

Electronic Theses and Dissertations, 2020-

2021

Representative-based Big Data Processing in Communications and Machine Learning

Mohsen Joneidi
University of Central Florida

 Part of the [Electrical and Computer Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd2020>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Joneidi, Mohsen, "Representative-based Big Data Processing in Communications and Machine Learning" (2021). *Electronic Theses and Dissertations, 2020-*. 703.
<https://stars.library.ucf.edu/etd2020/703>

REPRESENTATIVE-BASED BIG DATA PROCESSING IN COMMUNICATIONS AND
MACHINE LEARNING

by

MOHSEN JONEIDI
M.S. Sharif University of Technology, 2012

A Dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2021

Major Professor: Nazanin Rahnavard

© 2021 Mohsen Joneidi

ABSTRACT

The present doctoral dissertation focuses on representative-based processing proper for a big set of high-dimensional data. Compression and subset selection are considered as two main effective methods for representing a big set of data by a much smaller set of variables. Compressive sensing, matrix singular value decomposition, and tensor decomposition are employed as powerful mathematical tools to analyze the original data in terms of their representatives. Spectrum sensing is an important application of the developed theoretical analysis. In a cognitive radio network (CRN), primary users (PUs) coexist with secondary users (SUs). However, the secondary network aims to characterize PUs in order to establish a communication link without any interference with the primary network. A dynamic and efficient spectrum sensing framework is studied based on advanced algebraic tools. In a CRN, collecting information from all SUs is energy inefficient and computationally complex. A novel sensor selection algorithm based on the compressed sensing theory is devised which is compatible with the algebraic nature of the spectrum sensing problem. Moreover, some state-of-the-art applications in machine learning are investigated. One of the main contributions of the present dissertation is the introduction a versatile data selection algorithm which is referred as spectrum pursuit (SP). The goal of SP is to reduce a big set of data to a small-size subset such that the linear span of the selected data is as close as possible to all data. SP enjoys a low-complexity procedure which enables SP to be extended to more complex selection models. The kernel spectrum pursuit (KSP) facilitates selection from a union of non-linear manifolds. This dissertation investigates a number of important applications in machine learning including fast training of generative adversarial networks (GANs), graph-based label propagation, few shot classification, and fast subspace clustering.

TABLE OF CONTENTS

LIST OF FIGURES	ix
LIST OF TABLES	xvii
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: BACKGROUND	7
Singular Value Decomposition	8
Tensor Decomposition	9
Spectrum Sensing	12
Column Subset Selection Problem	15
Selection Based on Diversity	15
Representative Selection	17
CHAPTER 3: TENSOR-BASED SPECTRUM CARTOGRAPHY	18
Related Work and Preliminaries	20
CP Decomposition	22
Spline-based Surface Interpolation	23

The TRASC Algorithm	25
Alternating Least Squares Solution	30
Least Square Solution with Missing Entries	32
Implementation of TRASC for Power Map Reconstruction	33
Rank Estimation in a Dynamic Environment	34
Experiments	35
Conclusion	42
 CHAPTER 4: E-OPTIMAL SENSOR SELECTION	 44
Motivation	46
Problem Statement and Related Work	49
Convex Relaxation	50
Greedy Algorithms	51
Matrix subset selection	52
E-optimal sampling	55
RIP-Based Sensor Selection	58
Distributed Implementation	64
Successive Processing	66

Random Partitioning	67
Designed Partitioning	68
Reliability Estimation and Dynamic Sensor Selection	70
Optimization and Complexity	73
Experimental Results	76
Sensor Selection in CRNs	76
Data Selection for Supervised Learning	81
Synthesized Data	88
Dynamic Sensor Selection	91
Conclusion	95

CHAPTER 5: SPECTRUM PURSUIT: DATA REDUCTION BASED ON PRESERVING SPECTRAL STRUCTURE

Problem Statement and Related Work	100
Selection Based on Diversity	101
Representative Selection	102
Spectrum Pursuit (SP): Our Proposed Selection Method	103
The Spectrum Pursuit Algorithm	104

Sequential SP	106
A Lower Bound on Maximum Correlation	107
Upper Bound on Projection Error	109
Robustness to Perturbation	110
Residual Descent Implementation	111
Kernel SP: Selection based on a Locally Linear Model	113
Conclusion	115
 CHAPTER 6: APPLICATIONS OF DATA SUBSET SELECTION	 117
Representatives for Multi-PIE Dataset	117
Representatives To Generate Multi-view Images Using GAN	118
Representatives for ImageNet	121
Fast Subspace Clustering	123
Conclusion	125
 CHAPTER 7: MULTI-WAY SELECTION	 126
Two-way Spectrum Pursuit	129
N-way Spectrum Pursuit	132
Experimental Results	136

CUR Decomposition on Synthetic Data	136
Joint Sensor Selection and Channel Assignment	137
Informative Users/Contents Detection	139
Supervised Sampling	140
Experiments on Self-CP Decomposition	142
Conclusion	143
LIST OF REFERENCES	144

LIST OF FIGURES

2.1	An example illustrating the advantage of tensor vs. matrix decomposition.	11
2.2	A simple setup example for a cognitive radio network.	14
3.1	Schematic of CP decomposition to summation of rank-one tensors.	22
3.2	framework of the proposed joint tensor decomposition and surface interpolation. This scheme only shows a big picture of the proposed work. Practically, CP de- composition is implemented iteratively. In the proposed algorithm, interpolation is performed alongside iterations of CP decomposition. In other words, we solve a joint problem of decomposition and interpolation.	30
3.3	The practical implementation of TRASC with adaptive rank estimation in a dynamic radio environment.	35
3.4	Comparison between the original and the recovered spectrum maps. (a) The origi- nal power spectrum map. The grid is 50×50 , however, 6 columns and 6 rows of the original power spectrum map are measured. The power spectrum is measured at these locations only. (b) The recovered spectrum from missing and noisy sensed data using a plain CP decomposition and interpolating the unread measurements via CP reconstruction. (c) The recovered spectrum via block-term decomposition. (d) The plain 2D plate splines method is employed for interpolating the power spec- trum map. (e) The proposed method in [1] via the block-term decomposition which post-processed using 2D plate splines. (f) Our proposed method that employs CP decomposition and 2D plate splines jointly.	37

3.5	The interpolated spatial components using our proposed framework. The rank of CPD is assumed to be 2. A linear combination of these two factors is able to reconstructs the power spectrum map in any frequency band.	38
3.6	Spectrum map reconstruction error for several algorithms. (a) The rank for the tensor-based methods is assumed to be the best possible rank. (b) The rank for the tensor-based methods is assumed to differ from the best possible rank by +1. . .	39
3.7	The convergence behavior of the proposed framework. In each iteration of the proposed algorithm, all tensor CP factors are updated and a more accurate model is estimated for reconstruction of power spectrum map.	40
3.8	(a) The effect of assumed rank on the residual error of cost function (3.11). (b) Sensitivity to the proposed framework w.r.t. the assumed CP rank.	41
3.9	Sensitivity of BTM to the assumed rank.	41
3.10	(a) The number of structured measurements versus the normalized cartography error. (b) The number of random measurements versus the normalized cartography error. .	41
3.11	The impact of the smoothness parameter in 2D splines interpolation on the overall performance of the proposed framework.	42
3.12	The performance of the proposed framework in presence of moving sources in terms of the normalized cartography error. The speed is denoted as the number of paved grid points in T time slots where $T = 50$ time slots are considered.	42

4.1	Comparison of D-optimality and E-optimality for selecting 2 sensors in the 3D space. The gray area is the maximum achievable area by selecting the second sensor based on D-optimality. The shaded area is a well-shaped polygon obtained by E-optimality.	54
4.2	The main framework of the proposed reliability based sensor selection.	73
4.3	An example setup with 25 candidate points as transmitters.	77
4.4	Performance of different sensor selection algorithms in terms of number of selected sensors	78
4.5	Performance of the selection algorithm in presence of 0dB AWGN.	78
4.6	Performance of blind and data-aware RIP based sensor selection algorithms in terms of number of selected sensors.	79
4.7	Performance of our Data-aware algorithm.	80
4.8	The true spectrum in the area of interest along the selected sensors obtained by 3 methods in spatial domain	81
4.9	The error of estimated spectrum in the area of interest corresponding to Fig. 4.8. (Left) RIP based, Algorithm 1. (Middle) Online RIP based, Algorithm 2 while only 5% of sensors are sensed. (Right) Centralized RIP based, Algorithm 2 while all the sensors are sensed. λ is assumed 0.7	81
4.10	The projection error of the training data into the subspace spanned by the selected rows.	82
4.11	The projection error of the test data into the subspace spanned by the selected rows. .	84

4.12	Training data corresponding to the third subject of Extended Yale-B data set. This data set contains different angles of shadowing for each subject.	85
4.13	Comparison of the proposed E-optimal representatives versus K-medoids selection. .	85
4.14	12 selected images of digit 4 from 5842 images.	86
4.15	Performance of nearest subspace classifier learned by few data from each class. . . .	86
4.16	Running time of selecting few data from each class.	87
4.17	Error of subspace identification using selection.	89
4.18	Performance of the distributed selection algorithm in terms of number of distributed nodes.	90
4.19	Complexity of the distributed selection algorithm in terms of number of distributed nodes.	90
4.20	Performance of different static sensor selection algorithms in terms of number of selected sensors.	92
4.21	Performance of static and dynamic E-optimal-based sensor selection algorithms vs. the number of selected sensors.	93
4.22	The error of estimated spectrum in the area of interest. (Left) E-optimal, Algorithm 1. (Middle) Reliable E-optimal, Algorithm 2 after sensing in one time block. (Right) Reliable E-optimal, Algorithm 2 while all the sensors are sensed after 30 time blocks. γ is assumed 0.7	93
4.23	MSE error versus different values of γ	93

4.24	Reliability maps of 4 time blocks illustrate how the proposed framework evolves in time in order to select adapted sensors to the dynamic of network after state transition. Sensors within unreliable (red) areas have more chance of selection.	94
4.25	(a) A dynamic network with 3 states for the location of active PUs. The shaded blue squares represent active PUs. (b) The effect of reliable sensor selection for compensation of the model error in the reliable sensor selection procedure.	95
4.26	The effect of reliable sensor selection for compensation of the model error in the reliable sensor selection procedure.	95
5.1	A dataset consisting of 20 real images is considered as blue dots. The best possible subspace that spans the dataset is shown in green. However, the significant eigenfaces (green dots) are not among the dataset. We look for the best 3 out 20 real images whose span is the closest to the span of 3 green eigenfaces; the best subspace is shown in blue. There are $\binom{20}{3}$ possible combinations from which the best representatives must be selected.	100
5.2	Two consecutive iterations of SP algorithm. The first LSV of the residual matrix is a vector on \mathbb{S}^{N-1} and the goal is to find K samples which pursuit the spectral characteristics of dataset over iterations.	105
5.3	A toy example that illustrates the first iteration of the sequential SP (IPM). (Left) The most matched sample with the first left singular vector, v , is selected. (Right) The rest of samples are projected on the null space of the selected sample in order to continue selection in the lower dimensional subspace.	108

5.4	SP-RD algorithm does not select directly the most correlated sample with the first left singular vector. First, a small subset of samples which are correlated with the first left SV are grouped. Then, the sample which is the best minimizer for (5.7c) is selected.	112
5.5	(a) A dataset lies on a two dimensional manifold identified by two parameters, rotation and size. However, the rank of corresponding matrix to this dataset is a large number. (b) Linear embedding using linear PCA and selection using linear SP. (c) nonlinear embedding using tSNE[2] and selection using kernel-SP. Un-selected and selected samples are shown as red and black dots in the embedded space, respectively.	115
6.1	Selection of 10 representatives out of 520 images of a subject. IPM and SP select from more diverse angles.	117
6.2	Performance of different methods in terms of their accuracy for CSSP defined in (5.1). The proposed SP algorithm is compared with IPM [3], and DS3 [4], FFS [5], SMRS [6], 2phase [7], K-medoids [8] and volume sampling [9]. (Top) The ratio of projection error using selection algorithms to projection error of random selection for selecting K representatives from each subject, averaged over all the subjects. (Bottom) Running time of different algorithms versus number of input samples for selecting 10 samples. Our SP algorithm is slower than IPM, however, it is more accurate.	119

6.3	The ratio of projection error as a function of K selected samples for each class of Multi-PIE dataset. It is averaged for all 250 classes and the running time for selecting from the whole dataset is reported in the parenthesis. IPM and SP do not require any parameter, and Parameter P in SP-RD algorithm does not need tuning. It can be fixed according to the accessible computational power.	119
6.4	Multi-view face generation results for a sample subject in testing set using CR-GAN [10]. The network is trained on a selected subset of training set (9 images per subject) using random selection (first row), K-medoids (second row), DS3 [4] (third row), and SP (fourth row). The fifth row shows the results generated by the network trained on all the data (360 images per subject). SP generates closest results to the complete dataset.	120
6.5	Two approaches for subspace clustering. (a) Identify subspaces directly from ensemble of data. (b) First select a set of representatives and then identify subspaces accordingly. Our selection algorithm facilitates the second approach.	125
6.6	Accuracy of clustering of two synthetic clusters contaminated with noise and containing 10% outlier samples. Clustering of full data results in 96.47% accuracy using SSC [11].	125
7.1	Two columns and three rows from matrix \mathbf{X} are selected and organized in matrix \mathbf{C} and \mathbf{R} . The outer product of each pair of a selected column and a selected row constructs a rank-1 matrix, i.e., $\mathbf{c}_i \mathbf{r}_j^T$. The contribution amount of each pair is reflected in variable u_{ij} . The core matrix \mathbf{U} is the collection of all u_{ij} 's. The goal is to minimize $\ \mathbf{X} - \hat{\mathbf{X}}\ _F$ where $\hat{\mathbf{X}} = \mathbf{CUR}$	127

7.2	(a)-(d) Performance comparison in terms of the normalized error of CUR decomposition. (e) Convergence behavior of the proposed two-way SP for selecting 20 columns and 20 rows.	137
7.3	The behavior of the proposed algorithm w.r.t. the initial condition of selected subset. The initial cost function are corresponding to the initial set which are drawn randomly. The blue curves indicate the path of optimization alongside iterations of the TWSP algorithm. Here, 100 different realizations are studied.	137
7.4	The original spectrum map and its comparison with the interpolated map using sampled random sampling and our proposed method. The interpolation error is depicted versus number of sensed locations.	138
7.5	Comparison of the normalized prediction error with state-of-the-art algorithms obtained by CUR decomposition for simultaneous movies and users subset selection from Netflix dataset.	139
7.6	The classification accuracy of a fine tuned Resnet-18 network using a few selected data per each class.	141
7.7	Test accuracy in terms of improvement comparison with the test accuracy obtained by random selection.	142
7.8	Test accuracy in terms of improvement comparison with the test accuracy obtained by random selection.	143

LIST OF TABLES

4.1	Complexity of different selection strategies.	57
4.2	Accuracy of different classifiers using partial data for learning of Extended Yale-B dataset with 5 representatives.	84
4.3	Performance of selection algorithms in terms percentage of classification rate using a deep neural network learned by partial data. The original data set contains 60,000 training images. (Left) K-medoids, (Middle) two-phase. (Right) Random Selection. Each classifier is learned by only 200, 500, 1000 and 2000 training data out of 60,000. The performance of MLP using all 60,000 samples is 98.25 and it is 99.66 for the CapsNet architecture.	88
4.4	Running time (seconds) of data selection corresponding to Table 4.3.	88
4.5	Running time (seconds) of neural network learning corresponding to Table 4.3. Running time per epoch is reported.	88
6.1	Identity dissimilarities between real and generated images by network trained on reduced (using different selection methods) and complete dataset.	121
6.2	Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.	121

6.3	Top-1 classification accuracy (%) on ImageNet, using selected representatives from each class. Accuracy using all the labeled data (~ 1.2 M samples) is 46.86%. Numbers in () show the size of the selected representatives as a % of the full training set.	122
6.4	Unsupervised clustering accuracy for MNIST handwritten dataset.	124

CHAPTER 1: INTRODUCTION

Complex systems containing very large numbers of data-gathering devices have been developed in the last decade. The amount of gathered data in data-driven systems is expanding in an astonishing rate in the recent years. International Data Corporation (IDC) predicts the global data volume will grow to the order of ($\sim 10^{23}$ bytes) by 2025 [12]. However, dealing with a large number of sources of data is challenging especially when each data point belongs to a high-dimensional measurement. The emerging research area, *big data*, aims to address challenges of such complex systems. Representing the underlying structure of data by a succinct format is a crucial issue in the big data literature. For instance, dimension reduction techniques, compression, and different clustering-based approaches aim to extract a concise representation of data. A robust representation should be informative enough for reconstructing the original data given the representatives. Two popular representations are considered in this dissertation, 1) compression and 2) selection. Chapter 2 reviews basic data representation methods.

Compressive sensing (CS) has been a popular sampling scheme in the last decade. In CS, a compressed replica of the unknown variables is sensed via a measurement matrix. The unknown variables are organized in a vector and the sensed measurements can be interpreted as a low-dimensional representation of the unknown vector. Some efforts are conducted to sense an unknown matrix under the context of matrix sensing. In this dissertation multi-way sensing for reconstruction of a multi-way tensor given a compressed replica of the tensor of interest is studied. The proposed approach is employed for dynamic spectrum sensing in presence of collaborative sensor networks that are communicating over multi-band frequencies over time. Spatial, spectral and temporal occupancy patterns of the network are projected in entries of an unknown tensor. However, a compressed replica of the tensor of interest is available to sense. Since there exist redundancies in the original tensor, the compressed replica contains sufficient information in order

to reconstruct the original tensor. In Chapter 3 a tensor-based compressive sensing solution for spectrum sensing is presented.

Removing redundant sensors is an alternative for compression in order to exploit the underlying redundancy in sensor networks. This approach can be conducted by sensor selection. In the present dissertation, both approaches are studied jointly. In Chapter 4 the problem of E-optimal sensor selection for compressive sensing-based purposes is presented. Our ultimate goal is to reduce a big set of equations to a selected set of equations. This selection is conducted such that informative equations are remained. In other words, the solution of the big systems of equations before reduction will be as close as possible to that of resulted by the subset of selected equations.

Representatives obtained by compression or algebraic decomposition methods are often not easy to interpret. Furthermore, obtaining each representative implies processing all data or a large portion of data. In order to have a straightforward interpretation, it is desired to find the representatives by selection from data. There are some clustering approaches that select the representatives from data such as k-medoids clustering [13]. However these clustering methods assign each data to only one prototype which is the cluster centroid, while in the case of highly structured data only one prototype from data does not contain sufficient information to capture the underlying structure of the whole cluster. In this dissertation, it is shown that how we can select a subset of data such that their linear combination is able to approximate the ensemble of data. Moreover, the linear subspace spanned by selected data is extended to a non-linear manifold that encompasses the selected data.

Data-driven systems based on data reduction that make the best use of a significantly less amount of data are of great interest. For example, active learning (AL) [14] aims at addressing informative sample selection by training a model using a small number of labeled data, evaluating the trained model, and then querying the labels of selected representatives, which are used later for training a new model. In this context, preserving the underlying structure of data succinctly by

representatives is an essential concern. Although each application requires specific considerations, regardless of the underlying application a versatile cost function for selection can be devised. The versatile selection problem can be customized in order to be adapted to the target application. Two main general cost functions are studied in this dissertation. 1) Diversity-based selection and 2) Representative-based selection. The first track finds a subset of data such that they are as diverse as possible. The found solution using the second direction provides a subset which does not have necessarily diverse samples. However, the selected samples are accurate representatives for all samples including selected and unselected ones.

In some applications such as sensor selection from scattered sensors or sensor placement in an area, diversity-based selection is desired and there is no advantage to cover unselected sensors by employing a representative-based selection. On the other hand, in some applications such as training a machine learning system or video summarization it is essential that selected samples are suitable representatives from all data including unselected samples.

An example of big data system is wireless sensor networks, where the processing unit has to deal with an excessively large number of observations acquired by the various sensors. Often there exist some redundancies within the sensed data and they should be pruned. Sensor selection and sensor scheduling aim to address this problem. In many applications the sensor selection task is non-trivial and possibly consists of addressing an NP-hard problem (i.e., there are $\binom{M}{K}$ possibilities of choosing K distinct sensors out of M available ones). This essentially implies that an optimal solution cannot be efficiently computed, in particular when the number of sensors becomes excessively large. A convex relaxation of the original NP-hard problem has been suggested in [15]. The most prominent advantage of this approach over other methods is its practicality, thanks to many well-established computationally-efficient convex optimization techniques. In addition to convex relaxation, a sub-modular cost function as the criterion of sensor selection allows us to take advantage of greedy optimization methods for selecting sensors [16]. The existing studies on

sensor selection mostly consider heuristic approaches. For example, in [15] the volume of the reduced bases is considered. This method is called *D-optimality*. In addition, *A-optimality* [17] and *E-optimality* [17] are suggested as some other alternative heuristics already introduced in convex optimization. These heuristics are presented without any specific justification for sensor selection application. In Chapter 3 we are going to exploit E-optimal criteria more judiciously in favor of compressed sensing (CS) theoretical guarantees. The proposed approach is employed for dynamic sensor selection in cognitive radio networks. Moreover, a theoretical investigation is presented for the performance bound of E-optimal subset selection.

Dimension reduction techniques and clustering-based approaches aim to extract a concise representation of data. However, representatives or exemplars obtained by such methods are not helpful for selection an informative subset. Furthermore, obtaining each representative implies processing all or a large portion of data. Thus, it is desired to optimally select the representatives from data samples. There are some clustering approaches that select the representatives from data such as the k-medoids clustering [13]. These clustering methods assign each data sample to only one prototype which is the cluster center. However, in the case of more structured data only one prototype from data does not contain sufficient information to capture the underlying structure of the whole cluster. Randomly selecting K out of M data, while computationally simple, is inefficient in many cases, since non-informative or redundant instances may be among the selected ones. On the other hand, the optimal selection of data for a specific task implies solving an NP-hard problem [18]. For example, finding an optimal subset of K data samples from M to be employed in training a Deep Learning (DL) network with the best performance requires $\binom{M}{K}$ number of trial and errors, which is not tractable. It is essential to define a versatile objective function and to develop a method that efficiently selects the K samples that optimize the objective function. Let us assume the M data samples are organized as the columns of a matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$. The following is a general purpose cost function for subset selection, known as *column subset selection problem* (CSSP), which is an

open problem [19]:

$$\mathbb{S}^* = \underset{|\mathbb{S}| \leq K}{\operatorname{argmin}} \|\mathbf{A} - \pi_{\mathbb{S}}(\mathbf{A})\|_F^2, \quad (1.1)$$

where $\pi_{\mathbb{S}}$ is the linear projection operator on the span of K columns of \mathbf{A} indicated by set \mathbb{S} . This problem has been shown to be NP hard [18, 20]. Moreover, the cost function is not sub-modular [21] and greedy algorithms are not efficient to tackle Problem (1.1). Computer scientists and mathematicians during the last 30 years have proposed many tractable selection algorithms that guarantee an upper bound for the projection error $\|\mathbf{A} - \pi_{\mathbb{S}}(\mathbf{A})\|_F^2$. These works include algorithms based on QR decomposition of matrix \mathbf{A} with column pivoting (QRCF) [22, 23, 24]; methods based on volume sampling (VS) [25, 26, 27] and matrix subset selection algorithms [19, 28, 29]. However, the guaranteed upper bounds are very loose and the corresponding selection results are far from the actual minimizer of CSSP in practice. Interested readers are referred to [30, 28] and Sec. 2.1 in [31] for detailed discussions. For example, in VS it is shown that the projection error on the span of K selected samples is guaranteed to be less than $K + 1$ times of the projection error on the span of the K first left singular vectors; which is too loose for a large K . Recently, it was shown that VS performs even worse than random selection in some scenarios [32]. Moreover, some efforts have been made using convex relaxation and regularization. Fine tuning of these methods is not straightforward [6, 4, 33]. Moreover their cubical complexity is an obstacle to employ these methods for diverse applications.

In this work a new fast and accurate solution for Problem (1.1) is proposed with a linear time complexity. This algorithm is referred as spectrum pursuit (SP) which is presented in Chapter 5. Theoretical performance bounds of SP are also investigated. It is shown that the proposed approach reaches the best performance of subset selection in two asymptotic cases. SP has no parameter to be fine tuned and this desirable property makes it problem-independent. The simplicity of SP enables

us to extend the underlying linear model to more complex models such as nonlinear manifolds and graph-based models. The nonlinear extension of SP is introduced as *kernel-SP* (KSP) in Chapter 5. The superiority of the proposed algorithms is demonstrated in a wide range of applications.

In Chapter 6 some interesting applications of SP and KSP algorithms are presented. These applications include 1) Training GAN using reduced data, 2) Fast deep neural networks training, 3) fast subspace clustering, 4) few shot learning, 5) graph central node selection, 6) semi-supervised label propagation, and 7) open-set identification.

Chapter 7 presents a novel paradigm for multi-way selection from high-dimensional data structures. Joint selection of columns and rows from a matrix is a simple example of multi-way selection. The problem formulation and its practical solution are presented in addition to few interesting applications.

CHAPTER 2: BACKGROUND

The underlying investigated problems in the present dissertation are introduced in this section. First the tensor decomposition problem is presented. then spectrum sensing problem is introduced and a solution using low-dimensional representation is explained. The problem of sensor selection is reviewed and two main selection strategies are studied.

Notations: Throughout this dissertation, vectors, matrices, and tensors are denoted by bold lowercase, bold uppercase, and bold underlined uppercase letters, respectively. The positive orthant in \mathbb{R}^P is denoted by \mathbb{R}_+^P and it is defined as $\{\mathbf{x} | x_p \geq 0, \forall p = 1, \dots, P\}$, in which x_p is the p^{th} element of \mathbf{x} . If $\underline{\mathbf{T}} \in \mathbb{R}^{P \times F \times T}$ then $(\underline{\mathbf{T}})_{:,f,t}$ is a vector of length P , also known as a mode-1 fiber of $\underline{\mathbf{T}}$, defined by fixing all the indices but one. Similarly, we have mode-2 and mode-3 fibers. $\mathbf{T}_1, \mathbf{T}_2$, and \mathbf{T}_3 are unfolded matrices whose columns are fibers of the first, second and third mode of $\underline{\mathbf{T}}$, respectively. The Khatri-Rao product is denoted by \odot . Moreover, \circ denotes the outer product, i.e., entries of $\underline{\mathbf{T}} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ are calculated as $t_{pft} = a_p b_f c_t$. The outer product of two non-zero vectors is a rank-1 matrix, similarly the outer product of three non-zero vectors is a rank-1 tensor. The symbol $*$ denotes the element-wise (Hadamard) product. The n-mode product of a tensor, $\underline{\mathbf{X}}$, with a proper sized transformation matrix \mathbf{U} is a tensor and is denoted by $\underline{\mathbf{X}} \times_n \mathbf{U}$. It transfers each fiber of the n^{th} mode of the tensor to the corresponding fiber in the output tensor. Mathematically,

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_n \mathbf{U} \leftrightarrow \mathbf{Y}_n = \mathbf{U} \mathbf{X}_n \text{ for } n = 1, 2, 3,$$

Singular Value Decomposition

Throughout the present dissertation, we will see how we can decompose a high-dimensional and complicate-structured array into a set of low-dimensional and simple-structured components. The main goal behind these decomposition techniques is one simple idea. The number of unknown variables can be much lower by the low-dimensional representation. Thus, in presence of limited observations for estimating a set of unknown variables, a low-dimensional representation is a clever substitution in order to get a robust solution. There are several methods for decomposition of a matrix including singular value decomposition (SVD), QR decomposition, CUR decomposition and lower-upper decomposition. Here, we introduces SVD as the main tool for matrix decomposition which has three main favorable properties. (i) Any matrix with any size has a unique SVD. (ii) Its computation is linear w.r.t. size and it is proportional to the number of simple-structured components. (iii) It can be implemented in a recursive manner since components are orthogonal to each other. For example, finding two optimal components is equivalent to adding 1 optimal component to the solution for finding the best optimal component.

SVD has a tight relation with Eigen decomposition. However, Eigen decomposition is defined only for squared matrices. Assume matrix \mathbf{X} which is squared is written as $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$. Columns of matrix \mathbf{Q} are known as eigenvectors of \mathbf{X} . Diagonal entries of matrix $\mathbf{\Lambda}$ are known as eigenvalues. Since \mathbf{Q}^{-1} cancels the scale of each eigenvector, without loss of generality we can assume that eigenvectors are normal. If matrix \mathbf{X} is symmetric, eigenvectors will be orthogonal to each other. Then, \mathbf{Q}^{-1} will be equal to \mathbf{Q}^T . Now, we are able to define SVD of a general matrix \mathbf{X} with any size as follows,

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (2.1)$$

Here, the columns of matrix \mathbf{U} are eigenvectors of $\mathbf{X}\mathbf{X}^T$ which also are referred as left singular

vectors of \mathbf{X} . Similarly, the columns of matrix \mathbf{V} are eigenvectors of $\mathbf{X}^T \mathbf{X}$ which also are referred as right singular vectors of \mathbf{X} . It is easy to show that non-zero eigenvalues of $\mathbf{X}^T \mathbf{X}$ are equal to those of $\mathbf{X} \mathbf{X}^T$. Diagonal elements of matrix $\mathbf{\Sigma}$ are square root of eigenvalues of $\mathbf{X}^T \mathbf{X}$ which also called singular values. Since $\mathbf{X}^T \mathbf{X}$ is a positive-definite matrix, all of its eigenvalues are non-negative. Thus, all of singular values also are real and non-negative. SVD plays an important role in our proposed algorithms in practical applications of the present dissertation. The extension of SVD for higher dimensional tensors is discussed in the next section.

Span of a set of vectors is defined as the subspace that can be represented as a linear combination of that set of vectors. Let us split columns of \mathbf{U} into two separate matrices as \mathbf{U}_{\parallel} and \mathbf{U}_{\perp} . Columns corresponding to non-zero singular values are collected in \mathbf{U}_{\parallel} and that columns corresponding to zero singular values are collected in \mathbf{U}_{\perp} . Similarly, let us define \mathbf{V}_{\parallel} and \mathbf{V}_{\perp} by splitting columns of \mathbf{V} . All columns of a matrix are within the span of \mathbf{U}_{\parallel} in its SVD. Similarly, all rows of a matrix are within the span of \mathbf{V}_{\parallel} in its SVD. The span of \mathbf{U}_{\perp} is referred as the null-space from the column perspective and the span of \mathbf{V}_{\perp} is the null-space from row perspective. In other words, if we pick a vector from span of \mathbf{U}_{\perp} or span of \mathbf{V}_{\perp} , it will be orthogonal to all columns or all rows of the matrix, respectively.

Tensor Decomposition

Tensor-based methods have been employed in communications and coding frameworks since Sidiropoulos et. al. introduced them for blind code-division multiple access (CDMA) [34]. Recently, more advanced coding methods in communication systems are developed based on the tensor decomposition theory [35, 36, 37]. Moreover tensors are employed for massive MIMO channel estimation in millimeter wave scenarios [38]. The common idea of all tensor-based coding methods is to perform a joint blind symbol and channel estimation at the receiver, which is feasible due to mild

conditions for uniqueness of tensor CP decomposition [39].

Utilizing another signal processing model on top of the CP tensor model has also been studied extensively. For example, a joint problem of sensing and sparsifying for tensor compressive sensing is proposed in [40] and a joint encryption and compression method is proposed for medical image compression in [41]. In this dissertation, joint factorization of the sensed tensor to a latent tensor and a non-negative transformation matrix is studied. This problem is a generalization of three-way compressed sensing [42], when the transformation (sensing matrix) is not known and is not necessarily a compression matrix.

Our proposed tensor-based approach is mainly based on the CP decomposition [43], which factorizes a tensor into a sum of rank-one tensors. For example, a three-way tensor $\underline{\mathbf{X}} \in \mathbb{R}^{P \times F \times T}$ of rank R can be decomposed as

$$\underline{\mathbf{X}} = \sum_{r=1}^R \mathbf{a}_r^X \circ \mathbf{b}_r^X \circ \mathbf{c}_r^X = \llbracket \mathbf{A}_X, \mathbf{B}_X, \mathbf{C}_X \rrbracket, \quad (2.2)$$

where $\mathbf{a}_r^X \in \mathbb{R}^P$, $\mathbf{b}_r^X \in \mathbb{R}^F$ and $\mathbf{c}_r^X \in \mathbb{R}^T$ are *factor vectors* of the r^{th} rank-one component. The *factor matrices* refer to the collection of factor vectors from the rank-one components, i.e., $\mathbf{A}_X = [\mathbf{a}_1^X \ \mathbf{a}_2^X \ \dots \ \mathbf{a}_R^X]$ and likewise for \mathbf{B}_X and \mathbf{C}_X . The kruskal-rank, which also is referred to as k-rank, is a well-known criterion for deriving conditions on the uniqueness of tensor decomposition.

Figure 2.1 shows a $2 \times 2 \times 2$ tensor and provides a toy example that gives us an idea on the advantage of tensors over matrices. This tensor can be regarded as two consecutive 2×2 images in time in which a pixel is moving from location (1,1) to (2,2). Rank of this tensor is 2.¹ Decomposition to summation of rank one tensors is able to bring us 2 components. However, im-

¹ Similar to matrices, rank function is defined for tensors as the minimum number of rank-1 tensors that reconstructs the tensor. However, characteristics of tensor rank are quite different. Tensor rank varies based on the assumed structure on CP factors. Generally, rank of a tensor refers to unconstrained rank in which no structure is assumed for the factors, however, rank of a matrix refers to the spectral decomposition in which orthogonality is assumed implicitly.

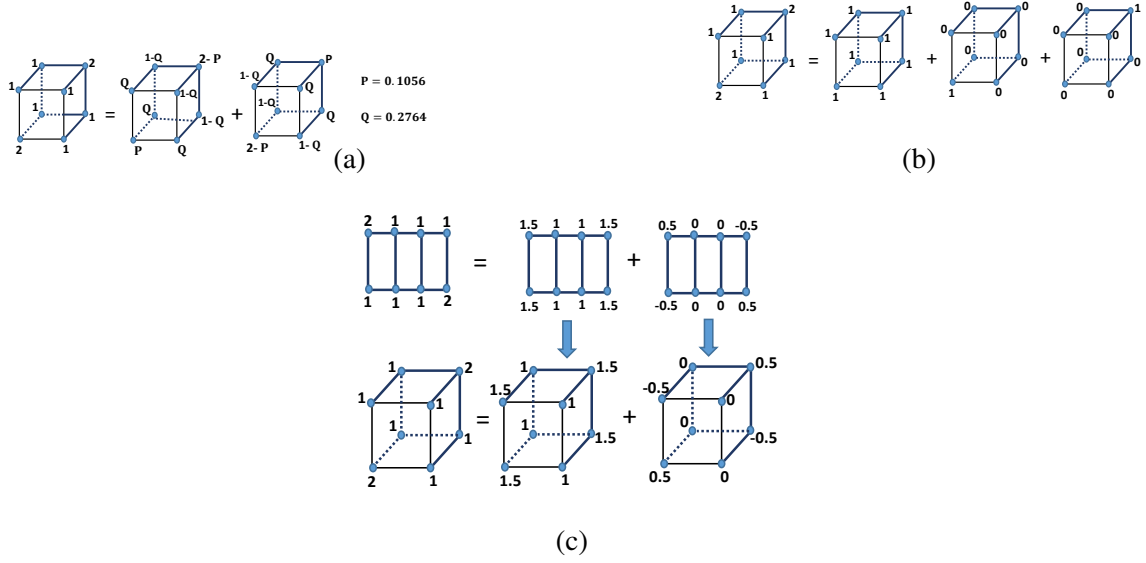


Figure 2.1: An example illustrating the advantage of tensor vs. matrix decomposition.

posing a proper structure increases the minimum number of required components. For example sparsity on CP factors come up with 3 distinguished shown tensors. Matricization of the given data brings us a 4×2 matrix. Clearly, the maximum rank is 2 and we are able to extract two rank one matrices. These rank one matrices are reshaped to their original 3-D locations in the tensor. As shown, the extracted rank one components from the corresponding matricization of the tensor are difficult to interpret while the structured rank one tensors extracted by the tensor decomposition can be interpreted simply to a background (all-one tensor) and two objects. In the simple example of Figure 2.1, the given tensor can be decomposed to the following three factors,

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Advantages of tensor-based processing over matrix-based processing:

- Multi-dimensional structures of data can be discovered by exploiting the tensor representation. Such structures cannot be easily exploited in matrix-based data analysis.

- As the inherent structures of data is preserved in tensor representation, we can easily exploit prior knowledge and the constraints imposed by the dynamics of the problem to achieve more accurate analysis in a concise mathematical formulation.
- Uniqueness of CP decomposition for multi-dimensional tensors is guaranteed under milder conditions as the dimension of tensor increases.

Spectrum Sensing

Consider an area of interest which contains up to P active PUs. We have deployed $N > P$ spectrum sensors in the area to measure the power of received signals at different frequency bands. The channel gain between the p^{th} PU and the n^{th} SU is denoted by γ_{np} . The bandwidth is broken into F frequency channels. For the sake of conciseness, assumptions for channel gains and signals of sources are borrowed from [44]. The received PSD at SU/sensor n at time slot t and frequency channel f , $y_n(t, f)$, can be written as follows under a set of mild conditions [44]²,

$$\begin{aligned} y_n(t, f) &= \sum_{p=1}^P \gamma_{np} x_p(t, f) + z_n(t, f) \\ &= \boldsymbol{\gamma}_n^T \mathbf{x}(t, f) + z_n(t, f), \quad t = 1, \dots, T \text{ and } f = 1, \dots, F. \end{aligned} \quad (2.3)$$

In this equation $\boldsymbol{\gamma}_n = [\gamma_{n1} \ \gamma_{n2} \ \dots \ \gamma_{nP}]^T$ is the set of channel gains between all PUs and the n^{th} sensor. The received noise at n^{th} sensor is represented by $z_n(t, f)$. Moreover, $\mathbf{x}(t, f) = [x_1(t, f) \ \dots \ x_P(t, f)]^T$ is the propagating power from each of P active PUs at time t and frequency bin f . Due to the non-negative nature of sensed power spectrum data, all the variables are considered to be non-negative. Let $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_{np}] \in \mathbb{R}^{N \times P}$ denote the collection of channel gains be-

²Please note that duration of each time slot is much more than sampling time. For example, each time slot consists of 256 samples that enable us to obtain the spectrum for each time slot. Moreover, transmitted signals of PUs are assumed to be independent of each other.

tween locations of active PUs and those of sensors. Equation (2.3) can be cast in the tensor format as

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{\Gamma} + \underline{\mathbf{Z}}. \quad (2.4)$$

The measured spectrum data is presented by the tensor $\underline{\mathbf{Y}}$ and it is modeled by a mod-1 product between a non-negative tensor and a non-negative matrix plus the noise tensor, $\underline{\mathbf{Z}}$. Since location of SUs and PUs are unknown, $\underline{\mathbf{X}}$ and $\mathbf{\Gamma}$ must be estimated *jointly*. In practice, only tensor $\underline{\mathbf{Y}}$ is given and the integer variable P is assumed as the maximum number of PUs in the underlying model. It is shown that CP factors of $\underline{\mathbf{Y}}$ and CP factors of $\underline{\mathbf{X}}$ are related in the presence of no noise ($\underline{\mathbf{Z}} = 0$). In this case, CP factors corresponding to the second and third ways of these two tensors are equal. Further, the first factor matrix of $\underline{\mathbf{Y}}$, denoted by \mathbf{A}_Y , is equal to $\mathbf{\Gamma} \mathbf{A}_X$ [45]³. Taking the noise tensor into the account results in different factor matrices for tensor $\underline{\mathbf{Y}}$ and tensor $\underline{\mathbf{X}} \times_1 \mathbf{\Gamma}$. However, it is shown that CP decomposition is robust against additive noise, i.e., if the energy of additive noise is bounded, the difference of CP factors of two tensors is also bounded [46].

Fig. 2.2 shows a simple CRN where 7 SUs sense the propagated spectrum from 3 PUs. In this figure, matrix $\mathbf{\Gamma}$ corresponds to a 7×3 matrix. Unique identification of both $\underline{\mathbf{X}}$ and $\mathbf{\Gamma}$ is not a trivial task. However, an estimated local optimal solution for the joint optimization (2.4) can be employed to recover the missing spectrum and de-noise the sensed spectrum in tensor $\underline{\mathbf{Y}}$.

The following problem aims to find the CP factors of the latent tensor, $\underline{\mathbf{X}}$, and the non-negative

³In this section we deal with two main tensors, $\underline{\mathbf{Y}}$, and $\underline{\mathbf{X}}$. CP decomposition of these tensors are written as $[[\mathbf{A}_Y, \mathbf{B}_Y, \mathbf{C}_Y]]$, and $[[\mathbf{A}_X, \mathbf{B}_X, \mathbf{C}_X]]$, respectively. In the presence of no noise, $\mathbf{A}_Y = \mathbf{\Gamma} \mathbf{A}_X$, $\mathbf{B}_Y = \mathbf{B}_X$, and $\mathbf{C}_Y = \mathbf{C}_X$.

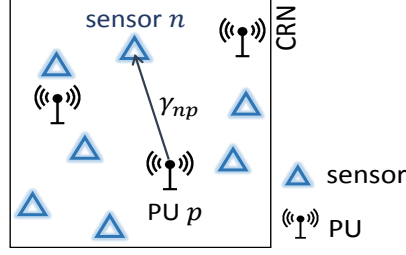


Figure 2.2: A simple setup example for a cognitive radio network.

channel gain matrix, $\mathbf{\Gamma}$,

$$\begin{aligned}
 (\hat{\mathbf{A}}_X, \hat{\mathbf{B}}_X, \hat{\mathbf{C}}_X, \hat{\mathbf{\Gamma}}) &= \underset{\mathbf{a}_r^X, \mathbf{b}_r^X, \mathbf{c}_r^X, \mathbf{\Gamma}}{\operatorname{argmin}} \|\underline{\mathbf{Y}} - \underline{\mathbf{X}} \times_1 \mathbf{\Gamma}\|_F^2 \\
 \text{subject to: } \underline{\mathbf{X}} &= \sum_{r=1}^R \mathbf{a}_r^X \circ \mathbf{b}_r^X \circ \mathbf{c}_r^X \\
 \mathbf{A}_X &\geq 0, \mathbf{B}_X \geq 0 \text{ and } \mathbf{C}_X \geq 0 \\
 \mathbf{\Gamma} &\geq 0.
 \end{aligned} \tag{2.5}$$

In this problem, $\mathbf{\Gamma}$ is an $N \times P$ matrix where $P < N$. The low-rank approximation of the sensed tensor, denoted by $\underline{\mathbf{Y}}^L$, can be estimated using the underlying low-rank model, i.e., $\underline{\mathbf{Y}}^L = \llbracket \hat{\mathbf{\Gamma}} \hat{\mathbf{A}}_X, \hat{\mathbf{B}}_X, \hat{\mathbf{C}}_X \rrbracket$. Entries of tensor $\underline{\mathbf{Y}}^L$ follow a low-rank pattern while tensor $\underline{\mathbf{Y}}$ corresponds to a high-rank tensor due to noisy measurements. This problem provides a unique solution under a set of conditions as will be discussed in the appendix. However, any stationary point for this problem provides a reconstructive model which can be employed for missing spectrum recovery and spectrum denoising. In other words, the set of CP factors and the channel gain matrix are a low-dimensional representation for $\underline{\mathbf{Y}}$. Contamination of tensor $\underline{\mathbf{Y}}$ by noise and missing entries does not affect the low-dimensional representation.

Column Subset Selection Problem

Let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M \in \mathbb{R}^N$ be M given data points of dimension N . We define an $N \times M$ matrix, \mathbf{A} , such that \mathbf{a}_m is the m^{th} column of \mathbf{A} , for $m = 1, 2, \dots, M$. The goal is to reduce this matrix into an $N \times K$ matrix, \mathbf{A}_R , based on an optimality metric. In this section, we introduce some related work on matrix subset selection and data selection.

Selection Based on Diversity

Consider a large system of equations $\mathbf{y} = \mathbf{A}^T \mathbf{w}$, which can be interpreted as a simple linear classifier in which \mathbf{y} is the vector of labels, \mathbf{A} represents the training data and \mathbf{w} is the classifier weights. An optimal sense for data selection is to reduce this system of equations to a smaller system, $\mathbf{y}_R = \mathbf{A}_R^T \hat{\mathbf{w}}$, such that the reduced subsystem estimates the same classifier as the original system, i.e., the estimation error of $\hat{\mathbf{w}}$ is minimized [47] over an assumed distribution for \mathbf{y} . A typical selection objective is to minimize $\|\mathbf{w} - \hat{\mathbf{w}}\|_2$. This criterion is referred to as the *A-optimal* design in the literature of optimization, which is mathematically equivalent to the following problem [48],

$$\begin{aligned} \hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmin}} \quad & \operatorname{tr} \left(\sum_{m=1}^M z_m \mathbf{a}_m \mathbf{a}_m^T \right)^{-1}, \\ \text{subject to} \quad & \|\mathbf{z}\|_0 = K \text{ and } \mathbf{z} \in \{0, 1\}^M, \end{aligned} \tag{2.6}$$

where $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_M]^T$ and z_m indicates the contribution of the m^{th} sample. According to the constraints only K samples can be selected in the reduced system. This is an NP-hard problem which can be solved via convex relaxation with computational complexity of $O(M^3)$ [49].

However, there are other criteria that have some interesting properties. For example *D-optimal*

design optimizes the determinant of a reduced matrix [49]. There are several other efforts in this area [9, 27, 26, 50, 51]. Inspired by the D-optimal design, volume sampling (VS), which has received lots of attention, considers a selection probability for each subset of data, which is proportional to the determinant (volume) of the reduced matrix [26, 52, 47]. The VS theory expresses that if $\mathbb{T} \subset \{1, 2, \dots, M\}$ is any subset with cardinality K , chosen with probability proportional to $\det(\mathbf{A}_{\mathbb{T}}^T \mathbf{A}_{\mathbb{T}})$, then⁴,

$$\mathbb{E}\{\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2\} \leq (K + 1)\|\mathbf{A} - \mathbf{A}_K\|_F^2, \quad (2.7)$$

where $\pi_{\mathbb{T}}(\mathbf{A})$ is a matrix representing the projection of columns of \mathbf{A} onto the span of selected columns indexed by \mathbb{T} . \mathbb{E} indicates expectation operator w.r.t. all the combinatorial selection of K rows of \mathbf{A} out of M . \mathbf{A}_K is the best rank- K approximation of \mathbf{A} , that can be obtained by singular value decomposition and $\|\cdot\|_F^2$ is the Frobenius norm. VS is not a deterministic selection algorithm, as it gives a probability of selection for any subset of samples, and for which only a loose upper bound for the expectation of projection error is guaranteed. In contrast, in the present work a deterministic algorithm is proposed based on direct minimization of projection error using a new optimization mechanism.

Diversity-based selection is very sensitive to outliers and in some applications these methods are employed for outlier detection [53, 54]. A set of outlier samples from a dataset has probably more diverse samples rather than a randomly sampled subset. Thus, diversity-based selection methods should consider outliers properly. Recently, an exemplar-based subspace clustering method is proposed using selection [5]. Their employed selection algorithm is based on selecting farthest sample from previously selected samples and infusing sparsity on the metric of selection. However, our proposed selection algorithm does not necessarily provide diverse samples far from each other.

⁴ $\mathbf{A}_{\mathbb{T}}$ is the selected columns of \mathbf{A} indexed by set \mathbb{T} .

Representative Selection

A method for sampling from a set of data is proposed by Elhamifar et. al. based on sparse modeling representative selection (SMRS) [6]. Their proposed cost function for data selection is the error of projecting all the data onto the subspace spanned by the selected data. Mathematically, the optimization problem in [6] can be written as (5.1) which is an NP-hard problem and the proposed method in [6] tackles this problem via convex relaxation. However, there is no guarantee that convex relaxation provides the best approximation for an NP-hard problem. Furthermore, such methods that try to solve the selection problem via convex programming are usually computationally too intensive for large datasets [6, 4, 33, 55]. In this dissertation, a new fast algorithm for solving Problem (5.4) is proposed.

Dissimilarity-based Sparse Subset Selection (DS3) algorithm selects a subset of data based on pairwise distance of all data to some target points [4]. DS3 considers a source dataset and its goal is to encode the target data according to pairwise dissimilarity between each sample of source and target datasets. This algorithm can be interpreted as the non-linear implementation of SMRS algorithm [4].

CHAPTER 3: TENSOR-BASED SPECTRUM CARTOGRAPHY

Spectrum cartography is a promising solution to address today's spectrum deficiency caused by the recent spike in demand for wireless technologies [56, 57, 58]. The licensed holders of the spectrum (a.k.a. primary users or PUs) often under-utilize this valuable resource [59]. It is desired to allow unlicensed or secondary users (SUs) to coexist with PUs. SUs are allowed to access the spectrum, given that they do not interfere with the licensed users. This necessitates a cognitive radio system to sense the spectrum usage and accordingly adapt its spectrum utilization [60, 61].

The spectrum sensing problem is approached using numerous methods [62]. These methods are ranged from per-bin spectrum sensing [63] to wide-band sensing [64]; non-cooperative sensing [65] to cooperative sensing [44]; centralized [66] to distributed [67]; and directional sensing using phased arrays [68] to omni-directional energy detectors. Our work focuses on cooperative centralized spectrum sensing using a set of simple energy detectors. Cooperative detection of spectrum opportunities requires collecting sensed measurements in a fusion center. However, dealing with a large amount of spectrum measurements is not a trivial task. Efficient representation using high-dimensional matrices/tensors is an attractive approach for analysis of sensed measurements. In this way, structured factorization of the received spectrum enables us to capture the underlying spectrum occupancy patterns [69, 70, 71]. In the present work, we model the propagated power from the primary transmitters at different locations, time slots, and frequency channels as a multi-dimensional tensor, which is referred to as *the power tensor*.

The radio frequency (RF) cartography problem leads to find the propagating power maps across a network at any frequency channel. This is an ill-posed problem, and therefore it is difficult to infer unique and meaningful interpretation for the estimated propagating power maps. To alleviate this issue, we consider the CANDECOMP/PARAFAC (CP) [72] model for the latent tensor and

we impose smoothness constraint for the interpolated spatial maps. The CP model represents a D -dimensional tensor via D factor matrices. Each factor matrix contains a set of bases that spans one way of the tensor. Here, we deal with 3-way tensors, i.e., $D = 3$. In the proposed framework, the CP factors capture the patterns of the PUs' activities over different dimensions of time, space, and frequency. We propose the *tensor-based radio spectrum cartography (TRASC)* algorithm to address the joint problem of tensor decomposition and map interpolation. CP decomposition [72] and Tucker decomposition [73] are two well-known tensor decomposition methods, which can be interpreted as two extensions of matrix singular value decomposition (SVD). In this section, we show that the CP model can fit to the spectrum cartography application.

It is worth noting that for simplicity, we assume a centralized fusion center. Moreover, we assume that our sensors are energy detectors with omni-directional antennas. It is straightforward to extend it to a distributed framework and/or scenarios in which sensors are equipped with directional antennas as presented in [74] for the matrix-based formulation. There are so many efforts form matrix-based data completion [75, 76, 77]. Our work is the extension for tensor completion. The main goals of the study in this section are summarized as follows:

- Dynamic spectrum cartography is modeled by a low-rank tensor and the relationship between the imposed rank and the dynamics of network is studied,
- A novel algorithm, referred to as TRASC is introduced that performs spatially smooth tensor completion using tensor decomposition and spatial interpolation.
- The applicability of TRASC for the spectrum cartography application, where propagation parameters such as the location of the transmitters, and fading characteristics are unknown, is demonstrated.

per.

Related Work and Preliminaries

Tensor decomposition and multi-way modeling are old problems in mathematics [78]. However, the first practical applications of tensors dates back to 1981 in Chemometrics [79]. Since then tensors have found diverse applications in signal processing [80], computer vision [81], and graph analysis [82].

Tensor-based methods have been employed in communications and coding frameworks since Sidiropoulos et. al. introduced them for blind code-division multiple access (CDMA) [34]. Recently, more advanced coding methods in communication systems are developed based on the tensor decomposition theory [35, 36, 37]. Moreover tensors are employed for massive MIMO channel estimation in millimeter wave scenarios [38]. The common idea of all tensor-based coding methods is to perform a joint blind symbol and channel estimation at the receiver, which is feasible due to mild conditions for uniqueness of tensor CP decomposition [39].

Recently, a tensor-based formulation of spectrum sensing has been proposed [83, 84], in which the received signals are modeled as a tensor. Since the received complex signals contain phase spectral information via Fourier transform, their formulation requires the sensors to be synchronized and the channel to be multipath-free. Moreover, it is assumed that the underlying signals are stationary (the network is static). In contrast, in our proposed framework, our formulation is based on the power spectrum density (PSD) of the received signals; hence, the sensors are not required to be synchronized. Moreover, as opposed to [84], we consider the fading and multipath effects using spatial maps which can be interpreted as a linear transformation. Fu et. al. proposed a power spectra separation scheme which does not need synchronization [71]. Moreover, they assume that the transmitted signals from sources are stationary over time. Thus, time variable is eliminated by taking the PSD from received signals of sensors. In contrast, our proposed scheme assumes that

the network is dynamic and the received signals at sensors are not stationary. We assume received signals can be treated as piece-wise stationary processes. Similar to the setup in [44], where a coherent block is considered, we define time slots such that dynamics of the network are assumed constant during a time slot and sufficient samples are sensed in order to estimate a reliable PSD for each time slot. By concatenation of PSD measurements from different sensors over time a three-way tensor can be constructed. In [85], a tensor-based detection algorithm is proposed for a receiver equipped with multiple antennas for non-cooperative spectrum sensing. However, in the present work omni-directional antennas are considered in a cooperative setup. In the presence of a large amount of training data, machine learning techniques are proposed to estimate opportunities for communication [86, 87]. Our proposed work focuses on an efficient mathematical model for model-based spectrum sensing in a cognitive radio network (CRN).

In the most recent work by Zhang et. al, block-term tensor decomposition (BTD) is considered to model the propagation using a set of low-rank matrices [1]. However, they acknowledge that low-rank assumption does not take the spatial details of propagation into the account. To solve this issue, they propose to employ a 2D interpolation as a post-processing step after tensor decomposition in order to capture more details which are eliminated after low-rank assumption. We have demonstrated that their main gain comes from interpolation not BTD. We show that the well-known CP decomposition suffices for modeling cartography problem while spatial CP factors are obtained through a 2D interpolation. In other words, we demonstrate that the joint problem of tensor decomposition and interpolation is a powerful framework for cartography. Next, a short introduction to CP decomposition and spline-based interpolation is presented.

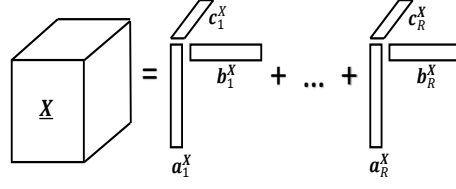


Figure 3.1: Schematic of CP decomposition to summation of rank-one tensors.

CP Decomposition

Our proposed tensor-based approach is mainly based on the *CP* decomposition [43], which factorizes a tensor into a sum of rank-one tensors. For example, a three-way tensor $\underline{\mathbf{X}} \in \mathbb{R}^{P \times F \times T}$ of rank R can be decomposed as

$$\underline{\mathbf{X}} = \sum_{r=1}^R \mathbf{a}_r^X \circ \mathbf{b}_r^X \circ \mathbf{c}_r^X = \llbracket \mathbf{A}_X, \mathbf{B}_X, \mathbf{C}_X \rrbracket, \quad (3.1)$$

where $\mathbf{a}_r^X \in \mathbb{R}^P$, $\mathbf{b}_r^X \in \mathbb{R}^F$ and $\mathbf{c}_r^X \in \mathbb{R}^T$ are *factor vectors* of the r^{th} rank-one component (Fig. 3.1). The *factor matrices* refer to the collection of factor vectors from the rank-one components, i.e., $\mathbf{A}_X = [\mathbf{a}_1^X \ \mathbf{a}_2^X \ \dots \ \mathbf{a}_R^X]$ and likewise for \mathbf{B}_X and \mathbf{C}_X . The kruskal-rank, also referred to as k-rank, is a well-known criterion for deriving conditions on the uniqueness of tensor decomposition.

The tensor CP rank may be referred to as unconstrained rank because there is no additional constraint on the factors \mathbf{a}_r^X , \mathbf{b}_r^X and \mathbf{c}_r^X . On the other hand, there is the constrained or structured rank in which the factors are restricted to be within a specific set [43]. For example, non-negative-rank and symmetric-rank are obtained by imposing nonnegativity and symmetry constraints on the factors, respectively.

Due to corruption by noise, typical tensors are not low rank and low-rank approximation should be employed in order to extract a set of meaningful components. However, low-rank approximation

of tensors is an ill-posed problem. This is because the set of tensors with the rank of at most R is not a closed set and optimization algorithms for finding CP factors result in an infimum solution which is not feasible [88]. There are some efforts for approximating rank with other functions such as nuclear norm¹ [90]. In contrast to the matrix rank minimization, tensor rank minimization cannot be relaxed easily using nuclear norm because the calculation of tensor nuclear norm is an NP-hard problem [90].

Alternating least squares (ALS) is a well-known method for finding the CP factors [91] of a tensor $\underline{\mathbf{X}}$. In this approach, factors are initialized randomly at the beginning and then updated iteratively.

Spline-based Surface Interpolation

Spectrum map of an area contains spatial correlation over neighboring locations. Our proposed framework estimates a set of *principal incomplete spectrum maps* such that the actual sensed data is a linear combination of them. The principal incomplete maps are estimated using CP decomposition. Interpolation of the principal incomplete maps (spatial CP factors) is the enabling step toward estimating the actual spectrum map for any arbitrary location. Please note that the contribution of each principal map can be estimated using CP decomposition and the same contribution coefficients are applied for reconstructing the full spectrum map using the interpolated principal maps. In the present work, *thin plate splines* (TPS) is employed for interpolating the spatial incomplete maps [92]. TPS is proposed for modeling climate data originally. However, it is an efficient model for capturing other kinds of spatial dependencies including spectrum cartography [92].

Assume we are given a set of locations (z_n, w_n) and their corresponding value y_n . The problem of surface interpolation can be cast as finding function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ such that $f(z_n, w_n)$ is as close as possible to y_n and at the same time a desirable property is satisfied for function f . Specifically, in

¹Nuclear norm is a well-known function to surrogate rank of matrices [89].

the TPS method a measure for smoothness of f is employed as follows [93],

$$I_f = \int \int \left(\frac{\partial^2 f}{\partial z^2} + 2 \frac{\partial^2 f}{\partial z \partial w} + \frac{\partial^2 f}{\partial w^2} \right) dz dw. \quad (3.2)$$

The coefficient I_f is called the bending energy of f . Function f is modeled by summation of n terms given by

$$f(z, w) = \sum_i \lambda_i r_i^\alpha \log(r_i), \quad (3.3)$$

where,

$$r_i = \sqrt{(z - z_i)^2 + (w - w_i)^2}. \quad (3.4)$$

Equation (4.17) describes the kernel of interpolation. Parameter α is the path-loss coefficient which is set to 2 for power spectrum propagation in free space [94]. The goal of interpolation is to find function f and it can be expressed mathematically as follows,

$$\underset{f}{\operatorname{argmin}} I_f \text{ s.t. } y_n = f(z_n, w_n).$$

Finding an optimized interpolating function is equivalent to estimating λ_i 's according to (4.17). Here, for simplicity of notation the minimization is shown w.r.t. function f .

In the next section, the thin plate spline interpolation is integrated with the CP decomposition in order to address the spectrum cartography problem under missing sensor measurements.

The TRASC Algorithm

This section presents our main contribution. First, the tensor-based problem formulation for spectrum cartography is introduced. The derived formulation is based on partitioning the area to a grid network and modeling the received spectrum at each grid point by a superposition of the propagating power from sources. Then, practical assumptions are elaborated to make the problem tractable and a discussion for rank estimation is exhibited. We suppose the locations of SUs are known and the goal is to find the propagating power and location of active PUs as the enabling step for reconstructing the power spectrum map at any arbitrary location and frequency. To achieve this, let us consider a set of grid points across the area of interest. Let N denote the number of grid points in the area of interest and G denotes the number of active primary users. The indices of sensors is a subset of $\{1, \dots, N\}$. The frequency bandwidth is broken into F frequency channels. The received PSD at location/sensor n at time t and frequency channel f can be written as [44]²,

$$\begin{aligned} y_n(t, f) &= \sum_{g=1}^G \phi_{ng} x_g(t, f) + z_n(t, f) \\ &= \boldsymbol{\phi}_n^T \mathbf{x}(t, f) + z_n(t, f), \end{aligned} \tag{3.5}$$

where, $\boldsymbol{\phi}_n = [\phi_{n1}, \dots, \phi_{nG}]^T$ in which ϕ_{ng} is the channel gain between the g^{th} active source and the n^{th} sensor. The total number of active sources is indicated by G . The propagation from the g^{th} active source is denoted by $x_g(t, f)$ and the collection for all grid points forms vector $\mathbf{x}(t, f) = [x_1(t, f), \dots, x_G(t, f)]^T$. We assume the measurements are available for T time slots and F frequency channels. Collaborative estimation of $\mathbf{x}(t, f) \in \mathbb{R}^G$ over each time slot and for each frequency bin requires collecting measurements of all sensors in vector $\mathbf{y}(t, f) = [y_1(t, f), \dots, y_N(t, f)]^T \in \mathbb{R}^N$. The following minimization problem has been proposed for esti-

²This form requires a set of mild conditions which is out of scope our present work and studied in [44]

mation of $\mathbf{x}(t, f)$ for each time and frequency independently [44]

$$\mathbf{x}(t, f) = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}(t, f) - \Phi \mathbf{x}\|_2, \quad (3.6)$$

where the n^{th} row of matrix Φ is ϕ_n^T . The regularized version of (3.6) using ℓ_1 constraint has previously been employed for collaborative spectrum estimation for a given Φ [44, 51].

Let us represent the collection of $\mathbf{x}(t, f)$ and $\mathbf{y}(t, f)$ for all frequencies and time slots as tensors $\underline{\mathbf{X}} = [x_{gtf}]$ and $\underline{\mathbf{Y}} = [y_{ntf}]$, respectively. That is $x_{gft} = x_g(t, f)$ and $y_{nft} = y_n(f, t)$. Tensor $\underline{\mathbf{X}} \in \mathbb{R}^{G \times F \times T}$ is referred to as power tensor and tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{N \times F \times T}$ is referred to as cartography tensor. Please note that $\mathbf{x}(t, f)$ is a vector in \mathbb{R}^G and $\mathbf{y}(t, f)$ is a vector in \mathbb{R}^N . These vectors point to the mode-1 fibers of $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$, respectively. We aim to estimate the propagation power from each active source using the accessible measurements in $\underline{\mathbf{Y}}$. Mathematically speaking, we have:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \Phi + \underline{\mathbf{Z}}, \quad (3.7)$$

Each entry of matrix $\Phi \in \mathbb{R}^{N \times G}$ represents the channel gain between the n^{th} grid point and the g^{th} source [44]. Estimate the power tensor, $\underline{\mathbf{X}}$, which is characterized by its CP factors, is an enabling step toward estimating non-sensed entries of the propagation tensor, $\underline{\mathbf{Y}}$. Due to narrow band communication, and the temporal correlation of power propagation at a transmitter, tensor $\underline{\mathbf{X}}$ is highly structured and can be modeled by a low-rank tensor using the CP decomposition as stated in (3.1). Each rank-1 tensor, i.e., $\mathbf{a}_r^X \circ \mathbf{b}_r^X \circ \mathbf{c}_r^X$, represents a principle pattern of the spectrum propagation. Matrix Φ consists of G number of vectorized maps in which each map is a reshaped version of a column of Φ . Let define tensor $\underline{\mathbf{\Gamma}} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times G}$ such that the g^{th} slice of $\underline{\mathbf{\Gamma}}$ is the reshaped version of the g^{th} column of Φ . The reshaping operator provides a spatial map in each slice of $\underline{\mathbf{\Gamma}}$ such that vicinity is preserved and entries are corresponding to the original area of the network. Without loss of generality we assume that N is a perfect square number and the area of

interest consists of $\sqrt{N} \times \sqrt{N}$ grid points. Moreover, let us define multiplication operator $\langle \cdot, \cdot \rangle$ such that

$$\langle \underline{\mathbf{X}}, \underline{\mathbf{\Gamma}} \rangle = \underline{\mathbf{X}} \times_1 \Phi. \quad (3.8)$$

It is interesting to mention that the CP factors of tensor $\underline{\mathbf{Y}}$ in the presence of no noise, i.e., $\underline{\mathbf{Z}} = 0$, can be stated as follows in terms of CP factors of tensor $\underline{\mathbf{X}}$:

$$\begin{aligned} \mathbf{a}_r^Y &= \Phi \mathbf{a}_r^X, \\ \mathbf{b}_r^Y &= \mathbf{a}_r^X, \\ \mathbf{c}_r^Y &= \mathbf{c}_r^X. \end{aligned} \quad (3.9)$$

In the proposed formulation tensor $\underline{\mathbf{X}}$ is assumed to be low-rank and this property is inherited to $\underline{\mathbf{Y}}$. Imposing low-rankness on channel gain matrices via Φ turns the problem into a block-term decomposition. However, in general this matrix is not low-rank even in absence of noise since propagation pattern is radial. Although a low-rank approximation always exists for the channel gain matrix, in practice there are some imperfections such as shadowing and obstacles which affects the low-rank approximation parameters. Therefore, CP decomposition is more parsimonious to model cartography tensor. Moreover, as it will be shown since there are only one parameter in CPD it is easier to fine tune it and the final performance is more robust to mismatch in rank selection. Tensor $\underline{\mathbf{\Gamma}}$, which is a reshaped version of Matrix Φ , plays a key role in our architecture. Each slice of this tensor contains an incomplete power propagation pattern from the g^{th} source. Interpolating these patterns alongside with tensor $\underline{\mathbf{X}}$ results in a completed cartography tensor, $\underline{\mathbf{Y}}$. Solving the following problem provides us with two outcomes: (i) the CP factors of the power

tensor and (ii) principle propagation maps and their interpolation functions.

$$\begin{aligned}
& \underset{\mathbf{a}_r^X, \mathbf{b}_r^X, \mathbf{c}_r^X, f_g}{\operatorname{argmin}} \quad \|\underline{\mathbf{M}} * (\underline{\mathbf{Y}} - \langle \underline{\mathbf{X}}, \underline{\mathbf{\Gamma}} \rangle)\|_F^2 + \alpha \sum_g I_g \\
& \text{subject to: } \underline{\mathbf{X}} = \sum_{r=1}^R \mathbf{a}_r^X \circ \mathbf{b}_r^X \circ \mathbf{c}_r^X, \quad \text{and} \\
& f_g(z_i, w_i) = \mathbf{\Gamma}_g(z_i, w_i)
\end{aligned} \tag{3.10}$$

Since sensors are distributed in a small subset of $\{1, \dots, N\}$, we need to reduce the problem to only the known information. Binary tensor $\underline{\mathbf{M}}$ shows the sensed entries of tensor $\underline{\mathbf{Y}}$ and operator $*$ refers to the element-wise product. Function f_g interpolates the incomplete power spectrum map corresponding to the g^{th} slice of $\underline{\mathbf{\Gamma}}$. The bending energy of $f_g(z, w)$ is denoted by I_g which is defined in (3.2) and variable α controls smoothness of the interpolation function. Each slice of tensor $\underline{\mathbf{\Gamma}}$ is a 2D incomplete surface since the sensed tensor $\underline{\mathbf{Y}}$ is incomplete. Interpolation of all slices of $\underline{\mathbf{\Gamma}}$ results in completion of $\underline{\mathbf{Y}}$. Please note that each slice of $\underline{\mathbf{\Gamma}}$ exploits a separate interpolating function. This problem can be regarded as a combination of CP decomposition and 2D surface interpolation. Here, we assume that sources are distinguished based on their spatial distance and their spectral signature over time. For example, if the spectral pattern of propagating power from a location is changed after a certain time, there is two distinguished sources.

In the proposed model, parameters G and R depend on the actual number of active users. However, the actual number of active users is unknown in a realistic network. Therefore, a large enough estimation can be substituted. In order to make the model more parsimonious, we assume that $R = G$. The proposed problem can then be simplified to the following form by elimination of

parameter G and the auxiliary variable $\underline{\mathbf{X}}$.

$$\begin{aligned} \underset{\phi_r, \mathbf{b}_r, \mathbf{c}_r, f_r}{\operatorname{argmin}} \quad & \|\underline{\mathbf{M}} * (\underline{\mathbf{Y}} - \sum_{r=1}^R \phi_r \circ \mathbf{b}_r \circ \mathbf{c}_r)\|_F^2 + \alpha \sum_r I_r \\ \text{subject to:} \quad & f_r(z_i, w_i) = \Gamma_r(z_i, w_i) \quad \forall i \in \Omega. \end{aligned} \quad (3.11)$$

In this problem ϕ_r is the vectorized replica of the r^{th} principle spectrum map and it can be interpreted as a CP factor of tensor $\underline{\mathbf{Y}}$ w.r.t. the spatial dimension. In other words, the first CP factor is regularized to have the minimum bending energy in order to have a smooth map over interpolated locations. Tensor $\underline{\mathbf{Y}}$ is an incomplete tensor w.r.t. non-sensed locations. However, a spatial interpolation can reveal us a completed principle map. Each principle map is interpolated using a function which is regularized to have a smooth behavior. This regularization is inspired by prior work in the literature of interpolation as discussed in Sec. . The interpolator function is constrained to be equal to the principle maps at the observed locations indexed in set Ω . The coordinate of observed locations are shown by (z_i, w_i) . Low-rank assumption for the cartography tensor is helpful for the making the problem identifiable. However, fine local details of the spectrum maps are lost in the found low-rank structure. Two dimensional interpolation is employed in order to keep details in the sensed data and to infuse these details for interpolation of non-sensed locations. Surface interpolation helps CP factors to obtain spatial smooth factors. Alternating least squares is the practical approach for solving the proposed problem in which each iteration corresponds to a least squares problem with missing entries. The steps for solving this problem will be explained in Section . It is worthwhile to mention that interpolating \mathbf{b}_r and \mathbf{c}_r results in increasing both spectral and temporal resolution. However, in the present work we study cartography which is focused on increasing the spatial resolution.

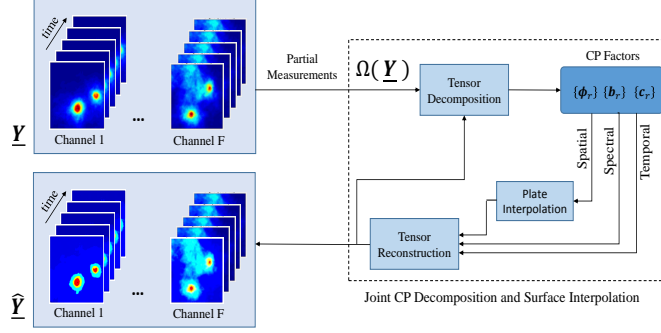


Figure 3.2: framework of the proposed joint tensor decomposition and surface interpolation. This scheme only shows a big picture of the proposed work. Practically, CP decomposition is implemented iteratively. In the proposed algorithm, interpolation is performed alongside iterations of CP decomposition. In other words, we solve a joint problem of decomposition and interpolation.

Our proposed optimization problem is based on assuming tensor \underline{Y} to be low-rank. The relation of rank to dynamics of the network is investigated in the appendix from the mathematical point of view. However, in a practical situation, a proper rank is chosen empirically which will be discussed in Sec. .

Alternating Least Squares Solution

The main proposed formulation is a joint optimization problem. In order to solve it, we employ block coordinate descent algorithm in order to break the main problem into a set of consecutive problems. Each subproblem is a minimization problem in terms of only one unknown parameter.

$$\min_{\Phi} \|M_1 * (Y_1 - \Phi(C \odot B)^T)\|_F^2 \quad (3.12a)$$

$$\min_B \|M_2 * (Y_2 - B(C \odot A)^T)\|_F^2 \quad (3.12b)$$

$$\min_{\mathbf{C}} \|\mathbf{M}_3 * (\mathbf{Y}_3 - \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T)\|_F^2 \quad (3.12c)$$

$$\min_{f_r} I_r \text{ s.t. } f_r(z_i, w_i) = \Gamma_r(z_i, w_i) \quad \forall i \in \Omega. \quad (3.12d)$$

The first three subproblems are borrowed from the plain CP decomposition. The last subproblem attempts to estimate R interpolating function for each spatial function. The solution of the last subproblem can be obtained by an interpolation technique. In our work we employ thin-plate splines method to solve the last subproblem as discussed in .

A fundamental difference with the plain CP decomposition is restriction of entries for only those that are sensed. Considering missing entries in a least square problem is discussed in the next subsection. Since, it is desired to obtain an online spectrum map, the first subproblem should be rewritten in terms of the last sensed spectrum slice over time. Moreover, considering the last time slot for updating the spatial maps helps us to model the changing sensed entries over time. Let \mathbf{Y}^t denote the entries of tensor $\underline{\mathbf{Y}}$ at the t^{th} time slot and \mathbf{M}^t shows the sensed entries. Thus, restricting the first subproblem in (3.12a) into the last time slot results in the following equation.

$$\min_{\Phi} \|\mathbf{M}^t * (\mathbf{Y}^t - \Phi(\mathbf{B} \text{diag}(\mathbf{C}^t))^T)\|_F^2. \quad (3.13)$$

Here, \mathbf{C}^t refers to the t^{th} row of \mathbf{C} . The closed form solution of the introduced subproblems considering the mask is discussed next.

Least Square Solution with Missing Entries

Our main proposed algorithm requires solving a general least square problem with missing entries along iterations. Let us formalize this problem as follows for a given matrix \mathbf{Y} , a given matrix \mathbf{U} , and known entries organized in a binary matrix denoted by \mathbf{M} ,

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{M} * (\mathbf{Y} - \mathbf{U}\mathbf{Z})\|_F^2 \quad (3.14)$$

The solution of this problem w.r.t. \mathbf{Z} given \mathbf{Y} , \mathbf{U} , and the mask is straightforward which is indicated in Alg. 1 referred to as missing entries least squares (MELS). Please note that MELS solves a basic optimization problem and our main proposed algorithm is built upon it iteratively. In MELS, each column of \mathbf{Z} can be computed independently.

Algorithm 1 Missing Entries Least Squares (MELS).

Require: $\mathbf{Y} \in \mathbb{R}^{N \times F}$, $\mathbf{U} \in \mathbb{R}^{N \times R}$, and the mask, \mathbf{M} .

Output: $\mathbf{Z} \in \mathbb{R}^{R \times F}$.

```

FOR  $f = 1, \dots, F$ 
2:    $\mathbf{D} = \operatorname{diag}(\mathbf{M}(:, f))$ 
3:    $\mathbf{W} = \mathbf{U}^T \mathbf{D} \mathbf{U}$ 
4:    $\mathbf{Z}(:, f) = \mathbf{W}^{-1} \mathbf{U}^T \mathbf{D} \mathbf{Y}(:, f)$ 
END FOR
```

Alg. 1 is a simple solution for (3.14) which is a basic problem in linear algebra. However, it plays a key role in our main algorithm. The solution of (3.14) is referred as $\text{MELS}(\mathbf{Y}, \mathbf{U}, \mathbf{M})$. Our main proposed algorithm which is presented later needs the solution for the least squares problem with missing entries iteratively. The main algorithm is discussed next which is built based on Alg. 1.

Implementation of TRASC for Power Map Reconstruction

In this section the practical algorithm for solving (3.11) is proposed. The sensed incomplete tensor is decomposed into a set of CP factors considering the known entries. Then, the spectral and temporal factors are kept and spatial factors are interpolated in each iteration of tensor decomposition. The power spectrum at any arbitrary location and in each frequency can be inferred via tensor reconstruction of CP factors. Alg. 2 presents the steps of the TRASC algorithm that is a *joint tensor decomposition and 2D interpolation for spectrum cartography*. Spectral and temporal CP factors, i.e., matrices \mathbf{B} and \mathbf{C} , are initialized by a plain CP decomposition on \mathbf{Y}_Ω where unknown entries are set to 0. The initial value for spatial factors is estimated using the MELS algorithm in Line 2 of the algorithm. In Alg. 2, TPS() refers to thin-plate splines interpolation method which is introduced in Sec. This interpolation needs a 2D incomplete plate as in input in addition to the indices of known entries and regularization parameter introduced in Sec. .

In Alg. 2, $\mathbf{Y}^t = \mathbf{Y}(:, :, t)$ is a slice of tensor \mathbf{Y} corresponding to the time slot t . Please note that \mathbf{Y}^t is an $N \times F$ matrix which has F columns. Each column has N elements corresponding to measurements of all spectrum sensors. In other words, each column of \mathbf{Y}^t is a 1D collection of all power spectrum sensors within the 2D network. The order for vectorization is arbitrary; however, in Line 8 of the algorithm, the operator reshape is the inverse operator for the employed vectorization. Here, Φ refers to the full spatial factors and \mathbf{A}^t refers to the incomplete spatial maps corresponding to time slot t . CP decomposition using ALS algorithm can be categorized as a block coordinate descent algorithm. It is shown that if optimization along any coordinate direction yields a unique minimum point then the main cost function is convergent using a coordinate descent method [95]. Line 3, Line 4 and Line 7 of the TRASC algorithm is identical to the conventional ALS algorithm for CP decomposition. The main variables are computed in these three steps. However, we define a dependent variable which is obtained by interpolation of the spatial factors.

Algorithm 2 Tensor-based Radio Spectrum Cartography (TRASC).

Require: \underline{Y}_Ω , R , Ω , and N .

Output: \underline{Y} .

```
1: Initialize  $\mathbf{B}$  and  $\mathbf{C}$  by CP factors of  $\underline{Y}_\Omega$ 
2:  $\Phi \leftarrow \text{MELS}(\mathbf{Y}_1^T, \mathbf{B} \odot \mathbf{C}, \mathbf{M})$  %solution of (3.12a)
   While (The stopping criterion is not met)
3:    $\mathbf{B} \leftarrow \text{MELS}(\mathbf{Y}_2^T, \mathbf{C} \odot \Phi, \mathbf{M})$  %solution of (3.12b)
4:    $\mathbf{C} \leftarrow \text{MELS}(\mathbf{Y}_3^T, \mathbf{B} \odot \Phi, \mathbf{M})$  %solution of (3.12c)
5:   FOR  $t = 1 \dots T$ 
6:      $\mathbf{F} = \mathbf{B} \text{diag}(\mathbf{C}_t)$ 
7:      $\mathbf{A}_\Omega^t = (\mathbf{F}^\dagger \mathbf{Y}_\Omega^t)^T$  %solution of (3.13)
8:     FOR  $r = 1 \dots R$ 
9:        $\mathbf{\Gamma}_r = \text{TPS}(\text{reshape}(\mathbf{a}_\Omega^t, [\sqrt{N}, \sqrt{N}]), \mathbf{M})$  %solution of (3.12d)
10:       $\phi_r = \text{vec}(\mathbf{\Gamma}_r)$ 
11:    END FOR
12:     $\mathbf{Y}^t = \Phi \mathbf{F}^T$ 
13:  END FOR
14: END While
```

Note that convergence of TRASC algorithm is inherited from convergence of the ALS algorithm for CP decomposition [96]. Thus, convergence of the main variables results in convergence of the dependent variable which is interpolated.

Rank Estimation in a Dynamic Environment

As it is shown in Sec. ??, the rank of sensed tensor in the absence of noise is proportional to dynamic parameters of the problem including the number of active sources and the number of changes in the spectral pattern of each source. Moreover, mobility of sources is equivalent to multiple active locations with distinguished temporal signatures. Thus, velocity of sources is another important parameter which affects the underlying rank of the sensed tensor.

Rank estimation considering the dynamics of network is a two-sided problem. The first aspect is

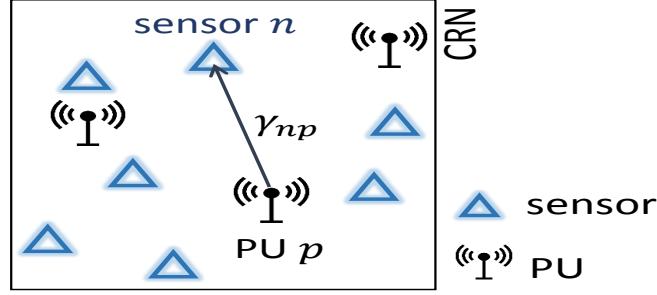


Figure 3.3: The practical implementation of TRASC with adaptive rank estimation in a dynamic radio environment.

that all parameters affecting the rank are unknown in the proposed framework. On the other hand, we do not need to estimate all of these parameters separately in order to estimate a fitted rank. In other words, we devise a practical method that estimates the proper rank experimentally. The impact of the unknown dynamic parameters is projected on the estimated rank. The least squares term in (3.11) which is the main objective is considered as a measure for estimating a fitted rank. We evaluate this term first by using a small number as the rank and solving the problem. Then, we increase the rank gradually to reach a point that the residual error does not improve further. In the experimental results we will see the performance of rank estimation and evaluate the sensitivity of estimated rank.

A practical framework of cartography method should include a block for rank estimation based on the dynamic behavior of network. Fig. 3.3 shows the diagram of the proposed dynamic cartography via TRASC equipped with a block for rank estimation.

Experiments

The TRASC algorithm is evaluated for dynamic spectrum cartography. The experimental setup is similar to that of [1]. Specifically, we consider $F = 16$ channels and our area of interest with the size of $50 \times 50 \text{ m}^2$ is discretized into 51 horizontal bins and 51 vertical bins, i.e., $N = 2601$.

The primary active users are considered static or mobile and $T = 100$ time slots are employed. The spatial propagation pattern of each transmitter is synthesized using a path-loss model and the spatial correlated log-normal shadowing model [97]. The attenuation from location (z, w) to (z_i, w_i) is expressed by

$$\|(z - z_i)^2 + (w - w_i)^2\|^{-\eta} 10^{\alpha_r(z,w)/10},$$

where,

$$\mathbb{E}\{\alpha_r(z, w)\alpha_r(z', w')\} = \sigma_r \exp(-\|(z - z')^2 + (w - w')^2\|/X_c).$$

Parameter X_c is called decorrelation distance which is ranged from 50 to 100 for a typical outdoor environment [97]. The spectral activity pattern of each transmitter is assumed as summation of three sinc functions as explained in [1] as follows:

$$b_r(f) = \sum_{i=1}^3 q_i^r \text{sinc}^2\left(\frac{(f - f_i^r)}{w_i^r}\right), \quad (3.15)$$

where, q_i^r follows a uniform distribution between 0.5 and 2. Moreover, f_i^r and w_i^r are the central frequency and the width parameter of each function, respectively. The width parameter is drawn from a uniform distribution between 2 and 4.

The performance of different spectrum sensing algorithms are evaluated via the cartography error which is defined as

$$e = \frac{\|\log(\mathbf{Y}) - \log(\hat{\mathbf{Y}})\|_F}{\|\log(\mathbf{Y})\|_F}. \quad (3.16)$$

Matrix \mathbf{Y} indicates the power spectrum map and $\hat{\mathbf{Y}}$ refers to the interpolated map using a set of measurements. Employing logarithm scale results in less bias for high power spectrum areas of the network. Thus, we have a more reliable measure for comparison of the interpolated low-power details in the area of interest. Two different sensing patterns are employed in our experiments as

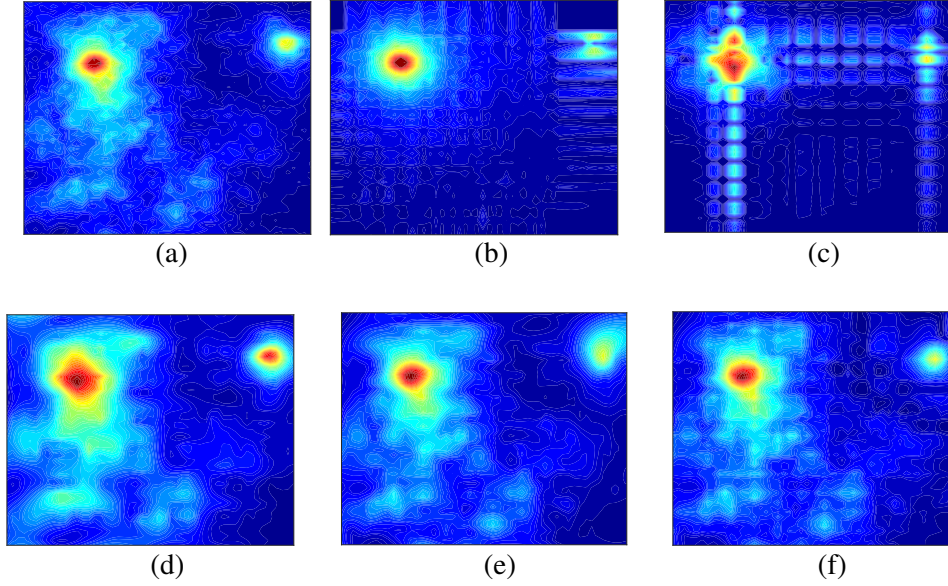


Figure 3.4: Comparison between the original and the recovered spectrum maps. (a) The original power spectrum map. The grid is 50×50 , however, 6 columns and 6 rows of the original power spectrum map are measured. The power spectrum is measured at these locations only. (b) The recovered spectrum from missing and noisy sensed data using a plain CP decomposition and interpolating the unread measurements via CP reconstruction. (c) The recovered spectrum via block-term decomposition. (d) The plain 2D plate splines method is employed for interpolating the power spectrum map. (e) The proposed method in [1] via the block-term decomposition which post-processed using 2D plate splines. (f) Our proposed method that employs CP decomposition and 2D plate splines jointly.

suggested in [1]. In the first pattern, a line of horizontal grid points and a line of vertical grid points are scanned. This pattern is referred as structured pattern. The second pattern corresponds to random sampling of grid points in the network.

Fig. 3.4 shows the original power spectrum map in a frequency band which is sampled in a small subset of locations versus the recovered power spectrum map using different recovery algorithms. In Fig. 3.4b the recovered spectrum using a plain CP decomposition is shown. In other words, only a low-rank decomposition is employed to interpolate the incomplete sensing results. The same strategy can be repeated using any other tensor decomposition model. Fig. 3.4c shows the interpolation result using the plain block-term tensor decomposition [1]. Kernel-based interpolation methods interpolate the incomplete set of measurements via neighborhood information. How-

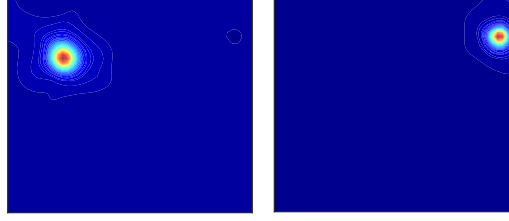


Figure 3.5: The interpolated spatial components using our proposed framework. The rank of CPD is assumed to be 2. A linear combination of these two factors is able to reconstructs the power spectrum map in any frequency band.

ever, these methods neglect the global correlation among measurements in space, spectral bands and time. Fig. 3.4d shows the interpolated map using 2D plate splines method [98]. Fig. 3.4e shows the result of coupled BTM where the pathloss gains are corrected using plate splines as a post-processing. In Fig. 3.4f, our proposed framework is evaluated which employs an iterative approach between model-based CP factors and neighborhood-based splines. The last two subfigures correspond to the joint methods that exploit both tensor-based decomposition and interpolation. As it can be seen, utilizing both techniques improves the accuracy of spectrum recovery. Our proposed joint method estimates the low-power source at the top right of the area more accurate. However, in the next simulations their performance will be compared quantitatively.

Fig. 3.5 exhibits two interpolated CP spatial components. At each iteration of TRASC, a set of spatial CP factors are estimated and interpolated. Plate splines method is utilized to obtain interpolated CP factors as the basic components to reconstruct the desired spectrum map based on them.

In the next experiment, we study the performance of two basic methods based on tensor decomposition and neighborhood interpolation in terms of the normalized error of cartography defined in (3.16). Moreover, in this experiment we consider the BTM method for cartography [1]. Fig. 3.6b shows the cartography error of different methods over time. Tensor-based methods need a fine tuning of rank in practice. In this experiment it is assumed that the assumed rank differs +1

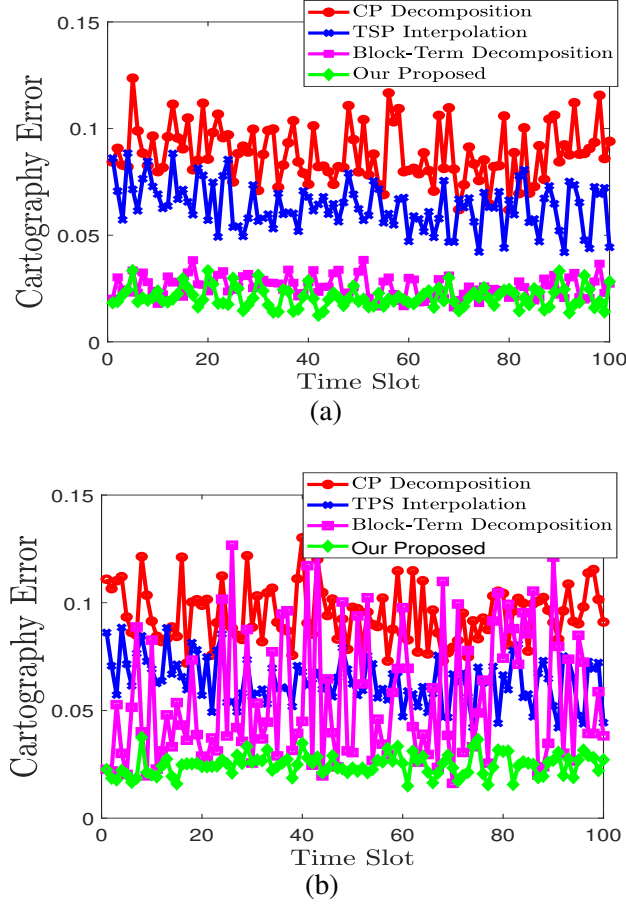


Figure 3.6: Spectrum map reconstruction error for several algorithms. (a) The rank for the tensor-based methods is assumed to be the best possible rank. (b) The rank for the tensor-based methods is assumed to differ from the best possible rank by +1.

from the best possible rank. As it can be seen and as it is mentioned in [1], the block-term tensor decomposition is highly sensitive to the rank. However, our proposed structured CP decomposition is not highly sensitive to the rank. In Fig.3.6b the cartography error for 100 time slots is plotted. Tensor-based methods also are compared with the naive interpolation on independent channels for each time slot using TPS interpolation. In some time slots, the performance of the coupled block-term decomposition is close to the performance of our framework. However, the coupled block-term decomposition is performing worse than TRASC in average over time.

Unlike the coupled block-term decomposition method [1], our framework is an iterative approach

which performs both tensor decomposition and 2D spline interpolation at each iteration. However, any iterative approach raises the convergence issue which must be investigated. Fig. 3.7 shows the performance of the proposed framework over iterations in terms of the normalized cartography error.

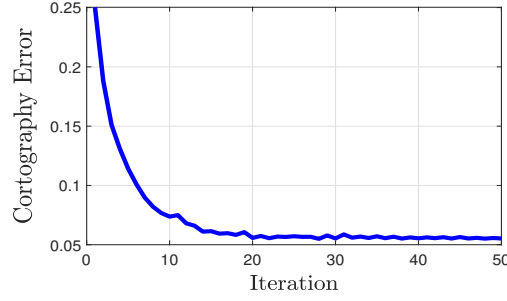


Figure 3.7: The convergence behavior of the proposed framework. In each iteration of the proposed algorithm, all tensor CP factors are updated and a more accurate model is estimated for reconstruction of power spectrum map.

An adaptive rank estimation method introduced in the previous section is presented next. Fig. 3.8 shows the residual error of the main cost function (3.11) versus the assumed rank. There are two static sources in the area and each one has three active bands based on (3.15). The cartography error can be interpreted as a generalized error for unseen grid points and the residual error represents the error of TRASC only for the sensed grid points. This concept is similar to the train error and the test error in machine learning systems. Increasing the rank improves the residual error, however, after a certain point it will cause over-fitting for the general cartography error.

The behavior of TRASC w.r.t. the assumed rank is smooth while the method based on BTD introduced in [1] is highly sensitive to the assumed rank of BTD. Fig. 3.9 shows the sensitivity of cartography based on BTD w.r.t. the assumed rank. As it can be seen, this method only performs efficiently for a specific rank. This problem makes BTD not viable in the cartography application.

In Fig. 3.10a, our proposed spectrum sensing framework is compared with the block-term tensor decomposition in terms of the number of sensed grid points. In structured sampling an entire line

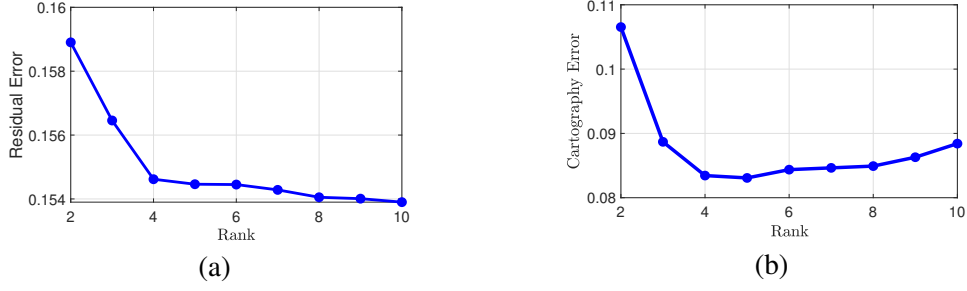


Figure 3.8: (a) The effect of assumed rank on the residual error of cost function (3.11). (b) Sensitivity to the proposed framework w.r.t. the assumed CP rank.

of horizontal/vertical grid points are sensed. However, in the random sampling the sensed grid points have no spatial structure.

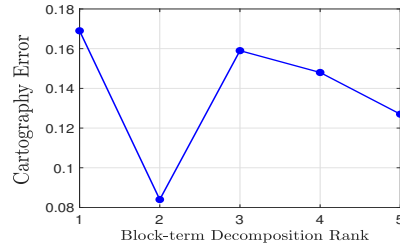


Figure 3.9: Sensitivity of BTM to the assumed rank.

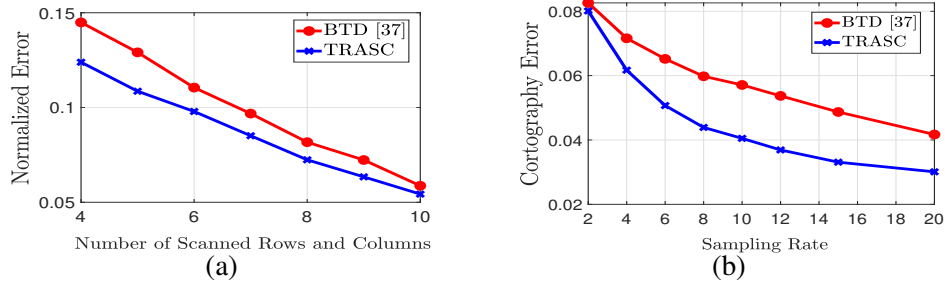


Figure 3.10: (a) The number of structured measurements versus the normalized cartography error. (b) The number of random measurements versus the normalized cartography error.

In each iteration of our framework, a 2D interpolation is applied on all spatial CP factors. The impact of the regularization parameter which controls the smoothness of interpolation functions is studied in Fig. 3.11. A low value for parameter α corresponds to a non-smooth interpolating function. However, there is a wide range for the smoothing parameter such that the cartography error is improved.

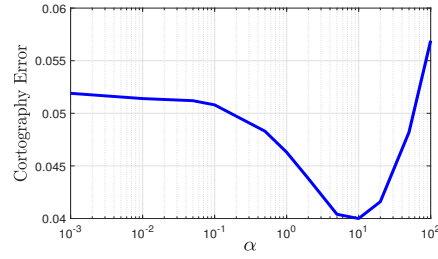


Figure 3.11: The impact of the smoothness parameter in 2D splines interpolation on the overall performance of the proposed framework.

A moving source can be modeled using multiple sources in different time slots since it is propagating from a grid point and it will be propagating from another neighboring point in the next time slot. Thus mobility increases the complexity and consequently the underlying rank of TRASC. Fig. 3.12 shows the cartography error of two moving sources while their speed is denoted as number of grid points that are paved in $T = 100$ time slots. As it can be seen, for a higher speed, a larger number for the rank needs to be considered.

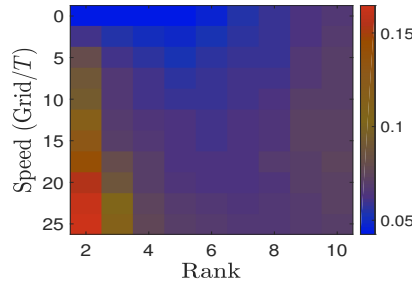


Figure 3.12: The performance of the proposed framework in presence of moving sources in terms of the normalized cartography error. The speed is denoted as the number of paved grid points in T time slots where $T = 50$ time slots are considered. .

Conclusion

A new framework for dynamic spectrum cartography is proposed. The thin plate spline interpolation method and the tensor CP decomposition algorithm are employed jointly in a unified framework. An iterative algorithm, referred to as TRASC, is introduced for estimating CP factors and the parameters of interpolation. The proposed joint decomposition and interpolation is used for

tensor completion to address the problem of spectrum cartography under the shadowing channel model and transmitters mobility. Our proposed cartography technique is shown to be as accurate as the state-of-the-art method while it is less sensitive to the model's parameters such as the assumed rank of tensors. It is shown that the rank of a cartography tensor depends on the dynamics of the problem such as the number of active sources and their number of spectral changes over time. In order to make the proposed approach practical, TRASC can be integrated with an adaptive rank estimator to adjust the rank over time according to the dynamics of the network.

CHAPTER 4: E-OPTIMAL SENSOR SELECTION

Collaborative estimation of a sparse vector \mathbf{x} by M potential measurements is considered. Each measurement is the projection of \mathbf{x} obtained by a regressor, i.e., $y_m = \mathbf{a}_m^T \mathbf{x}$. The problem of selecting K sensor measurements from a set of M potential sensors is studied where $K \ll M$ and K is less than the dimension of \mathbf{x} . In other words, we aim to reduce the problem to an under-determined system of equations in which a sparse solution is desired. Sparsity facilitates employing the compressive sensing approach where the compressed measurements are selected from a large number of potential sensors. This dissertation suggests selecting sensors in a way that their corresponding regressors construct a well conditioned measurement matrix. Our criterion is based on E-optimality, which is highly related to the restricted isometry property that provides some guarantees for sparse solution obtained by ℓ_1 minimization. Data/feature selection is an enabling step for processing a huge set of data. However, the proposed basic selection algorithm is not tractable for huge number of data. The proposed basic E-optimal selection is vulnerable to outlier and noisy data. The robust version of the algorithm is presented for distributed selection for big data sets. Moreover, an online implementation is proposed that involves partially observed measurements in a sequential manner. Our simulation results show the proposed method outperforms the other criteria for collaborative spectrum sensing in cognitive radio networks (CRNs).

Our suggested selection method is evaluated in machine learning applications. It is used to pick up the most informative features/data. Specifically, the proposed method is exploited for face recognition with partial training data.

Complex systems containing very large numbers of data-gathering devices, were developed in the last decade. However, dealing with large number of sources of data is challenging. The emerging research area, big data, aims to address challenges of such complex systems. Representing the

underlying structure of data by a succinct format is a crucial issue in the big data literature. For instance, dimension reduction techniques and different clustering-based approaches aim to extract a concise format of data. Representatives obtained by such methods are often not easy to interpret. Furthermore, obtaining each representative implies processing of all data or a large portion of data. In order to have a straightforward interpretation, it is desired to find the representatives by selection from data. There are some clustering approaches that select the representatives from data such as k-medoids clustering [13]. However these clustering methods assign each data to only one prototype which is the cluster representer, while in the case of highly structured data only one prototype from data does not contain sufficient information to capture the underlying structure of the whole cluster.

An example of big data system is wireless sensor networks, where the processing unit has to deal with an excessively large number of observations acquired by the various sensors. Often there exist some redundancies within the sensed data and they should be pruned. Sensor selection and sensor scheduling aim to address this problem. In many applications the sensor selection task is non-trivial and possibly consists of addressing an NP-hard problem (i.e., there are $\binom{M}{K}$ possibilities of choosing K distinct sensors out of M available ones). This essentially implies that an optimal solution cannot be efficiently computed, in particular when the number of sensors becomes excessively large. A convex relaxation of the original NP-hard problem has been suggested in [15]. The most prominent advantage of this approach over other methods is its practicality, thanks to many well-established computationally-efficient convex optimization techniques. In addition to convex relaxation, a sub-modular cost function as the criterion of sensor selection allows us to take advantage of greedy optimization methods for selecting sensors [16]. The existing studies on sensor selection mostly consider heuristic approaches. For example, in [15] the volume of the reduced bases is considered. This method is called *D-optimality*. In addition, *A-optimality* [17] and *E-optimality* [17] are suggested as some other alternative heuristics already introduced in

convex optimization. These heuristics are presented without any specific justification for sensor selection application. In this chapter we are going to exploit a criteria more judiciously in favor of compressed sensing (CS) theoretical guarantees.

Motivation

Compressed sensing is a technique by which sparse signals can be measured at a rate less than conventional Nyquist sampling theorem [99, 100]. There exist vast applications of CS in signal and image processing [101], channel estimation [102], cognitive radio [103] and spectrum sensing [104]. CS aims to recover a sparse vector, \mathbf{x} , using a small number of measurements \mathbf{y} . The CS problem can be formulated as,

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0 \quad \text{s.t. } \mathbf{y} = \Phi \mathbf{x}, \quad (4.1)$$

where, $\|\cdot\|_0$ represents the number of non-zero elements of a vector. $\Phi \in \mathbb{R}^{K \times N}$ is called measurement matrix that provides us K measurements collected in \mathbf{y} . These measurements sense from an unknown N dimensional vector. Exact solution of the above optimization problem is through the combinational search among all possible subsets. Due to its high computational burden, this algorithm is impractical for high dimension scenarios. Many sub-optimal algorithms have been proposed such as OMP [105], smoothed ℓ_0 [106] and basis pursuit [107]. Basis pursuit is based on relaxing ℓ_0 to ℓ_1 norm and is popular due to theoretical guarantees and reasonable computational burden [108]. The theoretical guarantees for ℓ_1 minimization arise from several sufficient conditions based on some suggested metrics. These include the mutual coherence [109], null space property [110], spark [111] and restricted isometry property (RIP) [112]. Except for the mutual coherence, none of these measures can be efficiently calculated for an arbitrary given measurement matrix Φ . For example, the RIP requires enumerating over an exponential number of index sets.

RIP is defined as follows.

Definition 1 [112] *A measurement matrix is said to satisfy symmetric form RIP of order S with constant δ_S if δ_S is the smallest number that*

$$(1 - \delta_S)\|\mathbf{x}\|_2^2 \leq \|\Phi\mathbf{x}\|_2^2 \leq (1 + \delta_S)\|\mathbf{x}\|_2^2, \quad (4.2)$$

holds for every S -sparse \mathbf{x} (i.e. \mathbf{x} contains at most S nonzero entries).

Based on this definition several guarantees are proposed in terms of δ_{2S} , δ_{3S} and δ_{4S} in [113] and [114] in order to guarantee recovering S -sparse vectors. By S -sparse we mean a vector that has S non-zero entries. In [115] an asymmetric form of definition 1 is introduced in order to more precisely quantify the RIP.

Definition 2 [115] *For a measurement matrix the asymmetric RIP constants δ_S^L and δ_S^U are defined as,*

$$\begin{aligned} \delta_S^L(\Phi) &= \underset{c \geq 0}{\operatorname{argmin}} (1 - c)\|\mathbf{x}\|_2^2 \leq \|\Phi\mathbf{x}\|_2^2, \quad \forall \mathbf{x} \in \mathcal{X}_S^N, \\ \delta_S^U(\Phi) &= \underset{c \geq 0}{\operatorname{argmin}} (1 + c)\|\mathbf{x}\|_2^2 \geq \|\Phi\mathbf{x}\|_2^2, \quad \forall \mathbf{x} \in \mathcal{X}_S^N, \end{aligned} \quad (4.3)$$

where, \mathcal{X}_S^N refers to the set of S -sparse vectors in \mathbb{R}^N .

[115] Although both the smallest and largest singular values of $\Phi_S^T \Phi_S^{-1}$ affect the stability of the reconstruction algorithms, the smaller eigenvalue is dominant for compressed sensing in that it allows distinguishing between sparse vectors, \mathcal{X}_S^N , given their measurements by Φ .

A sensor selection method inspired by the RIP of a matrix is designed. The goal is to reduce a

¹ \mathbb{S} represents a set with cardinality of S and Φ_S represents any combination of columns of Φ .

measurement matrix to only a small fraction of its rows, while optimizing the proposed RIP-based criterion. In other words we aim to reduce number of equations such that the reduced system of equations would be a well-conditioned inverse problem.

For many scenarios, the big data are modeled by matrices and tensors. While conventional numerical algebra has been of interest for decades in many fields of sciences, it has been revisited for analysis of large datasets. For example algebraic tools such as singular value decomposition and subspace clustering are well-known methods for data mining, however their essential considerations for big data analysis are studied recently under the context of big data [116, 117, 118]. To this aim, parallel, distributed, scalable, and randomized algorithms are developed based on novel optimization strategies such as ADMM (alternating direction method of multipliers) [119, 120, 121]. Selection strategies are helpful for big data analysis and there is a strong connection between matrix subset selection and other analysis methods based on low-rank data expression [31]. A modified matrix subset selection is proposed in Chapter III of [122] in which big data considerations are addressed by a randomized approach. In this section, a successive and a parallel algorithm are proposed to tackle big data scenarios. The parallel algorithm is designed based on distribution of data on machines. Moreover, theoretical bounds are studied.

The main objectives are summarized as,

- The link between matrix subset selection, especially volume sampling and sensor selection, is investigated,
- A new criterion for matrix subset selection is proposed, which results in a new sensor selection method,
- The suitability of the E-optimal criterion is discussed, which is equivalent to optimization of an upper bound for RIP coefficients in compressive sensing literature,

- An approximation for RIP coefficients is proposed and utilized to extend E-optimality to an RIP-based criteria, and
- Successive and parallel algorithms are proposed as practical algorithms for selection from large data sets. Their performances are compared with the centralized algorithm.

Problem Statement and Related Work

Solving the sensor selection problem by evaluating the performance for each of the possible choices of $\binom{M}{K}$ is impractical unless the sizes are sufficiently small.

Suppose we want to estimate a vector $\mathbf{x} \in \mathbb{R}^N$ from M linear measurements, corrupted by additive noise, given by

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\nu}, \quad (4.4)$$

where, $\mathbf{y} \in \mathbb{R}^M$ and $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\boldsymbol{\nu}$ is normally distributed with zero mean and σ^2 variance. In other words, we want to only select just K rows of \mathbf{A} to have K measurements out of maximum M measurements. The maximum likelihood (ML) estimator is given by [15],

$$\hat{\mathbf{x}}_{ML} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}. \quad (4.5)$$

The estimation error $\mathbf{x} - \hat{\mathbf{x}}$ has zero mean and the covariance matrix is equal to

$$\boldsymbol{\Sigma}_{ML} = \sigma^2 (\mathbf{A}^T \mathbf{A})^{-1}. \quad (4.6)$$

To involve selection operator in the equations let us first write the ML solution as follows,

$$\hat{\mathbf{x}}_{ML} = \left(\sum_{m=1}^M \mathbf{a}_m \mathbf{a}_m^T \right)^{-1} \sum_{m=1}^M y_m \mathbf{a}_m, \quad (4.7)$$

where, \mathbf{a}_m^T is the m^{th} row of \mathbf{A} . The estimation error is distributed in a high dimensional ellipsoid

that its center is located at origin and its shape is according to the covariance matrix of error [15]. Minimization of volume of this ellipsoid (D-optimality) is the heuristic used in [15] that results in the following problem:

$$\begin{aligned} \hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \log \det \left(\sum_{m=1}^M w_m \mathbf{a}_m \mathbf{a}_m^T \right)^{-1}, \\ \text{subject to } \|\mathbf{w}\|_0 = K \text{ and } \mathbf{w} \in B^M, \end{aligned} \quad (4.8)$$

where \mathbf{w} determines whether or not each column is involved and $B = \{0, 1\}$.

The computationally tractable algorithms are divided into two main categories, convex relaxation and greedy selection. The first approach approximates the search space to the nearest convex set and exploits convex optimization methods to solve the problem, while greedy methods gradually select suitable sensors or prune inefficient ones.

Convex Relaxation

A convex relaxation for (5.2) is proposed in [15] as given by

$$\begin{aligned} \hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \log \det \left(\sum_{m=1}^M w_m \mathbf{a}_m \mathbf{a}_m^T \right)^{-1}, \\ \text{subject to } \|\mathbf{w}\|_1 = K \text{ and } \mathbf{w} \in C^M, \end{aligned} \quad (4.9)$$

for which ℓ_0 norm is replaced by ℓ_1 norm and C , the continuous set $[0, 1]$, is used instead of B . Another heuristic (A-optimality) is proposed in [123] based on minimization of $\text{MSE} = E[\|x -$

$\hat{x} \|_2^2] = \sigma^2 \text{tr}(\sum_{m=1}^M \mathbf{a}_m \mathbf{a}_m^T)^{-1}$ given by,

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\text{argmin}} \|\mathbf{w}\|_1, \\ \text{subject to } \quad & \text{tr}(\sum_{m=1}^M w_m \mathbf{a}_m \mathbf{a}_m^T)^{-1} \leq \eta \text{ and } \mathbf{w} \in C^M, \end{aligned} \quad (4.10)$$

where, η is a regularization parameter. As η increases, the number of selected sensors would be decreased. There is a performance gap between the best subset and the heuristic solution of the convex relaxation for maximizing the volume. Although simulations show the gap is small in many cases, there is no guarantee that the gap between the performance of the chosen subset and the best performance is always small [15].

Greedy Algorithms

The greedy algorithms are faster than convex relaxation methods in addition to providing some guarantees for the optimality of the solution in the case of a sub modular condition [124]. For example, it is possible to rewrite (5.2) as the following sub-modular problem [16],

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\text{argmax}} \log \det(\sum_{m=1}^M w_m \mathbf{a}_m \mathbf{a}_m^T), \\ \text{subject to } \quad & \|\mathbf{w}\|_0 = K \text{ and } \mathbf{w} \in B^M. \end{aligned} \quad (4.11)$$

To solve this problem, we can select sensors sequentially. At the step t , a sensor will be selected that maximizes $\log \det\{(\sum_{m=1}^{t-1} \mathbf{a}_{S_m} \mathbf{a}_{S_m}^T) + \mathbf{a}_z \mathbf{a}_z^T\}$ with respect to \mathbf{a}_z in which S_m stacks the indices of the selected sensors in previous iterations and the obtained z is the index of the new selected sensor. Solving the maximization results in \mathbf{a}_{S_t} . This procedure will continue till $t = K$.

Matrix subset selection

The sensor selection problem is highly related to column/row sub-matrix selection, a fundamental problem in applied mathematics. There exists many efforts in this area [9, 27, 125, 50]. Generally, they aim at devising a computationally efficient algorithm in which the span of the selected columns/rows cover the columns/rows space as close as possible. Mathematically, a general guarantee can be stated as one of the following forms [27, 126],

$$\begin{aligned}\mathbb{E}\{\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2\} &\leq (K + 1)\|\mathbf{A} - \mathbf{A}_K\|_F^2, \\ \|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2 &\leq p(K, M, N)\|\mathbf{A} - \mathbf{A}_K\|_F^2,\end{aligned}$$

in which, $\pi_{\mathbb{T}}(\mathbf{A})$ represents projection of rows of \mathbf{A} on to the span of selected rows indexed by \mathbb{T} set. \mathbb{E} indicates expectation operator with respect to \mathbb{T} , i.e., all the combinatorial selection of K rows of \mathbf{A} out of M . Moreover, $p(K, M, N)$ is a polynomial function of the number of selected elements, the number of columns and the number of rows. \mathbf{A}_K is the best rank- K approximation of \mathbf{A} that can be obtained by singular value decomposition. The first form suggests the distribution of potential sets for selection and it expresses an upper bound for expected value of error. The second form guarantees existence of a deterministic subset that bounds the error by a polynomial function of the parameters.

Volume sampling is the most well-known approach to achieve the desired selection that satisfies one of the aforementioned bounds. The following theorem expresses the probabilistic form volume sampling.

[[27]] Let \mathbb{T} be a random K -subset of rows of a given matrix \mathbf{A} chosen with probability

$$Pr(\mathbb{T}) = \frac{\det(\mathbf{A}_{\mathbb{T}}\mathbf{A}_{\mathbb{T}}^T)}{\sum_{|\mathbb{U}|=K} \det(\mathbf{A}_{\mathbb{U}}\mathbf{A}_{\mathbb{U}}^T)}$$

Then,

$$\mathbb{E}\{\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2\} \leq (K + 1)\|\mathbf{A} - \mathbf{A}_K\|_F^2.$$

Volume sampling considers more probability of selection for those rows whose volume is greater. The volume of a subset of a matrix, $\mathbf{A}_{\mathbb{T}}$, is proportional to the determinant of $\mathbf{A}_{\mathbb{T}}\mathbf{A}_{\mathbb{T}}^T$. Thus, (5.2) aims to find the most probable subset according to volume sampling.

Volume sampling and D-optimality pursue the same heuristic objective. This heuristic does not promote a well-shaped matrix for compressive sensing purposes based on RIP. However, the analysis of optimization w.r.t the RIP coefficient is not an easy task due to the columns combinatorial behavior in addition to row selection for the basic sensor selection problem. To eliminate the column combinations, we consider all of the columns and consequently we come up with an optimization problem w.r.t the minimum eigenvalue that is known as E-optimality in the optimization literature [17, 51]. Assume a simple selection from rows of $\mathbf{A} \in \mathbb{R}^{100 \times 3}$. Each row of \mathbf{A} , associated with a sensor, corresponds to a point in \mathbb{R}^3 . We are to select 2 sensors out of 100 based on D-optimality and E-optimality [51]. Both solutions are initialized by the same sensor (sensor 1) and the criteria for the next selection varies. The D-optimal solution aims to maximize the surrounded area (gray area in Fig. 4.1) which is vulnerable to be an ill-shaped area while, E-optimal solution comes up with a well-shaped area due to maximizing the minimum eigenvalue (shaded area in Fig. 4.1).²

²The presented intuition about D-optimality and E-optimality relates to the condition number of a matrix in linear algebra [127]. Diverged eigenvalues results in a large condition number and an ill-conditioned system of equations; accordingly, we refer to the polygon of an ill-conditioned system of equations as ill-shaped where the vertexes of shape are the rows of the matrix. On the other hand, close eigenvalues correspond to a small condition number and

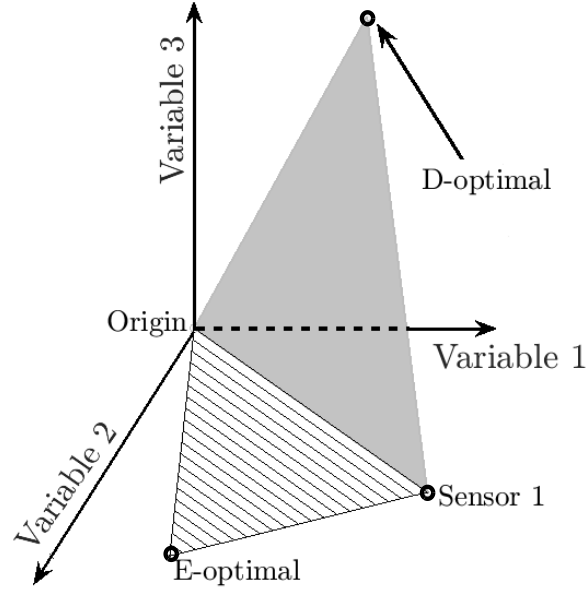


Figure 4.1: Comparison of D-optimality and E-optimality for selecting 2 sensors in the 3D space. The gray area is the maximum achievable area by selecting the second sensor based on D-optimality. The shaded area is a well-shaped polygon obtained by E-optimality.

The following simple example illustrates the effect of E-optimality. Consider two matrices, $\begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The determinant of both matrices are equal, thus D-optimality does not favor one over the other, however, the second matrix is optimum based on E-optimality.

As we will see in the next section, for selection of K rows of $\mathbf{A} \in \mathbb{R}^{M \times N}$, the E-optimal criterion is equivalent to optimizing the RIP coefficient of order N , which is an upper bound for any arbitrary order of RIP coefficients. In the next section E-optimality will be exploited to develop a new sampling method for which its performance guarantee is analyzed. E-optimal criterion suggests optimization of an upper bound for any order of RIP [7].

a well-conditioned system of equations. The corresponding polygon is referred as well-shaped in Fig 4.1. Having well-conditioned matrices, is a central concern in CS as evidenced by the role played by the RIP [128].

E-optimal sampling

Remark 1 promotes us to develop a new matrix subset selection method that reduces the matrix to have a well-conditioned sub-matrix in the CS sense. The dominant factor of RIP constant comes from the minimum eigenvalue of the reduced matrix. It suggests to exploit the following optimization problem for sensor selection,

$$\begin{aligned} \hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \quad & \left\| \left(\sum_{m=1}^M w_m \mathbf{a}_m \mathbf{a}_m^T \right)^{-1} \right\|, \\ \text{subject to} \quad & \|\mathbf{w}\|_0 = K \text{ and } \mathbf{w} \in B^M. \end{aligned} \quad (4.12)$$

In which, $\|\cdot\|$ denotes the spectral norm of a matrix that is defined as its maximum eigenvalue. The following lemma shows that the minimum eigenvalue is an upper bound for δ_S^L . For any $\mathbf{A} \in \mathbb{R}^{M \times N}$, the following inequality holds.

$$1 - \sigma_{\min}(\mathbf{A}\mathbf{A}^T) = \delta_N^L(\mathbf{A}) \geq \delta_{N-1}^L(\mathbf{A}) \geq \cdots \geq \delta_2^L(\mathbf{A}).$$

Proof: According to the definition of RIP constant δ_S and considering that the set of at most $S-1$ non-zero vectors is subset of the set of at most S non-zero vectors, it easily concluded that $\delta_S(\mathbf{A}) \geq \delta_{S-1}(\mathbf{A})$ for any $S = 2, \dots, N$.

Lemma suggests that E-optimality, i.e., minimization of δ_N^L , actually is equivalent an upper bound for an arbitrary order of RIP coefficient.

Similar to volume sampling, we design a probability of sampling according to their minimum eigenvalue.

Definition 3 Given a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, E-optimal sampling is defined as picking a subset of \mathbb{T} with the following probability,

$$Pr(\mathbb{T}) = \frac{\sigma_{\min}^2(\mathbf{A}_{\mathbb{T}})}{\sum_{|\mathbb{U}|=K} \sigma_{\min}^2(\mathbf{A}_{\mathbb{U}})}.$$

Definition 4 Given a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\bar{\delta}_K^L$ is defined as one minus the mean of minimum eigenvalues of \mathbf{A} 's sub-matrices with K columns. Mathematically, it can be expressed as follows,

$$\bar{\delta}_K^L(\mathbf{A}) = 1 - \mathbb{E}\{\sigma_{\min}^2(\mathbf{A}_{\mathbb{S}})\},$$

in which \mathbb{S} indicates a subset of K columns of \mathbf{A} .

Definition 5 [111] Given a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, the spark of \mathbf{A} is defined as the smallest number of columns that are linearly dependent. It can be stated as follows,

$$Spark(\mathbf{A}) = \min \|\mathbf{x}\|_0 \quad s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{0} \quad \text{and} \quad \mathbf{x} \neq \mathbf{0}.$$

The upper bound for spark is the rank of matrix plus 1. However any linear dependencies among some columns of the matrix may decrease the spark. Based on the above definitions we present the following theorem that expresses an upper bound for projection error of E-optimal sampling [7]. Assume spark of $\mathbf{A} \in \mathbb{R}^{M \times N}$ is greater than $K + 1$. E-optimal selection of K rows implies

$$\mathbb{E}\{\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2\} \leq \frac{M - K}{C(K + 1)} \frac{1 - \bar{\delta}_{K+1}^L(\mathbf{A}^T)}{1 - \bar{\delta}_K^L(\mathbf{A}^T)},$$

where C is a positive number a function of the dependencies of rows.

Proof: See appendix.

Table 4.1: Complexity of different selection strategies.

Algorithm	Complexity
Convex Optimization [15]	$O(M^3)$
Volume sampling [9]	$O(KNM^2\log M)$
Greedy Submodular Selection [16]	$O(MK^3)$
Greedy E-optimal selection (proposed)	$O(MNK^2)$

E-optimal sampling implies an upper bound for the expectation of projection error in a probabilistic manner. However, we need to select some sensors deterministically. To this aim, we propose the following iterative algorithm. Actually, this algorithm is an approximation for the maximum likelihood estimator in which the likelihood comes from the suggested probability in Definition 3.

Table 4.1 compares computational burden of three well-known selection methods with the proposed method. Convex relaxation is not able to work effectively for big data sets since the complexity of the algorithm grows with M^3 [15]. Complexity of volume sampling also depends on M^2 . Likewise, complexity of greedy algorithms which process data one-by-one increase linearly w.r.t size of data. However, in some big data scenarios we still need to decrease computational complexity w.r.t data size. To this aim in Section , two remedies are studied based on data partitioning.

Algorithm 3 Greedy E-Optimal Sensor Selection

Require: \mathbf{A} and K

- 1: **Initialization:** \mathbb{S} with a random sensor
 - 2: for $k = 1, \dots, K$
 - 3: for $m = 1, \dots, M$
 - 4: $\mathbb{T} = \mathbb{S} \cup \{m\}$
 - 5: $p(m) = \sigma_{\min}(\mathbf{A}_{\mathbb{T}})$
 - 6: end
 - 7: $s_k = \operatorname{argmax}_m p(m)$
 - 8: $\mathbb{S} = \mathbb{S} \cup s_k$
 - 9: end
-

RIP-Based Sensor Selection

The structure of the reduced measurement matrix plays a critical role in sparse recovery [129, 130]. Several criteria have been suggested to evaluate suitability of a measurement matrix including the mutual coherency and the RIP coefficient. In order to guarantee a well-conditioned matrix to recover a S -sparse vector, the criteria based on RIP depend on the RIP constant of order PS . Different guarantees suggest some bounds in terms of δ_{2S} , δ_{3S} and δ_{4S} , i.e., δ_{PS} for $P = 2, 3, 4$ [113] [114]. As Remark 1 suggests, the lower RIP constant defined in (4.3) is the dominant factor for compressive sensing. Thus, we employ the lower constant of order PS in (4.2) denoted by δ_{PS}^L (4.3) to propose the following problem for sensor selection,

$$\begin{aligned} \hat{\mathbf{W}} = \underset{w_{km} \in \{0,1\}}{\operatorname{argmin}} \delta_{PS}^L(\mathbf{W}\mathbf{A}), \\ \text{subject to } \|\mathbf{w}_k\|_0 = 1 \ \forall k = 1, \dots, K. \end{aligned} \quad (4.13)$$

In which $\mathbf{W} \in \mathbb{R}^{K \times M}$ reduces the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ by some selected rows. In other words, matrix \mathbf{W} is the selector operator and the goal of sensor selection is to estimate this matrix. P is a constant between 2 and 4 and \mathbf{w}_k is the k^{th} row of \mathbf{W} . In each row of \mathbf{W} there is only one entry 1 and all the other entries are zero, i.e., $\|\mathbf{w}_k\|_0 = 1$. According to the definition of RIP, the above problem can be cast to the following form,

$$\begin{aligned} \hat{\mathbf{W}} = \underset{w_{km} \in \{0,1\}}{\operatorname{argmax}} \min_{\mathbf{x}} \|\mathbf{W}\mathbf{A}\mathbf{x}\|_2^2, \\ \text{subject to } \|\mathbf{w}_k\|_0 = 1, \|\mathbf{x}\|_2 = 1 \text{ and } \|\mathbf{x}\|_0 \leq PS. \end{aligned} \quad (4.14)$$

This problem is a jointly combinatorial search with respect to both \mathbf{W} and \mathbf{x} . It is shown that finding the solution with respect to \mathbf{x} is NP-hard with a fixed \mathbf{W} [131]. On the other hand, with

a fixed \mathbf{x} , it is easy to show that the problem is sub-modular with respect to \mathbf{W} . The reduction matrix selects the most significant entries of the error $\mathbf{y} - \mathbf{A}\mathbf{x}$. In the next section we will propose an optimization algorithm that first approximates the solution w.r.t \mathbf{x} and then pursues a greedy method to update \mathbf{W} . Please note that by ignoring the last constraint, the problem turns into the E-optimal sensor selection.

Although matrix subset selection and sensor selection formulation are highly related to each other, they have their own approaches to the problem. Sensor selection aims to reduce a system of equations which is not specified for a fixed unknown vector. For instance, in Problem (4.14) we minimize w.r.t \mathbf{x} and Problem (5.2) is derived by minimizing expectation of estimation error of \mathbf{x} . However, a specific \mathbf{x} generates the corresponding values of potential sensors. So far we have assumed that we do not optimize the problem for a specific observed \mathbf{y} . If we have access to the measurements in a fusion center, we can exploit this information in the selection decision. To consider more valuable measurements, their values are involved in the following problem in which we call it data-aware RIP based sensor selection.

$$\begin{aligned} \hat{\mathbf{W}} = & \underset{w_{km} \in \{0,1\}}{\operatorname{argmax}} \min_{\mathbf{x}} \|\mathbf{W}\mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{W}(\mathbf{y} - \mathbf{A}\mathbf{x}(\odot))\|_2^2, \\ \text{s.t. } & \|\mathbf{w}_k\|_0 = 1, \|\mathbf{w}^{(m)}\|_0 \leq 1, \|\mathbf{x}\|_2 = 1 \text{ and } \|\mathbf{x}\|_0 \leq PS. \end{aligned} \quad (4.15)$$

This problem promotes the sensor selection to select sensors from areas with high vulnerability to error. In a same time, their corresponding bases construct a well-conditioned matrix based on RIP coefficient. $\mathbf{w}^{(m)}$ denotes the m^{th} column of \mathbf{W} . The constraint, $\|\mathbf{w}^{(m)}\|_0 \leq 1$ avoids repetitive selection of the same sensor. Note that repetitive selection may occur for large values of λ and there is no need for this constraint in (4.14) because a repetitive column results in a zero eigenvalue

while the cost function maximizes the minimum eigenvalue. By considering the model's error, we aim to compensate the error of model by an intelligent sensor selection. Aggregating all sensors' measurements in a fusion center is in contrast with the goal of sensor selection. However, we devise a dynamic framework that needs a partial set of sensors for adapting the sensor selection algorithm with the dynamic of the sensors. These measurements might be derived by a low-frequency sampling from all sensors or set of recent measured sensors. \mathbb{O} denotes the set of observed measurements and $\mathbf{x}(\mathbb{O})$ refers to the estimation based on the partial observed data. $\mathbf{A}\mathbf{x}(\mathbb{O})$ indicates the approximation of the measurements \mathbf{y} . $\mathbf{x}(\mathbb{O})$ is obtained by solving the following regression problem.

$$\mathbf{x}(\mathbb{O}) = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y}_{\mathbb{O}} - \mathbf{A}_{\mathbb{O}}\mathbf{x}\|_2^2 + \lambda_{LASSO}\|\mathbf{x}\|_1. \quad (4.16)$$

Where λ_{LASSO} regularizes sparsity. In order to obtain the error of model, we need to observe all the sensors, while we aim to keep the number of observed sensors limited. The following interpolation in terms of the observed sensors is exploited to derive the error of model for all sensors.

$$y_m = \frac{\sum_{j \in \mathbb{O}} \gamma_{mj} y_j}{\sum_{j \in \mathbb{O}} \gamma_{mj}}, \quad (4.17)$$

where γ_{mj} is a similarity function between m^{th} and j^{th} sensor. The estimated observation of unobserved sensors help us to evaluate their fidelity to the model. E.g., if the interpolated measurement of the m^{th} sensor, y_m , also satisfies $y_m \approx \mathbf{a}_m^T \mathbf{x}(\mathbb{O})$, it implies that this sensor can be predicted by some other sensors based on the model. Thus, this sensor is reliable and it does not maximize the cost function (4.15) significantly. This data-driven approach is inspired by dynamic sensor selection introduced in [132, 133]. For a given model \mathcal{M} on the data, dynamic sensor selection determines set \mathbb{S} such that the estimation error of the rest of sensors, \mathbb{S}^c , is minimized. The estimation is obtained based on the model, \mathcal{M} , and observed sensors, \mathbb{S} [132]. The assumed model in

our proposed approach is indicated in (4.4).

The parameter λ in (4.15) regularizes the weight of the energy of error and the RIP coefficient of selected bases. In other words, \mathbf{W} reduces the rows of \mathbf{A} in an optimal sense and simultaneously, it selects some vulnerable sensors to model's error. In the experimental results we will show the effect of the regularization parameter. According to our simulations, the importance of the main term of objective function is more than the energy of the model's error. Even by $\lambda = 0$ we have a well-spread set of selected sensors corresponding to a well-conditioned system of equations while, by $\lambda \rightarrow \infty$ a set of concentrated sensors would be concluded which corresponds to an ill-posed system of equations. Simulations show a relatively wide range of λ could be a good choice.

Finding RIP of a matrix requires solving an NP-hard problem [131]. Thus, for a large-scale problem, it is not feasible to search among all the subsets. A greedy algorithm is proposed to approximate the RIP of a matrix. To this end, let us consider the following problem.

$$\begin{aligned} \delta_{PS}(\mathbf{A}) = \\ 1 - \min \|\mathbf{A}\mathbf{x}\|_2^2 \quad \text{st: } \|\mathbf{x}\|_2 = 1 \text{ and } \|\mathbf{x}\|_0 \leq PS. \end{aligned} \tag{4.18}$$

The solution is approximated in (4.19). The suggested problem neglects the last constraint in (4.29) and obtains a solution, then projects the obtained solution to the feasible set spanned by the neglected constraint.

$$\begin{aligned} \tilde{\delta}_{PS}(\mathbf{A}) = \\ 1 - \|\mathbf{A}\Omega_{\ell_2}(T_{PS}\{\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x}\|_2^2 \text{ st: } \|\mathbf{x}\|_2 = 1 \})\|_2^2. \end{aligned} \tag{4.19}$$

In which, $T_{PS} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is the truncate function that keeps only PS most significant entries and makes the rest zero. As the truncated vector no longer satisfies the unit norm constraint, $\Omega_{\ell_2} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ normalizes the truncated vector to the unit ℓ_2 ball. The solution of the alternative problem denoted by $\tilde{\delta}_{PS}(\mathbf{A})$ can be solved efficiently using singular value decomposition.

$$\begin{aligned}\tilde{\delta}_{PS}(\mathbf{A}) &= 1 - \|\mathbf{A}\mathbf{x}^*\|_2^2, \quad \mathbf{x}^* = \Omega_{\ell_2}(T_{PS}\{\mathbf{U}(:, k)\}), \\ \mathbf{A} &= \mathbf{V}^T \mathbf{\Lambda} \mathbf{U},\end{aligned}\tag{4.20}$$

in which, $\mathbf{U}(:, k)$ is the k^{th} column³ of \mathbf{U} . In other words, \mathbf{x}^* is obtained by setting it to the normalized and truncated Eigenvector corresponding the minimum Eigenvalue. By exploiting the approximation of δ_{PS} the sensor selection problem can be cast as the following form,

$$\hat{\mathbf{W}} = \underset{w_{km} \in \{0,1\}}{\operatorname{argmin}} \tilde{\delta}_{PS}(\mathbf{W}\mathbf{A}) \quad \text{st: } \|\mathbf{w}_k\|_0 = 1, \forall k = 1, \dots, K. \tag{4.21}$$

By using the obtained approximation in (4.20), we conclude

$$\hat{\mathbf{W}} = \underset{w_{km} \in \{0,1\}}{\operatorname{argmax}} \|\mathbf{W}\mathbf{A}\mathbf{x}^*\|_2^2 \quad \text{st: } \|\mathbf{w}_k\|_0 = 1, \forall k = 1, \dots, K, \tag{4.22}$$

³The k^{th} column is represented by $\mathbf{U}(:, k)$ and the k^{th} row is represented by $\mathbf{U}(k, :)$ in Algorithms 1 and 2. Moreover, $\mathbf{U}(\mathbb{S}, :)$ represents the reduced matrix by some selected rows indicated by \mathbb{S} set.

in which,

$$\begin{aligned}\mathbf{x}^* &= \Omega_{\ell_2}(T_{PS}\{\mathbf{U}(:, K)\}) \\ \mathbf{W}\mathbf{A} &= \mathbf{V}^T\mathbf{\Lambda}\mathbf{U}.\end{aligned}\tag{4.23}$$

Algorithm 4 shows the steps of our proposed greedy algorithm to solve the obtained optimization problem. To evaluate each sensor we need to compute the most dominant k eigen components which implies performing singular value decomposition (SVD). However, truncated SVD up to the k^{th} component will be sufficient. A similar algorithm can be used to solve Problem (4.15). To this aim, Step 6 in Algorithm 4 should be modified to consider the error of m^{th} sensor, i.e., $p(m) = \|\mathbf{x}^*\|_2^2 + \lambda|y_m - \mathbf{a}_m^T \mathbf{x}(\mathbb{O})|$. However, it is not practical to have all the measurements at the fusion center. An online algorithm is proposed that observes one new measurement sequentially. In each sequence, the observed set of sensors is updated and this set is initialized by the output of Algorithm 4. In other words, the selected sensors in Algorithm 4 are sensed. Our data-aware algorithm needs an approximation of the observed data in terms of the corresponding reduced \mathbf{A} using (4.31).

As mentioned in the last section, the online data-aware framework, Algorithm 5, uses an interpolation as the prediction of unobserved measurements. It will be an enabling step for estimation of model's error in order to adapt the sensor selection to the measurements. The interpolation is based on weighted averaging of observed measurements where the weight is a similarity metric that depends on the underlying application. For example, we consider a simple channel gain between two sensors in CRNs simulations which is an inverse function of distance as the similarity criterion in 4.17.

The bottleneck of complexity order of Algorithm 4 at the k^{th} iteration is performing a truncated singular

value decomposition to obtain the first k eigen components. Thus, the complexity of the algorithm in the k^{th} iteration will be $O(kMN^2)$ [134]. Therefore, selection of K sensors implies complexity order of $O(K^2N^2M)$.

Algorithm 4 and Algorithm 5 can be implemented in a distributed manner similar to the proposed idea in Section . The selection procedure is as same as before in each machine but the number of data are decreased by factor C which is the number of machines. This makes complexity to $O(K^2N^2M')$ where, $M' = M/C$.

Distributed Implementation

Data summarizing is an enabling step for more complicated processing procedures. For example, computational burden for training a recognition system increases tremendously by the size of the training data. However, in some cases even data summarizing is not tractable due to the size of data. A naive approach for data summarizing is randomly sampling from data to make it sufficiently small.

There exist some attempts to design randomized algorithms for matrix subset selection [126]. The idea is based on combining deterministic and randomized methods, using a two-phase algorithm.

Algorithm 4 The blind RIP-based Sensor Selection

Require: \mathbf{A} , S and K

- 1: **Initialization:** $\mathbf{W} = \mathbf{0} \in \mathbb{R}^{K \times M}$ and $\mathbb{S} = \emptyset$
 - 2: for $k = 1, \dots, K$ (Optimization of the k^{th} row of \mathbf{W})
 - 3: for $m = 1, \dots, M$
 - 4: SVD on $\mathbf{A}(\mathbb{S} \cup m, :)$ to obtain \mathbf{U} in (4.23)
 - 5: $\mathbf{x}^* = \Omega_{\ell_2}(T_{PS}\{\mathbf{U}(:, k)\})$
 - 6: $p(m) = \|\mathbf{A}\mathbf{x}^*\|_2^2$
 - 7: end
 - 8: $s_k = \operatorname{argmax}_m p(m)$
 - 9: $\mathbb{S} = \mathbb{S} \cup s_k$ and $\mathbf{W}_{k, s_k} = 1$
-

Algorithm 5 The data-aware RIP-based Sensor Selection

Require: \mathbf{A} , S , K , λ and λ_{LASSO}

Initialization: \mathbb{O} = Output of Algorithm 1

- 2: while $\mathbb{O} \neq \{1, \dots, M\}$
 - $\mathbf{W} = \mathbf{0} \in \mathbb{R}^{K \times M}$, $\mathbb{S} = \emptyset$
 - 4: Observe 1 new measurement and update \mathbb{O}
Interpolate $\mathbf{y}_{\overline{\mathbb{O}}}$ using $\mathbf{y}_{\mathbb{O}}$ using (4.17)
 - 6: for $k = 1, \dots, K$
 for $\forall m \in \mathbb{S}^c$
 - 8: SVD on $\mathbf{A}(\mathbb{S} \cup m, :)$ to obtain \mathbf{U} in (4.23)
 $\mathbf{x}^* = \Omega_{\ell_2}(T_{PS}\{U(:, k)\})$
 - 10: $\mathbf{x}(\mathbb{O}) = LASSO(\mathbf{A}, \mathbf{y}_{\mathbb{O}}, \lambda_{LASSO})$ using (4.31)
 $p(m) = \|\mathbf{A}\mathbf{x}^*\|_2^2 + \lambda|y_m - \mathbf{a}_m^T \mathbf{x}(\mathbb{O})|$
 - 12: end
 $s_k = \operatorname{argmax}_m p(m)$
 - 14: $\mathbb{S} = \mathbb{S} \cup_{s_k}^m$ and $\mathbf{W}_{k, s_k} = 1$
 end
 - 16: $\mathbb{O} = \mathbb{O} \cup \mathbb{S}$ and return to 2.
-

The first phase selects $O(k \log(k))$ rows of the matrix. Then, deterministic subset selection finds exactly the k most informative rows of the matrix. This randomized algorithm achieves the following bound [126],

$$\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2 \leq O(k \log^{\frac{1}{2}}(k)) \|\mathbf{A} - \mathbf{A}_K\|_F^2,$$

This bound suggests us that a judiciously or even randomly set of rows of \mathbf{A} can provide us a submatrix with a close subspace to the original matrix. The submatrix might be more convenient to deal with, specially when the data size is big. In this section, data partitioning is studied as an enabling step for successive and parallel processing.

Successive Processing

In order to make the problem tractable, we can employ a method based on successive processing of partitioned data. Suppose data matrix \mathbf{A} is partitioned into C blocks that each block, \mathbf{A}_c , contains M_c rows of \mathbf{A} . At the first stage K rows are selected out of M_1 rows of the first partition. The selected rows are forwarded to the next stage in order to perform selection among M_2 data of the second part, as well as the already K selected rows. It means at the second stage there are $M_2 + K$ data and the goal is to select only K rows to feed to the next stage. Alg. 6 shows the steps of successive E-optimal sensor selection algorithm. In the experimental results section the performance of this method will be presented.

In addition to the successive method, there is another solution for scenarios that data can be independently processed over distributed machines in a parallel manner. The successive approach performs a series of selection procedures and all of these procedures can be implemented in a same machine. However, in some scenarios we have access to multiple processing nodes in a network. In this case it is desired to implement a distributed algorithm, which is able to process different part of data simultaneously. We study two methods for distributing data, random partitioning and designed partitioning.

Algorithm 6 Successive E-optimal row selection

Require: \mathbf{A} , C , and K

- 1: **Initialization:** \mathbb{S} by \emptyset .
 - 2: Partition \mathbf{A} to C parts (\mathbb{A}_c indicates the indices of \mathbf{A}_c).
 - 3: for $c = 1, \dots, C$
 - 4: $\mathbb{Z} = \mathbb{S} \cup \mathbb{A}_c$
 - 5: $\mathbb{S} \leftarrow$ select K rows of $\mathbf{A}_{\mathbb{Z}}$ using Alg. 1.
 - 6: end
-

Random Partitioning

In this section, the given matrix, \mathbf{A} , is randomly broken into $\{\mathbf{A}_c\}_{c=1}^C$, in which each submatrix contains M_c rows of the original matrix. In order to ensure that row space of each submatrix is close enough to the row space of the original matrix, we need to derive a lower bound on the number of members of each submatrix. To this aim we assume union of subspaces model for the whole data.

Assumption 1: The matrix \mathbf{A} can be expressed as a union of subspaces, i.e., $\mathbf{A} = [\mathbf{U}_1 \mathbf{Q}_1, \dots, \mathbf{U}_L \mathbf{Q}_L]^T$. Assume rank of \mathbf{A} is R and rank of each subspace is $\frac{R}{L}$, where, $\{\mathbf{U}_l \in \mathbb{R}^{N \times \frac{R}{L}}\}_{l=1}^L$ and $\{\mathbf{Q}_l \in \mathbb{R}^{\frac{R}{L} \times M'}\}_{l=1}^L$, and $M' = M/L \gg \frac{R}{L}$.

Assumption 1 implies that the original matrix, \mathbf{A} , is a union of L subspaces in which intrinsic dimension of each subspace is at most R/L . This assumption is reasonable for many scenarios in signal processing and data mining [135, 136]. The following lemma suggests an upper bound for the number of parallel machines in order to ensure that the row space of each portion of data is equal to the original data with a high probability.

Assume \mathbf{A} follows Assumption 1. If the rows of \mathbf{A} are equally partitioned among C parts and samples of each part are drawn uniformly at random and C satisfies the following inequality,

$$C \leq \frac{M}{L\xi(2 + (3/\xi)\log\frac{2L}{\delta})}, \quad (4.24)$$

where,

$$\xi = 10\gamma \max(R/L, \log M/L) \log \frac{2R}{\delta},$$

then the row space of each part spans row space of \mathbf{A} with probability at least $1 - 2\delta - 2\frac{L^4}{M^3}$. *proof:* See Appendix.

Proposition 1 *The order of minimum number of samples for each parallel machine is $O(R)$ in order to make sure that the span of selected rows is equal to that of the original matrix in each machine with a high*

probability.

This proposition is clearly derived by the steps of proof of Lemma in the appendix. It suggests that each machine needs a portion of data such that the required size of each portion is linearly dependent to the rank of the original matrix.

Assume K samples are drawn from each partition and KC samples are selected in the first phase. The second phase aims to select only the K most informative samples among the initial selection. Volume sampling and the proposed sampling method select the corners of data such that the selected points constructs a polygonal in which their vertexes are far from each other. However, the selected point could be outlier data, i.e., data is not concentrated about some selected samples. We need to ensure that each selected point represents a relatively large number of non-selected data. Selection algorithms that work based on relative structure of samples are complicated and they can not be used for the big data regime. To tackle this problem a concentration-based selection is performed in the second phase of selection on the KC selected data. K-medoids clustering is a generalization of K-means in which the data centers are selected from the sample points of data. In the first phase we ensure that all the vertexes of the hull of data are selected and in the second phase K-medoids algorithm shrinks the selected data to only K samples. This two-phase algorithm is the practical application which can be exploited for big data sets. As we will see in the simulation results, the overall two-phase process is faster than performing selection on the whole data using Alg. 3 and it is much faster than performing k-medoids algorithm for whole data. Alg. 7 shows steps of the proposed two-phase algorithm for selecting from big data. This algorithm is the robust and practical version of Alg. 3 for real scenarios which a huge number of noisy data are given.

Designed Partitioning

The best case scenario of data structure and partitioning is illustrated here in order to show an intuition on appropriate partitioning of data. Assume the data can be modeled by a union of orthogonal subspaces as described in the following assumption,

Algorithm 7 two-phase selection algorithm

Require: \mathbf{A} , K , C .

- 1: Assign $\mathbf{A}^{(c)} \forall c = 1, \dots, C$.
 - 2: for $c = 1, \dots, C$
 - 3: $\mathbf{U}^{(c)} \leftarrow \text{Algorithm 1}(\mathbf{A}^{(c)}, K)$.
 - 4: End for
 - 5: $\mathbf{U} = [\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(C)}]$.
 - 6: K-medoids to select K data from \mathbf{U} .
 - 7: END
-

Assumption 2: The matrix \mathbf{A} can be expressed as a union of orthogonal subspaces, i.e., $\mathbf{A} = [\mathbf{U}_1 \mathbf{Q}_1, \dots, \mathbf{U}_L \mathbf{Q}_L]$ and $\mathbf{U} = [\mathbf{U}_1, \dots, \mathbf{U}_L]$ is an orthogonal matrix.

Although this structure is restricted and real data do not follow the orthogonality constraint, it suggests us a sub-optimum data partitioning strategy. A subspace clustering algorithm as a partitioning method can reveal us the optimum partitions where the number of optimum parts is equal to L in Assumption 1. The following lemma shows that the optimum selection of K rows from all rows of \mathbf{A} is equivalent to the selection from only KC rows that found by C parallel machines, each machine reporting K rows. Assume \mathbf{A} follows Assumption 2 and it is partitioned by subspace clustering where $C = L$. Let \mathbb{S}^* denotes the solution of the following problem,

$$\underset{|\mathbb{S}|=K}{\operatorname{argmin}} \|\mathbf{A} - \pi_{\mathbb{S}}(\mathbf{A})\|_F^2. \quad (4.25)$$

Each parallel machine has to select K rows by solving following problem,

$$\mathbb{S}_c = \underset{\mathbb{S} \subset \mathbf{A}_c}{\operatorname{argmin}} \|\mathbf{A}_c - \pi_{\mathbb{S}}(\mathbf{A}_c)\|_F^2 \quad \text{s.t. } |\mathbb{S}| = K. \quad (4.26)$$

Then,

$$\mathbb{S}^* \subset \bigcup_{c=1}^C \mathbb{S}_c.$$

proof: See appendix.

Inspired by Lemma , we suggest to partition rows of \mathbf{A} according to the underlying subspaces and select K rows at each distributed node. The results of each distributed selection are aggregated in a center and the

final selection aims to select K rows from only those KC rows rather than all the rows of \mathbf{A} . Algorithm 8 summarizes the steps of partitioning based on subspace identification. This algorithm at each iteration, finds a subspace of \mathbf{A} and then it finds all the rows that lie on this subspace. These rows construct a partition. The rows of the found partition are removed from \mathbf{A} and this procedure is applied on the rest of rows, until all of rows are partitioned. At the stage 4 of the algorithm, inspired by Proposition 1, we need to take $O(R)$ rows in order to ensure that the sampled rows spans sufficiently close subspaces to the whole set of rows. At the stage 5, any subspace clustering method can be applied. This topic is a well-studied research and there are well-known algorithms such as K-subspaces, and sparse subspace clustering [137, 11].

Random partitioning requires selection of sufficiently large number of rows in order to be sure that all of the subspaces of rows are spanned. However, the designed partitioning needs only spanning a specific subspace. Aggregating rows belonging same subspace in the same partition results in more accurate distributed selection. However, this method needs prior processing on the top of the main selection algorithm.

Algorithm 8 Subspace-based data partitioning

Require: \mathbf{A} , R , ϵ .

- 1: $\mathbf{A}^{(1)} \leftarrow \mathbf{A}$
 - 2: $c = 1$
 - 3: WHILE all rows are not partitioned.
 - 4: $\mathbb{D} \leftarrow$ Select sufficiently large number of rows from $\mathbf{A}^{(c)}$.
 - 5: $\mathbf{U}_c \leftarrow$ Identify 1 subspace of $\mathbf{A}_{\mathbb{D}}$
 - 6: $\mathbb{A}_c \leftarrow$ Determine rows of $\mathbf{A}^{(c)}$ that lie on subspace \mathbf{U}_c by distance less than ϵ .
 - 7: $\mathbf{A}^{(c+1)} \leftarrow$ Remove rows of \mathbb{A}_c from $\mathbf{A}^{(c)}$.
 - 8: $c \leftarrow c + 1$
 - 9: END
-

Reliability Estimation and Dynamic Sensor Selection

Collaborative sensor networks may collect redundant information which results in a larger number of sensor nodes than is needed. While, pruning unnecessary data is essential, Algorithm 3 is measurement-independent and it reduces the underlying equations of the network to shrink the equations to a well-

conditioned set of sub-equations regardless of dynamic of the network. This measurement-independent approach is optimal in an averaged sense, i.e., for different possible measurements. It is appropriate for a static regime or initialization of a dynamic sensor selection. This section proposes a dynamic sensor selection framework which considers measurements for sensor selection. First of all, let us define the dynamic sensor selection systematically as follows,

Definition 6. (Dynamic Sensor Selection) [132]: For a given model \mathcal{M} on the data, determine set \mathbb{S} such that the estimation error of the rest of sensors, \mathbb{S}^c , is minimized. The estimation is obtained based on the model, \mathcal{M} , and observed sensors, \mathbb{S} .

We assume the compressed sensing model (4.4) for power spectrum sensing. Let us denote the obtained spectrum power vector by the subset \mathbb{S} of sensors at time t as $\mathbf{x}_{\mathbb{S}}^t$. A proper selection of \mathbb{S} enables to predicting the power spectrum throughout the network's area.

In order to keep track of the network's dynamic, we propose to sample most of the nodes in a low rate mode; while some selected nodes should provide us with data sampled at a high rate enabling estimation of a high temporal resolution power spectrum map. In this framework, there is no completely switched off sensors, but we collect data from low-sampling rate sensors to dynamically select the sensors with high sampling rate. Therefore, we have two following types of sensors in our proposed framework,

1. *High-sampling-rate selected (active) sensors*: These are a small fraction of sensors selected by an underlying sensor selection mechanism in order to access real-time data and generate a dynamic power spectrum map. The active sensors report their sensing at rate $f_h = 1$ sample per time block.
2. *Low-sampling-rate sensors*: All sensors collect and report their data in a low-rate mode, resulting in less bandwidth and power consumption. The low-rate data enables us to validate the estimated power spectrum map. The low-sampling sensors report their sensing at rate $f_l = \frac{1}{n_l}$ sample/time. I.e., 1 sample per n_l time blocks is collected. It should be mentioned

that the measurements from low-sampling rate sensors will not contribute in estimating \mathbf{x} . They will be used to determine the reliability of estimation as we will discuss below.

The dynamic sensor selection aims to select some sensors as the active-mode set. The rest of sensors are marked as power efficient low sampling rate sensors. If the active set is selected properly, the rest of sensors can be predicted accurately by the assumed model and the active selected sensors. The ability of sensing is assumed same for all sensors and only the sensing time is different. However, different bandwidth for sensing can be considered in a more sophisticated framework which is out of scope of the present work. Selected sensors contain sufficient information enabling them to predict the rest of sensors by the assumed model on the spectrum (4.32). Low sampling rate data may cause obsolete information vulnerable to large deviation from the model. Moreover, changes in the dynamic of network also may cause large deviations between the model's estimation and the low sampling rate data. The following expression defines a new metric called *reliability* for sensor m at time t .

$$r_m^{(t)} = \frac{\exp(-\sigma(t - t_m))}{1 + |y_m - E^{(t)}(m, \mathbb{S})|^2}, \quad \forall m \in \{1, \dots, M\} \quad (4.27)$$

in which,

$$E^{(t)}(m, \mathbb{S}) = \mathbf{a}_m^T \mathbf{x}^{(t-1)}(\mathbb{S})$$

In (4.27), $\mathbf{x}^{(t-1)}(\mathbb{S})$ is the estimation of power propagation at time $t - 1$ based on collected data from active sensors indexed by \mathbb{S} . Moreover, $E^{(t)}(m, \mathbb{S})$ is the estimation of the measurement of m^{th} sensor at time t . σ is a temporal forgetting factor. t_m is the last time that sensor m is sampled and the corresponding measurement is y_m . The reliability of each sensor consists of two terms. The numerator indicates how fresh is our observation. Obsolete data results in unreliable observation. The denominator shows the power of model for estimation of unseen regions. Accurate estimation of the observation of sensor m using the active demonstrates that the sensor m has a reliable

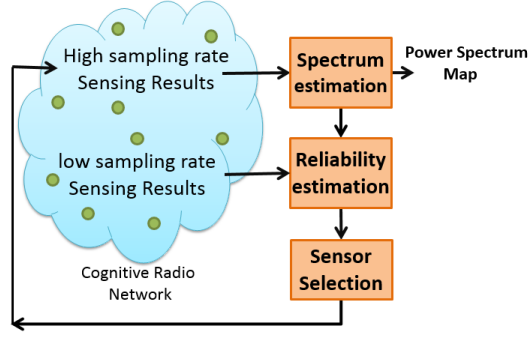


Figure 4.2: The main framework of the proposed reliability based sensor selection.

sensing. The proposed dynamic sensor selection framework is illustrated in Fig. 4.2. We propose to consider the reliability of sensors in the sensor selection procedure in order to determine a proper subset which is able to compensate large model's error for the low-rate sampled sensors. Mathematically speaking, the static E-optimal sensor selection algorithm is modified as follows,

$$\mathbb{S} = \underset{|\mathbb{S}| \leq K}{\operatorname{argmax}} \lambda_{\min}(A_{\mathbb{S}} A_{\mathbb{S}}^T) + \gamma \|u_{\mathbb{S}}\|_2^2, \quad (4.28)$$

in which, γ is the regularization parameter and $u_m = r_m^{-1}$ represents unreliability and $u_{\mathbb{S}}$ is the sub-vector of \mathbf{u} indexed by set \mathbb{S} . The superscript (t) is removed due to simplicity of notation. It means we are looking for unreliable sensors to select them for the next time slot in order to compensate the model's error.

Optimization and Complexity

In order to cast the dynamic sensor selection (4.28) in a tractable formulation, first let us rewrite the minimum eigenvalue as the following problem.

$$\lambda_{\min}(\mathbf{A}) = \min \|\mathbf{A}\mathbf{x}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_2 = 1. \quad (4.29)$$

Problem (4.28) can be written in the following form,

$$\begin{aligned} \mathbf{W}^{(t)} = \operatorname{argmax}_W \min_x & \|\mathbf{W}\mathbf{A}\mathbf{x}\|_2^2 + \gamma \|\mathbf{W}\mathbf{u}^{(t)}\|_2^2 \quad \text{s.t.} \\ \|\mathbf{x}\|_2 = 1, & W_{ij} \in \{0, 1\}, \|\mathbf{w}_k\|_0 = 1 \text{ and } \|\mathbf{w}^m\|_0 \leq 1. \end{aligned} \quad (4.30)$$

In which $\mathbf{W} \in \mathbb{R}^{K \times M}$ reduces the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ by some selected rows. \mathbf{w}_k represents the k^{th} row of \mathbf{W} and \mathbf{w}^m indicates the m^{th} column of \mathbf{W} . The last constraint $\|\mathbf{w}^m\|_0 \leq 1$ avoids repetitive selection of the same row (sensor). This problem implies eigenvalue optimization over combination of rows of \mathbf{A} that it is shown to be NP-hard [131]. Accordingly, we propose a greedy algorithm to solve (4.30).

Algorithm 9 shows the steps of our proposed greedy algorithm to solve the obtained optimization problem. This algorithm optimizes the reduction matrix row-by-row where the reliability of the non-selected sensors are being considered. Assume the algorithm aims to select a new sensor at the k^{th} iteration. Up to current iteration, $k - 1$ sensors already are selected. The algorithm evaluate the non-selected sensors one-by-one in order to find the sensor that maximizes the objective function. The objective function is a weighted summation of the minimum eigenvalue of the restricted set of rows (sensors) and their corresponding unreliability weights. To evaluate each sensor we need to compute the most dominant k eigen components which implies performing singular value decomposition (SVD). However, truncated SVD up to the k^{th} component will be sufficient. An online algorithm is proposed that observes the non-selected sensors with a low sampling rate as depicted in Fig. 4.2. In each sequence, the observed set of sensors is updated as well as their corresponding reliability weights. The first step to update the reliability is estimating the propagation using only

Algorithm 9 Reliable E-optimal Sensor Selection

Require: \mathbf{A} , S , K and r

Output: The selected set \mathbb{S} and reduction matrix \mathbf{W} .

Initialization: $\mathbf{W} = \mathbf{0} \in \mathbb{R}^{K \times M}$ and $\mathbb{S} = \emptyset$
for $k = 1, \dots, K$ (Optimization of the k^{th} row of \mathbf{W})
 for $\forall m \in \mathbb{S}^c$
 SVD: $\mathbf{A}(\mathbb{S} \cup m, :) = \mathbf{V}^T \mathbf{\Lambda} \mathbf{U}$
 $\mathbf{x}^* = \mathbf{U}(:, k)$
 $p(m) = \|\mathbf{A}\mathbf{x}^*\|_2^2 + \gamma u(m)$
 end
 $s_k = \text{argmax}_m p(m)$
 $\mathbb{S} = \mathbb{S} \cup s_k$ and $\mathbf{W}_{k, s_k} = 1$
end for

the current active sensors. To this aim the following problem must be solved.

$$\mathbf{x}^{(t)}(\mathbb{S}) = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{W}^{(t)}(\mathbf{y} - \mathbf{A}\mathbf{x})\|_2^2 + \lambda_{LASSO} \|\mathbf{x}\|_1. \quad (4.31)$$

Here λ_{LASSO} regularizes sparsity and \mathbf{W} indicates the reduction matrix to the selected set \mathbb{S} . Those sensors whose measurements are matched with the estimated power density map are marked as reliable. A consistent definition is proposed in (4.27) which considers the deviation of actual measurements from the estimation of the model as a metric for reliability. The subscript t is removed in Algorithm 2 for simplification. Algorithm 10 shows the overall process of spectrum sensing using the selected sensors.

The bottleneck of complexity order of Algorithm 9 at the k^{th} iteration is performing a truncated singular value decomposition to obtain the first k eigen components. Thus, the complexity of the algorithm in the k^{th} iteration will be $O(kMN^2)$ [134]. Therefore, selection of K sensors implies complexity order of $O(K^2MN^2)$.

Algorithm 10 Spectrum Sensing using Dynamic Sensor Selection

Require: \mathbf{A} , S , K , λ , f_l and λ_{LASSO} .

Output: Power spectrum for each time $\mathbf{x}^{(t)}$.

Initialization: \mathbb{S} = Output of Algorithm 1 and $\mathbf{x}(\mathbb{S})$ = Result of Problem (4.31)

- 2: for a new time block (t)
 sample $M \times f_l$ sensors
 - 4: Update $t_m = t$ for the sensed sensors
 Update reliability using (4.27)
 - 6: $\mathbb{S}^{(t)}$ = Output of Algorithm 9
 $\mathbf{x}^{(t)}(\mathbb{S}^{(t)})$ = Result of Problem (4.31)
 - 8: end for
-

Experimental Results

Our proposed schemes are evaluated in three cases including sensor selection in cognitive radio networks (CRNs), data selection for supervised learning, and performance evaluation using synthesized data. The underlying model is $\mathbf{y} = \mathbf{A}\mathbf{x}$. Matrix \mathbf{A} in CRNs is an array of channel gains from different locations of the network to locations of sensors. In the case of supervised learning, \mathbf{A} is the collection of training data. \mathbf{x} and \mathbf{y} are specified for a test data. However, it is desired that the trained system works for any test data. In the first case, we are estimating a specific \mathbf{x} which corresponds to a specific \mathbf{y} . While for supervised learning it is desired that the selected data constructs a well-conditioned inverse problem that is averagely appropriate for any test data. Thus, we exploit Algorithm 5 only for the first application where we access to the actual measurements of sensors in an online manner.

Sensor Selection in CRNs

The simulations are performed for collaborative spectrum sensing. Our goal is to estimate vector \mathbf{x} that indicates transmitted spectrum power at some candidate points. We assume a network setup the same as that of [138]. Consider N_s transmitters and M receivers in an area. The receivers

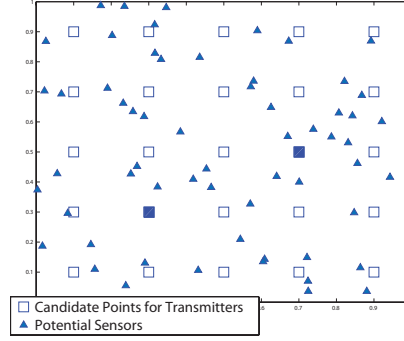


Figure 4.3: An example setup with 25 candidate points as transmitters.

receive a superposition of the transmitter signals. Figure 4.3 shows a setup consist of $N_s = 25$ potential transmitters and 2 active points. The received signals are contaminated by channel gain and additive noise, represented by,

$$\mathbf{y}_m = \mathbf{A}_m \mathbf{x} + \sigma_m^2 \mathbf{1}, \quad \forall m = 1 \dots M, \quad (4.32)$$

where, $\mathbf{1} \in \mathbb{R}^n$, $\mathbf{y}_m \in \mathbb{R}^n$ in which n is the number of frequency samples in each time slot. Moreover, \mathbf{A}_m^T contains the corresponding channel gains and σ_m^2 represents noise power at the m^{th} receiver. The following problem aims to estimate \mathbf{x}

$$\hat{\mathbf{x}} = \underset{\mathbf{x}, \sigma}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{A}\mathbf{x} - \sigma \otimes \mathbf{1}\|_2^2 + \gamma \|\mathbf{x}\|_1, \quad (4.33)$$

in which $\sigma \in \mathbb{R}^M$ indicates the noise level of each sensor. \mathbf{y} and \mathbf{A} are concatenation of \mathbf{y}_m and \mathbf{A}_m respectively and \otimes denotes kronecker multiplication. Each entry of \mathbf{x} determines the contribution of the s^{th} source on the sensed data. Due to scarce presence of active transmitters and their narrow band communication, $\|\mathbf{x}\|_1$ is exploited which encourages sparsity.

Suppose we have potentially 300 sensors and they are estimating an $\mathbf{x} \in \mathbb{R}^{36}$ that has only 5 active transmitters. Figure 4.20 shows the performance of different algorithms versus the number of

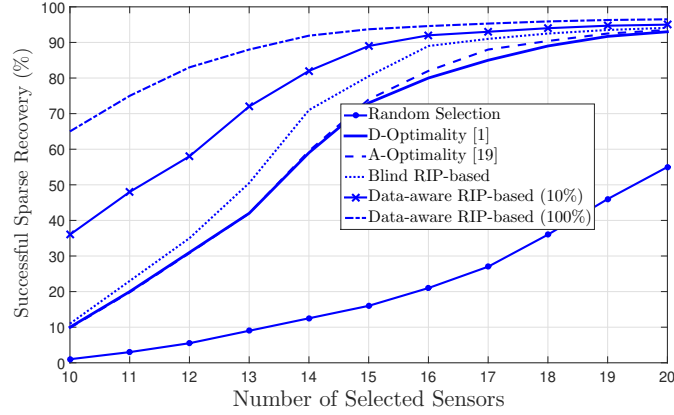


Figure 4.4: Performance of different sensor selection algorithms in terms of number of selected sensors

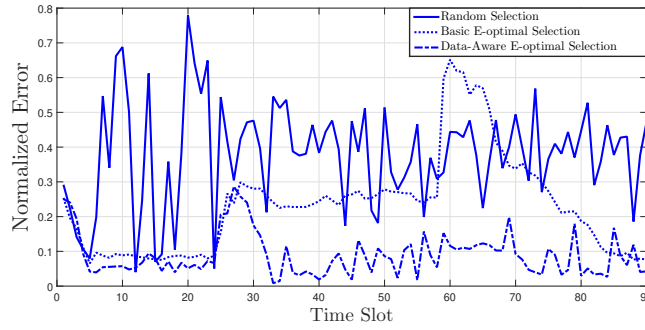


Figure 4.5: Performance of the selection algorithm in presence of 0dB AWGN.

selected sensors. Successful recovery is defined as true estimation of the support of sparse vector using the measurements.

For the first experiment, Problem (4.33) is solved 200 times by different selected sensor sets for each algorithm. Additive noise is not considered and the iterative re-weighted least square algorithm is employed to obtain the sparse solution [139]. As it can be seen in Figure 4.20, among the blind methods, sensor selection using RIP coefficient δ_{2S} has the best performance. In the case of known data in a fusion center, the information of sensed data has a great effect on the centralized

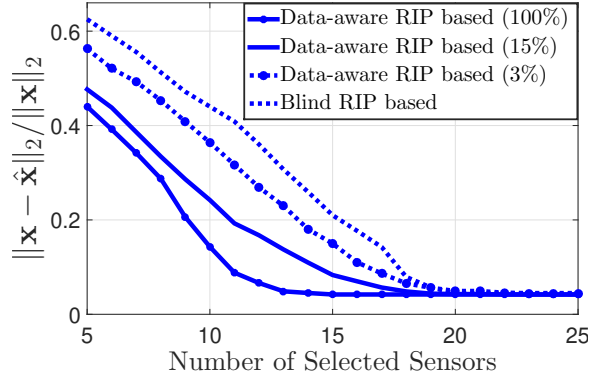


Figure 4.6: Performance of blind and data-aware RIP based sensor selection algorithms in terms of number of selected sensors.

estimation. The data-aware algorithm observes some sensors in an online manner.⁴

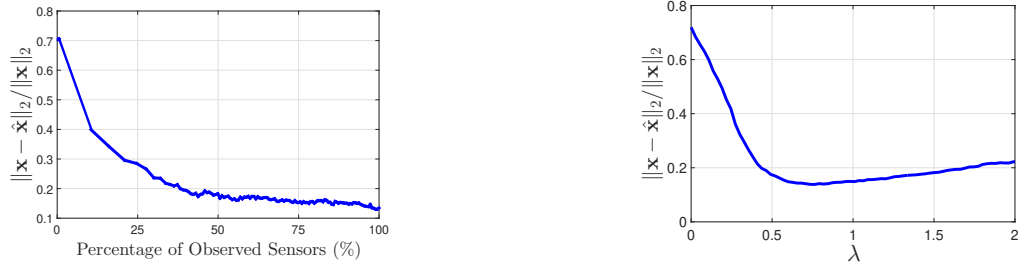
Fig. 4.5 shows performance of the proposed selection algorithm in a dynamic system where status of the network is changed at time slot 25 and 60. Switching of any propagation point causes a status change. This simulation is performed in presence of 0dB AWGN in addition to 6 tab multi-path fading. As it can be seen the blind algorithm performs better than random selection, however, the selected sensors are fixed and independent of the dynamic of system.

Fig. 4.21 exhibits the effect of involving sensors measurements in the data-aware sensor selection algorithm. Random sensing of only 3% of data (9 sensors within 300 sensors) prior to sensor selection makes an improvement in normalized estimation error; similarly, usage of 15% of data significantly improves the performance to be close to the centralized sensor selection which access to 100% of the data. The normalized error is defined as follows as the criterion for performance,

$$\text{normalized error} = \frac{\|\mathbf{x}^* - \mathbf{x}(\mathbb{O})\|_2}{\|\mathbf{x}^*\|_2}.$$

In which, \mathbf{x}^* is the ground truth solution.

⁴The initial sensors can be determined by our blind RIP-based sensor selection and then in each time slot a new sensor will be observed.



(a) Performance of data-aware algorithm versus the (b) MSE error versus different values of λ for the percentage of observed measurements, λ is assumed data-aware algorithm where 100% of measurements equal to 0.7.

Figure 4.7: Performance of our Data-aware algorithm.

Another simulation is performed to select 20 sensors out of 200 ones to determine the power spectrum in 36 candidate points while 10 of them are active. Figure 4.7(a) shows the performance of our proposed data-aware method in terms of MSE of the sparse vector estimation while λ is assumed 0.7. The performance improves as the number of observed sensors increases. The performance obtained by observation of 50% of data (100 sensors) is about that of all the sensors because of the redundancy among the sensors. It can be seen in Figure 4.7(b) that the error of estimation is significantly decreased by setting $\lambda = 0.7$. However, an efficient value of γ depends on the problem setup and should be tuned. Setting $\gamma = 0$ is equivalent to the static E-optimal sensor selection. Simulation shows the proposed reliable sensor selection performs better than the static sensor selection for a relatively wide range of γ , i.e., the problem is not very sensitive to well-tuning of this parameter.

Fig. 4.8 shows the power spectrum of a network in an area. We have potentially 200 sensors, however we are allowed to use only 8 sensors for collaborative spectrum estimation. The selected sensors using the blind and data-aware RIP based are marked in this figure. As it can be seen, the selected sensors of the blind RIP based are spread in the area while the selected sensors by the data-aware algorithm have a tendency to move toward the more eventful areas of the network.

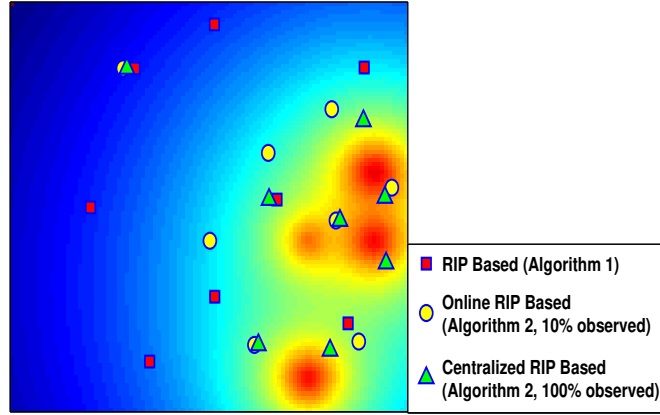


Figure 4.8: The true spectrum in the area of interest along the selected sensors obtained by 3 methods in spatial domain

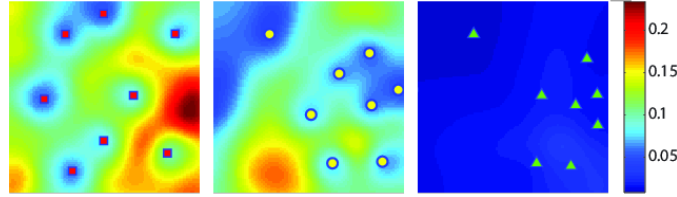


Figure 4.9: The error of estimated spectrum in the area of interest corresponding to Fig. 4.8. (Left) RIP based, Algorithm 1. (Middle) Online RIP based, Algorithm 2 while only 5% of sensors are sensed. (Right) Centralized RIP based, Algorithm 2 while all the sensors are sensed. λ is assumed 0.7

Figure 4.22 shows the error of estimated spectrum using different selected sensors in the setup of Fig. 4.8. To this end, first, the spectrum is estimated in all of sensors and the error is obtained by Euclidean distance of the estimated spectrum and the actual measurements, then a weighted averaging is performed to interpolate the spectrum error in every point.

Data Selection for Supervised Learning

In this section, the applicability of the proposed selection technique in feature and data selection is studied. This is a challenging problem in computer vision and machine learning [140].

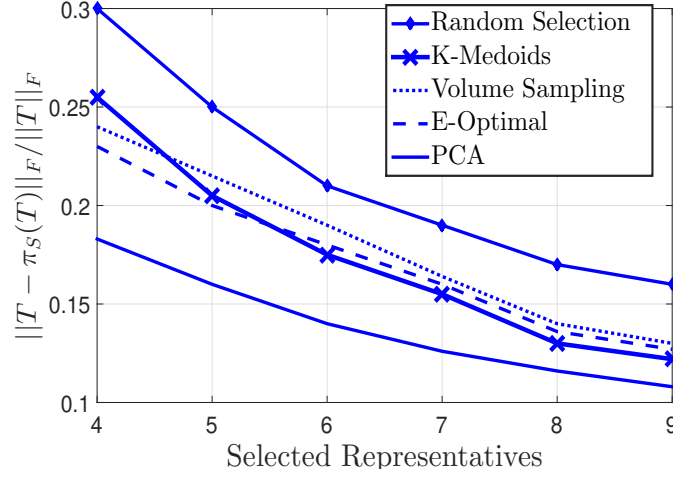


Figure 4.10: The projection error of the training data into the subspace spanned by the selected rows.

We evaluate the performance of our method as well as other algorithms for finding appropriate representatives for classification. The training data set is reduced to only some selected data for each class. The classifier is then trained solely by the reduced set. We assume that if the representative data are informative enough about the initial data set, the classification performance should be close to the comprehensive classifier. We compare our proposed algorithm with some standard methods for finding representatives. These methods are Kmedoids [141], volume sampling [126], and a simple random selection. Some basic classifiers are utilized for learning and evaluating the test data including nearest neighbor (NN), nearest subspace (NS) [142], sparse representation based classifier (SRC) [143], and linear support vector machine (SVM) [144].

Extended Yale-B face images dataset [145] is used to perform the simulations. The dataset consists of 38 subjects in which there exist 56 images for each subject. Data is split into two groups: train set and test set that contain 51 and 5 images, respectively. The selection algorithms aim to pick up a few training images among all 51 ones to train a general classifier which is able to identify the test images.

Fig. 4.10 shows the normalized error of the projection of training data on the subspace spanned by the representatives. In this figure matrix T is the collection of the training data which representatives are selected from them. It is obvious that PCA indicates the best normalized error ⁵. I.e., it can be interpreted as a lower bound for the projection error on any low-dimensional space. However, we aim to indicate the subspace only using few images of the training data set. The performance of random selection, K-medoids, D-optimal, and our suggested E-optimal selections are shown in this figure.

The projection error of test data is depicted in Fig. 4.11. In this figure matrix T is the collection of the test data which are not seen for selection procedure. Although the error of PCA representatives for training data is much less than the other methods due to over-learning of the bases, in the case of test data the performance of our suggested selection is approximately the same as that of PCA representatives. This means, we could span a generalized subspace by only using few selected images that are able to cover the desired signal space as well as PCA method that uses all of the training data.

Fig. 4.12 shows 40 images from the third subject of Extended Yale-B data set. As an example we are to select 6 images using K-medoids and our suggested algorithm. The results are shown in Fig. 4.13. The selected set of images using K-medoids do not contain the shadowing effect from the front side while our selection capture from different point of views.

The effect of data partitioning using the successive E-optimal selection on the performance of selection is studied for a larger data set. MISNT data set is used which contains 60,000 sample images of handwritten digits [146]. Two criteria are considered, the first one is recognition rate using the learned classifier by reduced data and the second criterion is running time for data selec-

⁵According to the definition of PCA, it spans the best low-rank subspace that minimizes the normalized error defined in Fig. 4.10 for a set of training data.

Table 4.2: Accuracy of different classifiers using partial data for learning of Extended Yale-B dataset with 5 representatives.

	NN	NS	SRC	SVM
Random	26.8%	45.3%	72.0%	55.7%
Kmedoids	39.0%	61.1%	82.6%	68.2%
Volume sampling	76.3%	71.6%	88.9%	85.3%
E-optimal	77.9%	82.6%	94.2%	90.0%
All Data	81.4%	95.8%	97.1%	98.7%

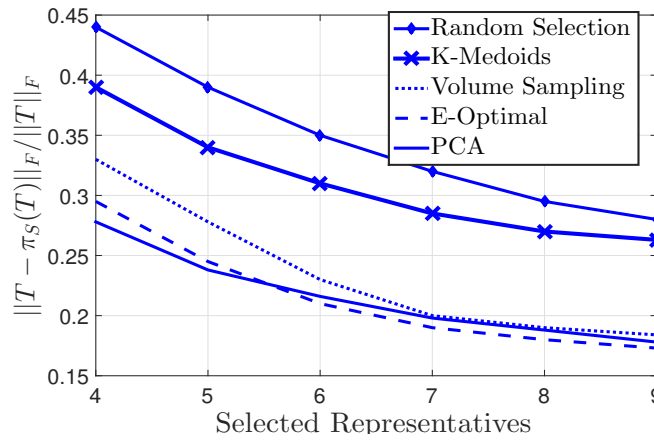


Figure 4.11: The projection error of the test data into the subspace spanned by the selected rows.

tion and data classification. Reducing the number of training data may decrease the performance of a classifier. A proper selection aims to preserve the recognition rate about the one using full data. On the other hand, reduced data make the training algorithm fast. Exploiting full data needs no process for selection but the training process needs a high amount of computations.

The basic E-optimal criterion is vulnerable to outlier data. It aims to select the most distinguished samples. However, unusual samples are probably different from each other and they satisfy the E-optimal criterion. The proposed two-phase algorithm first selects some candidates for final selection using E-optimal criterion and in the second phase the final selection reduces candidate samples to exact K selection. Fig. 4.14 shows the effect of two-phase algorithm on selection from



Figure 4.12: Training data corresponding to the third subject of Extended Yale-B data set. This data set contains different angles of shadowing for each subject.

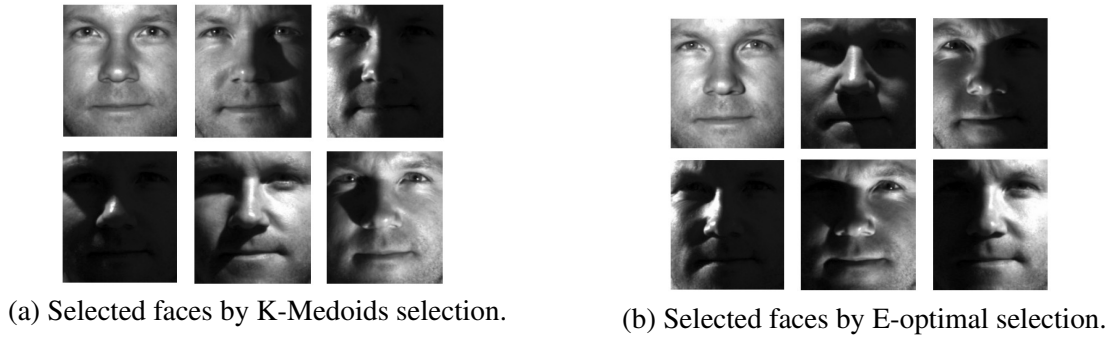


Figure 4.13: Comparison of the proposed E-optimal representatives versus K-medoids selection.

5842 samples of digit 4. The selected samples by E-optimal criterion are exceptional hand-written characters for digit 4. While, the two-phase algorithm selects visually proper representative for this class. Quantitative measures also will be demonstrated.

Sparse subspace classifier is learned by only few selected data. Four criteria are investigated for selection. D-optimal, the proposed E-optimal, K-medoids and the proposed two-phase algorithm are utilized for selection. D-optimal and E-optimal are vulnerable to outlier data as depicted in Fig. 4.14 (a). The k-medoids algorithm performs better than greedy algorithms for selection as it finds some points that data are concentrated around them. However, k-medoids algorithm is not tractable for real-time processing of big data. Our suggested two-phase algorithm outperforms K-



(a) E-optimal criterion on the whole data. (b) Two phase distributed selection based on E-optimality.

Figure 4.14: 12 selected images of digit 4 from 5842 images.

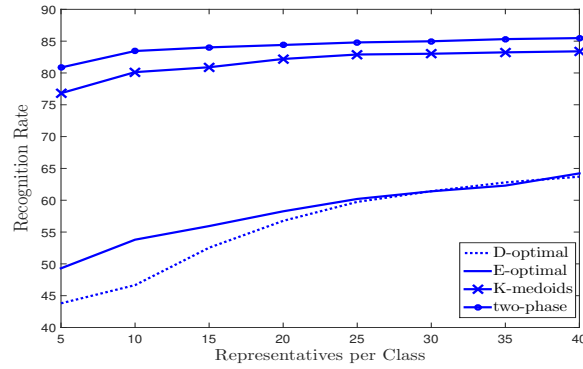


Figure 4.15: Performance of nearest subspace classifier learned by few data from each class.

medoids in terms successful classification rate. In addition to better representatives, our two-phase selection performs much faster than K-medoids algorithm. The running time of algorithms are shown in Fig. 4.16. Reducing the number of training signals saves a huge computation burden for training the classifier. In this figure algorithms are performed using an Intel Xeon CPU 3.7 Ghz and 8 GB RAM. A simple one nearest neighbor classifier needs 784 seconds to classify 5000 test images. While by selecting data it decreases to 2.83 seconds.

Deep learning achieves the best results for classification of MNIST data set. In order to compare the the effect of data selection on the state of the art method of classification, a deep neural network is learned with the selected data. MLP network and Capsules network [147] are employed to perform classification. Table 4.3 summarizes the accuracy of learned classifiers. The MLP

network has three layers and the hidden layer contains 1000 neurons. As it can be seen selected training set improves the classification rate. For example, the learned network using 2,000 random images is working worse than the network which is learned by 1,000 selected images as the training set. However, Capsule net which exhibits one of the best performances for MNIST data set is less sensitive to the input training data and the improvement is less than that of MLP. Table 4.4 shows processing time for selection from 60,000 images for K-medoids algorithm and our proposed two-phase algorithm. Please note that the proposed algorithm can be implemented parallel which reduces the running time significantly. However, the centralized algorithm is simulated. The effect of data reduction on the speed of learning a deep network is presented in Table 4.5. The running time for one epoch is reported. MLP needs 20 epochs for convergence and it takes 500 epochs for CapsNet to reach the best performance. Thus, running time of MLP for whole data is about 30 seconds and for CapsNet is about 266 minutes. While, using only 1000 samples the running time for MLP decreases to only 1 second and for CapsNet it takes less than 3 minutes. Deep learning simulations are performed on Chainer framework [148] using 1 GPU of Nvidia TitanX and 12 GB RAM.

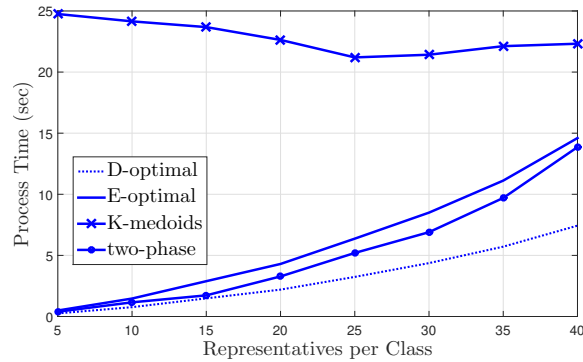


Figure 4.16: Running time of selecting few data from each class.

Table 4.3: Performance of selection algorithms in terms percentage of classification rate using a deep neural network learned by partial data. The original data set contains 60,000 training images. (Left) K-medoids, (Middle) two-phase. (Right) Random Selection. Each classifier is learned by only 200, 500, 1000 and 2000 training data out of 60,000. The performance of MLP using all 60,000 samples is 98.25 and it is 99.66 for the CapsNet architecture.

	200			500			1,000			2,000		
MLP	87.45	88.26	80.84	89.81	91.12	88.52	92.13	93.11	91.03	93.79	94.50	92.91
CapsNet	84.38	81.85	78.61	92.76	93.10	91.87	96.11	96.62	95.72	97.91	97.69	97.59

Table 4.4: Running time (seconds) of data selection corresponding to Table 4.3.

	200	500	1000	2000
K-medoids	193.4	218.4	433.2	1328
two-phase	31.61	160.8	191.3	280.6

Synthesized Data

In the first setup, data are generated on 10 subspaces within \mathbb{R}^{500} . Each subspace is ranked 4 and there are 1000 samples per each subspace. The goal is to select some data that spans all the rest of data. Obviously, the union of data is ranked at most 40. Thus, 40 data can be sufficient to build a subspace that reconstructs all data. Fig. 4.17 shows normalized projection error of data on the span of selected data. D-optimal and E-optimal criteria are able to select 40 samples that exactly recover all data. However, E-optimal solution achieves lower errors when the rank of data is not estimated accurately.

Table 4.5: Running time (seconds) of neural network learning corresponding to Table 4.3. Running time per epoch is reported.

	200	500	1000	2000	60,000
MLP	0.029	0.04	0.051	0.078	1.55
CapsNet	0.073	0.18	0.39	0.88	32.18

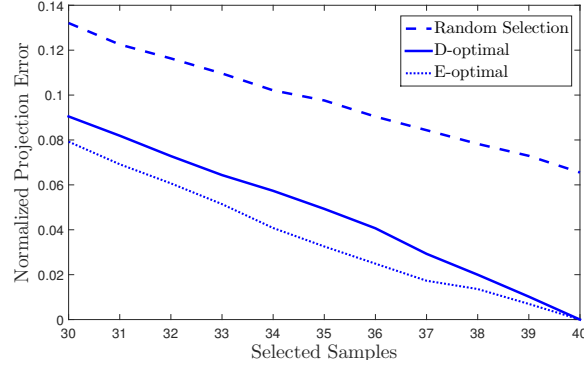


Figure 4.17: Error of subspace identification using selection.

Matrix \mathbf{A} is generated using the union of subspaces model. In order to make the synthetic data more realistic, we generalize Assumption 1 for simulations by considering non-uniform characteristics of subspaces. The rank and the number of members of each subspace are also considered random and the generated data is contaminated by noise. Moreover, the span of subspaces may be overlapped. Matrix \mathbf{A} is modeled by $\mathbf{X}\mathbf{D} + \mathbf{N}$. Rows of \mathbf{D} spans the synthesized row space and coefficients \mathbf{X} specify contribution of rows of \mathbf{D} for constructing each row of \mathbf{A} . Matrix \mathbf{N} represents the AWGN noise.

Fig. 4.18 compares the performance of two proposed distributed schemes with the lower bound indicated by PCA and the upper bound that corresponds to the random selection. The dimension of data is 100 and 50,000 data are generated. SNR is +20dB and $K = 10$ data are selected. $C = 1$ points to the non-distributed implementation. Random partitioning deteriorates the performance significantly. However, it is still much better than purely random selection. The performance of random selection is evaluated by averaging of performances of 100 different randomly selected rows. The subspace-based partitioning preserves the performance using the proposed pre-processing. K-subspaces algorithm is exploited for subspace identification [137]. The computational burden of the proposed distributed algorithm is demonstrated in Fig. 4.19 in terms of running time. As the number of distributed nodes increases, each node performs less computations.

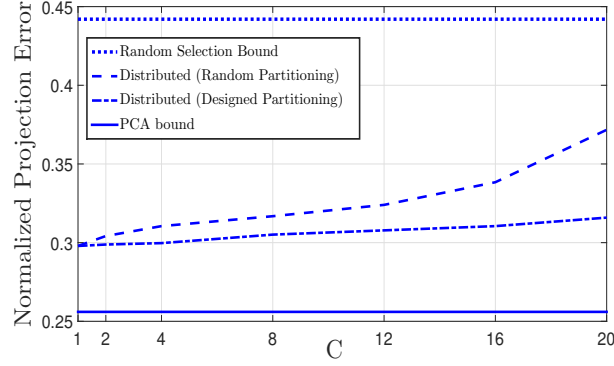


Figure 4.18: Performance of the distributed selection algorithm in terms of number of distributed nodes.

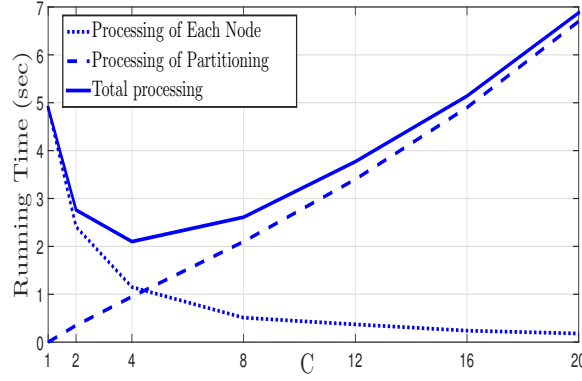


Figure 4.19: Complexity of the distributed selection algorithm in terms of number of distributed nodes.

However, the pre-processing for partitioning implies more complex operations. In this simulation both distributed selection and partitioning are performed by a same processor. In Fig. 4.19 the bottleneck of the designed partitioning scheme comes from complexity of partitioning. While, a more powerful processor can be employed in a fusion center to perform the needed pre-processing operations. Moreover, as only one subspace is needed in each succession, faster subspace identification algorithms can be employed rather than K -subspaces.

In this chapter the problem of sensor selection is considered and its relation to existing work on matrix subset selection is elaborated. We developed a new subset selection method as an extension

for the well-known volume sampling. Our criteria is based on E-optimality which is in favor of compressive sensing theory. Moreover the E-optimal criterion is extended to RIP-based sensor selection. Selection is an enabling step for efficient processing of a large amount of data, however for many cases selection from large data also is challenging. To this aim, successive and distributed implementation of the proposed algorithm are developed. Experimental results indicate the performance of our suggested sensor selection algorithm in cognitive radio networks' spectrum sensing as well as supervised learning with partial selected data.

Dynamic Sensor Selection

The simulations are performed for collaborative spectrum sensing. The setup for generating data are employed from [138]. Our goal is to estimate vector \mathbf{x} that indicates transmitted spectrum power at some candidate points.

For the first simulation suppose we have potentially 300 sensors and they are estimating an $\mathbf{x} \in \mathbb{R}^{36}$ that has only 5 active transmitters. The location of sensors are derived from a uniform distribution and the active transmitters are selected randomly and the results are averaged for 200 different realizations. The following linear measurements are sensed by sensors $m = 1 \dots M$,

$$y_m = \mathbf{a}_m^T \mathbf{x} + \nu_m,$$

where, ν_m indicates additive white Gaussian noise. a_{ms} shows the s^{th} entry of \mathbf{a}_m is the channel gain between the m^{th} sensor and the s^{th} potential source. The channel gain between two points is assumed by one over squared distance of two points. Since, the ability of sensors is considered the same over spectrum, thus the simulations are performed for a single spectrum band. The same procedure can be performed for multi-band spectrum regime independently. Figure 4.20 shows the performance of different static algorithms versus the number of selected sensors. Static refers to measurement-independent methods. In this experiment the SNR is set to +20dB. Successful

recovery is defined as true estimation of the support of sparse vector using the measurements. Problem (4.31) is solved 200 for each algorithm. The Sparse solution is obtained using the iterative re-weighted least square algorithm [139]. As it can be seen in Fig. 4.20, E-optimal based sensor selection has the best performance.

Fig. 4.21 exhibits the effect of involving reliability on the static sensor selection. Suppose there are 300 potential sensors and the low-sampling rate is set equal to $\frac{1}{30}$. It means in each time block 10 new measurements contribute to construct the reliability weights (4.27). Observation of new measurements of one time block makes an improvement in normalized estimation error; similarly, usage of 5 time blocks significantly improves the performance to be close to the estimation after 30 time blocks in which all the sensors are observed. The forgetting factor is set to 0 as the state of network is not changed during observation of 30 time blocks. Thus, aggregating the measurements without the forgetting factor is optimum. The normalized error, $\|\mathbf{x}^* - \mathbf{x}(\mathbb{S})\|_2 / \|\mathbf{x}^*\|_2$, is defined as the criterion for performance, where, \mathbf{x}^* is the ground truth solution.

Fig. 4.22 visualizes the error of spectrum sensing in the area of network for the setup of Fig. 4.21. We are to choose 8 sensors.

Fig. 7 shows that the error of estimation is significantly decreased by setting $\gamma = 0.7$ for the setup

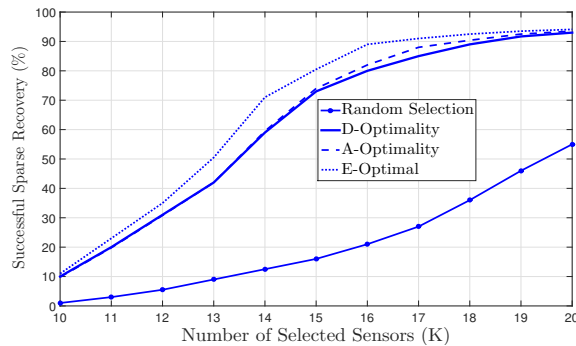


Figure 4.20: Performance of different static sensor selection algorithms in terms of number of selected sensors.

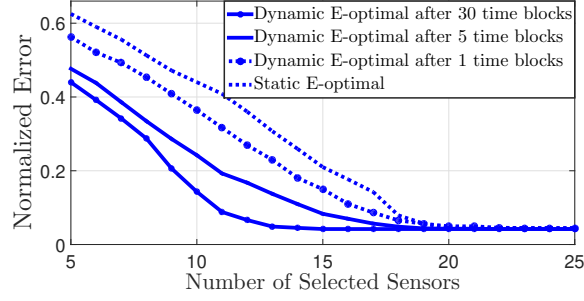


Figure 4.21: Performance of static and dynamic E-optimal-based sensor selection algorithms vs. the number of selected sensors.

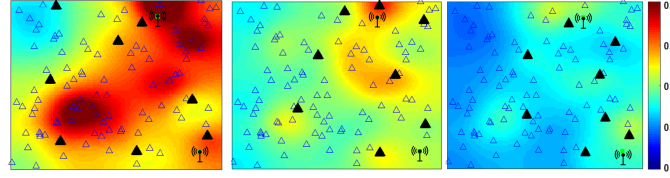


Figure 4.22: The error of estimated spectrum in the area of interest. (Left) E-optimal, Algorithm 1. (Middle) Reliable E-optimal, Algorithm 2 after sensing in one time block. (Right) Reliable E-optimal, Algorithm 2 while all the sensors are sensed after 30 time blocks. γ is assumed 0.7

of Fig. 4.21. However, an efficient value of γ depends on the problem setup and should be tuned. Setting $\gamma = 0$ is equivalent to the static E-optimal sensor selection. Simulation shows the proposed reliable sensor selection performs better than the static sensor selection for a relatively wide range of γ , i.e., the problem is not very sensitive to well-tuning of this parameter.

In addition to power spectrum map, the proposed framework is able to generate a new network profile which can provides us trustworthy of the estimated spectrum for each point of the network. We

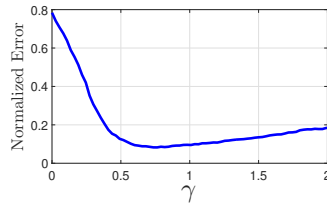


Figure 4.23: MSE error versus different values of γ .

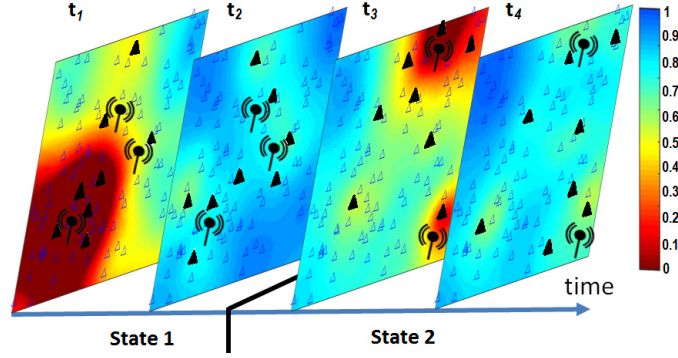
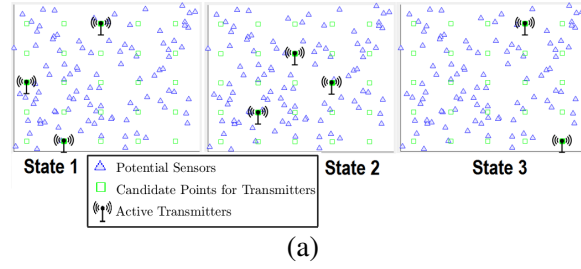


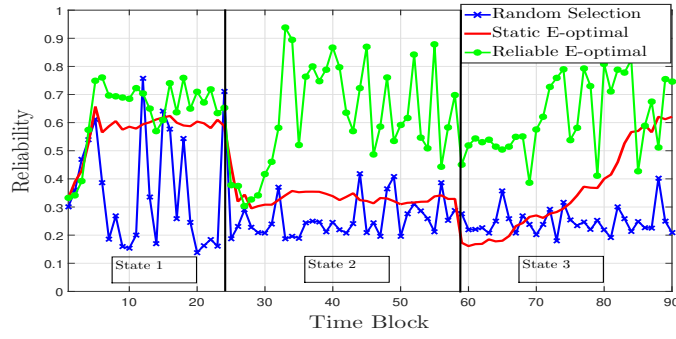
Figure 4.24: Reliability maps of 4 time blocks illustrate how the proposed framework evolves in time in order to select adapted sensors to the dynamic of network after state transition. Sensors within unreliable (red) areas have more chance of selection.

call this side output *reliability map*. Interpolation of the estimated reliability of sensors throughout the network's area, generates the reliability map. Fig. 4.24 visualizes the temporal effect of dynamic sensor selection using the reliability map. Unreliable areas are indicated by red and blue areas represent reliable estimation of spectrum. Reliable sensor selection aims to compensate unreliability by considering more chance for red regions. In the next time slot the error for those regions are compensated. In this figure, each state of the network corresponds to a specific set of active PUs.

Fig. 4.25a shows the location of active PUs for a dynamic network with 3 states. There are 90 time blocks and the state of network is changed in blocks 24 and 59. The forgetting factor is set to $0.1/(\Delta T)$ in which ΔT is the time difference of two consecutive time blocks. Fig. 4.25b and Fig. 4.26 show the performance of sensor selection in terms of average network reliability and the spurious error of spectrum sensing which is defined by $\|\hat{x}_{\text{spurious}}\|_1 = \sum_{i \notin \mathbf{x}^* \text{ support}} |x(\mathbb{S})_i|$. As it can be seen, the reliability is increased and the undesired power propagation is decreased by exploiting the dynamic framework, especially for the second state.



(a)



(b)

Figure 4.25: (a) A dynamic network with 3 states for the location of active PUs. The shaded blue squares represent active PUs. (b) The effect of reliable sensor selection for compensation of the model error in the reliable sensor selection procedure.

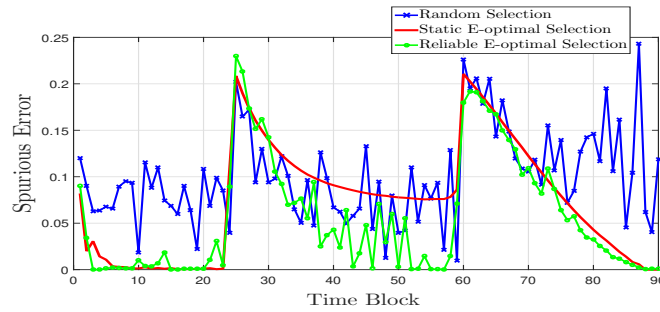


Figure 4.26: The effect of reliable sensor selection for compensation of the model error in the reliable sensor selection procedure.

Conclusion

The problem of sensor selection is considered and its relation to existing work on matrix subset selection is elaborated. We developed a new subset selection method as an extension of the well-

known volume sampling. Our criteria is based on E-optimality, which is in favor of compressive sensing theory. We extended the static E-optimal sensor selection to a dynamic sensor selection method that exploits the measurements in an online manner. The experimental results indicate the efficiency of our suggested sensor selection algorithm in cognitive radio networks' spectrum sensing.

CHAPTER 5: SPECTRUM PURSUIT: DATA REDUCTION BASED ON PRESERVING SPECTRAL STRUCTURE

Column subset selection problem (CSSP) aims to find a subset of a dataset such that the linear combination of selected samples approximates the entire dataset. This chapter presents an efficient solver for CSSP, referred to as Spectrum Pursuit (SP), in which samples are selected to pursue the most significant spectral components of the entire data. SP has a linear computational complexity w.r.t. number of original samples. This is a significant improvement over state-of-the-art techniques with cubical complexity. In addition to its simplicity, SP is an accurate solver for CSSP and is parameter-free that makes it a universal and efficient selection algorithm for many applications. At each iteration of SP, one selected sample is updated by capturing maximum information from the structure of the data based on spectral decomposition. We show that SP is an optimal selector for sampling from linear subspaces. Moreover, a tight upper bound for projection error on the span of selected data is presented. Furthermore, the superiority of SP is demonstrated on learning based on representatives for ImageNet dataset; training a generative adversarial network (GAN) to generate multi-view images on CMU Multi-PIE dataset, and fast subspace clustering on MNIST dataset.

With the ever-increasing proliferation of sensing devices, a massive amount of data is available for machine learning (ML) purposes. However, processing/labeling/communication of a large number of input data has remained challenging. Therefore, novel ML methods that make the best use of a significantly less amount of data are of great interest. For example, active learning (AL) [14] aims at addressing this problem by training a model using a small number of labeled data, evaluating the trained model, and then querying the labels of selected representatives, which are used later for training a new model. In this context, preserving the underlying structure of data succinctly by

representatives is an essential concern.

Dimension reduction techniques and clustering-based approaches aim to extract a concise representation of data. However, representatives or exemplars obtained by such methods are often not easy to interpret. Furthermore, obtaining each representative implies processing all or a large portion of data. Thus, it is desired to optimally select the representatives from data samples. There are some clustering approaches that select the representatives from data such as the k-medoids clustering [13]. These clustering methods assign each data sample to only one prototype which is the cluster center. However, in the case of more structured data only one prototype from data does not contain sufficient information to capture the underlying structure of the whole cluster. Randomly selecting K out of M data, while computationally simple, is inefficient in many cases, since non-informative or redundant instances may be among the selected ones. On the other hand, the optimal selection of data for a specific task implies solving an NP-hard problem [18]. For example, finding an optimal subset of K data samples from M to be employed in training a Deep Learning (DL) network with the best performance requires $\binom{M}{K}$ number of trial and errors, which is not tractable. It is essential to define a versatile objective function and to develop a method that efficiently selects the K samples that optimize the objective function. Let us assume the M data samples are organized as the columns of a matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$. The following is a general purpose cost function for subset selection, known as *column subset selection problem* (CSSP), which is an open problem [19]:

$$\mathbb{S}^* = \underset{|\mathbb{S}| \leq K}{\operatorname{argmin}} \|\mathbf{A} - \pi_{\mathbb{S}}(\mathbf{A})\|_F^2, \quad (5.1)$$

where $\pi_{\mathbb{S}}$ is the linear projection operator on the span of K columns of \mathbf{A} indicated by set \mathbb{S} . This problem has been shown to be NP hard [18, 20]. Moreover, the cost function is not sub-modular [21] and greedy algorithms are not efficient to tackle Problem (5.1). Computer scientists and

mathematicians during the last 30 years have proposed many tractable selection algorithms that guarantee an upper bound for the projection error $\|\mathbf{A} - \pi_{\mathbb{S}}(\mathbf{A})\|_F^2$. These works include algorithms based on QR decomposition of matrix \mathbf{A} with column pivoting (QRCP) [22, 23, 24]; methods based on volume sampling (VS) [25, 26, 27] and matrix subset selection algorithms [19, 28, 29]. However, the guaranteed upper bounds are very loose and the corresponding selection results are far from the actual minimizer of CSSP in practice. Interested readers are referred to [30, 28] and Sec. 2.1 in [31] for detailed discussions. For example, in VS it is shown that the projection error on the span of K selected samples is guaranteed to be less than $K + 1$ times of the projection error on the span of the K first left singular vectors; which is too loose for a large K . Recently, it was shown that VS performs even worse than random selection in some scenarios [32]. Moreover, some efforts have been made using convex relaxation and regularization. Fine tuning of these methods is not straightforward [6, 4, 33]. Moreover their cubical complexity is an obstacle to employ these methods for diverse applications.

Recently, we have proposed a low-complexity approach to solve CSSP, referred to as iterative projection and matching (IPM) [3]. IPM is a greedy algorithm that selects K consecutive and locally optimal samples, yet without the option of revisiting the previous selections and escaping from local optima.

In this chapter, we propose a solver for CSSP, referred to as *Spectrum Pursuit (SP)* and experiments show that SP provides an accurate solution for CSSP. Fig. 5.1 shows the concept behind the proposed algorithm for data selection. Assume we are not restricted to select representatives from data samples and we are allowed to generate pseudo-data and select them as representatives. In this scenario, the best K representatives are the first K spectral components of data according to the definition of singular value decomposition (SVD). However, the spectral components of data are not among data members. Our proposed algorithm aims to find K data samples such that their span is close to that of the first K spectrum of data obtained from the ensemble of data. Fig.

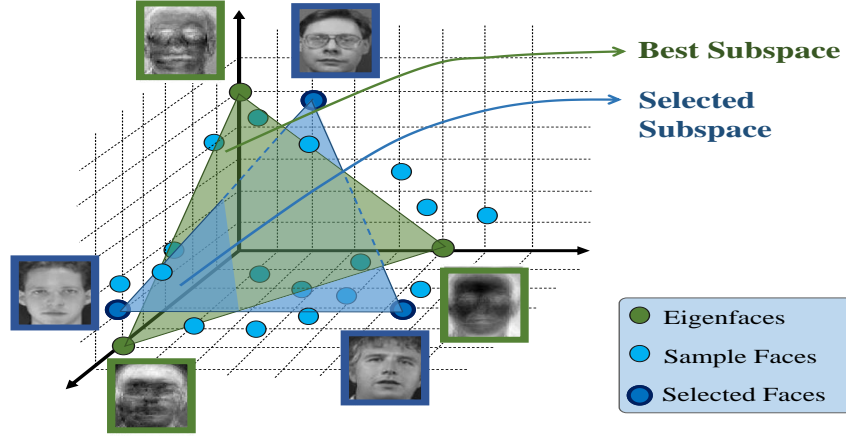


Figure 5.1: A dataset consisting of 20 real images is considered as blue dots. The best possible subspace that spans the dataset is shown in green. However, the significant eigenfaces (green dots) are not among the dataset. We look for the best 3 out of 20 real images whose span is the closest to the span of 3 green eigenfaces; the best subspace is shown in blue. There are $\binom{20}{3}$ possible combinations from which the best representatives must be selected.

5.1 shows the intuition of our proposed algorithm. In other words, Our proposed algorithm finds some pseudo-samples efficiently (which are not among our dataset) and then a few real samples are matched with the found pseudo-samples, iteratively. Iterative methods for finding low-dimensional representations are previously explored [149, 150]. Moreover, a tight upper bound for projection error of selection is derived.

Problem Statement and Related Work

Let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M \in \mathbb{R}^N$ be M given data points of dimension N . We define an $N \times M$ matrix, \mathbf{A} , such that \mathbf{a}_m is the m^{th} column of \mathbf{A} , for $m = 1, 2, \dots, M$. The goal is to reduce this matrix into an $N \times K$ matrix, \mathbf{A}_R , based on an optimality metric. In this section, we introduce some related work on matrix subset selection and data selection.

Selection Based on Diversity

Consider a large system of equations $\mathbf{y} = \mathbf{A}^T \mathbf{w}$, which can be interpreted as a simple linear classifier in which \mathbf{y} is the vector of labels, \mathbf{A} represents the training data and \mathbf{w} is the classifier weights. An optimal sense for data selection is to reduce this system of equations to a smaller system, $\mathbf{y}_R = \mathbf{A}_R^T \hat{\mathbf{w}}$, such that the reduced subsystem estimates the same classifier as the original system, i.e., the estimation error of $\hat{\mathbf{w}}$ is minimized [47] over an assumed distribution for \mathbf{y} . A typical selection objective is to minimize $\|\mathbf{w} - \hat{\mathbf{w}}\|_2$. This criterion is referred to as the *A-optimal* design in the literature of optimization, which is mathematically equivalent to the following problem [48],

$$\begin{aligned} \hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmin}} \quad & \operatorname{tr} \left(\sum_{m=1}^M z_m \mathbf{a}_m \mathbf{a}_m^T \right)^{-1}, \\ \text{subject to} \quad & \|\mathbf{z}\|_0 = K \text{ and } \mathbf{z} \in \{0, 1\}^M, \end{aligned} \tag{5.2}$$

where $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_M]^T$ and z_m indicates the contribution of the m^{th} sample. According to the constraints only K samples can be selected in the reduced system. This is an NP-hard problem which can be solved via convex relaxation with computational complexity of $O(M^3)$ [49].

However, there are other criteria that have some interesting properties. For example *D-optimal* design optimizes the determinant of a reduced matrix [49]. There are several other efforts in this area [9, 27, 26, 50, 51]. Inspired by the D-optimal design, volume sampling (VS), which has received lots of attention, considers a selection probability for each subset of data, which is proportional to the determinant (volume) of the reduced matrix [26, 52, 47]. The VS theory expresses that if $\mathbb{T} \subset \{1, 2, \dots, M\}$ is any subset with cardinality K , chosen with probability

proportional to $\det(\mathbf{A}_{\mathbb{T}}^T \mathbf{A}_{\mathbb{T}})$, then¹,

$$\mathbb{E}\{\|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2\} \leq (K + 1)\|\mathbf{A} - \mathbf{A}_K\|_F^2, \quad (5.3)$$

where $\pi_{\mathbb{T}}(\mathbf{A})$ is a matrix representing the projection of columns of \mathbf{A} onto the span of selected columns indexed by \mathbb{T} . \mathbb{E} indicates expectation operator w.r.t. all the combinatorial selection of K rows of \mathbf{A} out of M . \mathbf{A}_K is the best rank- K approximation of \mathbf{A} , that can be obtained by singular value decomposition and $\|\cdot\|_F^2$ is the Frobenius norm. VS is not a deterministic selection algorithm, as it gives a probability of selection for any subset of samples, and for which only a loose upper bound for the expectation of projection error is guaranteed. In contrast, in the present work a deterministic algorithm is proposed based on direct minimization of projection error using a new optimization mechanism.

Diversity-based selection is very sensitive to outliers and in some applications these methods are employed for outlier detection [53, 54]. A set of outlier samples from a dataset has probably more diverse samples rather than a randomly sampled subset. Thus, diversity-based selection methods should consider outliers properly. Recently, an exemplar-based subspace clustering method is proposed using selection [5]. Their employed selection algorithm is based on selecting farthest sample from previously selected samples and infusing sparsity on the metric of selection. However, our proposed selection algorithm does not necessarily provide diverse samples far from each other.

Representative Selection

A method for sampling from a set of data is proposed by Elhamifar et. al. based on sparse modeling representative selection (SMRS) [6]. Their proposed cost function for data selection is

¹ $\mathbf{A}_{\mathbb{T}}$ is the selected columns of \mathbf{A} indexed by set \mathbb{T} .

the error of projecting all the data onto the subspace spanned by the selected data. Mathematically, the optimization problem in [6] can be written as,

$$\operatorname{argmin}_{|\mathbb{T}|=K} \|\mathbf{A} - \pi_{\mathbb{T}}(\mathbf{A})\|_F^2. \quad (5.4)$$

This is an NP-hard problem and the proposed method in [6] tackles this problem via convex relaxation. However, there is no guarantee that convex relaxation provides the best approximation for an NP-hard problem. Furthermore, such methods that try to solve the selection problem via convex programming are usually computationally too intensive for large datasets [6, 4, 33, 55]. A new fast algorithm for solving Problem (5.4) is proposed.

Dissimilarity-based Sparse Subset Selection (DS3) algorithm selects a subset of data based on pairwise distance of all data to some target points [4]. DS3 considers a source dataset and its goal is to encode the target data according to pairwise dissimilarity between each sample of source and target datasets. This algorithm can be interpreted as the non-linear implementation of SMRS algorithm [4].

Spectrum Pursuit (SP): Our Proposed Selection Method

In this section, an iterative and computationally efficient algorithm is proposed for approximating the solution to the NP-hard selection problem (5.4). The proposed algorithm iteratively finds the best direction on the unit sphere², and then from the available samples in dataset selects the sample with the smallest angle to the found direction. In this section, we present details of our algorithm and investigate its properties.

²In unit sphere, every point corresponds to a unique direction.

The Spectrum Pursuit Algorithm

Projection of all the data onto the subspace spanned by the K columns of \mathbf{A} , indexed by \mathbb{T} , i.e., $\pi_{\mathbb{T}}(\mathbf{A})$, can be expressed by a rank- K factorization, $\mathbf{U}\mathbf{V}^T$. In this factorization, $\mathbf{U} \in \mathbb{R}^{N \times K}$, $\mathbf{V}^T \in \mathbb{R}^{K \times M}$, and \mathbf{U} includes the K columns of \mathbf{A} , indexed by \mathbb{T} which are normalized to have unit length. Therefore, optimization problem (5.4) can be restated as

$$\underset{\mathbf{U}, \mathbf{V}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}\mathbf{V}^T\|_F^2 \text{ s.t. } \mathbf{u}_k \in \mathbb{A}, \quad (5.5)$$

where, $\mathbb{A} = \{\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \dots, \tilde{\mathbf{a}}_M\}$, $\tilde{\mathbf{a}}_m = \mathbf{a}_m / \|\mathbf{a}_m\|_2$, and \mathbf{u}_k is the k^{th} column of \mathbf{U} . It should be noted that \mathbf{U} is restricted to be a collection of K normalized columns of \mathbf{A} , while there is no constraint on \mathbf{V} . Since Problem (5.5) involves a combinatorial search and is not easy to tackle, we modify (5.5) into two consecutive problems. The first sub-problem relaxes the constraint $\mathbf{u}_k \in \mathbb{A}$ in (5.5) to a moderate constraint $\|\mathbf{u}\| = 1$, and the second sub-problem reimposes the underlying constraint. These sub-problems are formulated as

$$(\mathbf{u}, \mathbf{v}) = \underset{\mathbf{u}, \mathbf{v}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}_{\bar{k}}\mathbf{V}_{\bar{k}}^T - \mathbf{u}\mathbf{v}^T\|_F^2 \text{ s.t. } \|\mathbf{u}\| = 1, \quad (5.6a)$$

$$m^{(k)} = \underset{m}{\operatorname{argmax}} |\mathbf{u}^T \tilde{\mathbf{a}}_m|. \quad (5.6b)$$

Here $m^{(k)}$ is the index of the k^{th} selected data point and $\mathbf{a}_{m^{(k)}}$ is the selected sample. Matrix $\mathbf{U}_{\bar{k}}$ is obtained by removing the k^{th} column of \mathbf{U} . Matrix $\mathbf{V}_{\bar{k}}$ is defined in a similar manner. Subproblem (5.6a) is equivalent to finding the first left singular vector (LSV) of $\mathbf{E}_k \triangleq \mathbf{A} - \mathbf{U}_{\bar{k}}\mathbf{V}_{\bar{k}}^T$. The constraint $\|\mathbf{u}\| = 1$ keeps \mathbf{u} on the unit sphere to remove scale ambiguity between \mathbf{u} and \mathbf{v} . Moreover, the unit sphere is a superset for \mathbb{A} and keeps the modified problem close to the recast problem (5.5). After solving for \mathbf{u} (which is not necessarily one of our data points), we find the data point that matches \mathbf{u} the most (makes the smallest angle with \mathbf{u}) in (5.6b).

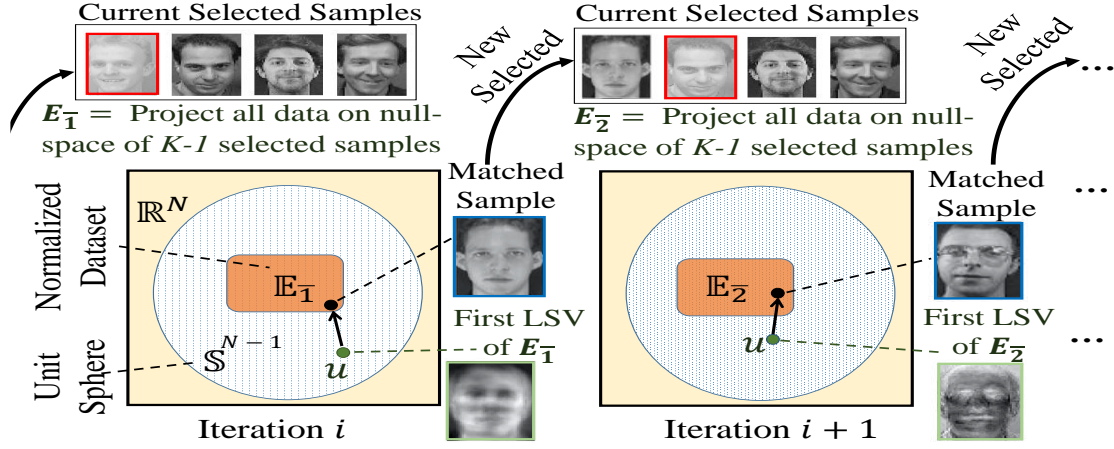


Figure 5.2: Two consecutive iterations of SP algorithm. The first LSV of the residual matrix is a vector on \mathbb{S}^{N-1} and the goal is to find K samples which pursue the spectral characteristics of dataset over iterations.

SP is a low-complexity algorithm with no parameters to be tuned. These properties in addition to its superior performance (as will be shown in many scenarios in Section 6) make SP very desirable for a wide range of applications.

Time complexity order of computing the first singular component of an $M \times N$ matrix is $O(MN)$ [151]. As the proposed algorithm only needs the first singular component for each selection, its time complexity is $O(KNM)$, which is much faster than convex relaxation-based algorithms with complexity $O(M^3)$ [49]. It is worthwhile to mention that the condition which needs to be satisfied for a good performance is $K \leq N < M$. This ensures that the calculated first LSV is reliable because when we access to a small number of data points (small M), the first LSV is highly dependent on each single data point. While given a large number of samples (large M), the first LSV is robust to changes in the dataset. SP also performs faster than the K-medoids algorithm and volume sampling, whose complexity are of the order $O(KN(M-K)^2)$ and $O(MKN \log N)$, respectively [152, 9]. The steps of the SP algorithm are elaborated in Algorithm 11 and Fig. 5.2 illustrates Problem (5.6) pictorially.

Algorithm 11 Spectrum Pursuit Algorithm

Require: \mathbf{A} and K

Output: $\mathbf{A}_{\mathbb{T}}$

1: **Initialization:**

$\mathbb{T} \leftarrow$ A random subset of $\{1, \dots, M\}$ with $|\mathbb{T}| = K$
 $\{\mathbb{T}_k\}_{k=1}^K \leftarrow$ Partition \mathbb{T} into K sets that each one has 1 element.
iter = 0

while the stopping criterion is not met

2: $k = \text{mod}(\text{iter}, K) + 1$
3: $\mathbf{U}_{\bar{k}} = \text{normalize column}(\mathbf{A}_{\mathbb{T} \setminus \mathbb{T}_k})$
4: $\mathbf{V}_{\bar{k}} = \mathbf{A}^T \mathbf{U}_{\bar{k}} (\mathbf{U}_{\bar{k}}^T \mathbf{U}_{\bar{k}})^{-1}$
5: $\mathbf{E}_{\bar{k}} = \mathbf{A} - \mathbf{U}_{\bar{k}} \mathbf{V}_{\bar{k}}^T$
6: $\mathbf{u}_k =$ first left singular-vector of $\mathbf{E}_{\bar{k}}$ by solving (5.6a)
7: $\mathbb{T}_k \leftarrow$ index of the most correlated data with \mathbf{u}_k (5.6b)
8: $\mathbb{T} \leftarrow \bigcup_{k'=1}^K \mathbb{T}_{k'}$
9: iter = iter + 1
end while

The following theorem shows that SP selects diversely from a union of subspaces.

Theorem 1 *Assume columns of \mathbf{A} lie on P clusters and each cluster forms a k_p -dimensional subspace in which $\sum_{p=1}^P k_p = K \leq N$. Selection of K samples using SP provides exactly k_p samples from each cluster.*

Sequential SP

An interesting property of Eigen decomposition is its sequential property. It means that the best rank- K approximation of a dataset is exactly equal to the best approximation with rank of $K - 1$ plus one new component which is orthogonal to the previously found components. However, a selection problem generally does not have a sequential solution. In other words, selection problems have a combinatorial solution and the best K subset of data is not related to the best $K - 1$ samples of data.

In some applications such as active learning or online data reduction, it is essential that the selection be sequential. In these scenarios, data are selected over time one by one consecutively,

and previously selected samples are already processed and cannot be replaced with another sample. The sequential version of SP is previously presented as the iterative projection and matching (IPM) algorithm [3]. IPM is a very simple and relatively accurate data selection algorithm. For the sake of consistency, to the sequential implementation of SP is referred as IPM in this manuscript. The steps in IPM are summarized in Alg. 12.

Algorithm 12 Iterative Projection and Matching Algorithm [3]

Require: A and K

Output: A_S

Initialize Alg. 11 with $S = \{\}$ and perform only K iterations

The first selected sample using IPM is the sample which is the most correlated with the first left singular vector (LSV) of data. The first selected sample is denoted by $m^{(1)}$. After selecting the first data point ($\mathbf{a}_{m^{(1)}}$), we project all data points onto the null space of the selected sample. This forms a new data matrix $A(\mathbf{I} - \tilde{\mathbf{a}}_{m^{(1)}}\tilde{\mathbf{a}}_{m^{(1)}}^T)$, where \mathbf{I} is an identity matrix. We select one more sample in the same fashion utilizing the new matrix to find the second data point. This process continues until we select K data points. It should be noted that the null space of selected sample(s) indicates a subspace that the selected sample(s) cannot span. Therefore, the next selected data is obtained by only searching in this null space. Fig. 5.3 shows an intuitive explanation of one iteration of IPM algorithm. First, the leading LSV is computed, and then the most correlated sample in the dataset is matched with the computed singular vector. Next, all data are projected onto the null space of the matched sample. The projected data are ready to perform one more iteration, if required.

A Lower Bound on Maximum Correlation

In this section, we derive a lower bound on the maximum of the absolute value of the correlation between columns of $\mathbf{E}_{\bar{k}}$ and the first LSV defined in Alg. 11.

Next, we present a lemma that guarantees the existence of a column in $\mathbf{E}_{\bar{k}}$ which is highly corre-

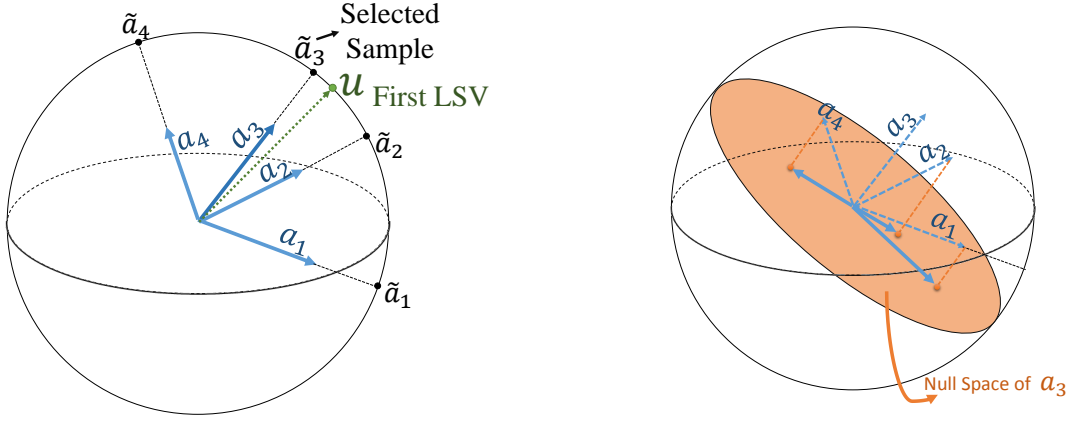


Figure 5.3: A toy example that illustrates the first iteration of the sequential SP (IPM). (Left) The most matched sample with the first left singular vector, \mathbf{u} , is selected. (Right) The rest of samples are projected on the null space of the selected sample in order to continue selection in the lower dimensional subspace.

lated with the first LSV, illustrating the fact that the direction of selected sample will not be distant from the optimal direction shown by the first LSV.

Lemma 1 *Let $\mathbf{e}_1^k, \mathbf{e}_2^k, \dots, \mathbf{e}_M^k \in \mathbb{R}^N$ construct columns of \mathbf{E}_k . In each iteration, let σ_1 , \mathbf{u} and \mathbf{v} denote the first singular value, the corresponding left and right singular vectors of \mathbf{E}_k , respectively. Then, there exists at least one column in \mathbf{E}_k (corresponds to a data point) such that the absolute value of its inner product with \mathbf{u} is greater than or equal to $\frac{\sigma_1}{\sqrt{M}}$. Hence, $\max_m |\mathbf{u}^T \mathbf{e}_m^k| \geq \frac{\sigma_1}{\sqrt{M}}$ for each iteration k .*

The following proposition states a lower bound on the maximum of the absolute value of the correlation between data points $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M$ and \mathbf{u} , when data are normalized on the unit sphere. First, let us define the following measure.

Definition 6 *Rank-oneness measure (ROM) of a rank R matrix \mathbf{A} with singular values $\sigma_1, \sigma_2, \dots, \sigma_R$ is defined as $\mathcal{R}_\mathbf{A} = \sqrt{\frac{\sigma_1^2}{\sum_{r=1}^R \sigma_r^2}} = \frac{\sigma_1}{\|\mathbf{A}\|_F}$.*

Proposition 2 Assume columns of $\mathbf{E}_{\bar{k}}$ are normalized to lie on the unit sphere. There exists at least one column, \mathbf{e}_i^k , such that the correlation coefficient between \mathbf{a}_i^k and the first left singular vector of $\mathbf{E}_{\bar{k}}$ is greater than or equal to $\mathcal{R}_{\mathbf{E}_{\bar{k}}}$.

Upper Bound on Projection Error

The main theoretical result of our work relates to the upper bound for projection error of selection of a new sample using SP. All proof for the theoretical results are provided in the appendix.

Theorem 2 If the columns of matrix \mathbf{A} contain M zero-mean samples in N dimensional space and \mathbf{a}_i is the first selected sample using SP, then,

$$\|\mathbf{A} - \pi_i(\mathbf{A})\|_F^2 \leq (1 + \mathcal{R}_{\mathbf{A}}^2(1 + \mathcal{R}_{\mathbf{A}})(1 - \mathcal{R}_{\mathbf{A}}))\|\mathbf{A} - \mathbf{A}_1\|_F^2,$$

where $\|\mathbf{A} - \pi_i(\mathbf{A})\|_F^2$ is the projection error on the span of the selected sample and \mathbf{A}_1 is the best rank-one approximation.

Obviously, $\|\mathbf{A} - \mathbf{A}_1\|_F^2$ is the lower bound for projection error based on the definition of SVD. However, this theorem states that the upper bound is a scale (≥ 1) of the lower bound and the scale is $1 + \mathcal{R}_{\mathbf{A}}^2(1 + \mathcal{R}_{\mathbf{A}})(1 - \mathcal{R}_{\mathbf{A}})$.

Proposition 3 Assume \mathbf{a}_i is the first selected sample using SP. Then,

$$\|\mathbf{A} - \pi_i(\mathbf{A})\|_F^2 \leq 1.25\|\mathbf{A} - \mathbf{A}_1\|_F^2.$$

When $\mathcal{R}_{\mathbf{A}} = 1$, the upper bound of projection error is equal to its lower bound since the dataset

is rank-one. Thus, any selection (even random selection) provides the same subspace which is equal to the subspace of rank-one approximation. On the other hand, when \mathcal{R}_A is too small,³ distribution of points in the dataset is symmetric. Thus, a specific data point does not have a priority to be selected. Therefore, for such datasets even random selection of a sample provides a close projection error in comparison to the best projection error. In other words, for very low-rank and very high-rank datasets, selection is not challenging and there are trivial solutions. The most challenging scenario for selection of a new sample occurs, when $\mathcal{R}_A = \sqrt{2}/2$ and the gap between the lower bound and the upper bound is maximized. In this case, the role of selection algorithm is more critical because the dataset is neither highly structured nor symmetrically-spread in the space. Interested readers are referred to the appendix.

Robustness to Perturbation

Data selection algorithms are vulnerable to outlier samples. Since outlier samples are more spread in the space of data, their span covers a wider subspace. However, the spanned subspace by outliers may not be a proper representative subspace. DS3 adds a penalty to the cost function in order to reject outliers [4]. Our proposed algorithm computes the first singular vector as the leading direction in each iteration. We show here that this direction is the most robust spectral component against changes in the data. First consider the autocorrelation matrix of data defined as, $\mathbf{C} = \sum_{m=1}^M \mathbf{a}_m \mathbf{a}_m^T$.

Eigenvectors of this matrix are equal to left singular vectors of \mathbf{A} . Adding a new row in \mathbf{A} does not change the size of matrix \mathbf{C} , but perturbs this matrix. The following lemma shows the robustness of eigenvectors of \mathbf{C} against perturbations.

³Please note that \mathcal{R}_A is greater than $1/\sqrt{N}$.

Lemma 2 Assume square matrix C and its spectrum $[\lambda_i, \mathbf{u}_i]$. Then, the following inequality holds,

$$\|\partial \mathbf{u}_i\|_2 \leq \sqrt{\sum_{j \neq i} \frac{1}{(\lambda_i - \lambda_j)^2}} \|\partial C\|_F.$$

Definition 7 The sensitivity coefficient of the i^{th} eigenvector of a square matrix is defined by, $s_i \triangleq \sqrt{\sum_{j \neq i} \frac{1}{(\lambda_i - \lambda_j)^2}}$.

It is easy to show that $s_1 < s_2$. Based on Lemma 2 and this definition the following proposition suggests a condition to satisfy $s_1 < s_i$, $\forall i \geq 2$.

Proposition 4 Assume square matrix C and its spectrum $[\lambda_i, v_i]$, where the gap between consecutive eigenvalues is decreasing. Then, $s_1 < s_i$, $\forall i \geq 2$.

The proofs of Propositions and Lemmas in this section are presented in the supplementary material. Moreover, the results of Proposition 2 and 4 are also verified in the appendix.

Residual Descent Implementation

The best minimizer for (5.6a) is the first LSV. However, restriction to data samples does not imply that the most correlated sample with the first LSV is necessarily the best minimizer for (5.6a), although in most cases the most correlated sample with the first LSV is the best minimizer. In order to find the best sample that minimizes (5.6a) at each iteration, we propose to collect a few samples that are correlated with the first LSV and compute residual error for these samples. Let Ω denote the set of P most correlated samples with the first LSV. The modified version of Problem

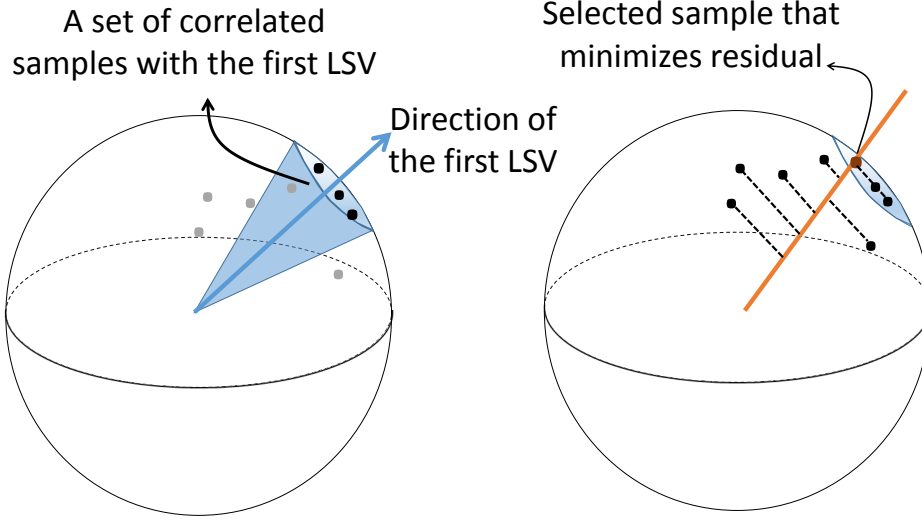


Figure 5.4: SP-RD algorithm does not select directly the most correlated sample with the first left singular vector. First, a small subset of samples which are correlated with the first left SV are grouped. Then, the sample which is the best minimizer for (5.7c) is selected.

(5.7) can be written as follows.

$$(\mathbf{u}, \mathbf{v}) = \underset{\mathbf{u}, \mathbf{v}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}_{\bar{k}} \mathbf{V}_{\bar{k}}^T - \mathbf{u} \mathbf{v}^T\|_F^2 \text{ s.t. } \|\mathbf{u}\| = 1, \quad (5.7a)$$

$$\Omega = \underset{|\Omega|=P}{\operatorname{argmax}} \sum_{c \in \Omega} |\mathbf{u}^T \tilde{\mathbf{e}}_c|, \quad (5.7b)$$

$$\mathbb{S}_k = \underset{c \in \Omega, \mathbf{v}}{\operatorname{argmin}} \|\mathbf{A} - \mathbf{U}_{\bar{k}} \mathbf{V}_{\bar{k}}^T - \mathbf{u} \mathbf{v}^T\| \text{ s.t. } \mathbf{u} = \tilde{\mathbf{a}}_c. \quad (5.7c)$$

Residual error over iterations of SP is not necessarily decreasing. To make the error monotonically decreasing, as a sufficient condition for convergence, we can include the index of the previously selected sample in for Ω for updating the k^{th} selected sample. Selection cost function (projection error) can be made monotonically decreasing by modifying set Ω over iterations. We refer to the modified algorithm as spectrum pursuit with residual descent (SP-RD). In order to select K sam-

ples, the lower bound for selection cost function is $\|\mathbf{A} - \mathbf{A}_K\|$ in which \mathbf{A}_K is the best rank- K approximation of \mathbf{A} . Since the cost function is monotonically decreasing, and it is lower bounded, the modified algorithm is convergent. However, simulation experiments show the plain SP algorithm is also observed to be empirically convergent for real and synthetic datasets. Alg. 13 and Fig. 5.4 show the steps in SP-RD algorithm.

Algorithm 13 Spectrum Pursuit Algorithm with Residual Descend (SP-RD)

Require: \mathbf{A} , P and K

Output: \mathbf{A}_S

1: **Initialization:**

$\mathbb{S} \leftarrow$ A random subset of $\{1, \dots, M\}$ with $|\mathbb{S}| = K$

$\{\mathbb{S}_k\}_{k=1}^K \leftarrow$ Partition \mathbb{S} into K sets that each one has 1 element.

iter = 0

while the stopping criterion is not met

2: $k = \text{mod}(\text{iter}, K) + 1$

3: $\mathbf{U}_{\bar{k}} = \text{normalize column}(\mathbf{A}_{\mathbb{S} \setminus \mathbb{S}_k})$

4: $\mathbf{V}_{\bar{k}} = \mathbf{A}^T \mathbf{U}_{\bar{k}} (\mathbf{U}_{\bar{k}}^T \mathbf{U}_{\bar{k}})^{-1}$

5: $\mathbf{E}_{\bar{k}} = \mathbf{A} - \mathbf{U}_{\bar{k}} \mathbf{V}_{\bar{k}}^T$

6: $\mathbf{u}_k =$ first left singular-vector of $\mathbf{E}_{\bar{k}}$ by solving (5.7a)

7: $\Omega \leftarrow$ indices of the most P correlated data with \mathbf{u} (5.7b)

8: $\Omega = \Omega \cup \mathbb{S}_k$

9: $\mathbb{S}_k \leftarrow$ the sample in Ω that minimizes (5.7c)

10: $\mathbb{S} \leftarrow \bigcup_{k'=1}^K \mathbb{S}_{k'}$

11: iter = iter + 1

end while

Kernel SP: Selection based on a Locally Linear Model

The goal of CSSP introduced in (5.1) is to select a subset of data whose *linear subspace* spans all data. Obviously, this model is not proper for general data types that mostly lie on nonlinear manifolds. Accordingly, we generalize (5.1) and propose the following selection problem in order to efficiently sample from a union of manifolds

$$\underset{|\mathbb{S}| \leq K}{\operatorname{argmin}} \sum_{m=1}^M \|\mathbf{a}_m - \pi_{\mathbb{S}_m}(\mathbf{a}_m)\|_F^2 \quad \text{s.t. } \mathbb{S}_m \subseteq \mathbb{S} \cap \Omega_m, \quad (5.8)$$

where Ω_m represent the indices of local neighbors of \mathbf{a}_m based on an assumed distance metric. This problem is simplified to CSSP in Problem (5.1) if Ω_m is assumed to be equal to $\{1, \dots, M\}$. Problem (5.8) is written for each column of \mathbf{A} separately in order to engage neighborhood for each data. This problem facilitates fitting a locally linear subspace for each data sample in terms of its neighbors. *Nonlinear* techniques demonstrates significant improvement upon linear methods for many scenarios [153, 154, 155].

Here we propose an extension of SP, referred to as kernel SP (KSP), to tackle the combinatorial search Problem (5.8). Manifold-based dimension reduction techniques and clustering algorithms do not provide prototypes suitable for data selection. However, inspired by spectral clustering of manifolds [156], main tool for nonlinear data analysis that partitions data into nonlinear clusters based on spectral components of the corresponding normalized similarity matrix, we formulate KSP as

$$\mathbb{S} = \underset{|\mathbb{S}| \leq K}{\operatorname{argmin}} \|\mathbf{L} - \pi_{\mathbb{S}}(\mathbf{L})\|_F^2, \quad (5.9)$$

where $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}^{-\frac{1}{2}}$, is the normalized similarity matrix of the data. Matrix $\mathbf{S} = [s_{ij}] \in \mathbb{R}^{M \times M}$ is defined as the similarity matrix of data and \mathbf{D} is a diagonal matrix and $d_{ii} = \sum_{j \neq i} s_{ij}$. The similarity matrix can be defined based on any similarity measure. A typical choice is a Gaussian kernel with parameter α . Note that problem (5.9) is the same as problem (5.1), where \mathbf{A} is replaced by \mathbf{L} . The steps of the KSP algorithm are summarized in Algorithm 14.

Fig. 5.5 illustrates the impact of nonlinear modeling on a toy example containing a set of 100×100 images where each image is a rotated and resized version of other images (Fig. 5.5(a)). Since none of the images lie on the linear subspace spanned by the rest of images, the ensemble of these data do not form a linear subspace. Therefore, this dataset is of high rank and the union of linear subspaces is not a proper underlying model for it. The KSP algorithm is implemented using a Gaussian kernel with parameter α , i.e., $s_{ij} \triangleq e^{-\alpha \|\mathbf{a}_i - \mathbf{a}_j\|^2}$. As shown in Fig. 5.5 (c), the nonlinear

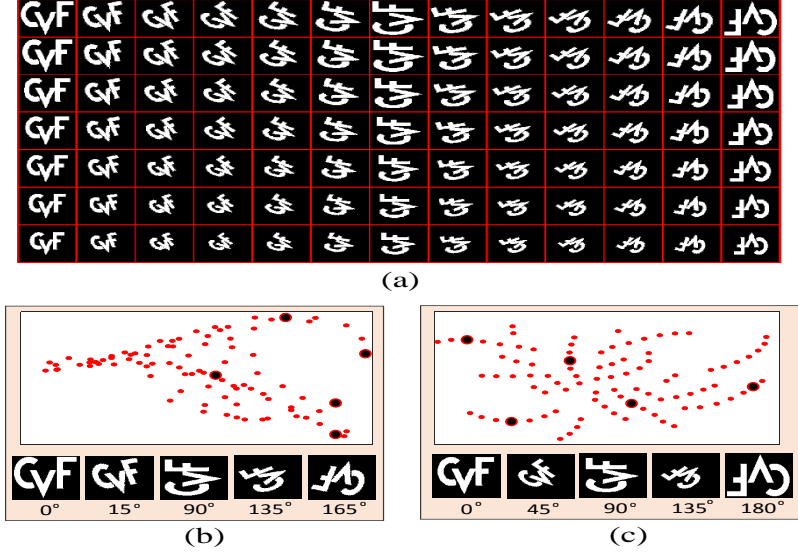


Figure 5.5: (a) A dataset lies on a two dimensional manifold identified by two parameters, rotation and size. However, the rank of corresponding matrix to this dataset is a large number. (b) Linear embedding using linear PCA and selection using linear SP. (c) nonlinear embedding using tSNE[2] and selection using kernel-SP. Un-selected and selected samples are shown as red and black dots in the embedded space, respectively.

selection algorithm has been able to discover the intrinsic structure of data and select data from more distinguished angles than that of Fig. 5.5 (b) in which the plain SP is applied.

Algorithm 14 Kernel Spectrum Pursuit

Require: A , α , and K

Output: \mathbb{S}

- 1: $S \leftarrow$ Similarity Matrix: $s_{ij} = e^{-\alpha \|a_i - a_j\|_2^2}$
 - 2: Form diagonal matrix D where $d_{ii} = \sum_{i \neq j} s_{ij}$
 - 3: $L = D^{-1/2} S D^{-1/2}$.
 - 4: $\mathbb{S} \leftarrow$ Apply SP on L with K (Alg. 11)
-

Conclusion

A novel approach to data selection from linear subspaces is proposed and its extension for selection from nonlinear manifolds is presented. The proposed SP algorithm demonstrates an accurate

solution for CSSP. Moreover, SP and KSP have shown superior performance in many applications which will be shown in the next chapter. The investigated fast and efficient deep learning frameworks, empowered by our selection methods, have shown that dealing with selected representatives is not only fast but can also be more effective.

CHAPTER 6: APPLICATIONS OF DATA SUBSET SELECTION

To validate our theoretical investigation and to empirically demonstrate the behavior and effectiveness of the proposed selection technique, we perform extensive sets of experiments considering several different scenarios. We divide our experiments into three different subsections.

In the first section we show the effectiveness of SP in selecting the most informative representatives, by training the classifier using only a few representatives from each class. Afterwards, in Section application of SP in fast subspace clustering is illustrated. It is shown that SP outperforms the state-of-the-art selection-based subspace clustering in terms of grouping accuracy.

Representatives for Multi-PIE Dataset

Here, we present our experimental results on CMU Multi-PIE Face Database [157]. We use 249 subjects from the first session with 13 poses, 20 illuminations, and two expressions. Thus, there are $13 \times 20 \times 2$ images per subject. Fig. 6.1 shows 10 selected images from 520 images



Figure 6.1: Selection of 10 representatives out of 520 images of a subject. IPM and SP select from more diverse angles.

of a subject. As it can be seen, the results of K-medoids and DS3 algorithms are concentrated on side views, while our selection provides images from more diverse angles. IPM and SP select from different angles, while the selected images by DS3 and K-medoids contain repetitious angles. Fig. 6.2 shows the performance of different selection algorithms in terms of normalized projection error and running time. It is evident that our proposed approach finds a better minimizer for CSSP defined in Equation (5.1) and is able to do so in orders of magnitude less time than the state of the art methods based on convex relaxation. The total computational burden of SP is more than that of IPM and SP solves CSSP more accurately.

Employing SP-RD which is introduced in Alg. 13 is able to further improve the accuracy of CSSP solution. However, SP-RD algorithm needs parameter P . As P increases, a better solution is achieved. Parameter P can be set according to the accessible computation power. Fig. 6.3 compares the performance of IPM and SP with SP-RD with three different parameters. In Fig. 6.3 in addition to error ratio, the running time for selecting 10 representatives per class for 249 classes is presented in the parentheses in a plot legend box on top left as an indicator for computational burden .

Representatives To Generate Multi-view Images Using GAN

Next, to investigate the effectiveness of the proposed selection, we use the selected samples to train a generative adversarial network (GAN) to generate multi-view images from a single-view input. For that, the GAN architecture proposed in [10] is employed. Following the experiment setup in [10], only 9 poses between $\frac{\pi}{6}$ and $\frac{5\pi}{6}$ are considered. Furthermore, the first 200 subjects are used for training and the rest for testing. Thus, the total size of the training set is 72,000, 360 per subject. All the implementation details are same as [10], unless otherwise is stated¹.

¹We use the code provided by the authors at <https://github.com/bluer555/CR-GAN>

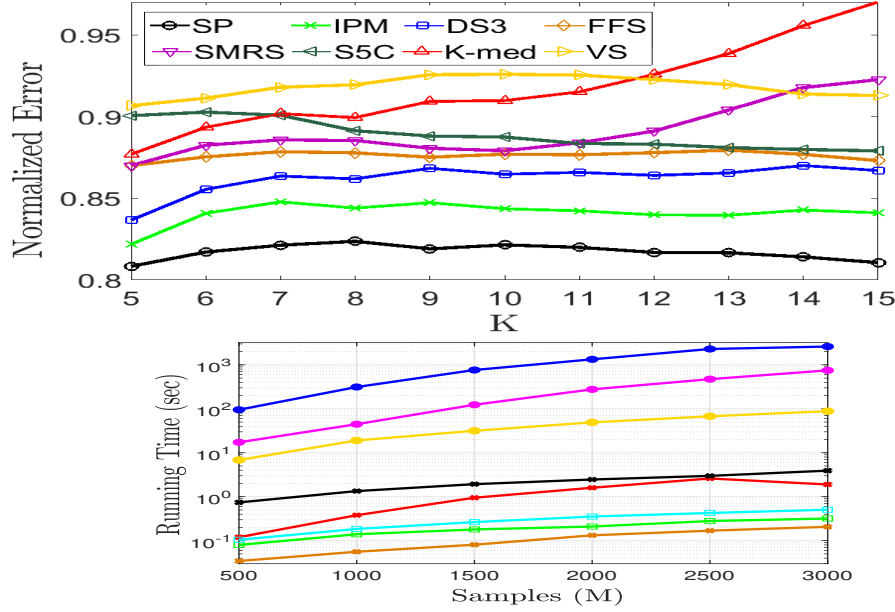


Figure 6.2: Performance of different methods in terms of their accuracy for CSSP defined in (5.1). The proposed SP algorithm is compared with IPM [3], and DS3 [4], FFS [5], SMRS [6], 2phase [7], K-medoids [8] and volume sampling [9]. (Top) The ratio of projection error using selection algorithms to projection error of random selection for selecting K representatives from each subject, averaged over all the subjects. (Bottom) Running time of different algorithms versus number of input samples for selecting 10 samples. Our SP algorithm is slower than IPM, however, it is more accurate.

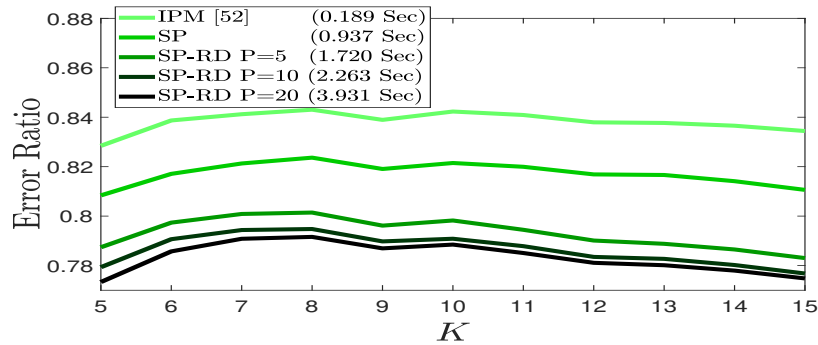


Figure 6.3: The ratio of projection error as a function of K selected samples for each class of Multi-PIE dataset. It is averaged for all 250 classes and the running time for selecting from the whole dataset is reported in the parenthesis. IPM and SP do not require any parameter, and Parameter P in SP-RD algorithm does not need tuning. It can be fixed according to the accessible computational power.

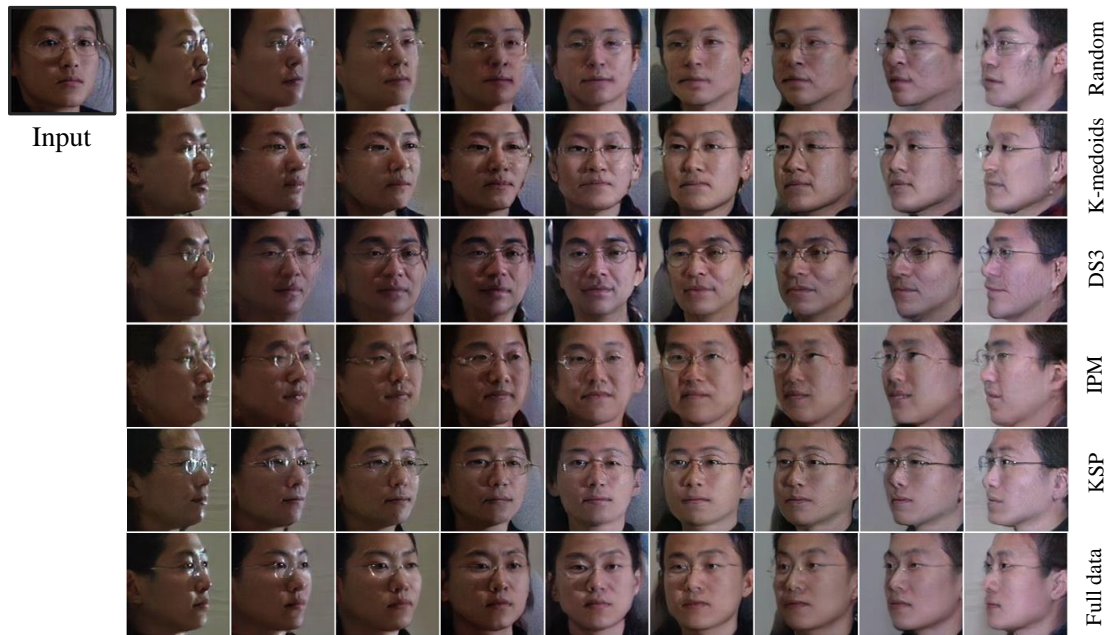


Figure 6.4: Multi-view face generation results for a sample subject in testing set using CR-GAN [10]. The network is trained on a selected subset of training set (9 images per subject) using random selection (first row), K-medoids (second row), DS3 [4] (third row), and SP (fourth row). The fifth row shows the results generated by the network trained on all the data (360 images per subject). SP generates closest results to the complete dataset.

We select only 9 images from each subject (1800 total subjects), and train the network with the reduced dataset for 300 epochs using the batch size of 36. Fig. 6.4 shows the generated images of a subject in the testing set, using the trained network on the reduced dataset, as well as using the complete dataset. The network trained on samples selected by SP (fourth row) is able to generate more realistic images, with fewer artifacts, compared to other selection methods (rows 1-3). Furthermore, compared to the results using all the data (row 5), it is clear that SP generates the closest results to the complete dataset. This is because, as demonstrated in Fig. 6.1, samples selected by SP cover more angles of the subject, resulting in better training of the GAN. See supplementary material for further experiments and sample outputs.

For a quantitative performance investigation, we evaluate the identity similarities between the real

Table 6.1: Identity dissimilarities between real and generated images by network trained on reduced (using different selection methods) and complete dataset.

Method	Rand	K-Med	FFS	DS3	IPM	SP
9 per subj	0.562	0.599	0.608	0.602	0.553	0.550
360 per subj	0.5364					

Table 6.2: Accuracy (%) of ResNet18 on UCF-101 dataset, trained using only the representatives selected by different methods. The accuracy using the full training set (9537 samples) is 82.23%.

Samples per class	1	2	3	4	5	6	7	8	9	10
Random	54.6	64.7	69.2	70.5	72.9	74.0	76.0	75.6	76.0	77.0
K-medoids	61.0	67.7	69.4	70.9	71.7	72.0	72.5	75.2	73.6	73.5
DS3[4]	60.8	69.1	74.0	75.2	74.9	75.3	75.8	77.0	77.6	76.6
IPM	62.3	69.4	72.1	73.5	75.3	77.0	77.2	77.4	77.5	78.1
SP	62.3	70.8	74.2	74.6	76.9	77.6	78.1	78.4	78.7	79.2

and generated images. For that, we feed each pair of real and generated images to a ResNet18², trained on MS-Celeb-1M dataset [159], and obtain 256-dimensional features. ℓ_2 distances of features correspond to the face dissimilarity. Table 6.1 shows the normalized ℓ_2 distances between the real and generated images, averaged over all the images in the testing set. Our method outperforms other selection methods in this metric as well. Thus, from Fig. 6.4 (qualitative) and Table 6.1 (quantitative), we can conclude that the SP-reduced training set contains more information about the complete set, compared to other selection methods.

Representatives for ImageNet

In this section, we use ImageNet dataset [160] to show the effectiveness of SP in selecting the representatives for image classification task. To this end, first, we extract features from images

²We use the naive ResNet18 architecture as described in [158].

Table 6.3: Top-1 classification accuracy (%) on ImageNet, using selected representatives from each class. Accuracy using all the labeled data ($\sim 1.2\text{M}$ samples) is 46.86%. Numbers in () show the size of the selected representatives as a % of the full training set.

Images per Class	1 (0.08%)	5 (0.4%)	10 (0.8%)	50 (4%)
Random	3.18	8.71	12.97	25.61
K-Medoids	11.78	17.01	17.56	26.86
IPM	12.50	21.69	25.26	30.77
SP	12.50	23.02	26.91	32.48

in an unsupervised manner, using the method proposed in [161]. Next, we perform selection in the learned 128-dimensional space and perform k -nearest neighbors (k -NN) using the learned similarity metric, following the experiments in [161]³. Here, we show that we can learn the feature space and the similarity metric in an unsupervised manner, as there is no shortage of unlabeled data, and use only a few labeled representatives to classify the data.

Due to the high volume of this dataset, selection methods based on convex-relaxation, such as DS3 [4] and SMRS [6], fail to select class representatives in a tractable time (as discussed before and shown in Fig. 6.2 for Multi-PIE dataset). Table 6.3 shows the top-1 classification accuracy for the testing set using k -NN. Using less than 1% of the labels, we can achieve an accuracy of more than 25%, showing the potential benefits of the proposed approach for dataset reduction. Classification accuracy of k -NN, using the learned similarity metric, reflects the representativeness of the selected samples, thus highlighting the fact that SP-selected samples preserve the structure of the data fairly well.

³We use the feature space generated by the ResNet50 backbone, as provided in <https://github.com/zhirongw/lemniscate.pytorch>

Fast Subspace Clustering

In this section, we apply our data selection method for carrying out fast subspace clustering. Subspace clustering is reviewed in [162]. In [163] and [164], sparse sub-space clustering is discussed. In subspace clustering, the goal is to identify some low-dimensional subspaces, the union of which encompasses the ensemble data. The data $\mathbf{A} \in \mathbb{R}^{N \times M}$ is a collection of points from G independent linear subspaces $\mathcal{S}_g, g \in [1, \dots, G]$. The g^{th} subspace contains M_g data points, the union of which forms the entire dataset. Let all the data in \mathcal{S}_g be denoted with $\mathbf{A}_g \in \mathbb{R}^{N \times M_g}$. In [163], the following problem is considered in order to find the similarity coefficients of the data, aka LASSO:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_1 \quad \text{subject to} \quad \mathbf{A} = \mathbf{AZ}, \text{diag}(\mathbf{Z}) = \mathbf{0}. \quad (6.1)$$

The coefficients in $\mathbf{Z} \in \mathbb{R}^{M \times M}$ represent the similarity, i.e. how each data element can be expressed as a summation of other data elements. The matrix \mathbf{Z} is used to form the symmetric matrix $\mathbf{W} = \mathbf{Z} + \mathbf{Z}^T$, which is fed into the spectral clustering algorithm. This framework is known as sparse subspace clustering (SSC) which exploits whole data to perform LASSO. Fig. 6.5 shows the idea behind sparse subspace clustering in which few critical data identify the underlying clusters and then whole data are clustered accordingly. In [5], authors propose a different method in achieving the matrix \mathbf{W} based on (k -NN). We use the same approach in finding the similarity matrix to feed it to the spectral clustering algorithm. Next, we apply SP to select the data and show that this selection method achieves better accuracy in subspace clustering in comparison to other known selection methods including the one introduced in [5]. This framework reduces the complexity in comparison to utilizing the entire data for subspace clustering (as in [163]) significantly, since finding representatives and performing regression on a subset of selected data reduces to the following computations:

$$\min_{\tilde{\mathbf{Z}}} \|\mathbf{A} - \mathbf{A}_{\mathcal{S}}\tilde{\mathbf{Z}}\|_F,$$

Table 6.4: Unsupervised clustering accuracy for MNIST handwritten dataset.

Method	Accuracy	clustering on all data	
Rand	40.5%	Kmeans	SSC[11]
K-medoids	46.5%	47.1%	48.2%
FFS [5]	49.4%		
DS3 [4]	48.3%		
IPM [3]	48.9%		
SP	52.0%		

where \mathbf{A}_S shows the selected columns of the data matrix and $\tilde{\mathbf{Z}} \in \mathbb{S}^{K \times M}$ indicates coefficients for samples of the entire data. In addition, we improve the accuracy further in comparison to the methods which initially select the data and apply the clustering, since the SP algorithm selects data evenly from a union of low-rank subspaces as shown in Theorem 1. Please note that the reduced regression problem is an over-determined system of equations and it does not require regularization. While the original problem is under-determined. Thus LASSO is employed to obtain a robust regression.

We implement the subspace clustering on synthesized and real datasets. First, we synthesize two 4-dimensional clusters in 20-dimensional space. We assign the data to two low-dimensional labeled classes. The synthetic data is contaminated with additive noise and 10% of the dataset includes outliers. The comparisons of different selection algorithms are shown in Fig. 6.6. For $K = 8$ samples, where K is the dimension of the union of the two clusters (rank-8 subspace), SP reaches the highest accuracy in subspace clustering. It is worth noting that SP achieves a maximum accuracy level extremely close to the maximum of 96.47% which is obtained on full data using SSC.

In the second set of experiments, we use the MNIST handwritten digit dataset for unsupervised clustering. First, we perform our subspace clustering into 10 clusters and then compare the clusters with the oracle labels in order to report a clustering accuracy score. Interestingly, the selection and clustering using SP reaches higher accuracy than performing clustering on the entire data as shown in Table 6.4.

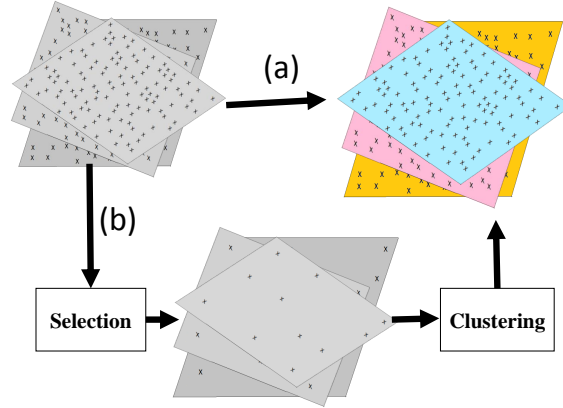


Figure 6.5: Two approaches for subspace clustering. (a) Identify subspaces directly from ensemble of data. (b) First select a set of representatives and then identify subspaces accordingly. Our selection algorithm facilitates the second approach.

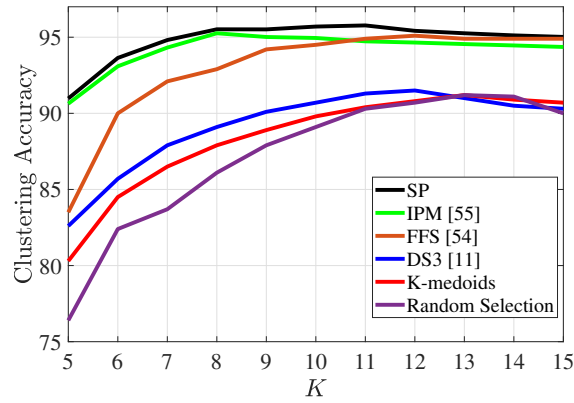


Figure 6.6: Accuracy of clustering of two synthetic clusters contaminated with noise and containing 10% outlier samples. Clustering of full data results in 96.47% accuracy using SSC [11].

Conclusion

Some applications of the proposed SP algorithm are presented in this chapter. It is shown that selection makes the main task faster while we can keep the accuracy as of whole data. This chapter only presented a selected applications. Interested readers are referred to published works in order to find more applications [3, 165].

CHAPTER 7: MULTI-WAY SELECTION

Matrix factorization provides a concise representation of data via low-rank approximation. Despite desirable uniqueness conditions and computational simplicity of the well-known singular value decomposition (SVD), it comes with some fundamental shortcomings. The intrinsic structure of data is not inherited to the singular components. Moreover, SVD implies orthogonality on the components which is irrelevant to the underlying structure of the original data. This enforced structure makes the bases, a.k.a. singular vectors, hard to interpret [166]. On the other hand, it is shown that borrowing bases from the actual samples of a dataset provides a robust representation, which can be employed in interesting applications where sampling is their heart [167]. This problem is studied under the literature of column subset selection problem (CSSP) [9, 19] and CUR decomposition [168, 169]. A general problem for CSSP and CUR can be written in the following form [170]:

$$(\mathbb{S}^c, \mathbb{S}^r) = \underset{\mathbb{S}^c, \mathbb{S}^r}{\operatorname{argmin}} \|\mathbf{X} - \pi_{\mathbb{S}^r}^r(\pi_{\mathbb{S}^c}^c(\mathbf{X}))\|_F^2, \quad (7.1)$$

where, $\mathbf{X} \in \mathbb{R}^{N \times M}$ is the data matrix containing M data points in an N -dimensional space. Here, $\pi_{\mathbb{S}^c}^c(\cdot)$ and $\pi_{\mathbb{S}^r}^r(\cdot)$ indicate column space projection and row space projection, respectively. These operators project all columns (rows) to a low-dimensional subspace spanned by selected columns (rows) of matrix \mathbf{X} indexed by the set \mathbb{S}^c (\mathbb{S}^r). The chronological order in applying $\pi_{\mathbb{S}^c}^c(\cdot)$ and $\pi_{\mathbb{S}^r}^r(\cdot)$ does not affect the problem since these operators are linear. Moreover, substituting $\pi_{\mathbb{S}^r}^r$ with the identity projection simplifies the problem to CSSP. Fig. 7.1 illustrates the structure of CUR matrix decomposition as a self-representative approach. The column-wise projection is a matrix multiplication from the left side and the row-wise projection is a matrix multiplication from the right side. Mathematically speaking,

$$\pi_{\mathbb{S}^r}^r(\pi_{\mathbb{S}^c}^c(\mathbf{X})) = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{X} \mathbf{R}^T (\mathbf{R} \mathbf{R}^T)^{-1} \mathbf{R}. \quad (7.2)$$

Matrix \mathbf{C} contains few columns indexed by set \mathbb{S}^c and matrix \mathbf{R} contains few rows indexed by set \mathbb{S}^r from the original matrix \mathbf{X} . A versatile metric for evaluating the performance of a data subset selection algorithm can be defined by the approximation error resulted from the projection of the entire data to the span of selected rows/columns. How close to the optimal selection an algorithm can reach, is determined by comparing its approximation error to the best low-rank approximation error specified by the spectral decomposition. Recently, we proposed a fast and accurate algorithm for solving CSSP which is called spectrum pursuit (SP) [165].

$$\mathbf{X} \cong \underbrace{\sum_{i=1}^2 \sum_{j=1}^3 u_{ij} \mathbf{c}_i \mathbf{r}_j^T}_{CUR} = \begin{matrix} u_{11} & + & u_{12} & + & u_{13} \\ \mathbf{c}_1 \mathbf{r}_1^T & & \mathbf{c}_1 \mathbf{r}_2^T & & \mathbf{c}_1 \mathbf{r}_3^T \\ + & u_{21} & + & u_{22} & + & u_{23} \\ \mathbf{c}_2 \mathbf{r}_1^T & & \mathbf{c}_2 \mathbf{r}_2^T & & \mathbf{c}_2 \mathbf{r}_3^T \end{matrix} = \hat{\mathbf{X}}$$

Figure 7.1: Two columns and three rows from matrix \mathbf{X} are selected and organized in matrix \mathbf{C} and \mathbf{R} . The outer product of each pair of a selected column and a selected row constructs a rank-1 matrix, i.e., $\mathbf{c}_i \mathbf{r}_j^T$. The contribution amount of each pair is reflected in variable u_{ij} . The core matrix \mathbf{U} is the collection of all u_{ij} 's. The goal is to minimize $\|\mathbf{X} - \hat{\mathbf{X}}\|_F$ where $\hat{\mathbf{X}} = \mathbf{CUR}$.

Inspired by SP, we propose a new algorithm to address the more general case of the CUR matrix decomposition. Extension to multi-way selection targets the following goals:

- A novel algorithm for CUR decomposition, referred to as two-way spectrum pursuit (TWSP), is proposed. TWSP provides an accurate solution for CUR decomposition.
- TWSP enjoys a linear complexity w.r.t. the number of columns and the number of rows of a matrix.
- TWSP is a parameter-free algorithm that only requires the number of desired columns and rows for selection. Thus, TWSP does not require any parameter fine-tuning.
- The TWSP algorithm is put to the test and investigated in a set of synthetic and real experiments.

- The role of the core matrix U in CUR decomposition is illustrated which shows the connection between selected columns and rows. Based on analysis of U , an interesting application for joint sensor/channel selection is presented.

Selecting the most diverse subset of data in an optimal sense is studied vastly [9, 26, 7]. However, these methods do not guarantee that the unselected columns are *well represented* by the selected ones. Further, outliers are selected with a high probability using such algorithms due to their diversity [165]. A more effective approach is selecting some *representatives* which are able to approximate the rest of data accurately [171] as defined as a special case of (7.1). This is an NP-hard problem [172] and there are several efforts for solving this problem [30, 9, 28, 3]. There are computationally expensive approaches based on convex relaxation [171, 173] that are not computationally feasible for large datasets since their complexity is of order $O(M^3)$, where M is the number of original columns. Recently, we proposed a new algorithm for solving CSSP with a linear complexity Which is called Spectrum pursuit (SP) [165]. The SP algorithm finds K columns of \mathbf{X} such that their span is close to that of the best rank- K approximation of \mathbf{X} . SP is an iterative approach where at each iteration one selected sample is optimized such that the ensemble of selected samples describes the whole dataset more accurately. SP finds representatives such that the column space is spanned accurately via consecutive rank-1 approximations. In this section, we extend the SP algorithm for selecting columns and rows jointly such that their outer product can represent the whole matrix accurately. A naive approach is applying SP algorithm on the matrix of interest to select a subset of columns and applying SP on its transpose in order to find a subset of rows. However, this approach is not efficient and we will compare it with our proposed approach which is optimized through a joint representation of selected columns and selected rows.

Two-way Spectrum Pursuit

The introduced joint column/row subset selection in (7.1) can be written as a CUR decomposition in the following form in which factor matrices must be drawn from actual columns/rows of the original matrix as

$$\begin{aligned} (C, U, R) &= \underset{C, U, R}{\operatorname{argmin}} \|X - CUR\|_F^2, \\ \text{s.t. } c_k &\in \mathbb{X}_c \text{ and } r_k \in \mathbb{X}_r. \end{aligned} \tag{7.3}$$

In this problem, \mathbb{X}_c , and \mathbb{X}_r indicate the set of normalized columns and rows of matrix X , respectively. Here, c_k and r_k^T denote the k^{th} column and the k^{th} row of C and R , respectively. In other words, $\mathbb{X}_c = \{X(:, m)/\|X(:, m)\|\}$ for all columns and $\mathbb{X}_r = \{X(n, :)/\|X(n, :)\|\}$ for all rows. Please note that replacing constraints in (7.3) with orthogonality constraint on c_k 's and on r_k 's results in the truncated singular value decomposition (SVD) with K most significant components. In this case, C and R contain the first K left singular vectors and the first K right singular vectors of X , respectively. Moreover, the core matrix will be diagonal and the entries will be singular values with diagonal entries as singular values. However, the underlying constraints in (7.3) turn the problem into a joint subset of row and column selection problem instead of matrix low-rank approximation problem.

To solve this complicated problem, we split it into two consecutive problems for optimization of the k^{th} selected column/row. Our optimization approach is alternative, i.e., a random subset of columns and rows are picked. Then, one column or row is considered to be replaced with a more efficient one at each iteration. Since scale of a vector does not change its span, without loss of generality assume that the column or the row subject of the optimization lie on the unit sphere. At

each iteration, a rank-1 component is optimized characterized by $\mathbf{c}\mathbf{g}^T$ or $\mathbf{h}\mathbf{r}^T$ given by

$$\underset{\mathbf{c}, \mathbf{W}, \mathbf{g}}{\operatorname{argmin}} \left\| \underbrace{\mathbf{X} - \mathbf{C}_k \mathbf{W} \mathbf{R}}_{\mathbf{E}^c} - \mathbf{c}\mathbf{g}^T \right\|_F^2 \text{ s.t. } \|\mathbf{c}\|_2 = 1 \quad (7.4a)$$

$$\underset{\mathbf{h}, \mathbf{Y}, \mathbf{r}}{\operatorname{argmin}} \left\| \underbrace{\mathbf{X} - \mathbf{C} \mathbf{Y} \mathbf{R}_k}_{\mathbf{E}^r} - \mathbf{h}\mathbf{r}^T \right\|_F^2 \text{ s.t. } \|\mathbf{r}\|_2 = 1 \quad (7.4b)$$

Matrix \mathbf{C}_k is the set of selected columns except the k^{th} one and \mathbf{R}_k is the set of selected rows except the k^{th} row. The first subproblem can be solved easily w.r.t. \mathbf{c} using singular value decomposition. In other words, $\mathbf{c}\mathbf{g}^T$ and $\mathbf{h}\mathbf{r}^T$ are the best rank-1 approximations of the residual \mathbf{E}^c and \mathbf{E}^r , respectively. The obtained \mathbf{c}/\mathbf{r} is the best column/row that can be added to the pool of selected columns/rows. However, the obtained vector is not available in the given dataset as a column or row since it is a singular vector which is a function of all columns/rows. The following step re-imposes the underlying constraints at each iteration,

$$\mathbb{S}_k^c = \underset{m}{\operatorname{argmax}} |\mathbf{x}_m^T \mathbf{c}|, \quad \forall \mathbf{x}_m \in \mathbb{X}_c \quad (7.5a)$$

$$\mathbb{S}_k^r = \underset{n}{\operatorname{argmax}} |\mathbf{x}_n^T \mathbf{r}|, \quad \forall \mathbf{x}_n \in \mathbb{X}_r \quad (7.5b)$$

Here, \mathbb{S}_k^c indicates a singleton that contains k^{th} selected column and \mathbb{S}_k^r corresponds to the k^{th} selected row. In each iteration, one column or one row is the subject of optimization. The impact of the latest estimation for that column/row on the representation is neglected. Then, an optimized replacement is found. At each iteration of TWSP, sub-problems in (7.4) are solved and their solutions are matched to the accessible column samples and row samples through matching equations in (7.5). The new selected column or row is stored in \mathbb{S}_k^c or \mathbb{S}_k^r , respectively. At each iteration we need to compute only the first singular vector and there are fast methods to do so [174]. The

pair of (7.4a) and (7.5a) optimizes and matches a column. Similarly, performing (7.4b) and (7.5b) provides us an optimized row. However, we do not update both of them in each iteration. In fact, we need to perform a column update or a row update in each iteration. It should be determined that which update (column or row) is more efficient in the current iteration. To this aim, first we choose a random previously selected column and a random previously selected row. Then, we find the best possible replacement column and the best possible replacement row. Accordingly, we choose whichever who minimizes the cost function more. The best modified column-wise subset is denoted by \tilde{S}^c and \tilde{S}^r denotes the best row-wise modified subset. Alg. 15 indicates the steps of TWSP algorithm. Here, \dagger refers to the Moore–Penrose pseudo-inverse operator. Iterations can be terminated either once CUR decomposition error is saturated or reaching a maximum number of iterations.

The proposed TWSP provides the selected columns and selected rows in order to form matrix C and R in CUR decomposition. It is straightforward to estimate the core matrix U . Mathematically,

$$U = C^\dagger X R^\dagger. \quad (7.6)$$

This matrix is a two-way compressed replica of the whole dataset and it contains valuable information in practice as will be discussed in Sec. . In general, the number of selected columns may differs from the desired number of rows. Here, K_1 refers to the number of columns and K_2 points to the number of rows. It is worthwhile to mention that the complexity order of TWSP is bottlenecked by computational burden for two pseudo-inverses and two singular vectors computation. Thus, the complexity can be expressed as $O(NK_1^2 + MK_2^2 + MN)$ per iteration and the algorithm needs $O(\max(K_1, K_2))$ iterations. In the next section, we evaluate the performance of our proposed algorithm.

The CSSP and CUR decomposition problems are NP-hard, i.e., finding an optimal solution re-

quires a combinatorial search to find the best columns and rows. The proposed TWSP algorithm minimizes the main cost function (7.1) in practice. However, there is no theoretical guarantee for convergence of the proposed TWSP algorithm Alg. 15. In order to improve the convergence behavior of TWSP, we evade updating both columns and rows in each iteration. Rather, we prioritize updating of a row or a column to the one which exhibits a smaller projection error and a better minimizer for the cost function per each iteration. The implementation steps of TWSP are summarized in Alg. 15.

N-way Spectrum Pursuit

Organization of high-volume data using a matrix and matrix subset selection are well-known solutions for efficient data representation and sampling. However, matrices are not suitable for more complex data. For example assume an fMRI dataset for a group of patients. Corresponding to each voxel (3D pixel) of brain there is an fMRI time-series. Thus, we measure a signal for each time slot of each voxel. Moreover, there are patients and each patient has independent measurements. Let denote the measurement at voxel (x, y, z) and time t for patient p using a function $f(x, y, z, t, p)$. This function is a 5-dimensional tensor. This organized and concise representation cannot be achieved using matrix-based representation. However, sampling from multi-dimensional tensors is not studied and it needs a deep insight to the problem. As the future direction of this proposal, multi-way data reduction problem will be introduced and solved efficiently.

The projection operator is denoted by $\pi_P(\mathbf{X})$ and defined as $\mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{X}$. This operator projects all columns of \mathbf{X} on the subspace spanned by columns of \mathbf{P} . For tensors, two projection operators are defined. 1) n-mode projection which is denoted by $\pi_P^{(n)}(\underline{\mathbf{X}})$ and 2) Coupled-

Algorithm 15 Two way spectrum pursuit (TWSP)

Require: $\mathbf{X} \in \mathbb{R}^{N \times M}$, K_1 and K_2 .

Output: \mathbb{S}^c and \mathbb{S}^r .

Initialization:

$\mathbb{S}^c \leftarrow$ A random subset of $\{1, \dots, M\}$ with $|\mathbb{S}^c| = K_1$

$\mathbb{S}^r \leftarrow$ A random subset of $\{1, \dots, N\}$ with $|\mathbb{S}^r| = K_2$

$\{\mathbb{S}_k^c\}_{k=1}^K \leftarrow$ Partition \mathbb{S}^c into K_1 subsets.

$\{\mathbb{S}_k^r\}_{k=1}^K \leftarrow$ Partition \mathbb{S}^r into K_2 subsets

$i = \text{rnd}(K_1)$ and $j = \text{rnd}(K_2)$

while a stopping criterion is not met

$\mathbb{S}_i^c = \mathbb{S}^c \setminus \mathbb{S}_i^c$

$\mathbf{C}_i \leftarrow$ remove column i in matrix \mathbf{C}

$\mathbf{W} = \mathbf{C}_i^\dagger \mathbf{X} \mathbf{R}^\dagger$

$\mathbf{E}^c = \mathbf{X} - \mathbf{C}_i \mathbf{W} \mathbf{R}$ (Null space projection)

$\mathbf{c} =$ find the first left singular vector of \mathbf{E}^c (7.4a)

$\mathbb{S}_i^c \leftarrow$ the most correlated column of \mathbf{E} with \mathbf{c} (7.5a)

$\tilde{\mathbb{S}}^c \leftarrow \bigcup_{i'=1}^{K_1} \mathbb{S}_{i'}^c$

$\mathbf{C} = \mathbf{X}(:, \tilde{\mathbb{S}}^c)$

$e^c = \min_U \|\mathbf{X} - \mathbf{C} \mathbf{U} \mathbf{R}\|_F$

$\mathbb{S}_j^r = \mathbb{S}^r \setminus \mathbb{S}_j^r$

$\mathbf{R}_j \leftarrow$ remove row j in matrix \mathbf{R}

$\mathbf{Y} = \mathbf{C}^\dagger \mathbf{X} \mathbf{R}_j^\dagger$

$\mathbf{E}^r = \mathbf{X} - \mathbf{C} \mathbf{Y} \mathbf{R}_j$ (Null space projection)

$\mathbf{r} =$ find the first right singular vector of \mathbf{E}^r (7.4b)

$\mathbb{S}_j^r \leftarrow$ the most correlated row of \mathbf{E} with \mathbf{r} (7.5b)

$\tilde{\mathbb{S}}^r \leftarrow \bigcup_{j'=1}^{K_2} \mathbb{S}_{j'}^r$

$\mathbf{R} = \mathbf{X}(\tilde{\mathbb{S}}^r, :)$

$e^r = \min_U \|\mathbf{X} - \mathbf{C} \mathbf{U} \mathbf{R}\|_F$

IF $e^c < e^r$

$\mathbb{S}^c \leftarrow \tilde{\mathbb{S}}^c$

$i = \text{rnd}(K_1)$

else

$\mathbb{S}^r \leftarrow \tilde{\mathbb{S}}^r$

$j = \text{rnd}(K_2)$

projection which is denoted by $\pi_{A,B,C}(\underline{\mathbf{X}})$. Mathematically,

$$\pi_P^{(n)}(\underline{\mathbf{X}}) = \underline{\mathbf{X}} \times_n \mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$$

and,

$$\pi_{A,B,C}(\underline{\mathbf{X}}) = \underset{\tilde{\underline{\mathbf{X}}}}{\operatorname{argmin}} \|\underline{\mathbf{X}} - \tilde{\underline{\mathbf{X}}}\|_F \text{ s.t. } \tilde{\underline{\mathbf{X}}} \in \Omega.$$

Set $\Omega \subset \mathbb{R}^{I \times J \times K}$ is a set of tensors that are linear combination of R tensors determined by $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ for $r = 1, \dots, R$. The coupled-projection can be computed by,

$$\operatorname{vec}(\pi_{A,B,C}(\underline{\mathbf{X}})) = \mathbf{P}(\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \operatorname{vec}(\underline{\mathbf{X}}).$$

In this equation the r^{th} column of \mathbf{P} is $\operatorname{vec}(\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r)$.

The final mission of the present dissertation is to design a tensor-based selection framework for reducing a multi-way structure of data into a set of fibers that reconstructs the original tensor. Two celebrated tensor decomposition algorithms are employed to this aim. Tucker decomposition and CP decomposition are extended to extract the principle fibers of the tensor.

Tucker decomposition and CP decomposition are powerful tools for tensors' analysis. However, they cannot be used directly for tensor fiber selection. The first step toward tensor subset selection is to define a concrete problem. It is straightforward to re-write Tucker decomposition in terms of factors matrices as the following optimization problem.

$$(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \underset{\mathbf{A}, \mathbf{B}, \mathbf{C}}{\operatorname{argmin}} \|\underline{\mathbf{X}} - \pi_C^{(3)}(\pi_B^{(2)}(\pi_A^{(1)}(\underline{\mathbf{X}})))\|_F^2 \quad (7.7)$$

Columns of factor matrices are optimized in order to minimize the cost function. Inspired by spectrum pursuit algorithm, we suggest the following problem in which columns of factor matrices must be drawn from the corresponding set of fibers from the original tensor. As our future work direction, we plan to extend Problem (7.7) in the same spirit of spectrum pursuit algorithm in order

to find the best subset fibers from different ways.

CP decomposition is considered as a special case of tucker decomposition. However, its simplicity, uniqueness conditions and straightforward interpretation make CP the most popular algorithm for decomposition of tensors. The self CP decomposition for multi-way selection is introduced as follows,

$$(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \underset{\mathbf{A}, \mathbf{B}, \mathbf{C}}{\operatorname{argmin}} \|\underline{\mathbf{X}} - \pi_{\mathbf{A}, \mathbf{B}, \mathbf{C}}(\underline{\mathbf{X}})\|_F^2 \quad (7.8)$$

$$\text{s.t. } \mathbf{A} \in \mathbb{X}_1, \mathbf{B} \in \mathbb{X}_2, \mathbf{C} \in \mathbb{X}_3. \quad (7.9)$$

Here, \mathbb{X}_n is the set of normalized fibers of tensor $\underline{\mathbf{X}}$ w.r.t. the n^{th} way. We plan to solve this problem efficiently and present interesting applications of multi-way data selection.

The implementation of the self-CP decomposition algorithm is summarized in Alg. 16. Matrix \mathbf{X}_j is the unfolded replica of tensor $\underline{\mathbf{X}}$ w.r.t. the j^{th} way. The m^{th} column of this matrix is denoted by \mathbf{x}_j^m .

Algorithm 16 Self-CP Decomposition Algorithm

Require: $\underline{\mathbf{X}}$, and K

Output: $\mathbb{S}_1, \mathbb{S}_2$, and \mathbb{S}_3

Initialize: $\mathbb{S}_1, \mathbb{S}_2$, and \mathbb{S}_3 with empty sets.

Initialize: $\underline{\mathbf{E}}$ by $\underline{\mathbf{X}}$

FOR: $k = 1, \dots, K$

$[\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3] \leftarrow \text{CPD on } \underline{\mathbf{E}} \text{ with Rank } 1$

FOR: $j = 1 : 3$

$\mathbb{S}_j \leftarrow \mathbb{S}_j \cup \operatorname{argmax}_m \langle \mathbf{x}_j^m, \mathbf{u}^j \rangle$

END FOR

$\underline{\mathbf{E}} \leftarrow \underline{\mathbf{E}} - \pi_{\mathbf{u}^1, \mathbf{u}^2, \mathbf{u}^3}(\underline{\mathbf{E}})$

END FOR

Experimental Results

In order to evaluate TWSP in terms of CUR decomposition accuracy or its performance on a machine-learning task, we apply the proposed TWSP on synthetic data as well as three real applications.

CUR Decomposition on Synthetic Data

In order to evaluate the general performance of TWSP, we compared it with the state-of-the-art methods for selecting columns and rows. In this regards, we created a 1000×2000 synthetic dataset. The dataset is generated by a rank-30 matrix contaminated with random noise. In Fig. 2, we have illustrated the CUR error for selecting a subset of rows and columns in the range of 2 to 20. The reconstruction error of CUR is normalized by $\|\mathbf{X}\|_F^2$. We employ SP as the state-of-the-art algorithm for column subset selection [165]. We perform SP on the data matrix \mathbf{X} and its transpose in order to, respectively, select a subset of columns and rows independently. Then, employing (7.6) results in a CUR decomposition. We refer to the algorithm in [168] as adaptive CUR. A more accurate algorithm for solving CUR decomposition results in a bigger blue region in Fig. 2. TWSP exhibits the best performance in this experiment. The convergence behavior of TWSP for this experiment is shown in Fig. 2 (d) for selecting 20 columns and 20 rows. The final solution of the TWSP algorithm depends on the initial selected columns and selected rows. However, regardless of the initial condition, the algorithm minimizes the cost function of CUR decomposition.

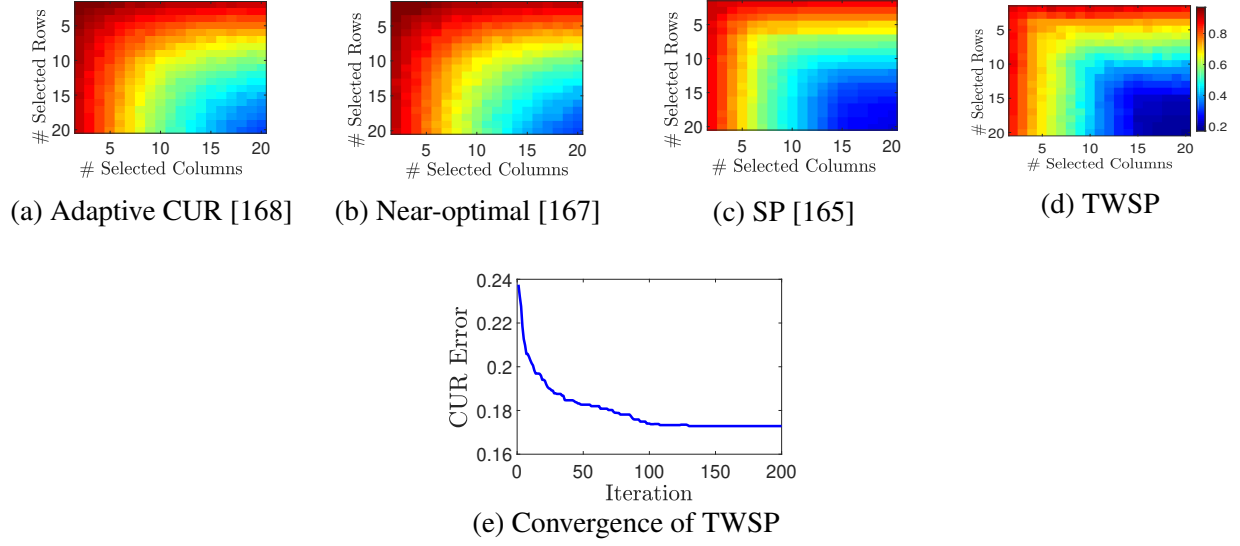


Figure 7.2: (a)-(d) Performance comparison in terms of the normalized error of CUR decomposition. (e) Convergence behavior of the proposed two-way SP for selecting 20 columns and 20 rows.

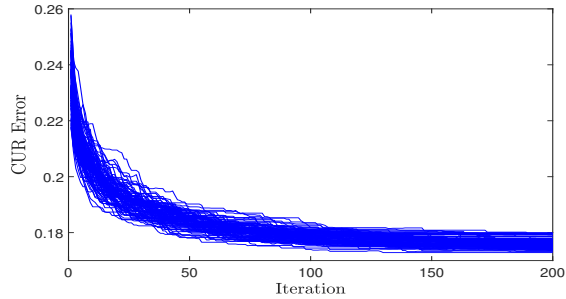


Figure 7.3: The behavior of the proposed algorithm w.r.t. the initial condition of selected subset. The initial cost function are corresponding to the initial set which are drawn randomly. The blue curves indicate the path of optimization alongside iterations of the TWSP algorithm. Here, 100 different realizations are studied.

Joint Sensor Selection and Channel Assignment

The output products of CUR decomposition are not limited to a subset of columns and rows. In some applications, interestingly, matrix \mathbf{U} is the most important output of a CUR decomposition. Entry (i, j) in \mathbf{U} indicates how important the cooperation of the i^{th} column and the j^{th} row is. This

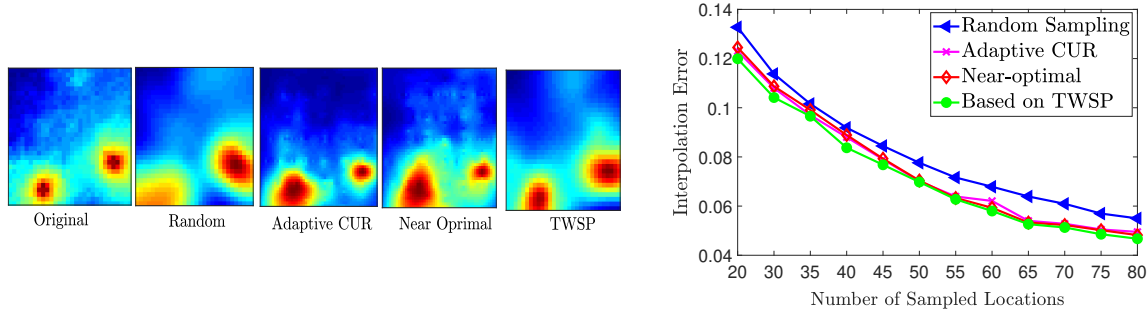


Figure 7.4: The original spectrum map and its comparison with the interpolated map using sampled random sampling and our proposed method. The interpolation error is depicted versus number of sensed locations.

interesting property is utilized for the problem of joint sensor selection and channel assignment in a cognitive radio network. To this aim the exact setup in [1] is considered with 900 grid points and 32 frequency channels. The received power magnitudes are organized in a 900×32 matrix. The only difference here, is that the uniform sampling pattern of sensing is replaced by the selection based on the CUR decomposition. Our proposed TWSP algorithm provides a fast and accurate solution for CUR decomposition. We select between 20 and 80 locations for spectrum sensing and all 32 channels. Each row of matrix U corresponds to a selected location and it has 32 entries. We are to assign F channels for each selected sensor. In other words, each location does not sense the whole spectrum and only F frequency channels are assigned for each sensor. The top- F entries in each row with the highest absolute value show the most important channels for the corresponding location to be sensed.

Fig. 7.4 shows the cartography error of spectrum sensing for the conventional random selection as introduced in [1] and our proposed optimized joint sensors and channels. For each sampled locations $F = 8$ channels out of 32 channels are sensed. The sampled spectrum map is interpolated using thin plane splines method [175] for both sampling methods. In addition to visual superiority in reconstruction of the spectrum map, channel assignment based on TWSP provides a better quantitative error.

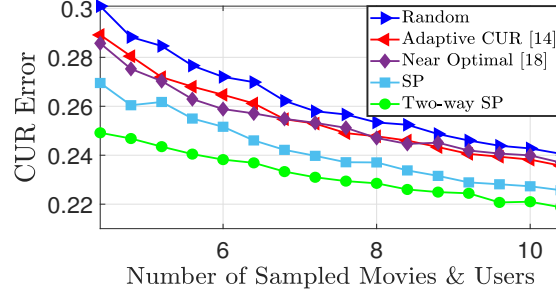


Figure 7.5: Comparison of the normalized prediction error with state-of-the-art algorithms obtained by CUR decomposition for simultaneous movies and users subset selection from Netflix dataset.

Informative Users/Contents Detection

Another problem gaining a lot of interest by streaming services providers is choosing a set of users for reflecting their feedbacks about different products. Therefore, it is crucial for such companies to find a subset of users and media products that reviews of those users for those specific products can leverage the most information about other users' unknown behavior. Each person has a limited scope of interest. For example, a user who only loves romance and action movie genres corresponds to a specific personality that is able to represents a cluster of users accurately. Moreover, his reviews for his area of interest are more valuable, not reviews for all genres.

As a result, there exist a demand for a reliable algorithm to simultaneously choose the most informative subset of users and movies. Such a subset is desirable for streaming companies to the extent that they are willing to give the users incentives to leave comprehensive reviews for those specific products. In this regards, we have evaluated our algorithm on Netflix Prize dataset containing 17,770 movies and 480,189 users. We have reduced the dataset to 990 movies and 4,727 users by considering only movies and users with most reviews. Then, we completed the dataset by Lin et al. method to have a ground truth [176]. Fig. 7.5 reveals that TWSP shows the best performance in terms of predicting scores for all users/movies based on a few selected users/movies.

Supervised Sampling

The proposed TWSP algorithm is an unsupervised data selection algorithm. In the presence of labels for our data, a naive approach is to select representatives from each class independently. However, considering both classes jointly is a more efficient way for data reduction. Assume we are given two classes data as $\mathbf{X}_1 \in \mathbb{R}^{N \times M_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{N \times M_2}$. The goal is to select K_1 samples from Class 1 and K_2 samples from Class 2. To this aim, we propose to construct the cross correlation of two classes as a kernel representation for both classes jointly. Matrix $\mathbf{X} = \mathbf{X}_2^T \mathbf{X}_1$ which has M_1 columns and M_2 rows is fed to TWSP algorithm in order to select K_1 columns and K_2 rows optimally.

Supervised sampling is performed on Kaggle cats and dogs dataset. The features are obtained by a trained Resnet-18 deep learning model [177]. Three mutually exclusive data subsets for training, validation, and testing are partitioned randomly from 2000 images of each class. The classification accuracy of 97.5% is achieved from a fine tuned Resnet-18 using the whole training set containing 1000 samples for each class. Afterwards, samples are selected by applying TWSP on the kernel feature matrix and the Resnet-18 is fine-tuned by using the sampled data. The model's accuracy is compared on the testing set with other sampling methods. Fig. 7.6 shows the performance of selection algorithms for different numbers of representatives per class. Using only two samples from each class, a classification accuracy of 82.3% can be achieved which is more than 15% improvement compared to random selection and more than 5% improvement compared to other competitors.

We have conducted further experiments to study the effectiveness of the proposed algorithm in the multi-class image classification problem. For this study, we use the Resnet-34 deep learning model pre-trained on CIFAR10 and trained on a subset of ImageNet Dataset comprising of 10 classes. The classes used for this experiment are Tench, Goldfish, Great white shark, Tiger shark, Hammerhead,

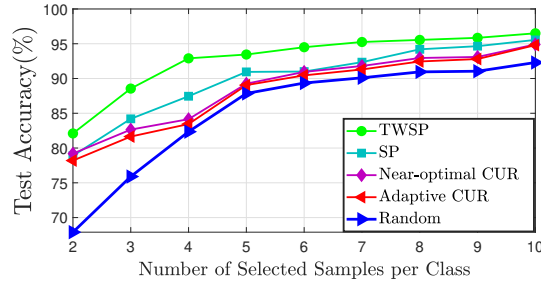


Figure 7.6: The classification accuracy of a fine tuned Resnet-18 network using a few selected data per each class.

Electric ray, Stingray, Cock, Hen and Ostrich. The original training data consists of 1300 images of each class, and the idea is to use Two-Way SP to select the data samples such that K samples of each class are used for the training purposes. After training Resnet-34 for 10 classes of CIFAR10, the feature vectors of all the training images are extracted such that a 1300×512 matrix is obtained for each class. Since, TWSP is applicable for the two-class problem, we employ the one-versus-all approach. In other words, a cross correlation matrix is generated for each class such that \mathbf{X}_1 has the dimensions 512×1300 and \mathbf{X}_2 has the dimensions 512×11700 where \mathbf{X}_1 represents the feature vector of the class for which samples will be selected while \mathbf{X}_2 represents the feature vectors of the remaining 9 classes. Hence, the number of cross correlation matrices is equal to the number of the classes and K samples are selected separately by applying TWSP on each cross correlation matrix. This step of generating convoluted matrices is different from binary classification where only one convoluted matrix was formed. Pre-trained Resnet-34 on CIFAR10 has been trained again by using the sampled-data but with the same parameters. The model's accuracy is subsequently compared on the validation set with other sampling methods.

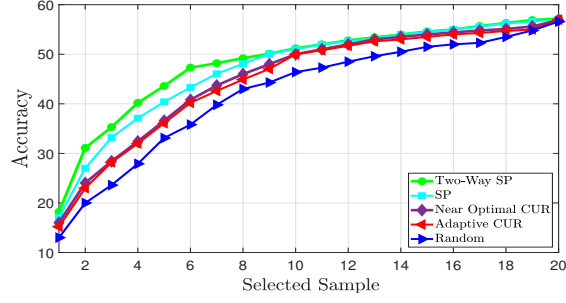


Figure 7.7: Test accuracy in terms of improvement comparison with the test accuracy obtained by random selection.

Experiments on Self-CP Decomposition

A simple extension of two-way spectrum pursuit is proposed as the general concept of Self-CP decomposition. In order to evaluate the performance of our proposed Self-CP decomposition, a synthetic tensor is generated which is the summation of a low-rank tensor plus a full-rank noise tensor. The dimension of tensor is considered as $30 \times 40 \times 50$ and the rank of the low-rank part is assumed as 5. From each way of the tensor, K fibers are selected. Fig. 7.8 shows the normalized reconstruction error versus the number of selected fibers from each way. CP decomposition provides us a set of vectors which are not actual fibers. The performance of rank- K CPD also is indicated as a lower bound for tensor reconstruction based on fiber selection. As it can be seen, a significant performance is achieved using the proposed method comparison with random selection of fibers. Assuming $K = 0$, results in a zero tensor as the estimation based on the selected fibers and the normalized error will be 1. As we select more fibers or increase the rank of CP, the normalized error decreases.

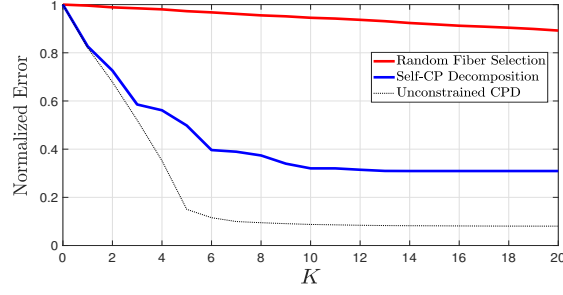


Figure 7.8: Test accuracy in terms of improvement comparison with the test accuracy obtained by random selection.

Conclusion

Two-way spectrum pursuit is proposed as an accurate and efficient algorithm for solving CUR decomposition. Some applications of the proposed algorithm are presented. However, they are not limited to the mentioned applications. Moreover, the proposed algorithm can be extended to n -way spectrum pursuit for efficient tensor subset selection.

LIST OF REFERENCES

- [1] G. Zhang, X. Fu, J. Wang, X.-L. Zhao, and M. Hong, “Spectrum cartography via coupled block-term tensor decomposition,” *IEEE Transactions on Signal Processing*, 2020.
- [2] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [3] A. Zaeemzadeh, M. Joneidi, N. Rahnavard, and M. Shah, “Iterative Projection and Matching: Finding Structure-Preserving Representatives and Its Application to Computer Vision,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5414–5423. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Zaeemzadeh_Iterative_Projection_and_Matching_Finding_Structure-Preserving_Representatives_and_Its_Application_CVPR_2019_paper.html
- [4] E. Elhamifar, G. Sapiro, and S. S. Sastry, “Dissimilarity based sparse subset selection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2182–2197, 2016.
- [5] C. You, C. Li, D. P. Robinson, and R. Vidal, “Scalable exemplar-based subspace clustering on class-imbalanced data,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–83.
- [6] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1600–1607.
- [7] M. Joneidi, A. Zaeemzadeh, B. Shahrabi, G.-J. Qi, and N. Rahnavard, “E-optimal Sensor Selection for Compressive Sensing-based Purposes,” *IEEE Transactions on Big Data*, p. To Appear in, 2018.
- [8] P. A. Vijaya, M. N. Murty, and D. K. Subramanian, “Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets,” *Pattern Recognition Letters*, vol. 25, no. 4, pp. 505–513, 2004.

- [9] A. Deshpande and L. Rademacher, "Efficient volume sampling for row/column subset selection," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010, pp. 329–338.
- [10] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas, "CR-GAN: Learning Complete Representations for Multi-view Generation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 942–948. [Online]. Available: <https://www.ijcai.org/proceedings/2018/131>
- [11] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [12] D. Reinsel, J. Gantz, and J. Rydning, "The digitization of the world from edge to core,(november)," 2018.
- [13] X. Jin and J. Han, "K-medoids clustering," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 564–565.
- [14] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.
- [15] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *Signal Processing, IEEE Transactions on*, vol. 57, no. 2, pp. 451–462, Feb 2009.
- [16] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 2572–2577.
- [17] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [18] A. Çivril, "Column subset selection problem is ug-hard," *Journal of Computer and System Sciences*, vol. 80, no. 4, pp. 849–859, 2014.

- [19] C. Boutsidis, M. W. Mahoney, and P. Drineas, “An improved approximation algorithm for the column subset selection problem,” in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2009, pp. 968–977.
- [20] Y. Shitov, “Column subset selection is np-complete,” *arXiv preprint arXiv:1701.02764*, 2017.
- [21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions - I,” *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.
- [22] T. F. Chan and P. C. Hansen, “Some applications of the rank revealing qr factorization,” *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 3, pp. 727–741, 1992.
- [23] T. F. Chan, “Rank revealing qr factorizations,” *Linear algebra and its applications*, vol. 88, pp. 67–82, 1987.
- [24] J. Xiao, M. Gu, and J. Langou, “Fast parallel randomized qr with column pivoting algorithms for reliable low-rank matrix approximations,” in *2017 IEEE 24th International Conference on High Performance Computing (HiPC)*. IEEE, 2017, pp. 233–242.
- [25] V. Guruswami and A. K. Sinop, “Optimal column-based low-rank matrix reconstruction,” in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012, pp. 1207–1214.
- [26] C. Li, S. Jegelka, and S. Sra, “Polynomial time algorithms for dual volume sampling,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5038–5047.
- [27] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, “Matrix approximation and projective clustering via volume sampling,” in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 1117–1126.
- [28] S. Paul, M. Magdon-Ismail, and P. Drineas, “Column selection via adaptive sampling,” in *Advances in neural information processing systems*, 2015, pp. 406–414.

- [29] Y. Wang and A. Singh, “Provably correct algorithms for matrix column subset selection with selectively sampled data,” *Journal of Machine Learning Research*, vol. 18, pp. 1–42, 2018.
- [30] C. Boutsidis, P. Drineas, and M. Magdon-Ismail, “Near-optimal column-based matrix reconstruction,” *SIAM Journal on Computing*, vol. 43, no. 2, pp. 687–717, 2014.
- [31] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.
- [32] M. Derezhinski, M. K. Warmuth, and D. J. Hsu, “Leveraged volume sampling for linear regression,” in *Advances in Neural Information Processing Systems*, 2018, pp. 2505–2514.
- [33] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, “From Keyframes to Key Objects: Video Summarization by Representative Object Proposal Selection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2016, pp. 1039–1048. [Online]. Available: <http://ieeexplore.ieee.org/document/7780487/>
- [34] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, “Blind parafac receivers for ds-cdma systems,” *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823, 2000.
- [35] M. N. da Costa, G. Favier, and J. M. T. Romano, “Tensor modelling of mimo communication systems with performance analysis and kronecker receivers,” *Signal Processing*, vol. 145, pp. 304–316, 2018.
- [36] M. Boutalline, I. Badi, B. Bouikhalene, and S. Safi, “Blind identification and equalization of cdma signals using the levenberg-marquardt algorithm,” *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, no. 7, pp. 1270–1275, 2015.
- [37] M. Sørensen, F. Van Eeghem, and L. De Lathauwer, “Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions,” *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4132–4145, 2017.

- [38] Y. Lin, S. Jin, M. Matthaiou, and X. You, “Tensor-based channel estimation for millimeter wave mimo-ofdm with dual-wideband effects,” *IEEE Transactions on Communications*, 2020.
- [39] G. Favier, M. N. Da Costa, A. L. De Almeida, and J. M. T. Romano, “Tensor space–time (tst) coding for mimo wireless communication systems,” *Signal Processing*, vol. 92, no. 4, pp. 1079–1092, 2012.
- [40] X. Ding, W. Chen, and I. J. Wassell, “Joint sensing matrix and sparsifying dictionary optimization for tensor compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3632–3646, 2017.
- [41] Q. Wang, M. Wei, X. Chen, and Z. Miao, “Joint encryption and compression of 3d images based on tensor compressive sensing with non-autonomous 3d chaotic system,” *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 1715–1734, 2018.
- [42] N. Sidiropoulos and A. Kyrillidis, “Multi-way compressed sensing for sparse low-rank tensors,” *Signal Processing Letters, IEEE*, vol. 19, no. 11, pp. 757–760, Nov 2012.
- [43] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [44] J. Bazerque and G. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, March 2010.
- [45] N. D. Sidiropoulos and A. Kyrillidis, “Multi-way compressed sensing for sparse low-rank tensors,” *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 757–760, 2012.
- [46] A. Bhaskara, M. Charikar, and A. Vijayaraghavan, “Uniqueness of tensor decompositions with applications to polynomial identifiability,” in *Conference on Learning Theory*, 2014, pp. 742–778.
- [47] M. Derezhinski and M. Warmuth, “Subsampling for Ridge Regression via Regularized Volume Sampling,” in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 716–725.

- [48] H. Jamali-Rad, H. Ramezani, and G. Leus, “Sparsity-aware multi-source $\{\text{RSS}\}$ localization,” *Signal Processing*, vol. 101, no. 0, pp. 174 – 191, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168414000802>
- [49] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, Feb 2009.
- [50] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel, “Greedy column subset selection for large-scale data sets,” *Knowledge and Information Systems*, vol. 45, no. 1, pp. 1–34, 2015.
- [51] M. Joneidi, A. Zaeemzadeh, and N. Rahnavard, “Dynamic Sensor Selection for Reliable Spectrum Sensing via E-optimal Criterion,” in *2017 IEEE 14th International Conference on Mobile Adhoc and Sensor Systems*. Orlando: IEEE Computer Society, 2017.
- [52] A. Nikolov, M. Singh, and U. T. Tantipongpipat, “Proportional Volume Sampling and Approximation Algorithms for A-Optimal Design,” *arXiv preprint arXiv:1802.08318*, 2018.
- [53] Z. E. Mariet and S. Sra, “Elementary symmetric polynomials for optimal experimental design,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2139–2148.
- [54] A. N. Dolia, S. F. Page, N. M. White, and C. J. Harris, “D-optimality for minimum volume ellipsoid with outliers,” in *Proc. 7th Int. Conf. Signal/Image Processing Pattern Recognition*. Citeseer, 2004, pp. 73–76.
- [55] H. Liu, Y. Liu, and F. Sun, “Robust Exemplar Extraction Using Structured Sparse Coding,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1816–1821, 8 2015. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909024>
- [56] O. H. Toma, M. López-Benítez, D. K. Patel, and K. Umebayashi, “Estimation of primary channel activity statistics in cognitive radio based on imperfect spectrum sensing,” *IEEE Transactions on Communications*, 2020.

- [57] W. Zhang, C.-X. Wang, X. Ge, and Y. Chen, “Enhanced 5g cognitive radio networks based on spectrum sharing and spectrum aggregation,” *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6304–6316, 2018.
- [58] W. Ahmad, N. Radzi, F. Samidi, A. Ismail, F. Abdullah, M. Jamaludin, and M. Zakaria, “5g technology: Towards dynamic spectrum sharing using cognitive radio networks,” *IEEE Access*, vol. 8, pp. 14 460–14 488, 2020.
- [59] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, May 2006.
- [60] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [61] A. Zaeemzadeh, M. Joneidi, N. Rahnavard, and G. Qi, “Co-spot: Cooperative spectrum opportunity detection using bayesian clustering in spectrum-heterogeneous cognitive radio networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 206–219, June 2018.
- [62] N. Muchandi and R. Khanai, “Cognitive radio spectrum sensing: A survey,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE, 2016, pp. 3233–3237.
- [63] Y. Chen, S. Su, H. Yin, X. Guo, Z. Zuo, J. Wei, and L. Zhang, “Optimized non-cooperative spectrum sensing algorithm in cognitive wireless sensor networks,” *Sensors*, vol. 19, no. 9, p. 2174, 2019.
- [64] B. Hamdaoui, B. Khalfi, and M. Guizani, “Compressed wideband spectrum sensing: Concept, challenges, and enablers,” *IEEE Communications Magazine*, vol. 56, no. 4, pp. 136–141, 2018.
- [65] S. Kapoor and G. Singh, “Non-cooperative spectrum sensing: A hybrid model approach,” in *2011 International Conference on Devices and Communications (ICDeCom)*, Feb 2011, pp. 1–5.

- [66] D. Cohen, A. Akiva, B. Avraham, and Y. C. Eldar, “Centralized cooperative spectrum sensing from sub-nyquist samples for cognitive radios,” in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 7486–7491.
- [67] J. Wei and X. Zhang, “Energy-efficient distributed spectrum sensing for wireless cognitive radio networks,” in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, Mar 2010, pp. 1–6.
- [68] H. Yazdani and A. Vosoughi, “On cognitive radio systems with directional antennas and imperfect spectrum sensing,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3589–3593.
- [69] Y. Luo, J. Dang, and Z. Song, “Optimal compressive spectrum sensing based on sparsity order estimation in wideband cognitive radios,” *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.
- [70] E. Dall’Anese, J. A. Bazerque, and G. B. Giannakis, “Group sparse lasso for cognitive network sensing robust to model uncertainties and outliers,” *Physical Communication*, vol. 5, no. 2, pp. 161 – 172, 2012, compressive Sensing in Communications.
- [71] X. Fu, N. D. Sidiropoulos, and W.-K. Ma, “Power spectra separation via structured matrix factorization.” *IEEE Trans. Signal Processing*, vol. 64, no. 17, pp. 4592–4605, 2016.
- [72] J. D. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
- [73] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [74] M. Joneidi, H. Yazdani, A. Vosoughi, and N. Rahnavard, “Source localization and tracking for dynamic radio cartography using directional antennas,” in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–9.

- [75] A. Esmaeili and F. Marvasti, "A novel approach to quantized matrix completion using huber loss measure," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 337–341, 2019.
- [76] A. Zaeemzadeh, M. Joneidi, B. Shahrashbi, and N. Rahnavard, "Robust target localization based on squared range iterative reweighted least squares," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 380–388.
- [77] M. Azghani, A. Esmaeili, K. Behdin, and F. Marvasti, "Missing low-rank and sparse decomposition based on smoothed nuclear norm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1550–1558, 2019.
- [78] R. B. Cattell, "The three basic factor-analytic research designs—their interrelations and derivatives," *Psychological bulletin*, vol. 49, no. 5, p. 499, 1952.
- [79] C. J. Appellof and E. R. Davidson, "Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents," *Analytical Chemistry*, vol. 53, no. 13, pp. 2053–2056, 1981.
- [80] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.
- [81] X. Guo, X. Huang, L. Zhang, L. Zhang, and A. ó. A. Plaza, "Support tensor machines for classification of hyperspectral remote sensing imagery," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3248–3264, 2016.
- [82] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova, "Clustering attributed graphs: models, measures and methods," *Network Science*, vol. 3, no. 3, pp. 408–444, 2015.
- [83] X. Fu, N. D. Sidiropoulos, and W.-K. Ma, "Tensor-based power spectra separation and emitter localization for cognitive radio," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2014, pp. 421–424.

- [84] X. Fu, N. D. Sidiropoulos, J. H. Tranter, and W. K. Ma, "A factor analysis framework for power spectra separation and multiple emitter localization," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6581–6594, Dec 2015.
- [85] T. Getu, W. Ajib, G. Kaddoum *et al.*, "Toward overcoming a hidden terminal problem arising in mimo cognitive radio networks: A tensor-based spectrum sensing algorithm," *IEEE Transactions on Vehicular Technology*, 2019.
- [86] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3005–3009, 2019.
- [87] J. Tian, P. Cheng, Z. Chen, M. Li, H. Hu, Y. Li, and B. Vucetic, "A machine learning-enabled spectrum sensing method for ofdm systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2019.
- [88] P. Comon, "Tensors: a brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, 2014.
- [89] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
- [90] S. Friedland and L.-H. Lim, "Nuclear norm of higher-order tensors," *Mathematics of Computation*, vol. 87, no. 311, pp. 1255–1281, 2018.
- [91] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 23, no. 7-8, pp. 393–405, 2009.
- [92] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Group-lasso on splines for spectrum cartography," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4648–4663, 2011.
- [93] G. Wahba, *Spline models for observational data*. SIAM, 1990.

- [94] A. M. Al-Samman, T. A. Rahman, M. Hindia, A. Daho, and E. Hanafi, "Path loss model for outdoor parking environments at 28 ghz and 38 ghz for 5g wireless networks," *Symmetry*, vol. 10, no. 12, p. 672, 2018.
- [95] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1997.
- [96] A. Uschmajew, "Local convergence of the alternating least squares algorithm for canonical tensor approximation," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 2, pp. 639–652, 2012.
- [97] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [98] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2547–2560, 2017.
- [99] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 375–391, 2010.
- [100] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, 2013.
- [101] B. Amizic, L. Spinoulas, R. Molina, and A. Katsaggelos, "Compressive blind image deconvolution," *Image Processing, IEEE Transactions on*, vol. 22, no. 10, pp. 3994–4006, Oct 2013.
- [102] X. Guan, Y. Gao, J. Chang, and Z. Zhang, "Advances in theory of compressive sensing and applications in communication," in *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*, Oct 2011, pp. 662–665.
- [103] Z. Yu, S. Hoyos, and B. M. Sadler, "Mixed-signal parallel compressed sensing and reception for cognitive radio," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3861–3864.

- [104] Y. Gwon, H. Kung, and D. Vlah, "Compressive sensing with optimal sparsifying basis and applications in spectrum sensing," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 5386–5391.
- [105] T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *Information Theory, IEEE Transactions on*, vol. 57, no. 7, pp. 4680–4688, July 2011.
- [106] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ_0 norm," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 289–301, Jan 2009.
- [107] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, vol. 20, pp. 33–61, 1998.
- [108] Q. Geng and J. Wright, "On the local correctness of ℓ_1 minimization for dictionary learning," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, 6 2014, pp. 3180–3184.
- [109] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, 2001.
- [110] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *J. Amer. Math. Soc.*, pp. 211–231, 2009.
- [111] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [112] E. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.
- [113] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathématique*, vol. 346, no. 9, pp. 589 – 592, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1631073X08000964>

- [114] M. Davies and R. Gribonval, “Restricted isometry constants where ℓ_p sparse recovery can fail for $0 < p \leq 1$,” *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2203–2214, May 2009.
- [115] J. D. Blanchard, C. Cartis, and J. Tanner, “Compressed Sensing: How Sharp Is the Restricted Isometry Property?” *SIAM Rev.*, vol. 53, no. 1, pp. 105–125, 2 2011. [Online]. Available: <http://dx.doi.org/10.1137/090748160>
- [116] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, “A unified framework for representation-based subspace clustering of out-of-sample and large-scale data,” *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2499–2512, 2016.
- [117] T. Sajana, C. M. S. Rani, and K. V. Narayana, “A survey on clustering techniques for big data mining,” *Indian Journal of Science and Technology*, vol. 9, no. 3, 2016.
- [118] W. Song, Z. Deng, L. Wang, B. Du, P. Liu, and K. Lu, “G-IK-SVD: parallel IK-SVD on GPUs for sparse representation of spatial big data,” *The Journal of Supercomputing*, pp. 1–18, 2016.
- [119] R. Zhang and J. T. Kwok, “Asynchronous Distributed ADMM for Consensus Optimization.” in *ICML*, 2014, pp. 1701–1709.
- [120] S. Scardapane, D. Wang, and M. Panella, “A decentralized training algorithm for echo state networks in distributed big data applications,” *Neural Networks*, vol. 78, pp. 65–74, 2016.
- [121] D. Hajinezhad, T.-H. Chang, X. Wang, Q. Shi, and M. Hong, “Nonnegative matrix factorization using ADMM: Algorithm and convergence analysis,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4742–4746.
- [122] P. Bühlmann, P. Drineas, M. Kane, and M. van der Laan, *Handbook of Big Data*. Chapman and Hall/CRC, 2016.
- [123] H. Jamali-Rad, A. Simonetto, and G. Leus, “Sparsity-aware sensor selection: Centralized and distributed algorithms,” *Signal Processing Letters, IEEE*, vol. 21, no. 2, pp. 217–220, Feb 2014.

- [124] G. Nemhauser and L. Wolsey, “Best algorithms for approximating the maximum of a submodular set function,” Universit?? catholique de Louvain, Center for Operations Research and Econometrics (CORE), CORE Discussion Papers RP 343. [Online]. Available: <http://EconPapers.repec.org/RePEc:cor:louvrp:-343>
- [125] M. Gu and S. C. Eisenstat, “Efficient algorithms for computing a strong rank-revealing qr factorization,” *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.
- [126] C. Boutsidis, M. W. Mahoney, and P. Drineas, “An improved approximation algorithm for the column subset selection problem,” in *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’09, Society for Industrial and Applied Mathematics. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009, pp. 968–977. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1496770.1496875>
- [127] P. Van Dooren, “Numerical Linear Algebra for Signal, Systems and Control,” *Draft notes prepared for the Graduate School in Systems and Control*, vol. 250, 2003.
- [128] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [129] A. Esmaeili and F. Marvasti, “Comparison of several sparse recovery methods for low rank matrices with random samples,” in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 191–195.
- [130] M. Joneidi, A. Zaeemzadeh, N. Rahnavard, and M. B. Khalilsarai, “Matrix coherency graph: A tool for improving sparse coding performance,” in *2015 International Conference on Sampling Theory and Applications (SampTA)*. IEEE, 2015, pp. 168–172.
- [131] A. Tillmann and M. Pfetsch, “The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 60, no. 2, pp. 1248–1259, Feb 2014.

- [132] C. C. Aggarwal, Y. Xie, and P. S. Yu, “On dynamic data-driven selection of sensor streams,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1226–1234.
- [133] G.-J. Qi, C. Aggarwal, D. Turaga, D. Sow, and P. Anno, “State-Driven Dynamic Sensor Selection and Prediction with State-Stacked Sparseness,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’15. New York, NY, USA: ACM, 2015, pp. 945–954. [Online]. Available: <http://doi.acm.org/10.1145/2783258.2783390>
- [134] M. Holmes, A. Gray, and C. Isbell, “Fast SVD for large-scale matrices,” in *Workshop on Efficient Machine Learning at NIPS*, vol. 58, 2007, pp. 249–252.
- [135] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.
- [136] S. V. Tenneti and P. P. Vaidyanathan, “A Unified Theory of Union of Subspaces Representations for Period Estimation,” *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5217–5231, 2016.
- [137] D. Wang, C. Ding, and T. Li, “K-subspace clustering,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 506–521.
- [138] J. Bazerque and G. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1847–1862, March 2010.
- [139] R. Chartrand and W. Yin, “Iteratively reweighted algorithms for compressive sensing,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 3869–3872.
- [140] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial intelligence*, vol. 97, no. 1, pp. 245–271, 1997.

- [141] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for K-medoids clustering,” *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.
- [142] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, “Clustering appearances of objects under varying illumination conditions,” in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, vol. 1. IEEE, 2003, pp. I–11.
- [143] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.
- [144] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.
- [145] K.-C. Lee, J. Ho, and D. J. Kriegman, “Acquiring linear subspaces for face recognition under variable lighting,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 684–698, 2005.
- [146] Y. LeCun and C. Cortes, “{MNIST} handwritten digit database,” 2010. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [147] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3859–3869.
- [148] S. Tokui, K. Oono, S. Hido, and J. Clayton, “Chainer: a Next-Generation Open Source Framework for Deep Learning,” in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015. [Online]. Available: http://learningsys.org/papers/LearningSys_2015_paper_33.pdf
- [149] A. Esmaeili, E. A. Kangarshahi, and F. Marvasti, “Iterative null space projection method with adaptive thresholding in sparse signal recovery,” *IET Signal Processing*, vol. 12, no. 5, pp. 605–612, 2018.

- [150] M. Sadeghi, M. Joneidi, M. Babaie-Zadeh, and C. Jutten, "Sequential subspace finding: a new algorithm for learning low-dimensional linear subspaces," in *21st European Signal Processing Conference (EUSIPCO 2013)*. IEEE, 2013, pp. 1–5.
- [151] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proceedings of the IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990. [Online]. Available: <http://ieeexplore.ieee.org/document/58320/>
- [152] P. A. Vijaya, M. N. Murty, and D. K. Subramanian, "Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 505–513, 2004.
- [153] D. K. Saxena and K. Deb, "Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 772–787.
- [154] M.-H. Yang, "Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods." in *Fgr*, vol. 2, 2002, p. 215.
- [155] M. Sedghi, G. Atia, and M. Georgiopoulos, "A multi-criteria approach for fast and outlier-aware representative selection from manifolds," *arXiv preprint arXiv:2003.05989*, 2020.
- [156] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou, "Spectral clustering on multiple manifolds," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1149–1161, 2011.
- [157] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 5 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0262885609001711>
- [158] K. Cao, Y. Rong, C. Li, X. Tang, and C. Change Loy, "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

- [159] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “MS-Celeb-1M: Challenge of Recognizing One Million Celebrities in the Real World,” *Electronic Imaging*, vol. 2016, no. 11, pp. 1–6, 2016. [Online]. Available: <http://www.ingentaconnect.com/content/10.2352/ISSN.2470-1173.2016.11.IMAWM-463>
- [160] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2009, pp. 248–255. [Online]. Available: <http://ieeexplore.ieee.org/document/5206848/>
- [161] Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin, “Unsupervised Feature Learning via Non-Parametric Instance Discrimination,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [162] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: a review,” *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.
- [163] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2790–2797.
- [164] Y. Sui, G. Wang, and L. Zhang, “Sparse subspace clustering via low-rank structure propagation,” *Pattern Recognition*, vol. 95, pp. 261–271, 2019.
- [165] M. Joneidi, S. Vahidian, A. Esmaeili, W. Wang, N. Rahnavard, B. Lin, and M. Shah, “Select to better learn: Fast and accurate deep learning using data selection from nonlinear manifolds,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7819–7829.
- [166] K. Hamm and L. Huang, “Perspectives on cur decompositions,” *Applied and Computational Harmonic Analysis*, vol. 48, no. 3, pp. 1088–1099, 2020.
- [167] S. Wang and Z. Zhang, “Improving cur matrix decomposition and the nyström approximation via adaptive sampling,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2729–2769, 2013.

- [168] M. W. Mahoney and P. Drineas, “Cur matrix decompositions for improved data analysis,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, 2009.
- [169] C. Boutsidis and D. P. Woodruff, “Optimal cur matrix decompositions,” *SIAM Journal on Computing*, vol. 46, no. 2, pp. 543–589, 2017.
- [170] A. Esmacili, M. Joneidi, M. Salimitari, U. Khalid, and N. Rahnavard, “Two-way spectrum pursuit for cur decomposition and its application in joint column/row subset selection,” *arXiv preprint arXiv:2106.06983*, 2021.
- [171] E. Elhamifar, G. Sapiro, and R. Vidal, “See all by looking at a few: Sparse modeling for finding representative objects,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1600–1607.
- [172] Y. Shitov, “Column subset selection is np-complete,” *Linear Algebra and its Applications*, 2020.
- [173] E. Elhamifar, G. Sapiro, and S. S. Sastry, “Dissimilarity-based sparse subset selection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2182–2197, 2015.
- [174] P. Comon and G. H. Golub, “Tracking a few extreme singular values and vectors in signal processing,” *Proceedings of the IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.
- [175] S. Üreten, A. Yongaçoğlu, and E. Petriu, “A comparison of interference cartography generation techniques in cognitive radio networks,” in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 1879–1883.
- [176] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *arXiv preprint arXiv:1009.5055*, 2010.
- [177] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.