard CCA cost with norm-one regularization coefficients that fully exploit the sparsity present in the sensor data (cross-)covariance matrices. Notice that, in the sensed field, only a few sensors are affected by the uncorrelated sources and the remaining sensors measure the noise. Thus, only the measurements from the informative sensors sensing the same sources exhibit statistical correlation, which accounts for the sparsity present in the sensor data (cross-)covariance matrices. Sparsity is an attribute present in many settings and has been extensively used in sparse regression, solving under-determined systems of equations as well as matrix decompositions into sparse factors [19, 64, 70, 72, 86, 87]. The resulting cost is minimized using coordinate descend techniques (e.g., [4]). A centralized algorithm is derived first that is well suited for networks equipped with a fusion center. Then, the alternating direction method of multipliers (ADMM) (see e.g., [5]) is put forth to formulate the novel sparse CCA framework as a separable optimization problem, and then combined with coordinate descent iterations to obtain a totally distributed algorithm that performs sensor clustering. The resulting distributed algorithm will require information exchanges only between single-hop neighboring sensors. Online implementations are also developed that allow real-time processing in settings where sensors are constantly acquiring data. The online schemes offer a more computationally and communication efficient algorithmic alternative compared to their batch counterpart, while compromising some sensor-clustering performance. For an increasing amount of sensor data it is proved that the sparsity-aware CCA framework is capable of perfectly clustering sensors into different groups based on their information (source) content, even in nonlinear settings and when sources do not overlap.

1.1.2 Heterogeneous Sensor Data Clustering

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 [60]. The acquired sensor measurements contain information about different phenomena of interest (with unknown number) such as thermal and/or pollution sources [41, 49, 68], 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 contain information about the present field sources. Before applying any data processing task such as estimation and detection [37] 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 they will identify regions of interest in the sensed field, such as pollution sources [41,49], 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 coming from the cost and energy considerations imposed by GPS equipment, see e.g., [23]. 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.

For the heterogeneous sensor setting, we put forth a new ℓ_1-norm and ℓ_2-norm regularized CCA to cluster sensor measurements into different informative groups and find the unknown number of sources. Different from the ℓ_1/ℓ_2 mixed norm CCA formulation in [77] in which each canonical loading is either penalized by ℓ_2-norm 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_1-norm and ℓ_2-norm regularization terms.

1.1.3 Application of Homogeneous Sensor Data Clustering: Field Reconstruction

An efficient central data fusion scheme will be derived to reconstruct the sensor measurements induced by multiple uncorrelated sources present in the field. Toward this end, a promising framework is proposed that carries out the following three tasks: i) Estimating the number of field sources; ii) Clustering the sensors based on which sources they are observing; and iii) Utilizing only the measurements of a few cluster head sensors to reconstruct *all* sensors' measurements at the fusion center (FC). To achieve the first task, moving-average (MA) filtering [18] is combined with principal component analysis (PCA) [6] to eliminate sensing noise variance and extract the number of principal components (PCs) in the sensor data covariance corresponding to the uncorrelated sources. To group sensors in clusters according to their source information, we realize that sensor measurements containing information about the same sources are statistically correlated. To exploit such spatial correlations, a norm-one regularized canonical correlation analysis technique [15] will be used to extract correlated sensor measurements and group them in clusters. PCA and adaptive filtering will also be utilized to result an improved clustering approach that groups sensors according to their information content. Normalized least mean squares is utilized to enable the FC to reconstruct each cluster's sensor measurements using only data from a few cluster head sensors that communicate their observations to the FC. Different from existing source-based clustering techniques [15, 16] the proposed framework is able to perform accurate clustering even when sensors observe multiple sources. The main idea is to separate the network into clusters of similar information content and use within each cluster the data of only one sensor to reconstruct the remaining sensor measurements at the FC. Such an approach will reduce significantly the number of scalars transmitted from the sensors to the FC potentially extending the sensors' lifespan.

1.1.4 Extension of Heterogeneous Sensor Data Clustering to Multiset Sensor Data

The work of clustering sensor data with two modalities is generalized to cluster more than two types of sensing measurements. Without giving the number of sources, PCA (or distributed PCA) is combined with MA filtering to determine the number of sources. Next, a M-CCA framework, that is simpler in the sense that no fourth-order polynomial terms are produced, will be penalized with norm-one terms to implement clustering of multiple types of sensor data according to their source information. Relying on the BCD and ADMM technologies, the proposed sparse M-CCA (SM-CCA) scheme is developed in both centralized and distributed ways, whose capability of correct clustering sensors is verified through numerical tests.

1.2 Prior Work

CCA is widely used in data analysis to extract the correlated components among two distinct data sets [6, 33]. CCA-based methods provide promising solutions to blind source separation [7, 42], detection of diseases [21], and genomic data integration [43] to name a few. A number of related sparse CCA methods have been proposed. The work in [74] applies the elastic net penalty, see e.g., [88], into standard CCA and derives an iterative regression procedure. It exploits the grouping effect from the ridge regression, and the shrinkage effect from the $Lasso$, see e.g., [70]. However, it does not optimize a given cost. In [29], the standard CCA is reformulated as a ℓ_1-regularized convex framework using a least-squares approach. This work focuses on a special scenario where one data set is in a primal representation (the input space), and another data set is in a dual representation (the kernel space). This approach is limited by the fact that the canonical vector for the dual representation must have nonnegative entries. The work in [77] proposes a sparse CCA scheme based on a forward greedy approach (sequentially picking entries of canonical vectors), in which upper bounds of the number of non-zero entries in the canonical vectors are known. An algorithm for obtaining sparse loadings for CCA iteratively was proposed in [52], while no optimization criterion was specified. The algorithm in [52] was extended to multiple canonical variables in [22], which performs robust estimation of the data covariance matrices. In [79], a penalized matrix decomposition with applications in sparse CCA was developed and the tuning parameters were chosen using cross-validation. The penalized CCA in [79] is extended to supervised sparse CCA in [81], which makes use of the measurement outcomes (e.g., the survival time of a patient in genomic research) to determine whether the canonical loadings obtained here are significant. The work in [20] extends the penalty in [79] to more general forms, including ℓ_1/ℓ_2 mixed-norm penalty, or weighted fusion penalty, see e.g., [36], combined with ℓ_1-norm regularization. However, application of this method requires prior knowledge of the sparsity structure in the canonical loadings. Centralized CCA approaches that work with biomedical data of different modalities can be found in [21, 67].

The aforementioned sparse CCA methods are generally challenged by the facts that either i) only consider one pair of canonical variables; or ii) they have prior information on the sparsity structure of the canonical loadings (vectors); or iii) require applicability of computationally intensive cross-validation to select the sparsity-controlling coefficients; or iv) do not use specified optimization criterion; or v) do not consider heterogeneous data sets from different sensing modalities.

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 [38]. K-means finds 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. To tackle the latter issue an intelligent K-means (iK-means) was 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 [3, 32, 35, 82] often implicitly assume knowledge of the cluster shapes or the multiple cluster configurations which are based on the available similarity measures. Thus, the challenge of applying these clustering approaches in our setting lies in the fact that the similarity between data entries containing information about the same source is unknown due to the different modalities of data sets and the unavailability of the underlying data models.

Typically, sensors are limited in terms of communication and computational capabilities. A straightforward approach to collect the information across all sensors to the FC is to allow each sensor forward its acquired measurements, possibly via multi-hop communications, to the FC. Such a process can place a heavy operational burden in all sensors due to the high communication cost. To tackle such a challenge, data aggregation techniques have become crucial to prolong the overall lifespan of a sensor network. There are four different strategies for data aggregation: i) centralized approaches, ii) in-network aggregation, iii) tree-based approaches, and iv) cluster-based aggregation [54]. The work in [2] considers a dynamic spanning-tree approach to minimize the energy consumption by taking into consideration the data traffic load. Support vector machines are used in [55] to reduce the redundant data and eliminate false data. An efficient cluster-based data aggregation scheme for heterogeneous sensor networks was developed in [45], where inter- (intra-) cluster data aggregation is performed to eliminate redundant data. The cluster-based approach in [57] uses a context-aware approach to validate data, while intra-cluster and inter-cluster redundancy is eliminated when sensors belong to the same cluster or neighboring clusters, respectively, for the validated data. The work in [63] focuses on the issues of accuracy, traffic load, redundancy elimination and delay when performing data aggregation, and proposed a model to address the aforementioned issues.

In the aforementioned cluster-based approaches, the geographical area is divided into multiple grid-based clusters, while the cluster heads are elected as those sensors with the highest energy and largest number of one-hop neighboring sensors, or as the sensors whose positions are closer to the centroid position of the cluster. Further, there is no basic principle in deciding the number of clusters needed. There are fundamental differences with the work proposed here. Specifically, the sensors will be clustered in groups according to their information content and the sources they sense (possibly multiple). Thus, the clusters here are formed based on the sensor information content and not according to an ad hoc splitting of the area monitored. Further, the number of clusters will be determined by the number of the underlying information sources in the monitored field. Last but not least, the clustering proposed here is done to facilitate a form of reduction in the number of data transmitted and achieve accurate reconstruction at the FC.

The field reconstruction problem has been previously addressed in the literature in [8] under the assumption that the monitored field is spatially governed by known partial differential diffusion equations. Distributed schemes to reconstruct two-dimensional diffusion fields are considered in [9]. An interesting work can also be found in [59], where an optimal dimensionality-reduced approximation method was developed to recover thermal maps. Based on Bayesian estimation and Kalman filtering, the papers by [50, 62, 80, 84] consider statistical estimation methods for non-static fields. Algorithms to estimate a single source's parameters are studied in [24,40,48] to fully recover the monitored field. The work in [51] puts forth an algorithm that can successfully reconstruct sensor signals as long as the field bandwidth is sufficiently small, the field adheres to a one-dimensional model and the sensor locations are known to the FC. A distributed cluster-based signal reconstruction for non-bandlimited fields is proposed in [58] by locally adapting the field model. However, the aforementioned schemes either assume the fields are driven by assumed known spatiotemporal diffusion equations or statistical models, or only take a single-source into consideration. Different from the existing methods, our method is attractive since it does not require all sensor to communicate with the FC, it does not require knowledge of the sensors' positions, it can address settings where the field consists of multiple spatially scattered sources and the sources-to-sensors propagation channels may be multipath. There is no need to estimate the source signals, and our focus is on reconstructing the field only in points of interest where sensors are deployed rather than the entire field.

1.3 Advantages of the Proposed Algorithms

Compared to the aforementioned sparse CCA approaches, our proposed regularized CCA exhibits several advantages: i) 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; ii) the related sparsity-controlling coefficients are selected through a computationally efficient heuristic way; iii) a specific minimization criterion is derived in our regularized CCA; iv) heterogeneous data sets from two or more than two different modalities are considered; and v) the proposed framework can obtain multiple pairs of canonical variables.

1.4 Notation

Bold face capital letters, bold face small letters, and normal font letters are respectively used for matrices, vectors and scalars. $\mathbb{E}(\cdot)$ denotes expectation, and $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_F$ denote norm-one, Euclidean norm, and Frobenius norm, respectively. For convenience, the same notation may represent different meanings in different chapters, while it holds the same meaning in the same chapter.

CHAPTER 2

CCA PRELIMINARIES

2.1 Standard CCA

Canonical Correlation Analysis (CCA) is a widely used method to find correlation structures in multi-view datasets, see e.g., [6]. There are many different ways to define CCA. Here we use one of the definitions in [6] since this version is later regularized in our novel sparse CCA formulations. The following proposition in [6] outlines the CCA formulation and solution.

Proposition 1. Let $\mathbf{X}$ be a p_1 vector-valued variate with mean $\boldsymbol{\mu}_x$ and covariance matrix $\mathbf{\Sigma}_x$ and $\mathbf{Y}$ be a p_2 vector-valued variate with mean $\boldsymbol{\mu}_y$ and covariance matrix $\mathbf{\Sigma}_y$. Denote $\mathbf{\Sigma}_{xy}$ and $\mathbf{\Sigma}_{yx}$ as the cross-covariance matrices between $\mathbf{X}$ and $\mathbf{Y}$. Suppose $\mathbf{\Sigma}_x$ and $\mathbf{\Sigma}_y$ are nonsingular. The $\mathbf{D} \in \mathbb{R}^{q \times p_1}$, $\mathbf{E} \in \mathbb{R}^{q \times p_2}$, and $\boldsymbol{\mu} \in \mathbb{R}^{q \times 1}$ with $\mathbf{D}\mathbf{\Sigma}_x\mathbf{D}^T = \mathbf{I}$, $\mathbf{E}\mathbf{\Sigma}_y\mathbf{E}^T = \mathbf{I}$, and $q \leq \min(p_1, p_2)$ that minimize

$$\mathbb{E}\{[\mathbf{EY} - \mathbf{DX} - \boldsymbol{\mu}]^T[\mathbf{EY} - \mathbf{DX} - \boldsymbol{\mu}]\} \tag{2.1}$$

are given by

$$\mathbf{D} = \mathbf{U}_1^T\mathbf{\Sigma}_x^{-1/2}, \ \mathbf{E} = \mathbf{U}_2^T\mathbf{\Sigma}_y^{-1/2}, \tag{2.2}$$

and

$$\boldsymbol{\mu} = \mathbf{E}\boldsymbol{\mu}_y - \mathbf{D}\boldsymbol{\mu}_x, \tag{2.3}$$

where the columns of $\mathbf{U}_1$ are the q principal eigenvectors of $\mathbf{\Sigma}_x^{-1/2}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_y^{-1}\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_x^{-1/2}$, while the columns of $\mathbf{U}_2$ are the q principal eigenvectors of $\mathbf{\Sigma}_y^{-1/2}\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_x^{-1}\mathbf{\Sigma}_{xy}\mathbf{\Sigma}_y^{-1/2}$.

The above proposition reveals that the goal of CCA is to find matrices $\mathbf{D}$ and $\mathbf{E}$ to maximize the correlation of linear combinations $\mathbf{DX}$ and $\mathbf{EY}$. Each row of $\mathbf{DX}$ or $\mathbf{EY}$ is called one *canonical variable*, and the canonical variables are orthogonal to each other. The coefficients for the linear combinations, saying the rows of $\mathbf{D}$ and $\mathbf{E}$, are called *canonical vectors*. The correlation between the canonical variables are called *canonical correlations*.

2.2 Literature Review of Sparse CCA

Recently, sparse representation technology is becoming increasingly popular and important in wide areas, for instance, machine learning [14,17,77], signal processing [64,65], and pattern recognition [34,56,76]. Sparsity inducing CCA is also attractive in theoretical studies [77,79] and practical applications, for example, gene expression [25,74,75,77] and in image classification [89]. As studied in [30], sparse CCA is formulated by representing two data sets in both primal and dual forms. This formulation minimizes the number of features in both primal and dual projections and maximizes the correlation between the two variable sets. By applying ridge penalization and elastic net penalization to the standard CCA, an iterative regression is proposed in [75] to identify candidate genes for incorporation in the pathway completing gene-expression networks. A forward greedy approach is introduced in [77], which deals with the variables sequentially. This algorithm reduces the computational complexity and also copes with high dimensional data through considering regularized covariance matrices. The latter method only considers sparse canonical variates for the first dimension, and is limited by the requirement of available number of zero entries in the canonical vectors.

2.3 Regularized CCA

Inspired by related centralized sparse decomposition approaches [64, 72, 87], along with the aforementioned existing sparse CCA alternatives, we construct a flexible regularization in CCA. When the known number of canonical variates (number of sources) is known, the ℓ_1 norm is used to penalize the canonical vectors, and while given the prior knowledge of the specified number of canonical variates both ℓ_1 and ℓ_2 will be utilized to regularize the canonical vectors. Different from all the existing applications of sparse CCA, we exploit a new application field in clustering data for sensor networks. Unlike the aforementioned sparse CCA algorithms, our regularized CCA is done in both a centralized manner and distributed fashion, where the processing is carried out across spatially scattered sensors via collaboration of neighboring units.

CHAPTER 3

HOMOGENEOUS DATA CLUSTERING

3.1 Problem Formulation

Consider a connected ad hoc network of p sensors which communicate only with neighboring sensors that are located within their transmission range. Let $\mathcal{N}_j$ denotes sensor j's single-hop neighbors for $j = 1, 2, ..., p$. The p sensors monitor a field, which is formed by M spatially uncorrelated zero-mean sources, denoted as the $r \times 1$ vectors, $\mathrm{s}_m(t)$, while $m = 1, 2, ..., M$ and $r \ll p$. The sources are assumed stationary and not necessarily white, thus they may exhibit temporal memory. Each sensor j acquires scalar measurements $\{x_j(t)\}$ during time instances $t = 0, 1, \ldots$ Each sensor measurement contains information about a subset (possibly one) of the M field sources. Thus, every sensor may contain information about a few of the M sources since they are located at different spatial positions. The sensor measurements adhere to the following generic nonlinear model

$$x_j(t) = \textstyle\sum_{m=1}^{M} h_{m,j}(\mathrm{s}_m(t)) + w_j(t) \tag{3.1}$$

where $h_{m,j}(\cdot)$ is a random scalar nonlinear mapping from $\mathbb{R}^r$ to $\mathbb{R}^1$, which will be negligible when sensor j is sufficiently far from source m (can be thought of as an attenuation factor) while $w_j(t)$ denotes white sensing noise with zero-mean.

Let $\boldsymbol{\chi}(t) := [x_1(t) \ldots x_p(t)]^T$ contain the measurements acquired across all sensors. As different sensors are affected by different sources, different entries in $\boldsymbol{\chi}(t)$ contain information of different sources. Let $\mathcal{S}^m$ denote the subset of entries of $\boldsymbol{\chi}(t)$ that contain information about source $\mathrm{s}_m(t)$, and let $\mathcal{S}^0$ denote the subset of sensors whose measurements do not contain information about any of the sources, e.g., they contain just noise.

For example consider a network consisting $p = 12$ sensors that observe a field with $M = 2$ sources, namely $\mathrm{s}_1(t)$ and $\mathrm{s}_2(t)$.

Assume that sensors S_4, S_5, S_6, S_7 sense source $\mathrm{s}_1(t)$, sensors S_8, S_9, S_{10} acquire measurements that are influenced by source $\mathrm{s}_2(t)$, while sensors $S_1, S_2, S_3, S_{11}, S_{12}$ just observe noise or irrelevant data. Thus, $\mathcal{S}^0 = \{1, 2, 3, 11, 12\}$, $\mathcal{S}^1 = \{4, 5, 6, 7\}$ and $\mathcal{S}^2 = \{8, 9, 10\}$. The union of the sensor clusters $\{\mathcal{S}^m\}_{m=1}^{M}$ contains all entries of $\boldsymbol{\chi}(t)$. This chapter aims at solving the problems: P1) identifying the noninformative sensors and informative sensors; and P2) clustering the entries of $\boldsymbol{\chi}(t)$ in groups $\mathcal{S}^m$ where the members of the same group contain information about the same source (within some ambiguity on the source identity). Toward this end, a novel distributed framework combining canonical correlation analysis (CCA) with norm-one regularization is proposed.

Given training data $\{\mathbf{x}(t), \mathbf{y}(t)\} \in \mathbb{R}^{pf \times 1}$ for $t = \{0, ..., N-1\}$ the CCA framework can be used to linearly extract common features from $\mathbf{x}(t)$ and $\mathbf{y}(t)$, see e.g., [6, Chpt. 10]. The training sequences that are going to be considered here are formed as

$$\mathbf{x}(t) = \left[\boldsymbol{\chi}^T(t-1), \boldsymbol{\chi}^T(t-2), ..., \boldsymbol{\chi}^T(t-f)\right]^T \tag{3.2}$$

$$\mathbf{y}(t) = \left[\boldsymbol{\chi}^T(t), \boldsymbol{\chi}^T(t+1), ..., \boldsymbol{\chi}^T(t+f-1)\right]^T \tag{3.3}$$

where the positive integer f denotes the memory length. Note that $\mathbf{x}(t)$ in (3.2) represents the past of $\boldsymbol{\chi}(t)$, and $\mathbf{y}(t)$ spans the future and present of the sensor measurements in $\boldsymbol{\chi}(t)$ with respect to time instant t. Both super-vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$ each of length $p \cdot f$ contain the information of the field sources summarized in $\mathrm{s}(t) := [\mathrm{s}_1^T(t), \ldots, \mathrm{s}_M^T(t)]^T$. Thus, $\mathrm{s}(t)$ can be viewed as the 'common' features present in both $\mathbf{x}(t)$ and $\mathbf{y}(t)$ and can be extracted by finding matrices $\mathbf{E}, \mathbf{D} \in \mathbb{R}^{q \times p \cdot f}$ with $q \leq p \cdot f$ that can be found via the minimization problem, see e.g., [6, 33, Chpt. 10]:

$$(\breve{\mathbf{D}}, \breve{\mathbf{E}}) = \arg\min (N^{-1}) \textstyle\sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}(t) - \mathbf{D}\mathbf{x}(t) - \hat{\mathbf{m}}\|_2^2 \tag{3.4}$$
$$\text{s.to } \mathbf{D}\hat{\mathbf{\Sigma}}_x\mathbf{D}^T = \mathbf{I} \text{ and } \mathbf{E}\hat{\mathbf{\Sigma}}_y\mathbf{E}^T = \mathbf{I},$$

where $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\mathbf{m}}_x$ correspond to the sample-average estimates of the covariance and mean of $\mathbf{x}(t)$, respectively. These estimates can be evaluated as $\hat{\boldsymbol{\Sigma}}_x := N^{-1}\sum_{t=0}^{N-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x)(\mathbf{x}(t) - \hat{\mathbf{m}}_x)^T$, and $\hat{\mathbf{m}}_x := N^{-1}\sum_{t=0}^{N-1}\mathbf{x}(t)$. Further, $\mathbf{I}$ is the identity matrix of proper dimensions. The covariance $\hat{\boldsymbol{\Sigma}}_y$ of $\mathbf{y}(t)$ can be defined in a similar way, while $\hat{\mathbf{m}} := \mathbf{E}\hat{\mathbf{m}}_y - \mathbf{D}\hat{\mathbf{m}}_x$.

The entries of the vectors $\breve{\mathbf{E}}\mathbf{y}(t)$ and $\breve{\mathbf{D}}\mathbf{x}(t)$, can be viewed as estimates of the sources $\mathbf{s}_m(t)$ that are present in both $\mathbf{x}(t)$ and $\mathbf{y}(t)$. Consider the 12-sensor example mentioned earlier, where $M = 2$ scalar sources ($r = 1$) are sensed, while $f = 1$ in forming $\mathbf{x}(t)$ and $\mathbf{y}(t)$ in (3.2). Recall that only entries $\{4, 5, 6, 7\}$ in $\boldsymbol{\chi}(t)$ contain information about $\mathbf{s}_1(t)$, whereas entries $\{8, 9, 10\}$ contain information about $\mathbf{s}_2(t)$ and the remaining just contain noise. Thus, when forming $\mathbf{D}_{1:}\mathbf{x}(t)$ and $\mathbf{E}_{1:}\mathbf{y}(t)$, the first rows $\mathbf{D}_{1:}$ and $\mathbf{E}_{1:}$ can zero-out irrelevant entries in $\mathbf{x}(t)$ and $\mathbf{y}(t)$ respectively without affecting the estimation performance. Specifically, entries $\{4, 5, 6, 7\}$ in $\mathbf{x}(t)$ and $\mathbf{y}(t)$ will contain information about $\mathbf{s}_1(t)$. Thus, the rows $\mathbf{E}_{1:}$ and $\mathbf{D}_{1:}$ can be selected such that they have nonzero entries only in positions $\{4, 5, 6, 7\}$, while the rest 8 entries can be set to zero (sparsity). Similarly, $\mathbf{E}_{2:}$ and $\mathbf{D}_{2:}$ can have nonzero entries only in positions $\{8, 9, 10\}$ which correspond to the entries of the $\mathbf{y}(t)$ and $\mathbf{x}(t)$ vectors that contain information about $\mathbf{s}_2(t)$. The remaining entries in $\mathbf{y}(t), \mathbf{x}(t)$ contain only noise and can be eliminated by setting the corresponding entries in the two rows of $\mathbf{E}$ and $\mathbf{D}$ equal to zero. Thus, by inducing proper sparsity patterns in the rows of $\mathbf{E}$ and $\mathbf{D}$ and recovering their corresponding supports (nonzero entries' indices) someone can identify which entries in $\boldsymbol{\chi}(t)$ acquire information about the same source and perform clustering. Traditional CCA as described in (3.4) is not capable to produce zero entries in $\mathbf{E}$ or $\mathbf{D}$. Toward this end, we put forth a ℓ_1-regularized CCA framework, which induces proper sparsity patterns in each row of $\mathbf{E}$ and $\mathbf{D}$.

3.2 ℓ_1-Regularized Canonical Correlations

In order to isolate noninformative entries in $\boldsymbol{\chi}(t)$ and identify the source-informative groups of entries within $\boldsymbol{\chi}(t)$, here norm-one regularization is incorporated in the standard CCA formulation in (3.4). The idea of utilizing norm-one to induce sparsity is well established in the literature, see e.g., [64, 70, 79, 87]. Pertinent sparse $\mathbf{E}$ and $\mathbf{D}$ matrices can be obtained using the sparsity-inducing CCA (S-CCA) formulation

$$
\begin{aligned}
&(\hat{\mathbf{D}}, \hat{\mathbf{E}}) = \text{arg min}_{\mathbf{D},\mathbf{E}} N^{-1} \textstyle\sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}(t) - \mathbf{D}\mathbf{x}(t) - \hat{\mathbf{m}}\|_2^2 \\
&+ \textstyle\sum_{\rho=1}^{q} \lambda_{E,\rho} ||\mathbf{E}_{\rho:}^T||_1 + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho} ||\mathbf{D}_{\rho:}^T||_1 \\
&+ \upsilon ||\mathbf{E}\hat{\boldsymbol{\Sigma}}_y \mathbf{E}^T - \mathbf{I}||_F^2 + \varepsilon ||\mathbf{D}\hat{\boldsymbol{\Sigma}}_x \mathbf{D}^T - \mathbf{I}||_F^2
\end{aligned}
\tag{3.5}
$$

where $\mathbf{E}_{\rho:}$ and $\mathbf{D}_{\rho:}$ correspond to the ρth row of $\mathbf{E}$ and $\mathbf{D}$ respectively. The sparsity-controlling coefficients $\lambda_{E,\rho}$ and $\lambda_{D,\rho}$ assume positive values and control the number of zero entries in $\mathbf{E}_{\rho:}$ and $\mathbf{D}_{\rho:}$, respectively. Further, the positive penalty coefficients υ and ε entailed in the last two terms in (3.5) are applied to forbid $\hat{\mathbf{D}}, \hat{\mathbf{E}}$ to be zero matrices, while facilitating the applicability of block coordinate descent techniques that will be utilized to derive centralized and distributed algorithms that tackle (3.5).

3.2.1 Centralized S-CCA (CS-CCA)

We first consider a centralized setting where a fusion center can gather all sensor measurements. Note that the cost in (3.5) is nonconvex with respect to (w.r.t.) $\mathbf{D}$ and $\mathbf{E}$. We come around this challenge by utilizing a block coordinate descent (BCD) solver, see e.g., [4, 71]. Specifically, the cost is minimized w.r.t. one entry of $\mathbf{D}$ (or $\mathbf{E}$), while keeping fixed the remaining entries of $\mathbf{D}$ (or $\mathbf{E}$) to their most up-to-date values. During each coordinate descent cycle all the entries of $\mathbf{D}$ and $\mathbf{E}$ will be updated.

Notice that the last two terms in (3.5) will produce fourth-order polynomial terms in the cost function when trying to minimize the latter cost w.r.t. a single entry of $\mathbf{D}$ or

$\mathbf{E}$ while fixing the remaining entries. To simplify the process of solving (3.5), we fix the second $\mathbf{D}$ and $\mathbf{E}$ in the last two terms of (3.5), respectively, to their most up-to-date value during the τth coordinate descent cycle, namely $\hat{\mathbf{D}}^{\tau-1}$ and $\hat{\mathbf{E}}^{\tau-1}$. Specifically, given the estimates $\hat{\mathbf{D}}^{\tau-1}$ and $\hat{\mathbf{E}}^{\tau-1}$ in the beginning of coordinate cycle τ, the minimization problem which is used to estimate the current updates of $\mathbf{D}$ and $\mathbf{E}$ at iteration τ can be formulated as

$$\begin{aligned}(\hat{\mathbf{D}}^{\tau}, \hat{\mathbf{E}}^{\tau}) = \text{arg min}_{\mathbf{D},\mathbf{E}} N^{-1} \textstyle\sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}(t) - \mathbf{D}\mathbf{x}(t) - \hat{\mathbf{m}}\|_2^2 \\ + \textstyle\sum_{\rho=1}^{q} \lambda_{E,\rho} ||\mathbf{E}_{\rho:}^T||_1 + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho} ||\mathbf{D}_{\rho:}^T||_1 \\ + \upsilon \|\mathbf{E}\hat{\boldsymbol{\Sigma}}_y (\hat{\mathbf{E}}^{\tau-1})^T - \mathbf{I}||_F^2 + \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x (\hat{\mathbf{D}}^{\tau-1})^T - \mathbf{I}||_F^2\end{aligned} \tag{3.6}$$

which will enable the derivation of closed-form and simple to implement iterates for the entries of $\mathbf{D}$ and $\mathbf{E}$ as will be detailed early on. To facilitate applicability of coordinate descent iterations, the cost in (3.6) can be rewritten w.r.t. $\mathbf{D}$ (while keeping $\mathbf{E}$ fixed)

$$\begin{aligned}\hat{\mathbf{D}}^{\tau} = \text{arg min}_{\mathbf{D}} N^{-1} \|\hat{\mathbf{E}}^{\tau-1}\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 \\ + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho} ||\mathbf{D}_{\rho:}^T||_1 + \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x (\hat{\mathbf{D}}^{\tau-1})^T - \mathbf{I}||_F^2\end{aligned} \tag{3.7}$$

where $\mathbf{X} := [\mathbf{x}(0) - \hat{\mathbf{m}}_x, ..., \mathbf{x}(N-1) - \hat{\mathbf{m}}_x]$ and $\mathbf{Y} := [\mathbf{y}(0) - \hat{\mathbf{m}}_y, ..., \mathbf{y}(N-1) - \hat{\mathbf{m}}_y]$ are $pf \times N$ matrices that contain the past and present/future data vectors in (3.2) and (3), shifted to zero mean.

Coordinate descent is applied in (3.6) (or equivalently (3.7) since $\mathbf{E}$ is fixed) to split the task of minimizing the cost in (3.7) into qpf scalar minimization subproblems, corresponding to each of the entries of the matrix $\mathbf{D}$. Specifically, the problem in (3.7) is minimized w.r.t. one entry of $\mathbf{D}$, say $\mathbf{D}(\alpha, \beta)$, while fixing matrix $\mathbf{E}$ as well as the $qpf - 1$ remaining entries of matrix $\mathbf{D}$ to their most recent updates. Then, the scalar update $\hat{\mathbf{D}}^{\tau}(\alpha, \beta)$ can be obtained as the following minimization problem

$$\begin{aligned}\hat{\mathbf{D}}^{\tau}(\alpha,\beta) &= \text{arg min}_{d}\|\boldsymbol{\psi}^{\tau}_{\alpha,\beta} - d\mathbf{h}^{\tau}_{\alpha,\beta}\|_2^2 + \lambda_{D,\alpha}|d| \tag{3.8}\\ &+ \|\breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} - d\breve{\mathbf{h}}^{\tau}_{\alpha,\beta}\|_2^2, \text{ for } \alpha = 1,...,q,\ \beta = 1,...,pf \tag{3.9}\end{aligned}$$

in which,

$$\begin{aligned}&\boldsymbol{\psi}^{\tau}_{\alpha,\beta} := N^{-0.5}([\hat{\mathbf{E}}^{\tau-1}\mathbf{Y}]_{\alpha:} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}^{\tau}(\alpha,\ell)\mathbf{X}_{\ell:} - \textstyle\sum_{\ell=\beta+1}^{pf}\hat{\mathbf{D}}^{\tau-1}(\alpha,\ell)\mathbf{X}_{\ell:})\\ &\mathbf{h}^{\tau}_{\alpha,\beta} := N^{-0.5}(\mathbf{X}_{\beta:}) \tag{3.10}\end{aligned}$$

$$\begin{aligned}&\breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} := \varepsilon^{0.5}(\mathbf{I}_{\alpha:} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}^{\tau}(\alpha,\ell)[\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T]_{\ell:} - \textstyle\sum_{\ell=\beta+1}^{pf}\hat{\mathbf{D}}^{\tau-1}(\alpha,\ell)[\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T]_{\ell:})\\ &\breve{\mathbf{h}}^{\tau}_{\alpha,\beta} := \varepsilon^{0.5}[\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T]_{\beta:} \tag{3.11}\end{aligned}$$

where $\mathbf{M}_{\alpha:}$ (or $[\mathbf{M}]_{\alpha:}$) and $\mathbf{M}_{:\ell}$ correspond to the αth row and ℓth column of matrix $\mathbf{M}$, respectively. Further, the minimization problem in (3.8) can be rewritten as

$$\hat{\mathbf{D}}^{\tau}(\alpha,\beta) = \text{arg min}_{d}\|\underline{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} - d\underline{\mathbf{h}}^{\tau}_{\alpha,\beta}\|_2^2 + \lambda_{D,\alpha}|d| \tag{3.12}$$

where $\underline{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} := [\boldsymbol{\psi}^{\tau}_{\alpha,\beta}, \breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}]^T$ and $\underline{\mathbf{h}}^{\tau}_{\alpha,\beta} := [\mathbf{h}^{\tau}_{\alpha,\beta}, \breve{\mathbf{h}}^{\tau}_{\alpha,\beta}]^T$. The minimization problem (3.12) corresponds to a scalar sparse regression problem. After applying the Karush-Kuhn-Tucker (KKT) conditions (see, e.g., [4]), and using Lemma 1 in [64] it turns out that

$$\hat{\mathbf{D}}^{\tau}(\alpha,\beta) = \mathbb{F}(\underline{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}, \underline{\mathbf{h}}^{\tau}_{\alpha,\beta}, 0, 0, \lambda_{D,\alpha}) \tag{3.13}$$

$$\begin{aligned}&\text{where } \mathbb{F}(\mathbf{p}_1,\mathbf{p}_2,p_3,p_4,\lambda) := \text{sgn}(\mathbf{p}_1^T\mathbf{p}_2 + p_3) \tag{3.14}\\ &\times (\max\left(0, \left(\left|\frac{\mathbf{p}_1^T\mathbf{p}_2 + p_3}{\|\mathbf{p}_2\|_2^2 + p_4}\right| - \left(\frac{\lambda}{2(\|\mathbf{p}_2\|_2^2 + p_4)}\right)\right)\right)\end{aligned}$$

Similarly, we can obtain the update $\hat{\mathbf{E}}^{\tau}(\alpha,\beta)$ after fixing the remaining entries of matrix $\mathbf{E}$ to their most recent updates, and set $\mathbf{D}$ to $\hat{\mathbf{D}}^{\tau-1}$ in (3.5). Then, the update $\hat{\mathbf{E}}^{\tau}(\alpha,\beta)$ is

$$\hat{\mathbf{E}}^{\tau}(\alpha,\beta) = \mathbb{F}(\bar{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}, \bar{\mathbf{h}}^{\tau}_{\alpha,\beta}, 0, 0, \lambda_{E,\alpha}) \tag{3.15}$$

where $\bar{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}$ and $\bar{\mathbf{h}}^{\tau}_{\alpha,\beta}$ are similar to $\underline{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}$ and $\underline{\mathbf{h}}^{\tau}_{\alpha,\beta}$, respectively, after doing the following substitutions: $\mathbf{Y} \rightarrow \mathbf{X}$, $\mathbf{X} \rightarrow \mathbf{Y}$, $\hat{\mathbf{E}}^{\tau-1} \rightarrow \hat{\mathbf{D}}^{\tau-1}$, $\hat{\mathbf{D}}^{\tau} \rightarrow \hat{\mathbf{E}}^{\tau}$, $\hat{\mathbf{D}}^{\tau-1} \rightarrow \hat{\mathbf{E}}^{\tau-1}$, $\hat{\boldsymbol{\Sigma}}_x \rightarrow \hat{\boldsymbol{\Sigma}}_y$, and $\varepsilon \rightarrow \upsilon$. The CS-CCA algorithm steps are:

Step 1) Initialize $\check{\mathbf{D}}^{(0)}$ and $\check{\mathbf{E}}^{(0)}$ randomly.

Step 2) For the τth coordinate descent, update $\hat{\mathbf{D}}^{\tau}(\alpha, \beta)$ and $\hat{\mathbf{E}}^{\tau}(\alpha, \beta)$ via (3.13) and (3.15) for $\alpha = 1, ..., q$ and $\beta = 1, \ldots, f$.

Step 3) If the S-CCA cost reduction in the current descent is larger than a pre-specified threshold go back to Step 2), otherwise exit and return $\hat{\mathbf{D}} = \hat{\mathbf{D}}^{\tau}$ and $\hat{\mathbf{E}} = \hat{\mathbf{E}}^{\tau}$.

Proposition 2: *Let* $\mathbf{D}^*$ *and* $\mathbf{E}^*$ *indicate a stationary point of* (3.5)*. As the coordinate cycles* τ *go to infinity, the updates* $\hat{\mathbf{D}}^{\tau}$ *and* $\hat{\mathbf{E}}^{\tau}$ *obtained from* (3.13) *and* (3.15) *in S-CCA will satisfy* $\|\hat{\mathbf{D}}^{\tau} - \mathbf{D}^*\|_F \leq \delta(\varepsilon)$ *and* $\|\hat{\mathbf{E}}^{\tau} - \mathbf{E}^*\|_F \leq \delta(\varepsilon)$*, where* $lim_{\varepsilon \rightarrow 0}\delta(\varepsilon) = 0$, and ε is the parameter in (3.5).

The result of Proposition 2 (proved in Appendix A) indicates that the iterates $\hat{\mathbf{D}}^{\tau}(\alpha, \beta)$ and $\hat{\mathbf{E}}^{\tau}(\alpha, \beta)$ can be brought arbitrarily close to a stationary point in (3.5) by selecting a sufficiently small ε in (3.5), since as ε goes to zero, distance $\delta(\varepsilon)$ will go to zero. Interestingly, although the original CCA cost in (6) is approximated by (7) to simplify the algorithmic implementation and complexity, the algorithmic iterates $\hat{\mathbf{D}}^{\tau}(\alpha, \beta)$ and $\hat{\mathbf{E}}^{\tau}(\alpha, \beta)$ are capable to approach a stationary point of the CCA cost in (6) arbitrarily close.

3.2.2 Distributed S-CCA (DS-CCA)

The centralized S-CCA scheme in Sec. 3.2.1 was developed under the assumption that the sequences $\mathbf{x}(t)$, $\mathbf{y}(t)$ in (3.2) and (3) are available at a central fusion center, which forms the updates $\hat{\mathbf{D}}^{\tau}$ and $\hat{\mathbf{E}}^{\tau}$. Here no central fusion center exists while sensors collect information in a distributed way and they are able to communicate only with their single-hop neighbors. A distributed algoithm is proposed, in which sensor j will update the submatrices $\mathbf{D}_j \in \mathbb{R}^{q\times f}$ and $\mathbf{E}_j \in \mathbb{R}^{q\times f}$ that contain the columns of $\mathbf{D}$ and $\mathbf{E}$ respectively with indices $j, p+j, 2p+j, \ldots, (f-1)p+j$, i.e.,

$$\begin{aligned}\mathbf{D}_j &:= [\mathbf{D}_{:j}, \mathbf{D}_{:p+j}, \mathbf{D}_{:2p+j}, .., \mathbf{D}_{:(f-1)p+j}]\\ \mathbf{E}_j &:= [\mathbf{E}_{:j}, \mathbf{E}_{:p+j}, \mathbf{E}_{:2p+j}, ..., \mathbf{E}_{:(f-1)p+j}].\end{aligned}$$

Further, let $\mathbf{x}(t,j)$ and $\mathbf{y}(t,j)$ correspond to the $f\times 1$ subvectors of $\mathbf{x}(t)-\hat{\mathbf{m}}_x$ and $\mathbf{y}(t)-\hat{\mathbf{m}}_y$ which are obtained after keeping their entries with indices $j, p+j, 2p+j, \ldots, (f-1)p+j$. After noticing that $\mathbf{D}\mathbf{x}(t) = \sum_{i=1}^{p} \mathbf{D}_i\mathbf{x}(t,i)$ and $\mathbf{E}\mathbf{y}(t) = \sum_{i=1}^{p} \mathbf{E}_i\mathbf{y}(t,i)$, then (3.5) can be reformulated as

$$\begin{aligned}&\arg\min_{\mathbf{D},\mathbf{E}} N^{-1}\textstyle\sum_{t=0}^{N-1} \|\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(t,i) - \sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(t,i)\|_2^2\\ &+\textstyle\sum_{i=1,\rho=1}^{p,q}\lambda_{D,\rho}\|\mathbf{D}_{i,\rho:}\|_1 + \sum_{i=1,\rho=1}^{p,q}\lambda_{E,\rho}\|\mathbf{E}_{i,\rho:}\|_1\\ &+ \upsilon\|N^{-1}\textstyle\sum_{t=0}^{N-1}[(\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(t,i))(\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(t,i))^T] - \mathbf{I}\|_F^2\\ &+ \varepsilon\|N^{-1}\textstyle\sum_{t=0}^{N-1}[(\sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(t,i))(\sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(t,i))^T] - \mathbf{I}\|_F^2\end{aligned} \tag{3.16}$$

in which $\mathbf{D}_{i,\rho:}$ and $\mathbf{E}_{i,\rho:}$ represent the ρth row of $\mathbf{D}_i$ and $\mathbf{E}_i$.

The distributed S-CCA will be derived by combining block coordinate descent (BCD) techniques along with the alternating direction method of multipliers (ADMM) [5, 61]. Specifically, BCD is used to split the minimization in (3.16) into p minimization subtasks where the cost in (3.16) is minimized w.r.t. to the block $\mathbf{D}_j$ (or $\mathbf{E}_j$) at sensor j for $j = 1, \ldots, p$. Meanwhile, ADMM will be employed to allow sensors estimate in a distributed fashion the global quantities $\hat{\mathbf{D}}^\tau \cdot (\mathbf{x}(t) - \hat{\mathbf{m}}_x)$ and $\hat{\mathbf{E}}^\tau \cdot (\mathbf{y}(t) - \hat{\mathbf{m}}_y)$ which will be necessary when minimizing the cost in (3.16) w.r.t. $\mathbf{D}_j$ and $\mathbf{E}_j$ at sensor j. As in Sec. 3.2.1, to avoid generating fourth-order terms in the last two summands in the cost in (3.16) we substitute one of the $\mathbf{D}$ and $\mathbf{E}$ with their latest update during iteration τ, i.e., the last term in (3.16) is replaced with

$$\varepsilon\|N^{-1}\sum_{t=0}^{N-1}[(\sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(t,i))(\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(t,i))^T] - \mathbf{I}\|_F^2, \tag{3.17}$$

Similarly, we can substitute the second last term in (3.16) with

$$v\|N^{-1}\sum_{t=0}^{N-1}[(\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(t,i))(\sum_{i=1}^{p}\hat{\mathbf{E}}_i^{\tau-1}\mathbf{y}(t,i))^T]-\mathbf{I}\|_F^2, \tag{3.18}$$

The first step is to minimize (3.16) w.r.t. $\mathbf{D}_j$ at sensor j. Toward this end, we fix all submatrices $\{\mathbf{D}_i\}_{i\neq j}$ and $\{\mathbf{E}_i\}_{i=1}^p$ to their latest updates. Then, the resulting cost can be written as a function of $\mathbf{D}_j$ as

$$\begin{aligned}&\|\hat{\mathbf{E}}^{\tau-1}\mathbf{Y}-\hat{\mathbf{D}}^{\tau-1}\mathbf{X}+\hat{\mathbf{D}}_j^{\tau-1}\mathbf{X}_j-\mathbf{D}_j\mathbf{X}_j\|_F^2\\&+\varepsilon\|N^{-1}(\hat{\mathbf{D}}^{\tau-1}\mathbf{X}-\hat{\mathbf{D}}_j^{\tau-1}\mathbf{X}_j+\mathbf{D}_j\mathbf{X}_j)(\hat{\mathbf{D}}^{\tau-1}\mathbf{X})^T-\mathbf{I}\|_F^2\\&+\textstyle\sum_{\rho=1}^{q}\lambda_{D,\rho}\|\mathbf{D}_{j,\rho:}\|_1,\ \text{where,}\end{aligned} \tag{3.19}$$

$$\begin{aligned}&\mathbf{X}:=[\mathbf{x}(0)-\hat{\mathbf{m}}_x,\ \mathbf{x}(1)-\hat{\mathbf{m}}_x,\ldots,\mathbf{x}(N-1)-\hat{\mathbf{m}}_x]\in\mathbb{R}^{pf\times N},\\&\mathbf{Y}:=[\mathbf{y}(0)-\hat{\mathbf{m}}_y,\ \mathbf{y}(1)-\hat{\mathbf{m}}_y,\ldots,\mathbf{y}(N-1)-\hat{\mathbf{m}}_y]\in\mathbb{R}^{pf\times N}\\&\mathbf{X}_j:=[\mathbf{x}(0,j),\ \mathbf{x}(1,j)\ldots\mathbf{x}(N-1,j)]\in\mathbb{R}^{f\times N},\ \text{and}\\&\hat{\mathbf{D}}^{\tau-1}\mathbf{X}-\hat{\mathbf{D}}_j^{\tau-1}\mathbf{X}_j=\textstyle\sum_{i=1,i\neq j}^{p}\hat{\mathbf{D}}_i^{\tau-1}\mathbf{X}_i.\end{aligned} \tag{3.20}$$

From (3.19) it follows that the 'global' terms $\hat{\mathbf{D}}^{\tau-1}[\mathbf{x}(t)-\hat{\mathbf{m}}_x]$ and $\hat{\mathbf{E}}^{\tau-1}[\mathbf{y}(t)-\hat{\mathbf{m}}_y]$ have to be available at every sensor j in order to update each entry of $\mathbf{D}_j$ via coordinate descent. However, this is not the case since these global quantities contain information from all sensors, and they are not physically available. To this end, ADMM will be utilized to express $\hat{\mathbf{D}}^{\tau-1}[\mathbf{x}(t)-\hat{\mathbf{m}}_x]$ or $\hat{\mathbf{E}}^{\tau-1}[\mathbf{y}(t)-\hat{\mathbf{m}}_y]$ at the solution of a separable convex minimization problem that can be solved in a distributed fashion and allow each sensor j to estimate these global quantities. Then, these estimates will be used to replace the corresponding quantities in (3.19) which will be further minimized w.r.t. one entry of $\mathbf{D}_j$ while fixing the rest.

Estimation of global quantities via ADMM:

To this end, note that $\hat{\mathbf{D}}^{\tau-1}\mathbf{X}=[\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(0)-\hat{\mathbf{m}}_x),\ldots,\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(N-1)-\hat{\mathbf{m}}_x)]$, and

$\hat{\mathbf{E}}^{\tau-1}\mathbf{Y} = [\hat{\mathbf{E}}^{\tau-1}(\mathbf{y}(0) - \hat{\mathbf{m}}_y), \ldots, \hat{\mathbf{E}}^{\tau-1}(\mathbf{y}(N-1) - \hat{\mathbf{m}}_y)]$. Sensor j can obtain estimates for the vectors $\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x)$ $t = 0, 1, ..., N-1$, by solving via ADMM the separable constrained minimization problem:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{i,t} &= \min_{\boldsymbol{\mu}_{i,t}} \textstyle\sum_{i=1}^{p} \|\boldsymbol{\mu}_{i,t} - p\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(t,i)\|_2^2 \\ &\text{s. to } \boldsymbol{\mu}_{i,t} = \boldsymbol{\mu}_{i',t}, i' \in \mathcal{N}_i, \text{ for } t = 0, \ldots, N-1, \end{aligned} \tag{3.21}$$

where $\boldsymbol{\mu}_{i,t} \in \mathbb{R}^{q\times 1}$ represents a local state vector at sensor i for estimating $\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x) = \sum_{i=1}^{p} \hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(t,i)$ which is the minimizer $\hat{\boldsymbol{\mu}}_{i,t}$ in (3.21). The equality constraints $\boldsymbol{\mu}_{i,t} = \boldsymbol{\mu}_{i',t}$ guarantee that all local estimates $\boldsymbol{\mu}_{i,t}$ will be equal across sensors. By employing the ADMM, see details in e.g., [5, 61], the subproblems (3.21) for $t = 0, \ldots, N-1$ will be tackled through updating the sensor j's local estimate, $\boldsymbol{\mu}_{j,t}$, along with the Lagrange multipliers $\{\mathbf{v}_{j,t}^{j'}\}_{j' \in \mathcal{N}_j}$ that correspond to the constraints $\boldsymbol{\mu}_{j,t} = \boldsymbol{\mu}_{j',t}$. Sensor j is responsible for carrying out the updating recursions (see details in Appendix B)

$$\mathbf{v}_{j,t}^{j'}(k) = \mathbf{v}_{j,t}^{j'}(k-1) + 0.5c[\boldsymbol{\mu}_{j,t}(k) - \boldsymbol{\mu}_{j',t}(k)] \tag{3.22}$$

$$\begin{aligned} \boldsymbol{\mu}_{j,t}(k+1) = [(2 + 2c|\mathcal{N}_j|)\mathbf{I}]^{-1} \times \Big[2p\hat{\mathbf{D}}_j^{\tau-1}\mathbf{x}(t,j) \\ - \textstyle\sum_{j' \in \mathcal{N}_j} ((\mathbf{v}_{j,t}^{j'}(k) - \mathbf{v}_{j',t}^{j}(k)) + c(\boldsymbol{\mu}_{j,t}(k) + \boldsymbol{\mu}_{j',t}(k)))\Big] \end{aligned} \tag{3.23}$$

where k corresponds to the ADMM iteration index, while c is a positive step-size. Using the convergence results in [61], as k goes to infinity, $\boldsymbol{\mu}_{j,t}(k)$ will converge to $\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x)$, no matter how the local estimates $\boldsymbol{\mu}_{j,t}(0)$ are initialized (here they are initialized at zero). Per coordinate cycle $\tau-1$, a finite number of K ADMM iterations are performed to estimate $\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x)$. A similar procedure is followed for estimating $\hat{\mathbf{E}}^{\tau-1}(\mathbf{y}(t) - \hat{\mathbf{m}}_y)$ across sensors. The corresponding local estimate for $\hat{\mathbf{E}}^{\tau-1}(\mathbf{y}(t) - \hat{\mathbf{m}}_y)$ at sensor j is denoted by $\boldsymbol{\eta}_{j,t}(k+1)$. A similar set of local iterations as the ones in (3.22) are employed at sensor j to update $\boldsymbol{\eta}_{j,t}(k+1)$. Further, let $\hat{\boldsymbol{\mu}}_{j,t}^{\tau}$ and $\hat{\boldsymbol{\eta}}_{j,t}^{\tau}$ to be the local estimates of $\hat{\mathbf{D}}^{\tau-1}(\mathbf{x}(t) - \hat{\mathbf{m}}_x)$ and $\hat{\mathbf{E}}^{\tau-1}(\mathbf{y}(t) - \hat{\mathbf{m}}_y)$, respectively, after running K iterations in the τth coordinate cycle, i.e., $\hat{\boldsymbol{\mu}}_{j,t}^{\tau} := \boldsymbol{\mu}_{j,t}(K)$ and $\hat{\boldsymbol{\eta}}_{j,t}^{\tau} := \boldsymbol{\eta}_{j,t}(K)$. Hence, after coordinate cycle τ the global quantities $\hat{\mathbf{D}}^{\tau-1}\mathbf{X}$ and $\hat{\mathbf{E}}^{\tau-1}\mathbf{Y}$ in (3.19) are replaced with sensor j's local estimates $\hat{\boldsymbol{\mu}}_j^{\tau} := [\hat{\boldsymbol{\mu}}_{j,0}^{\tau}, \ldots, \hat{\boldsymbol{\mu}}_{j,N-1}^{\tau}]$ and $\hat{\boldsymbol{\eta}}_j^{\tau} := [\hat{\boldsymbol{\eta}}_{j,0}^{\tau}, \ldots, \hat{\boldsymbol{\eta}}_{j,N-1}^{\tau}]$, respectively. Using the notation

$$\begin{aligned}
&\mathbf{M}_j^{\tau-1} := \hat{\boldsymbol{\eta}}_j^{\tau} - \hat{\boldsymbol{\mu}}_j^{\tau} + \hat{\mathbf{D}}_j^{\tau-1}\mathbf{X}_j, \\
&\mathbf{P}_j^{\tau-1} := \varepsilon^{0.5}N^{-1}(\hat{\boldsymbol{\mu}}_j^{\tau} - \hat{\mathbf{D}}_j^{\tau-1}\mathbf{X}_j)(\hat{\boldsymbol{\mu}}_j^{\tau})^T - \varepsilon^{0.5}\mathbf{I}, \\
&\mathbf{Q}_j^{\tau-1} := -\varepsilon^{0.5}N^{-1}\mathbf{X}_j(\hat{\boldsymbol{\mu}}_j^{\tau})^T \qquad (3.24)
\end{aligned}$$

the cost in (3.19) can be readily rewritten as

$$N^{-1}\|\mathbf{M}_j^{\tau-1} - \mathbf{D}_j\mathbf{X}_j\|_F^2 + \|\mathbf{P}_j^{\tau-1} - \mathbf{D}_j\mathbf{Q}_j^{\tau-1}\|_F^2 + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho}\|\mathbf{D}_{j,\rho:}\|_1 \qquad (3.25)$$

which will be tackled at sensor j to update the entries of sub matrix $\mathbf{D}_j$.

The cost (3.25) is minimized locally at sensor j w.r.t. one entry of $\mathbf{D}_j$, for instance, $\mathbf{D}_j(\alpha,\beta)$, while keeping the rest entries of $\mathbf{D}_j$ fixed. During τth cycle, the variable $\mathbf{D}_j(\alpha,\beta)$ in (3.25) can be obtained by minimizing

$$\hat{\mathbf{D}}_j^{\tau}(\alpha,\beta) = \text{arg min}_d \|\boldsymbol{\psi}_{j,\alpha,\beta}^{\tau} - d\mathbf{h}_{j,\alpha,\beta}^{\tau}\|_2^2 + \lambda_{D,\alpha}\|d\|_1 \qquad (3.26)$$

where $\alpha = 1,..,q$, $\beta = 1,..,f$, while

$$\mathbf{h}_{j,\alpha,\beta}^{\tau} := [N^{-0.5}\mathbf{X}_{j,\beta:}, \mathbf{Q}_{j,\beta:}^{\tau-1}]^T, \boldsymbol{\psi}_{j,\alpha,\beta}^{\tau} := [\boldsymbol{\psi}_{j,\alpha,\beta,1}^{\tau}, \boldsymbol{\psi}_{j,\alpha,\beta,2}^{\tau}]^T \qquad (3.27)$$

$$\begin{aligned}
&\boldsymbol{\psi}_{j,\alpha,\beta,1}^{\tau} := N^{-1/2}[\mathbf{M}_{j,\alpha:}^{\tau-1} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}_j^{\tau}(\alpha,\ell)\mathbf{X}_{j,\ell:} - \sum_{\ell=\beta+1}^{f}\hat{\mathbf{D}}_j^{\tau-1}(\alpha,\ell)\mathbf{X}_{j,\ell:}] \text{ and} \\
&\boldsymbol{\psi}_{j,\alpha,\beta,2}^{\tau} := \mathbf{P}_{j,\alpha:}^{\tau-1} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}_j^{\tau}(\alpha,\ell)\mathbf{Q}_{j,\ell:}^{\tau-1} - \sum_{\ell=\beta+1}^{f}\hat{\mathbf{D}}_j^{\tau-1}(\alpha,\ell)\mathbf{Q}_{j,\ell:}^{\tau-1}. \qquad (3.28)
\end{aligned}$$

Using the result in (3.13) for (3.12) it follows that the solution for $\hat{\mathbf{D}}_j^{\tau}(\alpha,\beta)$ is given as

$$\hat{\mathbf{D}}_j^{\tau}(\alpha,\beta) = \mathbb{F}(\boldsymbol{\psi}_{j,\alpha,\beta}^{\tau}, \mathbf{h}_{j,\alpha,\beta}^{\tau}, 0, 0, \lambda_{D,\alpha}), \qquad (3.29)$$

where $\mathbb{F}(\cdot)$ is defined in (4.22). Applying a process similar to the one for deriving (3.29), the update $\hat{\mathbf{E}}^{\tau}_{j}(\alpha,\beta)$ can be formed as

$$\hat{\mathbf{E}}^{\tau}_{j}(\alpha,\beta) = \mathbb{F}(\boldsymbol{\phi}^{\tau}_{j,\alpha,\beta}, \mathbf{g}^{\tau}_{j,\alpha,\beta}, 0, 0, \lambda_{E,\alpha}) \tag{3.30}$$

where $\boldsymbol{\phi}^{\tau}_{j,\alpha,\beta}$ and $\mathbf{g}^{\tau}_{j,\alpha,\beta}$ are similar to $\boldsymbol{\psi}^{\tau}_{j,\alpha,\beta}$ and $\mathbf{h}^{\tau}_{j,\alpha,\beta}$, respectively, after doing the following substitutions: $\hat{\boldsymbol{\mu}}^{\tau}_{j} \rightarrow \hat{\boldsymbol{\eta}}^{\tau}_{j}$, $\hat{\boldsymbol{\eta}}^{\tau}_{j} \rightarrow \hat{\boldsymbol{\mu}}^{\tau}_{j}$, $\hat{\mathbf{D}}^{\tau-1}_{j} \rightarrow \hat{\mathbf{E}}^{\tau-1}_{j}$, $\hat{\mathbf{D}}^{\tau}_{j} \rightarrow \hat{\mathbf{E}}^{\tau}_{j}$, $\mathbf{X}_{j} \rightarrow \mathbf{Y}_{j}$ and $\varepsilon \rightarrow v$.

In the beginning of τth coordinate cycle, the most-up-to-date $\hat{\mathbf{D}}^{\tau-1}_{j}$ and $\hat{\mathbf{E}}^{\tau-1}_{j}$ are available at sensor j. Then, K ADMM iterations will be run, nested in cycle τ, to estimate the global values, $\hat{\mathbf{D}}^{\tau-1}\mathbf{X}$ and $\hat{\mathbf{E}}^{\tau-1}\mathbf{Y}$, via the local estimates $\hat{\boldsymbol{\mu}}^{\tau}_{j}$ and $\hat{\boldsymbol{\eta}}^{\tau}_{j}$, respectively. During the ADMM iterations, sensor j has to communicate with its $|\mathcal{N}_j|$ neighboring sensors, which includes receiving vectors $\{\mathbf{v}^{j}_{j',t}(k), \boldsymbol{\mu}_{j',t}(k)\}$ and transmitting vectors, $\{\mathbf{v}^{j}_{j,t}(k)\}_{j'\in\mathcal{N}_j}, \boldsymbol{\mu}_{j,t}(k)$ from/to its neighboring sensors in set $\mathcal{N}_j$. In detail, sensor j receives $2NKq|\mathcal{N}_j|$ scalars in cycle τ, that correspond to the entries of the $q \times 1$ vectors $\mathbf{v}^{j}_{j',t}(k)$ and $\boldsymbol{\mu}_{j',t}(k)$, for $t = 0, ..., N-1$ and $j' \in \mathcal{N}_j$, needed to carry out the updates in (3.22) and (24). Meanwhile, sensor j will transmit the ADMM multipliers, $\{\mathbf{v}^{j'}_{j,t}(k)\}_{j'\in\mathcal{N}_j}$ and estimates $\boldsymbol{\mu}_{j,t}(k)$ to its single-hop neighbors, which accounts for $(|\mathcal{N}_j|+1)qN$ scalars per ADMM iteration and $(|\mathcal{N}_j|+1)qNK$ scalars in total. In summary, the total number of testing data N, the cardinality of neighborhood $|\mathcal{N}_j|$, the size of q, which depends on the number of sources and their dimensionality, and ADMM iterations K together decide the communication cost. In practice $q \ll p$, since only a few sources are sensed by many sensors. The DS-CCA scheme is tabulated as Alg. 1.

As $K \rightarrow \infty$, from the convergence claims in [61] it follows that $\hat{\boldsymbol{\mu}}^{\tau}_{j} \rightarrow \hat{\mathbf{D}}^{\tau-1}\mathbf{X}$ and $\hat{\boldsymbol{\eta}}^{\tau}_{j} \rightarrow \hat{\mathbf{E}}^{\tau-1}\mathbf{Y}$. Further, as the number of coordinate cycles $\tau \rightarrow \infty$, and nested coordinate iterations $K' \rightarrow \infty$ for $\hat{\mathbf{D}}^{\tau}_{j}(\alpha,\beta)$ and $\hat{\mathbf{E}}^{\tau}_{j}(\alpha,\beta)$, then the updates $\hat{\mathbf{D}}^{\tau}$ and $\hat{\mathbf{E}}^{\tau}$ as $\tau \rightarrow \infty$ approach $\delta(\varepsilon)$-close to a stationary point of the cost in (3.16) where $\lim_{\varepsilon\rightarrow 0}\delta(\varepsilon) =$

Algorithm 1 : DS-CCA

```
Initialize $\check{\mathbf{D}}_j^{(0)}$, $\check{\mathbf{E}}_j^{(0)}$ with the outcome of DS-CCA applied for $\{\lambda_{E,\rho} = \lambda_{D,\rho} = 0\}_{\rho=1}^{q}$ and initialized randomly.
for $\tau = 1, 2...$ do
Sensor $j$ forms estimates $\{\hat{\boldsymbol{\mu}}_{j,t}^{\tau}\}_{t=0}^{N-1}$ (and $\{\hat{\boldsymbol{\eta}}_{j,t}^{\tau}\}_{t=0}^{N-1}$) via K ADMM updating recursions
in (3.22)-(24) for $j = 1, \ldots, p$ nested in cycle $\tau$.
    for $j = 1, ..., p$ do
        Update $\hat{\mathbf{D}}_j^{\tau}(\alpha, \beta)$ via (3.29).
        Update $\hat{\mathbf{E}}_j^{\tau}(\alpha, \beta)$ via (3.30), for $\alpha = 1, ..., q$ and $\beta = 1, \ldots, f$.
        Repeat the updates for $K' \geq 1$ cycles.
    end for
end for
```

0 [similar arguments as in the proof of Proposition 2 can be used here]. As a termination criterion, the 'updating' error $\|\hat{\mathbf{D}}_j^{\tau} - \hat{\mathbf{D}}_j^{\tau-1}\|_F + \|\hat{\mathbf{E}}_j^{\tau} - \hat{\mathbf{E}}_j^{\tau-1}\|_F$ is checked until it drops below a desired tolerance.

3.3 Selection of λ

Proper selection of the sparsity-controlling coefficients in both CS-CCA and DS-CCA is critical to ensure that the zero and nonzero entries are placed in the right positions of the estimated $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$.

Next, a simple and sensible method is put forth to select the λ's. To simplify things, we set $\lambda_{D,\rho} = \lambda_{E,\rho} = \lambda_\rho$ for $\rho = 1, ..., q$. This selection is reasonable given that the support of the same rows in $\mathbf{D}$ and $\mathbf{E}$, ideally, should coincide as explained in Sec. 3.1. Let $\{\lambda_\rho^{\max}\}_{\rho=1}^{q}$ denote the smallest values of the sparsity controlling coefficients that result the ρth row of $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ obtained from CS-CCA (or DS-CCA) to be equal to zero.

Specifically, the proposed method here addresses two challenges: i) find a relatively large value to initialize λ_ρ which returns an all-zeros solution for the ρth row of $\mathbf{D}$ (and $\mathbf{E}_{\rho:}$), i.e., $\hat{\mathbf{D}}_{\rho:} = \mathbf{0}$; and ii) gradually decrease λ_ρ starting from the relatively large value set at i) and determine when to stop. The motivation is to start from an all-zeros solution and gradually let more and more nonzero values show up in $\mathbf{D}$ (and $\mathbf{E}$) by decreasing λ_ρ and reapplying S-CCA. The approach for selecting the initial value for λ_ρ and determining when to stop decreasing λ_ρ is explained next.

The first step (Step 1 in Alg. 2) is to estimate $\lambda_\rho^{\text{max}}$ (not available in closed form) via estimates $\hat{\lambda}_\rho^m$ for $\rho = 1, \ldots, q$. After randomly initializing $\hat{\lambda}_\rho^m$ and applying CS-CCA (or DS-CCA) the support sets of the estimates $\hat{\mathbf{D}}_{\rho:}$ and $\hat{\mathbf{E}}_{\rho:}$ are checked. If the support sets are nonempty (nonzero entries exist) then $\hat{\lambda}_\rho^m$ is increased by a factor of $\omega_2 > 1$. The estimates $\hat{\lambda}_\rho^m$ will keep increasing until the CS-CCA (or DS-CCA) gives an empty support for $\hat{\mathbf{D}}_{\rho:}$ and $\hat{\mathbf{E}}_{\rho:}$ in which case it is certain that $\lambda_\rho^{\text{max}}$ has been reached or exceeded.

If the support sets $\hat{\mathbf{D}}_{\rho:}$ and/or $\hat{\mathbf{E}}_{\rho:}$ are empty then $\hat{\lambda}_\rho^m$ has exceeded $\lambda\rho^{\text{max}}$ in which case Alg. 2 starts decreasing $\hat{\lambda}_\rho^m$ by a factor of $\omega_1 \in [1-\epsilon, 1)$ (close to one). The estimates $\hat{\lambda}_\rho^m$ will be decreased until when CS-CCA (or DS-CCA) gives a nonempty support for $\hat{\mathbf{D}}_{\rho:}$ and $\hat{\mathbf{E}}_{\rho:}$ in which case Step 1 is concluded. Note that the closer ω_1 is to one, the more accurate Step 1 will be in estimating $\lambda_\rho^{\text{max}}$.

Given the estimate $\hat{\lambda}_\rho^m$ from Step 1, Step 2 is focusing on recovering the indices of columns in $\mathbf{D}$ and $\mathbf{E}$ that are zero, denoted here as $\mathcal{C}$. Note that the index of a zero column indicates a sensor measurement acquiring only sensing noise. The estimate $\hat{\lambda}_\rho^m$ is scaled with factors $\omega_3 < 1$ and $\omega_4 < 1$, where $\omega_4 << \omega_3$. Two different column zero-entry sets, namely $\mathcal{C}_1$ (using ω_3) and $\mathcal{C}_2$ (using ω_4), are obtained after applying CS-CCA (or DS-CCA). Since $\omega_4 << \omega_3$ it is expected that $\mathcal{C}_1 \supseteq \mathcal{C}_2$. The reason for getting two different sets $\mathcal{C}_1$ and $\mathcal{C}_2$ is to identify which columns (noisy sensors) in $\mathbf{D}$ and $\mathbf{E}$ will be zero for both different scalings of $\hat{\lambda}_\rho^m$ using ω_4 and ω_3. This way the columns of $\mathbf{E}, \mathbf{D}$ that match with entries in $\mathbf{x}(t), \mathbf{y}(t)$ that contain information about a source (nonzero columns) can be distinguished from the columns that correspond to entries in $\mathbf{x}(t), \mathbf{y}(t)$ with just sensing noise (zero columns). Note that the proposed method is not optimizing a cost, however it will exhibit good performance as demonstrated in Sec. 3.6 and it is not computationally intensive.

The last (third) step is to select λ's that result estimates for $\mathbf{D}$ and $\mathbf{E}$ whose zero column index set coincides with $\mathcal{C}$ from Step 2. To this end, starting from $\hat{\lambda}_\rho^m$ obtained in Step 1 we gradually decrease their value by a factor $\omega_5 \in [1-\epsilon, 1)$ until the zero column index set of the $\mathbf{D}, \mathbf{E}$ estimates in CS-CCA (or DS-CCA) coincides with $\mathcal{C}$. In the numerical tests later on we set $\omega_1 = 0.75, \omega_2 = 1.5, \omega_3 = 0.1, \omega_4 = 0.01, \omega_5 = 0.95$. These parameter values exhibit acceptable behavior irrespective of the data processed, and there is no need to reselect them every time a new data set is processed.

3.4 Online Implementation

The CS-CCA and DS-CCA schemes derived in Sec. 3.2 are batch algorithms in the sense that first acquire data and then perform the processing. Such batch schemes are pertinent for settings where sensors acquire data for some limited time and then stop. However, in settings where sensors are constantly sensing new data a batch algorithm will eventually drain all storing and computational capabilities across sensors. To this end, online implementations for the S-CCA framework are derived here to allow real-time processing of the acquired sensor data and reduce computational complexity.

3.4.1 Online Centralized S-CCA (OCS-CCA)

To this end, starting from (3.5) we consider the following time-varying cost that accounts for a constant stream of sensor data

Algorithm 2 Selection of the λ's

```
-Step 1: Estimate $\{\lambda_\rho^{\max}\}_{\rho=1}^q$.
  Initialize $\{\hat{\lambda}_\rho^m > 0\}_{\rho=1}^q$ randomly.
  while(1)
  Find $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$ via CS-CCA (or DS-CCA) using $\hat{\lambda}_\rho^m$.
  If $\hat{\mathbf{D}}_{\rho:} \neq \mathbf{0}$ (or $\hat{\mathbf{E}}_{\rho:} \neq \mathbf{0}$)
   Update $\hat{\lambda}_\rho^m = \omega_2 \hat{\lambda}_\rho^m$ where $\omega_2 > 1$
  else if $\hat{\mathbf{D}}_{\rho:} = \mathbf{0}$ (or $\hat{\mathbf{E}}_{\rho:} = \mathbf{0}$)
   Update $\hat{\lambda}_\rho^m = \omega_1 \hat{\lambda}_\rho^m$, where $\omega_1 < 1$. Find $\check{\mathbf{D}}$ and $\check{\mathbf{E}}$ via
   CS-CCA (or DS-CCA) with updated $\hat{\lambda}_\rho^m$.
    If $\check{\mathbf{D}}_{\rho:} = \mathbf{0}$ (or $\check{\mathbf{E}}_{\rho:} = \mathbf{0}$)
         Update $\hat{\lambda}_\rho^m = \omega_1 \hat{\lambda}_\rho^m$.
    else if $\check{\mathbf{D}}_{\rho:} \neq \mathbf{0}$ (or $\check{\mathbf{E}}_{\rho:} \neq \mathbf{0}$)
          Break while
    endIf
  end If
  end while
-Step 2: Estimate zero column index set (denoted as $\mathcal{C}$) of $\mathbf{D}$ (or $\mathbf{E}$)
  Find zero column index set $\mathcal{C}_1$ of $\mathbf{D}$ and $\mathbf{E}$ estimates found via
  CS-CCA (or DS-CCA) using $\omega_3 \hat{\lambda}_\rho^m$ with $\omega_3 < 1$.
  Find zero column index set $\mathcal{C}_2$ of $\mathbf{D}$ and $\mathbf{E}$ estimates found via
  CS-CCA (or DS-CCA) using $\omega_4 \hat{\lambda}_\rho^m$ with $\omega_4 < \omega_3 < 1$.
  Evaluate $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$
-Step 3: Select $\{\hat{\lambda}_\rho\}_{\rho=1}^q$ to be used
  Starting from the earlier found $\hat{\lambda}_{\rho,0} = \hat{\lambda}_\rho^m$, iteratively decrease $\lambda_{\rho,n} = \omega_5 \lambda_{\rho,n-1}$ where $\omega_5 < 1$
  and apply CS-CCA (or DS-CCA). If resulting zero column index set for acquired $\hat{\mathbf{D}}$ (or $\hat{\mathbf{E}}$)
  matches $\mathcal{C}$ then stop.
```

$$
\begin{aligned}
&\text{arg min}_{\mathbf{D},\mathbf{E}}(t+1)^{-1}\textstyle\sum_{\tau=0}^{t}\|\mathbf{E}\mathbf{y}(\tau)-\mathbf{D}\mathbf{x}(\tau)-\hat{\mathbf{m}}_t\|_2^2\\
&+\textstyle\sum_{\rho=1}^{q}\lambda_{E,\rho}\|\mathbf{E}_{\rho:}^T\|_1+\sum_{\rho=1}^{q}\lambda_{D,\rho}\|\mathbf{D}_{\rho:}^T\|_1\\
&+\upsilon\|\mathbf{E}\hat{\boldsymbol{\Sigma}}_{y,t}(\hat{\mathbf{E}}^{t-1})^T-\mathbf{I}\|_F^2+\varepsilon\|\mathbf{D}\hat{\boldsymbol{\Sigma}}_{x,t}(\hat{\mathbf{D}}^{t-1})^T-\mathbf{I}\|_F^2,
\end{aligned}
\tag{3.31}
$$

where $\hat{\boldsymbol{\Sigma}}_{x,t}=(t+1)^{-1}\sum_{\tau=0}^{t}(\mathbf{x}(\tau)-\hat{\mathbf{m}}_{x,t})(\mathbf{x}(\tau)-\hat{\mathbf{m}}_{x,t})^T$ and $\hat{\mathbf{m}}_{x,t}=(t+1)^{-1}\sum_{\tau=0}^{t}\mathbf{x}(\tau)$ (similarly for $\hat{\boldsymbol{\Sigma}}_{y,t}$ and $\hat{\mathbf{m}}_{y,t}$) correspond to the online covariance and mean estimates. Further, $\hat{\mathbf{m}}_t:=\mathbf{E}\hat{\mathbf{m}}_{y,t}-\mathbf{D}\hat{\mathbf{m}}_{x,t}$.

As in Secs. 3.2.1 and 3.2.2, during time instant t, we split the minimization problem in (3.31) into multiple subproblems, which focus on solving one entry of $\mathbf{D}$ while keeping the rest of the entries of $\mathbf{D}$ and matrix $\mathbf{E}$ fixed. Using a similar way to derive the cost for $\mathbf{D}(\alpha,\beta)$ in (3.8)

$$
\begin{aligned}
\hat{\mathbf{D}}^t(\alpha,\beta)&=\text{arg min}_d\textstyle\sum_{\tau=0}^{t}(\psi_{\alpha,\beta,\tau}-d\cdot h_{\alpha,\beta,\tau})^2\\
&+\|\boldsymbol{\psi}'_{\alpha,\beta,t}-d\mathbf{h}'_{\alpha,\beta,t}\|_2^2+\lambda_{D,\alpha}|d|,\ \text{where}
\end{aligned}
\tag{3.32}
$$

$$
\begin{aligned}
\psi_{\alpha,\beta,\tau}&:=(t+1)^{-1/2}\{[\hat{\mathbf{E}}_{\alpha:}^{t-1}[\mathbf{y}(\tau)-\hat{\mathbf{m}}_{y,t}]-\textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}^t(\alpha,\ell)[\mathbf{x}(\tau)\\
&\quad-\hat{\mathbf{m}}_{x,t}]_\ell-\textstyle\sum_{\ell=\beta+1}^{pf}\hat{\mathbf{D}}^{t-1}(\alpha,\ell)[\mathbf{x}(\tau)-\hat{\mathbf{m}}_{x,t}]_\ell\}\\
h_{\alpha,\beta,\tau}&:=(t+1)^{-1/2}[\mathbf{x}(\tau)-\hat{\mathbf{m}}_{x,t}]_\beta\ \tau=0,\ldots,t\\
\boldsymbol{\psi}'_{\alpha,\beta,t}&:=\sqrt{\varepsilon}[\mathbf{I}_{\alpha:}-\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}^t(\alpha,\ell)\mathbf{B}_{t,\ell:}-\sum_{\ell=\beta+1}^{pf}\hat{\mathbf{D}}^{t-1}(\alpha,\ell)\mathbf{B}_{t,\ell:}]\\
\mathbf{h}'_{\alpha,\beta,t}&:=\sqrt{\varepsilon}\mathbf{B}_{t,\beta:}\ \text{where}\ \mathbf{B}_t:=\hat{\boldsymbol{\Sigma}}_{x,t}(\hat{\mathbf{D}}^{t-1})^T
\end{aligned}
\tag{3.33}
$$

Using the result in (3.13) the minimizer for (3.32) is

$$
\hat{\mathbf{D}}^t(\alpha,\beta)=\mathbb{F}(\boldsymbol{\psi}'_{\alpha,\beta,t},\mathbf{h}'_{\alpha,\beta,t},\Sigma_{\psi h,t}^{\alpha,\beta},\Sigma_{h^2,t}^{\beta},\lambda_{D,\alpha})
\tag{3.34}
$$

where $\Sigma^{\alpha,\beta}_{\psi h,t} := \sum_{\tau=0}^{t} \psi_{\alpha,\beta,\tau} \cdot h_{\alpha,\beta,\tau}$, $\Sigma^{\beta}_{h^2,t} := \sum_{\tau=0}^{t} h^2_{\alpha,\beta,\tau}$, for $\alpha = 1, ..., q$, and $\beta = 1, ..., pf$. Notice that the number of summands in $\Sigma^{\alpha,\beta}_{\psi h,t}$ and $\Sigma^{\beta}_{h^2,t}$ keeps increasing with time, thus there is a need to calculate them adaptively. Note that:

$$\Sigma^{\beta}_{h^2,t} = (t+1)^{-1} \sum_{\tau=0}^{t} ([\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\beta)^2 = \hat{\mathbf{\Sigma}}_{x,t}(\beta, \beta) \tag{3.35}$$

$$\begin{aligned} \Sigma^{\alpha,\beta}_{\psi h,t} &= (t+1)^{-1} \textstyle\sum_{\tau=0}^{t} \hat{\mathbf{E}}^{t-1}_{\alpha:}(\mathbf{y}(\tau) - \hat{\mathbf{m}}_{y,t})[\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\beta \\ &- (t+1)^{-1} \sum_{\ell=1}^{\beta-1} \hat{\mathbf{D}}^{t}(\alpha, \ell) \sum_{\tau=0}^{t} [\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\ell [\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\beta \\ &- (t+1)^{-1} \sum_{\ell=\beta+1}^{pf} \hat{\mathbf{D}}^{t-1}(\alpha, \ell) \sum_{\tau=0}^{t} [\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\ell [\mathbf{x}(\tau) - \hat{\mathbf{m}}_{x,t}]_\beta \end{aligned} \tag{3.36}$$

Next, it is delineated how the quantities in (3.35) can be found in an online fashion. Note that the first summand of $\Sigma^{\alpha,\beta}_{\psi h,t}$ in (3.35) contains the term $\mathbf{\Sigma}^{1,\beta}_{\psi h,t} - \hat{\mathbf{m}}^{1,\beta}_{\psi h,t}$, where $\mathbf{\Sigma}^{1,\beta}_{\psi h,t} := (t+1)^{-1} \sum_{\tau=0}^{t} \mathbf{y}(\tau)[\mathbf{x}(\tau)]_\beta$ and $\hat{\mathbf{m}}^{1,\beta}_{\psi h,t} := \hat{\mathbf{m}}_{y,t}[\hat{\mathbf{m}}_{x,t}]_\beta$. It follows readily that $\Sigma^{1,\beta}_{\psi h,t}$ and $\hat{\mathbf{m}}_{y,t}$ (or $\hat{\mathbf{m}}_{x,t}$) can be adaptively updated as

$$\Sigma^{1,\beta}_{\psi h,t} = t(t+1)^{-1} \Sigma^{1,\beta}_{\psi h,t-1} + (t+1)^{-1} \mathbf{y}(t)[\mathbf{x}(t)]_\beta, \tag{3.37}$$

$$\hat{\mathbf{m}}_{y,t} = t(t+1)^{-1} \hat{\mathbf{m}}_{y,t-1} + (t+1)^{-1} \mathbf{y}(t). \tag{3.38}$$

Thus, there is no need to store all data history as is the case in the batch algorithm CS-CCA. Further, there is a common part in the second and third terms in $\Sigma^{\alpha,\beta}_{\psi h,t}$ in (3.36), namely

$$\hat{\mathbf{\Sigma}}_{x,t}(\ell, \beta) = (t+1)^{-1} \textstyle\sum_{\tau=0}^{t} [\mathbf{x}(\tau)]_\ell [\mathbf{x}(\tau)]_\beta - [\hat{\mathbf{m}}_{x,t}]_\ell [\hat{\mathbf{m}}_{x,t}]_\beta,$$

both summands in $\hat{\mathbf{\Sigma}}_{x,t}(\ell, \beta)$ can be updated in an adaptive fashion as indicated in (3.37). Note that

$$\begin{aligned} \mathbf{R}_{x,t}(\ell, \beta) &:= (t+1)^{-1} \textstyle\sum_{\tau=0}^{t} [\mathbf{x}(\tau)]_\ell [\mathbf{x}(\tau)]_\beta \\ &= t(t+1)^{-1} \mathbf{R}_{x,t-1}(\ell, \beta) + (t+1)^{-1} [\mathbf{x}(t)]_\ell [\mathbf{x}(t)]_\beta. \end{aligned} \tag{3.39}$$

The same online mechanism can be used to update $\Sigma^{\beta}_{h^2,t}$ without the need to store all data history. Using the same procedure we can derive an updating formula for each of the entries of $\mathbf{E}$. Specifically, $\hat{\mathbf{E}}^t(\alpha,\beta)$ can be obtained as

$$\hat{\mathbf{E}}^t(\alpha,\beta) = \mathbb{F}(\check{\boldsymbol{\psi}}^{'}_{\alpha,\beta,t}, \check{\mathbf{h}}^{'}_{\alpha,\beta,t}, \check{\Sigma}^{\alpha,\beta}_{\psi h,t}, \check{\Sigma}^{\beta}_{h^2,t}, \lambda_{E,\alpha}) \tag{3.40}$$

where $\check{\Sigma}^{\alpha,\beta}_{\psi h,t}$ is obtained via the expression of $\Sigma^{\alpha,\beta}_{\psi h,t}$ in (3.35) after substituting $\hat{\mathbf{E}}^{t-1}$, $\hat{\mathbf{D}}^t$, $\mathbf{x}$, $\mathbf{y}$, $\hat{\mathbf{m}}_{y,t}$ and $\hat{\mathbf{m}}_{x,t}$ with $\hat{\mathbf{D}}^{t-1}$, $\hat{\mathbf{E}}^t$, $\mathbf{y}$, $\mathbf{x}$, $\hat{\mathbf{m}}_{x,t}$ and $\hat{\mathbf{m}}_{y,t}$, respectively. Similarly, $\check{\Sigma}^{\beta}_{h^2,t}$ can be obtained after making the same substitutions in $\Sigma^{\beta}_{h^2,t}$ in (3.35). The quantities $\check{\boldsymbol{\psi}}^{'}_{\alpha,\beta,t}$ and $\check{\mathbf{h}}^{'}_{\alpha,\beta,t}$ can be obtained from the corresponding quantities $\boldsymbol{\psi}^{'}_{\alpha,\beta,t}$, $\mathbf{h}^{'}_{\alpha,\beta,t}$ in (3.33) after applying the following substitutions: $\varepsilon \to v$, $\hat{\mathbf{D}}^t \to \hat{\mathbf{E}}^t$, $\hat{\mathbf{D}}^{t-1} \to \hat{\mathbf{E}}^{t-1}$ and $\mathbf{B}_t \to \hat{\boldsymbol{\Sigma}}_{y,t}(\hat{\mathbf{E}}^{t-1})^T$. Per time instant t one coordinate cycle is applied to update each entry of $\mathbf{D}$ (and $\mathbf{E}$).

3.4.2 Online Distributed S-CCA (ODS-CCA)

An online distributed S-CCA (ODS-CCA) is put forth here for the network setting considered also in Sec. 3.2.2. The starting point for building ODS-CCA will be the separable cost function introduced in (3.16) for DS-CCA, after replacing N with $t+1$ and making it time-varying as in (3.31). As in OCS-CCA the goal is to obtain at every time-instant t continuously refined sparse estimates $\hat{\mathbf{D}}^t$ and $\hat{\mathbf{E}}^t$. As in DS-CCA the resulting cost will be minimized in a coordinate fashion with respect to the $q \times f$ submatrices $\mathbf{D}_j$ (and $\mathbf{E}_j$), while fixing the remaining submatrices to their recent updates. When focusing on minimizing (3.16) (after replacing N with $t+1$) w.r.t. $\mathbf{D}_j$, the Euclidean norms in the first summand in (3.16) will be replaced with

$$\|\textstyle\sum_{i=1}^{p} \hat{\mathbf{E}}_i^{t-1}\mathbf{y}(\tau,i) - \sum_{i=1,i\neq j}^{p} \hat{\mathbf{D}}_i^{t-1}\mathbf{x}(\tau,i) - \mathbf{D}_j\mathbf{x}(\tau,j)\|_2^2,$$

where the most recent updates are used to set all $\mathbf{E}_i$ and $\mathbf{D}_i$, but $\mathbf{D}_j$. As in DS-CCA, K ADMM will be used to form local estimates for the *global* quantities $\sum_{i=1}^{p} \hat{\mathbf{E}}_i^{t-1}\mathbf{y}(\tau,i)$

and $\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{t-1}\mathbf{x}(\tau,i)$ for $\tau = 0,\ldots,t$. However, it is not hard to notice that during time instant t, t parallel ADMM schemes should run to estimate the aforementioned global quantities. As more and more data are acquired and t increases, the related complexity would be proportional to t and become eventually prohibitively high. To this end, we substitute the global terms $\sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(t,i)$ and $\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(t,i)$ in (3.16) with the updates $\sum_{i=1}^{p}\hat{\mathbf{E}}_i^{\tau-1}\mathbf{y}(\tau,i)$ and $\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(\tau,i)$ for $\tau = 0,\ldots,t$, where that data at time τ are multiplied with the latest update for $\mathbf{E}$ and $\mathbf{D}$ at time $\tau-1$, namely $\hat{\mathbf{E}}^{\tau-1}$ and $\hat{\mathbf{D}}^{\tau-1}$. This substitution at time t requires only the estimation of $\sum_{i=1}^{p}\hat{\mathbf{E}}_i^{t-1}\mathbf{y}(t,i)$ and $\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{t-1}\mathbf{x}(t,i)$ via K ADMM iterations, whereas there is no need to re-estimate the past quantities for $\tau = 0,\ldots,t-1$.

As in Sec. 3.2.2, let $\hat{\boldsymbol{\eta}}_j^{\tau}$ and $\hat{\boldsymbol{\mu}}_j^{\tau}$ denote the local estimates for $\sum_{i=1}^{p}\hat{\mathbf{E}}_i^{\tau-1}\mathbf{y}(\tau,i)$ and $\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(\tau,i)$ respectively obtained at sensor j after K ADMM iterations within the time interval $[\tau,\tau+1)$.

Then, the global terms $\sum_{i=1}^{p}\mathbf{E}_i\mathbf{y}(\tau,i)$ will be replaced with the local estimate $\hat{\boldsymbol{\eta}}_j^{\tau}$ at sensor j, while $\sum_{i=1,i\neq j}^{p}\mathbf{D}_i\mathbf{x}(\tau,i)$ with the local estimate $\hat{\boldsymbol{\mu}}_j^{\tau} - \hat{\mathbf{D}}_j^{\tau}\mathbf{x}(\tau,j)$ in (3.16). To prevent the presence of third and fourth-order terms resulting from the last summand in (3.31), this summand is replaced with the following approximate term [similarly to the ones in (3.17) and (3.18)]

$$\varepsilon\|(t+1)^{-1}\sum_{\tau=0}^{t}(\sum_{i=1}^{p}\mathbf{D}_i\mathbf{x}(\tau,i))(\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(\tau,i))^T - \mathbf{I}\|_F^2 \tag{3.41}$$

where the $\mathbf{D}_i$'s have been replaced with the past estimates $\hat{\mathbf{D}}_i^{\tau-1}$ when multiplying $\mathbf{x}(\tau, i)$. Then, after replacing the global terms $\sum_{i=1,i\neq j}^{p} \hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(\tau,i)$ and $\sum_{i=1}^{p} \hat{\mathbf{E}}_i^{\tau-1}\mathbf{y}(\tau,i)$ with their local estimates $\hat{\boldsymbol{\mu}}_j^{\tau}$ and $\hat{\boldsymbol{\eta}}_j^{\tau}$ at sensor j the cost

$$\begin{aligned}
&(t+1)^{-1}\sum\nolimits_{\tau=0}^{t}\|\hat{\boldsymbol{\eta}}_j^{\tau} - \hat{\boldsymbol{\mu}}_j^{\tau} + \hat{\mathbf{D}}_j^{\tau-1}\mathbf{x}(\tau,j) - \mathbf{D}_j\mathbf{x}(\tau,j)\|_2^2 \\
&+\varepsilon\|(t+1)^{-0.5}\sum\nolimits_{\tau=0}^{t}\Big(\hat{\boldsymbol{\mu}}_j^{\tau} - \hat{\mathbf{D}}_j^{\tau-1}\mathbf{x}(\tau,j) \\
&+\mathbf{D}_j\mathbf{x}(\tau,j)\big)\,(\hat{\boldsymbol{\mu}}_j^{\tau})^T - \mathbf{I}\|_F^2 + \sum\nolimits_{\rho=1}^{q}\lambda_{D,\rho}\|\mathbf{D}_{j,\rho:}\|_1
\end{aligned} \tag{3.42}$$

is obtained and will tackled at sensor j to update the entries of $\mathbf{D}_j$. The cost in (3.42) for $\mathbf{D}_j$ can be further written as

$$(t+1)^{-1}\sum\nolimits_{\tau=0}^{t}\|\boldsymbol{\phi}_{j,\tau} - \mathbf{D}_j\mathbf{x}(\tau,j)\|_2^2 + \sum\nolimits_{\rho=1}^{q}\lambda_{D,\rho}\|\mathbf{D}_{j,\rho:}\|_1 + \|\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,1}^{j} - \mathbf{D}_j\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,2}^{j}\|_F^2 \tag{3.43}$$

where $\boldsymbol{\phi}_{j,\tau} := \hat{\boldsymbol{\eta}}_j^{\tau} - \hat{\boldsymbol{\mu}}_j^{\tau} + \hat{\mathbf{D}}_j^{\tau-1}\mathbf{x}(\tau,j)$, $\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,1}^{j} = \sqrt{\varepsilon}\boldsymbol{\Sigma}_{\hat{\mu}^2,t}^{j} - \sqrt{\varepsilon}\boldsymbol{\Sigma}_{Dx\hat{\mu},t}^{j} - \sqrt{\varepsilon}\mathbf{I}$ and $\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,2}^{j} = -\sqrt{\varepsilon}\boldsymbol{\Sigma}_{x\hat{\mu},t}$, in which,

$$\begin{aligned}
\boldsymbol{\Sigma}_{\hat{\mu}^2,t}^{j} &:= (t+1)^{-1}\sum\nolimits_{\tau=0}^{t}\hat{\boldsymbol{\mu}}_j^{\tau}(\hat{\boldsymbol{\mu}}_j^{\tau})^T \\
\boldsymbol{\Sigma}_{Dx\hat{\mu},t}^{j} &:= (t+1)^{-1}\sum\nolimits_{\tau=0}^{t}\hat{\mathbf{D}}_j^{\tau-1}\mathbf{x}(\tau,j)(\hat{\boldsymbol{\mu}}_j^{\tau})^T \\
\boldsymbol{\Sigma}_{x\hat{\mu},t}^{j} &:= (t+1)^{-1}\sum\nolimits_{\tau=0}^{t}\mathbf{x}(\tau,j)(\hat{\boldsymbol{\mu}}_j^{\tau})^T,
\end{aligned} \tag{3.44}$$

Each of these quantities in (3.44) can be updated in an online fashion as, e.g., $\boldsymbol{\Sigma}_{\hat{\mu}^2,t}^{j} = t(t+1)^{-1}\boldsymbol{\Sigma}_{\hat{\mu}^2,t-1}^{j} + (t+1)^{-1}\hat{\boldsymbol{\mu}}_j^{t}(\hat{\boldsymbol{\mu}}_j^{t})^T$. The same updating process can be applied for the other two matrices in (3.44).

Following a similar strategy as before the cost in (3.43) is minimized at sensor j w.r.t. one entry of $\mathbf{D}_j$, say $\mathbf{D}_j(\alpha,\beta)$, while keeping the rest fixed in a coordinate descent fashion.

In detail, $\hat{\mathbf{D}}_j^t(\alpha,\beta)$ can be found as in (3.32) where the involved quantities $\boldsymbol{\psi}'_{\alpha,\beta,t}$, $\mathbf{h}'_{\alpha,\beta,t}$, $\psi_{\alpha,\beta,\tau}$ and $h_{\alpha,\beta,\tau}$ are replaced respectively by the local quantities at sensor j

$$\begin{aligned}\boldsymbol{\psi}'_{j,\alpha,\beta,t} := [\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,1}]_{\alpha,:} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}_j^t(\alpha,\ell)[\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,2}]_{\ell,:}\\ - \textstyle\sum_{\ell=\beta+1}^{f}\hat{\mathbf{D}}_j^{t-1}(\alpha,\ell)[\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,2}]_{\ell,:},\end{aligned} \tag{3.45}$$

and $\mathbf{h}'_{j,\alpha,\beta,t} := [\tilde{\boldsymbol{\Sigma}}_{x\hat{\mu},t,2}]_{\beta,:}$, while,

$$\begin{aligned}\psi_{j,\alpha,\beta,\tau} := (t+1)^{-1/2}[[\boldsymbol{\varphi}_{j,\tau}]_{\alpha} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}_j^t(\alpha,\ell)[\mathbf{x}(\tau,j)]_{\ell}\\ - \textstyle\sum_{\ell=\beta+1}^{f}\hat{\mathbf{D}}_j^{t-1}(\alpha,\ell)[\mathbf{x}(\tau,j)]_{\ell}],\end{aligned}$$

and $h_{j,\alpha,\beta,\tau} := (t+1)^{-1/2}[\mathbf{x}(\tau,j)]_{\beta}$.

Note that $\hat{\mathbf{D}}_j^t(\alpha,\beta)$ can be updated via (3.34) after using the quantities in (3.45) to form $\Sigma_{\psi h,t}^{j,\alpha,\beta} := \sum_{\tau=0}^{t}\psi_{j,\alpha,\beta,\tau}h_{j,\alpha,\beta,\tau}$ and $\Sigma_{h^2,t}^{j,\beta} := \sum_{\tau=0}^{t}h_{j,\alpha,\beta,\tau}^2$ that replace $\Sigma_{\psi h,t}^{\alpha,\beta}$ and $\Sigma_{h^2,t}^{\beta}$ respectively in (3.34). Specifically:

$$\begin{aligned}\Sigma_{\psi h,t}^{j,\alpha,\beta} := \Sigma_{\phi x,t}^{j,\alpha,\beta} - \textstyle\sum_{\ell=1}^{\beta-1}\hat{\mathbf{D}}_j^t(\alpha,\ell)\Sigma_{x_{\tau,j},t}(\ell,\beta)\\ - \textstyle\sum_{\ell=\beta+1}^{f}\hat{\mathbf{D}}_j^{t-1}(\alpha,\ell)\Sigma_{x_{\tau,j},t}(\ell,\beta) \text{ where}\end{aligned} \tag{3.46}$$

$$\begin{aligned}\Sigma_{\phi x,t}^{j,\alpha,\beta} &:= (t+1)^{-1}\textstyle\sum_{\tau=0}^{t}[\boldsymbol{\varphi}_{j,\tau}]_{\alpha}[\mathbf{x}(\tau,j)]_{\ell},\\ \Sigma_{x_{\tau,j},t}(\ell,\beta) &:= (t+1)^{-1}\textstyle\sum_{\tau=0}^{t}[\mathbf{x}(\tau,j)]_{\ell}[\mathbf{x}(\tau,j)]_{\beta}\end{aligned}$$

Note that the quantities $\Sigma_{\phi x,t}^{j,\alpha,\beta}$ and $\Sigma_{x_{\tau,j},t}(\ell,\beta)$ can be updated in an online fashion at sensor j as described in (3.37) which facilitates the updating of $\Sigma_{\psi h,t}^{j,\alpha,\beta}$. The updating of $\Sigma_{h^2,t}^{j,\beta} = \Sigma_{x_{\tau,j},t}(\beta,\beta)$ can be carried out in the exact same way. As mentioned earlier the update $\hat{\mathbf{D}}_j^t(\alpha,\beta)$ is formed, at sensor j, as

$$\hat{\mathbf{D}}^t(\alpha,\beta) = \mathbb{F}(\boldsymbol{\psi}'_{j,\alpha,\beta,t}, \mathbf{h}'_{j,\alpha,\beta,t}, \Sigma_{\psi h,t}^{j,\alpha,\beta}, \Sigma_{h^2,t}^{j,\beta}, \lambda_{D,\alpha}) \tag{3.47}$$

where $\boldsymbol{\psi}'_{j,\alpha,\beta,t}, \mathbf{h}'_{j,\alpha,\beta,t}$ given in (3.45). The same process can be repeated for obtaining updates for the entries of $\mathbf{E}$. ODS-CCA is summarized next.

At time t, each sensor j estimates the global quantities $\sum_{i=1}^{p}\hat{\mathbf{E}}_i^{t-1}\mathbf{y}_{t,i}$ and $\sum_{i=1}^{p}\hat{\mathbf{D}}_i^{t-1}\mathbf{y}_{t,i}$ by applying $2K$ ADMM iterations that result the estimates $\hat{\boldsymbol{\eta}}_j^t$ and $\hat{\boldsymbol{\mu}}_j^t$. This is a basic difference with the batch counterpart DS-CCA in Sec. III-B, where all t quantities $\hat{\mathbf{D}}^{t-1}\mathbf{x}(\tau)$ and $\hat{\mathbf{E}}^{t-1}\mathbf{y}(\tau)$ for $\tau = 0,\ldots,t$ need to be estimated at time instant t. That requires a number of $2tK$ ADMM iterations which are constantly growing with time in order to process the newly acquired data. Taking into account the communication complexity per ADMM iteration (see Sec. 3.2.2), here sensor j receives $2(2q|\mathcal{N}_j|K)$ scalars and transmits $2q(|\mathcal{N}_j|+1)K$ scalars from/to its neighbors in $\mathcal{N}_j$ after $2K$ ADMM iterations applied to compute $\hat{\boldsymbol{\eta}}_j^t$ and $\hat{\boldsymbol{\mu}}_j^t$. In contrast in DS-CCA in Sec. III-B, the latter quantities have to scale up by a factor t. Thus, DS-CCA has a lower computational and communication complexity. Nonetheless DS-CCA will demonstrate a better performance compared to ODS-CCA when clustering sensors.

3.5 S-CCA Properties

Next, it is shown that the S-CCA framework in (3.5) has the capability to return sparse estimates $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$, in which every row contains nonzero values at the entries corresponding to sensor measurements in $\mathbf{x}(t)$ and $\mathbf{y}(t)$ that contain information about the same source, whereas zeros correspond to entries in $\mathbf{x}(t)$ and $\mathbf{y}(t)$ that contain only noise. To establish this property it is assumed that a sufficiently high number of data are available ($N \to \infty$), in which case from the law of large numbers it follows that $\hat{\boldsymbol{\Sigma}}_x$ and $\hat{\boldsymbol{\Sigma}}_y$ converge to their ensemble counterpart $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_y$, respectively. Then, the sample-based cost in (3.5) converges to the ensemble-based cost

$$\begin{aligned}\arg\min_{\mathbf{D},\mathbf{E}}\text{tr}[&\mathbf{E}\boldsymbol{\Sigma}_y\mathbf{E}^T + \mathbf{D}\boldsymbol{\Sigma}_x\mathbf{D}^T - \mathbf{E}\boldsymbol{\Sigma}_{yx}\mathbf{D}^T - \mathbf{D}\boldsymbol{\Sigma}_{xy}\mathbf{E}^T]\\ &+ v||\mathbf{E}\boldsymbol{\Sigma}_y\mathbf{E}^T - \mathbf{I}||_F^2 + \varepsilon||\mathbf{D}\boldsymbol{\Sigma}_x\mathbf{D}^T - \mathbf{I}||_F^2\\ &+ \textstyle\sum_{\rho=1}^{q}\lambda_{E,\rho}||\mathbf{E}_{\rho:}^T||_1 + \sum_{\rho=1}^{q}\lambda_{D,\rho}||\mathbf{D}_{\rho:}^T||_1. \end{aligned} \tag{3.48}$$

Let $\mathbf{D}_e, \mathbf{E}_e$ denote one pair of minimizers in (3.48), while $\mathbf{\Sigma}_{xy}$ denotes the ensemble cross-covariance between $\{\mathbf{x}(t)\}_{t=0}^{N-1}$ and $\{\mathbf{y}(t)\}_{t=0}^{N-1}$. It is studied how the nonzero and zero entries are allocated across the rows of the estimates $\mathbf{D}_e, \mathbf{E}_e$. To facilitate the analysis, an entry of $\mathbf{D}_e$ (or $\mathbf{E}_e$), say $\mathbf{D}_e(\alpha, \beta)$, will be considered nonzero if $|\mathbf{D}_e(\alpha, \beta)| > \delta$, where δ is an arbitrarily small positive value. It is demonstrated next that for proper $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$, the S-CCA minimization framework returns rows $\mathbf{D}_{e,\rho:}$ and $\mathbf{E}_{e,\rho:}$ whose nonzero entries indices correspond to sensors sensing the same source signal for $\rho = 1, ..., q$, while the zero entries correspond to noisy sensor measurements.

In the subsequent analysis no assumptions are made about the data model which can be nonlinear as outlined in (3.1). Further, it is assumed that each sensor observes at most one source, thus the source-based sensor clusters $\mathcal{S}_m$ are not overlapping. This results nonoverlapping groups of correlated entries in $\mathbf{x}(t)$ and $\mathbf{y}(t)$. Since the sources are stationary the groups of correlated entries are the same in both $\mathbf{x}(t)$ and $\mathbf{y}(t)$, since $\mathbf{x}(t)$ contains a delayed version of the elements in $\mathbf{y}(t)$. Its follows readily that in the aforementioned setting the (cross-)covariance matrices $\mathbf{\Sigma}_x$,$\mathbf{\Sigma}_y$, $\mathbf{\Sigma}_{xy}$ and $\mathbf{\Sigma}_{yx}$ will be block diagonal after properly permuting their rows and columns. Let the M different groups of correlated entries in $\mathbf{x}(t)$, and $\mathbf{y}(t)$ be denoted by $\varsigma_1, \ldots, \varsigma_M$; data entries belonging to different groups ς_m are uncorrelated with each other since the different sources are uncorrelated. A proper permutation matrix $\mathbf{P}$ can be applied in $\mathbf{x}(t)$ (or $\mathbf{y}(t)$) such that entries that belong to the same group are contiguous in the permuted vector $\mathbf{x}_P(t) := \mathbf{P}\mathbf{x}(t) = [\mathbf{x}_{\varsigma_1}(t) \ldots \mathbf{x}_{\varsigma_M}(t)]^T$, where $\mathbf{x}_{\varsigma_m}(t)$ correspond to the sensor measurements acquired during interval $[t-1, t-f]$ that contain information about source $\mathbf{s}_m$ for $m = 1, \ldots, M$. Thus, $\mathbf{x}_P(t)$ (and $\mathbf{y}_P(t)$) has a block diagonal covariance matrix, namely $\mathbf{P}\mathbf{\Sigma}_x\mathbf{P}^T = \text{bdiag}(\mathbf{\Sigma}_{\mathbf{x}_{\varsigma_1}} \ldots \mathbf{\Sigma}_{\mathbf{x}_{\varsigma_M}})$, where $\mathbf{\Sigma}_{\mathbf{x}_{\varsigma_m}}$ corresponds to the covariance of $\mathbf{x}_{\varsigma_m}(t)$. A similar process can be applied to $\mathbf{y}(t)$ to obtain $\mathbf{y}_P(t) := \mathbf{P}\mathbf{y}(t) = [\mathbf{y}_{\varsigma_1}(t) \ldots \mathbf{y}_{\varsigma_M}(t)]^T$ whose covariance matrix is also block diagonal.

Let $\mathbf{v}(\mathcal{F})$ denote the entries of vector $\mathbf{v}$ with indices belonging to the set $\mathcal{F}$. It is established in Appendix C that

Theorem 1: For block diagonal covariance matrices $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_y$ and if $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^{q}$ are selected properly, then for any arbitrarily small $\delta > 0$ the minimization in (3.48) admits an optimal solution $\mathbf{D}_e, \mathbf{E}_e$ satisfying for $\rho = 1, ..., q$

$$||\mathbf{D}_{e,\rho:}(\bar{\mathcal{Z}}_{i_\rho})||_1 < \delta, \text{ and } ||\mathbf{D}_{e,\rho:}(\mathcal{Z}_{i_\rho})||_1 \geq \xi(\lambda_{D,\rho}) > 0 \tag{3.49}$$

$$||\mathbf{E}_{e,\rho:}(\bar{\mathcal{Z}}_{i_\rho})||_1 < \delta, \text{ and } ||\mathbf{E}_{e,\rho:}(\mathcal{Z}_{i_\rho})||_1 \geq \xi(\lambda_{E,\rho}) > 0 \tag{3.50}$$

where $\bar{\mathcal{Z}}_{i_\rho}$ is the complement of the support $\mathcal{Z}_{i_\rho}$ of the i_ρth dominant eigenvector $\mathbf{U}_{x,:i_\rho}$ of $\boldsymbol{\Sigma}_x$ (or $\boldsymbol{\Sigma}_y$) and $i_\rho \in 1, ..., q$. The constants $\xi(\lambda_{D,\rho}), \xi(\lambda_{E,\rho})$ depend only on $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ respectively and are strictly positive.

Theorem 1 states that S-CCA can generate optimal matrices ($\mathbf{D}_e$, $\mathbf{E}_e$) whose rows' support is a subset of the truth support of the q dominant eigenvectors in $\mathbf{U}_x$ and $\mathbf{U}_y$ (which have the same block diagonal structure as $\boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y$). This is possible since for the ρth row of $\mathbf{D}_e$ (and $\mathbf{E}_e$), there is a corresponding i_ρ column of the eigenvector matrix $\mathbf{U}_x$ (and $\mathbf{U}_y$) such that $||\mathbf{D}_{e,\rho:}(\bar{\mathcal{Z}}_{i_\rho})||_1 < \delta$ for arbitrarily small δ, while $||\mathbf{D}_{e,\rho:}(\mathcal{Z}_{i_\rho})||_1 \geq \xi(\lambda_{D,\rho}) > 0$ (strictly positive). Thus, all the nonzero entries of $\mathbf{D}_{e,\rho:}$, with magnitude exceeding δ, will have indices in $\mathcal{Z}_{i_\rho} :=$ support($\mathbf{U}_{x,:i_\rho}$) where $\rho = 1, \ldots, q$. This happens since: i) $||\mathbf{D}_{e,\rho:}(\bar{\mathcal{Z}}_{i_\rho})||_1$ can be made arbitrarily small, thus all entries of $\mathbf{D}_{e,\rho:}$ can be driven arbitrarily close to zero; and ii) $||\mathbf{D}_{e,\rho:}(\mathcal{Z}_{i_\rho})||_1$ is strictly positive while $\xi(\lambda_{D,\rho}) > \delta$ by making δ arbitrarily small. Thus, some of the entries of $\mathbf{D}_{e,\rho:}$ with indices in $\mathcal{Z}_{i_\rho}$ must have magnitude greater than δ. The number of nonzero entries in $\mathbf{D}_{e,\rho:}(\mathbf{Z}_{i_\rho})$ is determined by $\lambda_{D,\rho}$. Thus, if $\lambda_{D,\rho}$ is selected such that $||\mathbf{D}_{e,\rho:}||_0 = ||\mathbf{U}_{x,i_\rho}||_0$, then recovery of the whole support $\mathcal{Z}_{i_\rho}$ is ensured.

Note that for an infinite number of data the standard CCA solution $\breve{\mathbf{D}}$ and $\breve{\mathbf{E}}$ will also be sparse with the nonzero entries across the rows of $\breve{\mathbf{D}}$ (or $\breve{\mathbf{E}}$) pointing at the sensors sensing a common source. Thus, Theorem 1 states that the S-CCA formulation makes sense because they assign (non)zero entries in some meaningful positions rather than place them arbitrarily. However, as will become apparent via extensive numerical tests the true advantage of S-CCA (in terms of clustering sensors correctly) is in settings with a finite and small number of data in which case the corresponding $\breve{\mathbf{D}}$ and $\breve{\mathbf{E}}$ do not really give any insight on the informative sensors. Another important aspect of Theorem 1 is that perfect recovery of the different groups of informative sensors is ensured no matter what the underlying data model is (nonlinear in general); this is not the case in [65].

3.6 Numerical Tests

The performance of the batch CS-CCA and DS-CCA, as well as online OCS-CCA and ODS-CCA schemes is tested and compared with existing alternatives in terms of probability of correctly clustering sensor measurements based on their source content. The novel schemes will be compared with i) standard CCA [6, Chpt. 10]; ii) CS-CCA for zero sparsity-controlling coefficients ($\lambda = 0$); iii) the *centralized* sparse CCA scheme in [79] abbreviated as PMD; and iv) K-means algorithm (see e.g., [38]) used to cluster the different sensors into groups using the data vectors $\boldsymbol{\chi}_j := [x_j(0), \ldots, x_j(N-1)]^T$ acquired at sensor j over time-horizon $[0, N-1]$ assuming the number of sources M is known and the centroids initialized uniformly at random. The distributed algorithms put forth here (namely DS-CCA and ODS-CCA) are tested in a sensor network of $p = 15$ randomly placed sensors within a 2-D area which is represented by two normalized dimensions $[0, 1] \times [0, 1]$, while the sensor communication range is set to 0.4.

In the following numerical tests both linear and nonlinear data models are considered as specified next. An autoregressive evolution model is used for sources $\{\mathbf{s}_m\}_{m=1}^M$, i.e.,

$$\mathbf{s}_m(t) = \textstyle\sum_{\tau=1}^{L} \mathbf{F}_{m,\tau}\mathbf{s}_m(t-\tau) + \mathbf{u}_m(t) \tag{3.51}$$

in which $\mathbf{F}_{m,\tau} \in \mathbb{R}^{r\times r}$ correspond to the autoregressive coefficients, while L is the order of the AR process and $\mathbf{u}_m(t)$ is white perturbation noise with zero-mean and variance $\sigma^2_{u_m}\mathbf{I}_{r\times r}$. In the simulations the sources are scalar, i.e., $r = 1$.

First, three different non-overlapping scenarios (each sensor observes no more than one source) are considered to test the performance of CS-CCA and DS-CCA and compare it with other centralized approaches. The process order $L = 1$ in (3.51) is applied here, while the AR coefficients are selected such that $|F_{m,1}| \leq 1$. Moreover, the memory length parameter here is set $f = 1$. The first testing scenario treats a linear case where $h_{m,j}(\mathbf{s}_m(t)) = h_{m,j} \cdot s_m(t)$, where $h_{m,j}$ is normal if sensor j observes source m, otherwise $h_{m,j}$ is zero. Also note that $\mathbf{h}_m(s_m(t)) := [h_{m,1}(s_m(t)), ..., h_{m,p}(s_m(t))]^T$. In the non-overlapping configuration considered here sensors $\{1, ..., 5\}$ observe source $s_1(t)$, sensors $\{6, ..., 10\}$ observe source $s_2(t)$ while sensors $\{11, ..., 15\}$ sense just noise. In the second testing case (denoted as Nonlinear Case 1), sensors $\{1, ..., 5\}$ observe source $s_1(t)$ and the first five entries in $\mathbf{h}_1(s_1(t))$ are equal to $[h_{1,1}s_1(t), h_{1,2}s_1(t), h_{1,3}s_1(t), h_{1,4}s_1(t), h_{1,5}s_1(t)]$ while the last 10 entries are equal to zero. Sensors $\{6, ..., 10\}$ observe source $s_2(t)$ thus the corresponding entries in $\mathbf{h}_2(s_2(t))$ are given by $[h_{2,1}s_2^2(t), h_{2,2}s_2^2(t), h_{2,3}s_2^2(t), h_{2,4}s_2^2(t), h_{2,5}s_2^2(t)]$ while the rest are zero. Also, the coefficients $h_{m,j}$ are normally distributed. In the third testing case (denoted as Nonlinear Case 2), sensors $\{1, ..., 5\}$ observe source $s_1(t)$ and the first five entries in $\mathbf{h}_1(s_1(t))$ are equal to $[h_{1,1}s_1(t), h_{1,2}s_1^{1.1}(t), h_{1,3}s_1^{1.2}(t), h_{1,4}s_1^{1.3}(t), h_{1,5}s_1^{1.4}(t)]$ while the rest 10 entries are equal to zero. Sensors $\{6, ..., 10\}$ observe source $s_2(t)$, thus the corresponding entries in $\mathbf{h}_2(s_2(t))$ are given by $[h_{2,1}s_2(t), h_{2,2}s_2^{1.1}(t), h_{2,3}s_2^{1.2}(t), h_{2,4}s_2^{1.3}(t), h_{2,5}s_2^{1.4}(t)]$ while the rest are zero.

In Fig. 4.2 we compare the probability of correct sensor clustering among CS-CCA, DS-CCA for a different number of ADMM iterations ($K = 5$ and $K = 15$), standard CCA, PMD, K-means and CS-CCA for zero sparsity-controlling coefficients in the linear case. The sparsity-controlling coefficients λ^ρ_D or λ^ρ_E in 1) and 2) are selected using the algorithm in Sec. 3.3. The corresponding sparsity-controlling coefficients in PMD are selected by cross-validation (details in [79]). The multipliers $\mathbf{v}_{j,0}^{j'}$ and $\mathbf{w}_{j,0}^{j'}$ in the DS-CCA algorithm implementation are initialized to be zero. Fig. 4.2 depicts that CS-CCA achieves the best performance, while the probability of clustering sensor measurements reaches unity (corroborating Theorem 1) as the number of data vectors $\{\mathbf{x}(t), \mathbf{y}(t)\}_{t=0}^{N-1}$ goes to infinity ($N \to \infty$). It is also of interest that DS-CCA yields better performance than other centralized S-CCA approaches, e.g. PMD and K-means. Notice also that DS-CCA achieves a performance which improves as the number of ADMM iterations K increases leading to better estimates. Note that as K increases the DS-CCA performance curve will gradually overlap with the CS-CCA ($K \to \infty$). Also it can be seen that if sparsity is not employed, i.e., $\lambda_{E,\rho} = \lambda_{D,\rho} = 0$, then the performance of CS-CCA deteriorates significantly. In the same way standard CCA for small number of samples has much worse performance than CS-CCA and PMD, though as explained earlier for $N \to \infty$ the probability gradually reaches one since the standard CCA solution $\breve{\mathbf{D}}, \breve{\mathbf{E}}$ will have zeros at the right entries as $N \to \infty$. Similar conclusions can be drawn from Fig. 4.5, which shows the performance of the aforementioned schemes for the two nonlinear models considered here. This signifies the capability of CS-CCA to correctly cluster sensors even in nonlinear settings.

In Figs. 4.6 and 4.7 the performance of the online OCS-CCA and ODS-CCA algorithms is compared with the batch counterparts CS-CCA and DS-CCA, as well as K-means. The tests are carried for the linear setting in Fig. 4.6 and for the nonlinear case 2 in Fig. 4.7. The batch algorithm at every time instant t has to process all data $N = t$, leading to a prohibitively large complexity as explained in Secs. 3.4.1 and 3.4.2. This is not the case in the online counterparts that process new data in an incremental way and fixed complexity (not dependent on time). As expected the clustering performance of the online schemes is worse than the batch algorithms, though the probability of correctly clustering the sensors increases with t. Nonetheless, even the online schemes achieve better performance compared to the centralized schemes PMD and K-means.

The role of the memory length parameter f is examined next in the clustering performance of CS-CCA. To this end, two different AR model for the $M = 2$ sources are considered having order $L = 10$, while the AR-coefficients are selected such that $|F_{m,\tau}| < 1$ for $\tau = 1, ..., L$ and $m = 1, 2$. The clustering performance of CS-CCA is tested for the linear setting and nonlinear case 2, for two different memory length parameter values, namely $f = 1$ and $f = 5$. Fig. 3.5 indicates that increasing the memory length parameter in the current test setting boosts the performance of CS-CCA, especially in the nonlinear case. Clearly, the larger f is the more CS-CCA takes advantage of the temporal correlations present in the data due to the AR-10 source models.

The capability of CS-CCA to perfectly cluster sensors for an increasing number of sensor data was proved for a non-overlapping setting where each sensor can observe at most one field source. For the overlapping case where sensors could sense multiple sources there are no theoretical guarantees for perfect clustering so far. Nonetheless, in Fig. 3.6 CS-CCA is tested in an overlapping setting and compared with PMD and K-means. Specifically, a 15-sensor network is considered while there are $M = 3$ sources in the field evolving according to an AR-1 model. Sensors $\{1, 2, 3\}$ observe source $s_1(t)$, sensors $\{4, 5, 6\}$ observe source $s_2(t)$ and sensors $\{7, 8, 9\}$ observe both sources $s_1(t)$, and source $s_3(t)$. In the *linear* case the mappings in (3.1) are set such that entries $\{1, 2, 3\}$ in $\mathbf{h}_1(s_1(t))$ are set as $[h_{1,1}s_1(t), h_{1,2}s_1(t), h_{1,3}s_1(t)]$, entries $\{7, 8, 9\}$ are set as $[h_{1,4}s_1(t), h_{1,5}s_1(t), h_{1,6}s_1(t)]$ while the rest of the entries are set equal to zero. Entries $\{4, 5, 6\}$ in $\mathbf{h}_2(s_2(t))$ are set as $[h_{2,1}s_2(t), h_{2,2}s_2(t), h_{2,3}s_2(t)]$, while the rest of the entries are equal to zero. Similarly, entries $\{7, 8, 9\}$ in $\mathbf{h}_3(s_3(t))$ are set as $[h_{3,1}s_3(t), h_{3,2}s_3(t), h_{3,3}s_3(t)]$, and the remaining ones set to zero. For the *nonlinear* setting the mappings in (3.1) is set such that entries $\{1, 2, 3\}$ in $\mathbf{h}_1(s_1(t))$ are set as $[h_{1,1}s_1(t), h_{1,2}s_1(t), h_{1,3}s_1(t)]$, entries $\{7, 8, 9\}$ are set as

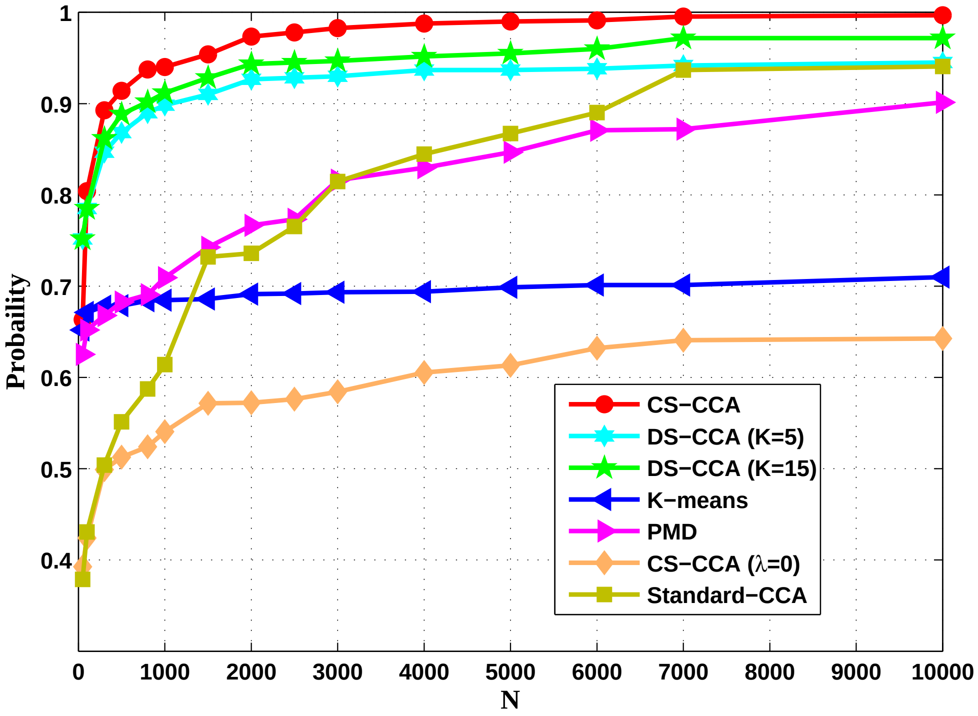

Figure 3.1. Probability of correctly clustering sensors (P_c) vs. number of data vectors $\mathbf{x}(t), \mathbf{y}(t)$ in a linear setting.

$[h_{1,4}s_1^2(t), h_{1,5}s_1^2(t), h_{1,6}s_1^2(t)]$ while the rest of the entries are set equal to zero. Entries $\{4, 5, 6\}$ in $\mathbf{h}_2(s_2(t))$ are set as $[h_{2,1}s_2^2(t), h_{2,2}s_2^2(t), h_{2,3}s_2^2(t)]$, while the rest of the entries are equal to zero. Similarly, entries $\{7, 8, 9\}$ in $\mathbf{h}_3(s_3(t))$ are set as $[h_{3,1}s_3^2(t), h_{3,2}$ $s_3^2(t), h_{3,3}s_3^2(t)]$, and the remaining ones set to zero. The coefficients $h_{m,j}$ are normally distributed. Fig. 3.6 shows that the CS-CCA framework achieves significantly better performance in both the linear and nonlinear settings w.r.t. PMD and K-means. The probability of correct sensor clustering is gradually increasing with N, especially for the linear model, reaching a probability much higher than PMD and K-means, which may not be one though.

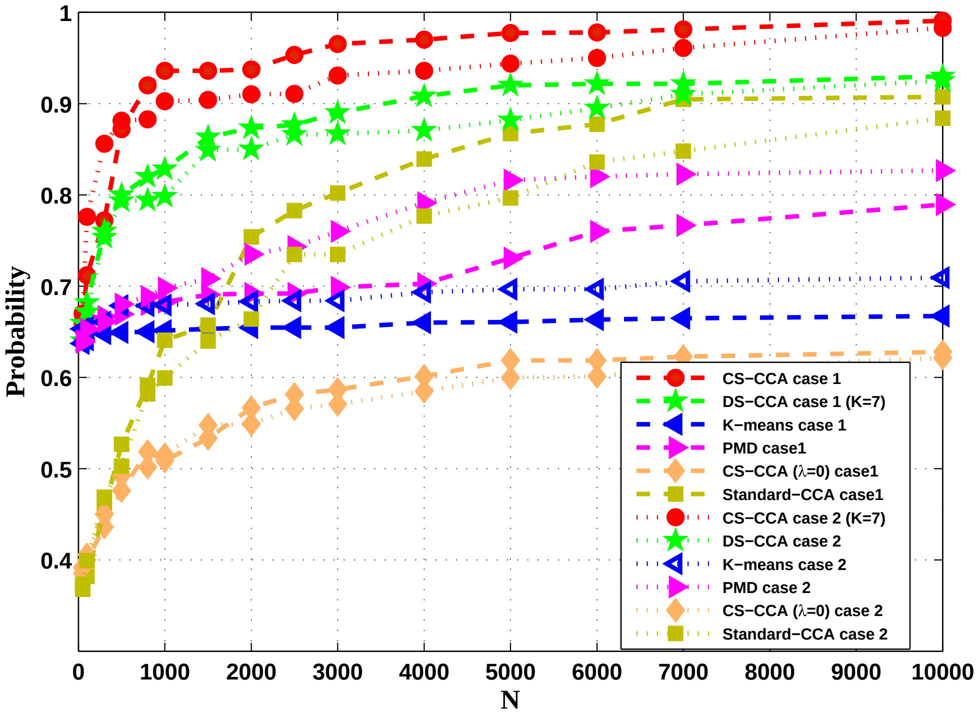

Figure 3.2. Probability of correctly clustering sensors vs. number of data vectors $\mathbf{x}(t), \mathbf{y}(t)$ in nonlinear settings.

3.7 Conclusions

A sparsity-inducing CCA framework was put forth and applied to clustering sensor measurements based on their source content. Norm-one regularization was utilized to impose the sparsity-requirements and recover the different sensor clusters. Relying on coordinate descent techniques a novel centralized algorithm (CS-CCA) is developed to minimize the associated cost and perform clustering. A distributed iterative approach (DS-CCA) that relies only on single-hop inter-sensor communications is further developed using the alternating direction method of multipliers. Online algorithmic implementations (OCS-CCA, ODS-CCA), having manageable communication, computational and storage cost are also derived for settings where sensors are constantly acquiring data. The potential of the proposed sparse-CCA framework in correctly recovering is established theoretically, while

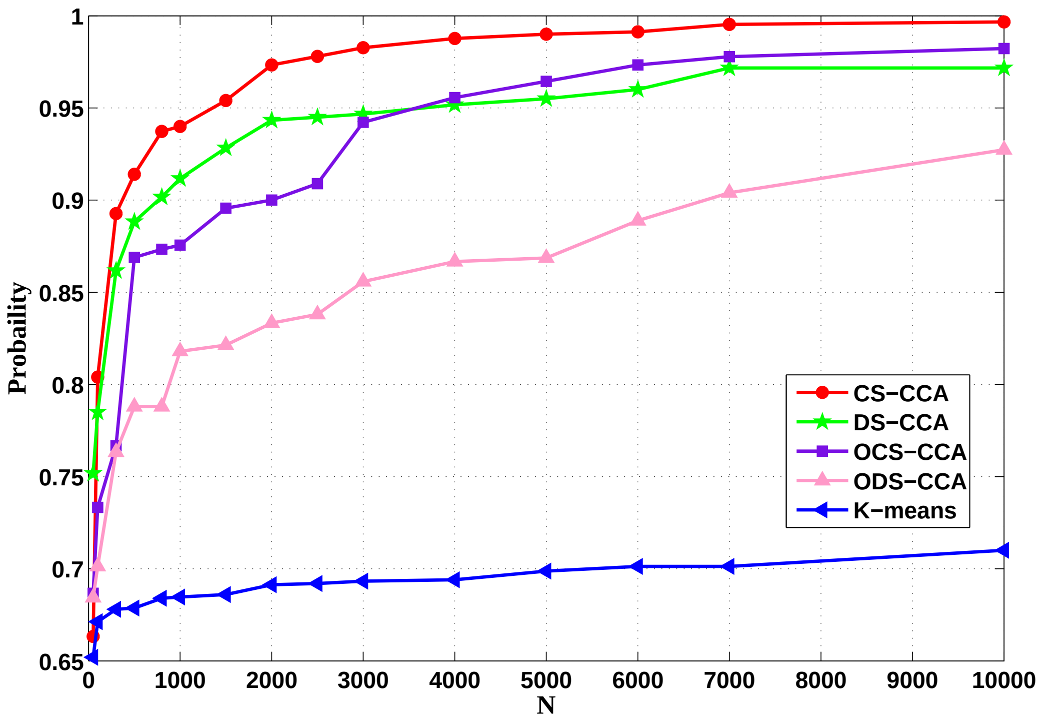

Figure 3.3. Probability of correctly clustering sensors for the online CS-CCA framework vs. time index t in a linear setting.

extensive numerical results demonstrate the advantages of the proposed approach over existing alternatives.

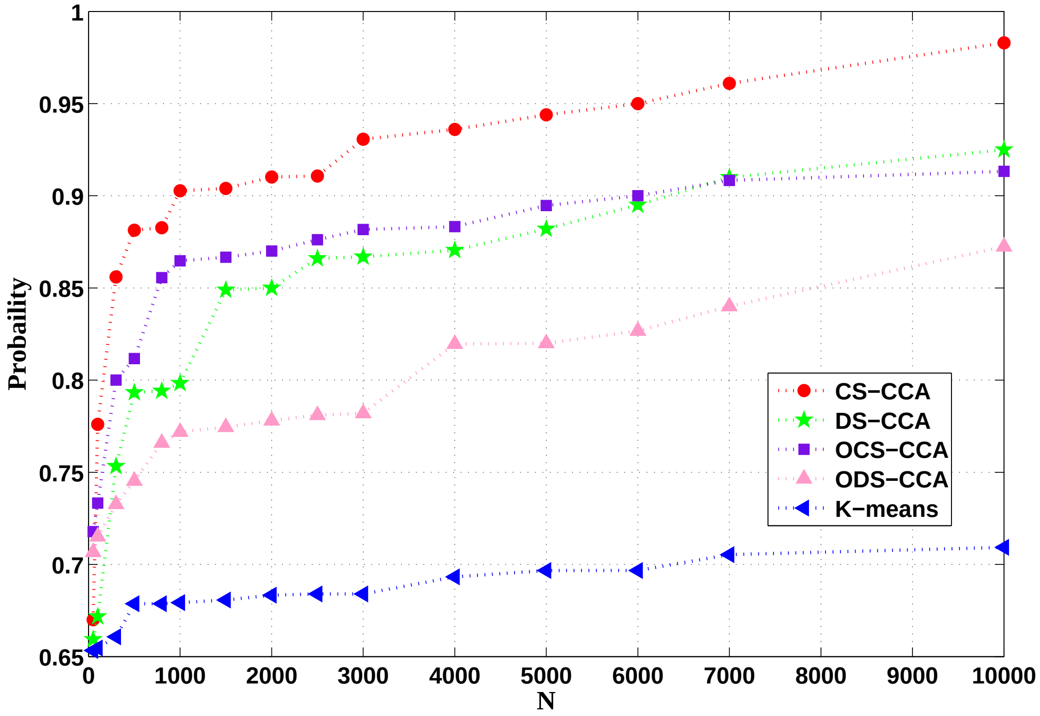

Figure 3.4. Probability of correctly clustering sensors for the online CS-CCA framework vs. time index t in a nonlinear setting.

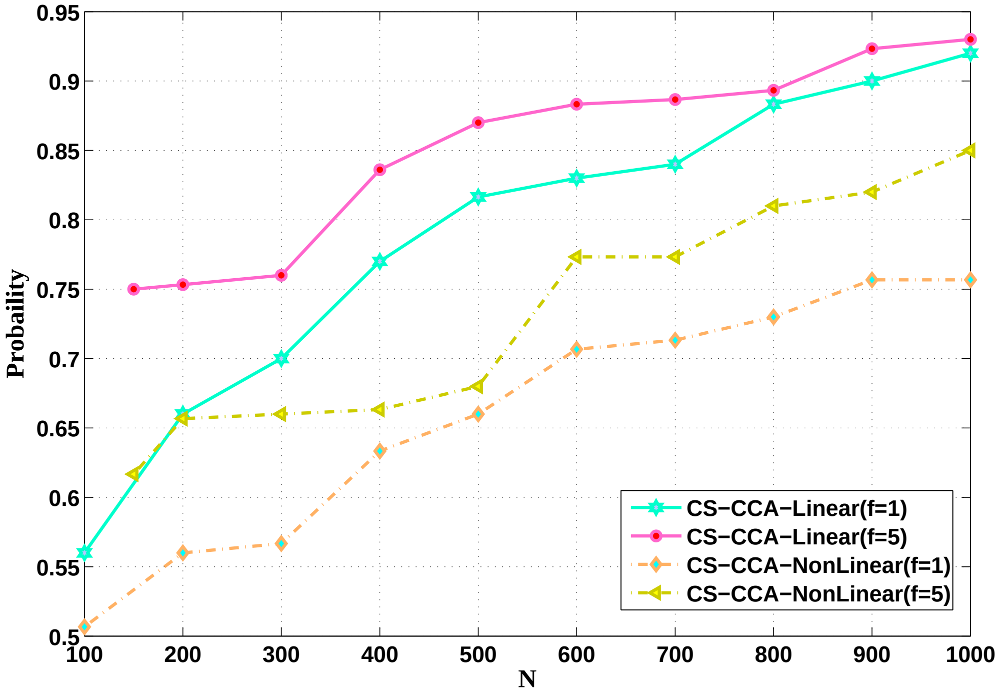

Figure 3.5. Probability of correctly clustering sensors vs. number of data vectors for different memory length f.

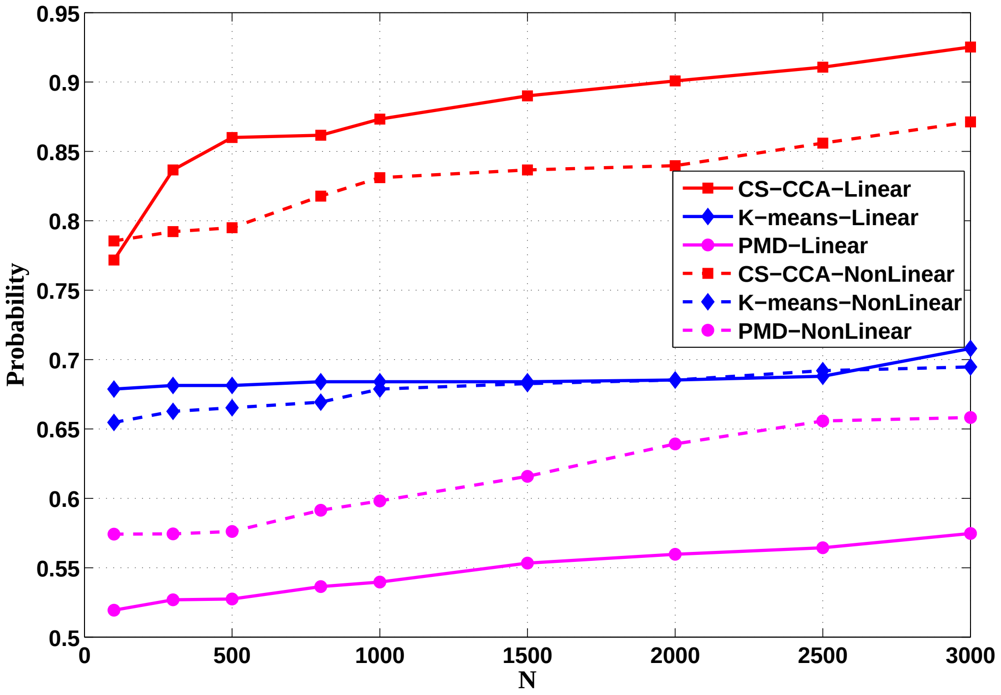

Figure 3.6. Probability of correctly clustering sensors vs. number of data vectors in source-overlapping (non)linear data settings.

CHAPTER 4

CLUSTERING OF HETEROGENEOUS DATA

4.1 Problem Formulation

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 CO_2) levels [41,49]. In the monitored field there are p_1 sensors of the first type, [see red circles 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 mth source whose intensity is denoted by random variable $s_m(t)$, for $m = 1, ..., M$, is located at position $\mathbf{l}_m \in \mathbb{R}^2$, while t denotes the time index and $t = 0, 1, ..., N - 1$. Let $\mathfrak{A}$ and $\mathfrak{B}$ denote the sets of the first and second type of sensors, with cardinality $|\mathfrak{A}| = p_1$ and $|\mathfrak{B}| = p_2$. 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 [8]. The field $f_1(\mathbf{r}, t)$ denotes the strength value of the field measured by sensors in $\mathfrak{A}$, and $f_2(\mathbf{r}, t)$ the strength of the field measured by sensors in $\mathfrak{B}$, at position $\mathbf{r} \in \mathbb{R}^{2\times 1}$ and time t. Diffusion fields are pertinent for modeling how heat or chemical substances are diffusing in space and time [8]. From the theory of Green's function, see e.g., [8], the two diffusion fields are modeled as

$$f_1(\mathbf{r}, t) = (g_1 * S_{d,1})(\mathbf{r}, t) \tag{4.1}$$

$$f_2(\mathbf{r}, t) = (g_2 * S_{d,2})(\mathbf{r}, t) \tag{4.2}$$

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

$$g_1(\mathbf{r}, t) = \frac{1}{4\pi\gamma_1 t} e^{-\frac{\|\mathbf{r}\|_2^2}{4t\gamma_1}} H(t) \tag{4.3}$$

$$g_2(\mathbf{r}, t) = \frac{1}{4\pi\gamma_2 t} e^{-\frac{\|\mathbf{r}\|_2^2}{4t\gamma_2}} H(t) \tag{4.4}$$

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., [8]. Further, $S_{d,1} \in \mathbb{R}^2$ and $S_{d,2} \in \mathbb{R}^2$ denote the sources' distribution at position $\mathbf{r}$ and time t, and are given as

$$S_{d,1}(\mathbf{r}, t) = \textstyle\sum_{m=1}^{M} \chi_{1,m}(s_m(t)) \cdot \delta((\mathbf{r} - \mathbf{L}_{s,m}), t) \tag{4.5}$$

$$S_{d,2}(\mathbf{r}, t) = \textstyle\sum_{m=1}^{M} \chi_{2,m}(s_m(t)) \cdot \delta((\mathbf{r} - \mathbf{L}_{s,m}), t) \tag{4.6}$$

where $\delta(\cdot)$ is the Dirac delta function, and $\chi_{1,m}$ and $\chi_{2,m}$ are random 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 measurements of the sensed fields, i.e.,

$$x_j(t) = f_1(\mathbf{l}_j^{\mathfrak{A}}, t) + w_j(t), \text{ for } j = 1, 2, ..., p_1 \tag{4.7}$$

$$y_i(t) = f_2(\mathbf{l}_i^{\mathfrak{B}}, t) + w_i(t), \text{ for } i = 1, 2, ..., p_2 \tag{4.8}$$

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 $m = 1, ..., M$. Let

$$\mathbf{x}_t := [x_1(t), x_2(t), ..., x_{p_1}(t)]^T \tag{4.9}$$

$$\mathbf{y}_t := [y_1(t), y_2(t), ..., y_{p_2}(t)]^T \tag{4.10}$$

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 $\mathbf{y}_t$ contain information about the same field sources since no location information is available.

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 $\mathbf{y}_t$ contain information about 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 contain information about the mth source for $m = 1, ..., M$, while $\mathcal{S}^0$ is the set of indices in $\mathbf{x}_t$ and $\mathbf{y}_t$ that contain just noise. For example, in Fig. 1 (bottom), there are $p_1 = 30$ sensors of the first type in $\mathfrak{A}$ (red circles), 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 circles 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 red 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

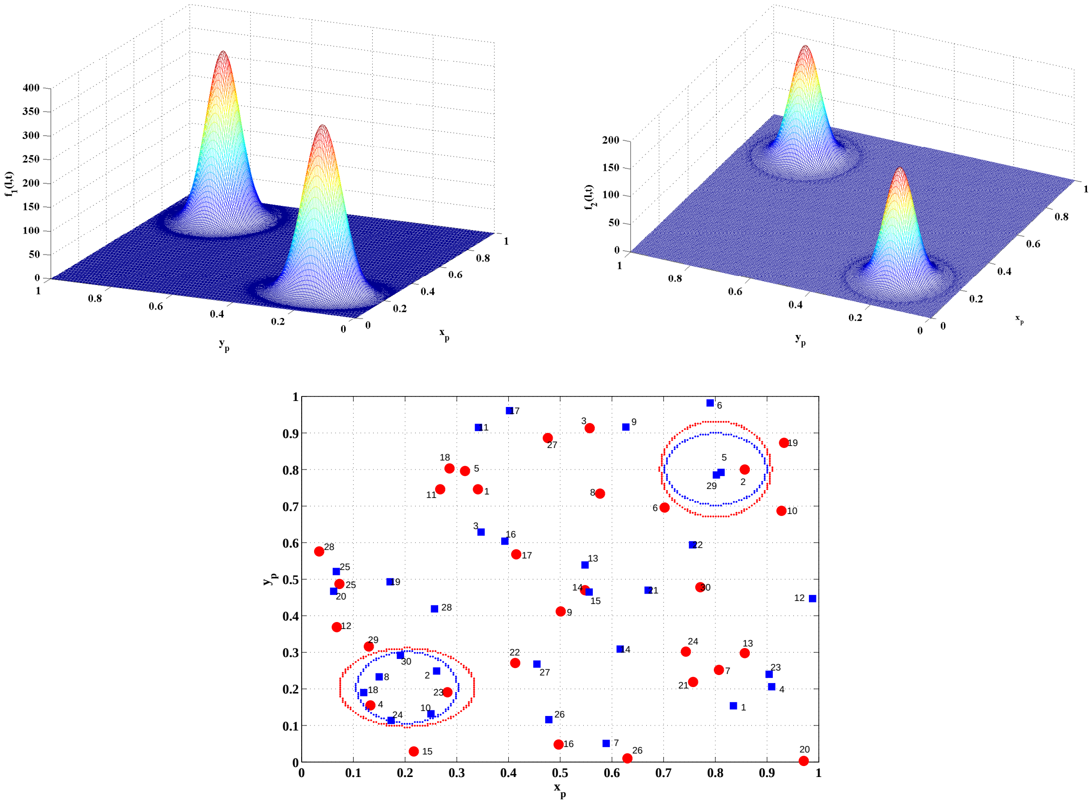

Figure 4.1. Diffusion field examples (top) sensed by a heterogeneous sensor network with two types of sensing units (bottom).

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, $\mathcal{S}_x^1$, $\mathcal{S}_x^2$, $\mathcal{S}_y^1$, $\mathcal{S}_y^2$, 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 contain information about the same source irrespective of the fact that they measure different quantities.

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 $\mathbf{y}_t$ contain information about the same 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 contain information about 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 contain information about 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 contain information about 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 [64, 70, 79, 83, 87], 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.

4.2 Regularized CCA

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 contain information about the same field source, norm-one regularization will be induced in the standard CCA formulation, motived by the work in [64, 70, 79, 87]. 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 the standard CCA, e.g., see [10,83] 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:

$$
\begin{aligned}
(\hat{\mathbf{D}}, \hat{\mathbf{E}}) &= \text{arg min}_{\mathbf{D},\mathbf{E}} N^{-1} \textstyle\sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}_t - \mathbf{D}\mathbf{x}_t - \hat{\boldsymbol{\mu}}\|_2^2 \\
&+ \upsilon \|\mathbf{E}\hat{\boldsymbol{\Sigma}}_y \mathbf{E}^T - \mathbf{I}\|_F^2 + \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x \mathbf{D}^T - \mathbf{I}\|_F^2 \\
&+ \textstyle\sum_{\rho=1}^{q} \lambda_{E,\rho} \|\mathbf{E}_{\rho:}\|_1 + \sum_{\rho=1}^{q} \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 \\
&+ \phi_D \textstyle\sum_{\rho=1}^{q} \|\mathbf{D}_{\rho:}\|_2 + \phi_E \sum_{\rho=1}^{q} \|\mathbf{E}_{\rho:}\|_2
\end{aligned} \tag{4.11}
$$

where the positive scalars $\lambda_{D,\rho}$ and $\lambda_{E,\rho}$ are controlling the sparsity (number of zeroes) in the ρth row of $\mathbf{D}$ and $\mathbf{E}$, respectively. Further, the nonnegative coefficients ϕ_D and ϕ_E multiplying the last norm-two terms in (4.11) control the number of nonzero rows in $\mathbf{D}$ and $\mathbf{E}$. The last two terms in (4.11) introduce group sparsity, e.g., [10, 83], and can zero out 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 (4.11) 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., [5] [1] will be applied. The centralized algorithm will be derived to tackle the problem in (4.11), and perform the desired clustering tasks in the heterogeneous sensor measurements.

4.2.1 Centralized Optimization of RCCA (CR-CCA)

A centralized setting is considered, where a fusion center (FC) collects the measurements acquired across all the sensors. Notice that the cost in (4.11) is a nonconvex function w.r.t. $\mathbf{D}$ and $\mathbf{E}$. Block coordinate descent [4] [71] can surpass this challenge by iteratively solving the cost in (4.11) 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

$$\hat{\mathbf{D}} = \text{arg min}_{\mathbf{D}} N^{-1} \textstyle\sum_{t=0}^{N-1} \|\hat{\mathbf{E}}\mathbf{y}_t - \mathbf{D}\mathbf{x}_t - \hat{\boldsymbol{\mu}}\|_2^2 \tag{4.12}$$
$$+ \varepsilon \|\mathbf{D}\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T - \mathbf{I}\|_F^2 + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho}\|\mathbf{D}_{\rho:}\|_1 + \phi_D \textstyle\sum_{\rho=1}^{q} \|\mathbf{D}_{\rho:}\|_2.$$

In order to solve the problem in (4.12), 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

$$\hat{\mathbf{D}}_{\rho:} = \text{arg min}_{\mathbf{D}_{\rho:}} N^{-1}\|\hat{\mathbf{E}}_{\rho:}\mathbf{Y} - \mathbf{D}_{\rho:}\mathbf{X}\|_2^2 + \lambda_{D,\rho}\|\mathbf{D}_{\rho:}\|_1$$
$$+ \phi_D\|\mathbf{D}_{\rho:}\|_2 + \varepsilon\|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:}\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T\|_2^2 \tag{4.13}$$

in which, $\mathbf{X} := [\mathbf{x}_0 - \hat{\boldsymbol{\mu}}_x, \mathbf{x}_1 - \hat{\boldsymbol{\mu}}_x, ..., \mathbf{x}_{N-1} - \hat{\boldsymbol{\mu}}_x] \in \mathbb{R}^{p_1 \times N}$ and $\mathbf{Y} := [\mathbf{y}_0 - \hat{\boldsymbol{\mu}}_y, \mathbf{y}_1 - \hat{\boldsymbol{\mu}}_y, ..., \mathbf{y}_{N-1} - \hat{\boldsymbol{\mu}}_y] \in \mathbb{R}^{p_2 \times N}$. Let τ denote as block coordinate cycle index. Note that the last term in (4.13) produces fourth-order polynomials in the cost function. To simplify the process of solving (4.13) w.r.t. $\mathbf{D}_{\rho:}$, we fix the second $\mathbf{D}$ in the last term of the cost in (4.13) to the most recent update $\hat{\mathbf{D}}^{\tau-1}$ during the τth 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 (4.13) can be rewritten as

$$\hat{\mathbf{D}}_{\rho:}^{\tau} = \text{arg min}_{\mathbf{D}_{\rho:}} N^{-1}\|\hat{\mathbf{E}}_{\rho:}^{\tau-1}\mathbf{Y} - \mathbf{D}_{\rho:}\mathbf{X}\|_2^2 + \lambda_{D,\rho}\|\mathbf{D}_{\rho:}\|_1$$
$$+ \phi_D\|\mathbf{D}_{\rho:}\|_2 + \varepsilon\|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:}\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 \tag{4.14}$$

To tackle the minimization problem in (4.14) we will employ ADMM. To this end, we introduce an auxiliary vector $\mathbf{b}_\rho \in \mathbb{R}^{1 \times p_1}$ and reformulate (4.14) as the following equivalent constrained minimization problem

$$(\hat{\mathbf{D}}_{\rho:}^{\tau}, \hat{\mathbf{b}}_{\rho}^{\tau}) = \text{arg min}_{\mathbf{D}_{\rho:}, \mathbf{b}_\rho} N^{-1}\|\hat{\mathbf{E}}_{\rho:}^{\tau-1}\mathbf{Y} - \mathbf{D}_{\rho:}\mathbf{X}\|_2^2 + \lambda_{D,\rho}$$
$$\|\mathbf{D}_{\rho:}\|_1 + \phi_D\|\mathbf{b}_\rho\|_2 + \varepsilon\|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:}\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T\|_2^2,$$
$$subject\ \ to\ \ \mathbf{b}_\rho = \mathbf{D}_{\rho:} \tag{4.15}$$

Notice that (4.15) amd (4.14) are equivalent in the sense that they have the same optimal solution. A necessary step in applying ADMM in (4.15) is the formation of augmented Lagrangian function

$$\mathcal{L}^{\tau}(\mathbf{D}_{\rho:}, \mathbf{b}_{\rho}, \mathbf{p}_{\rho}) = N^{-1}\|\hat{\mathbf{E}}_{\rho:}^{\tau-1}\mathbf{Y} - \mathbf{D}_{\rho:}\mathbf{X}\|_2^2 + \lambda_{D,\rho}\|\mathbf{D}_{\rho:}\|_1 + \phi_D\|\mathbf{b}_{\rho}\|_2$$
$$+ \varepsilon\|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:}\hat{\mathbf{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 + (\mathbf{D}_{\rho:} - \mathbf{b}_{\rho})\mathbf{p}_{\rho} + \frac{c}{2}\|\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}\|_2^2 \tag{4.16}$$

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 (4.16) is strictly convex w.r.t. $\mathbf{D}_{\rho:}$ and $\mathbf{b}_{\rho}$. ADMM is an iterative method, see e.g., [5], which involves the following three steps (details follow later):

Step 1) Minimize the augmented Lagrangian in (4.16) 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 (4.16) w.r.t. $\mathbf{b}_{\rho}$, while fixing $\mathbf{D}_{\rho:}$, and $\mathbf{p}_{\rho}$ to the ADMM updates $\hat{\mathbf{D}}_{\rho:}^{\tau;\kappa}$ and $\mathbf{p}_{\rho}^{\tau,\kappa-1}$ obtained at steps 1) and 2), to obtain update $\mathbf{b}_{\rho}^{\tau,\kappa}$.

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 coordinate cycle τ, and the κth ADMM iteration. These three steps are applied for K ADMM iterations, i.e., $\kappa = 1, ..., K$, nested inside cycle τ.

The first step in ADMM, during coordinate descent cycle τ, involves minimization of (4.16) 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

$$\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa} = \text{arg min}_{\mathbf{D}_{\rho:}} N^{-1}\|\hat{\mathbf{E}}_{\rho:}^{\tau-1}\mathbf{Y} - \mathbf{D}_{\rho:}\mathbf{X}\|_2^2 + \lambda_{D,\rho}\|\mathbf{D}_{\rho:}\|_1$$
$$+ \varepsilon\|\mathbf{I}_{\rho:} - \mathbf{D}_{\rho:}\hat{\mathbf{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T\|_2^2 + \frac{c}{2}\|\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}^{\tau,\kappa-1}\|_2^2 + \mathbf{D}_{\rho:}\mathbf{p}_{\rho}^{\tau,\kappa-1}. \tag{4.17}$$

The second step in ADMM involves minimizing (4.16) 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

$$\mathbf{b}_\rho^{\tau,\kappa} = \text{arg min}_{\mathbf{b}_\rho} \phi_D \|\mathbf{b}_\rho\|_2 + \frac{c}{2}\|\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa} - \mathbf{b}_\rho\|_2^2 - \mathbf{b}_\rho \mathbf{p}_\rho^{\tau,\kappa-1} \tag{4.18}$$

The third step involves updating the Lagrange multiplier vector

$$\mathbf{p}_\rho^{\tau,\kappa} = \mathbf{p}_\rho^{\tau,\kappa-1} + c(\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa} - \mathbf{b}_\rho^{\tau,\kappa})^T. \tag{4.19}$$

Using the convergence claims for ADMM in [5], it can be readily shown that the iterates $\hat{\mathbf{D}}_{\rho:}^{\tau,k}$ converge to $\lim_{k\to\infty}\hat{\mathbf{D}}_{\rho:}^{\tau,k} = \hat{\mathbf{D}}_{\rho:}^{\tau}$, where $\hat{\mathbf{D}}_{\rho:}^{\tau}$ for $\rho = 1, ..., q$ is the minimizer of (4.14) as the number of ADMM iterations goes to ∞.

To tackle the minimization problem in (4.17), 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 (4.17) each giving the update

$$\hat{\mathbf{D}}^{\tau,\kappa}(\rho,\beta) = \text{arg min}_d \|\boldsymbol{\zeta}_{\rho,\beta}^\tau - d \cdot \mathbf{h}_{\rho,\beta}^\tau\|_2^2 + \lambda_{D,\rho}|d| + 0.5c(d - \mathbf{b}_\rho^{\tau,\kappa-1}(\beta))^2 + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(\beta) \tag{4.20}$$

for $\beta = 1, 2, ..., p_1$, where

$$\boldsymbol{\zeta}_{\rho,\beta}^\tau := [\boldsymbol{\zeta}_{\rho,\beta}^{1,\tau}, \boldsymbol{\zeta}_{\rho,\beta}^{2,\tau}]^T$$

and

$$\mathbf{h}_{\rho,\beta}^\tau := [N^{-0.5}\mathbf{X}_{\beta:}, (\hat{\boldsymbol{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T)_{\beta:}]^T,$$

in which

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

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

As it has been shown in [64, Apdx. A], the closed form solution to (4.20) can be expressed as

$$\hat{\mathbf{D}}^{\tau,\kappa}(\rho,\beta) = \mathbb{F}(\boldsymbol{\zeta}^{\tau}_{\rho,\beta}, \mathbf{h}^{\tau}_{\rho,\beta}, \frac{c}{2}(\mathbf{b}^{\tau,\kappa-1}_{\rho}(\beta) - \frac{\mathbf{p}^{\tau,\kappa-1}_{\rho}(\beta)}{c})\;,\; \frac{c}{2}\;,\; \lambda_{D,\rho}\;) \tag{4.21}$$

$$\begin{aligned} &\text{where } \mathbb{F}(\mathbf{p}_1, \mathbf{p}_2, p_3, p_4, \lambda) := \text{sgn}(\mathbf{p}_1^T\mathbf{p}_2 + p_3) \\ &\times (\max\left(0, \left(\left|\frac{\mathbf{p}_1^T\mathbf{p}_2 + p_3}{\|\mathbf{p}_2\|_2^2 + p_4}\right| - \left(\frac{\lambda}{2(\|\mathbf{p}_2\|_2^2 + p_4)}\right)\right)\right). \end{aligned} \tag{4.22}$$

It is established in Apdx. C that the minimizer of (4.18) results the following update

$$\mathbf{b}^{\tau,\kappa}_{\rho} = c^{-1}\mathcal{S}_v(\mathbf{p}^{\tau,\kappa-1}_{\rho} + c \cdot \hat{\mathbf{D}}^{\tau,\kappa}_{\rho}, \phi_D) \tag{4.23}$$

where $\mathcal{S}_v(\mathbf{v}, \phi) = [1 - \frac{\phi}{\|\mathbf{v}\|_2}]_+\mathbf{v}$.

Next, starting from (4.11), 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 τth coordinate cycle, the estimate $\mathbf{E}^{\tau}_{\rho:}$ is obtained as

$$\begin{aligned} \hat{\mathbf{E}}^{\tau}_{\rho:} &= \text{arg min}_{\mathbf{E}_{\rho:}} N^{-1}\|\hat{\mathbf{D}}^{\tau-1}_{\rho:}\mathbf{X} - \mathbf{E}_{\rho:}\mathbf{Y}\|_2^2 + \lambda_{E,\rho}\|\mathbf{E}_{\rho:}\|_1 \\ &+ \phi_E\|\mathbf{E}^T_{\rho:}\|_2 + v\|\mathbf{I}_{\rho:} - \mathbf{E}_{\rho:}\hat{\boldsymbol{\Sigma}}_y(\hat{\mathbf{E}}^{\tau-1})^T\|_2^2. \end{aligned} \tag{4.24}$$

In order to solve the problem in (4.24), we follow a similar process as the one used to update $\hat{\mathbf{D}}^{\tau}_{\rho:}$ in (4.14). Then, we can obtain the updating recursions for $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j)$, $\breve{\mathbf{b}}^{\tau,\kappa}_{\rho} \in \mathbb{R}^{1\times p_2}$, and $\breve{\mathbf{p}}^{\tau,\kappa}_{\rho} \in \mathbb{R}^{p_2\times 1}$, in which the later two quantities are similar to $\mathbf{b}^{\tau,\kappa}_{\rho}$, and $\mathbf{p}^{\tau,\kappa}_{\rho}$, respectively. The update $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j)$ can be obtained as

$$\hat{\mathbf{E}}^{\tau,\kappa}(\rho, j) = \mathbb{F}(\;[\breve{\boldsymbol{\zeta}}^{1,\tau}_{\rho,j}, \breve{\boldsymbol{\zeta}}^{2,\tau}_{\rho,j}]^T\;, \breve{\mathbf{h}}^{\tau}_{\rho}\;, \frac{c}{2}(\breve{\mathbf{b}}^{\tau,\kappa-1}_{\rho}(j) - \frac{\breve{\mathbf{p}}^{\tau,\kappa-1}_{\rho}(j)}{c})\;,\; \frac{c}{2}\;,\; \lambda_{E,\rho}\;) \tag{4.25}$$

for $\rho = 1, ..., q$, and $j = 1, ..., p_2$, while $\breve{\boldsymbol{\zeta}}^{1,\tau}_{\rho,j}$, $\breve{\boldsymbol{\zeta}}^{2,\tau}_{\rho,j}$, and $\breve{\mathbf{h}}^{\tau}_{\rho}$ are obtained similarly to $\boldsymbol{\zeta}^{1,\tau}_{\rho,\beta}$, $\boldsymbol{\zeta}^{2,\tau}_{\rho,\beta}$, and $\mathbf{h}^{\tau}_{\rho}$, respectively, after substituting $\mathbf{X}$, $\mathbf{Y}$, $\hat{\boldsymbol{\Sigma}}_x$, $\hat{\mathbf{D}}^{\tau}$, $\hat{\mathbf{D}}^{\tau-1}$, $\hat{\mathbf{E}}^{\tau-1}$, β, p_1, and ε by $\mathbf{Y}$,

$\mathbf{X}$, $\hat{\boldsymbol{\Sigma}}_y$, $\hat{\mathbf{E}}^{\tau}$, $\hat{\mathbf{E}}^{\tau-1}$, $\hat{\mathbf{D}}^{\tau-1}$, j, p_2, and υ, respectively. Then, the auxiliary vector $\breve{\mathbf{b}}_{\rho}^{\tau,\kappa}$ can be obtained by

$$\breve{\mathbf{b}}_{\rho}^{\tau,\kappa} = c^{-1}\mathcal{S}_{\upsilon}(\breve{\mathbf{p}}_{\rho}^{\tau,\kappa-1} + c \cdot \hat{\mathbf{E}}_{\rho:}^{\tau,\kappa}, \phi_E). \tag{4.26}$$

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

$$\breve{\mathbf{p}}_{\rho}^{\tau,\kappa} = \breve{\mathbf{p}}_{\rho}^{\tau,\kappa-1} + c(\hat{\mathbf{E}}_{\rho:}^{\tau,\kappa} - \breve{\mathbf{b}}_{\rho}^{\tau,\kappa})^{T} \tag{4.27}$$

The entire updating process for implementing the centralized regularized CCA (CR-CCA) algorithm is tabulated as Algorithm 3.

Algorithm 3 : CR-CCA

```
Initialize $\mathbf{D}^{(0)}$ and $\mathbf{E}^{(0)}$ randomly. Initialize $\mathbf{b}_{\rho}^{0}$, $\breve{\mathbf{b}}_{\rho}^{0}$, $\mathbf{p}_{\rho}^{0}$, $\breve{\mathbf{p}}_{\rho}^{0}$ to $\mathbf{0}$ for $\rho = 1, 2, ..., q$.
for $\tau = 1, 2, \ldots,$ do
for $\rho = 1, 2, \ldots, q$ do
  for $\kappa = 1, 2, \ldots, K$ do
    Update $\hat{\mathbf{D}}^{\tau,\kappa}(\rho,\beta)$, $\mathbf{b}_{\rho}^{\tau,\kappa}$, and $\mathbf{p}_{\rho}^{\tau,\kappa}$ via (4.21), (4.23), and (4.19) for $\beta = 1, \ldots, p_1$.
    Update $\hat{\mathbf{E}}^{\tau,\kappa}(\rho,j)$, $\breve{\mathbf{b}}_{\rho}^{\tau,\kappa}$, and $\breve{\mathbf{p}}_{\rho}^{\tau,\kappa}$ via (4.25), (4.26), and (4.27) for $j = 1, \ldots, p_2$.
  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 $\rho = 1, \ldots, q$.
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.
end for
```

Notice that the updates in (4.21) and (4.25) for $\beta = 1, \ldots, p_1$ and $j = 1, \ldots, p_2$ 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. 3 it was assumed that $\tau = 1$, which implies that the quantities in (4.21) and (4.25) were updated for one cycle, and thus the τ_2 superscript was not included. However, notice that to solve (4.17) (as well as the corresponding minimization task when updating $\hat{\mathbf{E}}_{\rho:}^{\tau,k}$) the updates (4.21) [and correspondingly (4.25)] should be applied for an increasing number of nested coordinate iterations, i.e., $\tau_2 \to \infty$.

4.2.2 Distributed Implementation of RCCA (DR-CCA)

The proposed regularized (R-)CCA framework in (4.11) 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 $\mathfrak{A} \bigcup \mathfrak{B}$, 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 $\mathfrak{A} \bigcup \mathfrak{B}$ 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 $j \in \mathfrak{A} \bigcup \mathfrak{B}$, 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 jth 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 ith 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 $\mathfrak{A} \bigcup \mathfrak{B}$ is connected. The information acquired across sensors is spatially scattered, i.e., sensor $j \in \mathfrak{A}$ has available only measurements $\mathbf{x}_t(j)$ (jth entry in $\mathbf{x}_t$), while sensor $i \in \mathfrak{B}$ has available only the scalar measurement $\mathbf{y}_t(i)$ (ith entry in $\mathbf{y}_t$). Further, let $\bar{\mathbf{x}}_t(j)$ and $\bar{\mathbf{y}}_t(i)$ denote the jth and ith 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. 4.2 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.,

$$\mathbf{D}(\mathbf{x}_t - \hat{\boldsymbol{\mu}}_x) = p_1[\frac{1}{p_1} \textstyle\sum_{j=1}^{p_1} \mathbf{d}_j \bar{\mathbf{x}}_t(j)] \text{ and} \tag{4.28}$$

$$\mathbf{E}(\mathbf{y}_t - \hat{\boldsymbol{\mu}}_y) = p_2[\frac{1}{p_2} \textstyle\sum_{i=1}^{p_2} \mathbf{e}_i \bar{\mathbf{y}}_t(i)] \tag{4.29}$$

Substituting the previous quantities in the cost in (4.17), we obtain the following minimization problem

$$\begin{aligned}
&\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa} = \text{arg min}_{\,\mathbf{D}_{\rho:}} N^{-1} \textstyle\sum_{t=0}^{N-1} \| \sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i) \\
&- \textstyle\sum_{j=1}^{p_1} \mathbf{D}(\rho, j) \bar{\mathbf{x}}_t(j) \|_2^2 + \lambda_{D,\rho} \|\mathbf{D}_{\rho:}\|_1 + \mathbf{D}_{\rho:} \mathbf{p}_{\rho}^{\tau,\kappa-1} \\
&+ \varepsilon \|\mathbf{I}_{\rho:} - \frac{1}{N} \sum_{t=0}^{N-1} [(\sum_{j=1}^{p_1} \mathbf{D}(\rho, j) \bar{\mathbf{x}}_t(j))(\sum_{j=1}^{p_1} \hat{\mathbf{d}}_j^{\tau-1} \bar{\mathbf{x}}_t(j))^T] \|_2^2 \\
&+ \frac{c}{2} \|\mathbf{D}_{\rho:} - \mathbf{b}_{\rho}^{\tau,\kappa-1}\|_2^2.
\end{aligned} \tag{4.30}$$

Specifically, ADMM will be employed to allow sensors estimate in a distributed fashion the global quantities

$$\sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i) \bar{\mathbf{y}}_t(i),\ \sum_{j=1}^{p_1} \hat{\mathbf{D}}^{\tau-1}(\rho, j) \bar{\mathbf{x}}_t(j) \tag{4.31}$$

which correspond to the latest updates for the quantities at (4.28) and (26), at the beginning of cycle τ. 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 jth entry in vectors $\mathbf{b}_\rho$ and $\mathbf{p}_\rho$, namely $\mathbf{b}_\rho(j)$ and $\mathbf{p}_\rho(j)$, respectively, for $\rho = 1, \ldots, q$. Toward this end, the minimization problem in (4.30) 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 corresponds to the optimization variable)

$$
\begin{aligned}
&\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) = \text{arg min}_d N^{-1} \sum_{t=0}^{N-1} \| \sum_{i=1}^{p_2} \hat{\mathbf{E}}^{\tau-1}(\rho, i)\bar{\mathbf{y}}_t(i) \\
&- \textstyle\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i)\bar{\mathbf{x}}_t(i) + \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{x}}_t(j) - d \cdot \bar{\mathbf{x}}_t(j)\|_2^2 \\
&+ \lambda_{D,\rho}|d| + d\mathbf{p}_\rho^{\tau,\kappa-1}(j) + \varepsilon\|\mathbf{I}_{\rho:} - N^{-1} \sum_{t=0}^{N-1} [(\sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i)\bar{\mathbf{x}}_t(i) \\
&- \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{x}}_t(j) + d \cdot \mathbf{x}(t, j))(\sum_{i=1}^{p_1} \hat{\mathbf{d}}_i^{\tau-1}\bar{\mathbf{x}}_t(i))^T\|_2^2 \\
&+ \frac{c}{2}[d - \mathbf{b}_\rho^{\tau,\kappa-1}(j)]^2, \text{ for } j = 1, 2, ..., p_1,
\end{aligned} \tag{4.32}
$$

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). The necessity of estimating the quantities in (4.31) to allow sensor j update $\hat{\mathbf{D}}^{\tau,k}(\rho, j)$ is apparent from (4.32). 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 in (4.31) 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 (4.32) and enable sensor j to update $\hat{\mathbf{D}}^{\tau,k}(\rho, j)$.

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 $t = 0, ..., N-1$, by solving the separable constrained minimization problem:

$$
\begin{aligned}
&\text{min}_{\eta_{i,t,\rho}^{\tau,\kappa-1}} \sum_{i=1}^{p_1} \|\eta_{i,t,\rho}^{\tau,\kappa-1} - p_1\hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i)\bar{\mathbf{x}}_t(i)\|_2^2 \qquad (4.33) \\
&\text{s. to } \eta_{i,t,\rho}^{\tau,\kappa-1} = \eta_{i',t,\rho}^{\tau,\kappa-1}, i' \in \mathcal{N}_i^{\mathfrak{A}} \text{ for } t = 0, 1, ..., N-1
\end{aligned}
$$

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 constrains $\eta_{i,t,\rho}^{\tau-1} = \eta_{i',t,\rho}^{\tau,\kappa-1}, i' \in \mathcal{N}_i^{\mathfrak{A}}$ for $i = 1, ..., p_1$ 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, the minimization task in (4.33) can be tackled by sensor j through updating the sensor j's local estimate $\eta_{j,t,\rho}$, along with the Lagrange multipliers $\{v_{i,t,\rho}^{i',\tau,\kappa-1}\}_{i' \in \mathcal{N}_i^{\mathfrak{A}}}$, 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

$$v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota) = v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota - 1) + 0.5c_2[\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota) - \eta_{j',t,\rho}^{\tau,\kappa-1}(\iota)] \tag{4.34}$$

$$\begin{aligned}\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota + 1) &= [(2 + 2c_2|\mathcal{N}_j^{\mathfrak{A}}|)\mathbf{I}_q]^{-1} \times [2p_1\hat{\mathbf{D}}^{\tau-1}(\rho, j)\bar{\mathbf{x}}_t(j)\\ &- \sum\nolimits_{j' \in \mathcal{N}_j^{\mathfrak{A}}}((v_{j,t,\rho}^{j',\tau,\kappa-1}(\iota) - v_{j',t,\rho}^{j,\tau,\kappa-1}(\iota)) + c_2(\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota)\\ &+ \eta_{j',t,\rho}^{\tau,\kappa-1}(\iota)))]\end{aligned} \tag{4.35}$$

in which, ι is the network ADMM iteration index, and c_2 represents a positive step-size. The convergence results in [61] reveal that if $\iota \to \infty$, $\lim_{\iota\to\infty}\eta_{j,t,\rho}^{\tau,\kappa-1}(\iota) = \sum_{i=1}^{p_1} \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, i)$ $\bar{\mathbf{x}}_t(i)$, 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 $\hat{\boldsymbol{\eta}}_{j,t}^{\tau-1} := [\hat{\eta}_{j,t,1}^{\tau-1,K}, ..., \hat{\eta}_{j,t,q}^{\tau-1,K}]^T \in \mathbb{R}^{q\times 1}$ denote the corresponding estimates, which can be obtained in the beginning of τth coordinate cycle at sensor j.

A similar procedure (4.34)-(35) 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 (4.32) can be replaced with

$$\begin{aligned}
&\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) = \text{arg min}_d 0.5c[d - \mathbf{b}_\rho^{\tau,\kappa-1}(j)]^2 + \lambda_{D,\rho}|d| \\
&+ N^{-1}\sum_{t=0}^{N-1} \|\hat{\nu}_{f,t,\rho}^{\tau-1} - \hat{\eta}_{j,t,\rho}^{\tau,\kappa-1} - d \cdot \bar{\mathbf{x}}_t(j) + \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{x}}_t(j)\|_2^2 \\
&+ \varepsilon \|\mathbf{I}_{\rho:} - \frac{1}{N}\sum_{t=0}^{N-1}[(\hat{\eta}_{j,t,\rho}^{\tau,\kappa-1} - \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{x}}_t(j) + d \cdot \bar{\mathbf{x}}_t(j)) \\
&\times (\hat{\boldsymbol{\eta}}_{j,t}^{\tau-1})^T]\|_2^2 + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(j) \text{ for } j = 1, 2, ..., p_1.
\end{aligned} \tag{4.36}$$

For notational simplicity let

$$\hat{\boldsymbol{\nu}}_{f,\rho}^{\tau-1} := [\hat{\nu}_{f,0,\rho}^{\tau-1}, \hat{\nu}_{f,1,\rho}^{\tau-1}...\hat{\nu}_{f,N-1,\rho}^{\tau-1}] \in \mathbb{R}^{1\times N}, \tag{4.37}$$

$$\hat{\boldsymbol{\eta}}_j^{\tau-1} := [\hat{\boldsymbol{\eta}}_{j,0}^{\tau-1}, \hat{\boldsymbol{\eta}}_{j,1}^{\tau-1}...\hat{\boldsymbol{\eta}}_{j,N-1}^{\tau-1}] \in \mathbb{R}^{q\times N}, \tag{4.38}$$

$$\hat{\boldsymbol{\eta}}_{j,\rho}^{\tau,\kappa-1} := [\hat{\eta}_{j,0,\rho}^{\tau,\kappa-1}, \hat{\eta}_{j,1,\rho}^{\tau,\kappa-1}...\hat{\eta}_{j,N-1,\rho}^{\tau,\kappa-1}] \in \mathbb{R}^{1\times N} \tag{4.39}$$

$$\bar{\mathbf{X}}_j := [\bar{\mathbf{x}}_0(j), \bar{\mathbf{x}}_1(j)...\bar{\mathbf{x}}_{N-1}(j)] \in \mathbb{R}^{1\times N} \tag{4.40}$$

Moreover, the following notation is used

$$\begin{aligned}
&\mathbf{M}_{j,\rho}^{\tau,\kappa} := \hat{\boldsymbol{\nu}}_{f,\rho}^{\tau-1} - \hat{\boldsymbol{\eta}}_{j,\rho}^{\tau,\kappa-1} + \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{X}}_j, \\
&\mathbf{P}_{j,\rho}^{\tau,\kappa} := \varepsilon^{0.5}\mathbf{I}_{\rho:} - \varepsilon^{0.5}N^{-1}(\hat{\boldsymbol{\eta}}_{j,\rho}^{\tau,\kappa-1} - \hat{\mathbf{D}}^{\tau,\kappa-1}(\rho, j)\bar{\mathbf{X}}_j)(\hat{\boldsymbol{\eta}}_j^{\tau-1})^T, \\
&\mathbf{Q}_j^{\tau,\kappa} := \varepsilon^{0.5}N^{-1}\bar{\mathbf{X}}_j(\hat{\boldsymbol{\eta}}_j^{\tau-1})^T
\end{aligned} \tag{4.41}$$

then the cost in (4.36) can be rewritten as

$$\begin{aligned}
&\hat{\mathbf{D}}^{\tau,\kappa}(\rho, j) = \text{arg min}_d N^{-1}\|\mathbf{M}_{j,\rho}^{\tau,\kappa} - d \cdot \bar{\mathbf{X}}_j\|_2^2 \qquad (4.42) \\
&+ \lambda_{D,\rho}|d| + d \cdot \mathbf{p}_\rho^{\tau,\kappa-1}(j) + \|\mathbf{P}_{j,\rho}^{\tau,\kappa} - d \cdot \mathbf{Q}_j^{\tau,\kappa}\|_2^2 \\
&+ \frac{c}{2}(d - \mathbf{b}_\rho^{\tau,\kappa-1}(j))^2 \text{ for } j = 1, 2, ..., p_1.
\end{aligned}$$

Using a similar method to solve (4.14) in Centralized R-CCA framework, the solution to (4.42) can be expressed as

$$\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j) = \mathbb{F}(\mathbf{\Gamma}^{\tau,\kappa}_{\rho,j}, \mathbf{\Lambda}^{\tau,\kappa}_{\rho,j}, \frac{c}{2}(\mathbf{b}^{\tau,\kappa-1}_{\rho}(j) - \frac{\mathbf{p}^{\tau,\kappa-1}_{\rho}(j)}{c}), \frac{c}{2}) \tag{4.43}$$

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

After finishing the first ADMM step for (4.17), our focus is shifted to implementing the second step in (4.18) in a distributed fashion. Similar to the optimal solution for $\mathbf{b}^{\tau,\kappa-1}_{\rho}$ given by (4.23) in the Centralized R-CCA scheme, sensor $j \in \mathfrak{A}$ is responsible for updating the jth entry of $\mathbf{b}^{\tau,\kappa}_{\rho}$ in (4.23) which is given by

$$\frac{1}{c}[1 - \frac{\phi_D}{\|\mathbf{p}^{\tau,\kappa-1}_{\rho} + c(\hat{\mathbf{D}}^{\tau,\kappa}_{\rho:})^T\|_2}]_{+}(\mathbf{p}^{\tau,\kappa-1}_{\rho}(j) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j)) \tag{4.44}$$

where $\mathbf{p}^{\tau,\kappa-1}_{\rho}(j)$ and $\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j)$ are available for sensor $j \in \mathfrak{A}$. The challenge in calculating (4.44) is finding the global value $\|\mathbf{p}^{\tau,\kappa-1}_{\rho} + c(\hat{\mathbf{D}}^{\tau,\kappa}_{\rho:})^T\|_2$ in a distributed fashion. Let $m^{\tau,\kappa}_{\rho} := \|\mathbf{p}^{\tau,\kappa-1}_{\rho} + c(\hat{\mathbf{D}}^{\tau,\kappa}_{\rho:})^T\|_2^2$, then

$$m^{\tau,\kappa}_{\rho} = \textstyle\sum_{\ell=1}^{p_1}(\mathbf{p}^{\tau,\kappa-1}_{\rho}(\ell) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho,\ell))^2 \tag{4.45}$$

Notice that, $m^{\tau,\kappa}_{\rho}$ is an average-like quantity, and ADMM can be employed as in (35)-(36) to estimate $m^{\tau,\kappa}_{\rho}$ across sensors. Let $\hat{u}^{\tau,\kappa}_{j,\rho}$ denote sensor j's local estimate of the global quantity $m^{\tau,\kappa}_{\rho}$ after taking K_2 network ADMM iterations, say $u^{\tau,\kappa}_{j,\rho}(K_2)$. Thus, $\mathbf{b}^{\tau,\kappa}_{\rho}(j)$ can be obtained as

$$\mathbf{b}^{\tau,\kappa}_{\rho}(j) = \frac{1}{c}[1 - \frac{\phi_{D,\rho}}{\hat{u}^{\tau,\kappa}_{j,\rho}}]_{+}(\mathbf{p}^{\tau,\kappa-1}_{\rho}(j) + c\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j)). \tag{4.46}$$

Finally, the last task is to complete the third ADMM step in (4.19) in a distributed way. Sensor $j \in \mathfrak{A}$ needs to update $\mathbf{p}^{\tau,\kappa}_{\rho}(j)$ based on the most recent updated $\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j)$ and $\mathbf{b}^{\tau,\kappa}_{\rho}(j)$, which can be directly obtained by

$$\mathbf{p}^{\tau,\kappa}_{\rho}(j) = \mathbf{p}^{\tau,\kappa-1}_{\rho}(j) + c(\hat{\mathbf{D}}^{\tau,\kappa}(\rho,j) - \mathbf{b}^{\tau,\kappa}_{\rho}(j)). \tag{4.47}$$

Starting form (4.11), 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 $\breve{\mathbf{b}}_{\rho}^{\tau,\kappa}(i)$ and Lagrangian multiplier $\breve{\mathbf{p}}_{\rho}^{\tau,\kappa}(i)$. Specifically, let's denote $\hat{\nu}_{i,t,\rho}^{\tau,\kappa-1}$ as the local estimate of $\sum_{\ell=1}^{p_2} \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, \ell)\bar{\mathbf{y}}_t(\ell)$ after applying K_2 ADMM iterations across sensors in $\mathfrak{B}$. Define $\hat{\boldsymbol{\nu}}_{i,t}^{\tau-1} := [\hat{\nu}_{i,t,1}^{\tau-1,K}, ..., \hat{\nu}_{i,t,q}^{\tau-1,K}]^T \in \mathbb{R}^{q\times 1}$ and let

$$\hat{\boldsymbol{\nu}}_{i,\rho}^{\tau,\kappa-1} := [\hat{\nu}_{i,0,\rho}^{\tau,\kappa-1}, ..., \hat{\nu}_{i,N-1,\rho}^{\tau,\kappa-1}]^T \in \mathbb{R}^{q\times N} \tag{4.48}$$

$$\hat{\boldsymbol{\eta}}_{h,\rho}^{\tau-1} := [\hat{\boldsymbol{\eta}}_{h}^{\tau-1}]_{\rho:} \in \mathbb{R}^{1\times N} \tag{4.49}$$

$$\bar{\mathbf{Y}}_i := [\bar{\mathbf{y}}_0(i), ..., \bar{\mathbf{y}}_{N-1}(i)]^T \in \mathbb{R}^{q\times N} \tag{4.50}$$

where $h \in \mathfrak{A}$ corresponds to the neighboring sensor of $i \in \mathfrak{B}$. Using similar notation as in (4.41) let us define for sensor $i \in \mathfrak{B}$

$$\begin{aligned}
&\breve{\mathbf{M}}_{i,\rho}^{\tau,\kappa} := \hat{\boldsymbol{\eta}}_{h,\rho}^{\tau-1} - \hat{\boldsymbol{\nu}}_{i,\rho}^{\tau,\kappa-1} + \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, i)\bar{\mathbf{Y}}_i, \\
&\breve{\mathbf{P}}_{i,\rho}^{\tau,\kappa} := \upsilon^{0.5}\mathbf{I}_{\rho:} - \upsilon^{0.5}N^{-1}(\hat{\boldsymbol{\nu}}_{i,\rho}^{\tau,\kappa-1} - \hat{\mathbf{E}}^{\tau,\kappa-1}(\rho, i)\bar{\mathbf{Y}}_i)(\hat{\boldsymbol{\nu}}_{i}^{\tau-1})^T, \\
&\breve{\mathbf{Q}}_{i}^{\tau,\kappa} := \upsilon^{0.5}N^{-1}\bar{\mathbf{Y}}_j(\hat{\boldsymbol{\nu}}_{i}^{\tau-1})^T.
\end{aligned} \tag{4.51}$$

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

$$\begin{aligned}
&\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i) = \mathbb{F}([N^{-0.5}\breve{\mathbf{M}}_{i,\rho}^{\tau,\kappa}, \breve{\mathbf{P}}_{i,\rho}^{\tau,\kappa}]^T, [N^{-0.5}\bar{\mathbf{Y}}_i, \breve{\mathbf{Q}}_{i}^{\tau,\kappa}]^T, \\
&\frac{c}{2}(\breve{\mathbf{b}}_{\rho}^{\tau,\kappa-1}(i) - \frac{\breve{\mathbf{p}}_{\rho}^{\tau,\kappa-1}(i)}{c}), \frac{c}{2})
\end{aligned} \tag{4.52}$$

$$\breve{\mathbf{b}}_{\rho}^{\tau,\kappa}(i) = \frac{1}{c}(1 - \frac{\phi_E}{\hat{\breve{u}}_{i}^{\tau,\kappa}})_{+}(\breve{\mathbf{p}}_{\rho}^{\tau,\kappa-1}(i) + c\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)) \tag{4.53}$$

$$\breve{\mathbf{p}}_{\rho}^{\tau,\kappa}(i) = \mathbf{p}_{\rho}^{\tau,\kappa-1}(j) + c(\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i) - \breve{\mathbf{b}}_{\rho}^{\tau,\kappa}(i)) \tag{4.54}$$

where $\hat{\breve{u}}_{i}^{\tau,\kappa}$ denotes the local estimate at sensor $i \in \mathfrak{B}$ local estimate of $\|\mathbf{p}_{\rho}^{\prime,\tau,\kappa-1} + c(\hat{\mathbf{E}}_{\rho:}^{\tau,\kappa})^T\|_2^2$ after applying K_2 ADMM iterations. To summarize, the distributed regularized CCA(DR-CCA) algorithm is tabulated as Algorithm 3.

Algorithm 3: DR-CCA

```
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
$\mathbf{b}_\rho^0(j) = \breve{\mathbf{b}}_\rho^0(i) = \mathbf{p}_\rho^0(j) = \breve{\mathbf{p}}_\rho^0(i) = 0$ for $\rho = 1, 2, ..., q$, $j = 1, 2, ..., p_1$ and $i = 1, 2, ..., p_2$.
for $\tau = 1, 2, \ldots,$ do
    for $\rho = 1, 2, \ldots, q$ do
        for $\kappa = 1, 2, \ldots, K$ 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 (4.43), (4.46) and
            (4.47) respectively for $j = 1, \ldots, p_1$ and $j \in \mathfrak{A}$.
            Update $\hat{\mathbf{E}}^{\tau,\kappa}(\rho, i)$, $\breve{\mathbf{b}}_\rho^{\tau,\kappa}(i)$ and $\breve{\mathbf{p}}_\rho^{\tau,\kappa}(i)$ via (4.52), (4.53)
            and (4.54) respectively for $i = 1, \ldots, p_2$ 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 tolerance
$\epsilon$, then break.
end for
```

From the convergence claims in [61] it follows that, as $K_2 \to \infty$, $\hat{\boldsymbol{\eta}}_{j,\rho}^{\tau,\kappa-1} \to \hat{\mathbf{D}}_{\rho:}^{\tau,\kappa-1}\mathbf{X}$, $\hat{\boldsymbol{\eta}}_j^{\tau-1} \to \hat{\mathbf{D}}^{\tau-1}\mathbf{X}$, $\hat{\boldsymbol{\nu}}_{i,\rho}^{\tau,\kappa-1} \to \hat{\mathbf{E}}_{\rho:}^{\tau,\kappa-1}\mathbf{Y}$, and $\hat{\boldsymbol{\nu}}_i^{\tau-1} \to \hat{\mathbf{E}}^{\tau-1}\mathbf{X}$. Similary, as $K \to \infty$, the $\hat{\mathbf{D}}_{\rho:}^{\tau,\kappa}$ and $\hat{\mathbf{E}}_{\rho:}^{\tau,\kappa}$ obtained from (4.43) and (4.52) go to the optimizer of $\hat{\mathbf{D}}_\rho^\tau$ and $\hat{\mathbf{E}}^\tau$ of (4.14) and (4.24) , respectively. Further, as the block coordinate cycle $\tau \to \infty$ the updates $\hat{\mathbf{D}}^\tau$ and $\hat{\mathbf{E}}^\tau$ will approach $\delta(\varepsilon)$-close to a stationary point of the cost in (4.11) where $\lim_{\epsilon\to 0}\delta(\epsilon) = 0$.

4.3 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 ρth 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, the 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 $\hat{\mathbf{D}}\boldsymbol{\Sigma}_x\hat{\mathbf{D}}^T$ and given that there are M sources, it should ideally have rank equal to M. Thus, in an ideal noiseless setting $\hat{\mathbf{D}}\boldsymbol{\Sigma}_x\hat{\mathbf{D}}^T$ should have M nonzero eigenvalues corresponding the energy contributed by the M field sources. In practice, the ensemble $\hat{\mathbf{D}}\boldsymbol{\Sigma}_x\hat{\mathbf{D}}^T$ is estimated by the sample-average based estimator $\hat{\mathbf{D}}\hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{D}}^T$, which will be affected by noise and the usage of a finite number of samples estimating $\hat{\boldsymbol{\Sigma}}_x$. In general, rank $(\hat{\mathbf{D}}\hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{D}}^T) = q$ 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 $\rho = 1, ..., q$, we estimate the number of 'source-related' eigenvalues in $\hat{\mathbf{D}}\hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{D}}^T$ using the cumulative percent variance (CPV) approach, see. e.g. [27] and [90]. Specifically, the percentage of total variance captured by the first A largest eigenvalues of $\hat{\mathbf{D}}\hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{D}}^T$, namely $\lambda_{x,1} \geq \lambda_{x,2} \geq, ..., \geq \lambda_{x,A}$ is quantified as

$$CPV_A(\%) = \frac{\sum_{i=1}^{A} \lambda_{x,i}}{\sum_{i=1}^{q} \lambda_{x,i}} \times 100, \tag{4.55}$$

where q corresponds also to the total number of eigenvalues of $\hat{\mathbf{D}}\hat{\mathbf{\Sigma}}_x\hat{\mathbf{D}}^T$. 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 $\lambda_{D,1} = \lambda_{D,2} = ... = \lambda_{D,q}$ and $\lambda_{E,1} = \lambda_{E,2} = ... = \lambda_{E,q}$ 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 contain information about 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 $\rho = 1, ..., q$, 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, $\{\lambda_{D,\rho}, \lambda_{D,\rho}\}_{\rho=1}^q$ 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 ϕ_D, ϕ_E, $\{\lambda_{D,\rho}, \lambda_{D,\rho}\}_{\rho=1}^{q}$ in CR-CCA (or DR-CCA).

Fixing the coefficients $\lambda_{D,\rho} = \lambda_{D,0}$ and $\lambda_{E,\rho} = \lambda_{E,0}$ for $\rho = 1, ..., q$, 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 $\widetilde{\mathcal{R}}_D$ and $\widetilde{\mathcal{R}}_E$ 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 $\widetilde{\mathcal{R}}_D = \widetilde{\mathcal{R}}_E$. Since there is a possibility that a row of $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}}$, say $\hat{\mathbf{D}}_{\rho:}$ and $\hat{\mathbf{E}}_{\rho:}$, 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 $\widetilde{\mathcal{R}}_D = \widetilde{\mathcal{R}}_E$, 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 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 4. 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.

Algorithm 4: Sparsity-Controlling Coefficient Selection

```
• 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: $\phi_D = \phi_E = 0$, $\{\lambda_{D,\rho} = \lambda_{E,\rho} = 0\}_{\rho=1}^{q}$.
    Step 2.1) Find estimates $\lambda_D^{max}$, $\lambda_E^{max}$, 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 $\phi_D$, $\phi_E$, and $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^{q}$ alternatively.
    Initialize $\phi_D = \phi_E = \lambda_{D,\rho} = \lambda_{E,\rho} = 0$ for $\rho = 1, ..., q$.
    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 $\mathcal{C}_{E,c} \subset \mathcal{C}_E$: $\{\lambda_{E,\rho} = \lambda_{E,\rho} + \Delta_\lambda\}_{\rho=1}^{q}$;
      Run CR-CCA (or DR-CCA) and update $\mathcal{C}_{D,c}$ and $\mathcal{C}_{E,c}$.
    end
    3.2) Increase $\phi_D$ and $\phi_E$ gradually (using step size $\Delta_\phi$) 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 $\hat{\mathbf{E}}$.
    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 $\rho = 1, ..., q$.
      (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 $\rho = 1, ..., q$.
      (iii) Run CR-CCA (or DR-CCA), and update $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$, $\widetilde{\mathcal{R}}_D$, $\widetilde{\mathcal{R}}_E$.
      (iv) If $\widetilde{\mathcal{R}}_D = \widetilde{\mathcal{R}}_E = \mathcal{R}_c$
          $\mathbf{If}$ $\hat{\mathbf{D}}$, $\hat{\mathbf{E}}$ are $robust$, $\mathbf{then\ STOP}$.
          else, $\mathbf{ExitFlag} = \mathbf{1}$, and $\mathbf{Break\ while}$.
          $\mathbf{end}$
        $\mathbf{end}$
      (v) If the number of the common non-zero rows in $\hat{\mathbf{D}}$ and $\hat{\mathbf{E}} < |\mathcal{R}_c|$, $\mathbf{ExitFlag} = \mathbf{2}$
        and $\mathbf{Break\ while}$.
    $\mathbf{end}$
    If $\mathbf{ExitFlag} = \mathbf{2}$, then update $M_c = |\mathcal{R}_c|$.
    If $\mathbf{ExitFlag} = \mathbf{1}$, 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,..., 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 $M_c = |\mathcal{R}_c|$.
  $\mathbf{end}$
```

The $\mathbf{Break\ while}$ in Alg. 3 terminates the execution of the inner while loop, and the $\mathbf{STOP}$ 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 $\upsilon = 0.5$.

4.4 Simulation Results

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 [79] abbreviated as PMD; and $ii)$ intelligent K-Means (iK-Means) clustering approach in [13], [46] 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 [38].

Consider a setting in which $M = 3$ uncorrelated sources 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 $\mathrm{s}_1(t)$, $\mathrm{s}_2(t)$, and $\mathrm{s}_3(t)$, 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 $\mathrm{s}_1(t)$, $\mathrm{s}_2(t)$, and $\mathrm{s}_3(t)$, 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 $\mathrm{s}_1(t)$, while source $\mathrm{s}_2(t)$ 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 $\mathrm{s}_3(t)$, and the remaining sensors observe just sensing noise. The low number of sensors setting simulation results are depicted in Fig. 4.3, while Fig. 4.4 shows the performance for the $medium$ setting, and Fig. 4.2, Fig. 4.5, Fig. 4.6 correspond to the $high$ number of sensors setting. In Fig. 4.2, Fig. 4.3, Fig. 4.4, Fig. 4.5, and Fig. 4.7, 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. 4.2 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, $\lambda_{D,\rho}$s and $\lambda_{E,\rho}$s in CR-CCA and DR-CCA are selected using the Alg. 3 in Sec. 4.3. The sparsity-controlling coefficients in PMD are selected through cross-validation, whose details can be found in [79]. Fig. 4.2 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. 4.2 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 affect 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 affected 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, making the probability curve to be straight. The same conclusions with Fig. 4.2 can also be obtained from Fig. 4.3 and Fig. 4.4 under the low and $medium$ number of sensors setting, respectively. The results in Fig. 4.2, Fig. 4.3 and Fig. 4.4 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. 4.5 depicts the performance of DR-CCA for different network ADMM iterations, 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. Another interesting property is that CR-CCA outperforms DR-CCA, which is expected. Note that as $K_2 \to \infty$, 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. 4.3, and the number of data vector is set as $N = 500$. Fig. 4.6 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 τth block coordinate cycle, in the beginning of updating the ρth 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 κth 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 $\{v_{j,t,\rho}^{j',\tau,\kappa-1}(\ell), \eta_{j',t,\rho}^{\tau,\kappa-1}(\ell), w_{j,\rho}^{j',\tau,\kappa}(\ell),\ u_{j',\rho}^{\tau,\kappa}(\ell)\}_{t=0,\ell=1,j' \in \mathcal{N}_j^{\mathfrak{A}}}^{N-1,K_2}$. Here the number of network ADMM iterations is set as $K_2 = 20$, and 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., $\tau = 0, 1, ..., 99$, for different testing cases, and also apply the parameter selection algorithm in Alg. 3 to DR-CCA. The simulation results are shown in Fig. 4.7 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 $q = 5, 6, 7$.

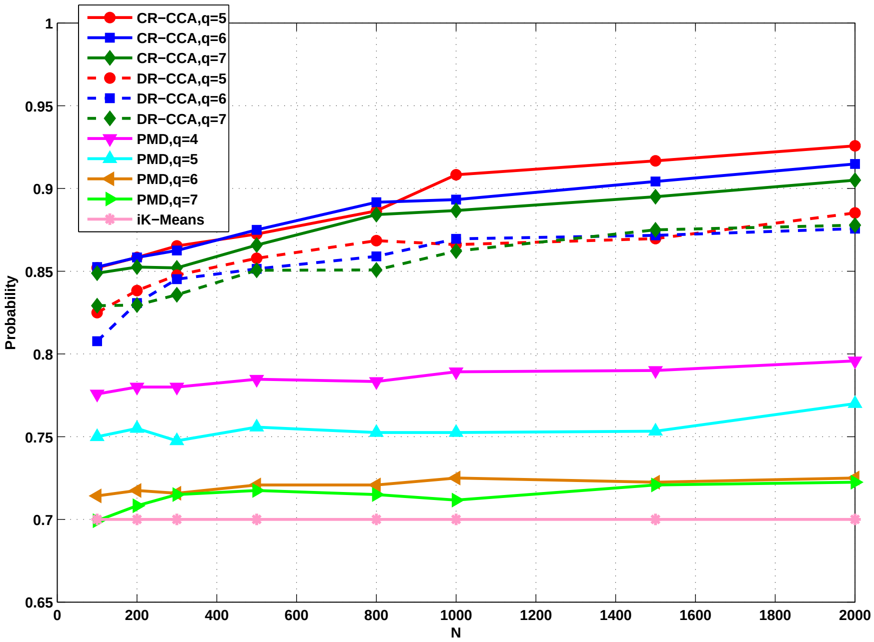

Figure 4.2. Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 60$ sensors.

4.5 Conclusion

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

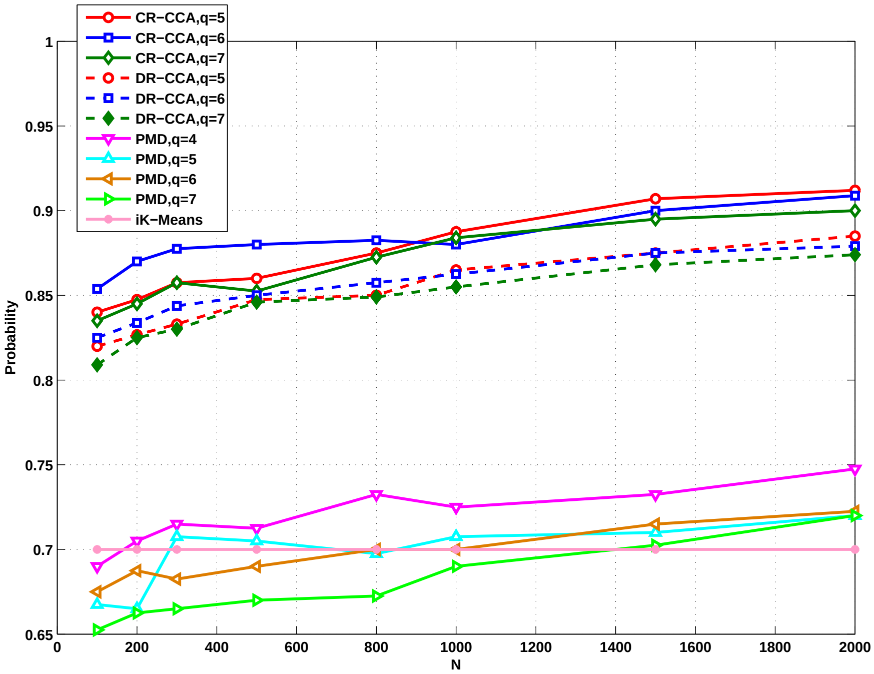

Figure 4.3. Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 20$ sensors.

collaborate. The potential of the proposed CR-CCA and DR-CCA in correctly recovering clusters for heterogeneous sensors, was demonstrated via extensive numerical tests.

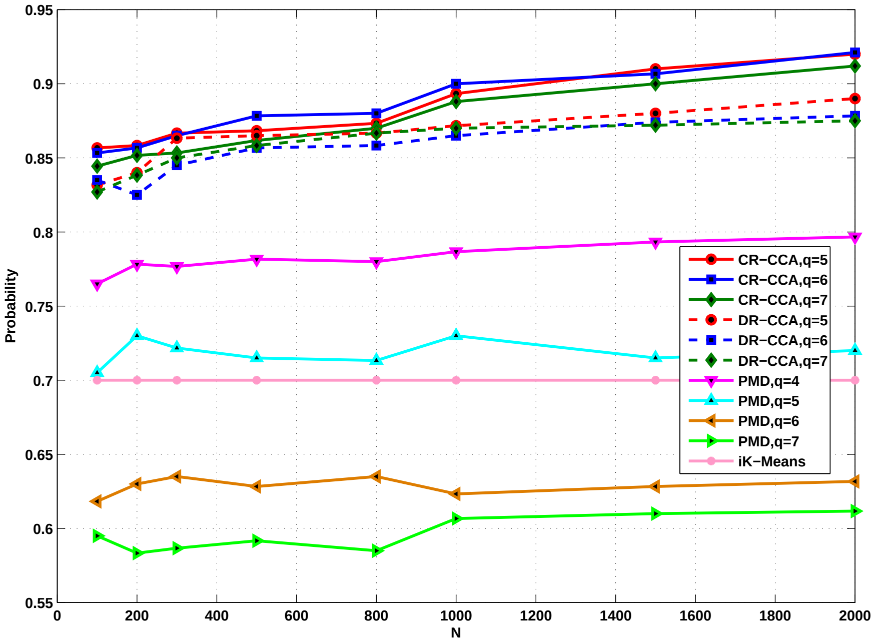

Figure 4.4. Probability of correctly clustering sensors vs. number of data vectors for $p_1 + p_2 = 30$ sensors.

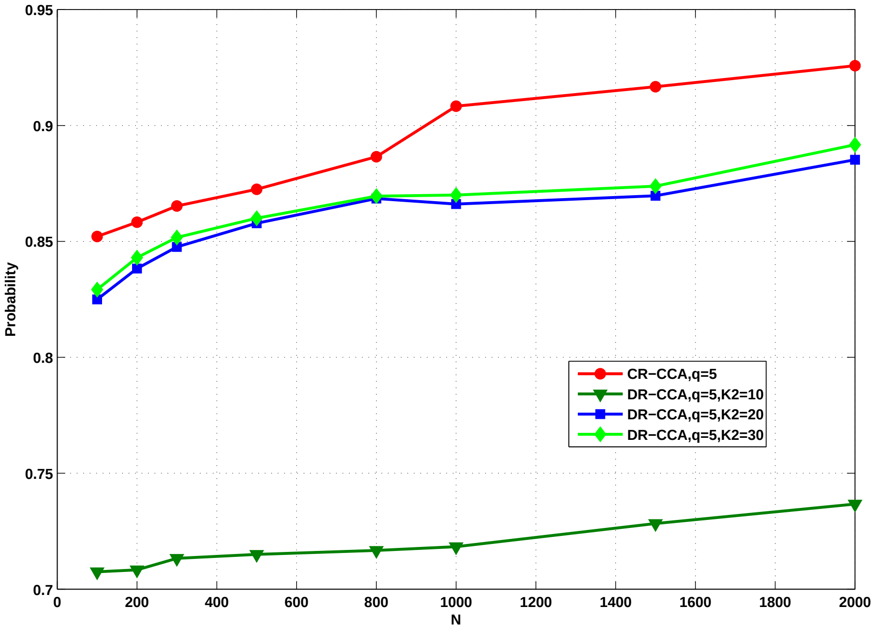

Figure 4.5. Probability of correctly clustering sensors vs. number of data vectors for different number of ADMM iterations.

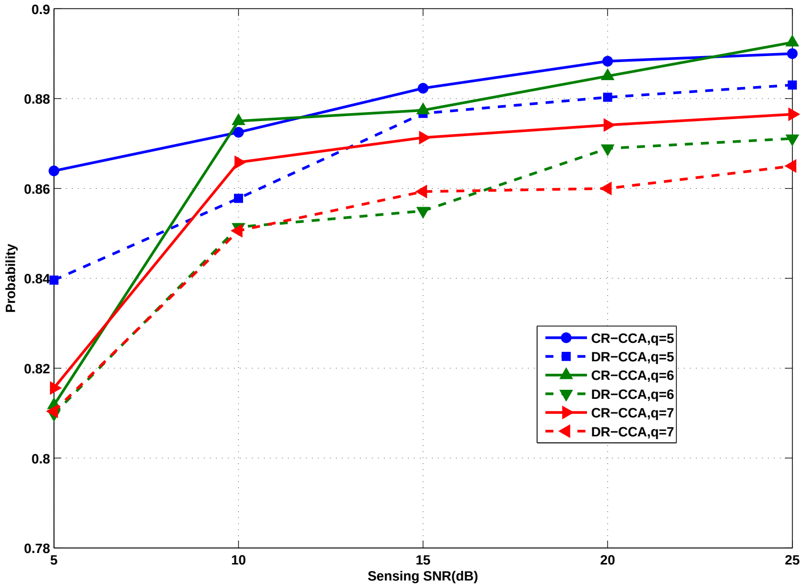

Figure 4.6. Probability of correctly clustering sensors vs. sensing SNR.

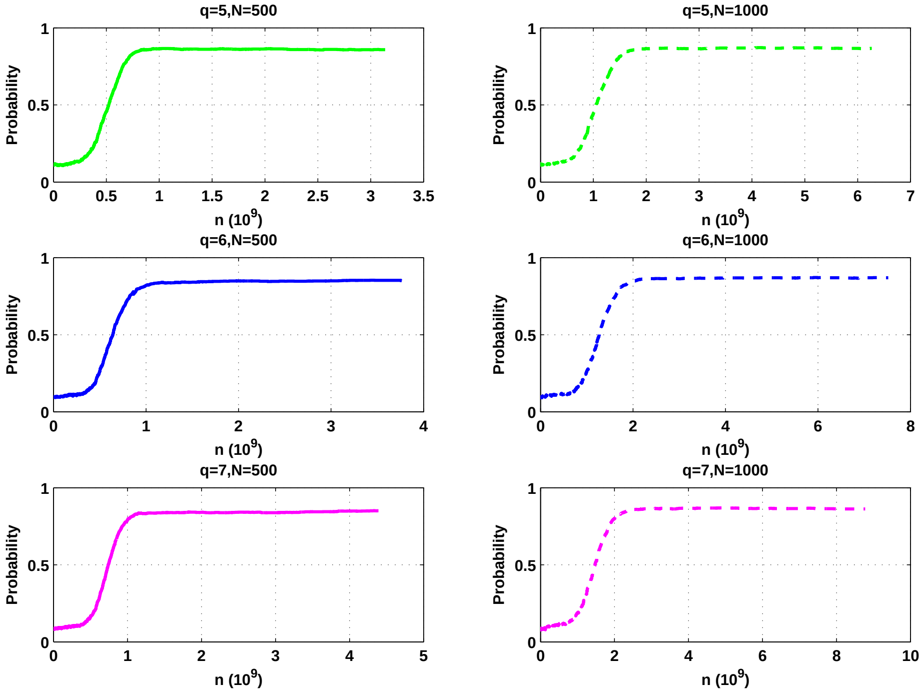

Figure 4.7. Probability of correctly clustering sensors vs. the average number of scalars communicated per sensor for $N = 500$ measurements (left) and $N = 1000$ measurements (right).

CHAPTER 5

COMMUNICATION EFFICIENT FIELD RECONSTRUCTION

5.1 Problem Description

Consider a field consisting of p spatially scattered sensors, which is generated by M underlying sources, while the number of sources M is unknown. The source signals placed in different spatial locations are modeled as random uncorrelated processes, namely $s_m(t)$, where m denotes the source index and t is time instant. We assume the source signals are wide sense stationary, which implies that their ensemble average is time-invariant.

The source signals are reaching the sensing units via multipath propagation channels. And the channel coefficients from the mth source to sensor j is modeled as a finite impulse response filter with coefficients $\mathbf{h}_{j,m} = [h_{j,m}(0), \ldots, h_{j,m}(L-1)]$, where L represents the maximum number of taps these filters can have. The channel coefficients are not available and modeled as random Gaussian variables. Inspired by the diffusion fields which are pertinent for modeling how heat or chemical substances are diffusing in space and time. We assume the energy of the source signals is decreasing exponentially with distance (corresponds to the propagation mechanics of a diffusion field [8]).

A sensor is considered to observe a source if the sensed source signal energy at the sensor's location is more than 10% of the signal power at the emission point, otherwise it is assumed that the sensor does not contain any information about that source. Thus, a threshold corresponding to 10% of a source's signal energy is used to determine which source every sensor observes. Given that the source signal power attenuates exponentially fast with propagation distance, the field sources are quite localized affecting a limited number of sensors in the monitored field. Let $\mathcal{S}_j$ contain the indices of the sources observed by sensor j, the measurement $x_j(t)$ adheres to the following model

$$x_j(t) = \sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) s_m(t-\tau) + w_j(t) \tag{5.1}$$

where, $c_{j,m}$ is a positive coefficient quantifying the attenuation experienced by the mth source signal at sensor j; for a diffusion field it can be quantified as (see details in [8])

$$c_{j,m} = \frac{1}{4\pi} e^{-||\mathbf{p}_j - \mathbf{p}_{s_m}||^2} \tag{5.2}$$

where $\mathbf{p}_j, \mathbf{p}_{s_m} \in \mathbb{R}^2$ correspond to the positions of sensor j and source $s_m(t)$, respectively, and $|| \cdot ||_2$ denotes Euclidean norm. Further, $w_j(t)$ corresponds to the zero-mean sensing noise with variance σ_w^2.

In existing sensing protocols developed for FC-based topologies, all sensor measurements at any time instant t have to be transmitted back to the FC for application dependent processing. Thus, at every t all p sensors transmit to the FC, leading to a communication cost of the order of $\mathcal{O}(pt)$. The communication cost is prohibitively high given that the number of sensors p is large for many different sensing applications such as environmental monitoring, surveillance and so on [12, 85].

The objective here is to utilize the spatial correlation among the sensor measurements to significantly reduce the communication load. The aforementioned goal can be divided into two steps: i) Identifying the set of sensors acquiring spatially correlated measurements; and ii) Learning the statistical models that the different correlated groups of sensor measurements follow. Toward this end, the first step boils down to determining the sets $\mathcal{M}_m$, $\forall\, m \in \{1, ..., M\}$ that contain the sensor indices whose measurements contain information about source $s_m(t)$.

The second step relies on the fact that a set of correlated sensor measurements are linearly dependent on common source signals. The main idea is to transmit the measurements of only one sensor in a correlated set $\mathcal{M}_m$, namely a head sensor, and rely on adaptive filtering to learn proper linear transformations (filtering coefficients) that can be used to reconstruct the data of all other sensors in $\mathcal{M}_m$ using only the head sensor measurements. Learning has to be performed under a setting where the source-to-sensor channel coefficients are unknown, and source signals are not available. In detail, in every set of correlated measurements $\mathcal{M}_m$, the head sensor will be designated as a reference sensor whose measurements will be used to linearly reconstruct all other sensors' measurements in $\mathcal{M}_m$. Thus, communication savings will be introduced by communicating to the FC only the head sensor measurements.

5.2 Learning and Reconstructing the Monitored Field

The proposed framework entails a training phase during which training data are acquired at the sensors and used to learn the statistical correlation structure of the monitored field. Specifically, three tasks are carried out here: i) determining the unknown number of sources M; ii) identifying the M sets of sensors with correlated measurements, i.e., $\{\mathcal{M}_m\}_{m=1}^{M}$; and iii) learning pertinent filters to reconstruct the sensor measurements in a correlated set $\mathcal{M}_m$ using only the head sensor's acquired measurements.

5.2.1 Determining the Number of Sources

A framework to identify the number of informative sources in the field and subsequently determine the correlated clusters is proposed. A novel combination of sample-averaging along with PCA is employed to effectively reduce the sensing noise variance and estimate the number of sources.

During training each sensor, say sensor j, acquires N measurements $x_j(t)$ for $t = 1, \ldots, N$ adhering to the model in (5.1). Each sensor then performs sample-averaging using a moving-average (MA) filter, producing the MA processed measurements

$$\bar{x}_j(t) = P^{-1} \textstyle\sum_{\ell=1}^{P} x_j(t + \ell - 1) \tag{5.3}$$

where P denotes the length of the MA filter performing averaging, with $t = 1, \ldots, N - P + 1$, while $N > P$.

The next step is to stack all the MA measurements in (5.3) in a vector $\bar{\mathbf{x}}_t := [\bar{x}_1(t) \ldots \bar{x}_p(t)]^T$ that can be formed at the FC. The MA data vectors are subsequently used to estimate the MA data covariance matrix via sample-averaging as

$$\hat{\mathbf{\Sigma}}_{\bar{x}} = (N - P + 1)^{-1} \sum_{t=1}^{N-P+1} [\bar{\mathbf{x}}_t - \mathbf{m}_{\bar{x}}][\bar{\mathbf{x}}_t - \mathbf{m}_{\bar{x}}]^T, \tag{5.4}$$

where $\mathbf{m}_{\bar{x}} := (N - P + 1)^{-1} \sum_{t=1}^{N-P+1} \bar{\mathbf{x}}_t$ denotes the sample-average estimate for the mean of $\bar{\mathbf{x}}_t$.

It is demonstrated in Appendix D that as the number of training data increases, while the length of the MA filter P is sufficiently large to make the sensing noise variance sufficiently small (arbitrarily close to zero for increasing P), then the number of eigenvalues of $\hat{\mathbf{\Sigma}}_{\bar{x}}$, whose amplitude is larger than the MA sensing noise variance σ^2/P, will be equal to the number of sources M. Effectively the MA filtering helps reducing the sensing noise variance while preserving the source signal power as shown in Appendix D. Moreover, MA filtering transforms the convolutive model in (5.1) into a simpler low dimensional linear model on which MA data $\bar{\mathbf{x}}_t$ adhere to. Then, PCA is utilized to determine the source-related principal eigenvalues.

5.2.2 Clustering Sensor Measurements

The first step in the proposed field reconstruction scheme is to determine clusters of correlated sensor measurements that contain information about common sources, namely the sensor subsets $\mathcal{M}_m$ which contains all sensors sensing source $s_m(t)$ for $m = 1, \ldots, M$. The norm-one regularized canonical correlations framework in [15] to cluster a set of sensor measurements based on source content, will be enhanced with a PCA scheme used to check the validity of the resulting clusters and achieve flawless clustering for sufficiently large number of measurements, even in settings where sensors observe more than one sources (overlapping sources). This is to be contrasted with the approach in [15] which may not perform well for an overlapping setting, where the clustering performance is sensitive to certain parameter selection.

The CS-CCA algorithm proposed in Sec. 3.2.1 is utilized here to cluster sensor measurements after substituting the two data sequences in (3.2) by $\boldsymbol{\chi}_t$ and $\boldsymbol{\psi}_t$, which are defined as:

$$\boldsymbol{\chi}_t = [\bar{\mathbf{x}}_{t-1}^T, \cdots, \bar{\mathbf{x}}_{t-f}^T]^T, \tag{5.5}$$

$$\boldsymbol{\psi}_t = [\bar{\mathbf{x}}_t^T, \cdots, \bar{\mathbf{x}}_{t+f-1}^T]^T \tag{5.6}$$

Nonetheless, the clustering performance of CS-CCA may not be perfect. To improve performance, an iterative combination of the CS-CCA framework and PCA framework is proposed next. As it will be shown in Sec. 5.2.3 this combination can lead to perfect sensor clustering as the number of measurements N, and MA filter length P are increasing.

The idea here is to determine how many sources are present in each of the estimated clusters $\hat{\mathcal{M}}_m$ obtained from CS-CCA when applied in all data. If a cluster has one source no further splitting is needed, whereas if a cluster contains information about multiple sources the goal is to identify which sources are contained in that cluster. For each of the clusters $\hat{\mathcal{M}}_m$ the corresponding sensor measurements are stacked in vectors $\bar{\mathbf{x}}_{m^1}^1(t)$, where the subscript indicates the cluster index and the superscript the iteration index of the alternating process between CS-CCA and PCA, while $t = 1, \ldots, \bar{N}$ and $m^1 \equiv m = 1, \ldots, M$. These vectors are used to form the sample-averaging covariance estimates

$$\mathbf{\Sigma}^1_{x,m^1} := \bar{N}^{-1} \sum_{\tau=1}^{\bar{N}} [\bar{\mathbf{x}}^1_{m^1}(\tau) - \mathbf{m}_{\bar{x}^1_{m^1}}][\bar{\mathbf{x}}^1_{m^1}(\tau) - \mathbf{m}_{\bar{x}^1_{m^1}}]^T, \tag{5.7}$$

where $\mathbf{m}_{\bar{x}^1_{m^1}}$ denotes the sample-average of vectors $\{\bar{\mathbf{x}}^1_{m^1}(t)\}_{t=1}^{\bar{N}}$. Here PCA can be applied to determine the number of source-related PCs, namely $M^1_{m^1}$ for $m^1 = 1, \ldots, M$. For sufficiently large P (length of MA filter) the noise-related eigenvalues in (5.7) will be significantly smaller than the source-related PCs which can be easily separated using thresholding. $M^1_{m^1}$ can be used to estimate the number of sources for which information is contained in each cluster $\hat{\mathcal{M}}^1_{m^1}$.

For each cluster $\hat{\mathcal{M}}^1_{m^1}$ for which $M^1_{m^1} = 1$ no further splitting is applied in the cluster since it has been estimated that only one source is present in the measurements in $\hat{\mathcal{M}}^1_{m^1}$. However, if $M^1_{m^1} > 1$ this implies that more than one sources are sensed by the measurements in $\hat{\mathcal{M}}^1_{m^1}$, and CS-CCA is applied within the cluster in order to separate the measurements further based on their information content. Toward this end, the CS-CCA formulation in (3.6) can be applied after using matrices $\mathbf{E}$ and $\mathbf{D}$ of size $M^1_{m^1} \times |\hat{\mathcal{M}}^1_{m^1}|f$ where $|\hat{\mathcal{M}}^1_{m^1}|$ denotes the number of measurements in $\hat{\mathcal{M}}^1_{m^1}$. Further, the $\boldsymbol{\chi}_t$ and $\boldsymbol{\psi}_t$ vectors will be of size $|\hat{\mathcal{M}}^1_{m^1}|f \times 1$ and constructed as described earlier using only the measurements in $\hat{\mathcal{M}}^1_{m^1}$. Thus, if $M^1_{m^1} > 1$ CS-CCA is employed (iteration index 2) to perform further splitting of cluster $\hat{\mathcal{M}}^1_{m^1}$ into $M^1_{m^1}$ clusters denoted as $\hat{\mathcal{M}}^2_{m^2}$, where $m^2 = 1, \ldots, M^1_{m^1}$ and $m \in \{1, \ldots, M\}$ for all these clusters that give more than one PCs after iteration 2. Then, PCA is applied again during iteration 3 to decide whether any of the clusters $\hat{\mathcal{M}}^2_{m^2}$ need to be spit further as described earlier. The aforementioned iterative alternation between CS-CCA and PCA is continued until all resulting clusters contain measurements whose corresponding covariance contains at most one source-related

PC, or the resulting cluster contains a single measurement which may have information about multiple sources.

Next, PCA is applied to (i) merge clusters that contain information about the same source; and (ii) further conclude which sources are contained in the single-measurement clusters which sense multiple sources. To carry out task (i) any two clusters whose covariance has a single source-related eigenvalue are combined into a single-cluster if the resulting cluster gives a measurement covariance matrix which also has a single source-related eigenvalue. PCA is used again to find the source-related PCs. If only one PC is present in the covariance of the resulting cluster then it is ensured that both clusters combined contain information about the same source and merging is valid, otherwise merging is not valid. Task (ii) focuses on identifying which single-source clusters carry information about the sources contained in these single-measurement clusters that sense more than one sources. After completion of task (i) each of the remaining single-measurement clusters are merged with those single-source clusters that contain measurements of sensors that are physical neighbors. This is done since sources sensed by the same sensor have to be geographically close in distance. In the merging of the neighboring clusters with a single-measurement cluster, we select only those neighboring clusters that, when merged with the single-measurement cluster, do not generate a covariance matrix with a number of PCs greater than the number of single-source neighboring clusters (which actually is equal to the number of sources contained in the single-measurement cluster). This process ensures that clustering is performed correctly for a sufficiently large number of measurements and points to the single-source clusters whose source content is contained in the examined single-measurement cluster.

5.2.3 Algorithmic Analysis and Practical Considerations

Interestingly, as the number of training data N and the length of the MA filter P are increasing arbitrarily large, alternating applicability of the CS-CCA and the PCA can achieve flawless clustering of the sensors according to their source content. Specifically, it is demonstrated in Appendix E:

Proposition 3 : Application of CS-CCA and PCA in an alternating fashion as proposed in Sec. 5.2.2, performs correct clustering of the sensors with probability one as the number of data N and the length of the MA filter P are increasing to infinity.

In practice both N and P are finite, which may create challenges when determining the source-related PCs and trying to separate them from the noise-related PCs when applying PCA to split or merge clusters. Specifically, in a scenario where there are sensors whose measurements contain information about multiple sources, CS-CCA may result clusters where some measurements contain information about a single source, while other measurements contain many sources. When N and P are sufficiently large applying PCA as delineated earlier will result multiple source-related PCs corresponding to the sources contained in a cluster. However, when N and P are small due to e.g., low sampling rates, it may be the case that some source-related PCs have small-amplitude and cannot be distinguished from noise-related PCs, especially if some sources have weak presence in the cluster measurements. A process is described next to separate within a cluster sensors that contain information about one source (single-source sensors), from sensors whose measurements are affected by multiple sources (multi-source sensors).

Within a cluster, say $\hat{\mathcal{M}}$, that potentially contains both single-source and multi-source sensors the goal is to extract out the singe-source sensors and place them in the correct single-source cluster. To this end, the process starts by picking a sensor's measurements inside a cluster as a reference signal and checking its ability to reconstruct (via linear filtering) another sensor's measurements within the cluster. Proper linear filters can be determined via the normalized least mean squares (NLMS) process detailed in Sec. 5.2.4. If the reference sensor picked within a cluster is a single-source sensor and there are other single-source sensors within the cluster observing the same source, then the NLMS reconstruction error will have a variance relatively close to the sensing noise variance σ_w^2, see e.g., [78]. Let us denote the subset that contains all these single-source sensors whose measurements can be reconstructed from the reference sensor as $\mathcal{N}_s$. This is to be contradicted with the case where either i) the reference sensor picked is a multi-source sensor, or ii) the remaining sensors within the cluster are multi-source sensors. If all reconstruction errors between the reference sensor and the rest of the sensors in the cluster have variance larger than σ_w^2, it can be concluded that the single-source set $\mathcal{N}_s$ will be empty.

If the resulting set $\mathcal{N}_s$ of potential single-source sensors is not empty, then these are removed from cluster $\hat{\mathcal{M}}$ and merged with that single-source cluster that contains information about the same source. This is found by employing PCA and checking which neighboring single-source cluster can be merged with the measurements from $\mathcal{N}_s$ without increasing the number of PCs in the resulting merged cluster. If set $\mathcal{N}_s$ is empty one possible scenario is that the reference sensor is single-source and the remaining measurements within the cluster examined correspond to multi-source sensors. To check such a case, the measurements corresponding to the reference sensor from cluster $\hat{\mathcal{M}}$ are merged with existing single-source clusters. If there is a single-source cluster, say $\hat{\mathcal{M}}'$ that when merged with the measurements of the reference sensor does not result increasing the number of PCs, then the reference sensor measurements are removed from cluster $\hat{\mathcal{M}}$ and merged with the single-source cluster $\hat{\mathcal{M}}'$. Then, among the remaining sensors in $\hat{\mathcal{M}}$ another one is picked as a reference and the process is repeated until all remaining sensors in $\hat{\mathcal{M}}$ have been used as reference sensors.

The aforementioned process is used to extract the single-source measurements from a cluster $\hat{\mathcal{M}}$ and merge them with those single-source clusters that have the same source information content. The remaining measurements will correspond to multi-source sensors and these multi-source sensors will be merged as explained in Sec. 5.2.2 with neighboring single-source clusters to identify their source content. The algorithm involving alternation between CS-CCA and PCA is summarized in Alg. 5.

Algorithm 5 Clustering via Alternating CS-CCA and PCA

```
1: Initialize: $\bar{\mathbf{x}}_t^0 := \bar{\mathbf{x}}_t$ for $t = 1, \ldots, \bar{N}$
2: Apply CS-CCA using the $\bar{N}$ measurements $\bar{\mathbf{x}}_t^0$ to obtain clusters $\hat{\mathcal{M}}_m$ for $m = 1, \ldots, M$.
3: Set $\hat{\mathcal{M}}_{m^1}^1 \equiv \hat{\mathcal{M}}_m$ and $m^1 \equiv m$.
4: for $k = 1, 2 \ldots, \kappa$ do
5:     Stack the measurements in cluster $\hat{\mathcal{M}}_{m^k}^k$ to form vectors $\{\{\bar{\mathbf{x}}_{m^k}^k(t)\}_{t=1}^{\bar{N}}\}_{m^k}$.
6:     Obtain covariance estimate $\mathbf{\Sigma}_{x,m^k}^k$ as in (5.7) and find the number of source-related PCs,
       namely $M_{m^k}^k$, by applying PCA.
7:     If $M_{m^k}^k = 1$ then $\hat{\mathcal{M}}_{m^k}^k$ is a single-source cluster, else if $M_{m^k}^k > 1$ CS-CCA is applied using
       $\{\bar{\mathbf{x}}_{m^k}^k(t)\}_{t=1}^{\bar{N}}$. to split $\hat{\mathcal{M}}_{m^k}^k$ into smaller clusters $\hat{\mathcal{M}}_{m^{k+1}}^{k+1}$
8:     If all obtained clusters so far have one PC or contain only a single-sensor measurement then
       break, else go back to 4.
9: end for
10: For each of the clusters obtained apply step for small $N, P$ in Sec. 5.2.3 to separate possible
    multi-source sensors from single-source sensors.
11: Merge the single-source clusters into a larger cluster if resulting cluster has only one source-
    related PC.
12: Merge multi-source clusters with neighboring single-source clusters to identify their source-
    content.
```

5.2.4 Learning Statistical Models

5.2.4.1 Single-Source Clusters

Consider a cluster $\hat{\mathcal{M}}_m$ obtained via the process in Sec. 5.2.2 and designated as a single-source cluster. Then, a head sensor i_m is designated in cluster $\hat{\mathcal{M}}_m$. Since all sensors in $\hat{\mathcal{M}}_m$ have correlated measurements, the goal here is to learn $|\hat{\mathcal{M}}_m| - 1$ linear filters that are able to generate all sensors' measurements in $\hat{\mathcal{M}}_m$ using as input the measurements acquired at the head sensor i_m for $m = 1, \ldots, M$. During the training phase, adaptive filtering is employed at the FC to learn the coefficients of the $|\hat{\mathcal{M}}_m| - 1$ linear filters. During the operational stage only the cluster head sensors transmit their measurements to the FC which can then reconstruct the other sensors' measurements in each cluster using only the head sensor data and the learnt filter coefficients. This process will reduce significantly the communication costs.

The head sensor i_m in each cluster $\hat{\mathcal{M}}_m$ is selected arbitrarily. During the training phase of the algorithm all sensors in $\hat{\mathcal{M}}_m$ send their measurements to the FC. The FC treats the head sensor measurements $x_{i_m}(t)$ as input, and each of the remaining $|\hat{\mathcal{M}}_m| - 1$ sensor measurements in $\hat{\mathcal{M}}_m$ is viewed as desired output of a linear filter whose coefficients are determined via the NLMS adaptive filtering approach, see e.g., [78]. Each filter is set to have $\bar{L}$ taps, where $\bar{L}$ is selected sufficiently large such that $L \leq \bar{L}$.

Consider sensor j within cluster $\hat{\mathcal{M}}_m$ with $j \neq i_m$ (not the head sensor). Sensors j and i_m belong to cluster $\hat{\mathcal{M}}_m$, thus they should be sensing a common source $s_m(t)$. Taking into account the data model in (5.1) and ignoring for now the sensing noise the measurements in the frequency domain are given as

$$\mathbf{X}_j(\omega) = c_{j,m}\mathbf{H}_{j,m}(\omega)\mathbf{S}_m(\omega), \tag{5.8}$$

$$\mathbf{X}_{i_m}(\omega) = c_{i_m,m}\mathbf{H}_{i_m,m}(\omega)\mathbf{S}_m(\omega) \tag{5.9}$$

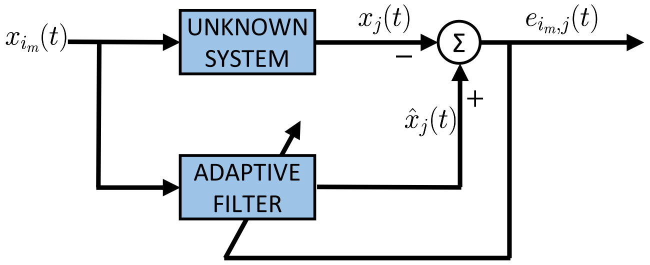

Figure 5.1. Adaptive filter block diagram..

where $\omega \in [-\pi, \pi]$ denotes frequency, while $\mathbf{H}_{j,m}(\omega)$, $\mathbf{H}_{i_m,m}(\omega)$ and $\mathbf{S}_m(\omega)$ denote the frequency responses of the source-to-sensor channels $\{h_{j,m}(\tau), h_{i_m,m}(\tau)\}_{\tau=0}^{L-1}$, and source $s_m(t)$, respectively. From (14), (15) it follows readily

$$\mathbf{X}_j(\omega) = \frac{c_{j,m}}{c_{i_m,m}} \cdot \frac{\mathbf{H}_{j,m}(\omega)}{\mathbf{H}_{i_m,m}(\omega)} \cdot \mathbf{X}_{i_m}(\omega). \tag{5.10}$$

Thus, the NLMS algorithm is trying to learn in time-domain (via a linear filter), the frequency response $\frac{c_{j,m}}{c_{i_m,m}} \cdot \frac{\mathbf{H}_{j,m}(\omega)}{\mathbf{H}_{i_m,m}(\omega)}$ associating the measurements at head sensor i_m, with the measurements of sensor j within single-source cluster $\hat{\mathcal{M}}_m$. Learning the filter in (5.10), via which the measurements of sensor sensor j can be reconstructed using as input the head sensor's measurements, involves the following three steps (see block diagram in Fig. 5.1):

Step 1: Evaluate the estimated output signal for sensor j

$$\hat{x}_j(t) := \mathbf{u}_{i_m}^T(t) \cdot \hat{\mathbf{w}}_{i_m,j}(t) \tag{5.11}$$

where $\mathbf{u}_{i_m}^T(t) := [x_{i_m}(t), x_{i_m}(t-1), \cdots, x_{i_m}(t-\bar{L}+1)]^T$ is the filter input vector, $\hat{\mathbf{w}}_{i_m,j}(t) := [w_{i_m,j,t}(0), w_{i_m,j,t}(1), \cdots, w_{i_m,j,t}(\bar{L}-1)]^T$ contains the filter coefficients used to learn the relationship in (5.10).

Step 2: Calculate the error signal $e_{i_m,j}(t)$

$$e_{i_m,j}(t) = \hat{x}_j(t) - x_j(t) \tag{5.12}$$

Step 3: Update the filter coefficients $\hat{\mathbf{w}}_{i_m,j}(t)$ as

$$\hat{\mathbf{w}}_{i_m,j}(t+1) = \hat{\mathbf{w}}_{i_m,j}(t) + \mu \frac{e_{i_m,j}(t)}{\zeta + \mathbf{u}_{i_m}^T(t)\mathbf{u}_{i_m}(t)} \mathbf{u}_{i_m}(t), \tag{5.13}$$

where μ is the step-size used in NLMS, and ζ is a positive constant. The coefficients are continuously updated at the FC until $\|\hat{\mathbf{w}}_{i_m,j}(t+1) - \hat{\mathbf{w}}_{i_m,j}(t)\|_2$ stops decreasing below a desired threshold. After NLMS algorithm has terminated the final filter coefficients will be denoted by $\hat{\mathbf{w}}_{i_m,j}$.

5.2.4.2 Multi-source clusters

When the cluster $\hat{\mathcal{M}}_m$ considered contains multi-source sensors some generalizations need to be performed to the aforementioned NLMS framework based adaptive filtering approach in order to reconstruct accurately the measurements of the multi-source sensors within $\hat{\mathcal{M}}_m$. Let's denote a multi-source sensor j in $\hat{\mathcal{M}}_m$, for each source $n \in \mathcal{S}_j$ a head sensor, namely i_n, is picked from a neighboring single-source cluster $\hat{\mathcal{M}}_n$ that contains information about source n. Since sensor j observes all sources in set $\mathcal{S}_j$ the corresponding frequency domain equations of (14), (15) are written here as

$$\mathbf{X}_j(\omega) = \sum_{n \in \mathcal{S}_j} c_{j,n} \mathbf{H}_{j,n}(\omega) \mathbf{S}_n(\omega) \tag{5.14}$$

$$\mathbf{X}_{i_n}(\omega) = c_{i_n,n} \mathbf{H}_{i_n,n}(\omega) \mathbf{S}_n(\omega) \tag{5.15}$$

where the same notation as in (14) is used. From (5.14), the following can be obtained

$$\mathbf{X}_j(\omega) = \sum_{n \in \mathcal{S}_j} \frac{c_{j,n} \mathbf{H}_{j,n}(\omega)}{c_{i_n,n} \mathbf{H}_{i_n,n}(\omega)} \mathbf{X}_{i,n}(\omega). \tag{5.16}$$

Comparing (5.16) with the single-source equivalent in (5.10) it turns out that $|\mathcal{S}_j|$ different filters, with frequency responses $\{\frac{c_{j,n}\mathbf{H}_{j,n}(\omega)}{c_{i_n,n}\mathbf{H}_{i_n,n}(\omega)}\}_{n \in \mathcal{S}_j}$, have to be learnt using the NLMS approach. Thus, $|\mathcal{S}_j|$ different adaptive filters will be learnt using NLMS via the following updating steps:

Step 1: Evaluate the filter output $\hat{x}_j(t)$

$$\hat{x}_j(t) = \sum_{n \in \mathcal{S}_j} \mathbf{u}_{i_n}^T(t) \cdot \hat{\mathbf{w}}_{i_n,j}(t) \tag{5.17}$$

where $\mathbf{u}_{i_n}(t) := [x_{i_n}(t), x_{i_n}(t-1), \cdots, x_{i_n}(t-\bar{L}+1)]^T$ corresponds to the input signals from head sensor i_n in single-source cluster $\hat{\mathcal{M}}_n$, and $\hat{\mathbf{w}}_{i_n,j}(t) := [w_{i_n,j,t}(0), w_{i_n,j,t}(1), \ldots, w_{i_n,j,t}(\bar{L}-1)]^T$ corresponds to the $\bar{L}$ coefficients of the adaptive filters used to learn the frequency response $\{\frac{c_{j,n}\mathbf{H}_{j,n}(\omega)}{c_{i_n,n}\mathbf{H}_{i_n,n}(\omega)}\}$. This step is done for all $n \in \mathcal{S}_j$.

Step 2: Calculate the error signal $e_{i_n,j}(t)$

$$e_{i_n,j}(t) = \hat{x}_j(t) - x_j(t) \tag{5.18}$$

Step 3: Update the filter coefficients $\mathbf{w}_{i_n,j}$ as

$$\hat{\mathbf{w}}_{i_n,j}(t+1) = \hat{\mathbf{w}}_{i_n,j}(t) + \mu \frac{e_{i_n,j}(t)\mathbf{u}_{i,n}(t)}{\zeta + \mathbf{u}_{i_n}^T(t)\mathbf{u}_{i_n}(t)}. \tag{5.19}$$

The coefficients $\hat{\mathbf{w}}_{i_n,j}(t+1)$ are iteratively updated until the update difference $\|\hat{\mathbf{w}}_{i_n,j}(t) - \hat{\mathbf{w}}_{i_n,j}(t-1)\|^2$ drops below a desired threshold. In fact, the three updating steps presented earlier could be applied periodically in a setting where the channel coefficients $h_{j,m,t}(\tau)$ are slowly-varying with time.

5.2.5 FC Field Reconstruction

During the operation stage of the proposed scheme, the FC is responsible for reconstructing all sensor measurements using only i) the measurements of the head sensors; and ii) the filter coefficients determined in Sec. 5.2.4. First, the reconstruction of the measurements of a sensor belonging to a single-source cluster $\hat{\mathcal{M}}_m$ is considered at the FC. Let sensor i_m denote the head sensor in cluster $\hat{\mathcal{M}}_m$ with measurements $x_{i_m}(t)$. Sensor j measurements are reconstructed using the filter $\hat{\mathbf{w}}_{i_m,j}$ obtained using NLMS in Sec. 5.2.4.1 as

$$\hat{x}_j(t) = \mathbf{u}_{i_m}^T(t) \cdot \hat{\mathbf{w}}_{i_m,j}, \tag{5.20}$$

where $\mathbf{u}_{i_m}$ contains the head sensor's measurements [cf. Sec. 5.2.4.1].

If sensor j belongs to a multi-source cluster $\hat{\mathcal{M}}_m$, then the FC can reconstruct its measurements readily using the filters $\hat{\mathbf{w}}_{i_n,j}$, obtained in Sec. 5.2.4.2, as follows

$$\hat{x}_j(t) = \sum_{n \in \mathcal{S}_j} \mathbf{u}_{i_n}^T(t) \cdot \hat{\mathbf{w}}_{i_n,j}, \tag{5.21}$$

where $\mathbf{u}_{i_n}(t)$ corresponds to the sensor measurements obtained at head sensor i_n in neighboring single-source cluster $\hat{\mathcal{M}}_n$ [cf. Sec. 5.2.4.2].

5.3 Communication and Computational Costs

In a setting with dense sensor deployment, the communication cost is an important figure of merit to quantify the efficacy of the algorithm. The proposed framework introduces a significant reduction in the communication cost with respect to a setting where all sensors have to transmit their measurements to the FC every time they sense new information. Specifically, consider a setting with M uncorrelated sources (number of single-source clusters) impacting the $N + T$ measurements acquired across the p sensors, where N corresponds to the number of the training data and T denotes the number of measurements obtained during the operational phase of the algorithm. The proposed scheme entails the transmission of $pN + MT$ scalars. In a standard approach, where all sensors transmit to the FC, the communication cost will be $p(N + T)$ scalars. The reduction in the communication cost takes place during the operation stage of the proposed scheme, the reason is that in practical settings the number of sensors p is much larger than the number of sources M, especially when the network consists of densely populated sensing units.

During the training stage of the algorithm, the computational complexity at the FC is $\mathcal{O}(Np^2)$ dominated by PCA. During the operational stage, the reconstruction formulas in (5.20) and (5.21), reveal that the computational complexity at the FC is of the order of $\mathcal{O}(\bar{L} \cdot T \cdot \sum_{m=1}^{M}(|\hat{\mathcal{M}}_m| - 1))$.

The scalability of the algorithm to handle data from large number of sensors is another important aspect in practical deployments. Employing CS-CCA and PCA for a large population of sensors may be challenging. Thus, from the perspective of a practical implementation, the entire field can be split in multiple, physically separated subfields which can be treated separately as described earlier in Secs. 5.2.1 and 5.2.2. In each of these subfields the proposed framework is utilized to identify their respective clusters. Then, the merging approach put forth in Sec. 5.2.2 can be applied in clusters obtained in neighboring subfields.

5.4 Numerical Tests

The proposed framework is tested here in terms of clustering efficiency, field reconstruction quality and communication efficiency. To this end, p sensors are deployed within a 2-D field occupying the normalized region $[0, 1] \times [0, 1]$. The field sources evolve in time according to a first-order autoregressive model (AR-1), i.e., $s_m(t) = F_m \cdot s_m(t-1) + u_m(t)$, for $m = 1, \ldots, M$ where F_m is the autoregressive coefficient which is selected such that $|F_m| \leq 1$, and $u_m(t)$ corresponds to zero-mean white perturbation noise with variance 0.1.

The propagation coefficients for the channels $\{h_{j,m}(l)\}_{j=1,l=1,m=1}^{p,L,M}$ in (5.1) are Gaussian distributed under the constraint that their energy is equal to one, i.e., $\{\sum_{\tau=0}^{L-1} h_{j,m}^2(\tau) = 1\}$, where $L = 3$. Furthermore, the memory length, namely f in (5.5), is set as $f = 1$. The sparsity-controlling coefficients, saying $\{\lambda_{D,\rho}, \lambda_{E,\rho}\}_{\rho=1}^{M}$, are chosen through the λ-selection scheme in Sec. III-C of [15]. The NLMS-based adaptive filter is considered to have $\bar{L} = 20$ taps and the NLMS step-size is set to $\mu = 0.01$. Furthermore, the sensing noise variance is set such that the signal-to-noise-ratio (SNR) in the numerical tests is 13dB. Note that all the simulation graphs below are obtained after averaging over 100 independent Monte Carlo runs. In this section, we apply our proposed algorithm in Sec. 5.2 to the following three scenarios:

$\mathbf{S1}$) *Non-overlapping case:* Three uncorrelated sources $s_1(t)$, $s_2(t)$ and $s_3(t)$ are considered to be at positions $[0.3, 0.4]$, $[0.8, 0.8]$, and $[0.8, 0.2]$, respectively. $p = 30$ sensors are considered here, and each source is sensed by 10 sensors. Each sensor acquires information about one source only, since sources do not overlap here.

$\mathbf{S2}$) *Overlapping case 1:* $p = 15$ sensors are deployed in the field in which $M = 2$ uncorrelated sources are present. Each source is sensed by 5 single-source sensors, the remaining 5 sensors observe both sources.

$\mathbf{S3}$) *Overlapping case 2:* There are $p = 30$ sensors and $M = 4$ sources in the field, see. Fig. 5.2. In detail, 4 sensors (denoted by the purple dots), 4 sensors (red dots), 4 sensors (pink dots), and 3 sensors (blue dots) observe source $s_1(t)$, $s_2(t)$, $s_3(t)$, and $s_4(t)$, respectively. Moreover, 3 sensors (the green dots) acquire information about source $s_1(t)$ and $s_2(t)$, 4 sensors (yellow dots) are affected by both sources $s_3(t)$ and $s_4(t)$, while the remaining sensors (black dots) are far away from any of the four sources and they only sense noise.

5.4.1 Clustering Performance

During the training phase, the number of sources needs to be determined and subsequently the sensor measurements should be clustered into M groups according to their source information content. First, it is demonstrated that the MA filtering approach in Sec. 5.2.1 eliminates the noise-related PCs in the MA data covariance to estimate the number of sources. The length of MA filter is set as $P = 30$ in the experiments. The relative strength (with respect to the summation of all nonzero eigenvalues) of each eigenvalue of the original data covariance $\hat{\mathbf{\Sigma}}_x$ and MA data covariance $\hat{\mathbf{\Sigma}}_{\bar{x}}$ is depicted for the non-overlapping case in Fig. 5.3, and for overlapping case 1 in Fig. 5.4. Clearly, when applying MA the

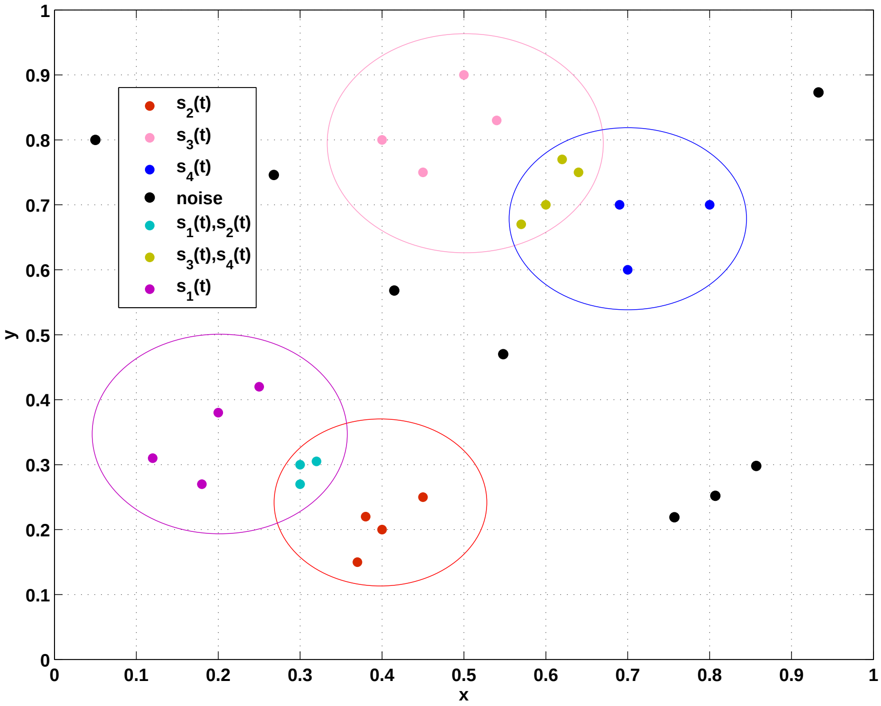

Figure 5.2. Configuration for overlapping case 2..

noise-related eigenvalues are effectively eliminated making easier to find the number of source-related eigenvalues which equals $M = 3$ in Fig. 5.3 and $M = 2$ in Fig. 5.4.

Next, we study the number of iterations alternating between PCA and CS-CCA as described in Sec. 5.2.2, namely the κ constant in Alg. 5, required to achieve flawless sensor clustering. Specifically, in Fig. 5.5, we plot the average number of PCA/CS-CCA iterations needed along with vertical lines showing the spread in the number of iterations versus the number of training data for all three different scenarios S1, S2 and S3. It can be seen that the most challenging case in terms of higher number of iterations required is scenario S3 where sources overlap. Further, it can also be inferred that as the number of training data increases the required number of iterations decreases since CS-CCA performs better in terms of clustering the sensor data and less iterations are required to reallocate

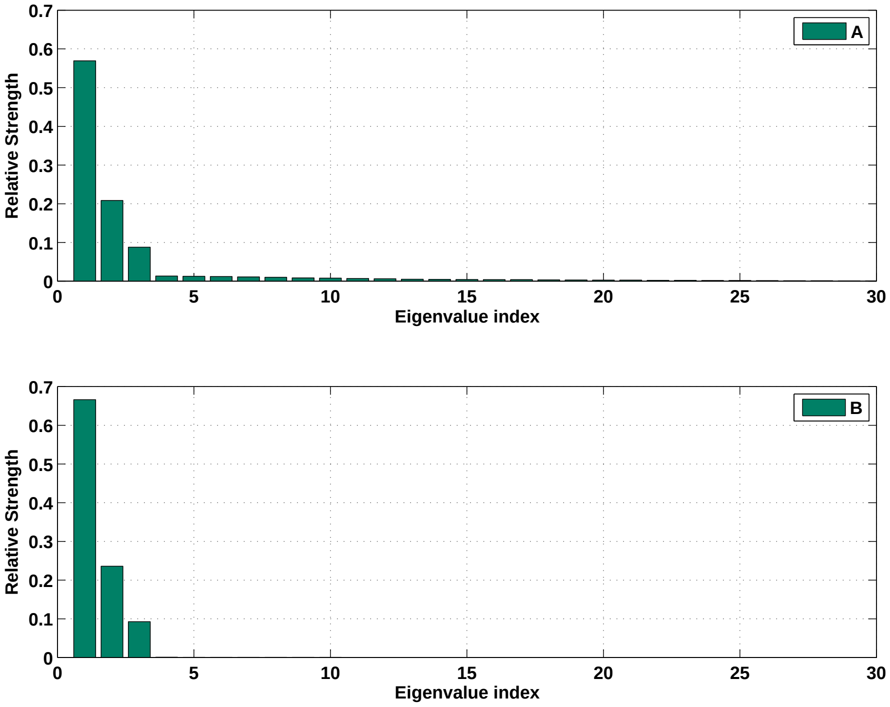

Figure 5.3. Relative strength of the eigenvalues in the data covariance before (A) and after (B) applying MA filtering for the non-overlapping case.

sensors that are wrongly clustered. The probability of correct clustering via CS-CCA is also studied in Figs. 5.6, 5.7 and 5.8 versus the number of training data samples available per sensor. Note that the different probability curves are plotted for a different number of CS-CCA/PCA iterations to demonstrate how the scheme in Sec. 5.2.2 gradually improves the clustering accuracy. Figs. 5.6, 5.7 and 5.8 depict the probability curves for scenarios S1, S2 and S3, respectively. It can be seen clearly in all cases that as the number of iterations increases the probability also increases and eventually reaches one after a certain number of iterations depending on the setting. Note also that the probability increases with the number of training data. Further, the improvement in probability is substantial for the initial

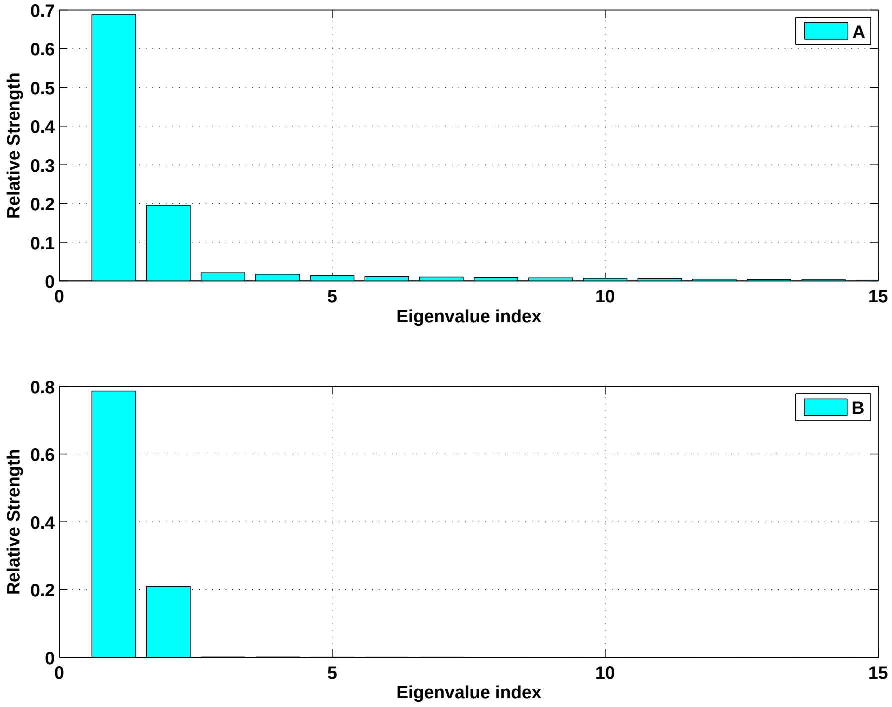

Figure 5.4. Relative strength of the eigenvalues in the data covariance before (A) and after (B) applying MA filtering in overlapping case 1.

iterations, while it declines as more and more iterations are applied and perfect clustering is reached.

5.4.2 Signal Reconstruction

The average reconstruction MSE at the FC is studied here versus the number of training data available. In detail, the performance metric utilized here is the normalized average recovery mean-square error defined as

$$MSE_n = \frac{1}{p}\sum_{j=1}^{p}\left[\frac{\sum_{t=1}^{T}(\hat{x}_j(t) - x_j(t))_2^2}{\sum_{t=1}^{T}(x_j(t))_2^2}\right], \tag{5.22}$$

where the data employed correspond to the data acquired after the training phase during the operational stage. Thus, the normalized average MSE demonstrates how efficient is the

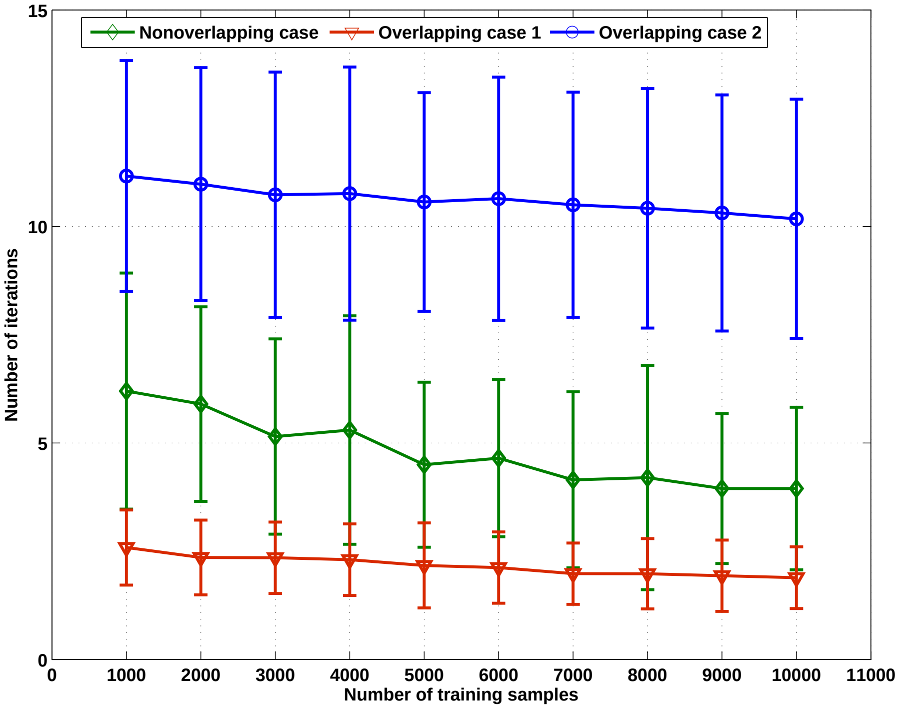

Figure 5.5. Number of iterations required for perfect clustering vs. number of training samples.

proposed algorithm in reconstructing data other than the training data (generalization) that follow the same statistical model. The reconstruction capability of the proposed framework in Sec. 5.2.4 via NLMS adaptive filtering and relying on the source cluster sensors is depicted in Fig. 5.9, for scenarios S1, S2 and S3. Fig. 5.9, corroborates that as the number of training sensor measurements increases the normalized MSE reduces showing the capability of the proposed NLMS-based reconstruction technique in learning the correlation patterns within clusters. Further, the average normalized MSE is similar for all different scenarios considered showing the effectiveness of the proposed scheme irrespective of the setting considered.

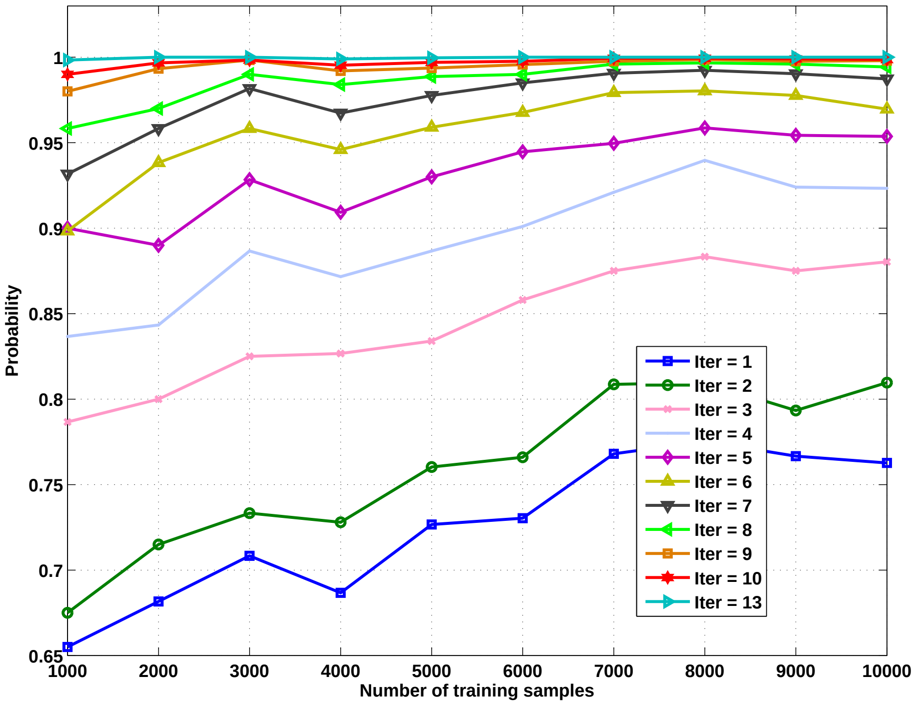

Figure 5.6. Probability of correct sensor clustering vs. number of training samples for different number of CS-CCA/PCA iterations for scenario S1.

5.4.3 Source-Sensor Channel Variation and Communication Efficiency

The adaptivity of the NLMS's reconstruction ability is tested in a setting where the source-sensor channel coefficients change at a given time. The test setting involves $p = 10$ sensors that observe the same source. The channel coefficients $\{h_j(\ell)\}_{j=1,\ell=1}^{10,3}$ are modified at time instant $t = 2000$, and sensor 1 is used as the head sensor whose measurements will be used as a reference signal when applying NLMS to reconstruct the measurements corresponding to the remaining 9 sensors. Fig. 5.10 depicts the relative error versus time t. The relative reconstruction MSE is evaluated here as

$$e_{MSE}(t) := \frac{\sum_{j=2}^{10}(\hat{x}_j(t) - x_j(t))^2}{\frac{1}{4000}\sum_{t=0}^{3999}\sum_{j=2}^{10}(x_j(t))^2}. \tag{5.23}$$

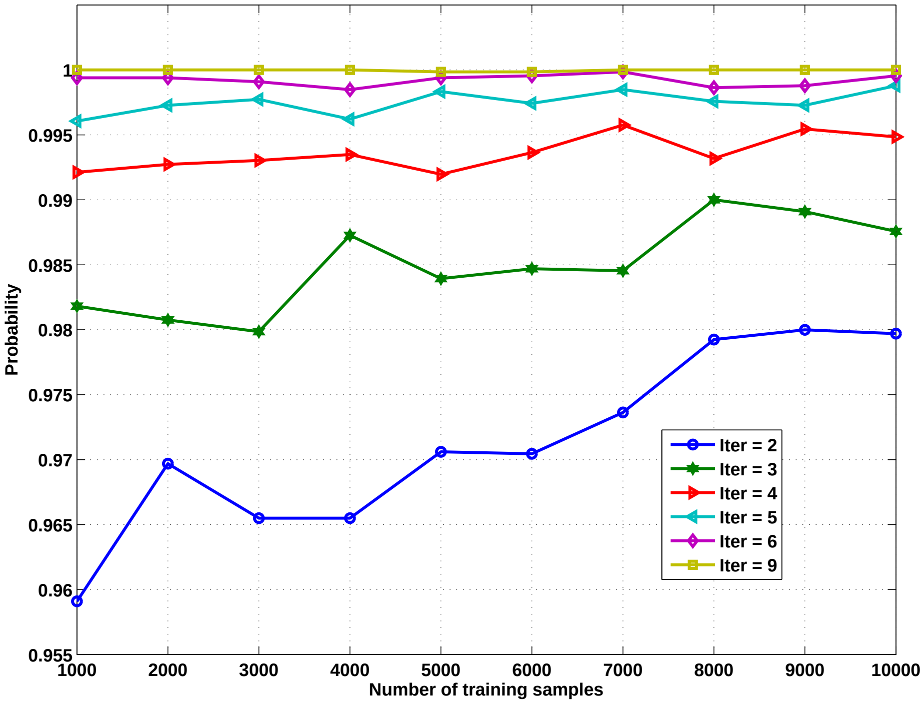

Figure 5.7. Probability of correct sensor clustering vs. number of training samples for different number of CS-CCA/PCA iterations for scenario S2.

Fig. 5.10 corroborates the adaptability of the proposed scheme in the event of changes in the channel coefficients. Clearly, when the source-sensor channel coefficients change, the NMLS approach, after an overshoot in the relative MSE at $t = 2000$ is able to eventually recover and reconstruct all the sensor measurements accurately . Finally, Fig. 5.11 illustrates the communication cost in terms of the number of scalars transmitted from the sensors to the FC versus time. The communication cost is depicted for the three different scenarios S1, S2 and S3. The training phase lasts up to $t = 5000$ where all sensors acquire $N = 5000$ measurements used as training data, while the remaining $T = 5000$ correspond to the measurements acquired during the operational stage. Notice that during the training phase all sensors transmit to the FC so there is no reduction in communication

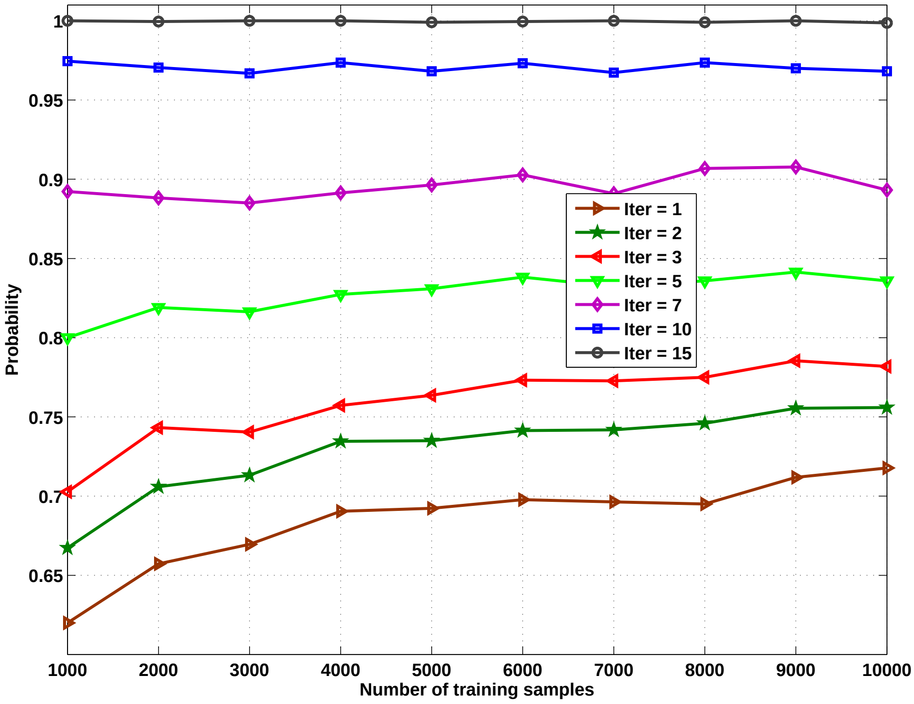

Figure 5.8. Probability of correct sensor clustering vs. number of training samples for different number of CS-CCA/PCA iterations for scenario S3.

cost during this period. However, during the operational phase where sensor clustering has been performed and only the head sensors of each cluster transmit information to the FC the reduction in communication cost is substantial compared to a setting where all sensors keep transmitting their measurements to the FC. Fig. 5.11 depicts the substantial reduction in the number of scalars that need to be communicated from the head sensors to the FC during the operational phase of the novel framework.

5.5 Conclusion

A novel framework was put forth for sensor clustering and communication efficient field reconstruction. Norm-one regularized canonical correlations were combined with

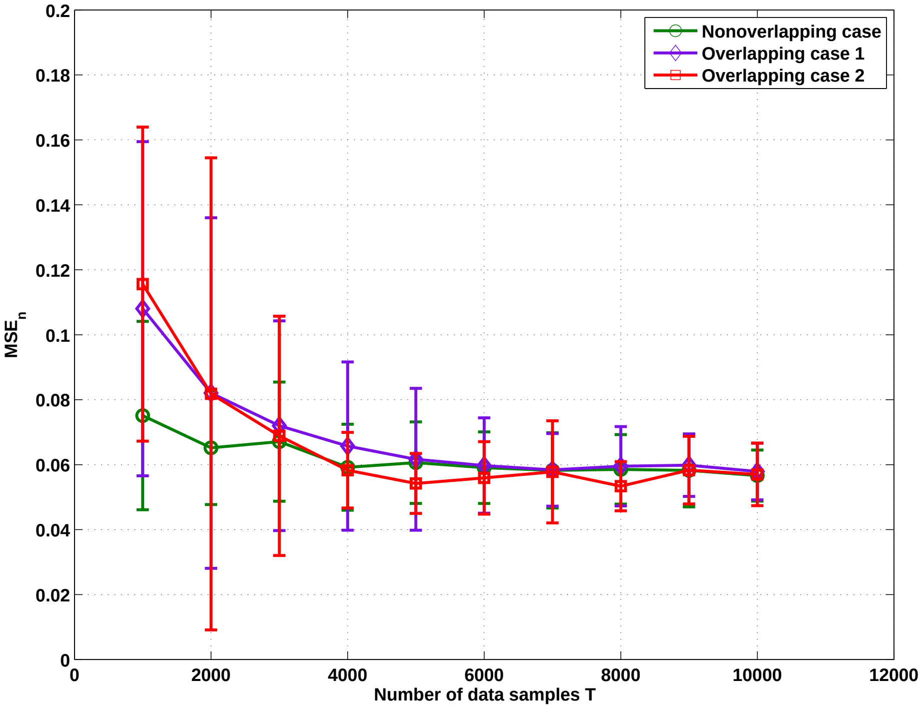

Figure 5.9. Normalized average reconstruction MSE for scenarios S1,S2 and S3.

principal component analysis and moving-average filtering to successfully cluster sensors according to their information content, as the number of training data and moving-average filter length goes to infinity. Utilizing data only from pertinent head sensors in each cluster, the FC reconstructs all sensors' measurements using effective normalized least mean squares techniques, reducing substantially the communication cost. Numerical experiments demonstrated the capability of the proposed novel approach in field reconstruction.

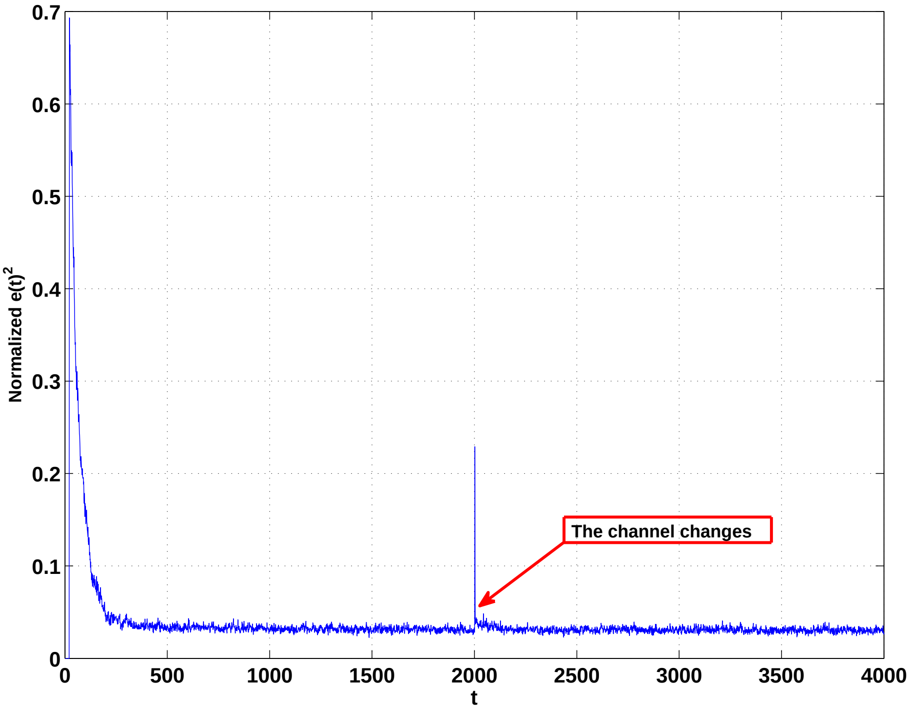

Figure 5.10. Relative average reconstruction MSE when source-sensor channels change at $t = 2000$.

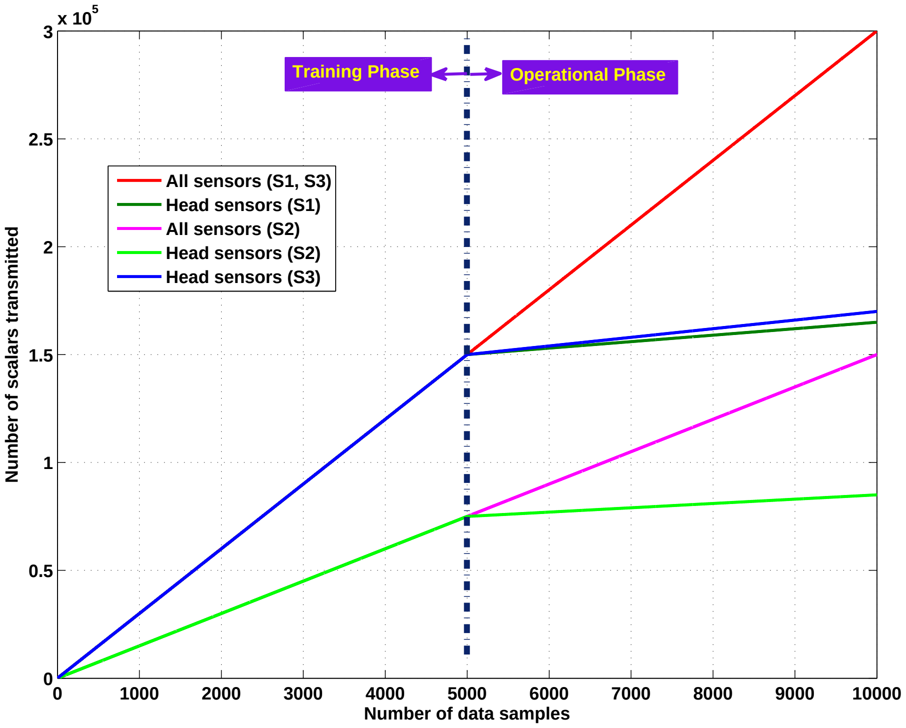

Figure 5.11. Communication cost versus time for the cases where all sensors, or only the head sensors transmit information to the FC.

CHAPTER 6

CLUSTERING OF MULTIMODAL HETEROGENEOUS DATA

6.1 Problem Statement

The ideas of matching different types of sensor measurements that contain information about the same sources can be generalized in settings where there are more than two different types of sensors. Oftentimes a variety of sensors measuring temperature pressure, carbon monoxide, carbon dioxide and so on are employed to get a better view and understanding of the sensed field. Our idea is to start from the multiset canonical correlation analysis (M-CCA) framework, see e.g., [23,40] that is capable to uncover common features from multiple (more than 2) sets of data and introduce the sparsity regularization terms introduced in Sec. 3.

Consider a connected network consisting of $\mathcal{K} \geq 2$ types of sensors, which generate $\mathcal{K}$ different data sets, and M uncorrelated scalar sources, namely $\{s_m(\tau)\}_{m=1}^{M}$, with the number of sources unknown. Let's denote $\mathcal{S}_a^b$ as the bth sensor of the ath type, and denote $\mathcal{S}_a$ as the sensor set for the ath type of sensors with cardinality $|\mathcal{S}_a| = p_a$. Each sensor, say sensor $\mathcal{S}_i^n$, acquires scalar measurements $\{\mathbf{x}_{i,\tau}(n)\}$ during time instances $\tau = 1, \cdots, t$. Each sensor contains the information about one of the M sources. The measurements of the sensor $\mathcal{S}_i^n$, adhere to the following model:

$$\mathbf{x}_{i,n}(\tau) = \sum_{m=1}^{M} h_{m,i,n}(s_m(\tau)) + w_{i,n}(\tau) \tag{6.1}$$

where $h_{m,i,n}(\cdot)$ is a random scalar linear mapping from $\mathbb{R}^1$ to $\mathbb{R}^1$, which equals to zero when sensor $\mathcal{S}_i^n$ is sufficiently far from source $s_m(\tau)$, and $w_{i,n}(\tau)$ denotes zero-mean sensing noise.

Let $\mathbf{x}_i(\tau) := [\mathbf{x}_{i,1}(\tau) - m_{i,1}, \cdots, \mathbf{x}_{i,n}(\tau) - m_{i,n}, \cdots, \mathbf{x}_{i,p_i}(\tau) - m_{i,p_i}]^T$ aggregate the measurements acquired across all the sensors in $\mathcal{S}_i$, where $m_{i,n}$ represents the sample-averaged expectation of $\mathbf{x}_{i,n}(\tau)$, i.e., $m_{i,n} = \frac{1}{t}\sum_{\tau=1}^{t}\mathbf{x}_{i,n}(\tau)$. Given $\mathcal{K}$ different data sets, $\mathbf{x}_1(\tau) \in \mathbb{R}^{p_1 \times 1}$, $\mathbf{x}_2(\tau) \in \mathbb{R}^{p_2 \times 1}$, ...,$\mathbf{x}_{\mathcal{K}}(\tau) \in \mathbb{R}^{p_{\mathcal{K}} \times 1}$ for $\tau = 1, ..., t$, M-CCA is looking for $q \times p_k$ matrices $\mathbf{D}_k$ and $p_k \times q$ matrices $\mathbf{E}_k$ such that the following summation of pairwise estimation errors is minimized, i.e.,

$$(\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}, \{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}) = \text{arg min}_{\mathbf{D}_j,\mathbf{E}_i} \sum_{i=1}^{\mathcal{K}-1} \sum_{j>i}^{\mathcal{K}} t^{-1} \sum_{\tau=1}^{t} \|\mathbf{x}_i(\tau) - \mathbf{E}_i\mathbf{D}_j\mathbf{x}_j(\tau)\|_2^2 \tag{6.2}$$

Unlike the formulation for standard CCA that involves two data sets and it is imposed the canonical variates to be as similar as possible, in the above M-CCA framework, only $\mathbf{D}_j\mathbf{x}_j(\tau)$ is used to estimate the common information present in $\mathbf{x}_i(\tau)$ and $\mathbf{x}_j(\tau)$, corresponding to the source signals and then based on the estimated source signals, matrices $\mathbf{E}_i$ are introduced to recover the sensor measurements $\mathbf{x}_i(\tau)$.

The objective of this chapter is to estimate the number of sources and cluster the sensors according to their source content. Toward this end, we will apply PCA along with moving-average (MA) filtering to determine the number of sources. Then, norm-one regularization will be combined with M-CCA to identify the set of sensors acquiring spatially correlated measurements. Ideally imposed sparsity in matrices $\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}$ and $\{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}$ makes these matrices behave such that each row of $\mathbf{D}_i$ and each column of $\mathbf{E}_i$ can have the same nonzero entry positions which correspond to the entries of the $\mathbf{x}_i(\tau)$ vector that contain information about the same source.

In order to develop the sparse M-CCA algorithm in both centralized and distributed fashions, we will impose the following assumptions in the multi-modality heterogeneous networks: $\mathcal{A}_1$) Each source affects $\mathcal{K}$ types of sensor measurements; $\mathcal{A}_2$) The communication graph for each different type of sensors is connected; and $\mathcal{A}_3$) Each sensor has $\mathcal{K}$ different types of neighboring sensors.

6.2 ℓ_1-Regularized Multiset Canonical Correlations

In order to isolate noninformative entries in $\mathbf{x}_i(\tau)$ and identify the source-informative groups of entries within $\mathbf{x}_i(\tau)$, norm-one is incorporated in the standard M-CCA formulation in (6.2). Proper sparse $\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}$ and $\{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}$ matrices can be obtained through the following sparsity-inducing M-CCA (SM-CCA) formulation:

$$(\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}, \{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}) = \text{arg min}_{\mathbf{D}_j,\mathbf{E}_i} \sum_{i=1}^{\mathcal{K}-1}\sum_{j>i}^{\mathcal{K}} t^{-1} \sum_{\tau=1}^{t} \|\mathbf{x}_i(\tau) - \mathbf{E}_i\mathbf{D}_j\mathbf{x}_j(\tau)\|_2^2 \qquad (6.3)$$
$$+ \sum_{i=1}^{\mathcal{K}-1}\sum_{\rho=1}^{q} \lambda_{i,\rho}^{E}\|\mathbf{E}_i(:,\rho)\|_1 + \sum_{j=2}^{\mathcal{K}}\sum_{\rho=1}^{q} \lambda_{j,\rho}^{D}\|\mathbf{D}_j(\rho,:)\|_1.$$

Note that the number of rows of $\mathbf{D}_j$ and the number of columns of $\mathbf{E}_i$, say q, is set as the estimated number of sources obtained in Sec. 5.2.1. Block coordinate descent techniques will be utilized to derive centralized and distributed approaches tackling the minimization problem of (6.3).

6.2.1 Centralized SM-CCA (CSM-CCA)

We consider a centralized setting where there exists a fusion center collecting all sensor measurements and solving the minimization problem in (6.3). In the beginning, one type of sensor data is applied to the proposed PCA along with the MA filtering scheme proposed in Sec. 5.2.1 to determine the number of sources and the value of q is set to the estimated number of sources.. Using the block coordinate descent solver, the cost in (6.3) is minimized w.r.t. one entry of $\{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}$ (or $\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}$) while keeping the remaining entries of $\{\mathbf{D}_j\}_{j=2}^{\mathcal{K}}$ (or $\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}$) to their most current updates. Let's denote the current coordinate descent cycle as z, and the most up-to-date updates of $\mathbf{D}_j$ and $\mathbf{E}_i$ as $\hat{\mathbf{D}}_j^z$ and $\hat{\mathbf{E}}_i^z$, respectively, for $i = 1, \cdots, \mathcal{K}-1$, $j = 2, \cdots, \mathcal{K}$. Specifically, in the beginning of coordinate cycle z, given the estimates $\{\hat{\mathbf{D}}_j^{z-1}\}_{j=2}^{\mathcal{K}}$ and $\{\hat{\mathbf{E}}_i^{z-1}\}_{i=1}^{\mathcal{K}-1}$, the minimization problem which is used to obtain the current updates $\{\hat{\mathbf{D}}_j^{z}\}_{j=2}^{\mathcal{K}}$ and $\{\hat{\mathbf{E}}_i^{z}\}_{i=1}^{\mathcal{K}-1}$ can be formulated as

$$(\{\hat{\mathbf{E}}_i^z\}_{i=1}^{\mathcal{K}-1}, \{\hat{\mathbf{D}}_j^z\}_{j=2}^{\mathcal{K}}) = \text{arg min}_{\mathbf{D}_j, \mathbf{E}_i} \sum_{i=1}^{\mathcal{K}-1} \sum_{j>i}^{\mathcal{K}} t^{-1} \sum_{\tau=1}^{t} \|\mathbf{x}_i(\tau) - \mathbf{E}_i \mathbf{D}_j \mathbf{x}_j(\tau)\|_2^2 \tag{6.4}$$
$$+ \sum_{i=1}^{\mathcal{K}-1} \sum_{\rho=1}^{q} \lambda_{i,\rho}^E \|\mathbf{E}_i(:,\rho)\|_1 + \sum_{j=2}^{\mathcal{K}} \sum_{\rho=1}^{q} \lambda_{j,\rho}^D \|\mathbf{D}_j(\rho,:)\|_1$$

To facilitate the application of coordinate descent iterations, we rewrite the cost in (6.4) w.r.t. $\mathbf{D}_j$ while fixing $\{\mathbf{E}_i\}_{i=1}^{\mathcal{K}-1}$ and $\{\mathbf{D}_{i'}\}_{i' \neq i}$

$$\hat{\mathbf{D}}_j^z = \text{arg min}_{\mathbf{D}} \sum_{i=1}^{j-1} t^{-1} \sum_{\tau=1}^{t} \|\mathbf{x}_i(\tau) - \hat{\mathbf{E}}_i^{z-1} \mathbf{D}_j \mathbf{x}_j(\tau)\|_2^2 + \sum_{\rho=1}^{q} \lambda_{j,\rho}^D \|\mathbf{D}_j(\rho,:)\|_1. \tag{6.5}$$

Coordinate descent is further applied in (6.5) to split it into qp_i subproblems, each of which corresponds to the minimization problem w.r.t. one entry of $\mathbf{D}_j$, say $\mathbf{D}_j(\alpha,\beta)$. After fixing the remaining $qp_i - 1$ entries to their most up-to-date values, the scalar update $\hat{\mathbf{D}}_j^z(\alpha,\beta)$ can be obtained by minimizing the following cost function:

$$\hat{\mathbf{D}}_j^z(\alpha,\beta) = \text{arg min}_d \sum_{i=1}^{j-1} t^{-1} \sum_{\tau=1}^{t} \|\boldsymbol{\chi}_{i,\tau,\alpha,\beta} - d \cdot \mathbf{E}_{x,\tau,i}(:, (\beta-1)q+\alpha)\|_2^2 + \lambda_{j,\alpha}^D \cdot |d| \tag{6.6}$$

where $\boldsymbol{\chi}_{i,\tau,\alpha,\beta} = \mathbf{x}_i(\tau) - \sum_{\ell=1,\ell\neq(\beta-1)q+\alpha}^{qp_j} \mathbf{d}_v(\ell) \cdot \mathbf{E}_{x,\tau,i}(:,\ell) \in \mathbb{R}^{p_i \times 1}$, and $\mathbf{E}_{x,\tau,i} := \mathbf{x}_j^T(\tau) \otimes \hat{\mathbf{E}}_i^{z-1}$, in which $\mathbf{d}_v := \mathbf{vec}(\hat{\mathbf{D}}_j^{z-1})$ and $\mathbf{vec}$ represents the operator of vectorization, and $\otimes$ denotes Kronecker product.

Let's define $\boldsymbol{\chi}_{\alpha,\beta} \in \mathbb{R}^{(p_1+p_2+\cdots+p_{j-1})t\times 1}$ and $\mathbf{E}_{x,\alpha,\beta} \in \mathbb{R}^{(p_1+p_2+\cdots+p_{j-1})t\times 1}$ as

$$\boldsymbol{\chi}_{\alpha,\beta} := \frac{1}{\sqrt{t}}[\boldsymbol{\chi}_{1,1,\alpha,\beta}^T, \cdots, \boldsymbol{\chi}_{1,t,\alpha,\beta}^T, \cdots, \boldsymbol{\chi}_{j-1,1,\alpha,\beta}^T, \cdots, \boldsymbol{\chi}_{j-1,t,\alpha,\beta}^T]^T \tag{6.7}$$

$$\mathbf{E}_{x,\alpha,\beta} := \frac{1}{\sqrt{t}}[(\mathbf{E}_{x,1,1}(:,(\beta-1)q+\alpha))^T, \cdots, (\mathbf{E}_{x,t,1}(:,(\beta-1)q+\alpha))^T,$$
$$\cdots, (\mathbf{E}_{x,1,j-1}(:,(\beta-1)q+\alpha))^T, \cdots, (\mathbf{E}_{x,t,j-1}(:,(\beta-1)q+\alpha))^T]^T \tag{6.8}$$

After applying (6.7) and (6.8) in the first term of (6.6), the cost in (6.6) can be rewritten as

$$\hat{\mathbf{D}}_j^z(\alpha,\beta) = \text{arg min}_d \|\boldsymbol{\chi}_{\alpha,\beta} - d\cdot \mathbf{E}_{x,\alpha,\beta}\|_2^2 + \lambda_{j,\alpha}^D \cdot |d| \tag{6.9}$$

After observing that (6.9) is a scalar sparse regression problem, it turns out that

$$\hat{\mathbf{D}}_j^z(\alpha,\beta) = \mathbb{F}(\boldsymbol{\chi}_{\alpha,\beta}, \mathbf{E}_{x,\alpha,\beta}, 0, 0, \lambda_{j,\alpha}^D) \text{ for } \alpha = 1,\cdots,q, \beta = 1,\cdots,p_j, j = 2,\cdots,\mathcal{K} \tag{6.10}$$

Next, we update each entry of $\hat{\mathbf{E}}_i^z$, say $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ for $i = 1,\cdots,\mathcal{K}-1$. After simple mathematical manipulations, the cost w.r.t. $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ can be formulated as

$$\hat{\mathbf{E}}_i^z(\beta,\alpha) = \text{arg min}_e \sum_{j>i}^{\mathcal{K}} t^{-1} \sum_{\tau=1}^{t} (\chi_{j,\tau} - e\cdot h_{j,\tau})^2 + \lambda_{i,\alpha}^E |e| \tag{6.11}$$

where $h_{j,\tau} = (\hat{\mathbf{D}}_j^{z-1}\mathbf{x}_j(\tau))(\alpha) \in \mathbb{R}^1$, and $\chi_{j,\tau} = \mathbf{x}_{i,\beta}(\tau) - m_{i,\beta} - \sum_{\ell=1,\ell\neq\alpha}^{q} \hat{\mathbf{E}}_i^{z-1}(\beta,\ell)\cdot(\hat{\mathbf{D}}_j^{z-1}\mathbf{x}_j(\tau))(\ell) \in \mathbb{R}^1$. Similarly, the update $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ can be obtained as

$$\hat{\mathbf{E}}_i^z(\beta,\alpha) = \mathbb{F}(\boldsymbol{\chi}_{i,\beta,\alpha}, \mathbf{h}_{i,\beta,\alpha}, 0, 0, \lambda_{i,\alpha}^E) \tag{6.12}$$

where $\boldsymbol{\chi}_{i,\beta,\alpha} = \frac{1}{\sqrt{t}}[\chi_{i+1,1},\cdots,\chi_{i+1,t},\cdots,\chi_{\mathcal{K},1},\cdots,\chi_{\mathcal{K},t}]^T \in \mathbb{R}^{(\mathcal{K}-i)t\times 1}$, and $\mathbf{h}_{i,\beta,\alpha} = \frac{1}{\sqrt{t}}[h_{i+1,1},\cdots,h_{i+1,t},\cdots,h_{\mathcal{K},1},\cdots,h_{\mathcal{K},t}]^T \in \mathbb{R}^{(\mathcal{K}-i)t\times 1}$. The CSM-CCA algorithm can be summarized as the following four steps:

$\mathbf{Step\ 1}$) Use PCA combined with MA to estimate the number of sources which is assigned to the value of q.

$\mathbf{Step\ 2}$) Initialize $\{\hat{\mathbf{D}}_j^0\}_{j=2}^{\mathcal{K}}$ and $\{\hat{\mathbf{E}}_i^0\}_{i=1}^{\mathcal{K}-1}$ randomly.

$\mathbf{Step\ 3}$) For the zth coordinate descent cycle, update $\hat{\mathbf{D}}_j^z(\alpha,\beta)$ and $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ via (6.10) and (6.12) for $\alpha = 1,\cdot,q$, $\beta = 1,\cdots,p_j$ (or p_i), $j = 2,\cdots,\mathcal{K}$, and $i = 1,\cdots,\mathcal{K}-1$.

$\mathbf{Step\ 4}$) If the CSM-CCA cost reduction in the current descent is larger than a pre-specified threshold go back to $\mathbf{Step\ 3}$), otherwise exit and return $\hat{\mathbf{D}}_j = \hat{\mathbf{D}}_j^z$ and $\hat{\mathbf{E}}_i = \hat{\mathbf{E}}_i^z$ for $j = 2,\cdots,\mathcal{K}$, and $i = 1,\cdots,\mathcal{K}-1$.

6.2.2 Distributed SM-CCA (DSM-CCA)

The SM-CCA scheme is redesigned for a setting where each sensor is able to talk only with its single-hop neighbors. In this distributed setting, sensor $\mathcal{S}_i^n$ will update the submatrices $\mathbf{D}_i(:,n) \in \mathbb{R}^{q\times 1}$ and $\mathbf{E}_i(n,:) \in \mathbb{R}^{1\times q}$ using its available measurements $\mathbf{x}_{i,n}(\tau)$ for $\tau = 1, \cdots, t$. First of all, a distributed PCA approach [66] combined with MA filtering is applied to find the number of sources. Specifically, a framework of locally estimating principal components vectors was proposed in [66], and the number of uncorrelated sources can be obtained by estimating the number of eigenvalues corresponding to source signals and denoted by q. Note that the mean of sensor data used in the distributed PCA is zeroed out after applying MA. Then, ADMM will be combined with BCD to solve the minimization problem of (6.3) in a distributed fashion. The basic theory of DSM-CCA is that sensors $\mathcal{S}_j^\beta$ and $\mathcal{S}_i^\beta$ respectively update (6.10) and (6.12) locally using their own available information.

Let's start from the solution of $\hat{\mathbf{D}}_j^z(\alpha,\beta)$ in (6.10), which will be updated by sensor $\mathcal{S}_j^\beta$ in the coordinate cycle z. Let's define that $\mathbf{x}_i(\tau,\beta) := \mathbf{x}_{i,\beta} - m_{i,\beta}$, then $\mathbf{x}_i(\tau) = [\mathbf{x}_i(\tau,1), \cdots, \mathbf{x}_i(\tau,p_i)]^T$. Notice that, (6.10) involves $\boldsymbol{\chi_{\alpha,\beta}}^T\mathbf{E}_{x,\alpha,\beta}$ and $\|\mathbf{E}_{x,\alpha,\beta}\|_2^2$, where the first term can be equivalently written as

$$\begin{aligned}
&\boldsymbol{\chi}_{\alpha,\beta}^T\mathbf{E}_{x,\alpha,\beta} = \frac{1}{t}\sum_{i=1}^{j-1}\sum_{\tau=1}^{t}\boldsymbol{\chi}_{i,\tau,\alpha,\beta}^T\mathbf{E}_{x,\tau,i}(:,(\beta-1)q+\alpha)\\
&= \frac{1}{t}\sum_{i=1}^{j-1}\sum_{\tau=1}^{t}(\mathbf{x}_i(\tau) - \sum_{\ell=1,\ell\neq(\beta-1)q+\alpha}^{qp_j}\mathbf{d}_v(\ell)\mathbf{E}_{x,\tau,i}(:,\ell))\mathbf{E}_{x,\tau,i}(:,(\beta-1)q+\alpha)\\
&= \frac{1}{t}\sum_{i=1}^{j-1}\sum_{\tau=1}^{t}\mathbf{x}_j(\tau,\beta)(\sum_{\ell=1}^{p_i}\mathbf{x}_i(\tau,\ell)\hat{\mathbf{E}}_i^{z-1}(\ell,\alpha)) + \frac{1}{t}\sum_{i=1}^{j-1}\sum_{\tau=1}^{t}\mathbf{x}_j^2(\tau,\beta)\hat{\mathbf{D}}_j^{z-1}(\alpha,\beta)\boldsymbol{\mathcal{E}}_i(\alpha,\alpha)\\
&- \frac{1}{t}\sum_{i=1}^{j-1}\sum_{\tau=1}^{t}\mathbf{x}_j(\tau,\beta)\cdot[\sum_{n=1}^{p_j}\mathbf{x}_j(\tau,n)\sum_{\rho=1}^{q}(\hat{\mathbf{D}}_j^{z-1}(\rho,n)\boldsymbol{\mathcal{E}}_i(\rho,\alpha))] \qquad (6.13)
\end{aligned}$$

where the α_1th row and α_2th column of matrix $\boldsymbol{\mathcal{E}}_i \in \mathbb{R}^{q\times q}$, saying $\boldsymbol{\mathcal{E}}_i(\alpha_1,\alpha_2)$ is $(\hat{\mathbf{E}}_i^{z-1}(:,\alpha_1))^T\cdot(\hat{\mathbf{E}}_i^{z-1}(:,\alpha_2))$ for $\alpha_1,\alpha_2 = 1,\cdots,q$, revealing that each entry of $\boldsymbol{\mathcal{E}}_i$ is a global value for the subnetwork consisting of sensors in the set of $\mathcal{S}_i$. Notice that, $\sum_{\ell=1}^{p_i}\mathbf{x}_i(\tau,\ell)\hat{\mathbf{E}}_i^{z-1}(\ell,\alpha)$

is also a global value across the subnetwork comprising of sensors in $\mathcal{S}_i$. After sensor $\mathcal{S}_j^{\beta}$ receiving $\boldsymbol{\mathcal{E}}_i(\rho, \alpha)$ from its neighboring sensors that belong to $\mathcal{S}_i$, $\sum_{n=1}^{p_j} \mathbf{x}_j(\tau, n) \sum_{\rho=1}^{q} (\hat{\mathbf{D}}_j^{z-1}$ (ρ, n) $\boldsymbol{\mathcal{E}}_i(\rho, \alpha))$ is the summation across all the sensors in $\mathcal{S}_j$. Thus, in order to obtain (6.13) purely relying on sensor $\mathcal{S}_j^{\beta}$'s accessible values, ADMM is applied to find all the concerned global terms in (6.13). In detail, each sensor in $\mathcal{S}_i$ runs K ADMM iterations to locally estimate the global scalars $\sum_{\ell=1}^{p_i} \mathbf{x}_i(\tau, \ell)\hat{\mathbf{E}}_i^{z-1}(\ell, \alpha)$ and $\boldsymbol{\mathcal{E}}_i(\rho, \alpha))$, whose estimated values from sensor $\mathcal{S}_i^{n_j^{\beta}}$ are denoted by $u_{i,\alpha}^{n_j^{\beta}}$ and $u_{i,\rho,\alpha}^{n_j^{\beta}}$, respectively, for $\tau = 1, \cdots, t,$, $i = 1, \cdots, j-1$, $\rho = 1, \cdots, q$. Let's define one of sensor $\mathcal{S}_i^{\beta}$'s neighbors in $\mathcal{S}_j$ as $\mathcal{S}_j^{n_i^{\beta}}$, and we define one of the neighboring sensors of sensor $\mathcal{S}_j^{\beta}$ in $\mathcal{S}_i$ as sensor $\mathcal{S}_i^{n_j^{\beta}}$, where $n_i^{\beta} \in \{1, \cdots, p_j\}$ and $p_j^{\beta} \in \{1, \cdots, p_i\}$. Thus, sensor $\mathcal{S}_j^{\beta}$ can communicate with sensor $\mathcal{S}_i^{n_j^{\beta}}$. Next, sensor $\mathcal{S}_i^{n_j^{\beta}}$ transmits $u_{i,\alpha}^{n_j^{\beta}}$ and $u_{i,\rho,\alpha}^{n_j^{\beta}}$ to sensor $\mathcal{S}_j^{\beta}$. Then, K ADMM iterations will be run by every sensor in $\mathcal{S}_j$, estimating $\sum_{n=1}^{p_j} \mathbf{x}_j(\tau, n) \sum_{\rho=1}^{q} (\hat{\mathbf{D}}_j^{z-1}(\rho, n) u_{i,\rho,\alpha}^{n_j^{\beta}})$ in a distributed way, and let's denote sensor $\mathcal{S}_j^{\beta}$'s estimate as $\bar{u}_{j,\alpha}^{\beta}$. Finally, sensor $\mathcal{S}_j^{\beta}$ is able to attain (6.13) only through communicating with its neighbors, which can be expressed as

$$\begin{aligned}\boldsymbol{\chi}_{\alpha,\beta}^T \mathbf{E}_{x,\alpha,\beta} =& \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} \mathbf{x}_j(\tau, \beta) u_{i,\alpha}^{n_j^{\beta}} + \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} \mathbf{x}_j^2(\tau, \beta) \hat{\mathbf{D}}_j^{z-1}(\alpha, \beta) u_{i,\alpha,\alpha}^{n_j^{\beta}} \\ & - \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} \mathbf{x}_j(\tau, \beta) \bar{u}_{j,\alpha}^{\beta}\end{aligned} \qquad (6.14)$$

Another term associated with (6.10) is $||\mathbf{E}_{x,\alpha,\beta}||_2^2$, which equals to

$$||\mathbf{E}_{x,\alpha,\beta}||_2^2 = \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} ||\mathbf{E}_{x,\tau,i}(:, (\beta-1)q + \alpha)||_2^2 = \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} \mathbf{x}_j^2(\tau, \beta) \cdot \boldsymbol{\mathcal{E}}_i(\alpha, \alpha) \quad (6.15)$$

Recall that sensor $\mathcal{S}_j^{\beta}$ receives the estimate of $\boldsymbol{\mathcal{E}}_i(\alpha, \alpha)$, saying $u_{i,\alpha,\alpha}^{n_j^{\beta}}$, so sensor $\mathcal{S}_j^{\beta}$ is able to get (6.15) by calculating the following term

$$||\mathbf{E}_{x,\alpha,\beta}||_2^2 = \frac{1}{t} \sum_{i=1}^{j-1} \sum_{\tau=1}^{t} \mathbf{x}_j^2(\tau, \beta) \cdot u_{i,\alpha,\alpha}^{n_j^{\beta}} \qquad (6.16)$$

According to (6.14) and (6.16), sensor $\mathcal{S}_j^\beta$ is capable of updating $\hat{\mathbf{D}}_j^z(\alpha,\beta)$. Thus, each entry of $\hat{\mathbf{D}}_j^z$ is going to be locally updated by its corresponding sensor. And it follows readily that, sensor $\mathcal{S}_j^\beta$ updates $\hat{\mathbf{D}}_j^z(\alpha,\beta)$ as

$$\hat{\mathbf{D}}_j^z(\alpha,\beta) = \mathbf{sgn}(\boldsymbol{\chi}_{\alpha,\beta}^T \mathbf{E}_{x,\alpha,\beta}) \times (\mathbf{max}(0, (|\frac{\boldsymbol{\chi}_{\alpha,\beta}^T \mathbf{E}_{x,\alpha,\beta}}{\|\mathbf{E}_{x,\alpha,\beta}\|_2^2}| - \frac{\lambda_{j,\alpha}^D}{2\|\mathbf{E}_{x,\alpha,\beta}\|_2^2}))) \tag{6.17}$$

where $\boldsymbol{\chi}_{\alpha,\beta}^T \mathbf{E}_{x,\alpha,\beta}$ and $\|\mathbf{E}_{x,\alpha,\beta}\|_2^2$ are obtained from (6.14) and (6.16), respectively.

Similarly, every entry of $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ can be updated by sensor $\mathcal{S}_i^\beta$ in a distributed fashion. From the solution of $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ in (6.12), it can be seen that sensor $\mathcal{S}_i^\beta$ needs locally calculate $\boldsymbol{\chi}_{i,\beta,\alpha}^T \mathbf{h}_{i,\beta,\alpha}$ as well as $\|\mathbf{h}_{i,\beta,\alpha}\|_2^2$, which can be written as

$$\begin{aligned}\boldsymbol{\chi}_{i,\beta,\alpha}^T \mathbf{h}_{i,\beta,\alpha} =& \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t} \mathbf{x}_i(\tau,\beta)\hat{\mathbf{D}}_j^{z-1}(\alpha,:)\mathbf{x}_j(\tau) \\ &- \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t} \hat{\mathbf{D}}_j^{z-1}(\alpha,:)\mathbf{x}_j(\tau) \sum_{\ell=1,\ell\neq\alpha}^{q} \hat{\mathbf{E}}_i(\beta,\ell)\hat{\mathbf{D}}_j^{z-1}(\ell,:)\mathbf{x}_j(\tau)\end{aligned} \tag{6.18}$$

and

$$\|\mathbf{h}_{i,\beta,\alpha}\|_2^2 = \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t} (\hat{\mathbf{D}}_j^{z-1}(\alpha,:)\mathbf{x}_j(\tau))^2 \tag{6.19}$$

where the global quantities $\hat{\mathbf{D}}_j^{z-1}(\ell,:)\mathbf{x}_j(\tau)$ for $\ell = 1, \cdots, q$, will be estimated by sensors in $\mathcal{S}_j$ after employing ADMM technology. And let's denote the estimation of $\hat{\mathbf{D}}_j^{z-1}(\ell,:)$ $\mathbf{x}_j(\tau)$ by sensor $\mathcal{S}_j^{n_i^\beta}$ as $\hat{u}_{j,\ell}^{n_i^\beta}$, where sensor $\mathcal{S}_j^{n_i^\beta}$ is the neighbor of sensor $\mathcal{S}_i^\beta$, which means that the estimation $\hat{u}_{j,\ell}^{n_i^\beta}$ is available for sensor $\mathcal{S}_i^\beta$. Toward this end, the two terms in (6.18) and (6.19) which are necessary in updating $\hat{\mathbf{E}}_i^z(\beta,\alpha)$, can be obtained by the following equations

$$\boldsymbol{\chi}_{i,\beta,\alpha}^T \mathbf{h}_{i,\beta,\alpha} = \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t} \mathbf{x}_i(\tau,\beta)\hat{\mathbf{D}}_j^{z-1}(\alpha,:)\mathbf{x}_j(\tau) \tag{6.20}$$

$$- \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t} \hat{\mathbf{D}}_j^{z-1}(\alpha,:)\mathbf{x}_j(\tau) \sum_{\ell=1,\ell\neq\alpha}^{q} \hat{\mathbf{E}}_i(\beta,\ell)\hat{u}_{j,\ell}^{n_i^\beta} \tag{6.21}$$

$$\|\mathbf{h}_{i,\beta,\alpha}\|_2^2 = \frac{1}{t}\sum_{j=i+1}^{\mathcal{K}}\sum_{\tau=1}^{t}(\hat{u}_{j,\ell}^{n_i^\beta})^2. \tag{6.22}$$

Meanwhile, using the results in (6.20) and (6.22), sensor $\mathcal{S}_i^\beta$ can update $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ as

$$\hat{\mathbf{E}}_i^z(\beta,\alpha) = \mathbf{sgn}(\boldsymbol{\chi}_{i,\beta,\alpha}^T\mathbf{h}_{i,\beta,\alpha}) \times \mathbf{max}(0,(|\frac{\boldsymbol{\chi}_{i,\beta,\alpha}^T\mathbf{h}_{i,\beta,\alpha}}{\|\mathbf{h}_{i,\beta,\alpha}\|_2^2}| - \frac{\lambda_{i,\alpha}^E}{2\|\mathbf{h}_{i,\beta,\alpha}\|_2^2})) \tag{6.23}$$

The DSM-CCA scheme is summarized as the following six steps:

$\mathbf{Step\ 1}$) Set q as estimated number of sources using distributed PCA [66] along with MA.

$\mathbf{Step\ 2}$) Initialize $\hat{\mathbf{D}}_j^0$ and $\hat{\mathbf{E}}_i^0$ with $q \times p_j$ and $p_i \times q$ matrices randomly, for $i = 1, \cdots, \mathcal{K}-1$, $j = 2, \cdots, \mathcal{K}$.

$\mathbf{Step\ 3}$) In cycle z, sensor $\mathcal{S}_i^{\bar{i}}$ forms estimates $u_{i,\alpha}^{\bar{i}}$, and $u_{i,\rho,\alpha}^{\bar{i}}$, via K ADMM updating recursions for $i = 1, \cdots, \mathcal{K} - 1$, $\alpha, \rho = 1, \cdots, q$, and $\bar{i} = 1 \cdots, p_i$. Sensor $\mathcal{S}_j^{\bar{j}}$ forms estimates $\bar{u}_{j,\alpha}^{\bar{j}}$ and $\hat{u}_{j,\ell}^{\bar{j}}$ via K ADMM iterations, for $j = 2, \cdots, \mathcal{K}$, $\alpha, \ell = 1, \cdots, q$, and $\bar{j} = 1, \cdots, p_j$.

$\mathbf{Step\ 4}$) Update $\hat{\mathbf{D}}_j^z(\alpha,\beta)$ via (6.17), for $j = 2, \cdots, \mathcal{K}$, $\alpha = 1, \cdots, q$, and $\beta = 1, \cdots, p_j$.

$\mathbf{Step\ 5}$) Update $\hat{\mathbf{E}}_i^z(\beta,\alpha)$ via (6.23), for $i = 1, \cdots, \mathcal{K} - 1$, $\alpha = 1, \cdots, q$, and $\beta = 1, \cdots, p_i$.

$\mathbf{Step\ 6}$) If the DSM-CCA cost reduction in current coordinate cycle drops below a desired tolerance, exit and return $\hat{\mathbf{D}}_j = \hat{\mathbf{D}}_j^z$ and $\hat{\mathbf{E}}_i = \hat{\mathbf{E}}_i^z$, otherwise go back to Step 3).

6.3 Simulation Results

The performance of the CSM-CCA and DSM-CCA is tested and compared with iK-Means [13] in terms of probability of correctly clustering sensor measurements according to their source information. The algorithms are tested in a sensor network consisting of $\mathcal{K} = 3$ types of sensors (15 sensors in each type) and $M = 2$ uncorrelated sources. The AR-1 model is used for the sources $\{s_m(\tau)\}_{m=1}^M$. In the testing scenario, sensors $\{\mathcal{S}_1^1, \mathcal{S}_1^2, \mathcal{S}_1^3, \mathcal{S}_1^6, \mathcal{S}_1^7, \mathcal{S}_2^1, \mathcal{S}_2^3, \mathcal{S}_2^5, \mathcal{S}_2^7, \mathcal{S}_2^9, \mathcal{S}_3^1, \mathcal{S}_3^2, \mathcal{S}_3^3, \mathcal{S}_3^4, \mathcal{S}_3^5\}$ observe source $s_1(\tau)$, source $s_2(\tau)$

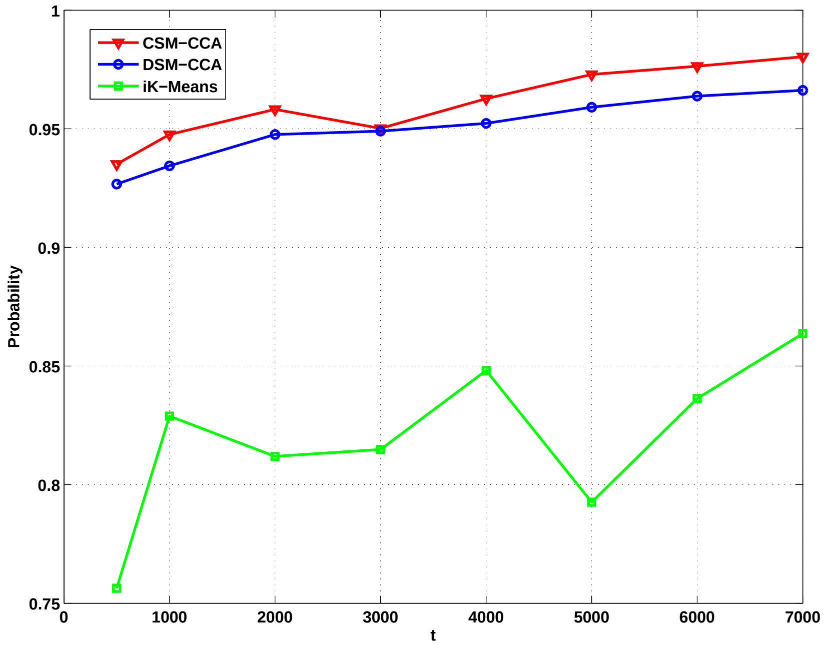

Figure 6.1. Probability of correctly clustering sensors vs. number of training data.

affects sensors $\{\mathcal{S}_1^4, \mathcal{S}_1^5, \mathcal{S}_1^8, \mathcal{S}_1^9, \mathcal{S}_1^{10}, \mathcal{S}_2^2, \mathcal{S}_2^4, \mathcal{S}_2^6, \mathcal{S}_2^8, \mathcal{S}_2^{10}, \mathcal{S}_3^6, \mathcal{S}_3^7, \mathcal{S}_3^8, \mathcal{S}_3^9, \mathcal{S}_3^{10}\}$, and the remaining 15 sensors are too far away from any of the sources to sense the sources. In the DSM-CCA algorithm, the ADMM iteration K is set to be $K = 20$. The sparsity-controlling coefficients are chosen using the $\lambda-$selection scheme proposed in 3.3. Fig. 6.1 depicts that CSM-CCA achieves the best performance, and DSM-CCA yields better performance than iK-Means. It is also of interest to notice that, as the number of training data t increases the higher probability of correctly clutering sensors for CSM-CCA and DSM-CCA also increases.

6.4 Conclusion

A novel ℓ_1-regularized M-CCA framework was put forth in this chapter and utilized to cluster different types of sensor data based on their source content. With the consideration of different kinds of network setting, FC-based network and ad-hoc network, CSM-CCA and DSM-CCA were proposed, respectively, which were translated to be minimization problems while the associated matrices in the formulation were applied to perform the clustering task. The numerical tests demonstrate that our proposed algorithm has the ability to correctly cluster multi-modalities of sensors.

CHAPTER 7

CONCLUDING REMARKS AND FUTURE RESEARCH

7.1 Conclusions

In this dissertation, the core objective is to cluster the sensor measurements based on their source content. Several different scenarios are considered: 1) the number of uncorrelated sources is given, 2) the number of sources is unknown, 3) there is a FC exists in the field, 4) the sensors are connected in an ad hoc way, 5) all the sensors sense the same kind of elements (homogeneous network), 6) multiple types of sensors comprising a heterogeneous network, 6) the FC and (or) sensors have limited capability of storing data (in an online setting), and 8) the sensors and (or) FC is capable of storing all the historical measurements (in a batch setting). For the sake of performing the sensor data clustering under different scenarios, different CCA-based frameworks are proposed. Specifically, CS-CCA and DS-CCA are designed to cope with the setting where the number of sources is known, while the former and latter focused on the case of 3) and 4), respectively. Further, both CS-CCA and DS-CCA are developed in an online fashion, which are denoted by OCS-CCA and ODS-CCA algorithms. Without knowing the number of sources, two novel solvers are provided, which are norm-two regularized algorithm and PCA based scheme. In Chapter 4, we apply the first solver in sparse CCA, and in Chapter 6, PCA combined with MA is utilized to carry out the clustering task. Different from the data models in Chapter 3 and Chapter 5, which are generated from the homogeneous networks, in Chapter 4 and Chapter 6, we deal with the clustering of multiple types of data sets. The numerical tests showed the capability of our proposed algorithms in grouping sensors according the sensors' source information in different scenarios, and proved that our algorithms surpassed the existing alternatives.

Two key techniques are utilized to solve the derived minimization frameworks, which are BCD and ADMM. The contribution of BCD is providing tractable solution to nonconvex minimization problems. Specifically, through the application of BCD, the matrices used to perform sensor clustering can be updated entry-by-entry in each coordinate cycle. It has been shown that as the coordinate cycle goes to infinity, the aforementioned matrices can behave perfectly in the sense that zero and nonzero entries will appear in the proper locations. ADMM played an important role in two aspects: 1) all the proposed distributed algorithms are associated with finding some global variables in a distributed fashion, which can be easily estimated by ADMM iterations; and 2) in the CR-CCA and DR-CCA schemes, the introduced norm-two terms make the cost function to be challenging, which was overcome by ADMM technique.

7.2 Future Research

The proposed CSM-CCA and DSM-CCA frameworks are batch algorithms in the sense that first acquire data and then perform the processing. Such batch schemes are pertinent for settings where sensors acquire data for some limited time and then stop. However, when sensors are constantly sensing new data, a batch algorithm will eventually drain all storing and computational capabilities across sensors. Furthermore, sensor data oftentimes are collected in challenging environments, whose statistical structure is not known and maybe dynamically changing with time. Obviously, when the phenomenon of interest are mobile or exhibit nonstationarity, our proposed batch algorithms will not work well in grouping sensors. To cope with these challenges, we will propose adaptive mechanisms, which can give more emphasis to the recent data and gradually forget the past, and also this kind of adaptive processing should have the ability to tackle the problems with limited number of data, which will not grow up dramatically with the time increases.

Moreover, the CS-CCA, DS-CCA, OCS-CCA, ODS-CCA, CR-CCA, DR-CCA, CSM-CCA and DSM-CCA schemes derived in this thesis are tested using synthetic data. In the future, those algorithms will be verified using real data.

In chapter 5, we proposed a new method based on CS-CCA, NLMS adaptive filter as well as PCA techniques, while in the training phase, every processing is done in a FC, which may introduce a heavy burden when the monitored field is very huge with extremely high number of sensors. Thus, it is necessary to fulfill the flawless clustering in a more efficient way, i.e., using a distributed scheme. Also, more detailed procedure will be proposed to deal with the time-varying environment, i.e, moving sources, the disappearing or appearing of sensors.

APPENDIX A

PROOF OF PROPOSITION 2

The proof consists of two parts: (I) It is shown that the updates $\check{\mathbf{D}}^{\tau}(\alpha,\beta)$ and $\check{\mathbf{E}}^{\tau}(\alpha,\beta)$ by applying BCD to the CCA-based cost (3.5) converge to a stationary point of the cost in (3.5); (II) It is demonstrated that the algorithmic updates obtained from applying BCD to the approximated cost in (7), namely $\hat{\mathbf{D}}^{\tau}(\alpha,\beta)$ and $\hat{\mathbf{E}}^{\tau}(\alpha,\beta)$, are arbitrarily close to the updates $\check{\mathbf{D}}^{\tau}(\alpha,\beta)$ and $\check{\mathbf{E}}^{\tau}(\alpha,\beta)$ obtained from the original CCA-based cost in (3.5), i.e., $|\hat{\mathbf{D}}^{\tau}(\alpha,\beta)-\check{\mathbf{D}}^{\tau}(\alpha,\beta)|\leq\delta(\varepsilon)$, where $\delta(\varepsilon)$ is a nonnegative quantity for which $\lim_{\varepsilon\to 0}\delta(\varepsilon)=0$. In detail:

A.1 Step I

First, minimize of the cost in (3.5) w.r.t. one entry of $\mathbf{D}$ (or $\mathbf{E}$), namely $\mathbf{D}(\alpha,\beta)$ (or $\mathbf{E}(\alpha,\beta)$), without the approximation introduced in (3.6). Let $\check{\mathbf{D}}^{\tau}$ and $\check{\mathbf{E}}^{\tau}$ denote the corresponding updates from entry-wise minimization of (3.5) . During the τth BCD cycle, the minimization of (3.5) w.r.t. $\mathbf{D}(\alpha,\beta)$ involves the cost

$$\begin{aligned}J^{\tau}_{\alpha,\beta}(d) &= \|\check{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}-d\check{\mathbf{h}}^{\tau}_{\alpha,\beta}\|_2^2+\lambda_{D,\alpha}|d|+\\ &\varepsilon[\|\boldsymbol{\psi}^{1,\tau}_{\alpha,\beta}-d\mathbf{h}^{1,\tau}_{\alpha,\beta}\|_2^2+(d^2\hat{\boldsymbol{\Sigma}}_x(\beta,\beta)+d\cdot 2h^{2,\tau}_{\alpha,\beta}+h^{3,\tau}_{\alpha,\beta})^2]\end{aligned} \tag{A.1}$$

where $\check{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}$ and $\check{\mathbf{h}}^{\tau}_{\alpha,\beta}$ can be obtained as in (3.10) after replacing the $\hat{\mathbf{D}}^{\tau-1}$ and $\hat{\mathbf{E}}^{\tau-1}$ updates with $\check{\mathbf{D}}^{\tau-1}$ and $\check{\mathbf{E}}^{\tau-1}$ which will be obtained via solving (A.1). Further,

$$\begin{aligned}h^{2,\tau}_{\alpha,\beta} &:= \textstyle\sum_{j=1,j\neq\beta}^{pf}\check{\mathbf{D}}^{\tau-1}(\alpha,j)\hat{\boldsymbol{\Sigma}}_x(j,\beta),\\ h^{3,\tau}_{\alpha,\beta} &:= \sum_{i,j=1,i,j\neq\beta}^{pf}\check{\mathbf{D}}^{\tau-1}(\alpha,j)\check{\mathbf{D}}^{\tau-1}(\alpha,i)\hat{\boldsymbol{\Sigma}}_x(j,i)-1\\ \boldsymbol{\psi}^{1,\tau}_{\alpha,\beta} &:= -\sum_{i,j=1,j\neq\beta}^{pf}\check{\mathbf{D}}^{\tau-1}(\alpha,j)\hat{\boldsymbol{\Sigma}}_x(j,i)\check{\mathbf{D}}^{\tau-1}(\mathcal{I}_\alpha,i)\\ \mathbf{h}^{1,\tau}_{\alpha,\beta} &:= -\sum_{i=1}^{pf}\hat{\boldsymbol{\Sigma}}_x(\beta,i)\check{\mathbf{D}}^{\tau-1}(\mathcal{I}_\alpha,i)\end{aligned} \tag{A.2}$$

where $\mathcal{I}_\alpha$ is an index set equal to $\{1, 2, \ldots, \alpha - 1, \alpha + 1, \ldots, q\}$ and $\check{\mathbf{D}}^{\tau-1}(\mathcal{I}_\alpha, i) := [\check{\mathbf{D}}^{\tau-1}(1, i) \ldots \check{\mathbf{D}}^{\tau-1}(\alpha - 1, i), \check{\mathbf{D}}^{\tau-1}(\alpha + 1, i) \ldots \check{\mathbf{D}}^{\tau-1}(q, i)]^T$ corresponds to a $q - 1 \times 1$ column vector. Let $\check{\mathbf{D}}^\tau(\alpha, \beta)$ denote the minimizer of (A.1) which is not available in closed form. This is to be contrasted with our algorithm, where instead of considering (A.1) we tackle (7) [or equivalently (9) and (10)] after making the approximation $\varepsilon\|\mathbf{D}\hat{\mathbf{\Sigma}}_x\mathbf{D}^T - \mathbf{I}\|_F^2$ with $\varepsilon\|\mathbf{D}\hat{\mathbf{\Sigma}}_x(\hat{\mathbf{D}}^{\tau-1})^T - \mathbf{I}\|_F^2$ (similary for the $\mathbf{E}$ term) leading to the simple closed-form updates (3.13) and (3.15).

Next, we are going to prove that the iterates acquired from (A.1) are convergent to a stationary point of the CCA-based cost in (3.5). Let $g(\{\mathbf{D}(\alpha, \beta), \mathbf{E}(\alpha, \beta)\}_{\alpha=1,\beta=1}^{q,pf})$ denote the S-CCA cost given in (3.5), which is defined over $\mathbb{R}^{2qpf \times 1}$ and

$$g_0(\{\mathbf{D}(\alpha, \beta), \mathbf{E}(\alpha, \beta)\}_{\alpha=1,\beta=1}^{q,pf}) := v\|\mathbf{E}\hat{\mathbf{\Sigma}}_y\mathbf{E}^T - \mathbf{I}\|_F^2 \\ + \varepsilon\|\mathbf{D}\hat{\mathbf{\Sigma}}_x\mathbf{D}^T - \mathbf{I}\|_F^2 + N^{-1}\textstyle\sum_{t=0}^{N-1} \|\mathbf{E}\mathbf{y}(t) - \mathbf{D}\mathbf{x}(t) - \hat{\mathbf{m}}\|_2^2. \tag{A.3}$$

Furthermore, let's consider the level set

$$\mathfrak{C}^0 := \{\{\mathbf{D}(\alpha, \beta), \mathbf{E}(\alpha, \beta)\}_{\alpha=1,\beta=1}^{q,pf} : \\ g(\{\mathbf{D}(\alpha, \beta), \mathbf{E}(\alpha, \beta)\}_{\alpha=1,\beta=1}^{q,pf}) \leq g(\hat{\mathbf{D}}^0, \hat{\mathbf{E}}^0)\} \tag{A.4}$$

where $\hat{\mathbf{D}}^0$ and $\hat{\mathbf{E}}^0$ are the $q \times pf$ matrices used to initialize S-CCA and selected such that $\|\hat{\mathbf{D}}^0\|_1 < \infty$, $\|\hat{\mathbf{E}}^0\|_1 < \infty$, from which it follows that $g(\hat{\mathbf{D}}^0, \hat{\mathbf{E}}^0) < \infty$. Further, from the form of $g(\cdot)$ it follows that any $\mathbf{D}$ and $\mathbf{E}$ in $\mathfrak{C}^0$ satisfy

$$\sum\nolimits_{\rho=1}^{q} \lambda_{E,\rho}\|\mathbf{E}_{\rho:}^T\|_1 + \textstyle\sum_{\rho=1}^{q} \lambda_{D,\rho}\|\mathbf{D}_{\rho:}^T\|_1 \leq g(\hat{\mathbf{D}}^0, \hat{\mathbf{E}}^0) < \infty.$$

Thus, (i) the set $\mathfrak{C}^0$ is compact (closed and bounded). Moreover, (ii) $g(\cdot)$ is continuous on $\mathfrak{C}^0$.

The cost in (A.1) used to obtain $\check{\mathbf{D}}^\tau(\alpha, \beta)$ can be written as

$$J_{\alpha,\beta}^\tau(d) = e_4 d^4 + e_3 d^3 + e_2 d^2 + e_1 d + \lambda_\alpha |d| + e_0 \tag{A.5}$$

where e_4, e_3, e_2, e_1 and e_0 depend on the quantities defined in (A.2). Notice that $e_4 > 0$ as the diagonal values of $\hat{\boldsymbol{\Sigma}}_x$ (or $\hat{\boldsymbol{\Sigma}}_y$) are positive, and $e_2 \geq 0$. If $e_1 = 0$, and $e_3 = 0$, then $J^{\tau}_{\alpha,\beta}(d)$ is symmetric around zero. In this case, as $e_2 \geq 0$, zero is the unique minimizer of $J^{\tau}_{\alpha,\beta}(d)$. If $e_1 \neq 0$ or $e_3 \neq 0$, there will be either one minimizer or two minimizers, where we can consistently select the larger (or smaller) minimizer. Thus, (iii) we can always ensure the uniquiness of minimizer per iteration. Also, (iv) function $g(\cdot)$ is regular at the unique minimizer, which is outlined in [71, (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$(g_0) = (-\infty, +\infty)^{2qpf \times 1}$ is an open set. Moreover, $g_0(\cdot)$ is G$\hat{a}$uteaux-differential over domain(g_0). In detail, the G$\hat{a}$uteaux derivative is defined as

$$g_0^{'}(\mathbf{H}; \boldsymbol{\Delta}_H) := \lim_{\epsilon \to 0}[g_0(\mathbf{H} + \epsilon\boldsymbol{\Delta}_H) - g_0(\mathbf{H})]/\epsilon \tag{A.6}$$

where $\mathbf{H}$ refers to either $\mathbf{D}$ or $\mathbf{E}$. After carrying out the necessary algebraic manipulations, it follows readily that $g_0^{'}(\mathbf{D}; \boldsymbol{\Delta}_D)$ (and $g_0^{'}(\mathbf{E}; \boldsymbol{\Delta}_E)$) exists for all $\boldsymbol{\Delta}_D$ (and $\boldsymbol{\Delta}_E$) $\in$domain(g_0), and it is equal to (similarly for $\mathbf{E}$)

$$\text{tr}[2(\mathbf{D}\mathbf{X}^{'} - \mathbf{E}\mathbf{Y}^{'})(\boldsymbol{\Delta_D}\mathbf{X}^{'})^T + 2\varepsilon(\mathbf{D}\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T - \mathbf{I})(\mathbf{D}\hat{\boldsymbol{\Sigma}}_x\boldsymbol{\Delta}_D^T - \boldsymbol{\Delta}_D\hat{\boldsymbol{\Sigma}}_x\mathbf{D}^T)] \tag{A.7}$$

whre $\mathbf{X}^{'} = 1/\sqrt{N}[\mathbf{x}(0) - \hat{\mathbf{m}}_x, ..., \mathbf{x}(N-1) - \hat{\mathbf{m}}_x]$ and $\mathbf{Y}^{'} = 1/\sqrt{N}[\mathbf{y}(0) - \hat{\mathbf{m}}_y, ..., \mathbf{y}(N-1) - \hat{\mathbf{m}}_y]$.

The properties (i), (ii), (iii) and (iv) ensure the iterates $\check{\mathbf{D}}^{\tau}(\alpha, \beta)$ and $\check{\mathbf{E}}^{\tau}(\alpha, \beta)$ will converge to a stationary point of $g(\cdot)$ [71, Thm. 4.1 (c)].

A.2 Step II

We demonstrate that the updates from (A.1), namely $\check{\mathbf{D}}^{\tau}(\alpha, \beta)$, can be arbitrarily close to the updates involved in the proposed algorithm in (14), i.e., $|\hat{\mathbf{D}}^{\tau}(\alpha, \beta) - \check{\mathbf{D}}^{\tau}(\alpha, \beta)| \leq \delta(\varepsilon)$, where $\delta(\varepsilon)$ is a nonnegative quantity for which $\lim_{\varepsilon \to 0}\delta(\varepsilon) = 0$.

Both updates $\hat{\mathbf{D}}^0(\alpha, \beta)$ and $\check{\mathbf{D}}^0(\alpha, \beta)$ can be initialized at same value. Assume now that at BCD iteration $\tau - 1$ it holds that $|\hat{\mathbf{D}}^{\tau-1}(\alpha, \beta) - \check{\mathbf{D}}^{\tau-1}(\alpha, \beta)| \leq \delta'(\epsilon)$ and $|\hat{\mathbf{E}}^{\tau-1}(\alpha, \beta) - \check{\mathbf{E}}^{\tau-1}(\alpha, \beta)| \leq \delta'(\epsilon)$, where $\lim_{\epsilon \to 0} \delta'(\epsilon) = 0$. Then, using the cost in (A.1), (13) can be written as

$$\begin{aligned} &\|\boldsymbol{\psi}^{\tau}_{\alpha,\beta} - d\mathbf{h}^{\tau}_{\alpha,\beta}\|_2^2 + \lambda_{D,\alpha}|d| + \|\breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} - d\breve{\mathbf{h}}^{\tau}_{\alpha,\beta}\|_2^2 \\ &= J^{\tau}_{\alpha,\beta}(d) + \phi(d, \varepsilon) \end{aligned} \tag{A.8}$$

where $\breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta}$ and $\breve{\mathbf{h}}^{\tau}_{\alpha,\beta}$ are defined in (3.11), and $\phi(d, \varepsilon) := \|\breve{\boldsymbol{\psi}}^{\tau}_{\alpha,\beta} - d\breve{\mathbf{h}}^{\tau}_{\alpha,\beta}\|_2^2 - \varepsilon[\|\boldsymbol{\psi}^{1,\tau}_{\alpha,\beta} - d\mathbf{h}^{1,\tau}_{\alpha,\beta}\|_2^2 - (d^2\hat{\boldsymbol{\Sigma}}_x(\beta, \beta) + d \cdot 2h^{2,\tau}_{\alpha,\beta} + h^{3,\tau}_{\alpha,\beta})^2] + \delta'(\epsilon) \cdot \Delta_\tau$, where Δ_τ is a finite coefficient. The continuity argument in [26, page 15] and (A.8) implies that, for any $\delta(\varepsilon)$, we can select ε sufficiently small such that $|\hat{\mathbf{D}}^{\tau}(\alpha, \beta) - \check{\mathbf{D}}^{\tau}(\alpha, \beta)| \leq \delta(\varepsilon)$, where $\lim_{\varepsilon \to 0}\delta(\varepsilon) = 0$. Thus, by induction it follows that $|\hat{\mathbf{D}}^{\tau}(\alpha, \beta) - \check{\mathbf{D}}^{\tau}(\alpha, \beta)| \leq \delta(\varepsilon)$ and $|\hat{\mathbf{E}}^{\tau}(\alpha, \beta) - \check{\mathbf{E}}^{\tau}(\alpha, \beta)| \leq \delta(\varepsilon)$ for any τ.

Thus, the updates $\hat{\mathbf{D}}^{\tau}(\alpha, \beta)$ (and $\hat{\mathbf{E}}^{\tau}(\alpha, \beta)$) in (14)-(15) will be $\delta(\varepsilon)$-close to a stationary point of the cost in (3.5) as iteration index $\tau \to \infty$.

APPENDIX B

DERIVATION OF ADMM RECURSIONS IN (3.22)

We introduce auxiliary variables $\mathbf{z}_{i,t}^{i^{'}}$ for $i^{'} \in \mathcal{N}_i$ and $t = 0, 1, ..., N-1$ and substitute the equality constraints in (3.21) with $\boldsymbol{\mu}_{i,t} = \mathbf{z}_{i,t}^{i^{'}}$ and $\boldsymbol{\mu}_{i^{'},t} = \mathbf{z}_{i^{'},t}^{i}$ for $i^{'} \in \mathcal{N}_i$ and $i \neq i^{'}$. Then the augmented Lagrangian function can be formed as

$$\begin{aligned}\mathcal{L}[\{\boldsymbol{\mu}_{i,t}\}_{i=1}^{p}, \mathbf{v}, \mathbf{w}] = \textstyle\sum_{i=1}^{p} \|\boldsymbol{\mu}_{i,t} - p\hat{\mathbf{D}}_i^{\tau-1}\mathbf{x}(t,i)\|_2^2 \\ + \textstyle\sum_{i=1}^{p} \sum_{i^{'} \in \mathcal{N}_i} [\mathbf{v}_{i,t}^{i^{'}}(\boldsymbol{\mu}_{i,t} - \mathbf{z}_{i,t}^{i^{'}}) + \mathbf{w}_{i,t}^{i^{'}}(\boldsymbol{\mu}_{i^{'},t} - \mathbf{z}_{i^{'},t}^{i})] \\ + 0.5c \textstyle\sum_{i=1}^{p} \sum_{i^{'} \in \mathcal{N}_i} [\|\boldsymbol{\mu}_{i,t} - \mathbf{z}_{i,t}^{i^{'}}\|_2^2 + \|\boldsymbol{\mu}_{i,t} - \mathbf{z}_{i^{'},t}^{i}\|_2^2]\end{aligned} \tag{B.1}$$

where $\mathbf{v}$ and $\mathbf{w}$ contain the Lagrange multipliers $\mathbf{v}_{i,t}^{i^{'}}$ and $\mathbf{w}_{i,t}^{i^{'}}$ corresponding to the constraints $\boldsymbol{\mu}_{i,t} = \mathbf{z}_{i,t}^{i^{'}}$ and $\boldsymbol{\mu}_{i^{'},t} = \mathbf{z}_{i^{'},t}^{i}$, respectively. Solving (B.1) involves three steps: Step 1 uses gradient ascent iterations to update the Lagrange multipliers; Step 2 updates $\boldsymbol{\mu}_{i,t}$ by minimizing (B.1) w.r.t. $\boldsymbol{\mu}_{i,t}$ while treating the rest quantities fixed; Step 3 minimizes (B.1) w.r.t. $\mathbf{z}_{i,t}^{i^{'}}$ while fixing the other variables. After reducing the redundant variables, the subproblems (3.21) for $t = 0, \ldots, N-1$ will be tackled through updating the sensor j's local estimate, $\boldsymbol{\mu}_{j,t}$, along with the Lagrange multipliers $\{\mathbf{v}_{j,t}^{j^{'}}\}_{j^{'} \in \mathcal{N}_j}$. Thus, sensor j is responsible for carrying out the updating recursions in (3.22)-(24).

APPENDIX C

PROOF OF THEOREM 1

The S-CCA framework in (3.5) for an infinite number of data converges to the ensemble counterpart in (3.48) which can be equivalently rewritten as

$$\begin{aligned}
&(\mathbf{D}_e, \mathbf{E}_e) \in \arg\min_{\mathbf{D},\mathbf{E}} \mathbb{E}[\|\mathbf{E}\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2] \\
&\upsilon\|\mathbf{E}\boldsymbol{\Sigma}_y\mathbf{E}^T - \mathbf{I}\|_F^2 + \varepsilon\|\mathbf{D}\boldsymbol{\Sigma}_x\mathbf{D}^T - \mathbf{I}\|_F^2 \\
&+ \textstyle\sum_{\rho=1}^{q} \lambda_{E,\rho} \sum_{j=1}^{pf} \mathbf{T}_E(\rho, j) + \sum_{\rho=1}^{q} \lambda_{D,\rho} \sum_{j=1}^{pf} \mathbf{T}_D(\rho, j)
\end{aligned} \tag{C.1}$$

subject to the constraints $|\mathbf{E}(\rho, j)| \leq \mathbf{T}_E(\rho, j)$ and $|\mathbf{D}(\rho, j)| \leq \mathbf{T}_D(\rho, j)$ for $\rho = 1, \ldots, r$ and $j = 1, \ldots, pf$, while $\mathbf{D}_e, \mathbf{E}_e$ indicate an optimal solution. The Lagrangian of (C.1) is

$$\begin{aligned}
&\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{T}_D, \mathbf{T}_E, \mathbf{L}_{1D}, \mathbf{L}_{2D}, \mathbf{L}_{1E}, \mathbf{L}_{2E}) \\
&= \text{tr}(\mathbb{E}[(\mathbf{E}\mathbf{y} - \mathbf{D}\mathbf{x})(\mathbf{E}\mathbf{y} - \mathbf{D}\mathbf{x})^T]) + \boldsymbol{\lambda}_E \mathbf{1}_{q\times 1}^T \mathbf{T}_E \mathbf{1}_{pf\times 1} \\
&+ \boldsymbol{\lambda}_D \mathbf{1}_{q\times 1}^T \mathbf{T}_D \mathbf{1}_{pf\times 1} + \upsilon\text{tr}((\mathbf{E}\boldsymbol{\Sigma}_y\mathbf{E}^T - \mathbf{I})(\mathbf{E}\boldsymbol{\Sigma}_y\mathbf{E}^T - \mathbf{I})^T) \\
&+ \varepsilon\text{tr}((\mathbf{D}\boldsymbol{\Sigma}_x\mathbf{D}^T - \mathbf{I})(\mathbf{D}\boldsymbol{\Sigma}_x\mathbf{D}^T - \mathbf{I})^T) + \text{tr}(\mathbf{L}_{1D}^T(\mathbf{D} - \mathbf{T}_D)) \\
&+ \text{tr}(\mathbf{L}_{2D}^T(-\mathbf{D} - \mathbf{T}_D)) + \text{tr}[\mathbf{L}_{1E}^T(\mathbf{E} - \mathbf{T}_E) + \mathbf{L}_{2E}^T(-\mathbf{E} - \mathbf{T}_E)]
\end{aligned} \tag{C.2}$$

where $\mathbf{L}_{1E}$, $\mathbf{L}_{2E}$, $\mathbf{L}_{1D}$ and $\mathbf{L}_{2D}$ are $\mathbb{R}^{q\times pf}$ matrices whose (ρ, j)th entry contains the Lagrange multiplier associated with the constraints $\mathbf{E}(\rho, j) \leq \mathbf{T}_E(\rho, j)$, $-\mathbf{E}(\rho, j) \leq \mathbf{T}_E(\rho, j)$, $\mathbf{D}(\rho, j) \leq \mathbf{T}_D(\rho, j)$ and $-\mathbf{D}(\rho, j) \leq \mathbf{T}_D(\rho, j)$, respectively. Also, let $\boldsymbol{\lambda}_E := [\lambda_{E,1}, ..., \lambda_{E,q}]^T$ and $\boldsymbol{\lambda}_D := [\lambda_{D,1}, ..., \lambda_{D,q}]^T$. The Karush-Kunh-Tucker (KKT) necessary optimality conditions, see e.g., [4], imply that the following gradients in (C.3) should be equal to $\mathbf{0}_{q\times pf}$ when evaluated at the optimum solution $\mathbf{D}_e$ and $\mathbf{E}_e$, $\mathbf{T}_D^*$ and $\mathbf{T}_E^*$, i.e.,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{D}} = \frac{\partial \mathcal{L}}{\partial \mathbf{E}} = \frac{\partial \mathcal{L}}{\partial \mathbf{T}_D} = \frac{\partial \mathcal{L}}{\partial \mathbf{T}_E} = \mathbf{0}_{q\times pf} \tag{C.3}$$

The equations in (C.3) result the following equalities satisfied at the optimum of (C.1)

$$\begin{aligned}
&(2 - 4\upsilon)\mathbf{E}_e\boldsymbol{\Sigma}_y + 4\upsilon\mathbf{E}_e\boldsymbol{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e\boldsymbol{\Sigma}_y - 2\mathbf{D}_e\boldsymbol{\Sigma}_{xy} \\
&+ \mathbf{L}_{1E}^* - \mathbf{L}_{2E}^* = \mathbf{0}_{q\times pf}
\end{aligned} \tag{C.4}$$

$$(2-4\varepsilon)\mathbf{D}_e\boldsymbol{\Sigma}_x + 4\varepsilon\mathbf{D}_e\boldsymbol{\Sigma}_x\mathbf{D}_e^T\mathbf{D}_e\boldsymbol{\Sigma}_x - 2\mathbf{E}_e\boldsymbol{\Sigma}_{yx} + \mathbf{L}_{1D}^* - \mathbf{L}_{2D}^* = \mathbf{0}_{q\times pf} \tag{C.5}$$

$$\mathbf{L}_{1D}^* + \mathbf{L}_{2D}^* = \boldsymbol{\lambda}_D\mathbf{1}_{q\times 1}\mathbf{1}_{pf\times 1}^T,\ \mathbf{L}_{1E}^* + \mathbf{L}_{2E}^* = \boldsymbol{\lambda}_E\mathbf{1}_{q\times 1}\mathbf{1}_{pf\times 1}^T$$

where the entries of the optimal Lagrange multipliers' matrices should be nonnegative. Moreover, the complementary slackness conditions imply for $\rho = 1, \dots, q$ and $j = 1, \dots, pf$ that

$$\mathbf{L}_{1D}^*(\rho, j)(\mathbf{D}_e(\rho, j) - \mathbf{T}_D^*(\rho, j)) = 0 \text{ and } \mathbf{L}_{2D}^*(\rho, j)(-\mathbf{D}_e(\rho, j) - \mathbf{T}_D^*(\rho, j)) = 0, \tag{C.6}$$

$$\mathbf{L}_{1E}^*(\rho, j)(\mathbf{E}_e(\rho, j) - \mathbf{T}_E^*(\rho, j)) = 0 \text{ and } \mathbf{L}_{2E}^*(\rho, j)(-\mathbf{E}_e(\rho, j) - \mathbf{T}_E^*(\rho, j)) = 0 \tag{C.7}$$

Consider the $q \times 1$ vector $\mathbf{e}_\rho := [0, \dots, 0, 1, 0, \dots, 0]^T$ where the only nonzero entry, equal to 1, exists in the ρth position, for $\rho = 1, \dots, q$. Multiplying (C.4) from the left and right with $\mathbf{e}_\rho^T$ and $\mathbf{E}_{e,\rho:}^T$, respectively, we obtain

$$(2-4\upsilon)\mathbf{E}_{e,\rho:}\boldsymbol{\Sigma}_y\mathbf{E}_{e,\rho:}^T - 2\mathbf{D}_{e,\rho:}\boldsymbol{\Sigma}_{xy}\mathbf{E}_{e,\rho:}^T + 4\upsilon\mathbf{E}_{e,\rho:}\boldsymbol{\Sigma}_y \cdot\mathbf{E}_e^T\mathbf{E}_e\boldsymbol{\Sigma}_y\mathbf{E}_{e,\rho:}^T + \sum_{j=1}^{pf}(\mathbf{L}_{1E}^*(\rho, j) - \mathbf{L}_{2E}^*(\rho, j))\mathbf{E}_e(\rho, j) = 0 \tag{C.8}$$

From the two equalities in (C.4) and (C.8), it follows that $\sum_{j=1}^{pf}(\mathbf{L}_{1E}^*(\rho, j) - \mathbf{L}_{2E}^*(\rho, j))$ $\mathbf{E}_e(\rho, j) = \lambda_{E,\rho}\|\mathbf{E}_{e,\rho:}\|_1$, then

$$(2-4\upsilon)\mathbf{E}_{e,\rho:}\boldsymbol{\Sigma}_y\mathbf{E}_{e,\rho:}^T - 2\mathbf{D}_{e,\rho:}\boldsymbol{\Sigma}_{xy}\mathbf{E}_{e,\rho:}^T + 4\upsilon\mathbf{E}_{e,\rho:}\boldsymbol{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e\boldsymbol{\Sigma}_y\mathbf{E}_{e,\rho:}^T = -\lambda_{E,\rho}\|\mathbf{E}_{e,\rho:}\|_1,\ \rho = 1, ..., q \tag{C.9}$$

Summing the q equations in (C.9) results

$$(1-2\upsilon)\text{tr}(\boldsymbol{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e) - \text{tr}(\boldsymbol{\Sigma}_{xy}\mathbf{E}_e^T\mathbf{D}_e) + 2\upsilon\text{tr}(\boldsymbol{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e\boldsymbol{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e) = -0.5\textstyle\sum_{\rho=1}^{q}\lambda_{E,\rho}\|\mathbf{E}_{e,\rho:}\|_1 \tag{C.10}$$

Using similar steps we obtain

$$
\begin{aligned}
&(1-2\varepsilon)\mathrm{tr}(\mathbf{\Sigma}_x\mathbf{D}_e^T\mathbf{D}_e)-\mathrm{tr}(\mathbf{\Sigma}_{yx}\mathbf{D}_e^T\mathbf{E}_e) && \text{(C.11)}\\
&+2\varepsilon\mathrm{tr}(\mathbf{\Sigma}_x\mathbf{D}_e^T\mathbf{D}_e\mathbf{\Sigma}_x\mathbf{D}_e^T\mathbf{D}_e)=-0.5\textstyle\sum_{\rho=1}^{q}\lambda_{D,\rho}\|\mathbf{D}_{e,\rho:}\|_1
\end{aligned}
$$

As $\sum_{\rho=1}^{q}\mathbf{D}_{e,\rho:}\mathbf{\Sigma}_{xy}\mathbf{E}_{e,\rho:}^T=\mathrm{tr}(\mathbf{\Sigma}_{xy}\mathbf{E}_e^T\mathbf{D}_e)=\mathrm{tr}(\mathbf{D}_e\mathbf{\Sigma}_{xy}\mathbf{E}_e^T)$, (C.9) and (C.11) can be rewritten as

$$
\begin{aligned}
&((1-2\upsilon)\mathbf{E}_{e,\rho:}\mathbf{\Sigma}_y-\mathbf{D}_{e,\rho:}\mathbf{\Sigma}_{xy}+2\upsilon\mathbf{E}_{e,\rho:}\mathbf{\Sigma}_y\mathbf{E}_e^T\mathbf{E}_e\mathbf{\Sigma}_y)\mathbf{E}_{e,\rho:}^T\\
&+0.5\lambda_{E,\rho}\|\mathbf{E}_{e,\rho:}\|_1=0,\ \rho=1,...,q && \text{(C.12)}\\
&((1-2\varepsilon)\mathbf{D}_{e,\rho:}\mathbf{\Sigma}_x-\mathbf{E}_{e,\rho:}\mathbf{\Sigma}_{yx}+2\varepsilon\mathbf{D}_{e,\rho:}\mathbf{\Sigma}_x\mathbf{D}_e^T\mathbf{D}_e\mathbf{\Sigma}_x)\mathbf{D}_{e,\rho:}^T\\
&+0.5\lambda_{D,\rho}\|\mathbf{D}_{e,\rho:}\|_1=0,\ \rho=1,...,q. && \text{(C.13)}
\end{aligned}
$$

Using (C.12) and (74), the cost in (C.1) can be rewritten as

$$
\begin{aligned}
&\textstyle\sum_{\rho=1}^{q}\frac{\lambda_{E,\rho}}{2}\|\mathbf{E}_{\rho:}\|_1+\sum_{\rho=1}^{q}\frac{\lambda_{D,\rho}}{2}\|\mathbf{D}_{\rho:}\|_1-\upsilon\mathrm{tr}(\mathbf{E}\mathbf{\Sigma}_y\mathbf{E}^T\mathbf{E}\mathbf{\Sigma}_y\mathbf{E}^T)\\
&-\varepsilon\mathrm{tr}(\mathbf{D}\mathbf{\Sigma}_x\mathbf{D}^T\mathbf{D}\mathbf{\Sigma}_x\mathbf{D}^T)+(\upsilon+\varepsilon)\mathrm{tr}(\mathbf{I}) && \text{(C.14)}
\end{aligned}
$$

Thus, the minimization problem in (C.1) is equivalent to

$$
\begin{aligned}
&(\mathbf{D}_e,\mathbf{E}_e)\in\arg\min_{\mathbf{D},\mathbf{E}}-\upsilon\mathrm{tr}(\mathbf{E}\mathbf{\Sigma}_y\mathbf{E}^T\mathbf{E}\mathbf{\Sigma}_y\mathbf{E}^T) && \text{(C.15)}\\
&-\varepsilon\mathrm{tr}(\mathbf{D}\mathbf{\Sigma}_x\mathbf{D}^T\mathbf{D}\mathbf{\Sigma}_x\mathbf{D}^T)+\sum_{\rho=1}^{q}\frac{\lambda_{E,\rho}}{2}\|\mathbf{E}_{\rho:}\|_1+\sum_{\rho=1}^{q}\frac{\lambda_{D,\rho}}{2}\|\mathbf{D}_{\rho:}\|_1\\
&\text{s. to the equality contraints in (C.12) and (76).}
\end{aligned}
$$

Given the properly selected sparsity-controlling coefficients $\boldsymbol{\lambda}_D$ and $\boldsymbol{\lambda}_E$, assume that the optimal solutions satisfy

$$
\begin{aligned}
&\|\mathbf{D}_{e,\rho:}\|_0=\ell_{d,\rho}\text{ and }\|\mathbf{D}_{e,\rho:}\|_1=\kappa_{d,\rho}\\
&\|\mathbf{E}_{e,\rho:}\|_0=\ell_{e,\rho}\text{ and }\|\mathbf{E}_{e,\rho:}\|_1=\kappa_{e,\rho} && \text{(C.16)}
\end{aligned}
$$

for $\rho = 1, .., q$, where $\ell_{d,\rho}$ (and $\ell_{e,\rho}$) denotes the number of nonzero entries in $\mathbf{D}_{\rho:}$ (and $\mathbf{E}_{\rho:}$).

Firstly, let's consider the simple case where $q = 1$. Let $\mathbf{D}_{1:} = ||\mathbf{D}_{1:}||_2\tilde{\mathbf{D}}_{1:}$, where $||\tilde{\mathbf{D}}_{1:}||_2 = 1$. Further, let $c_{d,1}\colon = ||\mathbf{D}_{1:}||_2$ and $\gamma_{d,1}\colon = \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T$. Thus, $\mathbf{D}_{1:}\mathbf{\Sigma}_x\mathbf{D}_{1:}^T = c_{d,1}^2\tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T = c_{d,1}^2\gamma_{d,1} \geq 0$. Similarly, let $\mathbf{E}_{1:} = ||\mathbf{E}_{1:}||_2\tilde{\mathbf{E}}_{1:}$, where $||\tilde{\mathbf{E}}_{1:}||_2 = 1$, $c_{e,1}\colon = ||\mathbf{E}_{1:}||_2$ and $\gamma_{e,1}\colon = \tilde{\mathbf{E}}_{1:}\mathbf{\Sigma}_y\tilde{\mathbf{E}}_{1:}^T$. Thus, $\mathbf{E}_{1:}\mathbf{\Sigma}_y\mathbf{E}_{1:}^T = c_{e,1}^2\gamma_{e,1} \geq 0$. In the same way, let us define the quantity $\gamma_{de,1} := \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_{xy}\tilde{\mathbf{E}}_{1:}^T$ that can be used to write $\mathbf{D}_{1:}\mathbf{\Sigma}_{xy}\mathbf{E}_{1:}^T = \mathbf{E}_{1:}\mathbf{\Sigma}_{yx}\mathbf{D}_{1:}^T = c_{d,1}c_{e,1}\gamma_{de,1}$. Then, the minimization problem in (C.15) can be rewritten as

$$\begin{aligned}
&\min\ -\upsilon c_{e,1}^4\gamma_{e,1}^2 - \varepsilon c_{d,1}^4\gamma_{d,1}^2 \text{ subject to} \qquad\qquad (C.17)\\
&(1-2\upsilon)c_{e,1}^2\gamma_{e,1} - c_{d,1}c_{e,1}\gamma_{de,1} + 2\upsilon c_{e,1}^4\gamma_{e,1}^2 = -0.5\lambda_{E,1}\kappa_{e,1}\\
&(1-2\varepsilon)c_{d,1}^2\gamma_{d,1} - c_{d,1}c_{e,1}\gamma_{de,1} + 2\varepsilon c_{d,1}^4\gamma_{d,1}^2 = -0.5\lambda_{D,1}\kappa_{d,1}\\
&0 \leq c_{d,1} \leq \kappa_{d,1},\ 0 \leq c_{e,1} \leq \kappa_{e,1}, 0 \leq \gamma_{e,1} \leq d_e^*,\\
&0 \leq \gamma_{d,1} \leq d_d^*,\ d_{min}(\mathbf{\Sigma}_{xy}) \leq \gamma_{de,1} \leq d_{max}(\mathbf{\Sigma}_{xy}),
\end{aligned}$$

where d_d^* is the maximum spectral radius among all possible $\ell_{d,1} \times \ell_{d,1}$ submatrices of $\mathbf{\Sigma}_x$ that are formed after keeping $\ell_{d,1}$ of its rows and columns with common indices that are determined by the indices of the $\ell_{d,1}$ nonzero entries in the optimal $\mathbf{D}_{e,1:} = ||\mathbf{D}_{e,1:}||_2\tilde{\mathbf{D}}_{e,1:}$, where $\tilde{\mathbf{D}}_{e,1:}$ is the optimal selection for $\tilde{\mathbf{D}}_{1:}$. This explains why $\gamma_{d,1} = \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T \leq d_d^*$ for any unit-vector $\tilde{\mathbf{D}}_{1:}$ for which $||\tilde{\mathbf{D}}_{1:}||_0 = \ell_{d,1}$. In the same way d_e^* is the maximum spectral radius among all possible $\ell_{e,1} \times \ell_{e,1}$ submatrices of $\mathbf{\Sigma}_y$, from which it follows that $\gamma_{e,1} = \tilde{\mathbf{E}}_{1:}\mathbf{\Sigma}_y\tilde{\mathbf{E}}_{1:}^T \leq d_e^*$ for any unit-vector $\tilde{\mathbf{E}}_{1:}$ for which $||\tilde{\mathbf{E}}_{1:}||_0 = \ell_{e,1}$, the optimal selection for $\tilde{\mathbf{E}}_{1:}$ will be denoted as $\tilde{\mathbf{E}}_{e,1:}$. Further, $d_{max}(\mathbf{\Sigma}_{xy})$ and $d_{min}(\mathbf{\Sigma}_{xy})$ denote the largest and smallest singular values of any $\ell_{d,1} \times \ell_{e,1}$ submatrix of $\mathbf{\Sigma}_{xy}$. Further, the third and fourth inequality constraints in (C.17) hold because $||\tilde{\mathbf{E}}_{1:}||_2 \leq ||\tilde{\mathbf{E}}_{1:}||_1$ and $||\tilde{\mathbf{D}}_{1:}||_2 \leq ||\tilde{\mathbf{D}}_{1:}||_1$.

In order to solve (C.17), we form its Lagrangian function

$$
\begin{aligned}
&\mathcal{L}_1(c_{d,1}, c_{e,1}, \gamma_{d,1}, \gamma_{e,1}, \gamma_{de,1}, \mathbf{v}_1) = -\upsilon c_{e,1}^4 \gamma_{e,1}^2 - \varepsilon c_{d,1}^4 \gamma_{d,1}^2 \\
&+ v_1^a[(1-2\upsilon)c_{e,1}^2 \gamma_{e,1} - c_{d,1}c_{e,1}\gamma_{de,1} + 2\upsilon c_{e,1}^4 \gamma_{e,1}^2 + 0.5\lambda_{E,1}\kappa_{e,1}] \\
&+ v_1^b[(1-2\varepsilon)c_{d,1}^2 \gamma_{d,1} - c_{d,1}c_{e,1}\gamma_{de,1} + 2\varepsilon c_{d,1}^4 \gamma_{d,1}^2 + 0.5\lambda_{D,1}\kappa_{d,1}] \\
&+ v_1^c(c_{d,1} - \kappa_{d,1}) - v_1^d c_{d,1} + v_1^e(c_{e,1} - \kappa_{e,1}) - v_1^f c_{e,1} \\
&+ v_1^g(\gamma_{d,1} - d_d^*) - v_1^h \gamma_{d,1} + v_1^i(\gamma_{e,1} - d_e^*) - v_1^j \gamma_{e,1} \\
&+ v_1^k[\gamma_{de,1} - d_{max}(\mathbf{\Sigma}_{xy})] + v_1^l[d_{min}(\mathbf{\Sigma}_{xy}) - \gamma_{de,1}] \qquad \text{(C.18)}
\end{aligned}
$$

where $\mathbf{v}_1 := [v_1^a, v_1^b, v_1^c, v_1^d, v_1^e, v_1^f, v_1^g, v_1^h, v_1^i, v_1^j, v_1^k, v_1^l]^T$ contains the multipliers for the equality and inequality constrains in the minimization task in (C.17). From the KKT necessary optimality conditions, it follows that $v_1^c, v_1^d, v_1^e, v_1^f, v_1^g, v_1^h, v_1^i, v_1^j, v_1^k, v_1^l$ assume nonnegative values.

Applying the KKT necessary optimality conditions in (C.18) involves: i) Differentiating $\mathcal{L}_1(.)$ w.r.t. $c_{d,1}, c_{e,1}, \gamma_{d,1}, \gamma_{e,1}, \gamma_{de,1}$ and making all these partial derivatives equal to zero; and ii) Utilizing the complementary slackness conditions, which make $v_1^{g*}(\gamma_{d,1}^* - d_d^*) = 0$, $v_1^{i*}(\gamma_{e,1}^* - d_e^*) = 0$, $v_1^{k*}(\gamma_{de,1}^* - d_{\text{max}}(\mathbf{\Sigma}_{xy})) = 0$ and $v_1^{l*}(d_{\text{min}}(\mathbf{\Sigma}_{xy}) - \gamma_{de,1}^*) = 0$, where the $*$ superscripts indicate the optimal multipliers. In the same way it turns out that $v_1^{d*} = v_1^{h*} = v_1^{f*} = v_1^{j*} = 0$. After applying these two steps, it follows (details omitted due to space considerations) that v_1^{g*} and v_1^{i*} are strictly positive from which the slackness conditions result that $\gamma_{d,1}^* = d_d^*$ and $\gamma_{e,1}^* = d_e^*$. Now recall that $\gamma_{d,1} = \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T$ for $||\tilde{\mathbf{D}}_{1:}||_2 = 1$ and $||\tilde{\mathbf{D}}_{1:}||_0 = \ell_{d,1}$. Thus, $\gamma_{d,1} = d_d^*$ when $\tilde{\mathbf{D}}_{1:} = \tilde{\mathbf{D}}_{e,1:}$. Similarly, $\gamma_{e,1} = d_e^*$ when $\tilde{\mathbf{E}}_{1:} = \tilde{\mathbf{E}}_{e,1:}$.

Recall that $d_d^* = \max_{\tilde{\mathbf{D}}_{1:}} \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T$, subject to $||\tilde{\mathbf{D}}_{1:}||_2 = 1$ and $||\tilde{\mathbf{D}}_{1:}||_0 = \ell_{d,1}$. It is demonstrated next that if $\tilde{\mathbf{D}}_{e,1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{e,1:}^T = d_d^*$, there must exist a column, namely the i_1th column of $\mathbf{U}_x$ with support $\mathcal{Z}_{i_1}$, where $\mathbf{U}_x$ is the eigenvector matrix of $\mathbf{\Sigma}_x$. Since $\mathbf{D}_{e,1:}$ is a scaled version of $\tilde{\mathbf{D}}_{e,1:}$, latter property implies that $||\mathbf{D}_{e,1:}(\bar{\mathcal{Z}}_{i_1})||_1 = 0$, while

$\|\mathbf{D}_{e,1:}(\mathcal{Z}_{i_1})\|_1 \geqslant \xi_1(\lambda_{D,1})$, where $\xi_1(\lambda_{D,1})$ is strictly positive. Further, let $\mathcal{G}_{m_1}$ denote the index set of the entries of m_1st diagonal block of $\mathbf{\Sigma}_x$ for which $\mathcal{G}_{m_1} = \mathcal{Z}_{i_1}$, while $m_1 \in \{1, \ldots, M\}$. It will be shown that $\mathcal{I}_1 :=$ support$(\mathbf{D}_{e,1:}) =$ support$(\tilde{\mathbf{D}}_{e,1:}) \subseteq \mathcal{G}_{m_1}$. To this end, let $\tilde{\mathbf{D}}_{1:} := [\tilde{\mathbf{D}}^1_{1:}, \tilde{\mathbf{D}}^2_{1:}, \ldots, \tilde{\mathbf{D}}^M_{1:}]$ in which every subvector $\tilde{\mathbf{D}}^m_{1:}$ has $|\mathcal{G}_m|$ entries (such that $\sum_{m=1}^{M} |\mathcal{G}_m| = pf$) and let $\mathcal{I}_{1,m} :=$ support$(\tilde{\mathbf{D}}^m_{1:})$ with $\sum_{m=1}^{M} |\boldsymbol{\mathcal{I}}_{1,m}| = \ell_{d,1}$, where $|\mathcal{I}_{1,m}|$ is the size of set $\mathcal{I}_{1,m}$. Then, it follows that,

$$\begin{aligned} \tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T &= \textstyle\sum_{m=1}^{M} \tilde{\mathbf{D}}^m_{1:}\mathbf{\Sigma}_{x,\mathcal{G}_m}(\tilde{\mathbf{D}}^m_{1:})^T \\ &\leqslant \textstyle\sum_{m=1}^{M} d_{max}(\mathbf{\Sigma}^{\ell_{d,1}}_{x,\mathcal{G}_m})\|\tilde{\mathbf{D}}^m_{1:}\|_2^2 \end{aligned} \tag{C.19}$$

where $d_{max}(\mathbf{\Sigma}^{\ell_{d,1}}_{x,\mathcal{G}_m})$ is the spectral radius of the $|\mathcal{G}_m| \times |\mathcal{G}_m|$ submatrix $\mathbf{\Sigma}^{\ell_{d,1}}_{x,\mathcal{G}_m}$, which is formed by keeping $\ell_{d,1}$ rows and columns of $\mathbf{\Sigma}_{x,\mathcal{G}_m} \in \mathbb{R}^{|\mathcal{G}_m|\times|\mathcal{G}_m|}$ with common indices. The inequality in (C.19) holds true because each subvector $\tilde{\mathbf{D}}^m_{1:}$ of $\tilde{\mathbf{D}}^{1:}$ can have at most $\ell_{d,1}$ nonzero entries. If $d_d^{\ell_{d,1}}$ denotes the maximum spectral radius that can be achieved by any $\ell_{d,1} \times \ell_{d,1}$ submatrix $\mathbf{\Sigma}^{\ell_{d,1}}_{x,\mathcal{G}_m}$ that is contained in a diagonal block $\mathbf{\Sigma}_{x,\mathcal{G}_m}$ for $m = 1, \ldots, M$, then since $\sum_{m=1}^{M} \|\tilde{\mathbf{D}}^m_{1:}\|_2^2 = 1$ and equation (C.19), it holds that $\tilde{\mathbf{D}}_{1:}\mathbf{\Sigma}_x\tilde{\mathbf{D}}_{1:}^T \leq d_d^{\ell_{d,1}}$. Then, it should hold that $d_d^{\ell_{d,1}} = d_d^*$. Then, the max value d_d^* can be attained if and only if the nonzero entry indices of the optimal $\tilde{\mathbf{D}}_{e,1:}$ satisfy $\mathcal{I}_1 :=$ support$(\tilde{\mathbf{D}}_{e,1:}) \subseteq \mathcal{G}_{m_1}$ for a $m_1 \in \{1, ..., M\}$. This further implies that there exists an eigenvector $\mathbf{U}_{x,:i_1}$, with support $\mathcal{Z}_{i_1} = \mathcal{G}_{m_1}$ for which $\mathcal{I}_1 \subseteq \mathcal{Z}_{i_1}$. Thus, it is deduced that $\|\tilde{\mathbf{D}}_{e,1:}(\bar{\mathcal{Z}}_{i_1})\|_1 = 0$ and $\|\tilde{\mathbf{D}}_{e,1:}(\mathcal{Z}_{i_1})\|_1 \geq \xi(\lambda_{D,1}) > 0$ since the $\ell_{d,1}$ nonzero entries have indices in $\mathcal{Z}_{i_1}$. Positivity of $\xi(\lambda_{D,1})$ is ensured since $\|\tilde{\mathbf{D}}_{e,1:}\|_2 = 1$ and $\lambda_{D,1}$ is selected such that $\mathbf{D}_{e,1:} \neq \mathbf{0}$.

Similarly, optimal $\mathbf{E}_{e,1:}$ satisfies $\mathcal{I}_1' :=$ support$(\mathbf{E}_{e,1:}) \subseteq \mathcal{G}_{m_1'}$ for $m_1' \in \{1, \ldots, M\}$ and $\mathcal{G}_{m_1'}$ is the index set for the m_1'st diagonal block of $\mathbf{\Sigma}_y$. In the same way, there exists an eigenvector $\mathbf{U}_{y,:i_1'}$, with support $\mathcal{Z}_{i_1'} = \mathcal{G}_{m_1'}$ for which $\mathcal{I}_1' \subseteq \mathcal{Z}_{i_1'}$. Thus, it is deduced that $\|\tilde{\mathbf{E}}_{e,1:}(\bar{\mathcal{Z}}_{i_1'})\|_1 = 0$ and $\|\tilde{\mathbf{E}}_{e,1:}(\mathcal{Z}_{i_1'})\|_1 \geq \xi'(\lambda_{E,1}) > 0$ since the $\ell_{e,1}$ nonzero entries have indices in $\mathcal{Z}_{i_1'}$.

Next, it is shown that $\mathcal{G}_{m_1'} = \mathcal{G}_{m_1} = \mathcal{Z}_{i_1'} = \mathcal{Z}_{i_1}$. From the constraints in (C.17), it follows that, if the optimal $\gamma^*_{e,1}, \gamma^*_d, c^*_d, c^*_e$ are all strictly positive, then, γ^*_{de} are strictly positive too. Recall that $\mathbf{\Sigma}_x$ and $\mathbf{\Sigma}_y$ have the same block diagonal structure then $\mathcal{G}_{m_1} = \mathcal{G}_{m_1'}$, otherwise $\mathcal{I}_1 \cap \mathcal{I}_1^{'} = \emptyset$ which would further imply that $\gamma^*_{de,1} = \tilde{\mathbf{E}}_{e,\rho:}\mathbf{\Sigma}_{yx}\tilde{\mathbf{D}}^T_{e,\rho:} = 0$ resulting a contradiction. Thus, $\mathcal{G}_{m_1'} = \mathcal{G}_{m_1}$ results $\mathcal{Z}_{i_1'} = \mathcal{Z}_{i_1}$.

Let's consider the more general case where $q > 1$. To this end, let $\mathbf{D}_{\rho:} = \|\mathbf{D}_{\rho:}\|_2\widetilde{\mathbf{D}}_{\rho:}$ with $\|\widetilde{\mathbf{D}}_{\rho:}\|_2 = 1$, and $\mathbf{E}_{\rho:} = \|\mathbf{E}_{\rho:}\|_2\widetilde{\mathbf{E}}_{\rho:}$ with $\|\widetilde{\mathbf{E}}_{\rho:}\|_2 = 1$, for $\rho = 1, ..., q$. Further, let $c_{d,\rho}\colon = \|\mathbf{D}_{\rho:}\|_2$, $\gamma_{d,\rho}\colon = \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{\rho:}$, $\gamma_{de,\rho}\colon = \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_{xy}\widetilde{\mathbf{E}}^T_{\rho:}$ and $\delta_{d,\rho j}\colon = \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{j:}$ ($j \neq \rho, j, \rho = 1, \ldots, q$). Notice that $\delta_{d,\rho j} = \delta_{d,j\rho}$, $\delta_{d,\rho j} \leq \gamma_{d,\rho}$, $\delta_{d,\rho j} \leq \gamma_{d,j}$ and $\gamma_{d,\rho} \leq d^*_{d,\rho}$. Where, $d^*_{d,\rho}$ corresponds to the maximum value that $\widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{\rho:}$ can attain when $\|\widetilde{\mathbf{D}}_{\rho:}\|_0 = \ell_{d,\rho}$, which can be equivalently written as $d^*_{d,\rho} = \max_{\widetilde{\mathbf{D}}_{\rho:}}\widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{\rho:}$ subject to the constraint that $\|\widetilde{\mathbf{D}}_{\rho:}\|_0 = \ell_{d,\rho}$. Moreover, let $c_{e,\rho}\colon = \|\mathbf{E}_{\rho:}\|_2$, $\gamma_{e,\rho}\colon = \widetilde{\mathbf{E}}_{\rho:}\mathbf{\Sigma}_y\widetilde{\mathbf{E}}^T_{\rho:}$ and $\delta_{e,\rho j}\colon = \widetilde{\mathbf{E}}_{\rho:}\mathbf{\Sigma}_y\widetilde{\mathbf{E}}^T_{j:}$ ($j \neq \rho$). Notice that $\delta_{e,\rho j} = \delta_{e,j\rho}$, $\delta_{e,\rho j} \leq \gamma_{e,\rho}$, $\delta_{e,\rho j} \leq \gamma_{e,j}$, and $\gamma_{e,\rho} \leq d^*_{e,\rho}$, in which, $d^*_{e,\rho}\colon = \max_{\widetilde{\mathbf{E}}_{\rho:}}\widetilde{\mathbf{E}}_{\rho:}\mathbf{\Sigma}_y\widetilde{\mathbf{E}}^T_{\rho:}$, subject to the constraint $\|\widetilde{\mathbf{E}}_{\rho:}\|_0 = \ell_{e,\rho}$. Recall that, $\boldsymbol{\lambda}_D$ and $\boldsymbol{\lambda}_E$ have been selected such that $\|\mathbf{D}^*_{\rho:}\|_1 = \kappa_{d,\rho}$ and $\|\mathbf{D}^*_{\rho:}\|_0 = \ell_{d,\rho}$, while $\|\mathbf{E}^*_{\rho:}\|_1 = \kappa_{e,\rho}$ and $\|\mathbf{E}^*_{\rho:}\|_0 = \ell_{e,\rho}$ for $\rho = 1, ..., q$.

In this case, let's denote the Lagrangian function of (C.15) as $\mathcal{L}_2(\cdot)$. Next, we apply KKT conditions to derive necessary conditions that the optimal solution of the minimization problem should satisfy, which involves the following three steps (assume that $v < 0.5$ and $\varepsilon < 0.5$): 1) Differentiating $\mathcal{L}_2(\cdot)$ w.r.t. $c_{d,\rho}$, $c_{e,\rho}$, $\gamma_{d,\rho}$, $\gamma_{e,\rho}$, $\gamma_{de,\rho}$, $\delta_{d,\rho j}$, and $\delta_{e,\rho j}$; 2) Setting the corrsponding derivatives equal to zero; 3) Applying the complementary slackness conditions for the optimal Lagrange multipliers.

Firstly, we consider the easier case where $r = 1$, which makes $q = M$. From Cauchy-Schwarz inequality and $\delta_{d,\rho j} \leq \gamma_{d,\rho}$, we know that the two sides are equal if and only if $\widetilde{\mathbf{D}}_{\rho:}$ and $\widetilde{\mathbf{D}}_{j:}$ are linearly dependent. As $\|\widetilde{\mathbf{D}}_{\rho:}\|_2 = \|\widetilde{\mathbf{D}}_{j:}\|_2 = 1$, if and only if $\widetilde{\mathbf{D}}_{\rho:} = \widetilde{\mathbf{D}}_{j:}$, then, $\delta_{d,\rho j} = \gamma_{d,\rho} = \gamma_{d,j}$. Consider that the M sources are nonoverlapping, there could not be two rows of $\mathbf{D}_e$ which have the same direction. Thus $\delta_{d,\rho j} < \gamma_{d,\rho}$ and $\delta_{d,\rho j} < \gamma_{d,j}$. Similarly, $\delta_{e,\rho j} < \gamma_{e,\rho}$ and $\delta_{e,\rho j} < \gamma_{e,j}$. Then after the aforementioned three steps, it follows that at an optimal point it must hold that $\gamma_{d,\rho} = d^*_{d,\rho}$, $\gamma_{e,\rho} = d^*_{e,\rho}$, $\gamma_{d,\rho j} = 0$ and $\gamma_{e,\rho j} = 0$ for $\rho = 1, ..., q$, $j = 1, ..., q, j \neq \rho$. Since $\gamma_{d,\rho j} = 0$, it follows that the optimal direction vector $\widetilde{\mathbf{D}}_{\rho:}$ should be selected such that $\widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{j:} = 0$ for $j \neq \rho$, while $\gamma^*_{e,\rho} = \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{\rho:}$ is equal to the maximum possible value $d^*_{d,\rho}$. The previous properties and since $\mathbf{D}_{e,\rho:} = c^*_{d,\rho}\widetilde{\mathbf{D}}_{\rho:}$, results that the direction vector for the ρth row of the optimal matrix $\mathbf{D}_e$, namely $\widetilde{\mathbf{D}}_{e,\rho:}$, should be selected such that

$$\begin{aligned} \widetilde{\mathbf{D}}_{e,\rho:} &= \text{arg max}_{\widetilde{\mathbf{D}}_{\rho:}} \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{\rho:} \\ &\text{s. to } \widetilde{\mathbf{D}}_{\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{j:} = 0 \, , \, \|\widetilde{\mathbf{D}}_{\rho:}\|_2 = 1 \, , \, \|\widetilde{\mathbf{D}}_{e,\rho:}\|_1 = \kappa_{d,\rho} \end{aligned} \tag{C.20}$$

where $\rho = 1, ..., q$, $j \neq \rho$. Using similar derivations as when $q = 1$, it can be shown that if $\widetilde{\mathbf{D}}_{e,\rho:}\mathbf{\Sigma}_x\widetilde{\mathbf{D}}^T_{e,\rho:} = d^*_{d,\rho}$, there must exist a column, namely the i_ρth column of $\mathbf{U}_x$ with support $\mathcal{Z}_{i_\rho}$. As $\mathbf{D}_{e,\rho:}$ is a scaled version of $\widetilde{\mathbf{D}}_{e,\rho:}$, it implies that $\|\mathbf{D}_{e,\rho:}(\bar{\mathcal{Z}}_{i_\rho})\|_1 = 0$, while $\|\mathbf{D}_{e,\rho:}(\mathcal{Z}_{i_\rho})\|_1 \geqslant \xi_\rho(\lambda_{D,\rho})$, where $\xi_\rho(\lambda_{D,\rho})$ is strictly positive.

Similarly, $\|\mathbf{E}_{e,\rho:}(\bar{\mathcal{Z}}_{i'_\rho})\|_1 = 0$, while $\|\mathbf{E}_{e,\rho:}(\mathcal{Z}_{i'_\rho})\|_1 \geqslant \xi'_\rho(\lambda_{E,\rho})$, where $\xi'_\rho(\lambda_{E,\rho})$ is strictly positive and $\mathcal{Z}_{i'_\rho} = \text{support}(\mathbf{U}_{y,:i'_\rho})$, while $\mathbf{U}_y$ is the eigenvector matrix of $\mathbf{\Sigma}_y$. Further, $\mathcal{G}_{m'_\rho} = \mathcal{G}_{m_\rho} = \mathcal{Z}_{i'_\rho} = \mathcal{Z}_{i_\rho}$, where $\mathcal{G}_{m_\rho}$ and $\mathcal{G}_{m'_\rho}$ are the index sets for the m_ρth diagonal block of $\mathbf{\Sigma}_x$ and the $m^{'}_\rho$th diagonal block of $\mathbf{\Sigma}_y$, respectively.

Using the similar way, we can prove the Theorem 1 for the more general case where $r > 1$.

APPENDIX D

IMPACE OF MOVING AVERAGE FILTER

Consider the measurement acquired at sensor j at time instant t

$$x_j(t) = \sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) s_m(t-\tau) + w_j(t) \tag{D.1}$$

where $\mathcal{S}_j$ corresponds to the set that contains the source indices observed by sensor j.

The MA filtering applied here boils down to form a running sample-average of P consecutive measurements at time instant t, i.e.,

$$\begin{aligned} \overline{x}_j(t) &= \frac{1}{P} \sum_{\ell=t}^{t-P+1} \left[\sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) s_m(\ell-\tau) + w_j(\ell) \right] \\ &= \sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) \left[P^{-1} \sum_{\ell=t}^{t-P+1} s_m(\ell-\tau) \right] + \frac{1}{P} \sum_{\ell=t}^{t-P+1} w_j(\ell) \\ &= \sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) \bar{s}_m(t-\tau) + \bar{w}_j(t), \end{aligned} \tag{D.2}$$

where $\bar{s}_m(t-\tau) := P^{-1} \sum_{\ell=t}^{t-P+1} s_m(\ell-\tau)$ and $\bar{w}_j(\ell)$ corresponds to the averaged noise term in (D.2).

Note that if the length of the MA filter P is selected sufficiently large compared to L, while the source signals have a time-invariant ensemble average (e.g., wide sense stationary processes) then in that case it holds that

$$\bar{s}_m(t) \approx \bar{s}_m(t-1) \approx \ldots \approx \bar{s}_m(t-L+1), \tag{D.3}$$

due to the averaging effect of a sufficiently large number of sensor measurements. Then, after utilizing (D.3) in the third equation in (D.2) we obtain

$$\begin{aligned} \overline{x}_j(t) &\approx \sum_{m \in \mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) \bar{s}_m(t) + \bar{w}_j(t) && \text{(D.4)} \\ &= \sum_{m \in \mathcal{S}_j} \bar{H}_{j,m} \bar{s}_m(t) + \bar{w}_j(t), && \text{(D.5)} \end{aligned}$$

where $\bar{H}_{j,m} := c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau)$.

From (D.4) it follows readily that the noise $\bar{w}_j(t)$ has zero-mean and variance σ_w^2/P. Thus, for sufficiently large P the noise can be made negligible and virtually be ignored. Next, it is demonstrated how the MA measurements can be utilized to obtain an accurate estimate for the number of sources present in the monitored field. Stacking all the MA measurements in (D.4) in a vector $\bar{\mathbf{x}}_t$ we obtain

$$\bar{\mathbf{x}}_t := [\bar{x}_1(t) \ldots \bar{x}_p(t)]^T = \sum_{m=1}^{M} \bar{\mathbf{h}}_m \bar{s}_m(t), \tag{D.6}$$

where $\bar{\mathbf{h}}_m$ is a $p \times 1$ vector whose jth entry is zero, namely $\bar{\mathbf{h}}_m[j] = 0$, if the measurements of sensor j do not contain information about source $s_m(t)$, otherwise $\bar{\mathbf{h}}_m[j] = c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau)$. If there is $\bar{N} := N - P + 1$ training sensor data vectors available, i.e., $\{\bar{\mathbf{x}}_t\}_{t=1}^{\bar{N}}$ and after subtracting the sample-average estimate for the mean of $\bar{\mathbf{x}}_t$, namely $\mathbf{m}_{\bar{x}} = \bar{N}^{-1} \sum_{t=1}^{\bar{N}} \bar{\mathbf{x}}_t$, then the MA data covariance matrix can be estimated as

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_{\bar{x}} &= \frac{1}{\bar{N}} \sum_{t=1}^{\bar{N}} [\bar{\mathbf{x}}_t - \mathbf{m}_{\bar{x}}][\bar{\mathbf{x}}_t - \mathbf{m}_{\bar{x}}]^T & \text{(D.7)} \\ &= \textstyle\sum_{m=1}^{M} \sum_{m'=1}^{M} \bar{\mathbf{h}}_m \bar{\mathbf{h}}_{m'}{}^T & \\ &\times \quad [\bar{N}^{-1} \textstyle\sum_{t=1}^{\bar{N}} [\bar{s}_m(t) - \bar{\bar{s}}_m][\bar{s}_{m'}(t) - \bar{\bar{s}}_m]], & \text{(D.8)} \end{aligned}$$

where $\bar{\bar{s}}_m := \bar{N}^{-1} \sum_{t=1}^{\bar{N}} \bar{s}_m(t)$. For sufficiently large number of training data $\bar{N}$ and since the sources are uncorrelated it turns out from Law of Large Numbers [39]

$$\bar{N}^{-1} \textstyle\sum_{t=1}^{\bar{N}} [\bar{s}_m(t) - \bar{\bar{s}}_m][\bar{s}_{m'}(t) - \bar{\bar{s}}_m] \approx \sigma_m^2 \delta(m - m') \tag{D.9}$$

with $\delta(m - m')$ denoting the Kronecker delta function and σ_m^2 corresponds to the variance of source $s_m(t)$. Thus, the MA data covariance matrix in (D.7) can be written as

$$\hat{\boldsymbol{\Sigma}}_{\bar{x}} \approx \textstyle\sum_{m=1}^{M} \sigma_m^2 \bar{\mathbf{h}}_m \bar{\mathbf{h}}_m^T. \tag{D.10}$$

As long as the column vectors $\bar{\mathbf{h}}_m$ are linearly independent it is easily seen that for sufficiently large number of training data N, and MA filtering length P the number of nonzero eigenvalues of the MA data covariance matrix estimate $\hat{\boldsymbol{\Sigma}}_{\bar{x}}$ is equal to the number of sources M. Note that for the random Gaussian channel coefficients considered in Sec. 5.1, the vectors $\{\bar{\mathbf{h}}_m\}_{m=1}^{M}$ are linearly independent.

APPENDIX E

PROOF OF PROPOSITION 3

Consider sensor j in which after applying the MA filter to its measurements following (5.1), and utilizing the result from (D.2) it follows

$$\bar{x}_j(t) = \sum_{m\in\mathcal{S}_j} c_{j,m} \sum_{\tau=0}^{L-1} h_{j,m}(\tau) \left[P^{-1}\sum_{p=t}^{t-P+1} s_m(p-\tau)\right] + \frac{1}{P}\sum_{p=t}^{t-P+1} w_j(p) \qquad \text{(E.1)}$$

where $\mathcal{S}_j$ corresponds to the set of sources that are affecting the measurements of sensor j. When the number of measurements P is selected sufficiently large, the noise variance in (E.1) can be made arbitrarily small as stated in Apdx. A. Thus, for sufficiently large P the MA measurements in (E.1) can be assumed approximately to be noise free which further implies that the sensor measurements are scaled versions of the modified source signal $\overline{s}_m(t)$. Since the modified source signal $\overline{s}_m(t)$ is formed by adding a set of P samples of the original source signal $s_m(t)$, and the original source signals are uncorrelated then the modified source signals are also uncorrelated.

It is demonstrated how the recursive interplay between PCA and CS-CCA in Sec. 5.2.2 ensures the correct clustering of the sensors according to their source content for sufficiently large MA filter length P and number of training data N. S-CCA at first is applied across all the sensors' MA measurements which are stacked in vectors $\bar{\mathbf{x}}(t)$ for $t = 1, \ldots, \bar{N}$ and used to form the MA data covariance matrix $\boldsymbol{\Sigma}_x^0$ using (5.7) and all measurements in $\bar{\mathbf{x}}(t)$. For sufficiently large P, the eigenvalues of $\boldsymbol{\Sigma}_x^0$ corresponding to the sources will be significantly larger in magnitude than the eigenvalues corresponding to noise which will have negligible magnitude. Thus, by selecting a proper threshold (dependent on the value of P) the source-related eigenvalues can be identified, and their cardinality will correspond to the actual number of sources M. Then, CS-CCA can be employed to identify the M different clusters $\hat{\mathcal{M}}_{m^1}^1$ for $m^1 = 1, \ldots, M$. For each of these clusters the corresponding sensor measurements are stacked in vectors $\bar{\mathbf{x}}_1^1(t), \bar{\mathbf{x}}_2^1(t), \ldots, \bar{\mathbf{x}}_M^1(t)$ for $t = 1, \ldots, \bar{N}$ respectively, where the subscript indicates the cluster index and the superscript the iteration index. The aforementioned vectors are used in the same way as during the first iteration to form via sample-averaging the covariance matrices $\mathbf{\Sigma}^1_{x,m^1}$ in (5.7). After PCA is applied to each $\mathbf{\Sigma}^1_{x,m^1}$, the number of sources contained in the mth cluster measurements $\bar{\mathbf{x}}^1_m(t)$ can be determined as delineated earlier. The alternating interplay between CS-CCA and PCA is continued until all resulting clusters either contain information about only one source, or have a single measurement that may contain information about multiple sources.

During the merging process, when PCA is applied for sufficiently large P and N as described in Sec. 5.2.2 then after a finite number of iterations, which will not exceed the number of sensors p (worst case scenario where all clusters contain a single-sensor measurements), this process will result M clusters that contain the measurements of sensors observing a single common source. If there are sensors that sense more than one sources, then there will also be clusters that contain measurements which have information about these sources. The source content of these multi-source clusters will be identified by employing the PCA process described in Sec. 5.2.2 which will determine the single-source clusters that share the same source content with the multi-source cluster.

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] A. Avokh and G. Mirjalily, "Dynamic Balanced Spanning Tree (DBST) for Data Aggregation in Wireless Sensor Networks," *Proc. of International Symposium on Telecommunication (IST)*, pp. 391–396, 2010.

[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] D. P. Bertsekas, *Nonlinear Programming.* 2nd Edition, Athena Scientific, Massachussets, 1999.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* 2nd Edition, Athena Scientific, Massachussets 1997.

[6] D. R. Brillinger, *Time Series: Data Analysis and Theory*. Expanded Edition, Holden Day, 1981.

[7] 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.

[8] 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.

[9] J. M. Bruce and P. L. Dragotti, “Consensus for the Distributed Estimation of Point Diffusion Sources in Sensor Networks,” *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015.

[10] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Convex Optimization with Sparsity-Inducing Norms,” *Optimization for Machine Learning*, MIT Press, 2011.

[11] A. Choromanska, and C. Monteleoni, “Online Clustering with Experts,” *In Proc. Of 15th Int. Conf. on Artificial Intelligence and statistics (AISTATS)*, La Palma, Canary Islands, pp. 1-18, Apr. 2012.

[12] J. H. Cui, J. Kong, M. Gerla, and S. Zhou, “The Challenges of Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications,” *IEEE Networks*, no. 20, pp. 12–18, 2006.

[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] B. Cheng, J. C. Yang, S. C. Yan, Y. Fu, and T. S. Huang, “Learning with ℓ_1-Graph for Image Analysis,” *IEEE Trans. Image Processing*, vol. 19, no. 4, pp. 858-866, Apr. 2010.

[15] J. Chen and I. D. Schizas, “Online Distributed Sparsity-Aware Canonical Correlation Analysis,” *IEEE Trans. on Signal Processing*, vol. 64, no. 3, pp. 688–703, 2016.

[16] J. Chen, I. D. Schizas, “Distributed information-based clustering of heterogeneous sensor data,” *Signal Processing*, available at http://dx.doi.org/10.1016/j.sigpro.2015.12.017 , 2016.

[17] D. Cai, X. He, and J. Han, “Spectral Regression: A Unified Approach for Sparse Subspace Learning,” *Proc. IEEE Seventh Int’l Conf. Data Mining (ICDM)*, 2007.

[18] J. M. Hoem, “A contribution to the statistical theory of linear graduation,” *Insurance: Mathematics and Economics*, vol. 3, no. 1, pp. 117, 1984, doi: 10.1016/0167-6687(84)90014-3.

[19] E. Candès, J. Romberg, and T. Tao, “Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information,” *IEEE Trans. on Info. Theory*, pp. 489–509, Feb. 2006.

[20] X. Chen, H. Liu, and J. G. Carbonell, “Structured Sparse Canonical Correlation Analysis,” *Proc. of Intl. Conf. on Artificial Intelligence and Stats. (AISTATS),* La Palma, Canary Islands, pp. 199–207, 2012.

[21] 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.

[22] J. Coleman, and J. Hardin, “Robust Sparse Canonical Correlation Analysis and PITCHf/x”, Tech. Report. Available: ”http://pages.pomona.edu/ jsh04747/Student%20Theses/JakeColeman13.pdf”

[23] 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.

[24] I. Dokmanic, J. Ranieri, A. Chebira, and M. Vetterli, “Sensor Networks for Diffusion Fields: Detection of Sources in Space and Time,” *Proc. of 49th Allerton Conf. Commun., Control, Comput. (Allerton)*, pp. 1552–1558, Monticello, Sep. 2011.

[25] P. Elena, T. David, and B. Joseph, “Sparse Canonical Correlation Analysis with Application to Genomic Data Integration,” *Statistical Applications in Genetics and Molecular Biology*, vol. 8, no. 1, pp. 1-34, 2009.

[26] A. V. Fiacco, “Introduction to Sensitivity and Stability Analysis in Nonlinear Programming”, Academic Press, 1983.

[27] D. Garcia-Alvarez, “ Fault Detection using Principal Component Analysis (PCA) in a Wastewater Treatment Plant (WWTP),” *Proc. of the International Students Scientific Conference*, 2009.

[28] I. Guedalia, M. London, and M. Werman, “An On-Line Agglomerative Clustering Method for Nonstationary Data,” *Neural Computation*, vol. 11, pp. 521-540, 1999.

[29] 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.

[30] D. R. Hardoon and J. Taylor, “Sparse Canonical Correlation Analysis,” *Machine Learning*, vol. 83, no. 3, pp. 331-353, 2011.

[31] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On Clustering Validation Techniques,” *Journal of Intelligent Information Systems*, 17, pp. 107-145, 2001.

[32] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning”, Springer, 2009.

[33] H. Hotelling, “Relations Between Two Sets of Variates,” *Biometrika*, vol. 28, pp. 321–377, 1936.

[34] R. He, B. G. Hu, W. S. Zheng, and Y. Q. Guo, “Two-Stage Sparse Representation for Robust Recognition on Larg-Scale Database,” *Proc. AAAI Conf. Artificial Intelligence*, 2010.

[35] A.K. Jain, M.N. Murty, and P.J. Flynn, “Data Clustering: A Review”, *ACM Computing Surveys*, 31(3), pp. 264-323, 1999.

[36] 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.

[37] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[38] S. Lloyd, “Least Squares Quantization in PCM,” *IEEE Trans. Inf. Theory,* vol. 28, no. 2, pp. 129–137, Mar. 1982.

[39] M. Loeve, *Probability Theory 1*. Fourth Edition, Springer Verlag, 1977.

[40] Y. M. Lu, P. L. Dragotti, and M. Vetterli, “Localizing Point Sources in Diffusion Fields from Spatiotemporal Samples,” *Proc. of the 9th Int. Conf. Sampl. Theory Appl. (SampTa)*, Singapore, May 2011.

[41] 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.

[42] Y. O. Li, T. Adali, W. Wang, and V. Calhoun, “Joint Blind Source Separation by Multiset Canonical Correlation Analysis”, *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3918–3929, 2009.

[43] 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.

[44] S. C. Madeira and A. L. Oliveira, “Biclustering Algorithms for Biological Data Analysis: A Survey,” *IEEE/ACM Trans. Comput. Biol. Bioinform.,* vol. 1, no. 1, pp. 24-45, 2004.

[45] D. Mantri, N. R. Prasad and R. Prasad, “BHCDA: Bandwidth efficient Heterogeneity Aware Cluster Based Data Aggregation for Wireless Sensor Network,” *Proc. of IEEE Intl. Conf. of Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1064–1069, Aug. 2013.

[46] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Boca Raton, FL: Chapman and Hall/CRC, 2005.

[47] J. B. MacQueen, "Some Methods for Classification and Analysis of MultiVariate Observations," *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, L. M. L. Camand J. Neyman, eds,* vol. 1, pp. 281–297, 1967.

[48] J. Matthes, L. Groll, and H. B. Keller, "Source Localization by Spatially Distributed Electronic Noses for Advection and Diffusion," *IEEE Trans. Signal Processing*, vol. 53, no. 5, pp. 1711–1719, May 2005.

[49] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Y. Li, and Y. Liu, "CitySee: Urban CO2 Monitoring with Sensors," *Proc. IEEE Int. Conf. Conput. Commun.*, pp. 1611–1619, 2012.

[50] A. Nehorai, B. Porat, and E. Paldi, "Detection and Localization of Vapor-Emitting Sources," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 243-253, Jan. 1995.

[51] A. Nordio, C. F. Chiasserini, and E. Viterbo, "Bandlimited field reconstruction for wireless sensor networks," *Tech Rep.*, Politecnico di Torino, Jan. 2006.

[52] E. Parkhomenko, D. Tritchiler, 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.

[53] 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.

[54] N. S. Patil and P. R. Patil, "Data Aggregation in Wireless Sensor Network", *IEEE Int. Conf. Computational Intelligence and Computing Research*, Dec. 2010.

[55] P. Patil and U. Kulkarni, "SVM based Data Redundancy Elimination for Data Aggregation in Wireless Sensor Networks, *in processing of Advances in Computing, Commu-*

nications and Informatics (ICACCI), International Conference IEEE, pp. 1309-1316, Aug. 2013.

[56] L. S. Qiao, S. C. Chen, and X. Y. Tan, "Sparsity Preserving Projections with Applications to Face Recognition," *Pattern Recognition*, vol. 43, no. 1, pp. 31-341-, 2010.

[57] S. Ramachandran, A. K. Gopi, G. V. Elumalai, and M. Chellapa, " REDD: Redundancy Eliminated Data Dissemination in Cluster Based Mobile Sinks," *ICRTIT, International Conference IEEE*, Jun. 2011.

[58] G. Reise, G. Matz, and K. Grochenig, "Distributed Field Reconstruction in Wireless Sensor Networks Based on Hybrid Shift-Invariant Spaces," *IEEE Trans. Signal Process.* vol. 60, no. 10, pp. 5426-5437, Oct. 2012.

[59] J. Ranieri, A. Vincenzi, A. Chebira, D. A. Alonso, and M. Vetterli, "EigenMaps: Algorithms for Optimal Thermal Maps Extraction and Sensor Placement on Multicore Processors," *Proc. 49th Design Autom. Conf. (DAC), ACM*, pp. 636–641, San Francisco, 2012.

[60] 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.

[61] 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.

[62] F. Sawo, K. Roberts, and U. D. Hanebeck, "Bayesian Estimation of Distributed Phenomena Using Discretized Representations of Partial Differential Equations," *Proc. 3rd IEEE Int. Conf. Inf. Control, Autom., Robot. (ICINCO)*, pp. 16–23, Aug. 2006.

[63] S. Sirsikar and S. Anavatti, "Issues of Data Aggregation Methods in Wireless Sensor Network: A Survey," *Proc. of 4th International Conference on Advances in Computing, Communication and Control (ICAC)*, vol. 49, pp. 194–201, 2015.

[64] 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.

[65] I. D. Schizas, “Distributed Informative-Sensor Identification using Sparsity-Aware Matrix Factorization,” *IEEE Transactions on Sig. Proc.*, vol. 61, no. 18, pp. 4610–4624, Sep. 2013.

[66] I. D. Schizas and A. Aduroja, “A Distributed Framework for Dimensionality Reducion and Denoising,” *IEEE Trans. on Signal Processing*, vol. 63, no. 23, pp. 6379–6394, Dec. 2015.

[67] 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.

[68] S. Simiċ 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.

[69] S. Theodoridis, and K. Koutroumbas, *Pattern Recognition*. Third Ed., Academic Press, 2006.

[70] 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.

[71] 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.

[72] M. O. Ulfarsson and V. Solo, “Sparse Variable PCA Using Geodesic Steepest Descent,” *IEEE Transactions on Signal Processing*, vol. 10, no. 12, pp. 5823–5832, 2008.

[73] J. Via, I. Santamaria, and J. Perez, "A Robust RLS Algorithm For Adaptive Canonical Correlation Analysis," *in Proc. of ICASSP*, Philadelphia, PA, pp. 365-368, Mar 2005.

[74] 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*, 7. Issue 1, Article 3, 2008.

[75] S. Waaijenborg and A. H. Zwinderman, "Sparse Canonical Correlation Analysis for Identifying, Connecting and Completing Gene-Expression Networks," *BMC Bioinformatics*, vol. 10, article 315, 2009.

[76] J. Wright, A. Yang, S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. Pattern Aanlysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, Feb. 2009.

[77] 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.

[78] B. Widrow and S. Steam, "Adaptive Signal Processing," 1st Edition, Prentice Hall, Mar. 1985.

[79] 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.

[80] J. Weimer, B. Sinopoli, and B. H. Krogh, "Multiple Source Detection and Localization in Advection-Diffusion Processes Using Wireless Sensor Networks," *Proc. 30th IEEE Real-Time Syst. Symp. (RTSS)*, pp. 333-342, Washington, DC, 2009.

[81] D. Witten and R. Tibshirani, “Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data,” *Statistical Applications in Genetics and Molecular Biology*, 8(1), pp. 1-27, 2009.

[82] R. Xu and D. Wunsch, II, “Survey of Clustering Algorithms,” *IEEE Trans. Neural Netw.,* vol. 16, no. 3, pp. 645–678, May 2005.

[83] 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.

[84] Z. Yong, Z. Hui, G. Dongqiang, and W. Zhihua, “Determination of Chemical Point Source Using Distributed Algorithm in Sensors Network,” *Proc. 24th Chin. Control Decision Conf. (CCDC)*, pp. 3373–3377, May, 2012.

[85] O. Younis, M. Krunz, and S. Ramasubramanian, “Node clustering in wireless sensor networks: recent developments and deployment challenges,” *IEEE Networks*, no. 20, pp. 20–25, 2006.

[86] H. Zou, “The Adaptive Lasso and its Oracle Properties,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[87] H. Zou, T. Hastie, and R. Tibshirani, “Sparse Principal Component Analysis,” *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, 2006.

[88] H. Zou, and T. Hastie, “Regularization and Variable Selection via the Elastic Net,” *Journal of the Royal Statistical Society, Series B*, 67, pp. 301-320, 2005.

[89] S. Zhu, D. Wang, K. Yu, T. Li, and Y. Gong, “Feature Selection for Gene Expression Using Model-Based Entropy, *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 7, no. 1, pp. 25-36, Jan.-Mar. 2010.

[90] D. Zumoffen and M. Basualdo, “From Large Chemical Plant Data to Fault Diagnosis Integrated to Decentralized Fault Tolerant Control: Pulp Mill Process Application,” *Industrial and Engineering Chemistry Research*, vol. 47, pp. 1201-1220, 2007.

BIOGRAPHICAL STATEMENT

Jia Chen was born in Yingtan, China, in 1987. She received her B.S. degree from Southwest Jiaotong University, China, in 2009, her M.S. degree from University of Electronic Science and Technology of China, China, in 2012. She is currently a PhD student in the University of Texas at Arlington in Electrical Engineering. Her research interests focus on machine learning, signal processing, and ad hoc networks.