HIGHER-ORDER DATA REDUCTION THROUGH CLUSTERING, SUBSPACE ANALYSIS AND COMPRESSION FOR APPLICATIONS IN FUNCTIONAL CONNECTIVITY BRAIN NETWORKS

By

Alp Ozdemir

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Doctor of Philosophy

2017

ABSTRACT

HIGHER-ORDER DATA REDUCTION THROUGH CLUSTERING, SUBSPACE ANALYSIS AND COMPRESSION FOR APPLICATIONS IN FUNCTIONAL CONNECTIVITY BRAIN NETWORKS

By

Alp Ozdemir

With the recent advances in information technology, collection and storage of higher-order datasets such as multidimensional data across multiple modalities or variables have become much easier and cheaper than ever before. Tensors, also known as multiway arrays, provide natural representations for higher-order datasets and provide a way to analyze them by preserving the multilinear relations in these large datasets. These higher-order datasets usually contain large amount of redundant information and summarizing them in a succinct manner is essential for better inference. However, existing data reduction approaches are limited to vector-type data and cannot be applied directly to tensors without vectorizing. Developing more advanced approaches to analyze tensors effectively without corrupting their intrinsic structure is an important challenge facing Big Data applications.

This thesis addresses the issue of data reduction for tensors with a particular focus on providing a better understanding of dynamic functional connectivity networks (dFCNs) of the brain. Functional connectivity describes the relationship between spatially separated neuronal groups and analysis of dFCNs plays a key role for interpreting complex brain dynamics in different cognitive and emotional processes. Recently, graph theoretic methods have been used to characterize the brain functionality where bivariate relationships between neuronal populations are represented as graphs or networks. In this thesis, the changes in these networks across time and subjects will be studied through tensor representations.

In Chapter 2, we address a multi-graph clustering problem which can be thought as a tensor

partitioning problem. We introduce a hierarchical consensus spectral clustering approach to identify the community structure underlying the functional connectivity brain networks across subjects. New information-theoretic criteria are introduced for selecting the optimal community structure. Effectiveness of the proposed algorithms are evaluated through a set of simulations comparing with the existing methods as well as on FCNs across subjects.

In Chapter 3, we address the online tensor data reduction problem through a subspace tracking perspective. We introduce a robust low-rank+sparse structure learning algorithm for tensors to separate the low-rank community structure of connectivity networks from sparse outliers. The proposed framework is used to both identify change points, where the low-rank community structure changes significantly, and summarize this community structure within each time interval.

Finally, in Chapter 4, we introduce a new multi-scale tensor decomposition technique to efficiently encode nonlinearities due to rotation or translation in tensor type data. In particular, we develop a multi-scale higher-order singular value decomposition (MS-HoSVD) approach where a given tensor is first permuted and then partitioned into several sub-tensors each of which can be represented as a low-rank tensor increasing the efficiency of the representation. We derive a theoretical error bound for the proposed approach as well as provide analysis of memory cost and computational complexity. Performance of the proposed approach is evaluated on both data reduction and classification of various higher-order datasets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

With the recent advances in information technology, collection and storage of higher-order datasets such as multidimensional data with multiple aspects have become much easier and cheaper than ever before. These higher-order datasets bring with themselves the problems of interpreting the data and extracting useful information. Tensors, also known as multiway arrays, provide natural representations for higher-order datasets and enable us to analyze them by preserving multilinear relations among the different modes. Tensor type data appears in many applications including computer vision where grey level images and image sequences can be represented as two- and three-dimensional datasets, respectively, neuroimaging where signals such as electroencephalogram (EEG) recordings across multiple subjects, channels and experimental conditions and multimodal images can be represented as high order tensors [1–3], and hyperspectral imaging where the images or videos obtained through remote sensing can be represented as 3-way and 4-way tensors [4, 5].

These higher-order tensors are often very high-dimensional and contain large amount of redundant information [6]. Summarizing these huge datasets in a succinct manner is essential for better inference. Data reduction can be performed in several different ways. Clustering approaches provide a compact way of summarizing the data which can be processed quickly and interpreted easily [7]. On the other hand, linear or nonlinear mapping of the high-dimensional data onto

lower dimensional subspaces or manifolds provides direct reduction of data dimensionality [8]. Approaches such as PCA/SVD for linear low-rank subspace approximation and manifold learning techniques for nonlinear approximation are well-developed for vector type data. However, these dimensionality reduction techniques become inadequate when dealing with higher-order tensors. Applying vector based algorithms to tensors requires vectorization which results in extremely long vectors and breaks the natural multilinear structure of the data. Analysis of these long vectors also requires massive computing power. Therefore, there is a need to develop clustering and subspace projection approaches for dimensionality reduction of tensor data.

1.1 Data Reduction by Clustering

One particular way of performing data reduction is clustering. There are a variety of clustering algorithms specific to problems from different fields i.e computer science, earth science, life science and economics [9–12]. The common objective of all the clustering algorithms is grouping the samples similar to each other based on a specified proximity measure i.e. Euclidean distance, cosine distance or Mahalanobis distance [13]. Clustering algorithms can be simply grouped into two categories as Hierachical algorithms and partitional algorithms [10]. Hierarchical clustering approaches construct a hierarchical structure from a distance matrix and make use of dendograms to visualize the distance. Hierarchical algorithms are commonly implemented through either agglomerative or divisive methods. Agglomerative algorithms perform clustering by merging individual data points while divisive methods start with the entire dataset and successively divide it into subclusters [14]. In contrast to hierachical approaches, partitional clustering algorithms directly assign samples into K clusters. Partitional algorithms can be also further divided into square-error based algorithms, graph-theoretic algorithms and mixture resolving algorithms [10]. Squared-error

based algorithms such as K-means perform clustering based on specified cost functions which maximize intercluster distance and intracluster similarity. Mixture resolving approaches evaluate the probability distributions of each cluster and group the data points based on the distribution's parameters that can be iteratively estimated by expectation maximization algorithm [15]. Graph-theoretic approaches make use of the similarity matrix formed using neighborhood information of data samples. Spectral properties carried by eigenvectors of the graph Laplacian assist in finding the optimum cut points of the graph [16].

However, all of these methods are mostly limited to clustering a single data set or unimodal data. In the case of multiway data it may be important to find a common cluster structure across one of the modes. Consensus clustering methods address this problem by combining multiple clustering results obtained by applying various algorithms to the same dataset [17, 18]. The final result is obtained by minimizing the total dissimilarity between the target partition and each individual partition [17]. Similar to consensus clustering, multiview clustering aims to obtain common clustering structure from multiple views of the same data [19]. Multiview clustering can equivalently be thought as a higher-order data clustering problem and tensor based approaches show superior performance compared to both single view and the other multiview approaches [20]. More recently, multi-way clustering problem has been defined by the use of N-way similarity measure instead of pairwise similarity measures [21]. Using N-way similarity measure yields N-way super symmetric affinity tensor and Shashua et al. [22] showed that applying nonnegative tensor factorization to N-way affinity tensor yields desired partitions. Moreover, He et al. [23] reformulated the multi-way clustering problem as a PARAFAC problem.

1.2 Higher-order Data Decomposition

Increased usage of higher-order datasets has prompted researchers to develop new multilinear analysis tools to better capture the hidden multilinear structures underlying the observed data. Advantages of using multiway analysis over two-way analysis in terms of uniqueness, robustness to noise and computational complexity have been shown in many studies [24–27]. Two of the most well-known higher-order decomposition methods are Tucker decomposition and Parallel Factor Analysis (PARAFAC). PARAFAC, also known as canonical decomposition (CANDECOMP), is an extension of PCA to tensors and represents the tensor as a sum of rank-1 tensors. PARAFAC yields a unique decomposition and provides the only possible combination of rank-one tensors [28, 29]. For an N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$, PARAFAC decomposition of $\mathcal{X}$ is:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ .... \circ \mathbf{a}_r^{(N)} \tag{1.1}$$

where R is a positive integer, λ_r is the weight of the rth rank-one tensor, $\mathbf{a}_r^{(i)} \in \mathbb{R}^{I_i}$ is the rth factor of ith mode with unit norm where $i \in \{1,\ 2, ...,\ N\}$ and $r \in \{1,\ 2, ...,\ R\}$, and "$\circ$" denotes the outer product of vectors. The main restriction of the PARAFAC model is that the factors across different modes only interacts factorwise. For example, for a 3-way tensor, the ith factor corresponding to the first mode only interacts with the ith factors of the second and third modes. However, this restriction also provides the same number of factors for each mode and yields a unique solution for PARAFAC model [24, 30].

Tucker decomposition is a natural extension of the SVD to N-way tensors and decomposes the tensor into a core tensor multiplied by a matrix along each mode [31]. One version of Tucker decomposition known as HoSVD is obtained by adding an orthogonality constraint to the component

matrices. HoSVD of N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is written as:

$$\begin{gathered} \mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)}, \\ \mathcal{X} = \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} s_{i_1,i_2,\ldots,i_N} c_{i_1}(\mathbf{U}^{(1)}) \circ c_{i_2}(\mathbf{U}^{(2)}) \ldots \circ c_{i_N}(\mathbf{U}^{(N)}) \end{gathered} \tag{1.2}$$

where the matrix $\mathbf{U}^{(n)}$ contains the left singular vectors of the matrix obtained by unfolding $\mathcal{X}$ along nth mode, $c_n(\cdot)$ is the nth column vector and core tensor $\mathcal{S}$ is obtained by $\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}^{(1),\top} \times_2 \mathbf{U}^{(2),\top} \ldots \times_N \mathbf{U}^{(N),\top}$. In contrast to PARAFAC, Tucker models allow interactions between the factors obtained across the modes and the core tensor includes the strength of these interactions. However, the main drawback of Tucker decomposition is that the factors are not necessarily unique [24]. For example, the effect of rotating one of the mode matrices can be eliminated by inversely rotating the core tensor.

Both of the PARAFAC and Tucker decompositions can be computed using alternating least squares (ALS) algorithms [2, 24]. ALS repeatedly estimates the component matrices one at a time while keeping the others fixed until convergence criteria are satisfied. However, ALS based algorithms do not guarantee convergence to the global optimum and changing the start points for the algorithm may yield different solutions. To accelerate the convergence of ALS, studies in line search techniques and gradient based approaches can be integrated in ALS algorithms [32–34]. Similar to PARAFAC, Tucker decomposition can also be performed by ALS algorithms and the closed form solution is based on computing the left singular vectors of the matrix obtained by unfolding the tensor along the related mode [2]. To speed up this process, [35] proposed to compute only the leading singular vectors which yields a truncated version of HoSVD. However, truncated HoSVD does not provide the optimal solution for the specified rank. [36] proposed an iterative technique known as higher order orthogonal iteration (HOOI) and showed that the truncated de-

composition obtained by HOOI provides a better fit to the original tensor.

Recently, alternatives to these major tensor decomposition techniques have also been proposed [30, 37–43]. De Lathauwer [37] proposed block-term decomposition (BTD) which unifies HoSVD and PARAFAC and defined the essential conditions for uniqueness. Block-term decomposition factorizes the tensor as a sum of several tensors of the same size but with a lower multilinear rank than the original tensor. BTD is a more generalized version of PARAFAC decomposition and allows us to model more complex data structures. However, the main challenge with the block-term decomposition is how to choose the appropriate block rank. When the order of the tensors increases, computation and storage of the core tensor obtained by Tucker decomposition becomes more costly. To reduce the memory requirements as well as computational complexity, various tensor decomposition techniques including hierarchical Tucker decomposition and tensor train decomposition have been proposed [31, 39, 40]. Hierarchical Tucker Decomposition (H-Tucker) recursively splits the modes based on a hierarchy and applies SVD. Resulting binary tree contains a subset of the modes at each node [39]. Alternatively, Tensor-Train Decomposition (T-Train) has been proposed to compress large tensor data into smaller core tensors obtained by a sequence of QR and SVD decompositions of matrices [40]. Real-valued datasets may exhibit different characteristics such as nonnegativity, sparsity, symmetricity and researchers have proposed various tensor decomposition techniques by including these constraints. For example, Brachat et al. decomposes symmetric tensors as the sum of rank-one symmetric tensors [41]. Real data containing frequency counts, pixel intensities and spectra, are nonnegative, and nonegativity constraint can also be imposed to tensor factorization [30]. Moreover, Tichavsky et al. [38] extended joint block diagonalization of nonsymmetric matrices to 3-way $N \times N \times N$ tensors by decomposing the tensor into a block diagonal core tensor multiplied by factor matrices along each mode. More recently, Allen et al. [42] took sparsity of the tensors into account and proposed sparse tensor decompo-

sitions (Sparse-HoSVD, Sparse-PARAFAC) where Sparse-PARAFAC regularizes the factors with an l_1-norm penalty and Sparse-HoSVD performs sparse PCA on matricized tensors .

1.3 Tensor Subspace Tracking

High-dimensional datasets often lie near a low dimensional subspace, and there has been a massive amount of subspace learning algorithms used in data reduction and compression including reconstructive methods (principal component analysis (PCA), independent component analysis (ICA), nonnegative matrix factorization (NMF) [44–46]), discriminative approaches (linear discriminant analysis (LDA), canonical correlation analysis (CCA) [47, 48]). To better deal with corrupted data and missing entries, researchers have focused on more robust subspace estimation techniques and the problem of separating a sparse matrix and a low-rank matrix from their sum has received a lot of attention. The final goal usually is to either find the column span of the low-rank matrix or the support of the sparse one. There has been a large amount of recent work on batch methods for low-rank + sparse recovery and its various extensions such as Principal Component Pursuit, Outlier Pursuit and Low-Leverage Decomposition [49–62]. However, the subspace that the dynamic signals and streaming datasets lie in tends to evolve in time and these batch algorithms are inconvenient to capture changing dynamics of the subspace. To address this issue, a considerable number of online subspace learning algorithms have been proposed i.e. SVD based approaches (incremental SVD, incremental PCA [63, 64]), recursive least squares based approaches (PAST, PETRELS [65, 66]) and robust approaches (GRASTA, REPROCS, online RPCA [67–70]). However, all of these approaches are appropriate for vector type datasets and applying them to higher-order datasets requires vectorization. Vectorizing the large datasets yields very long vectors and makes the optimization of the algorithms highly costly. Vectorizing process also corrupts the

multi-dimensional couplings that the higher-order data contains.

To better preserve the multilinear structures of dynamic higher-order datasets, recently tensor-based approaches have been proposed to track dynamic tensor subspaces such as dynamic tensor analysis, streaming tensor analysis and window based tensor analysis [71, 72]. However, these approaches provide computationally efficient frameworks for analysis of streaming datasets by recursively updating subspace information and do not address the robustness of the subspace estimates. Goldfarb and Qin extended robust PCA to tensors (HoRPCA) by solving low-rank + sparse recovery problem for general higher order tensors [73]. However, this method is highly computationally expensive and does not update the subspaces online. Li et al. [74] presented a robust subspace learning algorithm (RTSL) that incrementally updates the tensor subspace. Moreover, Nion et al. [75] proposed two adaptive approaches to track PARAFAC decompostion of 3-way tensors. These approaches suggest to update the PARAFAC decomposition at every time point based on simultaneous diagonalization or minimization of weighted least squares criterion. More recently, Mardani et al. [76] proposed an online subspace learning method based on nuclear norm minimization and extended this approach for matrices and higher order datasets. Extension of this algorithm to tensors takes advantage of parallel factor analysis (PARAFAC) model to minimize tensor rank and considers temporal information as one of the tensor modes. Similar to [76], OLSTEC proposed in [77] also tracks the subspace of partially observed higher-order data by making use of PARAFAC decomposition.

1.4 Tensor Based Approaches in Neuroscience Applications

Undoubtedly, the brain is one of the most complex biological systems in existence which has motivated researchers to study it through different imaging modalities including EEG, MEG and

MRI. However, all of these modalities provide either multilinear signals or multidimensional images which are most naturally represented by tensors. Recently, tensor based approaches have been widely used for analysis of these datasets [2]. For instance, Morup et al. [1] proposed to use PARAFAC model to decompose event-related EEG data into channel $\times$ frequency $\times$ time components and identified the components corresponding to visual event related potential (ERP) paradigm. In order to avoid degenerations due to time shifts that occur in EEG and fMRI signals and to better model the delays occurring in neuro-physiological systems, the shifted PARAFAC model has been proposed [78]. Tensor decomposition has also been used in BCI applications [79–84]. Lee et al. [79] presented non-negative tensor factorization by adding nonnegativity constraint to PARAFAC decomposition and showed that hidden multiway patterns found by NTF successfully classify the EEG signals. Similarly, generalized tensor discriminant analysis (GTDA) [85] provides discriminative multilinear subspace information for single trial EEG classification [80, 81]. Similarly, slice oriented decomposition (SOD) developed for 3-way tensors decomposes the tensors as outer product of slice matrices and features extracted by SOD have been employed in BCI applications [82]. Tucker models and extensions have also been used in BCI studies to extract features for classification of EEG signals [83, 84]. Moreover, brain source localization studies take advantage of tensor based methods [86–89]. These studies applied PARAFAC decomposition to time-frequency-electrode tensor constructed from EEG data and showed that some of the PARAFAC components are related to the artifacts while others are associated with the origins of epileptic seizure.

More recently, tensor based approaches were used in fMRI studies for several different applications. For example, Barnathan et al. [90] proposed a hybrid algorithm by combining tensor decompositions with wavelet transform and then applied it to fMRI to cluster motor tasks. They also showed that this hybrid method outperforms voxelwise analysis and methods depending only

on wavelet transform or tensor decompositions in terms of space, time, and accuracy. An extension of ICA for tensors was proposed in [91] and has been used used to obtain more accurate activation maps for multi-subject and multi-session fMRI analysis with increased robostness against deviation from model assumptions. In addition, EEG-fMRI fusion applications have taken advantage of tensor decomposition since multilinear approaches better reflect the intrinsic structure of multimodal multiway neuroimaging data [92, 93]. Tensor based approaches have also been used for compression of EEG signals and regression analysis of fMRI data [94–96].

In most of the studies mentioned above, the tensors are constructed from either the time-frequency distribution of the signals across channels or the time series across subjects for multi-channel EEG recordings or fMRI voxel intensity values. However, there is less work on employing tensors to represent the multivariate relationships among the different channels or voxels. Functional connectivity is defined as the statistical dependency between spatially remote neurophysiological events and is most commonly represented through the use of graph theoretic tools such as the adjacency/connectivity matrix. Recently, functional connectivity has been used to understand coordinated and integrated activity of the human brain [97, 98], and it has been shown that synchronization between different brain regions plays an important role in different cognitive and emotional processes [99, 100] as well as in various neurological and psychiatric disorders [101–104]. More recently, tools from graph theory have been employed to analyze the functional connectivity of the brain by associating nodes with distinct brain regions and edges with pairwise interactions between them [105, 106]. Moreover, it has been shown that functional connectivity networks change dynamically in short time scales and exhibit task-related patterns [107–111]. Tensors provide a natural tool to represent these dynamic functional connectivity networks constructed across subjects, time, frequency and experimental conditions. In recent work, tensor decomposition methods such as HoSVD have been used to summarize these high order datasets and to identify a small

number of network states [112, 113].

1.5 Organization of the Dissertation

This thesis makes fundamental contributions to data reduction of tensor type data with a particular focus on providing a better understanding of dynamic functional connectivity networks. In Chapter 2, we approach the problem of data reduction through clustering. In particular, we focus on obtaining a common community structure across multiple connectivity networks, which can be thought of partitioning a 3-way tensor. Therefore, unlike classical graph clustering this is a multi-graph clustering problem where the tensor corresponds to functional connectivity brain networks collected in time across multiple subjects. In order to understand the organization of functional connectivity networks, it is important to determine the community structure underlying these complex networks. Moreover, the study of functional networks is confounded by the fact that most neurophysiological studies consist of data collected from multiple subjects, thus, it is important to identify communities representative of all subjects. In Chapter 2, we propose a hierarchical consensus spectral clustering approach to address these problems. Furthermore, new information-theoretic criteria are introduced for selecting the optimal community structure. The proposed framework is applied to electroencephalogram (EEG) data collected during a study of error-related negativity (ERN) to better understand the community structure of functional networks involved in cognitive control.

The approach presented in Chapter 2 reduces the connectivity data across all subjects and within a given time interval into a single cluster structure. In Chapter 3, we propose an alternative way to reduce this high dimensional data through linear subspace estimation and update methods. In this approach, the dynamics of the connectivity networks are taken into account such that the data reduction is done across subjects for time intervals determined by the subspace tracking ap-

proach. Recent years have seen a growth of methods for subspace tracking of vector type data. The main contribution of this chapter is that we introduce a tensor based approach for tracking dynamic functional connectivity networks. The proposed framework introduces a robust low-rank+sparse structure learning algorithm for tensors to separate the low-rank community structure of connectivity networks from sparse outliers. The proposed framework is used to both identify change points, where the low-rank community structure of the FCN changes significantly, and summarize this community structure within each time interval. The proposed framework is applied to the study of cognitive control from electroencephalogram (EEG) data during a Flanker task.

In Chapter 4, we address the issue of data reduction through tensor decomposition. To this aim, we propose a novel multiscale analysis technique to efficiently encode nonlinearities in tensor type data. The proposed method constructs data-dependent multiscale dictionaries to better represent the data and consists of two major steps: 1) Constructing a tree structure by decomposing the tensor into a collection of permuted subtensors, and 2) Constructing multiscale dictionaries by applying HoSVD to each subtensor. We introduce different variations of the proposed MS-HoSVD method including a single scale and multi-scale decomposition along with an adaptive pruning method. We derive a theoretical error bound for the proposed approach as well as provide analysis of memory cost and computational complexity. Finally, we apply the proposed algorithm to data reduction of real datasets to illustrate the improvement in the compression performance compared to HoSVD, T-Train and H-Tucker deompositions. In addition, we show how the features obtained from multiscale representation provide advantages over regular HoSVD and T-Train features for classifying tensors containing nonlinearities such as rotation or translation.

Chapter 2

Hierarchical Spectral Consensus Graph Clustering for Group Analysis of Functional Brain Networks

2.1 Introduction

Functional connectivity is defined as the statistical dependency between spatially remote neurophysiological events [97] and is the key to understanding how the coordinated and integrated activity of the human brain takes place [98]. In recent years, many studies have suggested synchronization of neuronal oscillations as one plausible mechanism in the interaction of spatially distributed neural populations [114]. Moreover, it has been shown that synchronization between different brain regions plays an important role in different cognitive and emotional processes [99,100] as well as in various neurological and psychiatric disorders [101–104]. Synchronization refers to interdependencies among activities of different neuronal assemblies and requires the need to focus on the temporal dynamics of neural networks in the millisecond range. Therefore, neuroimaging techniques with high temporal resolution, such as electroencephalogram (EEG) [101, 115] and

magnetoencephalogram (MEG) [116], are the most appropriate tools.

Although phase synchrony is successful at quantifying pairwise interactions, it cannot completely describe the complex relationship between function and organization of the brain. Recently, research in the area of complex networks, in particular graph theoretic methods, has been used to characterize the relationship between the topology and the function of the brain [117–120]. The bivariate relationships between neuronal populations are represented as graphs where the nodes correspond to the individual sites and the edges to the strength of the interaction quantified by functional connectivity measures. The conventional approach to functional connectivity graph analysis extracts topological metrics either on the entire graph, i.e. global metrics, or at each node, i.e. local metrics. At the large topological scale, the small-world organization, whereby both integration (relatively high global-efficiency/low path length) and segregation (relatively high local-efficiency/clustering coefficient) of information between brain regions are supported, has been investigated thoroughly [121, 122]. The small-world model has also been shown to be significantly altered in various brain disorders and pathologies such as schizophrenia [123], autism [124], spinal cord injuries [125] and Alzheimer's disease [126]. At the local scale, the centrality or the degree of individual nodes can be computed and used to characterize the brain graph reorganization during different tasks and events. Although the global and local indices summarize the key aspects of the connectivity networks, they do not provide any information about the intermediate scale of network organization which is more accurately described by the community structure of the network [127, 128]. A community structure in a graph is defined as a densely connected set of nodes with sparse connections between communities in the network. It is hypothesized that the community structure of complex biological networks is indicative of robustness [127] and contributes to functionality [129] by compartmentalizing specific functions within certain cortical regions without perturbing the rest of the network [130]. Intra-cluster associations are thought

to describe the segregation of information processing while the inter-cluster associations testify to the integration of information processing across distant brain regions [131–133].

Identification of communities in the functional connectivity graphs has been originally addressed using methods like Principle Component Analysis (PCA) [134] and Independent Component Analysis (ICA) [135] which put non-physiological constraints in the obtained components such as orthogonality and independence. Recently, methods from spectral graph clustering have been used to detect communities [136, 137] by mapping the functional connections to a multi-dimensional subspace defined by a set of eigenvectors. However, these methods require *a priori* knowledge about the number of clusters and do not reveal a hierarchical decomposition of the network. Meunier and others [138–140] argue that most complex networks, including functional connectivity networks, possess a multi-scale community characteristic, i.e, are hierarchically decomposable into a finite number of modular levels. Therefore, a hierarchical decomposition of functional connectivity graphs is a more natural representation than conventional clustering approaches for community detection in brain networks.

A key challenge in identifying the community structure of brain networks is determining a common structure across multiple subjects. Current work either focuses on obtaining the community structure for the average connectivity network or on analyzing each subject individually and obtaining a common community structure using consensus clustering techniques [141]. Averaging neglects the variance across subjects and can be influenced by the outliers. Consensus clustering, also known as clustering ensembles, yields a stable and robust final clustering that is in agreement with the individual clusterings through a consensus function [18, 142]. Therefore, in this chapter we will introduce a hierarchical consensus based approach in which the best community structure is identified by combining information shared across multiple subjects.

In this chapter, we first quantify functional connectivity using a new time-varying measure

of phase synchrony and apply it to multichannel EEG data to quantify pairwise synchrony. The resulting connectivity matrices are treated as weighted undirected graphs representing each subject. We then introduce a new hierarchical graph partitioning method based on spectral graph theory, in particular the Fiedler bi-partitioning method [143]. This partitioning method is combined with two novel information theoretic criteria, homogeneity and completeness, to introduce a non-greedy consensus based hierarchical algorithm, the Fiedler Consensus Clustering Algorithm (FCCA), that is designed to reveal multiple levels of community organization common across subjects. Next, an information-theoretic quality measure is introduced to identify the optimal community structure. Finally, the proposed approach is applied to EEG data collected during a study of cognitive control in the brain based on the error-related negativity to test the approach on a known biological signal.

2.2 Background

2.2.1 Time-Varying Measure of Phase Synchrony

Phase synchronization within different frequency bands across the brain has been shown to be a plausible mechanism explaining neuronal integration [114, 144]. Two commonly used measures for quantifying time-varying phase synchrony are Hilbert transform and complex wavelet transform [145–147]. It has been observed that the two approaches are similar in their results with the wavelet based methods giving higher resolution phase synchrony estimates over time and frequency, especially at the low frequency range [145]. Although the wavelet based phase synchrony estimates address the issue of non-stationarity, they suffer from the resolution tradeoff, i.e. the frequency resolution is high at low frequencies and low at high frequencies. For this reason, there is a need for high time-frequency resolution phase distributions that can better track dynamic changes in phase synchrony. In the proposed work, pairwise functional connectivity will be quantified us-

ing a recently introduced time-frequency phase estimation method based on Reduced Interference Rihaczek distribution (RID-Rihaczek) [148, 149].

Reduced Interference Rihaczek Distribution (RID-Rihaczek) is given by:

$$C(t,\omega) = \iint \exp\left(\frac{-(\theta\tau)^2}{\sigma}\right) \exp\left(j\frac{\theta\tau}{2}\right) A(\theta,\tau) e^{-j(\theta t+\tau\omega)} d\tau d\theta, \tag{2.1}$$

where $\exp(-(\theta\tau)^2/\sigma)$ is the Choi-Williams kernel used to filter out the cross-terms, $A(\theta,\tau) = \int x(u+\frac{\tau}{2})x^*(u-\frac{\tau}{2})e^{j\theta u}du$ is the ambiguity function of the signal and $\exp(j\theta\tau/2)$ is the kernel corresponding to the Rihaczek distribution [150]. The phase difference between two signals based on this complex distribution is computed as

$$\Phi_{12}(t,\omega) = \arg\left[\frac{C_1(t,\omega)C_2^*(t,\omega)}{|C_1(t,\omega)||C_2(t,\omega)|}\right], \tag{2.2}$$

where $C_1(t,\omega)$ and $C_2(t,\omega)$ refer to the complex energy distributions of the two signals $x_1(t)$ and $x_2(t)$ respectively and a synchrony measure quantifying the intertrial variability of the phase differences, phase locking value (PLV), is defined as

$$PLV(t,\omega) = \frac{1}{N}\left|\sum_{k=1}^{N} \exp(j\Phi_{12}^k(t,\omega))\right|, \tag{2.3}$$

where N is the number of trials and $\Phi_{12}^k(t,\omega)$ is the time-varying phase estimate between two signals recorded at different electrodes for the kth trial. If the phase difference varies little across the trials, PLV is close to 1. Compared to the existing synchrony measures, in our previous work we have shown that RID-Rihaczek based phase synchrony measure is more robust to noise, has uniformly better time-frequency resolution with less bias, and perform superior at detecting actual

synchrony within a group of oscillators [148].

2.2.2 Graph Theory

Recent developments in the quantitative analysis of complex networks, based largely on graph theory, have been rapidly translated to studies of brain network organization [105, 151]. In this approach, the different regions of the brain correspond to the nodes in the network and the pairwise functional connectivity corresponds to the edges of the network. An undirected, connected, weighted graph, $\mathcal{G} = (V, E, \mathbf{W})$, consisting of a finite set of N nodes, $V = \{v_i | i \in \{1, 2, ..., N\}\}$, and a set of edges, E, associated with each node pair and a weighted adjacency matrix $\mathbf{W}$ can be used to represent these functional connectivity networks. For a binary graph, $w_{ij} \in \{0, 1\}$ and for a weighted graph $w_{ij} \in [0, 1]$. In an undirected graph, the edge weights are represented by a symmetric weighted adjacency matrix $\mathbf{W} = [w_{ij}]$ where $i, j \in \{1, 2, ..., N\}$. The sum of all elements along the ith row of matrix $\mathbf{W}$, $d_i = \sum_{j=1}^{N} w_{ij}$, is the degree of node v_i. When m graphs representing the same network are available, their adjacency matrices are represented as the set $\mathbb{W} = \{W^r\}$ where $r \in \{1, 2, ..., m\}$.

2.2.3 Spectral Clustering

A commonly used approach to identifying the community structure within graphs is spectral clustering thanks to its simple implementation and promising performance. Given a weighted and undirected graph $\mathcal{G}$, the spectrum of the graph is represented by the eigenvalues and eigenvectors of the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{W}$ is the adjacency matrix and $\mathbf{D}$ is the degree matrix containing degrees of nodes along the diagonal [152, 153]. Different versions of the Laplacian matrix, i.e. the symmetric normalized and the random walk normalized versions, have

been used leading to different versions of the spectral clustering algorithm. In this study, we use the symmetric version of the normalized Laplacian matrix defined as $\mathcal{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$ which yields more robust clustering solutions [136].

Since the normalized Laplacian matrix is a square, symmetric, and positive semi-definite matrix, its eigenvectors and eigenvalues are described by the equation $\mathcal{L}\mathbf{u}_i = \lambda\mathbf{u}_i$ and the eigenvectors, $\{u_1, u_2, ..., u_N\}$ are orthonormal and the eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_N\}$ are positive and real. Spectral clustering algorithm finds the spectrum of $\mathcal{G}$ through the eigendecomposition of its Laplacian matrix and embeds the original vertices in $\mathcal{G}$ to a low dimensional spectral domain formed by the graph spectrum. Typically, a subset of eigenvectors, $\{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k\}$ where $k < N$, is extracted and an optimization technique is iteratively applied to cluster centers within the data using algorithms such as k-means, fuzzy k-means [154], generalized synchronization cluster analysis [137], the Ng-Jordan-Weiss algorithm (NJW) [155], or power iteration clustering (PIC) [156]. This transformation enhances the intrinsic relationship among the original vertices leading to improved cluster identification in the new low dimensional space [136, 137, 152, 157].

An alternative to spectral clustering is to evaluate only one eigenvector for the purpose of bi-partitioning, i.e. identifying a minimal cut of the graph. This eliminates the problem of searching for the optimal set of eigenvectors. According to Holzrichter et. al. [158], the optimal minimal cut of a graph is defined by the eigenvector, $\mathbf{u}_2$, associated with the second smallest non-zero eigenvalue, λ_2, of the Laplacian matrix. This eigenvector is referred to as the Fiedler vector, $\mathbf{u}_F$, and defines a set of two clusters $\{C_1, C_2\}$ where

$$v_i \in \begin{cases} C_1 & \text{if } u_F(i) \geqslant 0 \\ C_2 & \text{if } u_F(i) < 0. \end{cases} \tag{2.4}$$

The Fiedler partition can be iteratively applied to each successive partition in order to achieve a

clustering with $k > 2$. In this study, the partitioning of a graph using the Fiedler vector will be referred to as FiedlerPartition(G), which partitions the nodes of graph G into two clusters, G_1 and G_2 such that $G_1 \cup G_2 = V$. Partitioning the graph according to the Fiedler vector generates a community structure in which the intra-cluster nodal relationships are maximally 'strong'. Repeating this partitioning process to the subsequent sub clusters reveals a hierarchical configuration of the network structure.

2.2.4 Consensus Clustering

In many clustering problems, it is common to apply different algorithms to the same data and then use a consensus method to combine the results [159]. In this chapter, a similar framework for obtaining a common community structure from multiple graphs is proposed. Three popular consensus clustering methods are consensus averaging, majority voting and the hypergraph partitioning algorithm (HPGA). The first approach averages m adjacency matrices to obtain $\widehat{\mathbf{W}} = [\widehat{w}_{ij}]$ where

$$\widehat{w}_{ij} = \frac{1}{m} \sum_{r=1}^{m} w_{ij}^{(r)}. \tag{2.5}$$

This approach is computationally efficient but loses the inter-subject variability.

The second commonly used approach for obtaining a common community structure across multiple graphs is to identify the community structure of each individual graph and then combine the information across the multiple community structures to identify a global community structure. The combination of community structure across multiple graphs or clustering solutions has been accomplished through different functions such as majority voting [160], mixture-model approach [161], and disagreement minimization methods [142].

Finally, HyperGraph-Partitioning Algorithm (HGPA) is used to extract common clustering structure [18, 162]. HGPA treats each cluster across all base clusterings as a hyperedge within a single global graph. The algorithm is a multilevel graph partitioning system which partitions this graph in three steps: 1) compress the graph by collapsing hyper-edges, 2) partition the compressed graph using a minimum cut objective function, and 3) decompress the partitions and repeat the process. HGPA has a computational complexity of $O(kNh)$ where h is the number of hyperedges. However, its overall complexity is dependent upon the total complexity of the clustering algorithms used to obtain the base clusterings. HGPA has the disadvantage of generating clusters of approximately equal sizes, even though in real networks, equally sized clusters are unlikely.

2.2.5 Modularity

The most commonly used cluster quality measure, modularity [163], compares a community structure to the expected community structure of a random graph such that there exists a high number of edges within clusters and low number of edges between clusters. Modularity for a weighted graph is defined as

$$Q = \frac{1}{2z} \sum_{ij} \left[w_{ij} - \frac{d_i d_j}{2z} \right] \sigma_{ij} \tag{2.6}$$

such that z is the sum of all edge weights in the graph and $\sigma_{ij} = 1$ if v_i and v_j are in the same cluster and 0 otherwise. Unfortunately, this definition of modularity does not always result in the highest value for the true community structure [164] and can reveal a suboptimal structure due to the simplistic random model computed through $\frac{d_i d_j}{2z}$ in the modularity equation [165].

2.2.6 Cohen's Kappa

One of the most commonly used measures to quantify the quality of an observed cluster with respect to the true structure is Cohen's Kappa measure. Cohen's Kappa [166] is a measure of agreement between two observers and is defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{2.7}$$

where p_o is the probability of observed agreement and p_e is the probability of expected agreement.

Cohen's Kappa measure can also be used to quantify the agreement between the ground truth clustering map (A) and the clustering map (B) obtained from the clustering algorithm as in Table-2.1. $A_{i,j}$ is equal to 1 if nodes i and j are assigned to the same cluster and 0 otherwise. Similarly, $B_{i,j}$ is equal to 1 if nodes i and j are assigned to the same cluster and 0 otherwise. In Table-2.1, a is the number of node pairs which are correctly identified as being in the same cluster, b is the number of node pairs which are falsely identified as being in the same cluster, c is the number of node pairs which are falsely identified as not being in the same cluster and d is the number of node pairs which are correctly identified as not being in the same cluster. Based on this observation, p_0 and p_e can be computed as:

$$\begin{aligned} p_o &= \tfrac{a+d}{a+b+c+d}, \\ p_e &= \tfrac{f1\times f2+g1\times g2}{(a+b+c+d)^2}. \end{aligned} \tag{2.8}$$

Standard error of kappa statistic is also known and is defined as [167]:

$$SE(\kappa) = \sqrt{\frac{p_o(1-p_o)}{(a+b+c+d)(1-p_e)^2}}. \tag{2.9}$$

Table 2.1: Agreement table for the observers used for computing the kappa score

		Observer 1 (Ground Truth)		
		$A_{i,j}=1$	$A_{i,j}=0$	
Observer 2 (Measurement)	$B_{i,j}=1$	a	b	$f2$=a+b
	$B_{i,j}=0$	c	d	$g2$=c+d
		$f1$=a+c	$g1$=b+d	

2.3 Information Theoretic Cluster Quality Measure

One problem with hierarchical clustering algorithms is how to determine the optimal number of clusters. In this section, we introduce a new measure to quantify the quality of the resulting clusters in the absence of 'ground truth' information, i.e. knowledge about the actual cluster structure.

2.3.1 Inter and Intra Edge Distribution

By the definition of a cluster, the pairwise connections within a cluster must be stronger than the inter-cluster connections. In this study, we propose measures that evaluate the quality of a particular clustering structure based on the distribution of the inter and intra-edge distributions across m graphs. These distributions will be defined similar to probability mass functions (pmfs). Prior to defining the pmfs of intra-cluster and inter-cluster edges, a function that maps the continuous edge values to a discrete alphabet is defined as $f : \mathbf{W}_i^{(r)} \rightarrow \mathbf{S}_i^{(r)}$ where $\mathbf{W}_i^{(r)} = \{w_{ij}^{(r)} \in [0,1]\}$ refers to the ith row of the rth adjacency matrix, $\mathbf{S}_i^{(r)} = \{s_{ij}^{(r)} \in \{1,2,...,N\}\}$, and $r \in \{1,2,...,m\}$. The elements of each row of the connectivity matrix across subjects are mapped to discrete integer values between 1 and N to eliminate the variation of edge strengths across subjects and extract only relational information about the pairwise edge strengths. We propose to use the rank function to do this mapping such that the node pair with the largest edge weight is assigned a 1 and the weakest node pair is assigned N.

When a particular cluster set $C = \{c_1, c_2, ..., c_k\}$ is identified, the probability mass function of intra-cluster ranks for a particular cluster, c_t, is defined as

$$P_{c_t}^{intra}(\beta) = \left\{ \frac{F_{c_t}^{intra}(\beta)}{\sum_{\beta=1}^{N} F_{c_t}^{intra}(\beta)} \right\}, \tag{2.10}$$

where

$$F_{c_t}^{intra}(\beta) = \sum_{r=1}^{m} \sum_{i,j}^{N} \delta(s_{ij}^r, \beta) \ \mid \ v_i, v_j \in c_t, \tag{2.11}$$

$t \in \{1, 2, \ldots, k\}$, $\beta \in \{1, 2, \ldots, N\}$, and

$$\delta(x, y) = \begin{cases} 1 & \text{if} \quad x = y \\ 0 & \text{otherwise.} \end{cases}$$

This function computes the frequency with which node pairs with varying strengths of connectivity are assigned to the same community.

Similarly, the probability mass function of inter-cluster ranks for cluster c_t is defined as

$$P_{c_t}^{inter}(\beta) = \left\{ \frac{F_{c_t}^{inter}(\beta)}{\sum_{\beta=1}^{N} F_{c_t}^{inter}(\beta)} \right\}, \tag{2.12}$$

where

$$F_{c_t}^{inter}(\beta) = \sum_{r=1}^{m} \sum_{i,j}^{N} \delta(s_{ij}^r, \beta) \ \mid \ v_i \in c_t; v_j \notin c_t. \tag{2.13}$$

This function computes the frequency with which node pairs with varying strengths of connectivity are assigned to different communities.

2.3.2 Homogeneity and Completeness

The next step is to quantify homogeneity and completeness, two principal characteristics which determine the quality of clustering. A homogeneous cluster contains only data points which belong to the same class while a complete cluster contains all possible data points within the sample space (Fig. 2.1).

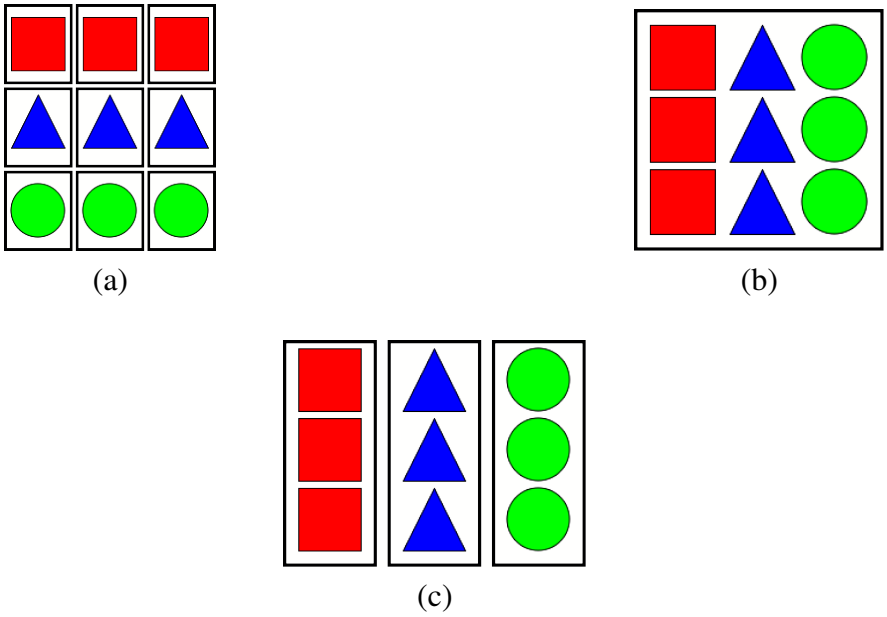


Figure 2.1: An illustration of variations in homogeneity and completeness in a group of objects where the true number of communities is 3; a) High homogeneity and low completeness; b) High completeness and low homogeneity; c) High homogeneity and completeness.

Similar measures such as F-measure [168] and V-measure [169], have been used in the literature to quantify the quality of a cluster. Both measures, however, require *a priori* knowledge of class labels. In most cases, this 'ground truth' is unknown and therefore alternative measures of cluster accuracy are needed. In this study, we propose new homogeneity and completeness measures which depend on the edges' strength and we quantify the quality of a clustering structure using the harmonic mean of the homogeneity and completeness measures similar to V-measure.

In this study, we use the observation that homogeneity is inversely related to the variance of

edge ranks within a cluster. If a particular cluster is homogeneous, then we would expect the pairwise connection strengths among the members of that cluster to be close to each other, thus implying the ranks of the weights to have small variance. We propose to quantify this variation through a measure of normalized entropy such that the lowest homogeneity score is obtained for a cluster containing a uniform distribution of ranks or large variation among the edge weights and the maximum homogeneity score is obtained for a cluster containing only one rank. Therefore, a normalized entropy measure of the cluster's intra-cluster rank distribution would be indicative of homogeneity. However, as cluster size gets smaller the intra-cluster rank distribution will naturally become more concentrated thus increasing the homogeneity. To account for this, we introduce a normalization term in the definition of homogeneity as follows: $\mathfrak{H}_{c_t} = \alpha_{C_t}\left(1 - \frac{H(P_{c_t}^{intra})}{log_2 N}\right)$ where $\alpha_{C_t} = \frac{|C_t|}{N}$ and this measure is always between 0 and 1.

Similarly, we define a metric using relative entropy between the inter-cluster rank and intra-cluster rank distributions to quantify completeness. Rank distributions for inter-edges and intra-edges are expected to be different from each other to maximize completeness. The similarity between two distributions is commonly quantified using divergence measures. In this study, we propose to use the Jensen-Shannon divergence measure [170] since it is symmetric and bounded. With respect to completeness, a divergence measure approaching 1 is synonymous with increased completeness. Completeness of a cluster is therefore defined as $\mathfrak{C}_{c_t} = JS(P_{c_t}^{intra}, P_{c_t}^{inter})$ where $JS(\mathfrak{p}, \mathfrak{q}) = \frac{1}{2}\left[\sum_i \mathfrak{p}(i) log_2 \frac{\mathfrak{p}(i)}{0.5(\mathfrak{p}(i)+\mathfrak{q}(i))} + \sum_i \mathfrak{q}(i) log_2 \frac{\mathfrak{q}(i)}{0.5(\mathfrak{p}(i)+\mathfrak{q}(i))}\right]$. Average completeness, $\hat{\mathfrak{C}} = \frac{1}{k}\sum_{t=1}^{k} \mathfrak{C}_{c_t}$, and average homogeneity, $\hat{\mathfrak{H}} = \frac{1}{k}\sum_{t=1}^{k} \mathfrak{H}_{c_t}$, are computed across all clusters.

Similar to the balanced F-score, our quality measure U for the final clustering structure is defined as the harmonic mean of the average completeness and average homogeneity as follows:

$$U = \frac{2}{\hat{\mathfrak{C}}^{-1} + \hat{\mathfrak{H}}^{-1}}. \tag{2.14}$$

2.4 Fiedler Consensus Clustering Approach

Since the literature has consistently demonstrated that the Fiedler vector is highly effective in partitioning graphs [171, 172], in this study, we use the Fiedler vector for performing consensus clustering across multiple weighted graphs. The original connectivity matrices are bi-partitioned into two clusters using the Fiedler partitioning method. This results in a cluster matrix for the r^{th} subject T^r such that

$$T^r(i,j) = \begin{cases} 1 & \text{if nodes } v_i, v_j \text{ are in the same cluster} \\ 0 & \text{otherwise} \end{cases}$$

and $r = \{1, 2, ..., m\}$. In order to find the common community structure across multiple graphs, we introduce a co-occurence matrix $\mathbf{P}$ where $P(i,j) = \sum_{r=1}^{m} \frac{T^r(i,j)}{m}$ and $P(i,j) \in [0,1]$. $P(i,j)$ is the probability that a pair of nodes are members of the same cluster across multiple graphs. The adjacency matrix reflects the strength of a direct relationship between a node pair, whereas $\mathbf{P}$ reflects the likeliness that a pair of nodes are in the same cluster across all subjects.

The Laplacian matrix of $\mathbf{P}$ is computed and the Fiedler vector is found to form a bi-partition of $\mathbf{P}$ into a community structure composed of clusters c_1 and c_{-1}. Since $\mathbf{P}$ represents the probability that a node pair should be clustered together, the Fiedler partition of $\mathbf{P}$ represents the community structure common to all graphs. The initial partition set, $C = \{c_1, c_{-1}\}$, contains 2 clusters but if $k > 2$ is desired, the process can be repeated by selecting a cluster in C to partition. In this case, c_1 or c_{-1} is selected based on the ζ values of each cluster. At each step of partitioning, weighted

Algorithm 2.1 Fiedler Consensus Clustering Algorithm

Input: m $N \times N$ dimensional graphs, $G = \{G^1, G^2, ..., G^m\}$ with vertices $V = \{v_1, v_2, .., v_N\}$ and edges $E^r = \{w_{ij}^r : v_i, v_j \in V\}$ such that $G^r = (V, E^r)$ and $r = \{1, 2, ..., m\}$.
Input: Number of clusters, k.
Output: k clusters $C = \{c_1, c_2, ..., c_k\}$ where $c_j \subset V$.
$C = \varnothing$
for $t = 2$ to k **do**
  $P' = \mathbf{0}_{|\mathbf{V}|\times|\mathbf{V}|}$
  for $r = 1$ to m **do**
    submatrix $\hat{G}^r \subset G^r | \hat{G}^r = (V, E^r)$
    (V_1, V_2) = SubRoutine(Fiedler Partition($\hat{G}^r$))
    $P'(i,j) = P'(i,j) + T^r(i,j)$ where $T^r(i,j) = \begin{cases} 1 & \text{if nodes } v_i, v_j \in V_1 \text{ or } v_i, v_j \in V_2 \\ 0 & \text{otherwise} \end{cases}$
  end for
  $P = \frac{P'}{m}$
  (V_1, V_2) = SubRoutine(Fiedler Partition(P))
  $C = C \cup \{V_1, V_2\}$.
  if $t \neq k$ **then**
    $score = 0_{t\times 1}$
    for $\gamma = 0 : 0.1 : 1$ **do**
      $\Lambda = [\zeta_1(\gamma)|\zeta_2(\gamma)|...|\zeta_t(\gamma)]$
      $j = \min_q\{\Lambda(q)\}$ and $q \in \{1, 2, ..., t\}$
      $score(j) = score(j) + 1$
    end for
    $V = c_i | i = \min_q\{score(q)\}$ and $q \in \{1, 2, ..., t\}$
    $E^r = \{w_{ij}^r : v_i, v_j \in V\}$
    $C = C \backslash \{c_i\}$
  end if
end for

sum of homogeneity and completeness is computed to select the cluster to be partitioned at the next level.

$$\zeta_{C_t} = \hat{\mathfrak{C}}_{C_t}\gamma + \hat{\mathfrak{H}}_{C_t}(1-\gamma) \tag{2.15}$$

where $\gamma \in [0, 1]$. For each cluster, a set of ζ values are created by choosing a range of $\gamma \in [0, 1]$. The cluster whose ζ values are lower than others' for a majority of γ values is selected for partitioning.

Next, sub-matrices are extracted from the original connectivity matrices such that they only

Algorithm 2.2 Fiedler Partition

Input: graph $G = (V, E)$.
Output: Vertex sets V_1 and V_2.
Compute Normalized Laplacian Matrix, L, of G.
Compute $|V|$ eigenvectors, $\mathbf{u}$, and eigenvalues, λ, of L.
Order eigenvalues in ascending order: $\lambda_1 \leqslant \lambda_2 \leqslant ... \leqslant \lambda_{|V|}$.
$u_F = u_i$ where $i = \min_q\{\lambda_q\}|\lambda_q \neq 0$.
Sort elements of u_F and find $u_F(i)$ which has maximum gap with ensuing element
for $j = 1$ to $|V|$ **do**
$$v_j \in \begin{cases} V_1 & \text{if } u_F(j) \leqslant u_F(i) \\ V_2 & \text{if } u_F(j) > u_F(i) \end{cases}$$
end for

contain the nodes of the chosen cluster. These sub-matrices are used to derive the new sub co-occurence matrix, $\mathbf{P}^y$, where $y = 1$ if cluster c_1 was selected and $y = -1$ if cluster c_{-1} was selected. The Fiedler partition will result in two new clusters, c_1^y and c_{-1}^y. The final cluster set is $C = \{c_{-y}, c_1^y, c_{-1}^y\}$ which is a concatenation of the two new clusters with the original cluster that was not chosen for bi-partitioning. Algorithm 2.1 describes this process for obtaining community structures for a given number of clusters.

2.5 Results

In this section, we will evaluate the effectiveness of the proposed Fiedler Consensus Clustering Algorithm for revealing the hierarchical community structure across multiple graphs. The optimal community structure will be identified by maximizing the quality measure U which is the harmonic mean of the homogeneity $\hat{\mathfrak{H}}$ and the completeness scores $\hat{\mathfrak{C}}$ as defined in Section-2.3.

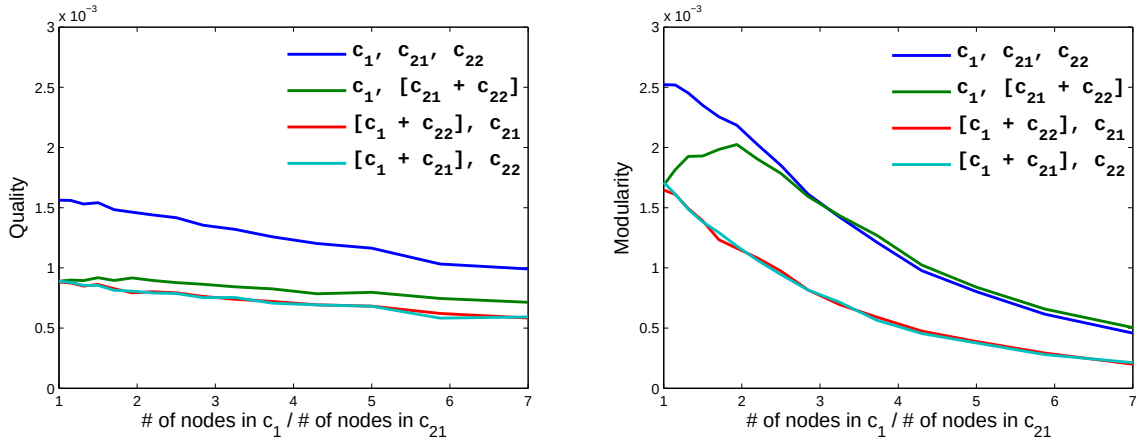
First, we will compare the traditional modularity measure versus the proposed quality measure in determining the optimal number of clusters for the FCCA. Then, we will compare the FCCA to other consensus clustering approaches including averaging, voting and HGPA for different types of

network structure: varying inter-cluster strengths, outliers within a group and overlapping clusters. Finally, the proposed clustering algorithm and the quality measure will be used to identify the community structure which best describes the multivariate relationships across multiple subjects from connectivity graphs obtained from EEG data. For the evaluation of computational complexity, we note that all data analysis has been performed on a 2.4 GHz Intel Core i5 processor running Windows 7.

2.5.1 Quality versus Modularity

Simulated networks consisting of 63 nodes and composed of 3 equal sized clusters were generated 100 times to evaluate the performance of the proposed quality measure, U, against modularity metric for determining the true community structure. The weights of the intra-cluster edges were selected from a truncated Gaussian distribution in $[0, 1]$ with a mean of $\mu_{intra} = 0.6$ and a standard deviation of $\sigma_{intra} = 0.1$. Similarly, inter-cluster edge weights were selected from a truncated Gaussian distribution with a mean of $\mu_{inter} = 0.1$ and a standard deviation of $\sigma_{inter} = 0.2$.

To compare the robustness of the quality metric with the modularity metric for identifying unequal size clusters, the number of nodes in the first cluster (c_1) was gradually increased from 21 to 49 while the sizes of the other two clusters (c_{21} and c_{22}) were decreased from 21 to 7. In order to evaluate the performance of the proposed quality measure and the standard modularity metric, all four possible partitionings of a 3 cluster network are considered, i.e. the three 2-cluster structures (c_1 and c_{21} as one single cluster vs. cluster c_{12}, c_1 and c_{22} as one single cluster vs. cluster c_{21}, and c_{21} and c_{22} as one single cluster vs. c_1) and the true 3 cluster structure. As seen in Fig. 2.2, the proposed quality metric always has its highest value for the true community structure, thus successfully identifying the correct community structure for each test condition while modularity tends to merge small clusters.



(a) Quality (b) Modularity

Figure 2.2: Average of quality metric (U) and modularity (Q) measure corresponding to defined clustering structures in a 3 cluster network with respect to the ratio of the number of nodes in cluster 1 to the number of nodes in the rest of the network over 100 trials.

2.5.2 Evaluation of FCCA for Varying Inter-cluster Strength

100 simulated networks consisting of 64 nodes and composed of 4 communities of equal size were generated 100 times to evaluate the performance of FCCA for varying inter-cluster edge strength, i.e. varying noise levels in the community structure. The weights of the intra-cluster edges were selected from a truncated Gaussian distribution in $[0, 1]$ with a mean of $\mu_{intra} = 0.8$ and a standard deviation of $\sigma_{intra} = 0.1$. The weights of the inter-cluster edges were selected from a truncated Gaussian distribution in with a mean of μ_{inter} and a standard deviation of $\sigma_{inter} = 0.2$. In order to evaluate the algorithms under different inter-cluster connectivity strengths, μ_{inter} was increased gradually from 0.4 to 0.7.

Using the proposed Fiedler Consensus Algorithm, the Averaging method, the Voting method, and HGPA, the networks were evaluated for $2 \leqslant k \leqslant 10$ communities. The best community structure was selected using the proposed quality measure, U. The accuracy of the resulting structure was quantified by Cohen's Kappa statistic which computes the agreement with the true community structure. Overall success was determined by computing the average Kappa value over 100 trials.

As shown in Table-2.2, FCCA and averaging method are more robust than other algorithms for varying inter-cluster edge strengths. These two algorithms accurately identified almost all clusters in all cases for $\mu_{inter} \leqslant 0.7$. Moreover, FCCA is computationally more efficient than voting and HGPA approaches.

Table 2.2: Average Cohen's Kappa and average standard error for identifying community structure in simulated 4-community-networks with varying inter-cluster strength

Method		μ_{inter}			
		0.4	0.5	0.6	0.7
FCCA	κ	0.9992 $\pm 1.4352 \times 10^{-5}$	1 ± 0	1 ± 0	0.9690 ± 0.0040
	time (sec.)	1.1106 ± 0.1415	1.2224 ± 0.2651	1.2709 ± 0.1516	1.4901 ± 0.4153
Ave.	κ	1 ± 0	1 ± 0	1 ± 0	0.9921 ± 0.0014
	time (sec.)	0.1763 ± 0.0219	0.1991 ± 0.0299	0.2131 ± 0.0223	0.2100 ± 0.0347
Voting	κ	0.9968 $\pm 1.9523 \times 10^{-4}$	1 ± 0	0.9905 ± 0.0011	0.8938 ± 0.0073
	time (sec.)	9.1640 ± 1.2854	11.1266 ± 1.0163	14.3203 ± 2.1477	41.6633 ± 4.2908
HGPA	κ	0.8385 ± 0.0092	0. 8576 ± 0.0080	0.9276 ± 0.0050	0.8101 ± 0.0107
	time (sec.)	1.6020 ± 0.3354	1.2207 ± 0.1861	1.2711 ± 0.1441	2.0018 ± 0.4222

2.5.3 Robustness to Outlier Graphs

In a lot of real world settings, the community structure across a population may not always be the same, i.e. there may be outliers in the group. In this subsection, we generate simulations to evaluate the performance of the different clustering methods in the case of outlier graphs. 100 simulated networks consisting of 64 nodes were generated 100 times using two different community structures. The majority of the networks had a 3-cluster structure where nodes 1 to 16 formed the first cluster, nodes 17 to 48 formed the second cluster and the last 16 formed the third cluster. The elements of the connectivity matrices ranged between 0 and 1. The weights of the intra-cluster edges

were selected from a truncated Gaussian distribution with a mean of $\mu_{intra} = 0.6$ and a standard deviation of $\sigma_{intra} = 0.1$. Similarly, inter-cluster edge weights were selected from a truncated Gaussian distribution with a mean of $\mu_{inter} = 0.3$ and a standard deviation of $\sigma_{inter} = 0.2$. Outlier networks were constructed to have 2 communities in which nodes 1 to 32 formed the first cluster and nodes 33 to 64 formed the second cluster. The edge weights were selected from a truncated Gaussian distribution with the intra-cluster edges having a mean of $\mu_{intra} = 0.8$ and a standard deviation of $\sigma_{intra} = 0.1$ while the inter-cluster edge weights had a mean of $\mu_{inter} = 0.1$ and a standard deviation of $\sigma_{inter} = 0.2$. To evaluate the robustness of the algorithms to outliers, the ratio of the outlier networks to the whole group was increased gradually from 15% to 30%.

Similar to the previous section, all networks were evaluated for $2 \leqslant k \leqslant 10$ communities and the best community structure was selected by choosing k which maximizes the quality measure U. Accuracy of the different clustering algorithms was quantified by computing the average Kappa value across 100 trials. As shown in Table 2.3, FCCA and the voting approach are more accurate in identifying the true community structure in the case of outlier graphs. Although the voting approach is more robust against the outliers, its high computational complexity makes the FCCA a useful alternative.

2.5.4 Detecting Overlapping Communities

Communities in a network may not always be distinctly separable from each other and may have an overlapping structure. In order to evaluate the performance of the different clustering algorithms in the case of overlapping communities, 100 simulated networks consisting of 64 nodes and composed of 4 equal size communities were generated 100 times. The weights of the intra-cluster edges were selected from a truncated Gaussian distribution with a mean of $\mu_{intra} = 0.8$ and a standard deviation of $\sigma_{intra} = 0.1$. Similarly, inter-cluster edge weights were selected from a truncated

Table 2.3: Average Cohen's Kappa and average standard error for identifying communities in a group of networks with outliers

Method		Outlier Rate			
		15%	20%	25%	30%
FCCA	κ	1 ± 0	1 ± 0	1 ± 0	0.7143 ± 0.0118
	time (sec.)	0.9672 ± 0.1819	0.9168 ± 0.1163	0.9429 ± 0.1245	1.0095 ± 0.1127
Ave.	κ	1 ± 0	0.7143 ± 0.0118	0.7143 ± 0.0118	0.7143 ± 0.0118
	time (sec.)	0.1716 ± 0.0313	0.1739 ± 0.0254	0.1771 ± 0.0281	0.1627 ± 0.0206
Voting	κ	1 ± 0	1 ± 0	1 ± 0	0.9304 ± 0.0031
	time (sec.)	9.0646 ± 0.9299	8.9715 ± 0.5380	9.0036 ± 0.7468	10.4778 ± 2.7145
HGPA	κ	0.5177 ± 0.0149	0.5164 ± 0.0150	0.4848 0.0154	0.4554 ± 0.0158
	time (sec.)	1.6271 ± 0.1875	1.5554 ± 0.1626	1.2809 ± 0.1719	1.2043 ± 0.0924

Gaussian distribution with a mean of $\mu_{inter} = 0.1$ and a standard deviation of $\sigma_{inter} = 0.2$. To provide overlap between communities, a subset of the inter-cluster edges between each pair of communities was selected from the same distribution as the intra-cluster edges. The number of strong inter-cluster edges was increased gradually from 75 to 125.

As in previous simulations, all networks were evaluated for $2 \leqslant k \leqslant 10$ communities and the optimal k was selected based on the maximizing the quality measure. Overall success was determined by computing the average Kappa value over 100 trials. As seen in Table 2.4, Fiedler Consensus Clustering Algorithm is more robust to overlapping communities in the network compared to the other methods.

2.5.5 Community Structure of the Brain During Error-Related Negativity

The time-varying phase synchrony measure is applied to a set of EEG data containing the error-related negativity (ERN) [173, 174]. The ERN is an event-related potential that occurs following

Table 2.4: Average Cohen's Kappa and average standard error for identifying overlapping clusters in simulated networks

Method		# of Inter-Cluster Edges		
		75	100	125
FCCA	κ	0.9964 $\pm 3.9437 \times 10^{-4}$	0.9887 ± 0.0013	0.9539 ± 0.0048
	time (sec.)	1.3631 ± 0.3329	1.2449 ± 0.2767	2.5533 ± 1.0191
Ave.	κ	0.9754 ± 0.0025	0.9424 ± 0.0052	0.8825 ± 0.0080
	time (sec.)	0.2186 ± 0.0560	0.2214 ± 0.0577	0.3817 ± 0.1246
Voting	κ	0.9938 $\pm 6.1236 \times 10^{-4}$	0.9694 ± 0.0022	0.9444 ± 0.0047
	time (sec.)	11.8611 ± 1.9694	15.0078 ± 2.8988	26.5411 ± 6.1363
HGPA	κ	0.8682 ± 0.0074	0.8374 ± 0.0094	0.7732 ± 0.0117
	time (sec.)	1.5070 ± 0.2811	1.5480 ± 0.2081	1.7893 ± 0.2913

performance errors in a speeded reaction time task. Previously reported EEG data [175] from 63-channels (10/20 system) were utilized. This included 91 undergraduate students (34 male) from the University of Minnesota (one of the original 92 participants were dropped due to artifacts rendering computation of the time-frequency phase synchrony (TFPS) values problematic). Full methodological details of the recording are available in the previous report [175]. The task was a common speeded-response letter (H/S) flanker, where error and correct response-locked trials from each subject were utilized. A random subset of correct trials was selected, to equate the number of error relative to correct trials for each participant. The EEG data are pre-processed by the spherical spline current source density (CSD) waveforms to sharpen event-related potential (ERP) scalp topographies and reduce volume conduction [176]. The CSD has fewer assumptions than many inverse transforms, attenuates volume conduction, and represents independent sources near the cortical surface [177]. Our previous work indicates that there is increased phase synchrony associated with ERN for the theta frequency band (4-7 Hz) and ERN time window (25-75 ms) for

Error responses compared to Correct responses [148]. For each subject and response type, the pairwise average phase locking value within the ERN time window and theta frequency band was computed using Equation (4) across trials yielding a 63×63 connectivity matrix indicating the average synchrony between brain regions.

First, the proposed Fiedler consensus clustering approach, averaging and voting methods were applied to the set of Error and Correct data in order to identify an optimal community structure. Hierarchical decomposition of the networks were evaluated for $2 < k < 15$ and the optimum k was selected by maximizing the quality metric U. The clustering results can be seen in Figs. 2.3 (FCCA) and 2.4 (averaging and voting methods, Figs. 4a and 4b, respectively). For FCCA, Error responses were best represented by a structure composed of 10 communities (Fig. 2.3a), and Correct responses with 7 communities (Fig. 2.3b), while Averaging and Voting methods identify 9 communities for Error responses and 5 communities for Correct responses (Fig. 2.4). As it can be seen, the averaging method yields two large clusters for both error and correct conditions unable to discriminate between error and correct responses and resolve the different subnetworks. Similarly, the voting method yields a large cluster for both error and correct responses with a couple of small frontal and lateral subnetworks. The proposed method, on the other hand, provides a more detailed view of the network separating the medial and lateral clusters from each other.

The obtained clusters from the three consensus clustering approaches are evaluated on each subject's network to quantify the agreement between the common cluster structure and each subject's connectivity graph. This agreement is quantified through the quality metric, U. As seen in Table 2.5, the community structure obtained by FCCA is more appropriate for each subject and yields a statistically significant higher score for both Error and Correct conditions $(p < 0.025)$.

From Fig. 2.3, we can see that the clusters identified by FCCA are more segregated and differentiated for Errors relative to Correct responses. For example, in the Correct condition one large

Table 2.5: Mean and Standard Deviation of quality metric (U) computed to quantify the consistency between the group's community structure and individual subjects' community structure for each subject, each response type and the three consensus clustering methods

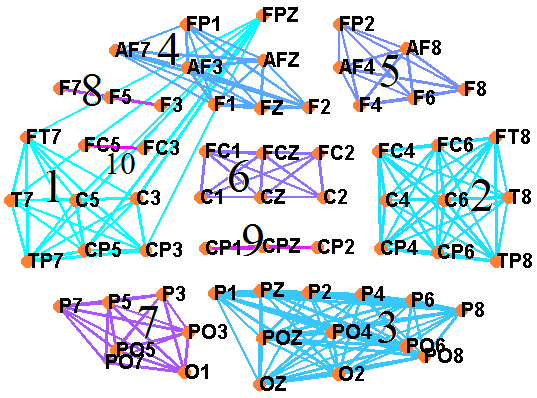
		Method		
		Averaging	Voting	FCCA
U	Error	0.0483 ± 0.0031	0.0524 ± 0.0040	0.0567 ± 0.0054
	Correct	0.0454 ± 0.0050	0.0426 ± 0.0054	0.0504 ± 0.0042

Table 2.6: Computation time for community structures obtained by the FCCA, Averaging and Voting methods

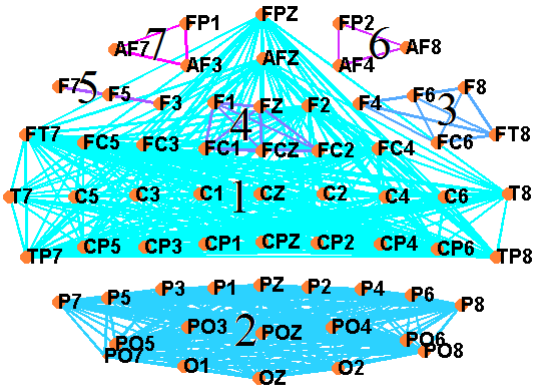
		Method		
		Averaging	Voting	FCCA
time	Error	2.7104	198.1623	11.8576
(sec.)	Correct	2.3326	212.5764	13.7596

cluster (1) accounts for the majority of prefrontal and motor regions, with a small cluster (5) consistent with separable activity in left-PFC regions. For Errors, on the other hand, separable clusters are apparent relative to left (1) and right (2) motor areas, and left (4, 8) and right (5) lateral-PFC regions (consistent with *a priori* hypotheses). Interestingly, one cluster in the Error condition (6) and one in the Correct (4) center on medial-frontal sites including FCz and Cz (6), consistent with the time-domain ERN and correct-related negativity (CRN) component topographies, respectively. Activity in parietal-occipital regions was characterized with similar clusters for both Correct (2) and Error conditions (3, 7).

An overall statistical assessment of the inter-modular relationships revealed that the grand mean was significantly greater for the Error relative to Correct conditions $(t(90) = 2.16, p < .033)$, while the same comparison for intra-modular pairs was not $(t(90) < .5)$. This provides support for the inference of increased functional connectivity related to error processing relative to correct. Next, to provide detailed information about these relationships, average intra-modular and inter-modular synchronies were computed for Correct and Error communities (presented in Fig. 2.5). These maps illustrate the amount of integration between different clusters. To provide statistical



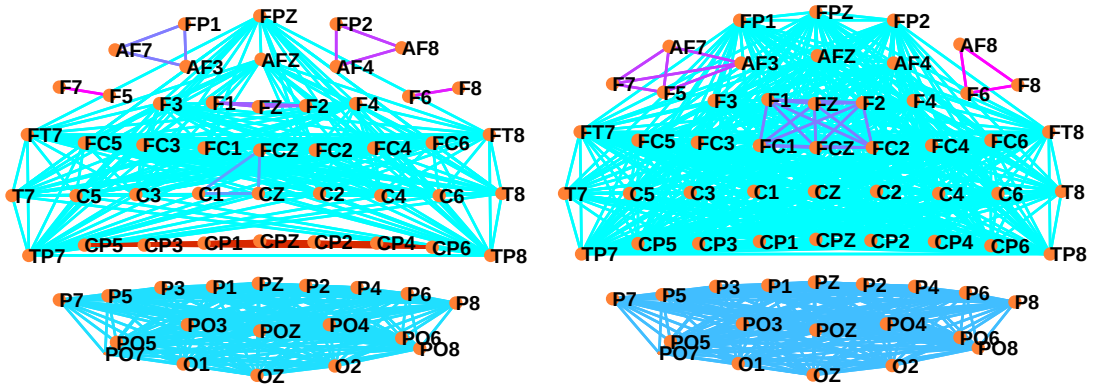
(a) Error



(b) Correct

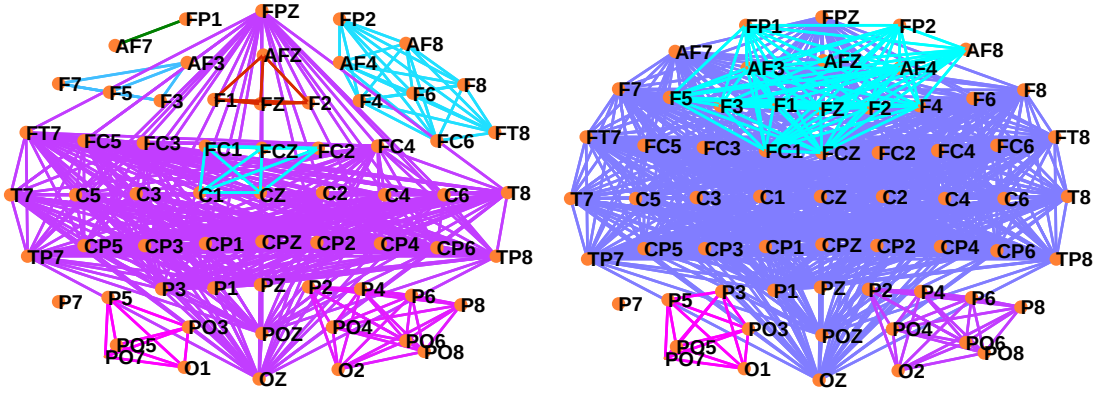
Figure 2.3: Cluster structures obtained by FCCA a) Error (k = 10); b) Correct (k = 7).

assessment of the inter-modular relationships within the Error and Correct conditions, t-tests were performed for each inter-modular bivariate pair relative to the grand mean of the inter-modular pairs. Resulting *t*-values and Bonferroni corrected *p*-values are presented in Figs. 3 b and c, respectively. While it was not appropriate to directly compare Error-Correct differences between individual clusters (as they were derived from separate cluster analyses), several observations about the individual clusters within Error or Correct conditions provide some interesting information at this level of analysis. First, motor-related clusters in the Error condition (1, 2) were significantly more related to each other than the across cluster average. Next, occipital clusters in both the Error (3, 7) and Correct (2) conditions evidenced decreases relative to the mean, suggesting a



Error Correct

(a) Averaging



Error Correct

(b) Voting

Figure 2.4: Cluster structures obtained by a) Averaging, Error (k = 9) and Correct (k = 5); b) Voting, Error (k = 9) and Correct (k = 5).

decrease in connectivity with visual processing areas during the ERN and CRN. For lateral-PFC regions, the smaller left-laterized clusters (8 and 5, respectively for error and correct conditions) were significantly associated with significantly increased connectivity with prefrontal areas and decreased connectivity with parietal occipital areas. Another *a priori* effect of interest was that both Error and Correct medial-frontal clusters (6 and 4, respectively) showed significant increases with left lateral-PFC clusters (4 and 5, respectively).

Fig. 2.6 illustrates the hierarchical structure obtained from FCCA for both response types. For

the error response, the initial partition yields one large frontal and one parietal cluster. Further decomposition provides the detailed construction of the frontal cluster. Similarly, for the correct response, the initial partition yields one frontal and one parietal partition with the subsequent partitions decomposing the frontal cluster into smaller sub-networks.

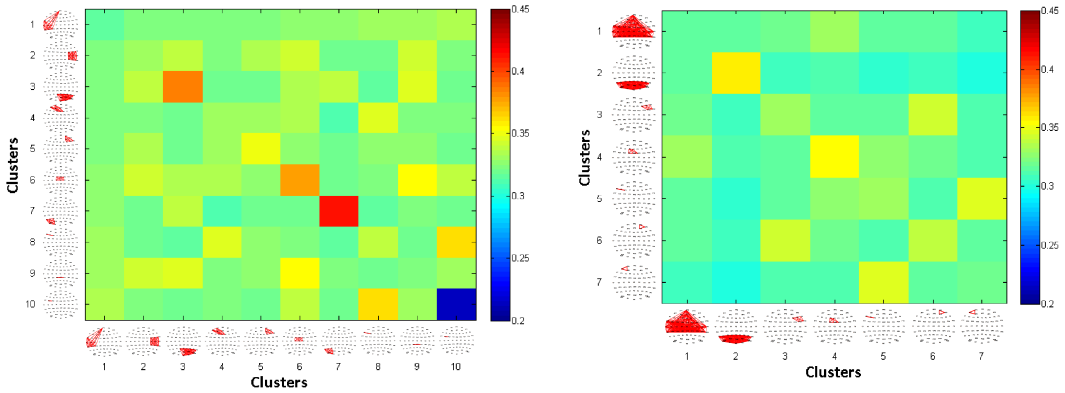
2.6 Conclusions

In this chapter, we proposed a new graph theoretic community detection approach to provide a detailed view of the organizational structure underlying the functional brain connectivity network through EEG recordings across multiple subjects. The main contributions include the hierarchical implementation of Fiedler vector based graph clustering, the introduction of an accurate and computationally efficient consensus clustering approach, the introduction of a new information-theoretic cluster quality measure, U, and a detailed study of the brain network involved in error processing.

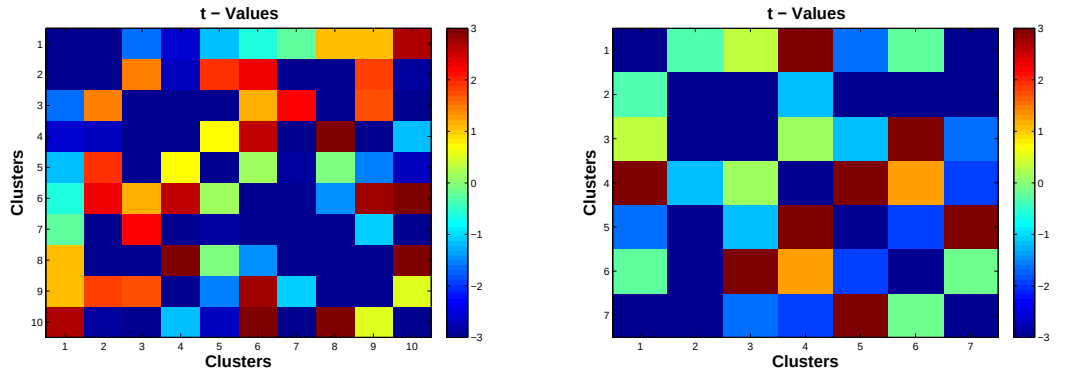
First, the well-known Fiedler vector based graph bi-partitioning method has been implemented to obtain a hierarchical decomposition of the functional connectivity networks. This hierarchical implementation is supported by previous work that suggests a hierarchical structure for functional connectivity networks [138–140]. Second, the proposed partitioning approach is modified to account for multiple subjects by first obtaining an initial bipartition of each subject's connectivity network and then by iteratively partitioning the co-occurrence matrix across subjects. As shown through simulations, FCCA is computationally more efficient than voting and is more accurate than averaging in the case of outliers and overlapping community structures. Moreover, the application of FCCA to EEG data produced clusters consistent with published work [178, 179], whereas voting and averaging methods failed to partition the frontal cluster into physiologically meaningful lateral

and medial frontal communities. Finally, a new cluster quality measure U based on optimizing the tradeoff between maximizing the divergence between clusters and minimizing the entropy of individual clusters was introduced to select the optimal number of clusters. This measure provides an alternative to the standard modularity measure which is known to fail for unequal cluster sizes and weighted networks [180].

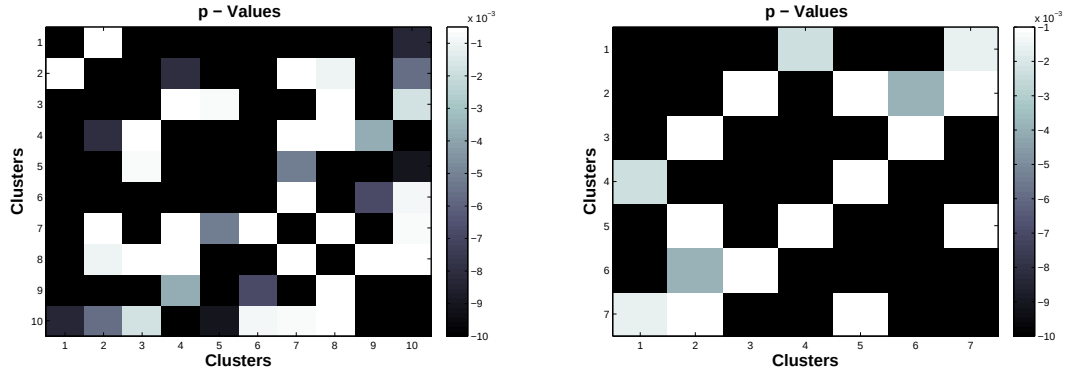
Future work will consider exploring single [181] and distributed dipole [182] source solutions to the inverse problem for extending this approach to the source domain.



(a)

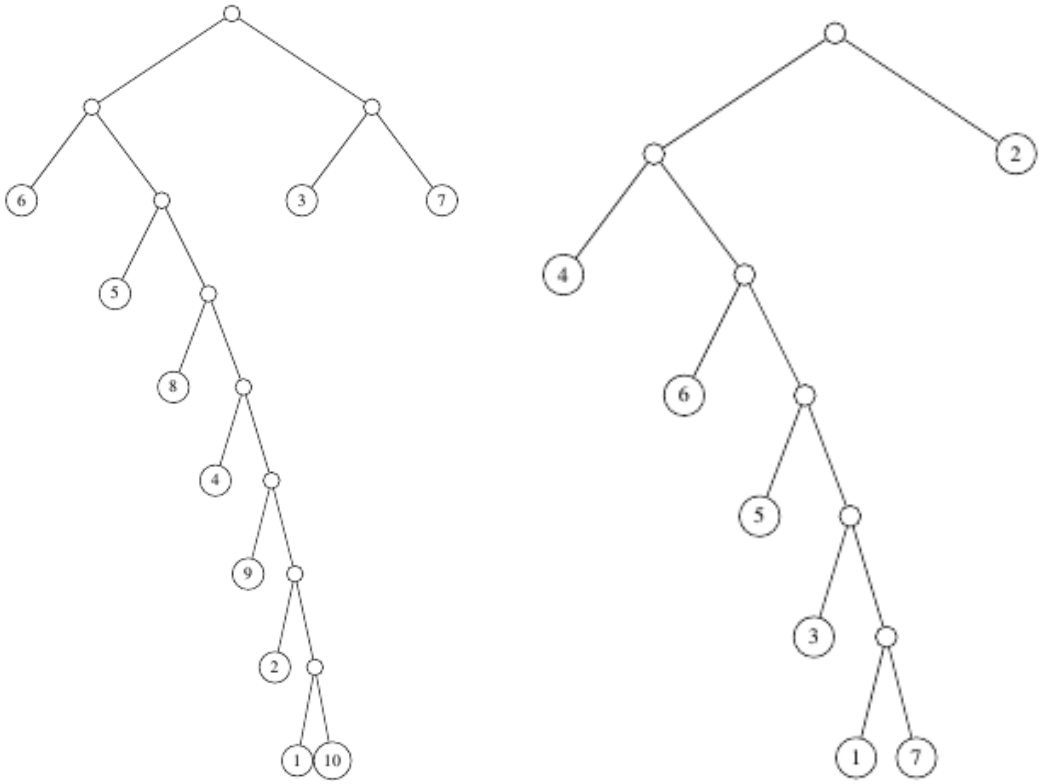


(b)



(c)

Figure 2.5: (a) Average phase synchrony between clusters for Error response (k = 10), and Correct response (k = 7). (b) *t*-values for inter-modular bivariate pair relative to the grand mean of the inter-modular pairs of ERN and CRN conditions and (c) corresponding *p*-values.



(a) (b)

Figure 2.6: Hierarchical structure for obtained by FCCA a) Error; b) Correct responses.

Chapter 3

Recursive Robust Low-rank + Sparse Structure Learning for Dynamic Tensors

3.1 Introduction

Advanced functional imaging techniques such as EEG and functional magnetic resonance imaging (fMRI) have enabled the study of the neuronal mechanisms underlying cognition in detail. These studies have revealed that transient synchronization, referred to as functional connectivity (FC), between spatially distributed neural populations is responsible for human cognition, perception and emotion [114, 183, 184]. Recently, tools from graph theory have been employed to analyze the functional connectivity of the brain by associating nodes with distinct brain regions and edges with pairwise interactions between them [105, 106]. Most of the current work on functional connectivity network analysis focuses on static networks where the networks correspond to average activity over a time and frequency window of interest. However, recent studies have shown that functional connectivity networks change dynamically in short time scales and exhibit task-related patterns [107–111]. This continuous formation and destruction of functional connectivity also controls the emergence of a unified neural process in cognition, perception and memory [111, 185, 186].

To better understand the brain dynamics, early studies focused on extracting graph theoretic

measures across time for a time-varying analysis of FC graphs [185, 187, 188]. For example, Valencia et al. [189] showed how the small-world structure of FC networks evolve during a visual stimulus. Similarly, Fallani et al. [190] presented a graph theoretical approach for FC networks to identify persistent edges during a motor task. Recently, dynamic FC network (dFCN) tracking approaches have been combined with network state estimation techniques. Allen et al. [191] assume that the FC network at each time point is at a distinct network state where the network states are determined through k-means clustering of dFCNs across time and subjects from resting state fMRI data. Similarly, Dimitriadis et al. [192] introduced FC microstates inspired by the EEG microstate literature [193]. In [194], the network states are obtained through clustering and Markov modelling to identify both FC-states and the transitions between them. Similarly in [195], the network states are identified by evolutionary clustering applied to FC edge timeseries. An alternative approach to dFCN analysis assumes that network states are made up of multiple building blocks. Leonardi et al. [196] propose a principal component analysis (PCA) based approach to reveal the intrinsic FC patterns named as eigenconnectivities, and describe the FC matrices as weighted sum of eigenconnectivities. Similarly, different PCA based approaches have been used to identify dynamics of both resting-state [197] and task-based EEG [198]. More recently, [199] compared clustering and SVD based approaches to identify task related and resting state FC dynamics and showed that FC patterns obtained from an SVD based approach better represent the task-based dynamics, while patterns obtained from a clustering based approach are more suitable to identify resting state FC dynamics.

In addition to approaches focused on unraveling the network states from dFCNs, recently introduced methods have also focused on detection of change points where the network structure considerably changes. Cribben et al. [200] presented a data-driven technique which first detects the temporal change points by a greedy partitioning scheme and then estimates a connectivity

graph for FC patterns in each temporal partition. Zhang et al. [201], presented dynamic Bayesian variable partition model which simultaneously identifies the state transitions and learns significant FC patterns in each state. Similarly, Ou et al. [202] proposed a Bayesian model to detect change points from functional brain interactions and evaluated the network structure using nonnegative matrix factorization within each temporal segment. However, all of these approaches make use of time-series data and are not based on the dFCNs constructed directly from data. Moreover, these approaches assume a multivariate Gaussian model for the underlying time series data and require individual analysis of each subject before inferring the group's network structure. Finally, these methods are computationally expensive as they rely either on greedy search or probabilistic metrics.

In this chapter, we propose a tensor based representation of dFCNs for tracking and summarizing the functional connectivity across time and subjects from task-based EEG data. First, we introduce a tensor subspace analysis method for robust low-rank + sparse structure recovery. This is motivated by the fact that FCNs are known to have a modular structure which translates to a low-rank connectivity matrix [151]. In the case of dFCNs, these low-rank structures can be assumed to change slowly in time, similar to EEG microstates [193], which can be described as a slowly changing subspace. Conventional subspace analysis methods such as PCA and SVD cannot deal with higher order data and existing tensor decomposition methods such as higher-order SVD (HoSVD) and parallel factor analysis (PARAFAC) are not robust to sparse outliers or noise in the data [44, 73, 203, 204]. The proposed recursive framework separates the low-rank part of the data from sparse noise components by identifying change points and updating the estimates of low-rank subspaces. Identified change points corresponding to the subspace change along the connectivity mode of the tensor are used to define time intervals of interest. The low-rank tensor within each time interval is then summarized through a recently introduced multiple network

clustering approach known as Fiedler Consensus Clustering Approach (FCCA) [205]. Finally, the proposed framework is applied to dFCNs constructed from EEG data collected during a study of error-related negativity.

The proposed framework offers three major contributions to both the literature in online tensor subspace tracking and dFCN analysis. First, unlike most of the current work which focuses on the dynamics of resting state fMRI networks, we focus on the dynamics of task related networks from EEG data. As EEG data has high temporal resolution and the data considered in this chapter is response locked, determining the actual time points where significant changes to network structure occurs is highly relevant. The proposed framework offers a way to determine time intervals during which the FCN has a common quasi-stationary pattern across time and subjects, i.e. slowly changing subspace structure similar to microstates [193]. Second, most of the current work reduces the high dimensionality of the dFCNs by vectorizing the connectivity matrices into long vectors before identifying the network states. This approach does not preserve the topological structure of the network. In the proposed work, we address this problem by keeping the network structure of FCNs intact by using tensor representations. Through tensor representation, we can capture the variability common to all subjects across time. Finally, the proposed low-rank plus sparse structure learning algorithm for tensors offers a novel way of recovering a low-rank subspace estimate along each mode of the data where the rank is defined through the Tucker rank. This rank definition is directly related to the modular structure of FCNs . The proposed approach separates the low-rank part of the data from sparse noise components along time and then summarizes the network within each time interval through clustering the extracted low-rank networks within the time interval. This yields better structure information as it is equivalent to denoising the networks.

3.2 Background

3.2.1 Robust Principal Component Analysis

High dimensional data mostly lies in a lower dimensional subspace and principal component analysis (PCA) is the most widely used technique to identify this lower dimensional subspace. Recently, PCA has been used for identifying network states from dynamic functional connectivity networks [196, 198]. However, it is known that PCA suffers from non-Gaussian corruptions and may find a completely wrong principal subspace in the presence of even a few outliers. These drawbacks have forced researchers to develop more robust subspace estimation techniques which is a significantly more difficult problem than standard PCA [206, 207].

Since the recent work by Candes et al. and Chandrasekharan et al. [208, 209], the general problem of separating a sparse matrix and a low-rank matrix from their sum has received a lot of attention. The final goal usually is to either find the column span of the low-rank matrix or the support of the sparse one. This is now commonly referred to as the "low-rank + sparse recovery" problem. There has been a large amount of recent work on batch methods for low-rank + sparse recovery and its various extensions including Principal Component Pursuit, Outlier Pursuit and Low-Leverage Decomposition [49–62, 209].

One of the well-known robust PCA methods is principal component pursuit (PCP) which assumes that the data matrix $\mathbf{M}$ has a low-rank part $\mathbf{L}$ and a sparse noise or outlier $\mathbf{S}$ as [208]:

$$\mathbf{M} = \mathbf{L} + \mathbf{S}. \tag{3.1}$$

It was shown that $\mathbf{L}$ can be efficiently estimated by solving the following optimization problem:

$$\begin{aligned} min \quad & \| \mathbf{L} \|_* + \lambda \| \mathbf{S} \|_1 \\ s.t. \quad & \mathbf{L} + \mathbf{S} = \mathbf{M}, \end{aligned} \tag{3.2}$$

where λ is the regularization parameter and $\| \mathbf{L} \|_* = \sum_{i=1}^{r} \sigma_i(\mathbf{L})$ denotes the nuclear norm where σ_i's are the first r singular values of the matrix $\mathbf{L}$. This problem has been solved by using convex optimization approaches, i.e. augmented Lagrange multiplier algorithm [208], accelerated proximal gradient approach [210].

In order to reduce the computational complexity and to achieve online subspace tracking, various approaches have also been proposed to solve the RPCA problem, i.e. GRASTA, PETRELS and REPROCS [63, 66, 67, 70, 204, 211–214]. These approaches first identify the subspace that the low rank data lies in, then recovers incoming low-rank measurement vectors from missing entries by considering this subspace information.

A recently introduced algorithm REPROCS recursively separates the low-rank part from sparse noise as follows. Let $\mathbf{M}_t \in \mathbb{R}^{n \times 1}$ be a time-series of measurement vectors written as $\mathbf{M}_t = \mathbf{L}_t + \mathbf{S}_t$ where $\mathbf{L}_t$ is the low-rank part which lies in a subspace spanned by $\mathbf{P}_t$ and $\mathbf{S}_t$ is the sparse noise vector. Let $\hat{\mathbf{P}}_t$ be an accurate estimate of the r-dimensional basis $\mathbf{P}_t$ at time t and $\hat{\mathbf{P}}_{t,\perp}$ be the orthogonal complement of $\hat{\mathbf{P}}_t$. Let $\alpha_t := \hat{\mathbf{P}}_t' \mathbf{L}_t$ be the projection of $\mathbf{L}_t$ onto $\hat{\mathbf{P}}_t$ and $\beta_t := (\hat{\mathbf{P}}_{t,\perp})' \mathbf{L}_t$ be a projection of $\mathbf{L}_t$ onto $\hat{\mathbf{P}}_{t,\perp}$. Then, $\mathbf{M}_t$ can be rewritten as $\mathbf{M}_t = \hat{\mathbf{P}}_t \alpha_t + \hat{\mathbf{P}}_{t,\perp} \beta_t + \mathbf{S}_t$. REPROCS first projects the measurement vector onto $\hat{\mathbf{P}}_{t,\perp}$ to approximately nullify the low-rank part $\mathbf{L}_t$. As $\mathbf{y}_t := (\hat{\mathbf{P}}_{t,\perp}')\mathbf{M}_t$, where $\mathbf{y}_t$ can be rewritten as $\mathbf{y}_t = (\hat{\mathbf{P}}_{t,\perp}')\mathbf{S}_t + \beta_t$, and the dimension of the projected data vector reduces to $n - r$. Since projecting $\mathbf{M}_t$ onto $\hat{\mathbf{P}}_{t,\perp}'$ nullifies the contribution of $\mathbf{L}_t$, β_t can be interpreted as small noise. Therefore, solving for n-dimensional $\mathbf{S}_t$ from $(n-r)$-dimensional $\mathbf{y_t}$ becomes a traditional sparse recovery problem. Once $\hat{\mathbf{S}}_t$ is recovered, $\mathbf{L}_t$ can be estimated as $\hat{\mathbf{L}}_t = \mathbf{M}_t - \hat{\mathbf{S}}_t$. Performance of this algorithm highly

depends on the correctness of the estimated low-rank subspace and the slowly changing subspace assumption [67, 204, 213, 214]. However, this method is limited to vector type measurements, and cannot be applied directly to higher order datasets such as tensors. In this chapter, we will present an extension of REPROCS to tensor type data.

Recently, tensor-based approaches have been proposed to track dynamic tensor subspaces such as dynamic tensor analysis, streaming tensor analysis and window based tensor analysis [71, 72]. However, these approaches provide computationally efficient frameworks for analysis of streaming datasets by recursively updating subspace information and do not address the robustness of the subspace estimates. Goldfarb and Qin extended robust PCA to tensors (HoRPCA) by solving low-rank + sparse recovery problem for general higher order tensors [73]. However, this method is highly computationally expensive and does not update the subspaces online, i.e. would not be useful for tracking subspace changes across time. Li et al. [74] presented a robust subspace learning algorithm (RTSL) that incrementally updates the tensor subspace. Moreover, Nion et al. [75] proposed two adaptive approaches to track PARAFAC decompostion of 3-way tensors. These approaches suggest to update the PARAFAC decomposition at every time point based on simultaneous diagonalization or minimization of weighted least squares criterion. More recently, Mardani et al. [76] proposed an online subspace learning method based on nuclear norm minimization and extended this approach for matrices and higher order datasets. Extension of this algorithm to tensors takes advantage of PARAFAC model to minimize tensor rank and considers temporal information as one of the tensor modes. Similar to [76], OLSTEC proposed in [77] also tracks the subspace of partially observed higher-order data using PARAFAC decomposition.

3.2.2 Tensor Algebra & Tensor Decompositions

An order N tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ where $x_{i_1,\, i_2,... \, i_N}$ corresponds to the $(i_1,\ i_2, ...,\ i_N)$th element of the tensor $\mathcal{X}$. Vectors obtained by fixing all indices of the tensor except the one that corresponds to n^{th} mode are called mode-n fibers.

Mode-n product The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times ... I_n \times ... \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted as $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} = (\mathcal{Y})_{i_1, i_2, ..., i_{n-1}, j, i_{n+1}, ..., i_N} = \sum_{i_n=1}^{I_n} x_{i_1, ..., i_n, ..., i_N} u_{j, i_n}$ and is of size $I_1 \times ... \times I_{n-1} \times J \times I_{n+1} \times ... \times I_N$.

Tensor matricization Process of reordering the elements of the tensor into a matrix is known as matricization or unfolding. The mode-n matricization of tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times ... I_n \times ... \times I_N}$ is denoted as $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{i \in \{1,...,N\}/\{n\}} I_i}$ and is obtained by arranging mode-n fibers to be the columns of the resulting matrix. Unfolding the tensor $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 ... \times_N \mathbf{U}_N$ along mode-n is equivalent to $\mathbf{Y}_{(n)} = \mathbf{U}_n \mathbf{X}_{(n)} (\mathbf{U}_N \otimes ... \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} ... \otimes \mathbf{U}_1)^\top$, where $\otimes$ is the matrix Kronecker product.

The n-Rank Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ be an N-way tensor, the *n-rank* of $\mathcal{X}$ is the collection of rank of mode matrices $\mathbf{X}_{(n)}$ and is denoted as:

$$rank_n(\mathcal{X}) = \left\{ rank(\mathbf{X}_{(1)}),\ rank(\mathbf{X}_{(2)}), ...,\ (\mathbf{X}_{(n)}) \right\}, \tag{3.3}$$

where $n = 1, 2, ..., N$.

Tucker decomposition Tucker decomposition is a form of higher order SVD. Any tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ can be decomposed as mode products of a core tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and N

mode matrices $\mathbf{U}_{(n)} \in \mathbb{R}^{I_n \times I_n}$ [215] [35].

$$\begin{aligned} \mathcal{X} &= \mathcal{S} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \ldots \times_N \mathbf{U}_{(N)}, \\ \mathcal{X} &= \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} s_{i_1,i_2,\ldots,i_N} c_{i_1}(\mathbf{U}_{(1)}) \circ \ldots \circ c_{i_N}(\mathbf{U}_{(N)}), \end{aligned} \tag{3.4}$$

where the matrix $\mathbf{U}_{(n)}$ contains the left singular vectors of $\mathbf{X}_{(n)}$ and $c_n(\cdot)$ is n^{th} column vector and $\mathcal{S}$ is obtained by $\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_{(1)}^\top \times_2 \mathbf{U}_{(2)}^\top \ldots \times_N \mathbf{U}_{(N)}^\top$.

3.2.3 Time-Varying Measure of Phase Synchrony

In this chapter, pairwise functional connectivity will be quantified using a recently introduced time-frequency phase estimation method based on Reduced Interference Rihaczek distribution (RID-Rihaczek) [148, 149]. Phase synchronization within different frequency bands across the brain has been shown to be a plausible mechanism explaining neuronal integration [114, 144].

Reduced Interference Rihaczek Distribution (RID-Rihaczek) is given by:

$$C(t,\omega) = \iint \exp\left(\frac{-(\theta\tau)^2}{\sigma}\right) \exp\left(j\frac{\theta\tau}{2}\right) A(\theta,\tau) e^{-j(\theta t+\tau\omega)} d\tau d\theta, \tag{3.5}$$

where $\exp(-(\theta\tau)^2/\sigma)$ is the Choi-Williams kernel used to filter out the cross-terms, $A(\theta,\tau) = \int x(u+\frac{\tau}{2}) x^*(u-\frac{\tau}{2}) e^{j\theta u} du$ is the ambiguity function of the signal $x(t)$ and $\exp(j\theta\tau/2)$ is the kernel corresponding to the Rihaczek distribution [150]. The phase difference between two signals, x_i and x_j, based on this complex distribution is computed as

$$\Phi_{ij}(t,\omega) = \arg\left[\frac{C_i(t,\omega) C_j^*(t,\omega)}{|C_i(t,\omega)||C_j(t,\omega)|}\right], \tag{3.6}$$

where $C_i(t,\omega)$ and $C_j(t,\omega)$ refer to the complex energy distributions of the two signals $x_i(t)$ and $x_j(t)$ respectively. A synchrony measure quantifying the intertrial variability of the phase differences, phase locking value (PLV), is defined as

$$PLV_{i,j}(t,\omega) = \frac{1}{\kappa}\left|\sum_{k=1}^{\kappa}\exp(j\Phi_{ij}^{k}(t,\omega))\right|, \tag{3.7}$$

where κ is the number of trials and $\Phi_{ij}^{k}(t,\omega)$ is the time-varying phase estimate between two signals recorded at electrodes i and j for the kth trial. If the phase difference varies little across the trials, PLV is close to 1. Compared to the existing synchrony measures, RID-Rihaczek based phase synchrony measure is more robust to noise, and has uniformly high time-frequency resolution with less bias [148].

In this chapter, we construct the functional connectivity matrices at each time point and for each subject as:

$$G_{i,j}(t) = \frac{1}{\omega_b - \omega_a}\sum_{\omega=\omega_a}^{\omega_b} PLV_{i,j}(t,\omega), \tag{3.8}$$

where the entries of the connectivity networks are computed as the average synchrony between pairs of nodes at time t averaged over a frequency band of interest which is the theta band. Once the individual connectivity matrices are constructed, a three-way tensor $\mathcal{X}_t \in \mathbb{R}^{N\times N\times S}$ is formed at each time point across all subjects as:

$$\mathcal{X}_t(i,j,s) = G_{i,j}^{s}(t) \tag{3.9}$$

where $i,j \in \{1,\ 2,..,\ N\}$ correspond to the nodes or brain regions in the network and $s \in \{1,\ 2,..,\ S\}$ is the subject.

3.2.4 Consensus Clustering and Fiedler Consensus Clustering Approach

In neuroscience problems, it is desirable to find a common network structure across subjects performing the same task or in the same population [216, 217]. Recently, we have introduced Fiedler Consensus Clustering Algorithm (FCCA) to address this issue to obtain a common community structure across multiple weighted graphs, where the graphs correspond to individual functional connectivity networks discussed in Section 3.2.3. [205]. One of the most common ways to partition a graph is spectral clustering. Spectral clustering generally uses the eigenvectors of the Laplacian matrix computed as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix of the graph and $\mathbf{D}$ is the degree matrix containing degrees of nodes along the diagonal with $D(i,i) = \sum_{j=1,\, j\neq i}^{N} A(i,j)$. The eigenvector corresponding to the second smallest non-zero eigenvalue of the Laplacian matrix provides the optimal minimal cut of a graph and this eigenvector is referred to as the Fiedler vector [158].

In FCCA, the original connectivity matrices are bi-partitioned into two clusters using the Fiedler partitioning method. This results in a cluster matrix for the r^{th} network $\mathbf{T}^r$ such that

$$\mathbf{T}^r(i,j) = \begin{cases} 1 & \text{if nodes } v_i, v_j \text{ are in the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

and $r = \{1, 2, ..., m\}$ where m is the number of networks. In order to find the common community structure across multiple graphs, we introduce a co-occurence matrix $\mathbf{W}$ where $W(i,j) = \sum_{r=1}^{m} \frac{\mathbf{T}^r(i,j)}{m}$ and $W(i,j) \in [0,1]$. $W(i,j)$ is the probability that a pair of nodes are members of the same cluster across multiple graphs. The adjacency matrix reflects the strength of a direct relationship between a node pair, whereas $\mathbf{W}$ reflects the likeliness that a pair of nodes are in the same cluster across all subjects.

The Laplacian matrix of $\mathbf{W}$ is computed and the Fiedler vector is found to form a bi-partition of $\mathbf{W}$ into a community structure composed of clusters c_1 and c_{-1}. Since $\mathbf{W}$ represents the probability that a node pair should be clustered together, the Fiedler partition of $\mathbf{W}$ represents the community structure common to all graphs. The initial partition set, $C = \{c_1, c_{-1}\}$, contains 2 clusters but if $k > 2$ is desired, the process can be repeated by selecting a cluster in C to partition. In this case, c_1 or c_{-1} is selected based on the quality score of each cluster (see [205] for more details on the particular quality score used in FCCA).

Next, sub-matrices are extracted from the original connectivity matrices such that they only contain the nodes of the chosen cluster. These sub-matrices are used to derive the new sub co-occurence matrix, $\mathbf{W}^y$, where $y = 1$ if cluster c_1 was selected and $y = -1$ if cluster c_{-1} was selected. The Fiedler partitioning is performed on the selected cluster to obtain two new clusters, c_1^y and c_{-1}^y. The final cluster set $C = \{c_{-y}, c_1^y, c_{-1}^y\}$ is a concatenation of the two new clusters with the original cluster that was not chosen for bi-partitioning. This method can be iterated until an optimal quality score or a desired number of clusters is achieved.

3.3 Higher-order Recursive Low-Rank + Sparse Structure Learning (Ho-RLSL)

3.3.1 Problem Statement

In this chapter, we will represent dynamic functional connectivity networks across subjects as a three-way dynamic tensor $\mathcal{M}_t \in \mathbb{R}^{N_1 \times N_2 \times N_3}$. We will assume that this tensor has a low-rank structure $\mathcal{L}_t$ with $rank(\mathbf{L}_t^{(i)}) \ll min(N_i, \prod_{k=1,k\neq i}^{3} N_k)$ along the connectivity and subject modes

corresponding to the modular network structure plus some sparse outlier connections $\mathcal{S}_t$ as:

$$\mathcal{M}_t = \mathcal{L}_t + \mathcal{S}_t. \tag{3.11}$$

Our goal is to separate the low-rank tensor $\mathcal{L}_t$ from its noisy version $\mathcal{M}_t$. This goal leads to the following optimization problem where the Tucker rank of $\mathcal{L}_t$ is minimized while simultaneously minimizing the l_1-norm of the noise part, $\mathcal{S}_t$ [73]:

$$\begin{aligned} &\min_{\mathcal{L}_t,\, \mathcal{S}_t} \ \| \mathcal{L}_t \|_* + \lambda \| \mathcal{S}_t \|_1 \\ &s.t.\ \mathcal{L}_t + \mathcal{S}_t = \mathcal{M}_t, \end{aligned} \tag{3.12}$$

where $\| \mathcal{L}_t \|_*$ is the nuclear norm of the low-rank tensor.

Since minimizing the nuclear norm is an NP hard problem, HoRPCA presented in [73] replaces it by its convex surrogate which is the sum of the nuclear norms of the mode-i unfoldings: $\sum_{i=1}^{3} \| \mathbf{L}_t^{(i)} \|_*$. One way to solve this optimization problem and obtain the low-rank and sparse parts of $\mathcal{M}_t$ is to use HoRPCA presented in [31]. However, there are two main drawbacks of applying HoRPCA for streaming or time-varying tensor data. First, it is very time consuming to compute HoRPCA in batch mode since all of the high dimensional data needs to be stored and processed. Second, HoRPCA yields different low-rank subspace information at each time point and subspace tracking as desired in this chapter requires additional metrics to compare the subspaces across time. To improve the computation efficiency and to better capture the evolving dynamics of the data, we propose to adapt and extend the projection based subspace update approach outlined in REPROCS. Thus, the optimization problem in equation (3.12) is solved in two steps. First, instead of determining the low-rank subspace at each time point, we propose to update the subspace across time by minimizing the Tucker norm, or the nuclear norm of each unfolding. As [208] has shown,

the minimizer of the nuclear norm can be obtained through singular value thresholding (SVT). As Algorithm 1 will show, we will use this approach to determine the low-rank subspace. In step 2, we project the observed tensor, $\mathcal{M}_t$, to a subspace orthogonal to the estimated low-rank subspace and transform equation (3.11) to a sparse recovery in noise problem as will be illustrated in equations (3.14) and (3.15) to obtain $\mathcal{S}_t$.

3.3.2 Algorithm Description

Suppose that we have a sequence of training tensors defined as $\mathcal{M}_{train}$ which do not contain any sparse information and are used for the initial estimate of the subspace in which each mode of $\mathcal{L}_t$ lies in. In the case of dynamic FCNs constructed from task-based EEG, this may correspond to the pre-stimulus activity. $\mathcal{M}_{train} \in \mathbb{R}^{N_1 \times N_2 \times N_3 \times t_{train}}$ can be considered as a 4-way tensor where the time information constitutes the 4^{th} mode and its full Tucker decomposition is

$$\mathcal{M}_{train} = \mathcal{C} \times_1 \mathbf{P}_0^{(1)} \times_2 \mathbf{P}_0^{(2)} \times_3 \mathbf{P}_0^{(3)} \times_4 \mathbf{P}_0^{(4)} \tag{3.13}$$

where $\mathbf{P}_0^{(1)}$, $\mathbf{P}_0^{(2)}$, $\mathbf{P}_0^{(3)}$ and $\mathbf{P}_0^{(4)}$ are the basis matrices along each mode with $\mathbf{P}_0^{(i)} \in \mathbb{R}^{N_i \times N_i}$. Let $\hat{\mathbf{P}}_0^{(i)}$s be the truncated version of $\mathbf{P}_0^{(i)}$ obtained by keeping the columns with the singular values greater than σ_{min}. $\hat{\mathbf{P}}_0^{(i)} \in \mathbb{R}^{N_i \times r_0^{(i)}}$s where $i \in \{1,\ 2,\ 3\}$ give the initial subspace information for $\mathcal{L}_t$ and $r_0^{(i)}$ is the rank of $\hat{\mathbf{P}}_0^{(i)}$. The goal is to estimate $\mathcal{L}_t$ and $\mathcal{S}_t$ for each $t > t_{train}$ by recursively updating its corresponding basis $\mathbf{P}_t^{(i)}$s. The $\mathcal{L}_t$'s are assumed to satisfy a slowly changing low-rank subspace model which will be detailed in section 3.3.3.

Let $\mathbf{P}_t$ be the set of projection matrices which form the basis for the subspaces in which each mode of $\mathcal{L}_t$ lies in $\mathbf{P}_t = \{\mathbf{P}_t^{(1)}, \mathbf{P}_t^{(2)}, \mathbf{P}_t^{(3)}\}$. Assume $\mathbf{P}_t$ has been accurately predicted using past estimates of $\mathcal{L}_t$ such that the projection of the new basis at time t to the orthogonal complement

of the past estimates $\left|\left|\left(\mathbf{I}-\hat{\mathbf{P}}_{t-1}^{(i)}(\hat{\mathbf{P}}_{t-1}^{(i)})^{\top}\right)\mathbf{P}_{t}^{(i)}\right|\right|_2$ is small. Then $\mathcal{M}_t$ is projected to the space orthogonal to $\hat{\mathbf{P}}_{t-1}^{(i)}$s defined through the projection operators $\boldsymbol{\phi}_t^{(i)} = \mathbf{I}-\hat{\mathbf{P}}_{t-1}^{(i)}(\hat{\mathbf{P}}_{t-1}^{(i)})^{\top}$ to obtain $\mathcal{Y}_t$ as $\mathcal{Y}_t = \mathcal{M}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)}$, which can be rewritten as:

$$\begin{aligned}\mathcal{Y}_t &= (\mathcal{L}_t + \mathcal{S}_t) \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)},\\ \mathcal{Y}_t &= \beta_t + \mathcal{S}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)},\end{aligned} \tag{3.14}$$

where $\beta_t = \mathcal{L}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)}$. Since $\parallel \boldsymbol{\phi}_t^{(i)}\mathbf{P}_t^{(i)} \parallel_2$ is small, the projection of $\mathcal{L}_t$ to $\boldsymbol{\phi}_t^{(i)}$s will yield small $\parallel \beta_t \parallel_F$ (see Appendix). Notice that, although the projection matrices $\boldsymbol{\phi}_t^{(i)}$'s are of size $N_i \times N_i$, they have rank $N_i - rank(\hat{\mathbf{P}}_t^{(i)})$. Therefore, obtaining $\mathcal{S}_t$ from $\mathcal{Y}_t$ can be represented as sparse recovery problem in small noise. Since $\hat{\mathbf{P}}_t^{(i)}$'s are dense and restricted isometry constants (RIC) of measurement matrices ($\boldsymbol{\phi}_t^{(i)}$) are small [214], we can accurately recover $\mathcal{S}_t$ from $\mathcal{Y}_t$ by solving following problem:

$$\begin{aligned}\hat{\mathcal{S}}_t &= arg\min \parallel \mathcal{S}_t \parallel_1,\\ s.t.\ \parallel \mathcal{Y}_t - \mathcal{S}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)} \parallel_F &\leqslant \epsilon.\end{aligned} \tag{3.15}$$

To recover $\mathcal{S}_t$ from $\mathcal{Y}_t$, we apply serial recovery procedure for compressed tensors, known as generalized tensor compressive sensing - serial (GTCS-S) presented in [218]. This algorithm repeatedly unfolds the compressed tensor along one of the modes and applies l_1 optimization to recover its columns. Once $\hat{\mathcal{S}}_t$ is recovered, $\mathcal{L}_t$ can be estimated as $\hat{\mathcal{L}}_t = \mathcal{M}_t - \hat{\mathcal{S}}_t$.

Algorithm 3.1 Higher-order Recursive Low-Rank + Sparse Structure Learning

Input: $\mathcal{M}_t$, $\hat{\mathbf{P}}_0^{(i)}s$
Output: $\hat{\mathcal{L}}_t$, $\hat{\mathcal{S}}_t$, t_j
for $t > 0$ **do**
  for i=1:3 **do**
    $\boldsymbol{\phi}_t^{(i)} = \mathbf{I} - \hat{\mathbf{P}}_{t-1}^{(i)}(\hat{\mathbf{P}}_{t-1}^{(i)})^\top$
  end for
  $\mathcal{Y}_t = \mathcal{M}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)}$
  Recover $\hat{\mathcal{S}}_t$ from $\mathcal{Y}_t$ by using GTCS-S algorithm [218].
  Estimate $\hat{\mathcal{L}}_t \leftarrow \mathcal{M}_t - \hat{\mathcal{S}}_t$
  if $mod(t - t_j + 1, \alpha) = 0$ **then**
    for i=1:3 **do**
      $\mathbf{D}^{(i)} = \left[\hat{\mathbf{L}}_{t_j+(k-1)\alpha}^{(i)} \cdots \hat{\mathbf{L}}_{t_j+k\alpha-1}^{(i)}\right]$
      $\hat{\mathbf{P}}_{(t)}^{(i)} = deleteDirection(\mathbf{D}, \hat{\mathbf{P}}_{(t-1)}^{(i)})$
      $\hat{\mathbf{P}}_{(t)}^{(i)} = addDirection(\mathbf{D}, \hat{\mathbf{P}}_{(t)}^{(i)})$
    end for
    if $\hat{\mathbf{P}}_{(t)}^{(1)} \neq \hat{\mathbf{P}}_{(t-1)}^{(1)}$ or $\hat{\mathbf{P}}_{(t)}^{(2)} \neq \hat{\mathbf{P}}_{(t-1)}^{(2)}$ or $\hat{\mathbf{P}}_{(t)}^{(3)} \neq \hat{\mathbf{P}}_{(t-1)}^{(3)}$ **then**
      $j \leftarrow j + 1, t_j \leftarrow t$
      $\hat{\mathbf{P}}_{(j)}^{(1)} \leftarrow \hat{\mathbf{P}}_{(t)}^{(1)}, \hat{\mathbf{P}}_{(j)}^{(2)} \leftarrow \hat{\mathbf{P}}_{(t)}^{(2)}, \hat{\mathbf{P}}_{(j)}^{(3)} \leftarrow \hat{\mathbf{P}}_{(t)}^{(3)}$
    end if
  else
    $\hat{\mathbf{P}}_{(t)}^{(1)} \leftarrow \hat{\mathbf{P}}_{(t-1)}^{(1)}, \hat{\mathbf{P}}_{(t)}^{(2)} \leftarrow \hat{\mathbf{P}}_{(t-1)}^{(2)}, \hat{\mathbf{P}}_{(t)}^{(3)} \leftarrow \hat{\mathbf{P}}_{(t-1)}^{(3)}$
  end if
end for

3.3.3 Slowly Changing Subspace & Change Points

The following assumptions are made to define slowly changing subspace along each mode of the tensor:

1. Let t_j denote the change points of the low-dimensional subspaces that $\mathbf{L}_t^{(i)}$s are in. Note that the subspaces along each mode can vary independently from the others and as such t_js are the collection of all change points across modes. Assume that for τ large enough, any τ length subsequence of $\mathbf{L}_t^{(i)}$s lies in low-dimensional subspaces, i.e. $max_t rank(\left[\mathbf{L}_{t-\tau+1}^{(i)}...\mathbf{L}_t^{(i)}\right] \ll \min(\tau, N_i, \prod_{k=1,k\neq i}^{3} N_k)$.

2. $\mathcal{L}_t$ lies in a low dimensional subspace that changes slowly along each mode i.e. $\mathcal{L}_t = \mathcal{A}_t \times_1 \mathbf{P}_t^{(1)} \times_2 \mathbf{P}_t^{(2)} \times_3 \mathbf{P}_t^{(3)}$ with $\mathbf{P}_t^{(i)} = \mathbf{P}_j^{(i)}$ for all $t_j \leqslant t \leqslant t_{j+1}, j = 1, 2, ...J$ where J is the maximum number of change points. $\mathbf{P}_j^{(i)}$ is an $N_i \times r_j^{(i)}$ basis matrix where $r_j^{(i)} \ll min(N_i, \prod_{k=1,k\neq i}^{3} N_k)$.

3. At the change points, t_j, at least one of the $\mathbf{P}_j^{(i)}$'s changes as $\mathbf{P}_j^{(i)} = \left[\mathbf{P}_{j-1}^{(i)}\mathbf{P}_{j,add}^{(i)}\right]$, $\mathbf{P}_j^{(i)} = \left[\mathbf{P}_{j-1}^{(i)} \backslash \mathbf{P}_{j,del}^{(i)}\right]$ or $\mathbf{P}_j^{(i)} = \left[(\mathbf{P}_{j-1}^{(i)} \backslash \mathbf{P}_{j,del}^{(i)}), \mathbf{P}_{j,add}^{(i)}\right]$ where $\mathbf{P}_{j,add}^{(i)}$ is a $N_i \times c_{j,add}^{(i)}$ basis matrix with $(\mathbf{P}_{j,add}^{(i)})^\top \mathbf{P}_{j-1}^{(i)} = 0$, i.e., the new directions added to the projection matrix are orthogonal to the previous directions and $\mathbf{P}_{j,del}^{(i)}$ is a $N_i \times c_{j,del}^{(i)}$ matrix of deleted basis columns.

4. There exists constants $c_{max}^{(i)}$ such that $0 \leqslant c_{j,add}^{(i)} \leqslant c_{max}^{(i)} < r_0^{(i)}$. $0 \leqslant \sum_{i=1}^{j}(c_{i,add} - c_{i,del}) \leqslant c_{dif}^{(i)}$ is required to imply $r_t^{(i)} \leqslant r_0^{(i)} + c_{dif}^{(i)} := r_{max}^{(i)}$. The number of change points $J \ll \min_i \left((N_i - r_0^{(i)} - c_{dif}^{(i)})/c_{max}^{(i)}\right)$, so $r_{max}^{(i)} + Jc_{max}^{(i)} \ll N_i$. Moreover, $(\prod_{k\neq i,k=1}^{2} N_k)(t_{j+1} - t_j) \gg r_0^{(i)} + c_{dif}^{(i)}$ helps to ensure $max_t rank(\left[\mathbf{L}_{t-\tau+1}^{(i)}...\mathbf{L}_t^{(i)}\right] \ll \min(\tau, N_i, \prod_{k=1,k\neq i}^{3} N_k)$.

5. The projection of $\mathcal{L}_t$ along the new added directions, $\mathcal{A}_{t,add} = \mathcal{L}_t \times_1 \mathbf{P}_{j,add}^{\top,(1)} \times_2 \mathbf{P}_{j,add}^{\top,(2)} \times_3 \mathbf{P}_{j,add}^{\top,(3)}$ is initially small, i.e $\max_{t_j \leqslant t \leqslant t_{j+\alpha}} \parallel \mathcal{A}_{t,add} \parallel_\infty \leqslant \gamma_{add}$ and $\gamma_{add} \ll min(\parallel \mathcal{L}_t \parallel_F, \parallel \mathcal{S}_t \parallel_F)$, but can increase gradually.

In order to enable a more efficient online implementation, the low-rank subspaces $\mathbf{P}_t^{(i)}$s are

estimated and updated every α samples, where α is selected empirically. Similar to the projection PCA (p-PCA) procedure used in [67], mode-i unfoldings $\hat{\mathbf{L}}_t^{(i)}$'s of the last α $\hat{\mathbf{L}}_t$'s are concatenated as $\mathbf{D}^{(i)} = \left[\hat{\mathbf{L}}^{(i)}_{t_j+(k-1)\alpha} \cdots \hat{\mathbf{L}}^{(i)}_{t_j+k\alpha-1}\right]$ with $k \in \{1,\ 2,\ ...,\ K\}$ where K is the maximum number of length α windows and $\mathbf{D}^{(i)}$s are projected onto subspaces which are orthogonal to $\hat{\mathbf{P}}^{(i)}_{(j-1)}$s as follows: $\mathbf{D}^{(i)}_{proj} = (I - \hat{\mathbf{P}}^{(i)}_{(j-1)}(\hat{\mathbf{P}}^{(i)}_{(j-1)})^\top)\mathbf{D}^{(i)}$. Then PCA is applied to find the subspace which spans $\mathbf{D}^{(i)}_{proj}$. Let $\mathbf{P}^{(i)}_{j,add}$ be the truncated basis that spans this subspace obtained by keeping the eigenvectors with eigenvalues greater than σ_{min}. $\mathbf{P}^{(i)}_{j,add}$ and previous subspace estimate $\hat{\mathbf{P}}^{(i)}_{(j-1)}$ together yield the new subspace estimate as: $\hat{\mathbf{P}}^{(i)}_{(j)} = [\hat{\mathbf{P}}^{(i)}_{(j-1)}\mathbf{P}^{(i)}_{j,add}]$. During the update step, some of the existing directions can also be deleted from the projection matrix by finding the ones with eigenvalues lower than σ_{min} (see Algorithms 3.2 and 3.3). If there are any added or deleted directions, it means that there is a change point. It is also important to note that, the tensor subspace estimation implemented in this study finds subspaces along each mode individually without taking other modes into account. Thus, the proposed method is not optimized like higher-order orthogonal iteration (HOOI) [36] but offers a computationally efficient way of estimating subspaces along each mode.

Algorithm 3.2 Delete Direction

1: Input: **D**: data, **P**: input basis matrix
2: Output: **Q**: output basis matrix
3: $\lambda = \frac{1}{w} diag((\mathbf{P}'\mathbf{D}(\mathbf{P}'\mathbf{D})^\top))$ where w is the number of columns of $\mathbf{D}$.
4: i= find($\lambda < \sigma_{min}$)
5: $\mathbf{Q} = [\mathbf{P} \backslash \mathbf{P}(i,:)]$

Algorithm 3.3 Add Direction

1: Input: $\mathbf{D}$: data, $\mathbf{P}$: input basis matrix
2: Output: $\mathbf{Q}$: output basis matrix
3: Projection: compute $\mathbf{D}_{proj} \leftarrow (\mathbf{I} - \mathbf{P}\mathbf{P}')\mathbf{D}$
4: PCA: compute $\frac{1}{w}\mathbf{D}_{proj}\mathbf{D}'_{proj} = \mathbf{U}\lambda\mathbf{U}'$ where w is the number of columns in $\mathbf{D}$.
5: i= find($diag(\lambda) > \sigma_{min}$)
6: $\mathbf{Q} = [\mathbf{P}\ \mathbf{U}(i,:)]$

3.3.4 Computational Complexity

In this section, we offer a comparison of the computational complexity of the proposed approach with respect to REPROCS applied to our data in vectorized form. Let the 3-way tensor be of size $N \times N \times N$. For the time points which do not require subspace update, computational complexity of the proposed approach is equivalent to the complexity of l_1 regularization $O(N^3)$ multiplied by the total number of fibers to recover for each mode to obtain the sparse component and is equal to $3N^2O(N^3)$. However, if we use REPROCS after vectorizing the data, complexity for the same operations become $O((N^3)^3) = O(N^9)$. For the time points which require basis update, there is an additional cost of covariance matrix computation and eigenvalue decomposition. For our approach, covariance matrix computations for the three modes have a complexity of $3O((\alpha N) \times N^4) = 3O(\alpha N^5)$ operations whereas eigenvalue decompositions cost $3O(N^3)$. However, REPROCS requires $O(\alpha(N^3)^2) = O(\alpha N^6)$ operations for covariance matrix computation and $O((N^3)^3) = O(N^9)$ operations for eigenvalue decomposition.

3.4 Results

3.4.1 Simulated Networks

The proposed framework is first applied to three simulated dynamic tensors $\mathcal{X}_t \in \mathbb{R}^{64\times64\times60}$ for $t \in \{1,\ 2, ...,\ 80\}$. For each network type, 20 simulations of the tensors are generated where each frontal slice $\mathbf{X}_t(:,:,i)$ corresponds to a weighted and undirected network. In our experiments, to generate the networks, we used three well-known network models known as the modular small-world network, hierarchical modular small-world network and overlapped modular networks 3.1. Distinct regions in the brain which are strongly connected within themselves are specialized for different processes in the brain and, this phenomenon is known as functional segregation. Presence of these specialized neuronal groups appear as different modules in brain networks. Moreover, some of the nodes in a module may have more specialized function which yields hiearchical structure in a network while some of the nodes belong to multiple clusters resulting in overlapping modules [219]. All of these network structures are illustrated in Fig. 3.1.

For the experiments including modular small world network model, initially, the networks contain 2 equal size modules. After $t = 20$, both of the modules are slowly divided into two smaller modules of size 16 nodes each. After $t = 60$, the network structure evolves back to the initial structure. For the second set of experiments with the hierarchical modular small world network model, the networks contain 2 equal size modules at the beginning. After $t = 20$, both of the modules are slowly divided into two smaller modules of size 16 nodes each while establishing hierarchical structure. After $t = 60$, the network structure evolves back to the initial structure. For the third set of experiments with overlapping modules, the networks contain 2 equal size non-overlapping modules at the beginning. After $t = 20$, 25% of the nodes start to belong to both modules. After $t = 60$, network structure evolves back to the initial non-overlapping structure.

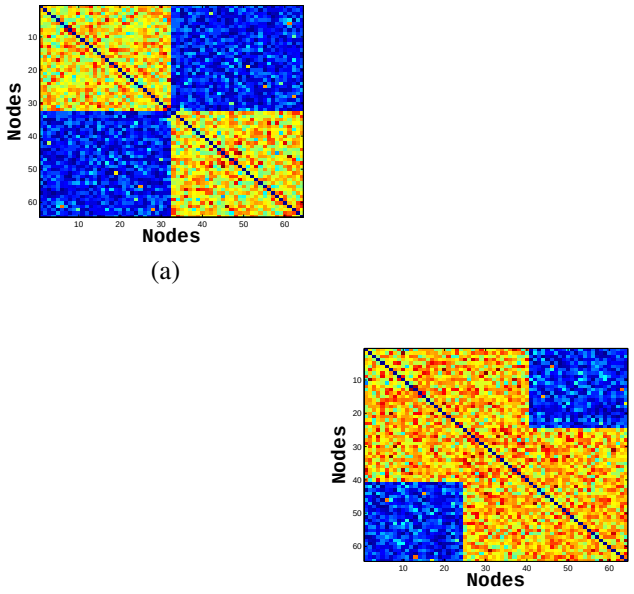


Figure 3.1: Illustration of network structures: (a) Modular network (b) Hierarchical modular network, (c) Overlapping modules.

For all of the experiments, intra-cluster edge values were selected from $N(0.6,\ 0.1)$ and truncated to the interval $[0, 1]$ while the inter-cluster edge values were selected from $N(0.1,\ 0.1)$. Moreover, these networks were corrupted by a sparse noise matrix $\mathbf{E}_t$ whose sparsity varies from 10% to 40% and $e_{i,j} \sim beta(4, 2)$. Proposed algorithm is applied with $\alpha = 5$ and σ_{min} is determined as 10% of highest singular value obtained from the initial subspace estimate along that mode using the first 5 time points. The proposed algorithm is compared to an implementation without the sparse recovery step similar to performing standard HoSVD at each time point. Mean squared error which quantifies the error between estimated and original low-rank components for all of the network models are computed for both algorithms as

$$\text{MSE} = \frac{1}{T_{end} - T_{start} + 1} \sum_{t=T_{start}}^{T_{end}} \frac{\| \mathcal{L}_t - \hat{\mathcal{L}}_t \|_F^2}{\prod_{i=1}^{3} N_i}, \tag{3.16}$$

where T_{start} and T_{end} are the start and end points of the detected time interval and N_i is the size

of the tensor along ith mode.

Tables 3.1, 3.2 and 3.3 show that Ho-RLSL is more robust than HoSVD for sparse outliers with smaller MSE values. As the sparsity level of the noise increases, the difference in performance between the two algorithms also increases. Complexity of the network structure, i.e. modular vs. hierarchically modular, also affects the accuracy of the algorithms and increased structural complexity results in increased error as seen in Tables 3.1, 3.2 and 3.3.

Table 3.1: Average MSE over time computed for low-rank components during detected time intervals obtained by Ho-RLSL and HoSVD for modular network structure under varying noise sparsity levels.

Noise Level	Method	Interval-1	Interval-2	Interval-3
10%	Ho-RLSL	0.0046	0.0094	0.0045
	HoSVD	0.0097	0.0144	0.0095
20%	Ho-RLSL	0.0085	0.0167	0.0084
	HoSVD	0.0233	0.0302	0.0229
30%	Ho-RLSL	0.0136	0.0260	0.0137
	HoSVD	0.0409	0.0496	0.0405
40%	Ho-RLSL	0.0215	0.0387	0.0216
	HoSVD	0.0613	0.0713	0.0607

Table 3.2: Average MSE over time computed for low-rank components during detected time intervals obtained by Ho-RLSL and HoSVD for hierarchical modular network structure under varying noise sparsity levels.

Noise Level	Method	Interval-1	Interval-2	Interval-3
10%	Ho-RLSL	0.0149	0.0265	0.0137
	HoSVD	0.0202	0.0319	0.0189
20%	Ho-RLSL	0.0184	0.0333	0.0171
	HoSVD	0.0335	0.0465	0.0322
30%	Ho-RLSL	0.0240	0.0377	0.0225
	HoSVD	0.0511	0.0663	0.0497
40%	Ho-RLSL	0.0346	0.0521	0.0326
	HoSVD	0.0712	0.0875	0.0699

Table 3.3: Average MSE over time computed for low-rank components during detected time intervals obtained by Ho-RLSL and HoSVD for network structure with overlapping modules under varying noise sparsity levels.

Noise Level	Method	Interval-1	Interval-2	Interval-3
10%	Ho-RLSL	0.0043	0.0088	0.0045
	HoSVD	0.0091	0.0107	0.0094
20%	Ho-RLSL	0.0080	0.0195	0.0083
	HoSVD	0.0225	0.0250	0.0228
30%	Ho-RLSL	0.0131	0.0336	0.0135
	HoSVD	0.0399	0.0435	0.0404
40%	Ho-RLSL	0.0209	0.0509	0.0215
	HoSVD	0.0602	0.0640	0.0606

3.4.2 Effect of Network Size on Computation Time and Performance

The proposed framework is applied to two simulated dynamic tensors $\mathcal{X}_t \in \mathbb{R}^{64\times 64\times 60}$ and $\mathcal{X}_t \in \mathbb{R}^{128\times 128\times 60}$ for $t \in \{1,\ 2, ...,\ 80\}$ to see the effect of network size on the computation time and performance of the algorithm. 10 simulations of the tensors are generated where each frontal slice $\mathbf{X}_t(:, :, i)$ corresponds to a weighted and undirected network and the third mode corresponds to the number of subjects. In these experiments, to generate the networks, we used modular small-world networks. For the experiments, initially, the networks contain 2 equal size modules. After $t = 20$, both of the modules are slowly divided into two smaller modules of equal size. After $t = 60$, the network structure evolves back to the initial structure.

For the experiments, intra-cluster edge values were selected from $N(0.6,\ 0.1)$ and truncated to the interval $[0, 1]$ while the inter-cluster edge values were selected from $N(0.1,\ 0.1)$. Moreover, these networks were corrupted by a sparse noise matrix $\mathbf{E}_t$ whose sparsity is 10% and $e_{i,j} \sim beta(4, 2)$. Proposed algorithm is applied with $\alpha = 5$ and σ_{min} is determined as 10% of highest singular value obtained from the initial subspace estimate along that mode using the first 5 time points. Mean squared error which quantifies the error between estimated and original low-rank components for both network sizes are computed for both algorithms as described in Section 3.4.1.

All of the simulations were run on a computer with Intel(R) Core(TM) i5-2500T CPU and 4.00 GB memory. Table 3.4 shows that doubling the number of nodes in the network increases the computation time almost three times. Moreover, increased network size yields better low-rank estimation for the complex network structures. As seen in Table 3.4, MSE computed for the second time interval where the networks contain more modules significantly decreases with the increased network size. This is due to the fact that with more modules in the network the number of nodes in a module decreases making the subspace estimation more challenging. When the number of nodes in the network increases, the subspace estimation becomes more accurate.

Table 3.4: Average computation time for Ho-RLSL for dynamic tensors containing 64×64 and 128×128 networks with average MSE over time computed for low-rank components during detected time intervals.

Tensor Size	Time (sec)	Interval-1	Interval-2	Interval-3
$64 \times 64 \times 60$	2.3916×10^4	0.0046	0.0090	0.0045
$128 \times 128 \times 60$	6.3397×10^4	0.0045	0.0065	0.0046

3.4.3 EEG Data

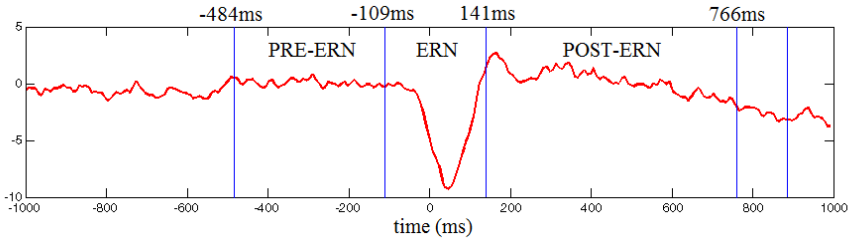
The proposed tensor tracking approach is applied to a set of connectivity graphs constructed from EEG data containing the error-related negativity (ERN) and correct-related negativity (CRN). The ERN is a brain potential response that occurs following performance errors in a speeded reaction time task usually 25-75 ms after the response [220]. Previous work [221] indicates that there is increased coordination between the lateral prefrontal cortex (lPFC) and medial prefrontal cortex (mPFC) within the theta frequency band (4-8 Hz) and ERN time window. EEG data from 63-channels was collected in accordance with the 10/20 system on a Neuroscan Synamps2 system (Neuroscan, Inc.) sampled at 128 Hz from 91 subjects. Full methodological details of the recording are available in the previous report [220]. The task was a common speeded-response letter

(H/S) flanker, where error and correct response-locked trials from each subject were utilized. A random subset of correct trials was selected, to equate the number of error relative to correct trials for each participant. The EEG data are pre-processed by the spherical spline current source density (CSD) waveforms to sharpen event-related potential (ERP) scalp topographies and reduce volume conduction [222]. The CSD has fewer assumptions than many inverse transforms, attenuates volume conduction, and represents independent sources near the cortical surface [177]. For each subject and response type, the pairwise phase locking value in the theta frequency band was computed as described in eqn. 3.8 [148]. We constructed 3-way tensors at each time point $\mathcal{X}_t \in \mathbb{R}^{63\times 63\times 91}$ for both ERN and CRN data separately where the first and second mode represent the adjacency matrix of the connectivity graphs while the third mode corresponds to the subjects for $t \in \{1,\ 2, ...,\ 256\}$.

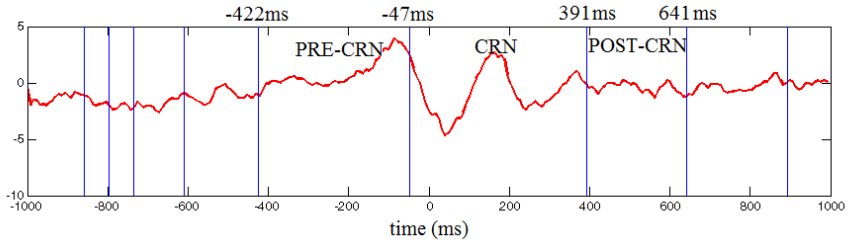
In this section, the method described in Section 3.3 is applied to tensors constructed from the connectivity networks as described in Section 3.2.3. The connectivity networks corresponding to the first 10 time points $t \in \{1,\ 2, ...,\ 10\}$ and all subjects was used in the training step to obtain initial subspace information of the low-rank component $\mathcal{L}_t$. This training data is used to obtain an initial subspace estimate and 10 time points approximately correspond to 78 ms of data. It is assumed that during this time period the connectivity networks are almost stationary and the low-rank structure does not change significantly. Then the proposed approach was applied to the remaining time points with $\alpha = 8$ and $\sigma_{min} = 0.11$. Since the connectivity networks constructed by the phase synchrony measure are symmetric, the basis matrices $\mathbf{P}_t^{(1)}$ and $\mathbf{P}_t^{(2)}$ corresponding to the first two modes are identical to each other as $\mathbf{P}_t^{(1)} = \mathbf{P}_t^{(2)}$ for each time point t. This is due to the fact that the unfolding of the 3-mode tensor along modes 1 and 2 yield identical matrices with identical subspaces and does not change the general algorithm introduced in Section 3.3.

In Fig. 3.2, change points corresponding to the connectivity mode $\mathbf{L}_t^{(1)}$ are given. It can be

seen that for ERN networks there is an interval (-109, 141) ms corresponding to the response time and the ERN response. There is also a longer interval (141, 766) ms corresponding to the Pe, the error-related positivity which usually occurs 200-500ms after making an incorrect response, following the error negativity (ERN). Similar time intervals are obtained for CRN, with the biggest difference being a longer time interval around the response time as the physiological response for CRN is not as pronounced as the one for ERN.



(a)



(b)

Figure 3.2: Average of (a) ERN and (b) CRN waveforms and the detected change points corresponding to the connectivity mode.

In order to better interpret the network structure corresponding to different time intervals, FCCA reviewed in Section 3.2.4 was applied to the sets of 63×63 networks obtained from low-rank tensors $\hat{\mathcal{L}}_t \in \mathbb{R}^{63\times63\times91}$ within each time interval (pre-ERN, ERN, post-ERN, pre-CRN, CRN, post-CRN) where $(\#\text{ of input adjacency matrices}) = (\#\text{ of subjects})\times(\#\text{ time points in the interval})$. As seen in Fig. 3.3, the network structure in pre-ERN interval is similar to the network structure of the post-ERN interval, while the identified modules are more segregated in the ERN interval relative to the pre and post-ERN. This is in line with previous results indicating that separable clusters

are apparent relative to left and right motor areas, and left and right lateral-PFC regions during ERN [205]. For the CRN clusters, we observed that segregation of the lateral and central areas is quite limited with one large fronto-central cluster present during all CRN intervals. The exception are small frontal- and central clusters during the CRN (a similar two were observed during the ERN). This is consistent with the idea that medial frontal regions are activated during correct trials, but to a smaller extent than during errors.

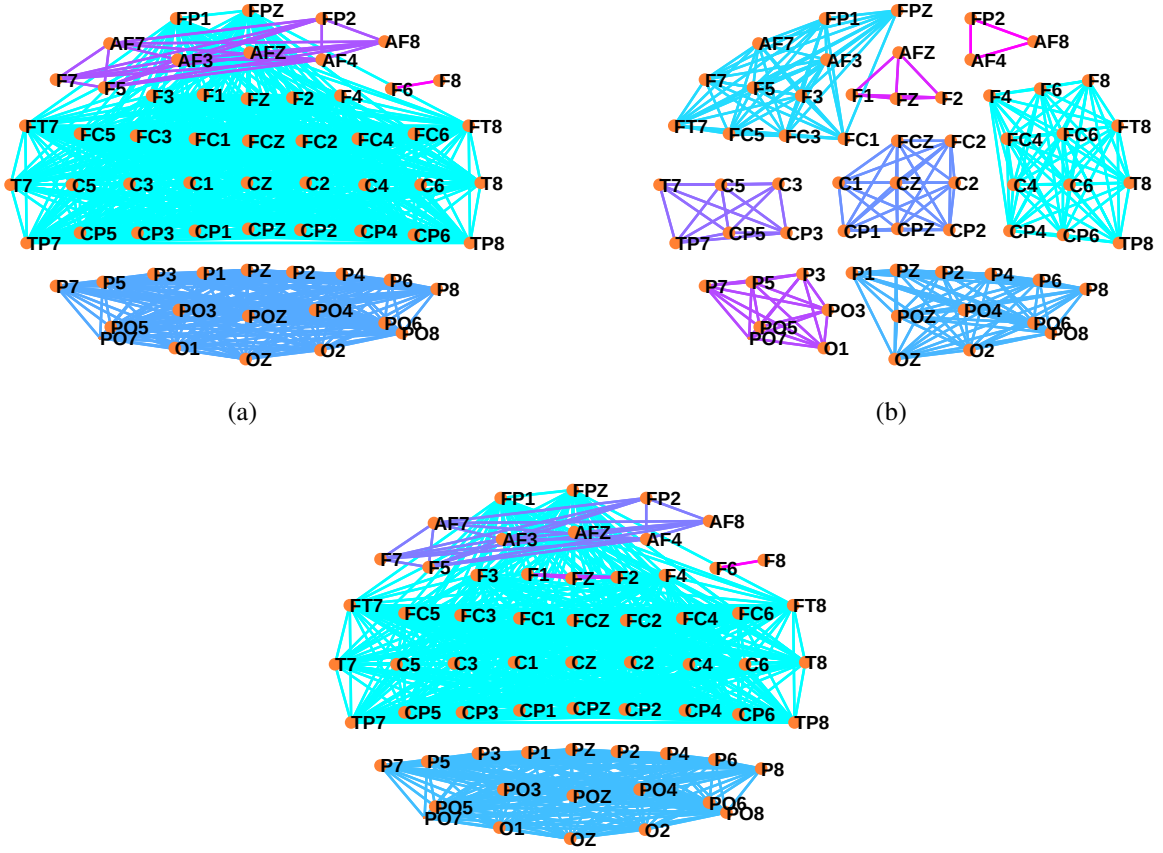


Figure 3.3: Network structures for the low-rank components of ERN networks obtained by Ho-RLSL: (a) pre-ERN, (b) ERN, (c) post-ERN.

To show the denoising performance of the proposed method compared to HoSVD, we applied HoSVD to the EEG networks with the same α and σ_{min} values used in Ho-RLSL. First, HoSVD detected pre-ERN, ERN and post-ERN intervals very similar to the ones obtained from Ho-RLSL

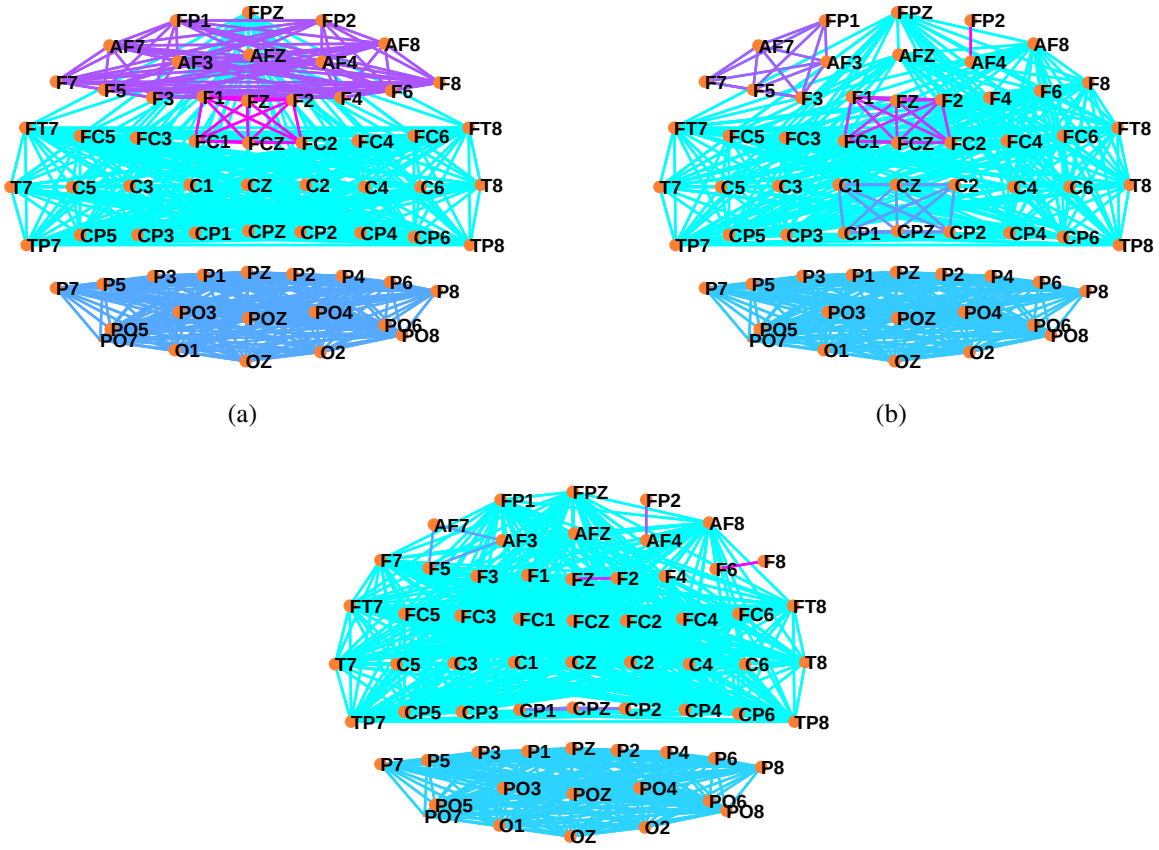


Figure 3.4: Network structures for the low-rank components of CRN networks determined obtained by Ho-RLSL: (a) pre-CRN, (b) CRN, (c) post-CRN.

(see Table 3.5). The change point at the end of the pre-ERN interval was shifted by α time points. We then used FCCA to identify the common network structure for each interval (Fig. 3.5). As seen in Figs. 3.3 and 3.5, both HoSVD and Ho-RLSL yield the same network structure for the ERN interval, while HoSVD yields more noisy networks for pre- and post-ERN intervals. When the physiological response is strong such as during the ERN, the network is less noisy yielding the exact cluster structure both for Ho-RLSL and HoSVD. However, when the networks are noisier such as for the pre-ERN interval, our algorithm provides a cleaner low-rank approximation which yields more distinct cluster structures. For CRN networks, Ho-RLSL and HoSVD detected very different pre-CRN, CRN and post-CRN intervals (see Table 3.5), and we cannot compare the network

structures obtained from both algorithms. In particular, HoSVD detects a very long pre-CRN time interval and very short CRN and post-CRN intervals. The CRN and post-CRN intervals detected by Ho-RLSL align better with well-known ERPs such as P300.

Table 3.5: Detected ERN and CRN intervals by Ho-RLSL and HoSVD

Interval	Ho-RLSL	HoSVD
Pre-ERN	-0.484ms to -0.109ms	-0.484ms to -0.047ms
ERN	-0.109ms to 0.141ms	-0.047ms to 0.141 ms
Post-ERN	0.141ms to 0.766ms	0.141ms to 0.766ms
Pre-CRN	-0.422ms to -0.047ms	-0.734ms to -0.109ms
CRN	-0.047ms to 0.391ms	-0.109ms to 0.141 ms
Post-CRN	0.391ms to 0.641ms	0.141ms to 0.391ms

3.5 Conclusions

In this chapter, we introduced a new recursive low-rank + sparse structure learning algorithm for tensor type data to track dynamic modular structure of functional connectivity networks constructed from EEG recordings across multiple subjects. To this aim, a recent subspace tracking approach, REPROCS, was adapted and extended to tensor type data. This extension offers several novelties with respect to the original algorithm. In original REPROCS, the measurements at each time point are vectors and the algorithm uses the fact that each measurement vector is coming from a low-rank subspace. However, in our case, low-rank corresponds to a low Tucker rank, i.e. the matricized version of the tensor along each mode is a low-rank matrix implying it can be reconstructed through the outer product of a small number of eigenvectors. This is particularly suitable for dealing with data that has a modular structure such as FCNs. Using REPROCS to analyze this type of dataset would require vectorization which breaks the intrinsic low-rank structure of each measurement along with increased computational complexity. The proposed approach yields robust estimation of the low-rank component of a dynamic 3-way tensor at each time

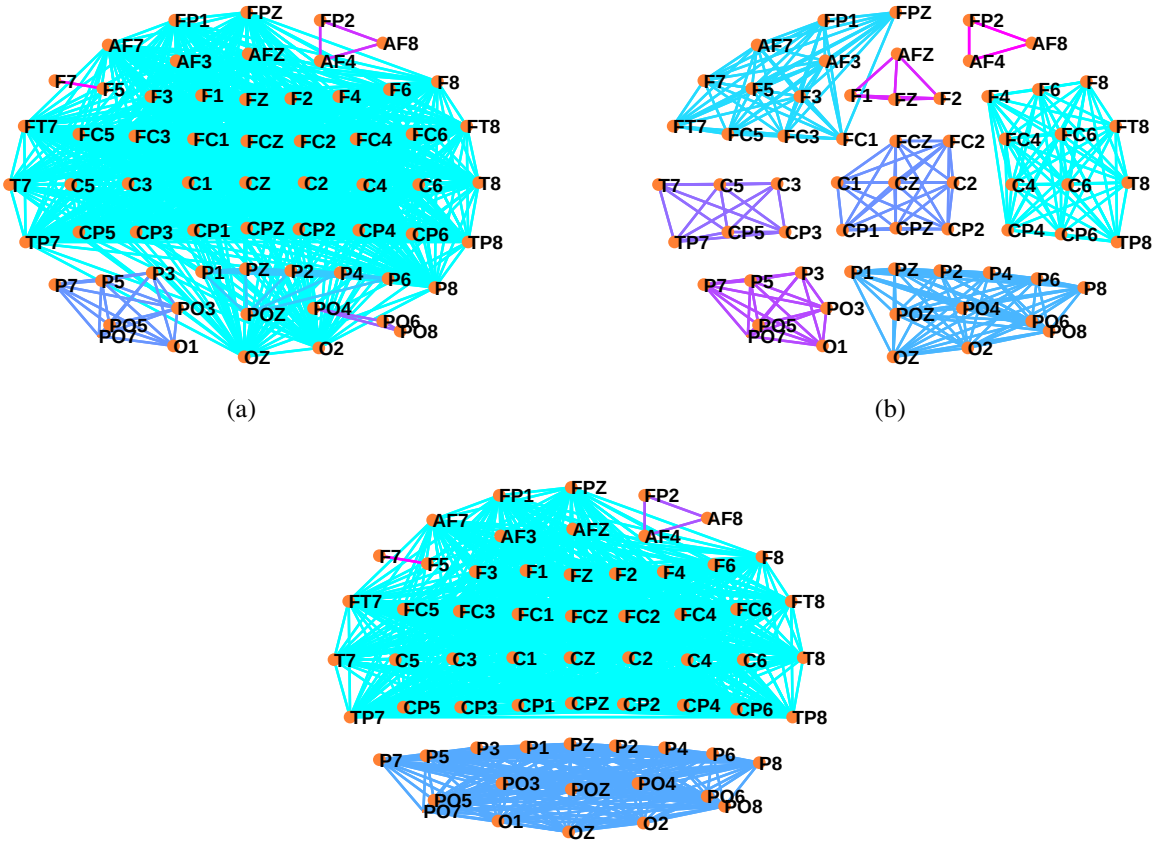


Figure 3.5: Network structures for the low-rank components of ERN networks obtained by HoSVD: (a) pre-ERN, (b) ERN, (c) post-ERN.

point and provides a recursive way to update the subspace information. This approach identifies the time points where the low-rank subspaces change and recursively updates these subspaces to improve the low-rank approximation to data. The proposed approach is first applied to a set of simulated networks for performance evaluation, and is then used to separate low-rank and sparse parts of dFCNs across multiple subjects. Low-rank component of dFCNs obtained for the detected time intervals are summarized by using a recently introduced multiple graph clustering approach, FCCA. The low-rank subspace approximation to each time interval provides a denoised version of the original network, thus improving the quality of the clustering results. The results indicate a clear change in the community structure before and after the physiological response. Moreover,

the detected community structures are in line with previous hypothesis regarding the networks involved in cognitive control [221].

One main concern about the proposed algorithm is the selection of parameters α and σ_{min}. Both α and σ_{min} are data-dependent parameters and their selection requires some *a priori* information. First, α should be smaller than the time interval between two consecutive change points in order to identify each one of them. For example, in our simulations change points appear at $t = 40$ and $t = 60$ and there are 20 time points between consecutive change points. We selected α as 5 in our simulations and we identified the change points correctly. When we apply our algorithm with $\alpha = 10$ we can still identify the change points. However, if we select $\alpha = 30$, we identify the first change point at $t = 60$ which corresponds to the beginning of the third interval. Therefore, the subspace estimate is based on the information from the second time interval and cannot do a good job of representing the third time interval and the algorithm fails. Moreover, if we keep α very small, then the algorithm will be more susceptible to instantaneous noise and we will have a lot of incorrect change points or false positives. For the real dataset, since we know something about the dynamics of EEG, we selected α to be at least as long as the duration of ERNs (50-75 ms) with $\alpha = 8$. Selection of σ_{min} is also dependent on the datasets. For example, in our simulations, we selected σ_{min} as 10% of the maximum eigenvalue. If we select it too small, the algorithm includes many basis vectors which correspond to noise and both the low-rank assumption and the algorithm fail. If we select σ_{min} too high, then the algorithm does not update the low-rank subspace for the slow changes and the algorithm again fails. Selecting σ_{min} for real dataset is more difficult, because gaps between eigenvalues may not be very clear. For example, for our EEG datasets, the first eigenvalue was much larger than the others and using a threshold around 10% of the maximum eigenvalue did not work. In this case, we chose $\sigma_{min} = 0.11$ empirically. If we select it lower, the algorithm includes many basis vectors at every time window which cause the algorithm to fail.

When we increased σ_{min}, we lost a lot of the change points.

Future work will consider extending the proposed approach to higher order tensors by including various experimental conditions, different frequency bands and multiple modalities. Future work will also consider extensions of linear low-rank subspace models to unions of subspaces or manifolds. Modifications to the Ho-RLSL algorithm such as predicting support of the sparse component [204] or using partial subspace knowledge for the low-rank component [223] will further improve the performance of Ho-RLSL.

Chapter 4

Multiscale Analysis for Higher-order Tensors

4.1 Introduction

Data in the form of multidimensional arrays, also referred to as tensors, arise in a variety of applications including chemometrics, hyperspectral imaging, high resolution videos, neuroimaging, biometrics and social network analysis [4, 5, 224]. These applications produce massive amounts of data collected in various forms with multiple aspects and high dimensionality. Tensors, which are multi-dimensional generalizations of matrices, provide a useful representation for such data. A crucial step in many applications involving higher-orders tensors is multiway reduction of the data to ensure that the reduced representation of the tensor retains certain characteristics. Early multiway data analysis approaches reformatted the tensor data as a matrix and resorted to methods developed for classical two-way analysis. However, one cannot discover hidden components within multiway data using conventional matrix decomposition methods as matrix based representations cannot capture multiway couplings focusing on standard pairwise interactions. To this end, many different types of tensor decomposition methods have been proposed in literature [37,40,225–227].

In contrast to the matrix case, where data reduction is often accomplished via low-rank repre-

sentations such as singular value decomposition (SVD), the notion of rank for higher order tensors is not uniquely defined. CANDECOMP/PARAFAC (CP) decomposition and Tucker decomposition are two of the most widely used tensor decomposition methods for data reduction [30,35]. For CP, the goal is to approximate the given tensor as a weighted sum of Rank-1 tensors, where rank-1 tensor refers to the outer product of n vectors, with n being equal to the order of the tensor. Tucker model allows for interactions between the factors from the different modes resulting in a typically dense but small core tensor. This model also introduces the notion of Tucker rank or n-rank, which refers to the n-tuple of ranks corresponding to the tensor unfoldings along each mode. Therefore, low rank approximation with the Tucker model can be obtained by projections onto low-rank factor matrices. Unlike CP decomposition, Tucker decomposition is in general non-unique. To obtain meaningful and unique representation by the Tucker decomposition, orthogonality, sparsity and non-negativity constraints are often imposed on the factors yielding Non-Negative Tensor Factorization (NTF) and Sparse Non-Negative Tucker Decomposition [228–230]. Tucker decomposition with orthogonality constraints on the factors, is known as Higher-Order Singular Value Decomposition (HoSVD) or Multilinear SVD [35]. HoSVD can simply be computed by flattening the tensor in each mode and calculating the n-mode singular vectors corresponding to that mode.

With the emergence of multidimensional big data, classical tensor representation and decomposition methods have become inadequate since the size of these tensors exceeds available working memory and the processing time is very long. In order to address the problem of large-scale tensor decomposition, several block-wise tensor decomposition methods have been proposed [37]. The basic idea is to partition a big data tensor into smaller blocks and perform tensor related operations block-wise using suitable tensor format. Preliminary approaches relied on a hierarchical tree structure and reduced the storage of d-dimensional arrays to the storage of auxiliary three-dimensional ones such as the tensor-train decomposition (T-Train), also known as the matrix product state

(MPS) decomposition, [40] and Hierarchical Tucker Decomposition (H-Tucker) [39]. In particular, in the area of large volumetric data visualization, tensor based multiresolution hierarchical methods such as TAMRESH have attracted attention [231]. However, all of these methods are interested in fitting a low-rank model to data which lies near a *linear* subspace, thus being limited to learning linear structure.

Similar to the research efforts in tensor reduction, low-dimensional subspace and manifold learning methods have also been extended for higher order data clustering and classification applications. In early work in the area, Vasilescu and Terzopoulos [232] extended the eigenface concept to the tensorface by using higher order SVD and taking different modes such as expression, illumination and pose into account. Similarly, 2D-PCA for matrices has been used for feature extraction from face images without converting the images into vectors [233]. He et al. [234] extended locality preserving projections [235] to second order tensors for face recognition. Dai and Yeung [236] presented generalized tensor embedding methods such as the extensions of local discriminant embedding methods [237], neighborhood preserving embedding methods [238], and locality preserving projection methods [235] to tensors. Li et al. [239] proposed a supervised manifold learning method for vector type data which preserves local structures in each class of samples, and then extended the algorithm to tensors to provide improved performance for face and gait recognition. Similar to vector-type manifold learning algorithms, the aim of these methods is to find an optimal *linear* transformation for the tensor-type training data samples without vectorizing them and mapping these samples to a low dimensional subspace while preserving the neighborhood information.

In this chapter, we propose a novel multi-scale analysis technique to efficiently approximate tensor type data using locally linear low-rank approximations. The proposed method constructs data-dependent multiscale dictionaries to better represent the data. The proposed algorithm con-

sists of two major steps: 1) Constructing a tree structure by partitioning the tensor into a collection of permuted subtensors, and 2) Constructing multiscale dictionaries by applying HoSVD to each subtensor. The major contributions of the proposed framework are three fold. First, we introduce a multi-scale tensor approximation method which allows the user to approximate a given tensor for given memory and processing power constraints allowing flexibility in the decomposition. The computational complexity and memory requirements of the proposed method are given in comparison to conventional HoSVD. Second, the proposed tensor partitioning method clusters the tensor data across each mode to obtain subtensors that are composed of similar entries. In this manner, we show through a theoretical error analysis that the resulting subtensors can approximate the locally linear structure more efficiently. Finally, we introduce different variations of the method for adaptively pruning the tree obtaining a better trade-off between compression rate and reconstruction error. The proposed method is evaluated for two common signal processing applications: data reduction and classification. The efficiency and accuracy of the proposed method for data reduction and classification is evaluated for different tensor type data and compared to state-of-the-art tensor decomposition methods including HoSVD, T-Train and H-Tucker decompositions.

Although this chapter focuses on the integration of a single existing tensor factorization technique (i.e., the HoSVD) into a clustering-enhanced multiscale approximation framework, we would like to emphasize that the ideas presented herein are significantly more general. In principal, for example, there is nothing impeding the development of multiscale variants of other tensor factorization approaches (e.g., CP, T-Train, H-Tucker, etc.) in essentially the same way. In this chapter, it is demonstrated that the use of the HoSVD as part of a multiscale approximation approach leads to improved compression and classification performance over standard HoSVD approaches. However, this chapter should additionally be considered as evidence that similar improvements are also likely possible for other tensor factorization-based compression and classification schemes, as well

as for other related applications.

4.2 Background

4.2.1 Tensor Notation and Algebra

A multidimensional array with N modes $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is called a tensor, where $x_{i_1,i_2,..i_N}$ denotes the $(i_1, i_2, ..i_N)^{th}$ element of the tensor $\mathcal{X}$. The vectors in $\mathbb{R}^{I_n}$ obtained by fixing all of the indices of such a tensor $\mathcal{X}$ except for the one that corresponds to its nth mode are called its *mode-n fibers*. Let $[N] := \{1, \ldots, N\}$ for all $N \in \mathbb{N}$. Basic tensor operations are reviewed below (see, e.g., [30], [240], [241]).

Tensor addition and multiplication by a scalar: Two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be added using component-wise tensor addition. The resulting tensor $\mathcal{X} + \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ has its entries given by $(\mathcal{X} + \mathcal{Y})_{i_1,i_2,..i_N} = x_{i_1,i_2,..i_N} + y_{i_1,i_2,..i_N}$. Similarly, given a scalar $\alpha \in \mathbb{R}$ and a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ the rescaled tensor $\alpha\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ has its entries given by $(\alpha\mathcal{X})_{i_1,i_2,..i_N} = \alpha\, x_{i_1,i_2,..i_N}$.

Mode-n products: The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots I_n \times \ldots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted as $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}$, $(\mathcal{Y})_{i_1,i_2,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,\ldots,i_n,\ldots,i_N} u_{j,i_n}$. It is of size $I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$. The following facts about mode-n products are useful (see, e.g., [30], [241]).

Lemma 1. *Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, $\alpha, \beta \in \mathbb{R}$, and $\mathbf{U}^{(n)}, \mathbf{V}^{(n)} \in \mathbb{C}^{J_n \times I_n}$ for all $n \in [N]$. The following are true:*

(a) $(\alpha\mathcal{X} + \beta\mathcal{Y}) \times_n \mathbf{U}^{(n)} = \alpha \left(\mathcal{X} \times_n \mathbf{U}^{(n)}\right) + \beta \left(\mathcal{Y} \times_n \mathbf{U}^{(n)}\right)$.

(b) $\mathcal{X} \times_n \left(\alpha\mathbf{U}^{(n)} + \beta\mathbf{V}^{(n)}\right) = \alpha \left(\mathcal{X} \times_n \mathbf{U}^{(n)}\right) + \beta \left(\mathcal{X} \times_n \mathbf{V}^{(n)}\right)$.

(c) If $n \neq m$ *then* $\mathcal{X} \times_n \mathbf{U}^{(n)} \times_m \mathbf{V}^{(m)} = \left(\mathcal{X} \times_n \mathbf{U}^{(n)}\right) \times_m \mathbf{V}^{(m)} = \left(\mathcal{X} \times_m \mathbf{V}^{(m)}\right) \times_n \mathbf{U}^{(n)} = \mathcal{X} \times_m \mathbf{V}^{(m)} \times_n \mathbf{U}^{(n)}$.

(d) If $\mathbf{W} \in \mathbb{C}^{P \times J_n}$ *then* $\mathcal{X} \times_n \mathbf{U}^{(n)} \times_n \mathbf{W} = \left(\mathcal{X} \times_n \mathbf{U}^{(n)}\right) \times_n \mathbf{W} = \mathcal{X} \times_n \left(\mathbf{W}\mathbf{U}^{(n)}\right) = \mathcal{X} \times_n \mathbf{W}\mathbf{U}^{(n)}$.

Tensor matricization: Process of reordering the elements of the tensor into a matrix is known as matricization or unfolding. The mode-n matricization of a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is denoted as $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$ and is obtained by arranging $\mathcal{Y}$'s mode-n fibers to be the columns of the resulting matrix. Unfolding the tensor $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)} =: \mathcal{X} \bigtimes_{n=1}^{N} \mathbf{U}^{(n)}$ along mode-n is equivalent to

$$\mathbf{Y}_{(n)} = \mathbf{U}^{(n)} \mathbf{X}_{(n)} (\mathbf{U}^{(N)} \otimes \ldots \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \ldots \otimes \mathbf{U}^{(1)})^{\top}, \tag{4.1}$$

where $\otimes$ is the matrix Kronecker product. In particular, (4.1) implies that the matricization $\left(\mathcal{X} \times_n \mathbf{U}^{(n)}\right)_{(n)} = \mathbf{U}^{(n)} \mathbf{X}_{(n)}$.[1]

It is worth noting that trivial inner product preserving isomorphisms exist between a tensor space $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and any of its matricized versions (i.e., mode-n matricization can be viewed as an isomorphism between the original tensor vector space $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and its mode-n matricized target vector space $\mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$). In particular, the process of matricizing tensors is linear. If, for example, $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ then one can see that the mode-n matricization of $\mathcal{X} + \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is $(\mathcal{X} + \mathcal{Y})_{(n)} = \mathbf{X}_{(n)} + \mathbf{Y}_{(n)}$ for all modes $n \in [N]$.

Tensor Rank: Unlike matrices, which have a unique definition of rank, there are multiple rank definitions for tensors including *tensor rank* and *tensor n-rank*. The *rank* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$

[1] Simply set $\mathbf{U}^{(m)} = \mathbf{I}$ (the identity) for all $m \neq n$ in (4.1). This fact also easily follows directly from the definition of the mode-n product.

is the smallest number of rank-one tensors that form $\mathcal{X}$ as their sum. The *n-rank* of $\mathcal{X}$ is the collection of ranks of unfoldings $\mathbf{X}_{(n)}$ and is denoted as:

$$n\text{-rank}(\mathcal{X}) = \left(\text{rank}(\mathbf{X}_{(1)}),\ \text{rank}(\mathbf{X}_{(2)}), ...,\ \text{rank}(\mathbf{X}_{(N)})\right). \tag{4.2}$$

Tensor inner product: The inner product of two same sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is the sum of the products of their elements.

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} x_{i_1,i_2,...,i_N} y_{i_1,i_2,...,i_N}. \tag{4.3}$$

It is not too difficult to see that matricization preserves Hilbert-Schmidt/Frobenius matrix inner products. That is, that $\langle \mathcal{X}, \mathcal{Y} \rangle = \left\langle \mathbf{X}_{(n)}, \mathbf{Y}_{(n)} \right\rangle_{\text{F}} = \text{Trace}\left(\mathbf{X}_{(n)}^{\top} \mathbf{Y}_{(n)}\right)$ holds for all $n \in [N]$. If $\langle \mathcal{X}, \mathcal{Y} \rangle = 0$, $\mathcal{X}$ and $\mathcal{Y}$ are *orthogonal*.

Tensor norm: Norm of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is the square root of the sum of the squares of all its elements.

$$\| \mathcal{X} \| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} x_{i_1,i_2,...,i_N}^2}. \tag{4.4}$$

The fact that matricization preserves Frobenius matrix inner products also means that it preserves Frobenius matrix norms. As a result we have that $\|\mathcal{X}\| = \left\|\mathbf{X}_{(n)}\right\|_{\text{F}}$ holds for all $n \in [N]$. If $\mathcal{X}$ and $\mathcal{Y}$ are orthogonal and also have unit norm (i.e., have $\|\mathcal{X}\| = \|\mathcal{Y}\| = 1$) we will say that they are an *orthonormal* pair.

4.2.2 Some Useful Facts Concerning Mode-n Products and Orthogonality

Let $\mathbf{I} \in \mathbb{R}^{I_n \times I_n}$ be the identity matrix. Given a (low rank) orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{I_n \times I_n}$ one can decompose any given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ into two orthogonal tensors using Lemma 1 (b)

$$\mathcal{X} = \mathcal{X} \times_n \mathbf{I} = \mathcal{X} \times_n ((\mathbf{I} - \mathbf{P}) + \mathbf{P}) = \mathcal{X} \times_n (\mathbf{I} - \mathbf{P}) + \mathcal{X} \times_n \mathbf{P}.$$

To check that the last two summands are orthogonal one can use (4.1) to compute that

$$\langle \mathcal{X} \times_n (\mathbf{I} - \mathbf{P}), \mathcal{X} \times_n \mathbf{P} \rangle = \Big\langle (\mathbf{I} - \mathbf{P})\mathbf{X}_{(n)}, \mathbf{P}\mathbf{X}_{(n)} \Big\rangle_{\mathrm{F}} = \mathrm{Trace}\left(\mathbf{X}_{(n)}^{\top}(\mathbf{I} - \mathbf{P})\mathbf{P}\mathbf{X}_{(n)}\right) = 0.$$

As a result one can also verify that the Pythagorean theorem holds, i.e., that $\|\mathcal{X}\|^2 = \|\mathcal{X} \times_n \mathbf{P}\|^2 + \|\mathcal{X} \times_n (\mathbf{I} - \mathbf{P})\|^2$.

If we now regard $\mathcal{X} \times_n \mathbf{P}$ as a low rank approximation to $\mathcal{X}$ then we can see that its approximation error

$$\mathcal{X} - \mathcal{X} \times_n \mathbf{P} = \mathcal{X} \times_n (\mathbf{I} - \mathbf{P})$$

is orthogonal to the low rank approximation $\mathcal{X} \times_n \mathbf{P}$, as one would expect. Furthermore, the norm of its approximation error satisfies $\|\mathcal{X} \times_n (\mathbf{I} - \mathbf{P})\|^2 = \|\mathcal{X}\|^2 - \|\mathcal{X} \times_n \mathbf{P}\|^2$. By continuing to use similar ideas in combination with lemma 1 for all modes one can prove the following more general Pythagorean result (see, e.g., theorem 5.1 in [241]).

Lemma 2. *Let* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *and* $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ *be an orthogonal projection matrix for all* $n \in [N]$. *Then,*

$$\begin{aligned}\left\| \mathcal{X} - \mathcal{X} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)} \right\|^2 &=: \left\| \mathcal{X} - \mathcal{X} \times_{n=1}^{N} \mathbf{U}^{(n)} \right\|^2 \\ &= \sum_{n=1}^{N} \left\| \mathcal{X} \times_{h=1}^{n-1} \mathbf{U}^{(h)} \times_n \left(\mathbf{I} - \mathbf{U}^{(n)} \right) \right\|^2.\end{aligned}$$

4.2.3 Higher Order Singular Value Decomposition (HoSVD)

Any tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ can be decomposed as mode products of a core tensor $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with N orthogonal matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ each of which is composed of the left singular vectors of $\mathbf{X}_{(n)}$ [35]:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \ldots \times_N \mathbf{U}^{(N)} = \mathcal{C} \bigtimes_{n=1}^{N} \mathbf{U}^{(n)} \tag{4.5}$$

where $\mathcal{C}$ is computed as

$$\mathcal{C} = \mathcal{X} \times_1 \left(\mathbf{U}^{(1)} \right)^\top \times_2 \left(\mathbf{U}^{(2)} \right)^\top \ldots \times_N \left(\mathbf{U}^{(N)} \right)^\top. \tag{4.6}$$

Let $\mathcal{C}_{i_n=\alpha}$ be a subtensor of $\mathcal{C}$ obtained by fixing the nth index to α. This subtensor satisfies the following properties:

- all-orthogonality: $\mathcal{C}_{i_n=\alpha}$ and $\mathcal{C}_{i_n=\beta}$ are orthogonal for all possible values of n, α and β subject to $\alpha \neq \beta$.

$$\langle \mathcal{C}_{i_n=\alpha}, \mathcal{C}_{i_n=\beta} \rangle = 0 \; when \; \alpha \neq \beta. \tag{4.7}$$

- ordering:

$$\| \mathcal{C}_{i_n=1} \| \geqslant \| \mathcal{C}_{i_n=2} \| \geqslant ... \geqslant \| \mathcal{C}_{i_n=I_n} \| \geqslant 0 \tag{4.8}$$

for all possible values of n.

4.3 Multiscale Analysis of Higher-order Datasets

In this section, we present a new tensor decomposition method named Multiscale HoSVD (MS-HoSVD) for an Nth order tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$. The proposed method recursively applies the following two-step approach: (i) Low-rank tensor approximation, followed by (ii) Decomposing the residual (original minus low-rank) tensor into subtensors.

A tensor $\mathcal{X}$ is decomposed using HoSVD as follows:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} ... \times_N \mathbf{U}^{(N)}, \tag{4.9}$$

where the $\mathbf{U}^{(n)}$'s are the left singular vectors of the unfoldings $\mathbf{X}_{(n)}$. The low-rank approximation of $\mathcal{X}$ is obtained by

$$\hat{\mathcal{X}}_0 = \mathcal{C}_0 \times_1 \hat{\mathbf{U}}^{(1)} \times_2 \hat{\mathbf{U}}^{(2)} ... \times_N \hat{\mathbf{U}}^{(N)} \tag{4.10}$$

where $\hat{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times r_n}$s are the truncated matrices obtained by keeping the first r_n columns of $\mathbf{U}^{(n)}$ and $\mathcal{C}_0 = \mathcal{X} \times_1 \left(\hat{\mathbf{U}}^{(1)}\right)^\top \times_2 \left(\hat{\mathbf{U}}^{(2)}\right)^\top ... \times_N \left(\hat{\mathbf{U}}^{(N)}\right)^\top$. The multilinear-rank of $\hat{\mathcal{X}}_0$, $\{r_1, ..., r_N\}$, can either be given a *priori*, or an energy criterion can be used to determine the minimum number of singular values to keep along each mode as:

$$r_n = \arg\min_i \sum_{l=1}^{i} \sigma_l^{(n)} \quad s.t. \quad \frac{\sum_{l=1}^{i} \sigma_l^{(n)}}{\sum_{l=1}^{I_n} \sigma_l^{(n)}} > \tau, \tag{4.11}$$

where $\sigma_l^{(n)}$ is the lth singular value of the matrix obtained from the SVD of the unfolding $\mathbf{X}_{(n)}$, and τ is an energy threshold. Once $\hat{\mathcal{X}}_0$ is obtained, the tensor $\mathcal{X}$ can be written as

$$\mathcal{X} = \hat{\mathcal{X}}_0 + \mathcal{W}_0, \tag{4.12}$$

where $\mathcal{W}_0$ is the residual tensor.

For the first scale analysis, to better encode the details of $\mathcal{X}$, we adapted an idea similar to the one presented in [242, 243]. The 0^{th} scale residual tensor, $\mathcal{W}_0$ is first decomposed into subtensors as follows. $\mathcal{W}_0 \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is unfolded across each mode yielding $\mathbf{W}_{0,(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode-n fibers of $\mathcal{W}_0$. For each mode, rows of $\mathbf{W}_{0,(n)}$ are partitioned into c_n non-overlapping clusters using a clustering algorithm such as local subspace analysis (LSA) [244] in order to encourage the formation of new subtensors which are intrinsically lower rank, and therefore better approximated via a smaller HoSVD at the next scale. The Cartesian product of the partitioning labels coming from the N modes yields $K = \prod_{i=1}^{N} c_i$ disjoint subtensors $\mathcal{X}_{1,k}$ where $k \in [K]$.

Let J_0^n be the index set corresponding to the nth mode of $\mathcal{W}_0$ with $J_0^n = [I_n]$, and let $J_{1,k}^n$ be the index set of the subtensor $\mathcal{X}_{1,k}$ for the nth mode, where $J_{1,k}^n \subset J_0^n$ for all $k \in [K]$ and $n \in [N]$. Index sets of subtensors for the nth mode satisfy $\bigcup_{k=1}^{K} J_{1,k}^n = J_0^n$ for all $n \in [N]$. For example, the index set of the first subtensor $\mathcal{X}_{1,1}$ can be written as $J_{1,1}^1 \times\ J_{1,1}^2 \times \ldots \times J_{1,1}^N$ and the kth subtensor $\mathcal{X}_{1,k} \in \mathbb{R}^{\left|J_{1,k}^1\right| \times \left|J_{1,k}^2\right| \times \cdots \times \left|J_{1,k}^N\right|}$ is obtained by

$$\begin{aligned} \mathcal{X}_{1,k}(i_1,\ i_2, \ldots,\ i_N) &= \mathcal{W}_0(J_{1,k}^1(i_1),\ J_{1,k}^2(i_2), \ldots,\ J_{1,k}^N(i_N)), \\ \mathcal{X}_{1,k} &= \mathcal{W}_0(J_{1,k}^1 \times J_{1,k}^2 \times \ldots \times J_{1,k}^N), \end{aligned} \tag{4.13}$$

where $i_n \in \left[\left|J^n_{1,k}\right|\right]$. Low-rank approximation for each subtensor is obtained by applying HoSVD as:

$$\hat{\mathcal{X}}_{1,k} = \mathcal{C}_{1,k} \times_1 \hat{\mathbf{U}}^{(1)}_{1,k} \times_2 \hat{\mathbf{U}}^{(2)}_{1,k} \ldots \times_N \hat{\mathbf{U}}^{(N)}_{1,k}, \tag{4.14}$$

where $\mathcal{C}_{1,k}$ and $\hat{\mathbf{U}}^{(n)}_{1,k} \in \mathbb{R}^{|J^n_{1,k}| \times r^{(n)}_{1,k}}$s correspond to the core tensor and low-rank projection basis matrices of $\mathcal{X}_{1,k}$, respectively. We can then define $\hat{\mathcal{X}}_1$ as the 1^{st} scale approximation of $\mathcal{X}$ formed by mapping all of the subtensors onto $\hat{\mathcal{X}}_{1,k}$ as follows:

$$\hat{\mathcal{X}}_1(J^1_{1,k} \times J^2_{1,k} \times \ldots \times J^n_{1,k}) = \hat{\mathcal{X}}_{1,k}. \tag{4.15}$$

Similarly, 1^{st} scale residual tensor is obtained by

$$\mathcal{W}_1(J^1_{1,k} \times J^2_{1,k} \times \ldots \times J^n_{1,k}) = \mathcal{W}_{1,k}, \tag{4.16}$$

where $\mathcal{W}_{1,k} = \mathcal{X}_{1,k} - \hat{\mathcal{X}}_{1,k}$. Therefore, $\mathcal{X}$ can be rewritten as:

$$\mathcal{X} = \hat{\mathcal{X}}_0 + \mathcal{W}_0 = \hat{\mathcal{X}}_0 + \hat{\mathcal{X}}_1 + \mathcal{W}_1. \tag{4.17}$$

Continuing in this fashion the j^{th} scale approximation of $\mathcal{X}$ is obtained by partitioning $\mathcal{W}_{j-1,k}$s into subtensors $\mathcal{X}_{j,k}$s and fitting a low-rank model to each one of them in a similar fashion. Finally, the j^{th} scale decomposition of $\mathcal{X}$ can be written as:

$$\mathcal{X} = \sum_{i=0}^{j} \hat{\mathcal{X}}_i + \mathcal{W}_j. \tag{4.18}$$

Algorithm 4.1 describes the pseudo code for this approach.

Algorithm 4.1 Multiscale HoSVD

Input: $\mathcal{X}$: tensor , $\mathbf{C} = (c_1,\ c_2,\ ...,\ c_N)$: the desired number of clusters for each mode, s_H: the highest scale of MS-HoSVD.
Output: T: Tree structure containing the MS-HoSVD decomposition of $\hat{\mathcal{X}}$.
Create an empty tree T
Create an empty list L
Add the node containing $\mathcal{X} =: \mathcal{X}_{0,1}$ to L with $\mathrm{Parent}(0,1) = \varnothing$ (i.e., this is the root of the the tree).
while L is not empty. **do**
    Pop a node corresponding to $\mathcal{X}_{s,t}$ (the tth subtensor from sth scale) from the list L where $s \in \{0, ..., s_H\}$ and $t \in \{1, ..., K^s\}$.
    $\mathcal{C}_{s,t}, \left\{\hat{\mathbf{U}}_{s,t}^{(n)}\right\} \leftarrow$ truncatedHOSVD($\mathcal{X}_{s,t}$).
    Add the node containing $\mathcal{C}_{s,t}, \left\{\hat{\mathbf{U}}_{s,t}^{(n)}\right\}$ to T as a child of $\mathrm{Parent}(s,t)$.
    if $s < s_H$ **then**
        Compute $\mathcal{W}_{s,t} = \mathcal{X}_{s,t} - \hat{\mathcal{X}}_{s,t}$.
        Create K subtensors $\mathcal{X}_{s+1,K(t-1)+k}$ with $J^n_{s+1,K(t-1)+k}$ from $\mathcal{W}_{s,t}$ where $k \in \{1,\ 2,\ ...,\ K\}$ and $n \in \{1,\ 2,\ ...,\ N\}$.
        Add K nodes containing $\mathcal{X}_{s+1,K(t-1)+k}$ and $\left\{J^n_{s+1,K(t-1)+k}\right\}$ to L with $\mathrm{Parent}(s+1, K(t-1)+k) = (s,t)$.
    end if
end while

4.3.1 Memory Cost of the First Scale Decomposition

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times .... \times I_N}$ be an Nth order tensor. To simplify the notation, assume that the dimension of each mode is the same, i.e. $I_1 = I_2 = ... = I_N = I$. Assume $\mathcal{X}$ is approximated by HoSVD as:

$$\hat{\mathcal{X}} = \mathcal{C}_H \times_1 \mathbf{U}_H^{(1)} \times_2 \mathbf{U}_H^{(2)} ... \times_N \mathbf{U}_H^{(N)}, \tag{4.19}$$

by fixing the rank of each mode matrix as $\mathrm{rank}(\mathbf{U}_H^{(i)}) = r_H$ for $i \in \{1,\ 2, ...,\ N\}$. Let $\mathbb{F}(\cdot)$ be a function that quantifies the memory cost, then the storage cost of $\mathcal{X}$ decomposed by HoSVD is $\mathbb{F}(\mathcal{C}_H) + \sum_{i=1}^{N}(\mathbb{F}(\mathbf{U}_H^{(i)})) \approx r_H^N + NIr_H$.

For multiscale analysis at scale 1, $\hat{\mathcal{X}} = \hat{\mathcal{X}}_0 + \hat{\mathcal{X}}_1$. The cost of storing $\hat{\mathcal{X}}_0$ is $\mathbb{F}(\mathcal{C}_0) + \sum_{i=1}^{N}(\mathbb{F}(\hat{\mathbf{U}}^{(i)})) \approx r_0^N + NIr_0$ where the rank of each mode matrix is fixed at $\mathrm{rank}(\mathbf{U}^{(i)}) = r_0$

for $i \in \{1, 2, ..., N\}$. The cost of storing $\hat{\mathcal{X}}_1$ is the sum of the storage costs for each of the $K = \prod_{i=1}^{N} c(i)$ subtensors $\hat{\mathcal{X}}_{1,k}$. Assume $c(i) = c$ for all $i \in \{1, 2, ..., N\}$ yielding c^N equally sized subtensors, and that each $\hat{\mathcal{X}}_{1,k}$ is decomposed using the HoSVD as $\hat{\mathcal{X}}_{1,k} = \mathcal{C}_{1,k} \times_1 \hat{\mathbf{U}}_{1,k}^{(1)} \times_2 \hat{\mathbf{U}}_{1,k}^{(2)} ... \times_N \hat{\mathbf{U}}_{1,k}^{(N)}$. Let the rank of each mode matrix be fixed as rank$(\hat{\mathbf{U}}_{1,k}^{(i)}) = r_1$ for all $i \in \{1, 2, ..., N\}$ and $k \in \{1, 2, ..., K\}$. Then, the memory cost for the first scale is $\sum_{k=1}^{K} \left(\mathbb{F}(\mathcal{C}_{1,k}) + \sum_{i=1}^{N} \mathbb{F}(\hat{\mathbf{U}}_{1,k}^{(i)}) \right) \approx c^N \left(r_1^N + \frac{NIr_1}{c} \right)$. Choosing $r_1 \lesssim \frac{r_0}{c^{(N-1)}}$ ensures that the storage cost does not grow exponentially so that $\mathbb{F}(\hat{\mathcal{X}}_1) < \mathbb{F}(\hat{\mathcal{X}}_0)$ since the total cost becomes approximately equal to $r_0^N \left(1 + \frac{1}{c^{N^2-2N}} \right) + 2NIr_0$. Thus, picking $r_0 \approx r_H/2$ can now provide lower storage cost for the first scale analysis than for HoSVD.

4.3.2 Computational Complexity

The computational complexity of MS-HoSVD at the first scale is equal to the sum of computational complexity of computing HoSVD at the parent node, partitioning into subtensors and computing HoSVD for each one of the subtensors. Computational complexity of HoSVD of an N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ where $I_1 = I_2 = ... = I_N = I$ is $\mathcal{O}\left(NI^{(N+1)}\right)$ [245]. By assuming that the partitioning is performed using K-means (via Lloyd's algorithm) with $c_i = c$ along each mode, the complexity partitioning along each mode is $\mathcal{O}\left(NI^N ci\right)$, where i is the number of iterations used in Lloyd's algorithm. Finally, the total complexity of applying the HoSVD to c^N equally sized subtensors is $\mathcal{O}\left(c^N N(I/c)^{(N+1)}\right)$. Therefore, first scale MS-HoSVD has a total computational complexity of $\mathcal{O}\left(NI^{(N+1)} + NI^N ci + c^N N(I/c)^{(N+1)}\right)$. Note that this complexity is similar to that of the HoSVD whenever ci is small compared to I. The runtime complexity of these multiscale methods can be reduced even further by computing the HoSVDs for different subtensors in parallel whenever possible, as well as by utilizing distributed and parallel SVD algorithms such as [246] when computing all the required HoSVD decompositions.

4.3.3 A Linear Algebraic Representation of the Proposed Multiscale HoSVD Approach

Definitions: Though the tree-based representation of the proposed MS-HoSVD approach used above in Algorithm 4.1 is useful for algorithmic development, it is somewhat less useful for theoretical error analysis. In this subsection we will develop formulas for the proposed MS-HoSVD approach which are more amenable to error analysis. In the process we will also formulate a criterion which, when satisfied, guarantees that the proposed fist scale MS-HoSVD approach produces an accurate multiscale approximation to a given tensor.

We can construct full size first scale subtensors of the residual tensor $\mathcal{W}_0 \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ from (4.12), $\mathcal{X}|_k \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ for all $k \in [K]$, using the index sets $J_{1,k}^n$ from (4.13) along with diagonal restriction matrices. Let $\mathbf{R}_k^{(n)} \in \{0,\ 1\}^{I_n \times I_n}$ be the diagonal matrix with entries given by

$$\mathbf{R}_k^{(n)}(i,j) = \begin{cases} 1, & \text{if } i = j, \text{ and } j \in J_{1,k}^n \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

for all $k \in [K]$, and $n \in [N]$. We then define

$$\mathcal{X}|_k := \mathcal{W}_0 \bigtimes_{n=1}^{N} \mathbf{R}_k^{(n)} = \mathcal{W}_0 \times_1 \mathbf{R}_k^{(1)} \times_2 \mathbf{R}_k^{(2)} \ldots \times_N \mathbf{R}_k^{(N)}. \tag{4.21}$$

Thus, the kth subtensor $\mathcal{X}|_k$ will only have nonzero entries, given by $\mathcal{W}_0(J_{1,k}^1 \times \ldots \times J_{1,k}^N)$, in the locations indexed by the sets $J_{1,k}^n$ from above. The properties of the index sets $J_{1,k}^n$ furthermore guarantee that these subtensors all have disjoint support. As a result both

$$\mathcal{W}_0 = \sum_{k=1}^{K} \mathcal{X}|_k \tag{4.22}$$

and

$$\left\langle \mathcal{X}|_k, \mathcal{X}|_j \right\rangle = 0 \text{ for all } j, k \in [K] \text{ with } j \neq k$$

will always hold.

Recall that we want to compute the HoSVD of the subtensors we form at each scale in order to create low-rank projection basis matrices along the lines of those in (4.14). Toward this end we compute the top $r_k^{(n)} \leqslant rank(\mathbf{R}_k^{(n)}) = |J_{1,k}^n|$ left singular vectors of the mode-n matricization of each $\mathcal{X}|_k$, $\mathbf{X}|_{\mathbf{k}(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$, for all $n \in [N]$. Note that $\mathbf{X}|_{\mathbf{k}(n)} = \mathbf{R}_k^{(n)} \mathbf{X}|_{\mathbf{k}(n)}$ always holds for these matricizations since $\mathbf{R}_k^{(n)}$ is a projection matrix.[2] Thus, the top $r_k^{(n)}$ left singular vectors of $\mathbf{X}|_{\mathbf{k}(n)}$ will only have nonzero entries in locations indexed by $J_{1,k}^n$. Let $\hat{\mathbf{U}}_k^{(n)} \in \mathbb{R}^{I_n \times r_k^{(n)}}$ be the matrix whose columns are these top singular vectors. As a result of the preceding discussion we can see that $\hat{\mathbf{U}}_k^{(n)} = \mathbf{R}_k^{(n)} \hat{\mathbf{U}}_k^{(n)}$ will hold for all $n \in [N]$ and $k \in [K]$. Our low rank projection matrices $\mathbf{Q}_k^{(n)} \in \mathbb{R}^{I_n \times I_n}$ used to produce low rank approximations of each subtensor $\mathcal{X}|_k$ can now be defined as

$$\mathbf{Q}_k^{(n)} := \hat{\mathbf{U}}_k^{(n)} \left(\hat{\mathbf{U}}_k^{(n)} \right)^\top. \tag{4.23}$$

As a consequence of $\hat{\mathbf{U}}_k^{(n)} = \mathbf{R}_k^{(n)} \hat{\mathbf{U}}_k^{(n)}$ holding, combined with the fact that $\left(\mathbf{R}_k^{(n)} \right)^\top = \mathbf{R}_k^{(n)}$ since each $\mathbf{R}_k^{(n)}$ matrix is diagonal, we have that

$$\begin{aligned} \mathbf{Q}_k^{(n)} &:= \hat{\mathbf{U}}_k^{(n)} \left(\hat{\mathbf{U}}_k^{(n)} \right)^\top = \mathbf{R}_k^{(n)} \hat{\mathbf{U}}_k^{(n)} \left(\mathbf{R}_k^{(n)} \hat{\mathbf{U}}_k^{(n)} \right)^\top \\ &= \mathbf{R}_k^{(n)} \hat{\mathbf{U}}_k^{(n)} \left(\hat{\mathbf{U}}_k^{(n)} \right)^\top \left(\mathbf{R}_k^{(n)} \right)^\top = \mathbf{R}_k^{(n)} \mathbf{Q}_k^{(n)} \mathbf{R}_k^{(n)} \end{aligned} \tag{4.24}$$

holds for all $n \in [N]$ and $k \in [K]$. Using (4.24) combined with the fact that $\mathbf{R}_k^{(n)}$ is a projection

[2] Here we are implicitly using (4.1).

matrix we can further see that

$$\begin{aligned}\mathbf{R}_k^{(n)}\mathbf{Q}_k^{(n)} = \mathbf{R}_k^{(n)}\left(\mathbf{R}_k^{(n)}\mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\right) &= \mathbf{R}_k^{(n)}\mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)} = \mathbf{Q}_k^{(n)} = \mathbf{R}_k^{(n)}\mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)} \\ &= \left(\mathbf{R}_k^{(n)}\mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\right)\mathbf{R}_k^{(n)} = \mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\end{aligned} \tag{4.25}$$

also holds for all $n \in [N]$ and $k \in [K]$.

1-scale Analysis of MS-HoSVD: Using this linear algebraic formulation we are now able to re-express the the 1^{st} scale approximation of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$, $\hat{\mathcal{X}}_1 \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$, as well as the 1^{st} scale residual tensor tensor, $\mathcal{W}_1 \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$, as follows (see (4.15) – (4.17)). We have that

$$\begin{aligned}\hat{\mathcal{X}}_1 = \sum_{k=1}^{K}\left(\mathcal{X}|_k \bigtimes_{n=1}^{N}\mathbf{Q}_k^{(n)}\right) &= \sum_{k=1}^{K}\left(\mathcal{W}_0 \bigtimes_{n=1}^{N}\mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\right) && \text{(Using Lemma 1 and (4.21))} \\ &= \sum_{k=1}^{K}\left(\mathcal{W}_0 \bigtimes_{n=1}^{N}\mathbf{Q}_k^{(n)}\right) && \text{(Using the properties in (4.25))} \\ &= \sum_{k=1}^{K}\left(\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \bigtimes_{n=1}^{N}\mathbf{Q}_k^{(n)}\right) && \text{(Using (4.12))}\end{aligned} \tag{4.26}$$

holds. Thus, we see that the residual error $\mathcal{W}_1$ from (4.17) satisfies

$$\mathcal{X} = \hat{\mathcal{X}}_0 + \sum_{k=1}^{K}\left(\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \bigtimes_{n=1}^{N}\mathbf{Q}_k^{(n)}\right) + \mathcal{W}_1. \tag{4.27}$$

Having derived (4.27) it behooves us to consider when using such a first scale approximation of $\mathcal{X}$ is actually better than, e.g., just using a standard HoSVD-based 0^{th} scale approximation of $\mathcal{X}$ along the lines of (4.12). As one might expect, this depends entirely on (i) how well the 1^{st} scale partitions (i.e., the restriction matrices utilized (4.20)) are chosen, as well as on (ii) how

well restriction matrices of the type used in (4.20) interact with the projection matrices used to create the standard HoSVD-based approximation in question. Toward understanding these two conditions better, recall that $\hat{\mathcal{X}}_0 \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$ in (4.27) is defined as

$$\hat{\mathcal{X}}_0 = \mathcal{X} \bigtimes_{n=1}^{N} \mathbf{P}^{(n)} = \mathcal{X} \times_1 \mathbf{P}^{(1)} \times_2 \mathbf{P}^{(2)} \cdots \times_N \mathbf{P}^{(N)} \tag{4.28}$$

where the orthogonal projection matrices $\mathbf{P}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ are given by $\mathbf{P}^{(n)} = \hat{\mathbf{U}}^{(n)} \left(\hat{\mathbf{U}}^{(n)}\right)^{\top}$ for the matrices $\hat{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times r_n}$ used in (4.10). For simplicity let the ranks of the $\mathbf{P}^{(n)}$ projection matrices momentarily satisfy $r_1 = r_2 = \cdots = r_N =: r_0$ (i.e., let them all be rank $r_0 < \max_n\{\text{rank}(\mathbf{X}_{(n)})\}$). Similarly, let all the ranks, $r_k^{(n)}$, of the 1^{st} scale projection matrices $\mathbf{Q}_k^{(n)}$ in (4.23) be r_1 for the time being.

Motivated by, e.g., the memory cost analysis of Section 4.3.1 above, one can now ask when the multiscale approximation error, $\|\mathcal{W}_1\|$, resulting from (4.27) will be less than a standard HoSVD-based approximation error, $\|\mathcal{X} - \bar{\mathcal{X}}_0\|$, where

$$\bar{\mathcal{X}}_0 := \mathcal{X} \bigtimes_{n=1}^{N} \bar{\mathbf{P}}^{(n)} = \mathcal{X} \times_1 \bar{\mathbf{P}}^{(1)} \times_2 \bar{\mathbf{P}}^{(2)} \cdots \times_N \bar{\mathbf{P}}^{(N)}, \tag{4.29}$$

and each orthogonal projection matrix $\bar{\mathbf{P}}^{(n)}$ is of rank $\bar{r}_n = r_H \geqslant 2r_0 \geqslant r_0 + c^{N-1} r_1$ $\big($i.e., where each $\bar{\mathbf{P}}^{(n)}$ projects onto the top r_H left singular vectors of $\mathbf{X}_{(n)}\big)$. In this situation having both $\|\mathcal{W}_1\| < \|\mathcal{X} - \bar{\mathcal{X}}_0\|$ and $r_H \geqslant 2r_0 \geqslant r_0 + c^{N-1} r_1$ hold at the same time would imply that one could achieve smaller approximation error using MS-HoSVD than using HoSVD while simultaneously achieving better compression (recall Section 4.3.1). In order to help facilitate such analysis we prove error bounds in Appendix that are implied by the choice of a good partitioning scheme for the residual tensor $\mathcal{W}_0$ in (4.20) – (4.22).

In particular, with respect to the question concerning how well the 1^{st}-scale approximation error, $\|\mathcal{W}_1\|$, from (4.27) might compare to the HoSVD-based approximation error $\|\mathcal{X} - \bar{\mathcal{X}}_0\|$ we can use the following notion of an *effective partition of* $\mathcal{W}_0$. The partition of $\mathcal{W}_0$ formed by the restriction matrices $\mathbf{R}_k^{(n)}$ in (4.20) – (4.22) will be called *effective* if there exists another *pessimistic* partitioning of $\mathcal{W}_0$ via (potentially different) restriction matrices $\left\{\tilde{\mathbf{R}}_k^{(n)}\right\}_{k=1}^{K}$ together with a bijection $f : [K] \to [K]$ such that

$$\sum_{n=1}^{N} \left\| \mathcal{X}|_k \times_n \left(\mathbf{I} - \mathbf{Q}_k^{(n)}\right)\right\|^2 \leqslant \sum_{n=1}^{N} \left\| \mathcal{W}_0 \times_n \tilde{\mathbf{R}}_{f(k)}^{(n)} \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right) \bigtimes_{h\neq n}^{N} \tilde{\mathbf{R}}_{f(k)}^{(h)} \right\|^2 \tag{4.30}$$

holds for each $k \in [K]$. In (4.30) the $\left\{\tilde{\mathbf{P}}^{(n)}\right\}$ are the orthogonal projection matrices obtained from the HoSVD of $\mathcal{W}_0$ with ranks $\tilde{r}_n = r_H$ $\big($i.e., where each $\tilde{\mathbf{P}}^{(n)}$ projects onto the top $\tilde{r}_n = r_H$ left singular vectors of the matricization $\mathbf{W}_{0,(n)}\big)$. In Appendix, we show that (4.30) holding for $\mathcal{W}_0$ implies that the error $\|\mathcal{W}_1\|$ resulting from our 1^{st}-scale approximation in (4.27) is less than an upper bound of the type often used for HoSVD-based approximation errors of the form $\|\mathcal{X} - \bar{\mathcal{X}}_0\|$ (see, e.g., [241]). In particular, we prove the following result.

Theorem 1. *Suppose that* (4.30) *holds. Then, the first scale approximation error given by MS-HoSVD in* (4.27) *is bounded by*

$$\|\mathcal{W}_1\|^2 = \left\|\mathcal{X} - \hat{\mathcal{X}}_0 - \hat{\mathcal{X}}_1\right\|^2 \leqslant \sum_{n=1}^{N} \left\|\mathcal{X} \times_n \left(\mathbf{I} - \bar{\mathbf{P}}^{(n)}\right)\right\|^2,$$

where $\left\{\bar{\mathbf{P}}^{(n)}\right\}$ *are low-rank projection matrices of rank* $\bar{r}_n = \tilde{r}_n = r_H$ *obtained from the truncated HoSVD of* $\mathcal{X}$ *as per* (4.29).

Proof. See Appendix. □

Theorem 1 implies that $\|\mathcal{W}_1\|$ *may be less* than $\|\mathcal{X} - \bar{\mathcal{X}}_0\|$ when (4.30) holds. It does not,

however, actually prove that $\|\mathcal{W}_1\| \leqslant \|\mathcal{X} - \bar{\mathcal{X}}_0\|$ holds whenever (4.30) does. In fact, directly proving that $\|\mathcal{W}_1\| \leqslant \|\mathcal{X} - \bar{\mathcal{X}}_0\|$ whenever (4.30) holds does not appear to be easy. It also does not appear to be easy to prove the error bound in theorem 1 without an assumption along the lines of (4.30) which simultaneously controls both (i) how well the restriction matrices utilized to partition $\mathcal{W}_0$ in (4.21) are chosen, as well as (ii) how poorly (worst case) restriction matrices interact with the projection matrices used to create standard HoSVD-based approximations of $\mathcal{W}_0$ and/or $\mathcal{X}$. The development of simpler and/or weaker conditions than (4.30) which still yield meaningful error guarantees along the lines of theorem 1 is left as future work.

Considering condition (4.30) above, we note that experiments show that it is regularly satisfied on real datasets when (i) the effective restriction matrices $\left\{\mathbf{R}_k^{(n)}\right\}_{k=1}^{K}$ in (4.20) – (4.22) are first formed by clustering the rows of each unfolding of $\mathcal{W}_0$ using, e.g., local subspace analysis (LSA), after which (ii) pessimistic restriction matrices $\left\{\tilde{\mathbf{R}}_k^{(n)}\right\}_{k=1}^{K}$ are randomly generated in order to create another (random) partition of $\mathcal{W}_0$ into K different disjoint subtensors for comparison. The bijection f can then be created by, e.g., (i) sorting the left-hand side errors in (4.30) for each $k \in [k]$, (ii) sorting the right-hand side errors in (4.30) for each $k \in [K]$, and then (iii) matching the largest left-hand and right-hand errors for comparision, the second largest left-hand and right-hand errors for comparision, etc.. When checked in this way the sorted right-hand side errors often dominate (entrywise) the sorted left-hand side errors for various reasonable ranks $\bar{r}_n = \tilde{r}_n = r_H = r_0 + c^{N-1}r_1$ (as a function of r_0 and r_1 with, e.g., $c = 2$) on every dataset considered in Section 4.4 below, thereby verifying that (4.30) does indeed regularly hold.

We refer the reader to the strong empirical performance of MS-HoSVD in Section 4.4 for additional evidence supporting the utility of (4.27) as a means of improving the compression performance of standard HoSVD-based compression techniques. In addition, we further refer the reader to Section 4.5 where it is empirically demonstrated that MS-HoSVD is also capable of se-

lecting more informative features than HoSVD-based methods for the purposes of classification. These two facts together provide strong evidence that combining the use of clustering-enhanced multiscale approximation with existing tensor factorization techniques can lead to improved performance in multiple application domains.

4.3.4 Adaptive Pruning in Multiscale HoSVD for Improved Performance

In order to better capture the local structure of the tensor, it is important to look at higher scale decompositions. However, as the scale increases, the storage cost and computational complexity will increase making any gain in reconstruction error potentially not worth the additional memory cost. For this reason, it is important to carefully select the subtensors adaptively at higher scales. To help avoid the redundancy in decomposition structure we propose an adaptive pruning method across scales.

In adaptive pruning, the tree is pruned by minimizing the following cost function $\mathbb{H} = Error + \lambda \cdot Compression$ similar to the rate-distortion criterion commonly used by compression algorithms where λ is the trade-off parameter [247]. To minimize this function we employ a greedy procedure similar to sequential forward selection [248]. First, the root node which stores $\hat{\mathcal{X}}_0$ is created and scale-1 subtensors $\hat{\mathcal{X}}_{1,k}$ are obtained from the 0th order residual tensor $\hat{\mathcal{W}}_0$ as discussed in Section 4.3. These subtensors are stored in a list and the subtensor which decreases the cost function the most is then added to the tree structure under its parent node. Next, scale-2 subtensors belonging to the added node are created and added to the list. All of the scale-1 and scale-2 subtensors in the list are again evaluated to find the subtensor that minimizes the cost function. This procedure is repeated until the cost function $\mathbb{H}$ converges or the decrease is minimal. A pseudocode of the algorithm is given in Algorithm 4.2. It is important to note that this algorithm is suboptimal similar to other greedy search methods.

Algorithm 4.2 Multiscale HoSVD with Adaptive Pruning

Input: $\mathcal{X}$: tensor , $\mathbf{C} = (c_1,\ c_2,\ ...,\ c_N)$: the desired number of clusters for each modes, s_H: the highest scale of MS-HoSVD.
Output: T: Tree structure containing the MS-HoSVD decomposition of $\hat{\mathcal{X}}$.
Create an empty tree T.
Create an empty list L.
Add node containing $\mathcal{X}$ to L.
while There is a node in L that decreases the cost function $\mathbb{H}(T)$. **do**
    Find the node corresponding to $\mathcal{X}_{s,t}$ (the tth subtensor from sth scale) in the list L that decreases $\mathbb{H}$ the most where $s \in \{0, ..., s_H\}$ and $t \in \{1, ..., K^s\}$.
    $\mathcal{C}_{s,t}, \left\{\hat{\mathbf{U}}_{s,t}^{(n)}\right\} \leftarrow$ truncatedHOSVD($\mathcal{X}_{s,t}$).
    Add the node containing $\mathcal{C}_{s,t}, \left\{\hat{\mathbf{U}}_{s,t}^{(n)}\right\}$ to T.
    if $s < s_H$ **then**
        Compute $\mathcal{W}_{s,t} = \mathcal{X}_{s,t} - \hat{\mathcal{X}}_{s,t}$.
        Create K subtensors $\mathcal{X}_{s+1,K(t-1)+k}$ with $J^n_{s+1,K(t-1)+k}$ from $\mathcal{W}_{s,t}$ where $k \in \{1,\ 2,\ ...,\ K\}$ and $n \in \{1,\ 2,\ ...,\ N\}$.
        Add K nodes containing $\mathcal{X}_{s+1,K(t-1)+k}$ and $\left\{J^n_{s+1,K(t-1)+k}\right\}$ to L.
    end if
end while

4.4 Data Reduction

In this section, we demonstrate the performance of MS-HoSVD for tensor type data representation on 3-mode and 4-mode real datasets compared with three other tensor decompositions: HoSVD, H-Tucker and T-Train. The performance of tensor decomposition methods are evaluated in terms of reconstruction error and compression rate. In the tables and figures below the error rate refers to the normalized tensor approximation error $\frac{\|\mathcal{X}-\hat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F}$ and the compression rate is computed as $\frac{\#\text{ total bits to store }\hat{\mathcal{X}}}{\#\text{ total bits to store }\mathcal{X}}$. Moreover, we show the performance of the proposed adaptive tree prunning strategy for data reduction.

4.4.1 Datasets

4.4.1.1 PIE dataset

A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{244\times320\times138}$ is created from PIE dataset [249]. The tensor contains 138 images from 6 different yaw angles and varying illumination conditions collected from a subject where each image is converted to gray scale. Fig. 4.1 illustrates the images from different frames of the PIE dataset.

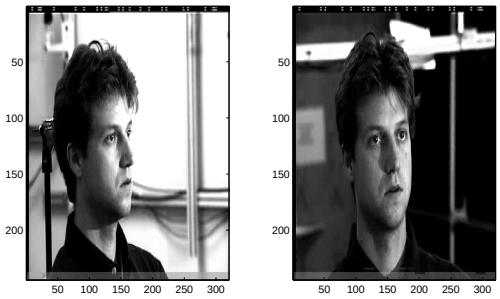


Figure 4.1: Sample frames from PIE dataset corresponding to the 30th (left) and 80th (right) frames.

4.4.1.2 COIL-100 dataset

The COIL-100 database contains 7200 images collected from 100 objects where the images of each object were taken at pose intervals of 5°. A 4-mode tensor $\mathcal{X} \in \mathbb{R}^{128\times128\times72\times100}$ is created from COIL-100 dataset [250]. The constructed 4-mode tensor contains 72 images of size 128×128 from 100 objects where each image is converted to gray scale. In Fig. 4.2, sample images of four objects taken from different angles can be seen.

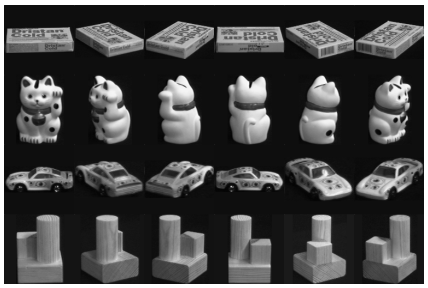


Figure 4.2: Image samples of four different objects from COIL-100 dataset from varying pose angles (from 0° to 240° with 60° increments).

4.4.1.3 The Cambridge Hand Gesture Dataset

The Cambridge hand gesture database consists of 900 image sequences of nine gesture classes of three primitive hand shapes and three primitive motions where each class contains 100 image sequences (5 different illuminations $\times$ 10 arbitrary motions $\times$ 2 subjects). In Fig. 4.3, sample image sequences collected for nine hand gestures can be seen. The created 4-mode tensor $\mathcal{X} \in \mathbb{R}^{60\times80\times30\times900}$ contains 900 image sequences of size $60 \times 80 \times 30$ where each image is converted to gray scale.

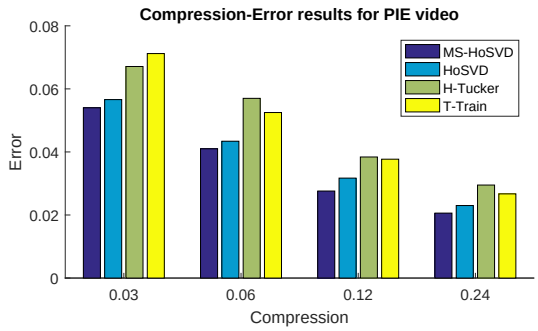
Figure 4.3: Illustration of nine different classes in Cambridge Hand Gesture Dataset.

4.4.2 Data Reduction Experiments

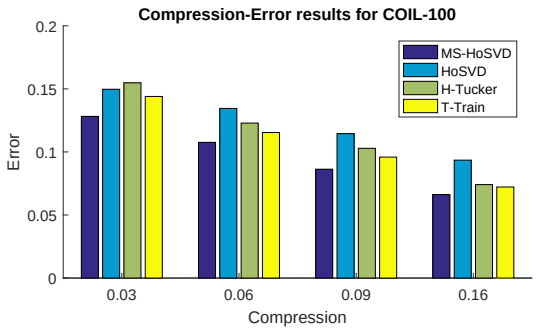
In this section, we evaluate the performance of MS-HoSVD for 1 and 2-scale decompositions compared to HoSVD, H-Tucker and T-Train decompositions. In the following experiments, tensor

partitioning is performed by LSA and the cluster number along each mode is chosen as $c_i = 2$. The rank used in HoSVD is selected adaptively using the energy criterion as described in Section 4.3. In our experiments, we performed MS-HoSVD with $\tau = 0.7$ and $\tau = 0.75$ and we kept τ the same for each scale. For the same compression rates as the MS-HoSVD, the reconstruction error of HoSVD, H-Tucker and T-Train models are computed.

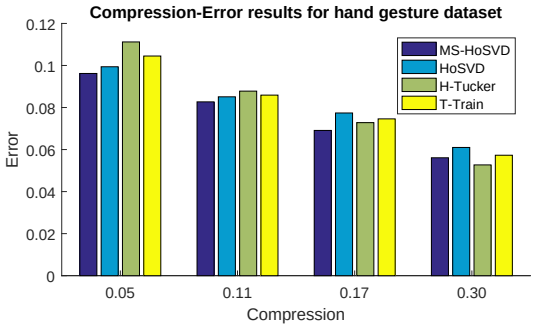
Fig. 4.4 explores the interplay between compression rate and approximation error for MS-HoSVD in comparison to HoSVD, H-Tucker and T-Train for PIE, COIL-100 and hand gesture datasets. Starting from the left on Figs. 4.4(a), 4.4(b) and 4.4(c), the first two compression rates correspond to 1-scale MS-HoSVD with $\tau = 0.7$ and $\tau = 0.75$, respectively while the last two are obtained from 2-scale approximation with $\tau = 0.7$ and $\tau = 0.75$, respectively. As seen in Fig. 4.4, MS-HoSVD outperforms other approaches with respect to reducing PIE, COIL-100 and hand gesture tensors at varying compression rates. Moreover, adding 2nd scale increases the storage requirements while decreasing the error of MS-HoSVD. Fig. 4.5 illustrates the influence of scale on the visual quality of the reconstructed images. As expected, introducing additional finer scales into a multiscale approximation of video data improves image detail in each frame. Morevoer, data reduction performance of T-Train seen better than H-Tucker in most of the experiments.



(a) PIE



(b) COIL-100



(c) Hand Gesture

Figure 4.4: Compression rate versus Normalized Reconstruction Error for MS-HoSVD (dark blue), HoSVD (light blue), H-Tucker (green) and T-Train (yellow) for a) PIE, b) COIL-100 and c) Hand Gesture datasets. Starting from the left for all (a), (b) and (c), the first two compression rates correspond to 1-scale MS-HoSVD with $\tau = 0.7$ and $\tau = 0.75$ while the last two are obtained from 2-scale approximation with $\tau = 0.7$ and $\tau = 0.75$, respectively. MS-HoSVD provides lower error than HoSVD, H-Tucker and T-Train.

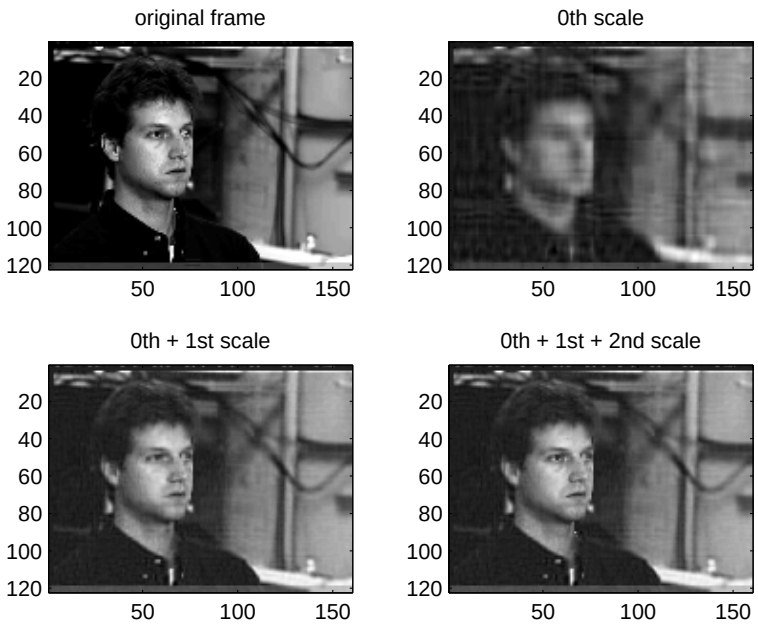


Figure 4.5: A single frame of the PIE dataset showing increasing accuracy with scale.

4.4.3 Data Reduction with Adaptive Tree Prunning

In this section, we evaluate the performance of adaptive tree pruning multiscale decompositions. In the pruning experiments, clustering is performed by LSA and the cluster number along each mode is chosen as $c_i = 2$. The rank used in HoSVD is selected adaptively based on the energy threshold which is 0.7. A pruned version of MS-HoSVD with 2-scale analysis that minimizes the cost function $\mathbb{H} = Error + \lambda \cdot Compression$ for 2-scale analysis is implemented for PIE, COIL-100 and Hand Gesture datasets with varying λ values as reported in Tables 4.1, 4.2 and 4.3. As λ increases, reducing the compression rate becomes more important and the algorithm prunes the leaf nodes more. For example, a choice of $\lambda = 0.75$ prunes all of the nodes corresponding to the second scale subtensors for PIE data (see Table 4.1).

As it can be seen from Tables 4.1, 4.2 and 4.3, the optimal tradeoff between reconstruction error and compression rate are achieved at different λ values for different data. For example, for

PIE data, increasing λ value does not provide much change in reconstruction error while increasing the compression accuracy. On the other hand, for COIL-100, $\lambda = 0.75$ provides a good tradeoff between reconstruction error and compression rate. On the other hand, small changes on λ yield significant effects on pruning the subtensors of 2-scale decomposition of hand gesture data. Fig. 4.6 illustrates the performance of the pruning algorithm on PIE dataset. Applying pruning with $\lambda = 0.25$ increases the reconstruction error from 0.0276 to 0.0506 while reducing the compression rate by 4 (Table 4.1). As seen in Fig. 4.6, the 2nd scale approximation obtained by adaptive pruning algorithm preserves most of the facial details in the image.

Table 4.1: Reconstruction error and compression rate computed for pruned tree structure obtained by applying MS-HoSVD with 2-scales to PIE data.

λ	0	0.22	0.25	0.30	0.75
Normalized error	0.0276	0.0395	0.0506	0.0530	0.0540
Compression	0.1241	0.0809	0.0377	0.0284	0.0261
Scales of subtensors	0+1+2	0+1+2	0+1+2	0+1+2	0+1

Table 4.2: Reconstruction error and compression rate computed for pruned tree structure obtained by applying MS-HoSVD with 2-scales to COIL-100 dataset.

λ	0	0.25	0.50	0.75	0.80
Normalized error	0.0857	0.0867	0.0913	0.1060	0.1207
Compression	0.0863	0.0840	0.0734	0.0526	0.0347
Scales of subtensors	0+1+2	0+1+2	0+1+2	0+1+2	0+1+2

Table 4.3: Reconstruction error and compression rate computed for pruned tree structure obtained by applying MS-HoSVD with 2-scales to Hand Gesture dataset.

λ	0	0.25	0.26	0.27	0.28
Normalized error	0.0691	0.0869	0.0913	0.0946	0.0999
Compression	0.1694	0.1056	0.0827	0.0698	0.0514
Scales of subtensors	0+1+2	0+1+2	0+1+2	0+1+2	0+1

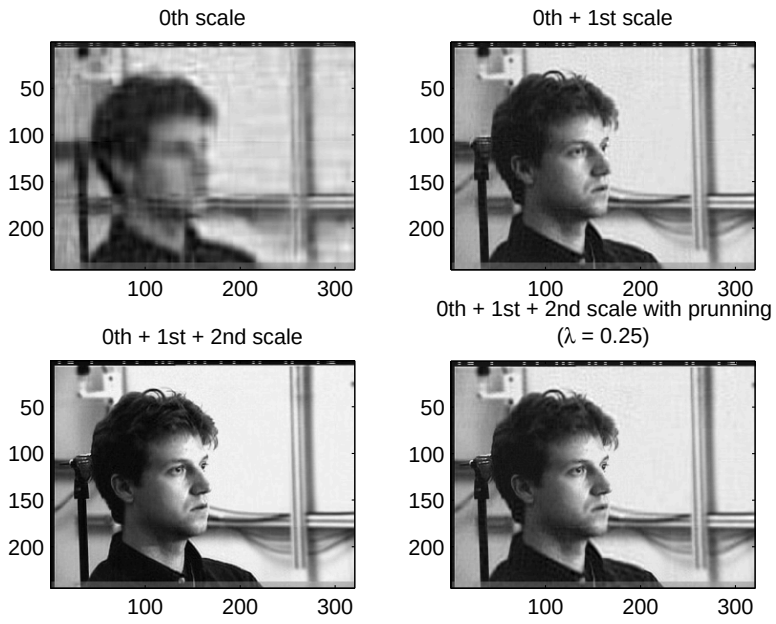
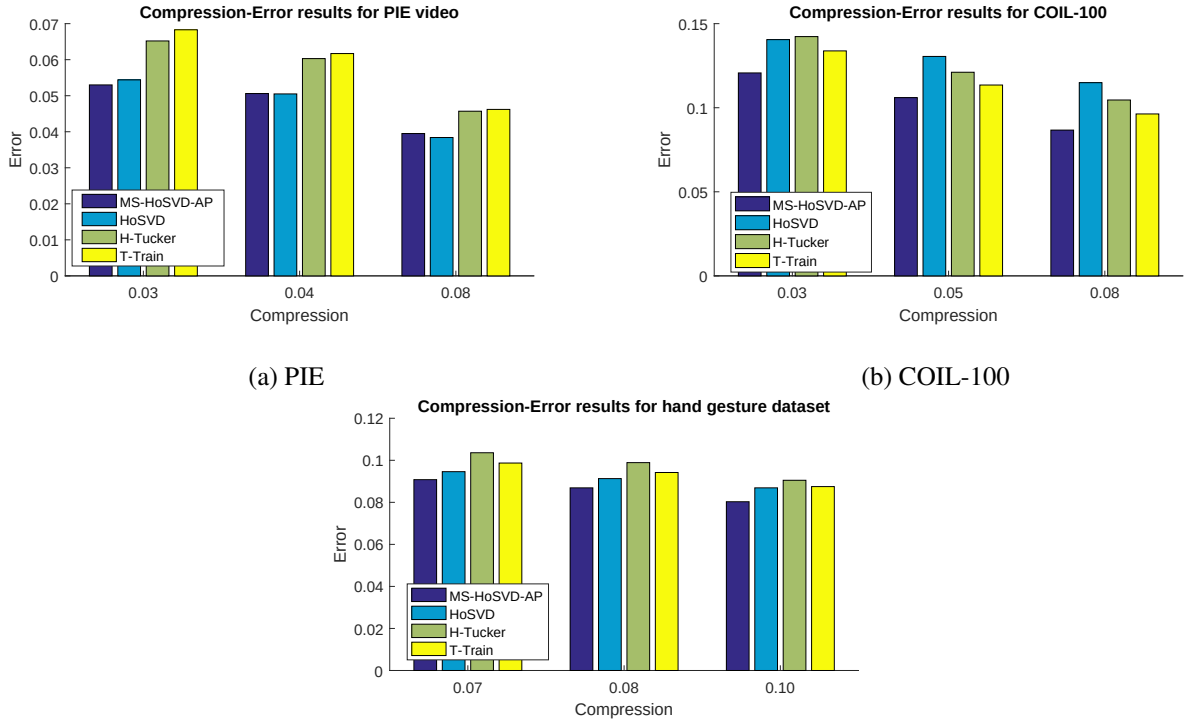


Figure 4.6: Reconstruction error and compression rate computed for pruned tree structure obtained by applying MS-HoSVD with 2-scales to PIE dataset. Top-left and right image is the sample frame by reconstructing the tensor using only 0th scale and recontruction by using 0th and 1st scales respectively. Bottom-left image is a sample frame for reconstructed using 2-scale approximation with all the subtensors and the bottom-right image is the reconstruction of 2 scale analysis with pruning approach where $\lambda = 0.25$.

Performance of the pruning algorithm reported in Tables 4.1, 4.2 and 4.3 is also compared with HoSVD, H-Tucker and T-Train decompositions in Fig. 4.7. As seen in Fig. 4.7 (b) and (c), MS-HoSVD outperforms other approaches for compressing COIL-100 and Hand Gesture datasets at varying compression rates. However, for PIE data, the performance of MS-HoSVD and HoSVD are very close to each other while both approaches outperform H-Tucker and T-Train, as can be seen in Fig. 4.7 (a). In Fig. 4.8, sample frames of PIE data reconstructed by T-Train (top-left), H-Tucker (top-right), HoSVD (bottom-left) and pruned MS-HoSVD with 2-scales (bottom-right) are shown. It can be easily seen that the reconstructed images by H-Tucker and T-Train are more blurred than the ones obtained by HoSVD and MS-HoSVD. One can also see the facial details captured by MS-HoSVD are clearer than HoSVD although the performances of both algorithms

are very similar to each other. The reason for capturing facial details better by MS-HoSVD is that the higher scale subtensors encode facial details.



(a) PIE

(b) COIL-100

(c) Hand Gesture

Figure 4.7: Compression rate versus Normalized Reconstruction Error for MS-HoSVD with adaptive pruning (dark blue), HoSVD (light blue), H-Tucker (green) and T-Train (yellow) for a) PIE, b) COIL-100 and c) Hand Gesture datasets. 2-scale MS-HoSVD tensor approximations are obtained using $\tau = 0.7$ for each scale and varying pruning trade-off parameter λ.

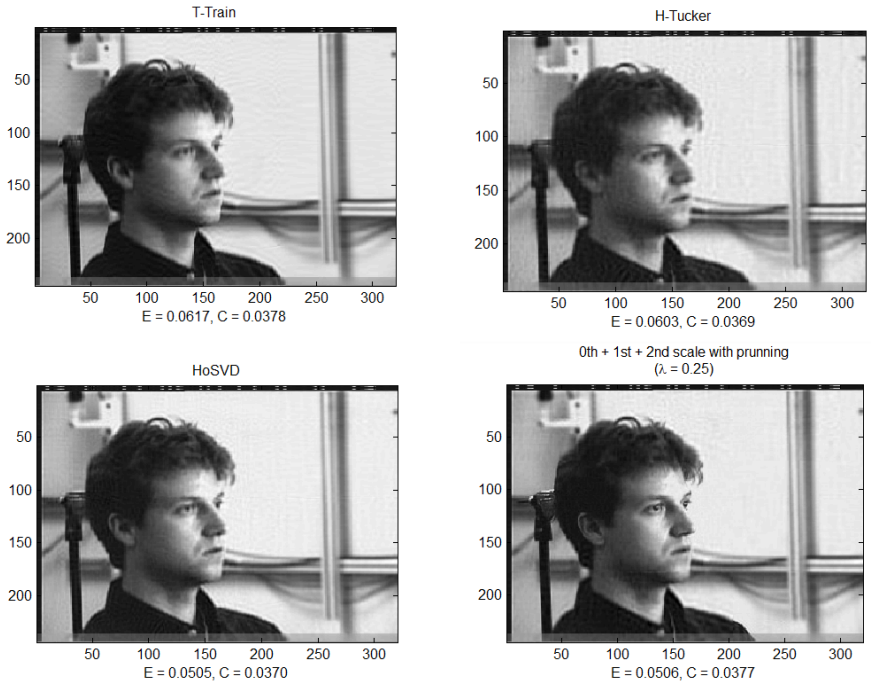


Figure 4.8: Reconstructed frame samples from PIE data compressed by T-Train (top-left), H-Tucker (top-right), HoSVD (bottom-left) and pruned MS-HoSVD with 2-scales (bottom-right). 2-scale MS-HoSVD tensor approximation is obtained using $\tau = 0.7$ for each scale and $\lambda = 0.25$.

4.5 Feature Extraction and Classification

In this section, we evaluate the features extracted from MS-HoSVD for classification of 2-mode and 3-mode tensors containing object images and hand gesture videos. Discrimination power and classification accuracy of MS-HoSVD features are compared to the features extracted by HoSVD and T-Train.

4.5.1 COIL-100 Image Dataset

For computational efficiency, each image was downsampled to a gray-scale image of 32×32 pixels. Number of images per object used for training data was gradually increased from 18 to 54 and selected randomly. A 3-mode tensor $\mathcal{X}^{tr} \in \mathbb{R}^{32 \times 32 \times I_{tr}}$ is constructed from training images

where $I_{tr} \in 100 \times \{18,\ 36,\ 54\}$ and the rest of the images are used to create the testing tensor $\mathcal{X}^{te} \in \mathbb{R}^{32\times32\times I_{te}}$ where $I_{te} = 7200 - I_{tr}$.

4.5.2 The Cambridge Hand Gesture Dataset

For computational efficiency, each image was downsampled to a gray-scale image of 30×40 pixels. Number of image sequences used for training data gradually increased from 25 to 75 per gesture and selected randomly. A 4-mode tensor $\mathcal{X}^{tr} \in \mathbb{R}^{30\times40\times30\times I_{tr}}$ is constructed from training image sequences where $I_{tr} \in 9 \times \{25,\ 50,\ 75\}$ and the rest of the image sequences are used to create the testing tensor $\mathcal{X}^{te} \in \mathbb{R}^{30\times40\times30\times I_{te}}$ where $I_{te} = 900 - I_{tr}$.

4.5.3 Classification Experiments

4.5.3.1 Training

For MS-HoSVD, the training tensor $\mathcal{X}^{tr}$ is decomposed using 1-scale MS-HoSVD as follows. Tensor partitioning is performed by LSA and the cluster number along each mode is chosen as $c = \{2,\ 3,\ 1\}$ yielding 6 subtensors for COIL-100 dataset and $c = \{2,\ 2,\ 3,\ 1\}$ yielding 12 subtensors for hand gesture dataset. We did not partition the tensor along the last mode that corresponds to the classes to make the comparison with other methods fair. The rank used in 0th scale is selected adaptively depending on the energy criterion with $\tau = 0.7$, while full rank decomposition is used for the 1st scale. Decomposing training data $\mathcal{X}^{tr}$ by MS-HoSVD as

$$\mathcal{X}^{tr}_{j,k} = \mathcal{C}^{tr}_{j,k} \times_1 \hat{\mathbf{U}}^{tr,(1)}_{j,k} \times_2 \hat{\mathbf{U}}^{tr,(2)}_{j,k} \ldots \times_N \hat{\mathbf{U}}^{tr,(N)}_{j,k}, \tag{4.31}$$

provides core tensors $\mathcal{C}_{j,k}^{tr}$ and factor matrices $\mathbf{U}_{j,k}^{tr,(i)}$ for jth scale and kth subtensor. Feature tensor $\mathcal{S}_{j,k}^{tr}$ for the training data is then created by projecting $\mathcal{X}_{j,k}^{tr}$s onto the first $N-1$ factor matrices $\mathbf{U}_{j,k}^{tr,(i)}$ as:

$$\mathcal{S}_{j,k}^{tr} = \mathcal{X}_{j,k}^{tr} \times_1 \left(\hat{\mathbf{U}}_{j,k}^{tr,(1)}\right)^\top \times_2 \left(\hat{\mathbf{U}}_{j,k}^{tr,(2)}\right)^\top ... \times_{N-1} \left(\hat{\mathbf{U}}_{j,k}^{tr,(N-1)}\right)^\top. \tag{4.32}$$

Unfolding the feature tensors $\mathcal{S}_{j,k}^{tr}$ along the sample mode N and concatenating them to each other yields a high dimensional feature vectors for the training samples. From these vectors, N_f features with the highest Fisher Score [251] are selected to form the lower-dimensional feature vectors $\mathbf{x}^{tr} \in \mathbb{R}^{N_f \times 1}$ for each training sample where the number of features (N_f) is determined by maximizing the discrimination score. For HoSVD and T-Train, full rank decompositions are computed and feature vectors are created by selecting N_f features with the highest Fisher Score from the core tensors as described above.

4.5.3.2 Testing

To create feature vectors from the testing samples, first, the testing tensor $\mathcal{X}_{te}$ is projected onto $\mathbf{U}_{j,k}^{tr,(i)}$ where $i \in [N-1]$ as:

$$\mathcal{S}_{j,k}^{te} = \mathcal{X}_{j,k}^{te} \times_1 \left(\hat{\mathbf{U}}_{j,k}^{tr,(1)}\right)^\top \times_2 \left(\hat{\mathbf{U}}_{j,k}^{tr,(2)}\right)^\top ... \times_{N-1} \left(\hat{\mathbf{U}}_{j,k}^{tr,(N-1)}\right)^\top. \tag{4.33}$$

Similar to the training step, unfolding the feature tensors $\mathcal{S}_{j,k}^{te}$ along the sample mode N and concatenating them with each other yields high dimensional feature vectors for the testing samples. The features corresponding to the features selected from the training step are used to form the feature vectors for testing samples $\mathbf{x}^{te} \in \mathbb{R}^{N_f \times 1}$. Once the feature vectors are obtained, 1-NN

is used to classify the test samples using the Euclidean distance. A similar two-step procedure is used to create feature vectors for HoSVD and T-Train for testing data. First, the testing tensor is projected onto factor matrices obtained from training data, then, N_f number of features are selected to create feature vectors.

As seen in Tables 4.4 and 4.5, features obtained from MS-HoSVD have greater Fisher score on average and classify the images better than both HoSVD and T-Train. It is also seen that the performance of HoSVD, T-Train and MS-HoSVD become close to each other as the size of the training dataset increases as expected. The reason behind the improved performance of MS-HoSVD is that MS-HoSVD captures the variations and nonlinearities across the modes such as rotation or translation better than the other methods. In both of the datasets used in this section, the images are rotated across the different frames. Since these nonlinearities are encoded in the higher scale (1st scale) features while the average characteristics , which are the same as HoSVD, are captured by the lower scale (0th scale) MS-HoSVD features the classification performance of the MS-HoSVD is slightly better than HoSVD. However, MS-HoSVD features, have much higher Fisher score than HoSVD features on average which indicates that classification using 1-NN with Euclidean distance may not be able to capture this difference in discrimination power since it treats all of the features obtained from 0th and 1st scales equally. It is also seen that T-Train features are not as good as MS-HoSVD and HoSVD features for capturing rotations and translations in the data and requires larger training set to reach the performance of MS-HoSVD and HoSVD.

Table 4.4: 1NN classification results for COIL-100 dataset over 20 trials with $N_f = 100$.

Training Size	Method	Accuracy mean $\pm$ std.	FS (10^3) mean $\pm$ std.
25%	MS-HoSVD	**93.71 $\pm$ 1.28**	1.11 $\pm$ 0.38
	HoSVD	93.07 $\pm$ 1.33	0.25 $\pm$ 0.03
	T-Train	92.29 $\pm$ 2.23	0.26 $\pm$ 0.02
50%	MS-HoSVD	**97.41 $\pm$ 0.69**	1.01 $\pm$ 0.38
	HoSVD	97.10 $\pm$ 0.83	0.24 $\pm$ 0.02
	T-Train	96.99 $\pm$ 1.09	0.24 $\pm$ 0.02
75%	MS-HoSVD	**98.38 $\pm$ 0.65**	0.97 $\pm$ 0.34
	HoSVD	98.16 $\pm$ 0.72	0.26 $\pm$ 0.01
	T-Train	98.25 $\pm$ 0.41	0.25 $\pm$ 0.01

Table 4.5: 1NN classification results for hand gesture dataset over 20 trials with $N_f = 200$.

Training Size	Method	Accuracy mean $\pm$ std.	FS (10^3) mean $\pm$ std.
25%	MS-HoSVD	**75.40 $\pm$ 3.87**	6.37 $\pm$ 1.35
	HoSVD	75.01 $\pm$ 3.99	5.51 $\pm$ 0.50
	T-Train	69.20 $\pm$ 2.63	7.55 $\pm$ 0.32
50%	MS-HoSVD	**83.86 $\pm$ 3.12**	6.04 $\pm$ 1.68
	HoSVD	83.15 $\pm$ 2.90	4.01 $\pm$ 0.21
	T-Train	78.97 $\pm$ 2.25	5.64 $\pm$ 0.36
75%	MS-HoSVD	**87.47 $\pm$ 2.07**	5.14 $\pm$ 1.57
	HoSVD	86.93 $\pm$ 2.31	3.68 $\pm$ 0.17
	T-Train	85.64 $\pm$ 2.57	4.59 $\pm$ 0.21

4.6 Applications on fMRI

Advances in information technology are making it possible to collect increasingly massive amounts of multidimensional, multi-modal neuroimaging data such as functional magnetic resonance imaging (fMRI). Current fMRI datasets involve multiple variables including multiple subjects, as well as both temporal and spatial data. These high dimensional datasets pose a challenge to the signal processing community to develop data reduction methods that can exploit their rich structure and extract meaningful summarizations. In this section, we demonstrate the performance of MS-HoSVD for compressing 4-way tensor containing fMRI volume sequences compared with HoSVD and 4D-Wavelet transform.

4.6.1 Data Description and Preprocessing

The data used in this section is obtained from 1000 Functional Connectomes Project [252] which has aggregated previously collected test-retest imaging datasets from more than 36 labs around the world. The data acquired from above url is referred to as Bangor which contains open-eye resting state fMRI scans of 20 male participants aged between 19-38 (Magnet: 3T, TR = 2, 34 slices, 265 time points).

The data were pre-processed using CONN functional connectivity toolbox [253]. First, structural images were co-registered to the mean functional image for each subject and normalized to MNI space. Then, slice timing correction and motion correction were performed for each functional images. The functional images were warped to Talairach Daemon atlas [254] provided by CONN toolbox and smoothed with an 4-mm FWHM Gaussian kernel. Confounds such as motion parameters obtained from reallignment and bold signals obtained from white matter and CSF masks were regressed out and band-pass (0.008-0.09 Hz) temporal filtering was applied to func-

tional images of each subject. After pre-processing, the fMRI dataset can be represented as a 4-mode tensor $\mathcal{X}^m \in \mathbb{R}^{109\times91\times91\times265}$ for each subject $m \in \{1,\ 2,\ ...,\ 20\}$ where the first three modes correspond to the preprocessed volume data and the fourth mode to time.

4.6.2 Results

In this section, we evaluated the performance of the MS-HoSVD in comparison to HoSVD and 4-D Wavelet for both compression and error rate on fMRI data. $\mathcal{X}^m$s obtained from preprocessing are decomposed by using MS-HoSVD yielding $\hat{\mathcal{X}}^m_{MS}$, HoSVD $\hat{\mathcal{X}}^m_{HO}$ and wavelet $\hat{\mathcal{X}}^m_{W}$, respectively. In the following experiments, clustering is performed by by local subspace analysis (LSA) [244] and the number of clusters along each mode is chosen as $c_i = 4$. The rank used in truncated HoSVD is selected adaptively depending on the energy criterion. Energy criterion determines the minimum number of singular values kept during the SVD of the unfolded tensors along each mode such that the cumulative energy is above a certain threshold. For MS-HoSVD, the energy thresholds are selected as 0.7 and 0.95 for the SVDs computed for 0th and 1st scales, respectively. For HoSVD, the energy threshold increased gradually from 0.990 to 0.999 with a step size of 0.0005 to compare the reconstruction error at similar compression ratios (experiment-1) and the compression rate for the similar error rates (experiment-2). For the 4-D Wavelet compression, 2-scale 1-D temporal Wavelet transform followed by a 2-scale 3-D spatial Wavelet transform with Db3 wavelet functions were applied. Significant wavelet coefficients were selected to have similar compression ratio close to MS-HoSVD. In Table 4.6 the error rate refers to the normalized tensor approximation error $\frac{\|\mathcal{X}-\hat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F}$ and the compression ratio is computed as $\frac{\#\text{ total bits to store }\mathcal{X}}{\#\text{ total bits to store }\hat{\mathcal{X}}}$. As seen in Table, 4.6, MS-HoSVD provides reduced error (experiment-1) and better compression than HoSVD (experiment-2) for 20 subjects. MS-HoSVD also provides smaller reconstruction error than 4-D Wavelet (Table 4.6).

Table 4.6: Average compression ratio (mean±st.dev) and reconstruction error (mean±st.dev) obtained by MS-HoSVD, HoSVD and 4-D Wavelet over 20 subjects.

	MS-HoSVD	HoSVD (Exp-1)	HoSVD (Exp-2)	4D-Wavelet
Compression Ratio	10.3275 ±0.6287	10.3736 ±0.5955	8.3068 ±0.6427	10.3456 ±0.1605
Reconstruction Error	0.0231 ±0.0018	0.0404 ±0.0061	0.0228 ±0.0040	0.0493 ±0.0026

Once low-rank approximations are obtained, mean ROI signals $\mathbf{Y}^m \in \mathbb{R}^{88\times265}$, $\mathbf{Y}^m_{MS} \in \mathbb{R}^{88\times265}$, $\mathbf{Y}^m_{HO} \in \mathbb{R}^{88\times265}$ and $\mathbf{Y}^m_{W} \in \mathbb{R}^{88\times265}$ corresponding to $\mathcal{X}^m$, $\hat{\mathcal{X}}^m_{MS}$, $\hat{\mathcal{X}}^m_{HO}$ and $\hat{\mathcal{X}}^m_{W}$ are computed for each subject using Talairach Daemon atlas. Connectivity networks for each subject m are denoted as $\mathbf{A}^m \in \mathbb{R}^{88\times88}$, $\mathbf{A}^m_{MS} \in \mathbb{R}^{88\times88}$, $\mathbf{A}^m_{HO} \in \mathbb{R}^{88\times88}$, $\mathbf{A}^m_{W} \in \mathbb{R}^{88\times88}$, and are constructed by computing the correlation coefficient between all ROIs in $\mathbf{Y}^m$, $\mathbf{Y}^m_{MS}$, $\mathbf{Y}^m_{HO}$ and $\mathbf{Y}^m_{W}$, respectively. Significant connections ($p \leqslant 0.01$, Bonferroni corrected) for each method were determined by performing t-tests for each edge of connectivity matrices over subjects. Table 4.7 shows the miss and false alarm rates for the connectivity networks constructed using MS-HoSVD, HoSVD and 4-D Wavelet in comparison to the original network. As it can be seen from Table 2, we obtain lower error rates for MS-HoSVD compared to HoSVD and Wavelet for all of the experiments.

Table 4.7: Comparisons of probability of miss (P_{Miss}) and probability of false alarm (P_{FA}) obtained by MS-HoSVD, HoSVD and 4-D Wavelet.

	MS-HoSVD	HoSVD (Exp-1)	HoSVD (Exp-2)	4D-Wavelet
P_{Miss}	0.0020	0.0033	0.0031	0.0023
P_{FA}	0	0.0006	0	0

4.7 Conclusions

In this chapter, we proposed a new multi-scale tensor decomposition technique for better approximating the local nonlinearities in generic tensor data. The proposed approach constructs a tree structure by considering similarities along different fibers of the tensor and decomposes the tensor into lower dimensional subtensors hierarchically. A low-rank approximation of each subtensor is then obtained by HoSVD. We also introduced a pruning strategy to find the optimum tree structure by keeping the important nodes and eliminating redundancy in the data. The proposed approach is applied to a set of 3-way and 4-way tensors to evaluate its performance on both data reduction and classification applications.

Future work will consider automatic selection of parameters such as the number of clusters and the appropriate rank along each mode.The computational efficiency of the proposed method can be improved through parallelization of the algorithm such as parallel construction of subtensors and parallel implementation of HoSVD [255]. This efficient implementation will enable the implementation of finer scale decompositions for higher order and higher dimensional tensors. Proposed algorithm currently constructs the tree structure based on decomposing the tensor using HoSVD. The proposed tensor decomposition structure can also be implemented using other tensor decomposition methods such as PARAFAC and tensor-train decompositions.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis makes fundamental contributions to data reduction of tensor type data with a particular focus on providing a better understanding of dynamic functional connectivity brain networks. In Chapter 2, we discuss the problem of tensor data reduction through clustering where the tensor contains functional connectivity brain networks collected in time across multiple subjects, and we approach the problem as a multi-graph clustering problem. To determine the common community structure underlying the functional connectivity networks of all subjects, we propose a hierarchical consensus spectral clustering approach, FCCA. To obtain a hierarchical decomposition, Fiedler vector based graph bi-partitioning method is applied first to the data and later to its partitionins repeatedly. Multiple subjects are taken into account by first obtaining an initial bipartition of each subject's connectivity network and then by iteratively partitioning the co-occurrence matrix across subjects. Furthermore, new information-theoretic cluster quality measures are introduced for selecting the optimal community structure as an alternative to the standard modularity measure. This measure depends on optimizing the tradeoff between maximizing the divergence between clusters and minimizing the entropy of individual clusters. It is also shown that FCCA is computationally more efficient than standard consensus clustering approaches such as voting and is more accurate

than averaging in the case of outliers and overlapping community structures. Moreover, the application of the proposed algorithm to functional connectivity networks constructed from EEG data during a study of ERN produced clusters consistent with published work [178,179] whereas voting and averaging methods failed to identify meaningful lateral and medial frontal communities.

In Chapter 3, we proposed an alternative way to reduce this high dimensional data through linear subspace estimation and update methods. In this approach, the dynamics of the connectivity networks are taken into account such that the data reduction is done across subjects for time intervals determined by the subspace tracking approach. For this purpose, we introduced a new recursive low-rank + sparse structure learning algorithm for tensor type data in order to track the modular structure of functional connectivity networks through EEG recordings across multiple subjects. This approach yields robust estimation of the low-rank component of a dynamic 3-way tensor at each time point and provides a recursive way to update the subspace information. The proposed approach, Ho-RLSL, identifies the time points where the low-rank subspaces change and recursively updates these subspaces to improve the low-rank approximation of data. The proposed approach is used to separate low-rank and sparse parts of dFCNs across multiple subjects. Low-rank component of dFCNs obtained for the detected time intervals are summarized using FCCA. The low-rank subspace approximation to each time interval provides a denoised version of the original network, thus improving the quality of the clustering results. The results indicate a clear change in the community structure before and after the physiological response. Moreover, the detected community structures are in line with previous hypothesis regarding the networks involved in cognitive control [221].

In Chapter 4, we proposed a new tensor decomposition technique which better approximates the underlying nonlinear structure of tensors compared to HoSVD. The proposed method constructs data-dependent multiscale dictionaries to better represent the data. The multiscale structure

of the proposed decomposition is obtained by constructing a tree structure depending on the local similarities in the tensor. Based on these local similarities, the proposed method decomposes the tensor into lower dimensionsional subtensors hierarchically. Low-rank structure of each subtensor obtained by HoSVD provides finer linear approximation for the points in the region of interest. Moreover, a pruning strategy which keeps important nodes in the tree is introduced to find sub-optimal tree structure. The proposed approach is applied to 3-way and 4-way tensors containing simulated and real datasets to illustrate the improvement in the compression performance compared to HoSVD and Wavelet transform. In addition, it is shown that features obtained from multiscale representation provide better classification performance than using HoSVD features for tensors containing nonlinearities such as rotation or translation.

5.2 Future Work

There are still remaining challenges involving data reduction of high-dimensional and higher-order tensors. Some of these challenges and possible solutions include:

- Unsupervised model selection: Tensor based approaches have been widely used in a variety of fields e.g. computer vision, data science and biomedical imaging and, identifying the most appropriate model such as Tucker, PARAFAC or Tensor-Train to exploit the multilinear structure of the data is one of the most important challenges. Bro et al. [256] has already addressed this issue and proposed an approach known as Core consistency diagnostics (CORCONDIA) to select Tucker versus CP model. Core consistency metric provides a larger value if the Tucker core is close to super diagonal tensor indicating that the CP model is more appropriate. However, this metric is limited to Tucker and CP models. Therefore, there is a need to develop new metrics appropriate for other models including Tensor-Train

and Hierarchical Tucker. Using such a metric may also enable us to select appropriate model to use in proposed multiscale tensor decomposition structure and may improve compression performance.

- Robust rank selection: Finding the appropriate rank for the selected model is another crucial problem for tensor based approaches and robust rank selection algorithms improve the compression performance of the selected models directly. Several different approaches have been already proposed to identify rank for both CP and Tucker decomposition [257–264]. For CP rank, Papalexakis et al. [259] proposed fast and exact algorithm for CORCONDIA which scales with tensor size while [262] and [260] used Bayesian approaches. To identify Tucker rank, Timmerman et al. [263] introduced difference in fit (DIFFIT) procedure which finds the combination of dimensions that gives the best-fit by assigning different ranks to each mode. Similar to CP, Bayesian approaches have been also used to identify Tucker rank [262]. Adapting these methods to Ho-RLSL to estimate low-rank subspace dimension in a robust way may enhance the denoising performance of the proposed algorithm. Moreover, robust rank selection may also improve the compression performance of MS-HoSVD.

- Efficient algorithms: In order to overcome the curse of dimensionality emerging from higher-order data, developing scalable tools for tensor decomposition has become inevitable. Recently, many approaches have been developed to compute Tucker and CP decompositions efficiently [255, 265–269]. Tsourakakis [265] showed that an accurate low-rank Tucker approximation of the tensor can be computed much faster from a sparsified version of the tensor. Similarly, Papalexakis et al. [267] proposed a fast and parallelizable method using random sampling techniques for speeding up CP decomposition which produces sparse outer product approximations for sparse tensors. Scalable and distributed implementations

of tensor mode-N product and singular value decomposition, which are essential to both CP and Tucker decompositions have also been developed [246, 266]. More recently, Austin et al. [269] proposed the first distributed memory implementation of a parallel algorithm for computing a Tucker decomposition. Adapting these techniques to the implementation of proposed Ho-RLSL and MS-HoSVD will surely reduce the time complexity.

APPENDIX

Proofs for Chapter 3

β_t Is Small

In this section, we will prove that β_t is small and that eqn (3.14) can be treated as a sparse recovery in noise problem. Define the subspace estimation error for ith mode as

$$\text{SE}(\mathbf{P}^{(i)}, \hat{\mathbf{P}}^{(i)}) := \| \ (I - \hat{\mathbf{P}}^{(i)}\hat{\mathbf{P}}^{\top,(i)})\mathbf{P}^{(i)} \ \|_F = \epsilon_i, \tag{1}$$

where $\mathbf{P}^{(i)}$ and $\hat{\mathbf{P}}^{(i)}$ are true and estimated basis matrices of the ith mode, respectively. $\mathbf{P}_j^{(i)} = \left[\mathbf{P}_{j-1}^{(i)}\mathbf{P}_{j,new}^{(i)}\right]$ where $\mathbf{P}_{j,new}^{(i)}$ is a $n_i \times c_{j,new}^{(i)}$ basis matrix with $(\mathbf{P}_{j,new}^{(i)})^\top \mathbf{P}_{j-1}^{(i)} = 0$. Since the low-rank tensor at time t, $\mathcal{L}_t$, has components $\mathcal{A}_m$s with $m \in \{1, 2, ..., 8\}$ which are the projections of $\mathcal{L}_t$ in both the previous subspace $\mathbf{P}_{j-1}^{(i)}$s and $\mathbf{P}_{j,new}^{(i)}$s. Therefore, $\mathcal{L}_t$ can be written as the sum of these components:

$$\begin{aligned}
\mathcal{L}_t = \ & \mathcal{A}_1 \times_1 \mathbf{P}_{j-1}^{(1)} \times_2 \mathbf{P}_{j-1}^{(2)} \times_3 \mathbf{P}_{j-1}^{(3)} \\
& + \mathcal{A}_2 \times_1 \mathbf{P}_{j,new}^{(1)} \times_2 \mathbf{P}_{j-1}^{(2)} \times_3 \mathbf{P}_{j-1}^{(3)} \\
& + \mathcal{A}_3 \times_1 \mathbf{P}_{j-1}^{(1)} \times_2 \mathbf{P}_{j,new}^{(2)} \times_3 \mathbf{P}_{j-1}^{(3)} \\
& + \mathcal{A}_4 \times_1 \mathbf{P}_{j-1}^{(1)} \times_2 \mathbf{P}_{j-1}^{(2)} \times_3 \mathbf{P}_{j,new}^{(3)} \\
& + \mathcal{A}_5 \times_1 \mathbf{P}_{j,new}^{(1)} \times_2 \mathbf{P}_{j,new}^{(2)} \times_3 \mathbf{P}_{j-1}^{(3)} \\
& + \mathcal{A}_6 \times_1 \mathbf{P}_{j,new}^{(1)} \times_2 \mathbf{P}_{j-1}^{(2)} \times_3 \mathbf{P}_{j,new}^{(3)} \\
& + \mathcal{A}_7 \times_1 \mathbf{P}_{j-1}^{(1)} \times_2 \mathbf{P}_{j,new}^{(2)} \times_3 \mathbf{P}_{j,new}^{(3)} \\
& + \mathcal{A}_8 \times_1 \mathbf{P}_{j,new}^{(1)} \times_2 \mathbf{P}_{j,new}^{(2)} \times_3 \mathbf{P}_{j,new}^{(3)}
\end{aligned} \tag{2}$$

where we redefine the old and new parts of the projection as

$$
\begin{aligned}
\mathcal{A}_{t,*} = \mathcal{A}_1 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j-1}^{\top,(1)} \times_2 \mathbf{P}_{j-1}^{\top,(2)} \times_3 \mathbf{P}_{j-1}^{\top,(3)} \\
\mathcal{A}_2 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j,new}^{\top,(1)} \times_2 \mathbf{P}_{j-1}^{\top,(2)} \times_3 \mathbf{P}_{j-1}^{\top,(3)} \\
\mathcal{A}_3 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j-1}^{\top,(1)} \times_2 \mathbf{P}_{j,new}^{\top,(2)} \times_3 \mathbf{P}_{j-1}^{\top,(3)} \\
\mathcal{A}_4 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j-1}^{\top,(1)} \times_2 \mathbf{P}_{j-1}^{\top,(2)} \times_3 \mathbf{P}_{j,new}^{\top,(3)} \\
\mathcal{A}_5 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j,new}^{\top,(1)} \times_2 \mathbf{P}_{j,new}^{\top,(2)} \times_3 \mathbf{P}_{j-1}^{\top,(3)} \\
\mathcal{A}_6 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j,new}^{\top,(1)} \times_2 \mathbf{P}_{j-1}^{\top,(2)} \times_3 \mathbf{P}_{j,new}^{\top,(3)} \\
\mathcal{A}_7 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j-1}^{\top,(1)} \times_2 \mathbf{P}_{j,new}^{\top,(2)} \times_3 \mathbf{P}_{j,new}^{\top,(3)} \\
\mathcal{A}_{t,new} = \mathcal{A}_8 = \ & \mathcal{L}_t \times_1 \mathbf{P}_{j,new}^{\top,(1)} \times_2 \mathbf{P}_{j,new}^{\top,(2)} \times_3 \mathbf{P}_{j,new}^{\top,(3)}.
\end{aligned}
\tag{3}
$$

Assumptions:

1. Assume that subspace estimation error is $\epsilon_i = \| \ (I - \hat{\mathbf{P}}^{(i)}\hat{\mathbf{P}}^{\top,(i)})\mathbf{P}^{(i)} \ \|_F \leqslant r_0^{(i)}\zeta$ for $\zeta \ll 1$.

2. Let $\mathbf{l}_i^{(k)}$ be the ith column of $\mathbf{L}_{t,(k)}$ and assume that $\| \ \mathbf{l}_i^{(k)} \ \|_F \leqslant \gamma_{*,k}$, $\gamma_{*,k} \leqslant \frac{1}{\sqrt{\zeta r_J^{(k)}}}$ and $\gamma_{new,k} << \gamma_{*,k}$.

3. Define $\gamma_* = min_{k \in \{1,\ 2,\ 3\}}(\gamma_{*,k}\sqrt{\prod_{i=1, i\neq k}^{3} N_i})$ and assume that $\| \ \mathcal{L}_t \ \|_F \leqslant \gamma_*$.

4. Define $\gamma_{new} = min_{k \in \{1,\ 2,\ 3\}}(\gamma_{new,k}\sqrt{\prod_{i=1, i\neq k}^{3} N_i})$ and assume that $\gamma_{new} \ll \gamma_*$.

β_t is defined as $\beta_t = \mathcal{L}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)}$ where $\boldsymbol{\phi}_t^{(i)} = \mathbf{I} - \hat{\mathbf{P}}_{t-1}^{(i)}(\hat{\mathbf{P}}_{t-1}^{(i)})^{\top}$ and its norm is:

$$
\begin{aligned}
\| \beta_t \|_F = & \| \mathcal{L}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \times_3 \boldsymbol{\phi}_t^{(3)} \|_F \\
\leqslant & \epsilon_1\epsilon_2\epsilon_3||\mathcal{A}_{t,*}||_F + \epsilon_2\epsilon_3||\mathcal{A}_2||_F \\
& +\epsilon_1\epsilon_3||\mathcal{A}_3||_F + \epsilon_1\epsilon_2||\mathcal{A}_4||_F \\
& +\epsilon_3||\mathcal{A}_5||_F + \epsilon_2||\mathcal{A}_6||_F \\
& +\epsilon_1||\mathcal{A}_7||_F + ||\mathcal{A}_{t,new}||_F
\end{aligned} \tag{4}
$$

From section 5.2:

$$
\begin{aligned}
\| \mathcal{A}_2 \|_F \leqslant & \gamma_* \sqrt{\text{rank}(\mathbf{P}_{j,new}^{(1)})\text{rank}(\mathbf{P}_{j-1}^{(2)})\text{rank}(\mathbf{P}_{j-1}^{(3)})} \\
\vdots & \\
\| \mathcal{A}_7 \|_F \leqslant & \gamma_* \sqrt{\text{rank}(\mathbf{P}_{j-1}^{(1)})\text{rank}(\mathbf{P}_{j,new}^{(2)})\text{rank}(\mathbf{P}_{j,new}^{(3)})}
\end{aligned} \tag{5}
$$

Let $\bar{N} = max_i(N_i)$, $\bar{r}_J = max_i(r_J^{(i)})$ and $\bar{\gamma_*} = max_k(\gamma_{*,k}) \cdot \bar{N}$ where $\bar{\gamma}_* > \gamma_*$. Define $\bar{\gamma}_{new} = max_k(\gamma_{new,k})$ and assume that $\bar{\gamma}_{new} \ll \bar{\gamma}_*$.

Note that;

- First term:

$$
\begin{aligned}
\epsilon_1\epsilon_2\epsilon_3||\mathcal{A}_{t,*}||_F & \leqslant (\bar{r}_{j-1}\zeta)^3 \cdot \bar{N} \cdot \bar{\gamma}_* \\
& \leqslant (\bar{r}_{j-1}\zeta)^3 \cdot \bar{N} \cdot \frac{1}{\sqrt{\zeta \bar{r}_J}} \\
& \leqslant \bar{N} \cdot \zeta^{5/2} \cdot (\bar{r}_{j-1})^2.
\end{aligned} \tag{6}
$$

- Last term:

$$
||\mathcal{A}_{t,new}||_F \leqslant \bar{N} \cdot \bar{\gamma}_{new,1}. \tag{7}
$$

- Cross terms:

$$\begin{aligned}\epsilon_2\epsilon_3||\mathcal{A}_2||_F &\leqslant (\bar{r}_{j-1}\zeta)^2 \cdot \bar{\gamma}_*\sqrt{\bar{r}_{j,new}\bar{r}_{j-1}\bar{r}_{j-1}} \\ &\leqslant \bar{N} \cdot (\bar{r}_{j-1}\zeta)^2 \cdot \frac{1}{\sqrt{\zeta\bar{r}_J}}\bar{r}_{j-1}\sqrt{\bar{r}_{j,new}} \\ &\leqslant \bar{N} \cdot \zeta^{3/2} \cdot (\bar{r}_{j-1})^2 \cdot (\bar{r}_{j,new})^{1/2}.\end{aligned} \tag{8}$$

$$\begin{aligned}\epsilon_3||\mathcal{A}_5||_F &\leqslant (\bar{r}_{j-1}\zeta) \cdot \bar{\gamma}_*\sqrt{\bar{r}_{j,new}\bar{r}_{j,new}\bar{r}_{j-1}} \\ &\leqslant \bar{N} \cdot (\bar{r}_{j-1}\zeta) \cdot \frac{1}{\sqrt{\zeta\bar{r}_J}}\bar{r}_{j,new}\sqrt{\bar{r}_{j-1}} \\ &\leqslant \bar{N} \cdot \zeta^{1/2} \cdot (\bar{r}_{j-1})^{1/2} \cdot \bar{r}_{j,new}.\end{aligned} \tag{9}$$

Therefore;

$$\begin{aligned}\| \beta_t \|_F \leqslant\ & \bar{N} \cdot \zeta^{5/2} \cdot (\bar{r}_{j-1})^2 + 3 \cdot \bar{N} \cdot \zeta^{3/2} \cdot (\bar{r}_{j-1})^2 \cdot (\bar{r}_{j,new})^{1/2} \\ & +3 \cdot \bar{N} \cdot \zeta^{1/2} \cdot (\bar{r}_{j-1})^{1/2} \cdot \bar{r}_{j,new} + \bar{N} \cdot \bar{\gamma}_{new}. \\ \| \beta_t \|_F \leqslant\ & \bar{N} \cdot \zeta^{1/2} \cdot \bar{\gamma}_*^{(-4)} + 3 \cdot \bar{N} \cdot \zeta^{1/2} \cdot (\bar{r}_{j-1})^1 \cdot \bar{\gamma}_*^{(-2)} \cdot (\bar{r}_{j,new})^{1/2} \\ & +3 \cdot \bar{N} \cdot \bar{\gamma}_*^{(-1)} \cdot \bar{r}_{j,new} + \bar{N} \cdot \bar{\gamma}_{new}.\end{aligned} \tag{10}$$

Since ζ is small and $\bar{\gamma}_*$ is large, the last term is dominant in the upper bound. Thus, β_t can be considered as a noise by the slow subspace change assumption $\| \gamma_{new} \|_F \ll \| \mathcal{S}_t \|_F$.

Upper bound for Frobenious Norm of Cross Terms

Upper bound for the norm of the cross terms is derived as follows:

$$\begin{aligned}
\| \mathcal{A}_{t,2} \|_F = & \| \mathcal{L}_t \times_1 \mathbf{P}_{j,new}^{\top,(1)} \times_2 \mathbf{P}_{j-1}^{\top,(2)} \times_3 \mathbf{P}_{j-1}^{\top,(3)} \|_F \\
\leqslant & \| \mathbf{P}_{j,new}^{\top,(1)} \mathbf{L}_{t,(1)} \|_F \cdot \| \mathbf{P}_{j-1}^{(3)} \otimes \mathbf{P}_{j-1}^{(2)} \|_F \\
\leqslant & \sqrt{r_{j,new}^{(1)}} \cdot \gamma_* \cdot \sqrt{\text{tr}\left[(\mathbf{P}_{j-1}^{\top,(3)} \otimes \mathbf{P}_{j-1}^{\top,(2)})(\mathbf{P}_{j-1}^{(3)} \otimes \mathbf{P}_{j-1}^{(2)}) \right]} \\
= & \sqrt{r_{j,new}^{(1)}} \cdot \gamma_* \cdot \sqrt{\text{tr}\left[(\mathbf{P}_{j-1}^{\top,(3)} \mathbf{P}_{j-1}^{(3)})^\top \otimes (\mathbf{P}_{j-1}^{\top,(2)} \mathbf{P}_{j-1}^{(2)}) \right]} \\
= & \sqrt{r_{j,new}^{(1)}} \cdot \gamma_* \cdot \sqrt{\text{tr}\left[(\mathbf{P}_{j-1}^{\top,(3)} \mathbf{P}_{j-1}^{(3)})^\top \right] \cdot \text{tr}\left[(\mathbf{P}_{j-1}^{\top,(2)} \mathbf{P}_{j-1}^{(2)}) \right]} \\
= & \gamma_* \sqrt{r_{j,new}^{(1)} r_{j-1}^{(2)} r_{j-1}^{(3)}}.
\end{aligned} \tag{11}$$

Proofs for Chapter 4

In order to facilitate error analysis for the 1-scale MS-HoSVD that is similar to the types of error analysis available for various HoSVD-based low-rank approximation strategies (see, e.g., [241]), we will engage in a more in depth discussion of condition (4.30) herein. Recall that the partition of $\mathcal{W}_0$ formed by the restriction matrices $\mathbf{R}_k^{(n)}$ in (4.20) – (4.22) is called *effective* if there exists another *pessimistic* partitioning of $\mathcal{W}_0$ via restriction matrices $\left\{ \tilde{\mathbf{R}}_k^{(n)} \right\}_{k=1}^K$ together with a bijection $f : [K] \to [K]$ such that

$$\sum_{n=1}^N \left\| \mathcal{X}|_k \times_n \left(\mathbf{I} - \mathbf{Q}_k^{(n)} \right) \right\|^2 \leqslant \sum_{n=1}^N \left\| \mathcal{W}_0 \times_n \tilde{\mathbf{R}}_{f(k)}^{(n)} \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)} \right) \bigtimes_{h \neq n}^N \tilde{\mathbf{R}}_{f(k)}^{(h)} \right\|^2 \tag{12}$$

holds for each $k \in [K]$. In (12) the $\left\{ \tilde{\mathbf{P}}^{(n)} \right\}$ are the orthogonal projection matrices obtained from the HoSVD of $\mathcal{W}_0$ with ranks $\tilde{r}_n \geqslant \bar{r}_n \geqslant r_n$ (i.e., where each $\tilde{\mathbf{P}}^{(n)}$ projects onto the top $\tilde{r}_n$ left singular vectors of the matricization $\mathbf{W}_{0,(n)}$). Below we will show that (12) holding for $\mathcal{W}_0$ implies that the error $\|\mathcal{W}_1\|$ resulting from our 1^{st}-scale approximation in (4.27) is less than

an upper bound of the type given for a high-rank standard HoSVD-based approximation (4.29) in [241].

Here we will begin with a lemma that shows our subtensor-based approximation of $\mathcal{W}_0$ is accurate whenever (12) is satisfied.

Lemma 3. *Let* $\mathcal{W}_0 = \mathcal{X} - \hat{\mathcal{X}}_0 \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$. *Suppose that* $\left\{\mathbf{R}_k^{(n)}\right\}$ *is a collection of effective restriction matrices that form an effective partition of* $\mathcal{W}_0$ *with respect to a pessimistic partition formed via pessimistic restriction matrices* $\left\{\tilde{\mathbf{R}}_k^{(n)}\right\}$ *as per* (12) *above. Similarly, let* $\tilde{\mathbf{P}}^{(n)}$ *be the rank* $\tilde{r}_n \geqslant \bar{r}_n \ \forall n$ *orthogonal projection matrices from* (12) *obtained via the truncated HoSVD of* $\mathcal{W}_0$ *as above. Then,*

$$\begin{aligned}\left\|\mathcal{W}_0 - \hat{\mathcal{X}}_1\right\|^2 &= \left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) - \textstyle\sum_{k=1}^{K}\left(\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_{n=1}^{N} \mathbf{Q}_k^{(n)}\right)\right\|^2 \\ &\leqslant \textstyle\sum_{n=1}^{N}\left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_n \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right)\right\|^2.\end{aligned}$$

Proof. We have that

$$\begin{aligned}\left\|\mathcal{W}_0 - \hat{\mathcal{X}}_1\right\|^2 &= \left\|\mathcal{W}_0 - \sum_{k=1}^{K} \mathcal{W}_0 \bigtimes_{n=1}^{N} \mathbf{Q}_k^{(n)}\right\|^2 && \text{Use (4.12) and (4.26)} \\ &= \left\|\sum_{k=1}^{K} \mathcal{W}_0 \bigtimes_{n=1}^{N} \mathbf{R}_k^{(n)} - \sum_{k=1}^{K} \mathcal{W}_0 \bigtimes_{n=1}^{N} \mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\right\|^2 && \text{Use (4.21), (4.22), and (4.25)} \\ &= \left\|\sum_{k=1}^{K} \mathcal{W}_0 \bigtimes_{n=1}^{N} \left(\mathbf{R}_k^{(n)} - \mathbf{Q}_k^{(n)}\mathbf{R}_k^{(n)}\right)\right\|^2 && \text{Use Lemma 1} \\ &= \sum_{k=1}^{K}\left\|\mathcal{X}|_k \bigtimes_{n=1}^{N} \left(\mathbf{I} - \mathbf{Q}_k^{(n)}\right)\right\|^2. && \text{Use Lemma 1, (4.21), and (4.25)}\end{aligned}$$

(13)

Applying lemmas 1 and 2 to (13) we can now see that

$$\left\|\mathcal{W}_0 - \hat{\mathcal{X}}_1\right\|^2 = \sum_{k=1}^{K}\sum_{n=1}^{N}\left\|\mathcal{X}|_k \bigtimes_{h=1}^{n-1} \mathbf{Q}_k^{(h)} \times_n \left(\mathbf{I} - \mathbf{Q}_k^{(n)}\right)\right\|^2$$
$$\leqslant \sum_{k=1}^{K}\sum_{n=1}^{N}\left\|\mathcal{X}|_k \times_n \left(\mathbf{I} - \mathbf{Q}_k^{(n)}\right)\right\|^2$$

since the $\mathbf{Q}_k^{(n)}$ matrices are orthogonal projections. Using assumption (12) we now get that

$$\left\|\mathcal{W}_0 - \hat{\mathcal{X}}_1\right\|^2 \leqslant \sum_{k=1}^{K}\sum_{n=1}^{N}\left\|\mathcal{W}_0 \times_n \tilde{\mathbf{R}}_k^{(n)}\left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right) \bigtimes_{h\neq n}^{N} \tilde{\mathbf{R}}_k^{(h)}\right\|^2$$
$$= \sum_{n=1}^{N}\left\|\mathcal{W}_0 \times_n \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right)\right\|^2$$

where we have used the fact that the pessimistic restriction matrices $\tilde{\mathbf{R}}_k^{(n)}$ partition $\mathcal{W}_0$ in the last line. □

Lemma 3 indicates that the error in approximating $\mathcal{W}_0$ via low-rank approximations of its effective subtensors is potentially smaller than the error obtained by approximating $\mathcal{W}_0$ via (higher rank) truncated HoSVDs whenever (12) holds.[1] The following theorem shows that this good error behavior extends to the entire 1^{st} scale approximation provided by (4.27) whenever (12) holds.

Theorem 2 (Restatement of Theorem 1)**.** *Let* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$. *Suppose that* (12) *holds. Then, the first scale approximation error given by MS-HoSVD* (4.27) *is bounded by*

$$\|\mathcal{W}_1\|^2 = \left\|\mathcal{X} - \hat{\mathcal{X}}_0 - \hat{\mathcal{X}}_1\right\|^2 \leqslant \sum_{n=1}^{N}\left\|\mathcal{X} \times_n \left(\mathbf{I} - \bar{\mathbf{P}}^{(n)}\right)\right\|^2$$

[1] That is, the upper bound on the error provided by Lemma 3 is less than or equal to the upper bound on the error for truncated HoSVDs provided by, e.g., [241] when/if (12) holds.

where $\left\{\bar{\mathbf{P}}^{(n)}\right\}$ *are low-rank projection matrices of rank* $\bar{r}_n \geqslant r_n$ *obtained from the truncated HoSVD of* $\mathcal{X}$ *as per* (4.29).

Proof. Using (4.12) and (4.17) together with lemma 3 we can see that

$$\begin{aligned}\|\mathcal{W}_1\|^2 = \left\|\mathcal{X} - \hat{\mathcal{X}}_0 - \hat{\mathcal{X}}_1\right\|^2 = \left\|\mathcal{W}_0 - \hat{\mathcal{X}}_1\right\|^2 &\leqslant \sum_{n=1}^{N} \left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_n \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right)\right\|^2 \\ &\leqslant \sum_{n=1}^{N} \left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\right\|^2 \end{aligned} \tag{14}$$

where $\bar{\mathbf{Q}}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ is the orthogonal projection matrix of rank $\tilde{r}_n$ which projects onto the subspace spanned by the top $\tilde{r}_n$ left singular vectors of $\mathbf{X}_{(n)}$. Here (14) holds because the orthogonal projection matrices $\tilde{\mathbf{P}}^{(n)}$ are chosen in (12) so that $\tilde{\mathbf{P}}^{(n)}\mathbf{W}_{0,(n)}$ is a best possible rank $\tilde{r}_n$ approximation to $\mathbf{W}_{0,(n)}$. As a result, we have that

$$\begin{aligned}\left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_n \left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right)\right\|^2 &= \left\|\left(\mathbf{I} - \tilde{\mathbf{P}}^{(n)}\right)\mathbf{W}_{0,(n)}\right\|_{\mathrm{F}}^2 \\ &\leqslant \left\|\left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\mathbf{W}_{0,(n)}\right\|_{\mathrm{F}}^2 \\ &= \left\|\left(\mathcal{X} - \hat{\mathcal{X}}_0\right) \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\right\|^2 \end{aligned} \tag{15}$$

must hold for each $n \in [N]$.

Continuing from (14) we can use the definition of $\hat{\mathcal{X}}_0$ in (4.28) to see that

$$\begin{aligned}\|\mathcal{W}_1\|^2 &\leqslant \sum_{n=1}^{N} \left\|\left(\mathcal{X} - \mathcal{X} \bigtimes_{h=1}^{N} \mathbf{P}^{(h)}\right) \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\right\|^2 \\ &= \sum_{n=1}^{N} \left\|\mathcal{X} \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right) - \mathcal{X} \bigtimes_{h=1}^{N} \mathbf{P}^{(h)} \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\right\|^2 \end{aligned} \tag{16}$$

by lemma 1. Due to the definition of $\bar{\mathbf{Q}}^{(n)}$ together with the fact that its rank is $\tilde{r}_n \geqslant r_n$ we can see that $\left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right)\mathbf{P}^{(n)} = \mathbf{0}$. As a consequence, lemma 1 implies that $\mathcal{X} \times_{h=1}^{N} \mathbf{P}^{(h)} \times_n$

$\left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right) = \mathbf{0}$ for all $n \in [N]$. Continuing from (16) we now have that

$$\|\mathcal{W}_1\|^2 \leqslant \sum_{n=1}^{N} \left\| \mathcal{X} \times_n \left(\mathbf{I} - \bar{\mathbf{Q}}^{(n)}\right) \right\|^2. \tag{17}$$

Again appealing to the definition of both $\bar{\mathbf{Q}}^{(n)}$ and $\bar{\mathbf{P}}^{(n)}$ in (4.29), combined with the fact that $\tilde{r}_n \geqslant \bar{r}_n$, finally yields the desired result. □

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, "Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg," *NeuroImage*, vol. 29, no. 3, pp. 938–947, 2006.

[2] A. Cichocki, "Tensor decompositions: A new concept in brain data analysis?" *arXiv preprint arXiv:1305.0395*, 2013.

[3] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[4] D. Letexier, S. Bourennane, and J. Blanc-Talon, "Nonorthogonal tensor matricization for hyperspectral image filtering," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, no. 1, pp. 3–7, 2008.

[5] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 8, pp. 1415–1428, 2009.

[6] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*. Springer, 2011, pp. 19–49.

[7] C. C. Aggarwal and C. K. Reddy, *Data clustering: algorithms and applications*. CRC Press, 2013.

[8] A. Cichocki, "Era of big data processing: a new approach via tensor networks and tensor decompositions," *arXiv preprint arXiv:1403.2048*, 2014.

[9] M. Steinbach, G. Karypis, V. Kumar *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, no. 1. Boston, 2000, pp. 525–526.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[11] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of computational biology*, vol. 6, no. 3-4, pp. 281–297, 1999.

[12] A. C. Cameron, J. B. Gelbach, and D. L. Miller, "Robust inference with multiway clustering," *Journal of Business & Economic Statistics*, 2012.

[13] R. Xu, D. Wunsch *et al.*, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, 2005.

[14] D. Fasulo, "An analysis of recent work on clustering algorithms," *Department of Computer Science & Engineering, University of Washington*, 1999.

[15] P. Smyth, "Model selection for probabilistic clustering using cross-validated likelihood," *Statistics and computing*, vol. 10, no. 1, pp. 63–72, 2000.

[16] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[17] N. Nguyen and R. Caruana, "Consensus clusterings," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 607–612.

[18] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[19] S. Bickel and T. Scheffer, "Multi-view clustering." in *ICDM*, vol. 4, 2004, pp. 19–26.

[20] X. Liu, S. Ji, W. Glanzel, and B. De Moor, "Multiview partitioning via tensor methods," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 5, pp. 1056–1069, 2013.

[21] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 838–845.

[22] A. Shashua, R. Zass, and T. Hazan, "Multi-way clustering using super-symmetric non-negative tensor factorization," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 595–608.

[23] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions On*, vol. 32, no. 11, pp. 2006–2021, 2010.

[24] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 1, pp. 6–20, 2009.

[25] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener, "Modeling and multiway analysis of chatroom tensors," in *Intelligence and Security Informatics*. Springer, 2005, pp. 256–268.

[26] F. Estienne, N. Matthijs, D. Massart, P. Ricoux, and D. Leibovici, "Multi-way modelling of high-dimensionality electroencephalographic data," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 1, pp. 59–72, 2001.

[27] S. Gourvénec, I. Stanimirova, C.-A. Saby, C. Airiau, and D. Massart, "Monitoring batch processes with the statis approach," *Journal of chemometrics*, vol. 19, no. 5-7, pp. 288–300, 2005.

[28] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[29] ——, "Rank, decomposition, and uniqueness for 3-way and n-way arrays," *Multiway data analysis*, vol. 33, pp. 7–18, 1989.

[30] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[31] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.

[32] M. Rajih, P. Comon, and R. A. Harshman, "Enhanced line search: A novel method to accelerate parafac," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1128–1147, 2008.

[33] D. Nion and L. De Lathauwer, "An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma," *Signal Processing*, vol. 88, no. 3, pp. 749–755, 2008.

[34] A.-H. Phan, P. Tichavsky, and A. Cichocki, "Fast alternating ls algorithms for high order candecomp/parafac tensor factorizations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 19, pp. 4834–4846, 2013.

[35] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[36] ——, "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[37] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms-part ii: definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.

[38] P. Tichavsky, A. H. Phan, and A. Cichocki, "Non-orthogonal tensor diagonalization, a tool for block tensor decompositions," *arXiv preprint arXiv:1402.1673*, 2014.

[39] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2010.

[40] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[41] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, "Symmetric tensor decomposition," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 525–529.

[42] G. Allen, "Sparse higher-order principal components analysis," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 27–36.

[43] J. Chen and Y. Saad, "On the tensor svd and the optimal low rank orthogonal approximation of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 4, pp. 1709–1734, 2009.

[44] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[45] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.

[46] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[47] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural networks for signal processing IX*, vol. 1, no. 1, p. 1, 1999.

[48] B. Thompson, "Canonical correlation analysis," *Encyclopedia of statistics in behavioral science*, 2005.

[49] T. Zhang and G. Lerman, "A novel m-estimator for robust pca," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 749–808, 2014.

[50] M. McCoy, J. A. Tropp *et al.*, "Two proposals for robust pca using semidefinite programming," *Electronic Journal of Statistics*, vol. 5, pp. 1123–1160, 2011.

[51] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.

[52] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[53] A. Ganesh, K. Min, J. Wright, and Y. Ma, "Principal component pursuit with reduced linear measurements," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1281–1285.

[54] G. Mateos and G. B. Giannakis, "Robust pca as bilinear decomposition with outlier-sparsity regularization," *Signal Processing, IEEE Transactions on*, vol. 60, no. 10, pp. 5176–5190, 2012.

[55] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *Information Theory, IEEE Transactions on*, vol. 57, no. 11, pp. 7221–7234, 2011.

[56] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[57] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[58] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex deconvolution, with applications," Tech. Rep., 2012.

[59] M. Mardani, G. Mateos, and G. Giannakis, "Decentralized sparsity-regularized rank minimization: Algorithms and applications," *Signal Processing, IEEE Transactions on*, vol. 61, no. 21, pp. 5374–5388, 2013.

[60] M. Mardani and G. Giannakis, "Robust network traffic estimation via sparsity and low rank," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4529–4533.

[61] M. Mardani, G. Mateos, and G. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *Information Theory, IEEE Transactions on*, vol. 59, no. 8, pp. 5186–5205, 2013.

[62] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[63] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Computer VisionECCV 2002*. Springer, 2002, pp. 707–720.

[64] Y. Li, "On incremental and robust subspace learning," *Pattern recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.

[65] B. Yang, "Projection approximation subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 43, no. 1, pp. 95–107, 1995.

[66] Y. Chi, Y. C. Eldar, and R. Calderbank, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 23, pp. 5947–5959, 2013.

[67] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *Signal Processing, IEEE Transactions on*, vol. 62, no. 16, pp. 4284–4297, 2014.

[68] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[69] J. Feng, H. Xu, and S. Yan, "Online robust pca via stochastic optimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 404–412.

[70] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1568–1575.

[71] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: dynamic tensor analysis," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 374–383.

[72] J. Sun, S. Papadimitriou, and S. Y. Philip, "Window-based tensor analysis on high-dimensional and multi-aspect streams." in *ICDM*, vol. 6, 2006, pp. 1076–1080.

[73] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.

[74] J. Li, G. Han, J. Wen, and X. Gao, "Robust tensor subspace learning for anomaly detection," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 89–98, 2011.

[75] D. Nion and N. D. Sidiropoulos, "Adaptive algorithms to track the parafac decomposition of a third-order tensor," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2299–2310, 2009.

[76] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *Signal Processing, IEEE Transactions on*, vol. 63, no. 10, pp. 2663–2677, 2015.

[77] H. Kasai, "Online low-rank tensor subspace tracking from incomplete data by cp decomposition using recursive least squares," *arXiv preprint arXiv:1602.07067*, 2016.

[78] M. Mørup, L. K. Hansen, S. M. Arnfred, L.-H. Lim, and K. H. Madsen, "Shift-invariant multilinear decomposition of neuroimaging data," *NeuroImage*, vol. 42, no. 4, pp. 1439–1450, 2008.

[79] H. Lee, Y.-D. Kim, A. Cichocki, and S. Choi, "Nonnegative tensor factorization for continuous eeg classification," *International journal of neural systems*, vol. 17, no. 04, pp. 305–317, 2007.

[80] J. Li, L. Zhang, D. Tao, H. Sun, and Q. Zhao, "A prior neurophysiologic knowledge free tensor-based scheme for single trial eeg classification," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 2, pp. 107–115, 2009.

[81] J. Li and L. Zhang, "Regularized tensor discriminant analysis for single trial eeg classification in bci," *Pattern Recognition Letters*, vol. 31, no. 7, pp. 619–628, 2010.

[82] Q. Zhao, C. F. Caiafa, A. Cichocki, L. Zhang, and A. H. Phan, "Slice oriented tensor decomposition of eeg data for feature extraction in space, frequency and time domains," in *Neural Information Processing*. Springer, 2009, pp. 221–228.

[83] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear theory and its applications, IEICE*, vol. 1, no. 1, pp. 37–68, 2010.

[84] ——, "Extended hals algorithm for nonnegative tucker decomposition and its applications for multiway analysis and classification," *Neurocomputing*, vol. 74, no. 11, pp. 1956–1969, 2011.

[85] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1700–1715, 2007.

[86] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C.-G. Bénar, and I. Merlet, "Eeg extended source localization: tensor-based vs. conventional methods," *NeuroImage*, vol. 96, pp. 143–157, 2014.

[87] E. Acar, C. Aykut-Bingol, H. Bingol, R. Bro, and B. Yener, "Multiway analysis of epilepsy tensors," *Bioinformatics*, vol. 23, no. 13, pp. i10–i18, 2007.

[88] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmini, and W. Van Paesschen, "Canonical decomposition of ictal scalp eeg reliably detects the seizure onset zone," *NeuroImage*, vol. 37, no. 3, pp. 844–854, 2007.

[89] M. Weis, F. Römer, M. Haardt, D. Jannek, and P. Husar, "Multi-dimensional space-time-frequency component analysis of event related eeg data using closed-form parafac," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 349–352.

[90] M. Barnathan, V. Megalooikonomou, C. Faloutsos, S. Faro, and F. B. Mohamed, "Twave: high-order analysis of functional mri," *Neuroimage*, vol. 58, no. 2, pp. 537–548, 2011.

[91] C. F. Beckmann and S. M. Smith, "Tensorial extensions of independent component analysis for multisubject fmri analysis," *Neuroimage*, vol. 25, no. 1, pp. 294–311, 2005.

[92] S. Ferdowsi, V. Abolghasemi, and S. Sanei, "Eeg-fmri integration using a partially constrained tensor factorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6191–6195.

[93] E. Karahan, P. A. Rojas-Lopez, M. L. Bringas-Vega, P. A. Valdes-Hernandez, and P. A. Valdes-Sosa, "Tensor analysis and fusion of multimodal brain images," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1531–1559, 2015.

[94] J. Dauwels, K. Srinivasan, M. R. Reddy, and A. Cichocki, "Near-lossless multichannel eeg compression based on matrix and tensor decompositions," *Biomedical and Health Informatics, IEEE Journal of*, vol. 17, no. 3, pp. 708–714, 2013.

[95] J. Dauwels, K. Srinivasan, M. Ramasubba Reddy, and A. Cichocki, "Multi-channel eeg compression based on matrix and tensor decompositions," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 629–632.

[96] X. Li, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," *arXiv preprint arXiv:1304.5637*, 2013.

[97] L. Lee, L. M. Harrison, and A. Mechelli, "A report of the functional connectivity workshop, dusseldorf 2002," *Neuroimage*, vol. 19, no. 2, pp. 457–465, 2003.

[98] C. Stam, W. De Haan, A. Daffertshofer, B. Jones, I. Manshanden, A. V. C. Van Walsum, T. Montez, J. Verbunt, J. De Munck, B. Van Dijk *et al.*, "Graph theoretical analysis of magnetoencephalographic functional connectivity in alzheimer's disease," *Brain*, vol. 132, no. 1, pp. 213–224, 2009.

[99] V. Krause, A. Schnitzler, and B. Pollok, "Functional network interactions during sensorimotor synchronization in musicians and non-musicians," *Neuroimage*, vol. 52, no. 1, pp. 245–251, 2010.

[100] Q. Luo, D. Mitchell, X. Cheng, K. Mondillo, D. Mccaffrey, T. Holroyd, F. Carver, R. Coppola, and J. Blair, "Visual awareness, emotion, and gamma band synchronization," *Cerebral Cortex*, vol. 19, no. 8, pp. 1896–1904, 2009.

[101] M. A. Kramer, F.-L. Chang, M. E. Cohen, D. Hudson, and A. J. Szeri, "Synchronization measures of the scalp electroencephalogram can discriminate healthy from alzheimer's subjects," *International journal of neural systems*, vol. 17, no. 02, pp. 61–69, 2007.

[102] M. Arthuis, L. Valton, J. Régis, P. Chauvel, F. Wendling, L. Naccache, C. Bernard, and F. Bartolomei, "Impaired consciousness during temporal lobe seizures is related to increased long-distance cortical–subcortical synchronization," *Brain*, vol. 132, no. 8, pp. 2091–2101, 2009.

[103] K. Maharajh, P. Teale, D. C. Rojas, and M. L. Reite, "Fluctuation of gamma-band phase synchronization within the auditory cortex in schizophrenia," *Clinical Neurophysiology*, vol. 121, no. 4, pp. 542–548, 2010.

[104] J. Altenburg, R. J. Vermeulen, R. L. Strijers, W. P. Fetter, and C. J. Stam, "Seizure detection in the neonatal eeg with synchronization likelihood," *Clinical neurophysiology*, vol. 114, no. 1, pp. 50–55, 2003.

[105] C. J. Stam and J. C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear Biomedical Physics*, vol. 1, no. 1, pp. 1–19, 2007.

[106] E. T. Bullmore and D. S. Bassett, "Brain graphs: graphical models of the human brain connectome," *Annual review of clinical psychology*, vol. 7, pp. 113–140, 2011.

[107] R. Prabhakaran, S. E. Blumstein, E. B. Myers, E. Hutchison, and B. Britton, "An event-related fmri investigation of phonological–lexical competition," *Neuropsychologia*, vol. 44, no. 12, pp. 2209–2221, 2006.

[108] R. Goebel, F. Esposito, and E. Formisano, "Analysis of functional image analysis contest (fiac) data with brainvoyager qx: From single-subject to cortically aligned group general linear model analysis and self-organizing group independent component analysis," *Human brain mapping*, vol. 27, no. 5, pp. 392–401, 2006.

[109] P. Boveroux, A. Vanhaudenhuyse, M.-A. Bruno, Q. Noirhomme, S. Lauwick, A. Luxen, C. Degueldre, A. Plenevaux, C. Schnakers, C. Phillips *et al.*, "Breakdown of within-and between-network resting state functional magnetic resonance imaging connectivity during propofol-induced loss of consciousness." *Anesthesiology*, vol. 113, no. 5, 2010.

[110] J. Schrouff, V. Perlbarg, M. Boly, G. Marrelec, P. Boveroux, A. Vanhaudenhuyse, M.-A. Bruno, S. Laureys, C. Phillips, M. Pélégrini-Issac *et al.*, "Brain functional integration decreases during propofol-induced loss of consciousness," *Neuroimage*, vol. 57, no. 1, pp. 198–205, 2011.

[111] C. Chang and G. H. Glover, "Time–frequency dynamics of resting-state brain connectivity measured with fmri," *Neuroimage*, vol. 50, no. 1, pp. 81–98, 2010.

[112] N. Leonardi and D. Van De Ville, "Identifying network correlates of brain states using tensor decompositions of whole-brain dynamic functional connectivity," in *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*. IEEE, 2013, pp. 74–77.

[113] A. G. Mahyari and S. Aviyente, "Identification of dynamic functional brain network states through tensor decomposition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2099–2103.

[114] F. Varela, J.-P. Lachaux, E. Rodriguez, and J. Martinerie, "The brainweb: phase synchronization and large-scale integration," *Nature reviews neuroscience*, vol. 2, no. 4, pp. 229–239, 2001.

[115] M. G. Knyazeva, M. Jalili, A. Brioschi, I. Bourquin, E. Fornari, M. Hasler, R. Meuli, P. Maeder, and J. Ghika, "Topography of eeg multivariate phase synchronization in early alzheimer's disease," *Neurobiology of aging*, vol. 31, no. 7, pp. 1132–1144, 2010.

[116] J. Żygierewicz, C. Sielużycki, R. König, and P. Durka, "Event-related desynchronization and synchronization in meg: framework for analysis and illustrative datasets related to discrimination of frequency-modulated tones," *Journal of neuroscience methods*, vol. 168, no. 1, pp. 239–247, 2008.

[117] J. R. Petrella, "Use of graph theory to evaluate brain networks: a clinical tool for a small world?" *Radiology*, vol. 259, no. 2, pp. 317–320, 2011.

[118] B. C. Van Wijk, C. J. Stam, and A. Daffertshofer, "Comparing brain networks of different size and connectivity density using graph theory," *PloS one*, vol. 5, no. 10, p. e13701, 2010.

[119] X.-N. Zuo, R. Ehmke, M. Mennes, D. Imperati, F. X. Castellanos, O. Sporns, and M. P. Milham, "Network centrality in the human functional connectome," *Cerebral Cortex*, vol. 22, no. 8, pp. 1862–1875, 2012.

[120] M. Boersma, D. J. Smit, H. de Bie, G. C. M. Van Baal, D. I. Boomsma, E. J. de Geus, H. A. Delemarre-van de Waal, and C. J. Stam, "Network analysis of resting state eeg in the developing young brain: structure comes with maturation," *Human brain mapping*, vol. 32, no. 3, pp. 413–425, 2011.

[121] M. E. Lynall, D. S. Bassett, R. Kerwin, P. J. McKenna, M. Kitzbichler, U. Muller, and E. Bullmore, "Functional connectivity and brain networks in schizophrenia," *The Journal of Neuroscience*, vol. 30, no. 28, pp. 9477–9487, 2010.

[122] B. C. Bernhardt, Z. Chen, Y. He, A. C. Evans, and N. Bernasconi, "Graph-theoretical analysis reveals disrupted small-world organization of cortical thickness correlation networks in temporal lobe epilepsy," *Cerebral cortex*, vol. 21, no. 9, pp. 2147–2157, 2011.

[123] Y. Liu, M. Liang, Y. Zhou, Y. He, Y. Hao, M. Song, C. Yu, H. Liu, Z. Liu, and T. Jiang, "Disrupted small-world networks in schizophrenia," *Brain*, vol. 131, no. 4, pp. 945–961, 2008.

[124] P. Barttfeld, B. Wicker, S. Cukier, S. Navarta, S. Lew, and M. Sigman, "A big-world network in asd: dynamical connectivity analysis reflects a deficit in long-range connections and an excess of short-range connections," *Neuropsychologia*, vol. 49, no. 2, pp. 254–263, 2011.

[125] F. D. V. Fallani, L. Astolfi, F. Cincotti, D. Mattia, M. G. Marciani, S. Salinari, J. Kurths, S. Gao, A. Cichocki, A. Colosimo *et al.*, "Cortical functional connectivity networks in normal and spinal cord injured patients: evaluation by graph analysis," *Human brain mapping*, vol. 28, no. 12, pp. 1334–1346, 2007.

[126] K. Supekar, V. Menon, D. Rubin, M. Musen, and M. D. Greicius, "Network analysis of intrinsic functional brain connectivity in alzheimer's disease," *PLoS computational biology*, vol. 4, no. 6, p. e1000100, 2008.

[127] M. Chavez, M. Valencia, V. Navarro, V. Latora, and J. Martinerie, "Functional modularity of background activities in normal and epileptic brain networks," *Phys. Rev. Lett.*, vol. 104, no. 11, p. 118701, Mar 2010.

[128] M. Valencia, M. Pastor, M. Fernández-Seara, J. Artieda, J. Martinerie, and M. Chavez, "Complex modular structure of large-scale brain networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 2, p. 023119, 2009.

[129] R. Guimera and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, pp. 895–900, 2005.

[130] M. Kirschner and J. Gerhart, "Evolvability," *Proceedings of the National Academy of Sciences*, vol. 95, no. 15, pp. 8420–8427, 1998.

[131] R. S. J. Frackowiak, K. J. Friston, C. Frith, R. Dolan, and C. J. Price, *Human Brain Function*, 2nd ed. Academic Press, 2003.

[132] O. Sporns, G. Tononi, and R. Kötter, "The human connectome: A structural description of the human brain," *PLoS Comput Biol*, vol. 1, no. 4, p. e42, 09 2005.

[133] S. L. Bressler, "Large-scale cortical networks and cognition," *Brain Research Reviews*, vol. 20, no. 3, pp. 288 – 304, 1995.

[134] R. Baumgartner, L. Ryner, W. Richter, R. Summers, M. Jarmasz, and R. Somorjai, "Comparison of two exploratory data analysis methods for fmri: fuzzy clustering vs. principal component analysis," *Magnetic Resonance Imaging*, vol. 18, no. 1, pp. 89–94, 2000.

[135] A. Meyer-Baese, A. Wismueller, and O. Lange, "Comparison of two exploratory data analysis methods for fmri: unsupervised clustering versus independent component analysis," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 3, pp. 387–398, 2004.

[136] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[137] C. Allefeld, "Eigenvalue decomposition as a generalized synchronization cluster analysis," *International Journal of Bifurcation and Chaos*, vol. 17, no. 10, pp. 3493–3497, 2008.

[138] D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore, "Hierarchical modularity in human brain functional networks," *Frontiers in Neuroinformatics*, vol. 3, no. 37, 2009.

[139] C. Zhou, L. Zemanová, G. Zamora, C. C. Hilgetag, and J. Kurths, "Hierarchical organization unveiled by functional connectivity in complex brain networks," *Physical review letters*, vol. 97, no. 23, p. 238103, 2006.

[140] L. Zemanová, C. Zhou, and J. Kurths, "Structural and functional clusters of complex brain networks," *Physica D: Nonlinear Phenomena*, vol. 224, no. 1, pp. 202–212, 2006.

[141] D. Meunier, R. Lambiotte, and E. T. Bullmore, "Modular and hierarchically modular organization of brain networks," *Frontiers in neuroscience*, vol. 4, 2010.

[142] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 4, 2007.

[143] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory," *Czechoslovak Mathematical Journal*, vol. 25, no. 4, pp. 619–633, 1975.

[144] E. Rodriguez, N. George, J.-P. Lachaux, J. Martinerie, B. Renault, and F. J. Varela, "Perception's shadow: long-distance synchronization of human brain activity," *Nature*, vol. 397, no. 6718, pp. 430–433, 1999.

[145] M. Le Van Quyen, J. Foucher, J.-P. Lachaux, E. Rodriguez, A. Lutz, J. Martinerie, and F. J. Varela, "Comparison of hilbert transform and wavelet methods for the analysis of neuronal synchrony," *Journal of neuroscience methods*, vol. 111, no. 2, pp. 83–98, 2001.

[146] P. Tass, M. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, and H.-J. Freund, "Detection of n: m phase locking from noisy data: application to magnetoencephalography," *Physical Review Letters*, vol. 81, no. 15, p. 3291, 1998.

[147] J.-P. Lachaux, A. Lutz, D. Rudrauf, D. Cosmelli, M. Le Van Quyen, J. Martinerie, and F. Varela, "Estimating the time-course of coherence between single-trial brain signals: an introduction to wavelet coherence," *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 32, no. 3, pp. 157–174, 2002.

[148] S. Aviyente, E. M. Bernat, W. S. Evans, and S. R. Sponheim, "A phase synchrony measure for quantifying dynamic functional integration in the brain," *Human brain mapping*, vol. 32, no. 1, pp. 80–93, 2011.

[149] S. Aviyente and A. Y. Mutlu, "A time-frequency-based approach to phase and phase synchrony estimation," *Signal Processing, IEEE Transactions on*, vol. 59, no. 7, pp. 3086–3098, 2011.

[150] A. Rihaczek, "Signal energy distribution in time and frequency," in *IEEE Transactions: Information Theory*, vol. 14, 1968, pp. 369–274.

[151] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

[152] J. Forman, P. Clemons, S. Schreiber, and S. Haggarty, "Spectralnet - an application for spectral graph analysis and visualization," *BMC Bioinformatics*, vol. 6, no. 1, p. 260, 2005.

[153] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," *SIAM International Conference on Data Mining*, pp. 76–84, 2005.

[154] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[155] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 849–856.

[156] F. Lin and W. W. Cohen, "Power iteration clustering," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 655–662.

[157] C. Ding and X. He, "Linearized cluster assignment via spectral ordering," in *Proceedings of the twenty-first international conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, p. 30.

[158] M. Holzrichter and S. Oliveira, "A graph based method for generating the fiedler vector of irregular problems," in *In Lecture Notes in Computer Science*, 1999, pp. 978–985.

[159] N. Nguyen and R. Caruana, "Consensus clusterings," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 607–612.

[160] A. P. Topchy, M. H. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 225–232.

[161] A. P. Topchy, A. K. Jain, and W. F. Punch, "A mixture model of clustering ensembles," in *SDM*. SIAM, 2004, pp. 379–390.

[162] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in vlsi domain," in *IEEE Trans. Very Larg Scale Integration (VLSI) Systems*, 1999, pp. 69–529.

[163] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[164] K. Steinhaeuser and N. V. Chawla, "Identifying and evaluating community structure in complex networks," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413 – 421, 2010.

[165] M. Ovelgonne and A. Geyer-Schulz, "Cluster cores and modularity maximization," *Data Mining Workshops, International Conference on*, vol. 0, pp. 1204–1213, 2010.

[166] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[167] J. B. Garner, "The standard error of cohen's kappa," *Statistics in medicine*, vol. 10, no. 5, pp. 767–775, 1991.

[168] C. J. Rijsbergen, "Information retrieval," *Journal of the American Society for Information Science*, vol. 30, no. 6, pp. 374–375, 1979.

[169] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 410–420, 2007.

[170] J. Lin, "Divergence measures based on the shannon entropy," *Information Theory, IEEE Transactions on*, vol. 37, no. 1, pp. 145–151, 1991.

[171] M. C. V. Nascimento and A. C. P. L. F. de Carvalho, "Spectral methods for graph clustering - a survey," *European Journal of Operational Research*, vol. 211, no. 2, pp. 221 – 231, 2011.

[172] S. Yu, X. Liu, L. C. Tranchevent, W. Glänzel, J. A. K. Suykens, B. De Moor, and Y. Moreau, "Optimized data fusion for k-means laplacian clustering," *Bioinformatics*, vol. 27, pp. 118–126, January 2011.

[173] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke, "Effects of crossmodal divided attention on late erp components. ii. error processing in choice reaction tasks," *Electroencephalography and clinical neurophysiology*, vol. 78, no. 6, pp. 447–455, 1991.

[174] W. J. Gehring, B. Goss, M. G. Coles, D. E. Meyer, and E. Donchin, "A neural system for error detection and compensation," *Psychological science*, vol. 4, no. 6, pp. 385–390, 1993.

[175] J. R. Hall, E. M. Bernat, and C. J. Patrick, "Externalizing psychopathology and the error-related negativity," *Psychological Science*, vol. 18, no. 4, pp. 326–333, 2007.

[176] J. Kayser and C. E. Tenke, "Principal components analysis of laplacian waveforms as a generic method for identifying erp generator patterns: Ii. adequacy of low-density estimates," *Clinical neurophysiology*, vol. 117, no. 2, pp. 369–380, 2006.

[177] C. E. Tenke and J. Kayser, "Generator localization by current source density (csd): Implications of volume conduction and field closure at intracranial and scalp resolutions," *Clinical neurophysiology*, vol. 123, no. 12, pp. 2328–2345, 2012.

[178] M. Bolaños, E. M. Bernat, B. He, and S. Aviyente, "A weighted small world network measure for assessing functional connectivity," *Journal of neuroscience methods*, vol. 212, no. 1, pp. 133–142, 2013.

[179] J. F. Cavanagh, M. X. Cohen, and J. J. B. Allen, "Prelude to and resolution of an error: Eeg phase synchrony reveals cognitive control dynamics during action monitoring," *The Journal of Neuroscience*, vol. 29, no. 1, pp. 98–105, 2009.

[180] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[181] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave de Peralta, "Eeg source imaging," *Clinical neurophysiology*, vol. 115, no. 10, pp. 2195–2222, 2004.

[182] C. S. Herrmann, M. H. Munk, and A. K. Engel, "Cognitive functions of gamma-band activity: memory match and utilization," *Trends in cognitive sciences*, vol. 8, no. 8, pp. 347–355, 2004.

[183] M. Steriade, P. Gloor, R. Llinas, F. L. Da Silva, and M.-M. Mesulam, "Basic mechanisms of cerebral rhythmic activities," *Electroencephalography and clinical neurophysiology*, vol. 76, no. 6, pp. 481–508, 1990.

[184] S. L. Bressler, "Large-scale cortical networks and cognition," *Brain Research Reviews*, vol. 20, no. 3, pp. 288–304, 1995.

[185] R. M. Hutchison, T. Womelsdorf, E. A. Allen, P. A. Bandettini, V. D. Calhoun, M. Corbetta, S. Della Penna, J. H. Duyn, G. H. Glover, J. Gonzalez-Castillo *et al.*, "Dynamic functional connectivity: promise, issues, and interpretations," *Neuroimage*, vol. 80, pp. 360–378, 2013.

[186] F. Freyer, K. Aquino, P. A. Robinson, P. Ritter, and M. Breakspear, "Bistability and non-gaussian fluctuations in spontaneous cortical activity," *The Journal of Neuroscience*, vol. 29, no. 26, pp. 8512–8524, 2009.

[187] J. Tang, S. Scellato, M. Musolesi, C. Mascolo, and V. Latora, "Small-world behavior in time-varying graphs," *Physical Review E*, vol. 81, no. 5, p. 055101, 2010.

[188] P. Grindrod, M. C. Parsons, D. J. Higham, and E. Estrada, "Communicability across evolving networks," *Physical Review E*, vol. 83, no. 4, p. 046120, 2011.

[189] M. Valencia, J. Martinerie, S. Dupont, and M. Chavez, "Dynamic small-world behavior in functional brain networks unveiled by an event-related networks approach," *Physical Review E*, vol. 77, no. 5, p. 050905, 2008.

[190] F. D. V. Fallani, V. Latora, L. Astolfi, F. Cincotti, D. Mattia, M. G. Marciani, S. Salinari, A. Colosimo, and F. Babiloni, "Persistent patterns of interconnection in time-varying cortical networks estimated from high-resolution eeg recordings in humans during a simple motor act," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 22, p. 224014, 2008.

[191] E. A. Allen, E. Damaraju, S. M. Plis, E. B. Erhardt, T. Eichele, and V. D. Calhoun, "Tracking whole-brain connectivity dynamics in the resting state," *Cerebral cortex*, p. bhs352, 2012.

[192] S. Dimitriadis, N. Laskaris, and A. Tzelepi, "On the quantization of time-varying phase synchrony patterns into distinct functional connectivity microstates (fcμstates) in a multi-trial visual erp paradigm," *Brain topography*, vol. 26, no. 3, pp. 397–409, 2013.

[193] T. Koenig, L. Prichep, D. Lehmann, P. V. Sosa, E. Braeker, H. Kleinlogel, R. Isenhart, and E. R. John, "Millisecond by millisecond, year by year: normative eeg microstates and developmental stages," *Neuroimage*, vol. 16, no. 1, pp. 41–48, 2002.

[194] S. Ma, V. D. Calhoun, R. Phlypo, and T. Adalı, "Dynamic changes of spatial functional network connectivity in healthy individuals and schizophrenia patients using independent vector analysis," *NeuroImage*, vol. 90, pp. 196–206, 2014.

[195] R. F. Betzel, M. A. Erickson, M. Abell, B. F. O'Donnell, W. P. Hetrick, and O. Sporns, "Synchronization dynamics and evidence for a repertoire of network states in resting eeg," *Frontiers in computational neuroscience*, vol. 6, 2012.

[196] N. Leonardi, J. Richiardi, M. Gschwind, S. Simioni, J.-M. Annoni, M. Schluep, P. Vuilleumier, and D. Van De Ville, "Principal components of functional connectivity: a new approach to study dynamic brain connectivity during rest," *NeuroImage*, vol. 83, pp. 937–950, 2013.

[197] S. Mehrkanoon, M. Breakspear, and T. W. Boonstra, "Low-dimensional dynamics of resting-state cortical activity," *Brain topography*, vol. 27, no. 3, pp. 338–352, 2014.

[198] A. Y. Mutlu, E. Bernat, and S. Aviyente, "A signal-processing-based approach to time-varying graph analysis for dynamic brain network identification," *Computational and mathematical methods in medicine*, vol. 2012, 2012.

[199] N. Leonardi, W. R. Shirer, M. D. Greicius, and D. Van De Ville, "Disentangling dynamic networks: Separated and joint expressions of functional connectivity patterns in time," *Human brain mapping*, vol. 35, no. 12, pp. 5984–5995, 2014.

[200] I. Cribben, R. Haraldsdottir, L. Y. Atlas, T. D. Wager, and M. A. Lindquist, "Dynamic connectivity regression: determining state-related changes in brain connectivity," *Neuroimage*, vol. 61, no. 4, pp. 907–920, 2012.

[201] J. Zhang, X. Li, C. Li, Z. Lian, X. Huang, G. Zhong, D. Zhu, K. Li, C. Jin, X. Hu *et al.*, "Inferring functional interaction and transition patterns via dynamic bayesian variable partition models," *Human brain mapping*, vol. 35, no. 7, pp. 3314–3331, 2014.

[202] J. Ou, Z. Lian, L. Xie, X. Li, P. Wang, Y. Hao, D. Zhu, R. Jiang, Y. Wang, Y. Chen *et al.*, "Atomic dynamic functional interaction patterns for characterization of adhd," *Human brain mapping*, vol. 35, no. 10, pp. 5262–5278, 2014.

[203] M. H. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 377–391, 2006.

[204] C. Qiu and N. Vaswani, "Automated recursive projected cs (reprocs) for real-time video layering," in *International Conference on Computer Vision and Pattern Recognition, CVPR 2012*, vol. 130. Citeseer, 2012.

[205] A. Ozdemir, M. Bolanos, E. Bernat, and S. Aviyente, "Hierarchical spectral consensus clustering for group analysis of functional brain networks," *Biomedical Engineering, IEEE Transactions on*, vol. 62, no. 9, pp. 2158–2169, 2015.

[206] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[207] S. Roweis, "Em algorithms for pca and spca," *Advances in neural information processing systems*, pp. 626–632, 1998.

[208] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[209] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[210] A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast algorithms for recovering a corrupted low-rank matrix," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*. IEEE, 2009, pp. 213–216.

[211] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1494–1501.

[212] Y. Li, L.-Q. Xu, J. Morphett, and R. Jacobs, "An integrated algorithm of incremental and robust pca," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1. IEEE, 2003, pp. I–245.

[213] C. Qiu, N. Vaswani, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5954–5958.

[214] C. Qiu and N. Vaswani, "Recursive sparse recovery in large but structured noisepart 2," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 864–868.

[215] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," *Contributions to mathematical psychology*, pp. 109–127, 1964.

[216] D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore, "Hierarchical modularity in human brain functional networks," *Frontiers in neuroinformatics*, vol. 3, 2009.

[217] D. Meunier, S. Achard, A. Morcom, and E. Bullmore, "Age-related changes in modular organization of human brain functional networks," *Neuroimage*, vol. 44, no. 3, pp. 715–723, 2009.

[218] S. Friedland, Q. Li, and D. Schonfeld, "Compressive sensing of sparse tensors," *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4438–4447, 2014.

[219] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[220] J. R. Hall, E. M. Bernat, and C. J. Patrick, "Externalizing psychopathology and the error-related negativity," *Psychological Science*, vol. 18, no. 4, pp. 326–333, 2007.

[221] J. F. Cavanagh, M. X. Cohen, and J. J. Allen, "Prelude to and resolution of an error: Eeg phase synchrony reveals cognitive control dynamics during action monitoring," *The Journal of Neuroscience*, vol. 29, no. 1, pp. 98–105, 2009.

[222] J. Kayser and C. E. Tenke, "Principal components analysis of laplacian waveforms as a generic method for identifying erp generator patterns: Ii. adequacy of low-density estimates," *Clinical neurophysiology*, vol. 117, no. 2, pp. 369–380, 2006.

[223] J. Zhan and N. Vaswani, "Robust pca with partial subspace knowledge," in *Information Theory (ISIT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 2192–2196.

[224] F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, "Decomposing eeg data into space–time–frequency components using parallel factor analysis," *NeuroImage*, vol. 22, no. 3, pp. 1035–1045, 2004.

[225] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[226] L. Cheng, Y.-C. Wu, and H. V. Poor, "Probabilistic tensor canonical polyadic decomposition with orthogonal factors," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 663–676, 2015.

[227] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6315–6328, 2015.

[228] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 792–799.

[229] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari, "Non-negative tensor factorization using alpha and beta divergences," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3. IEEE, 2007, pp. III–1393.

[230] ——, "Novel multi-layer non-negative tensor factorization with sparsity constraints," in *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2007, pp. 271–280.

[231] S. K. Suter, M. Makhynia, and R. Pajarola, "Tamresh–tensor approximation multiresolution hierarchy for interactive volume visualization," in *Computer Graphics Forum*, vol. 32, no. 3pt2. Wiley Online Library, 2013, pp. 151–160.

[232] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2. IEEE, 2002, pp. 511–514.

[233] J. Yang, D. Zhang, A. F. Frangi, and J.-y. Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 131–137, 2004.

[234] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in neural information processing systems*, 2005, pp. 499–506.

[235] X. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16. MIT, 2004, p. 153.

[236] G. Dai and D.-Y. Yeung, "Tensor embedding methods," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 330.

[237] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 846–853.

[238] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1208–1213.

[239] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 2, pp. 342–352, 2008.

[240] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[241] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A1027–A1052, 2012.

[242] A. Özdemir, M. A. Iwen, and S. Aviyente, "Multiscale tensor decomposition," in *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE, 2016, pp. 625–629.

[243] A. Ozdemir, M. A. Iwen, and S. Aviyente, "Locally linear low-rank tensor approximation," in *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015, pp. 839–843.

[244] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 94–106.

[245] A. Karami, M. Yazdi, and G. Mercier, "Compression of hyperspectral images using discerete wavelet transform and tucker decomposition," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 444–450, 2012.

[246] M. Iwen and B. Ong, "A distributed and incremental svd algorithm for agglomerative data analysis on large networks," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 4, pp. 1699–1718, 2016.

[247] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1993.

[248] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[249] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1615–1618, 2003.

[250] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.

[251] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," *arXiv preprint arXiv:1202.3725*, 2012.

[252] B. B. Biswal, M. Mennes, X.-N. Zuo, S. Gohel, C. Kelly, S. M. Smith, C. F. Beckmann, J. S. Adelstein, R. L. Buckner, S. Colcombe *et al.*, "Toward discovery science of human

brain function," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4734–4739, 2010.

[253] S. Whitfield-Gabrieli and A. Nieto-Castanon, "Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks," *Brain connectivity*, vol. 2, no. 3, pp. 125–141, 2012.

[254] J. L. Lancaster, M. G. Woldorff, L. M. Parsons, M. Liotti, C. S. Freitas, L. Rainey, P. V. Kochunov, D. Nickerson, S. A. Mikiten, and P. T. Fox, "Automated talairach atlas labels for functional brain mapping," *Human brain mapping*, vol. 10, no. 3, pp. 120–131, 2000.

[255] A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5450–5463, 2015.

[256] R. Bro and H. A. Kiers, "A new efficient method for determining the number of components in parafac models," *Journal of chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.

[257] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2016.

[258] J. P. C. da Costa, M. Haardt, and F. Romer, "Robust methods based on the hosvd for estimating the model order in parafac models," in *Sensor Array and Multichannel Signal Processing Workshop, 2008. SAM 2008. 5th IEEE*. IEEE, 2008, pp. 510–514.

[259] E. E. Papalexakis and C. Faloutsos, "Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5441–5445.

[260] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian cp factorization of incomplete tensors with automatic rank determination," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015.

[261] M. Araujo, S. Papadimitriou, S. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis, and D. Koutra, "Com2: fast automatic discovery of temporal (comet) communities," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 271–283.

[262] M. Mørup and L. K. Hansen, "Automatic relevance determination for multi-way models," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 352–363, 2009.

[263] M. E. Timmerman and H. A. Kiers, "Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima," *British journal of mathematical and statistical psychology*, vol. 53, no. 1, pp. 1–16, 2000.

[264] H. A. Kiers and A. Kinderen, "A fast method for choosing the numbers of components in tucker3 analysis," *British Journal of Mathematical and Statistical Psychology*, vol. 56, no. 1, pp. 119–125, 2003.

[265] C. E. Tsourakakis, "Mach: Fast randomized tensor decompositions," in *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 2010, pp. 689–700.

[266] I. Jeon, E. E. Papalexakis, U. Kang, and C. Faloutsos, "Haten2: Billion-scale tensor decompositions," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 1047–1058.

[267] E. Papalexakis, C. Faloutsos, and N. Sidiropoulos, "Parcube: Sparse parallelizable tensor decompositions," *Machine Learning and Knowledge Discovery in Databases*, pp. 521–536, 2012.

[268] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, 2016.

[269] W. Austin, G. Ballard, and T. G. Kolda, "Parallel tensor compression for large-scale scientific data," in *Parallel and Distributed Processing Symposium, 2016 IEEE International*. IEEE, 2016, pp. 912–922.