DYNAMIC NETWORK ANALYSIS WITH APPLICATIONS TO FUNCTIONAL NEURAL CONNECTIVITY

By

Arash Goilbagh Mahyari

A DISSERTATION

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

Electrical Engineering – Doctor of Philosophy

2017

ABSTRACT

DYNAMIC NETWORK ANALYSIS WITH APPLICATIONS TO FUNCTIONAL NEURAL CONNECTIVITY

By

Arash Goilbagh Mahyari

Contemporary neuroimaging techniques provide neural activity recordings with increasing spatial and temporal resolution yielding rich multichannel datasets that can be exploited for detailed description of anatomical and functional connectivity patterns in the brain. Studies indicate that the changes in functional connectivity patterns across spatial and temporal scales play an important role in a wide range of cognitive and executive processes such as memory and attention as well as in the understanding the causes of many neural diseases and psychopathologies such as epilepsy, Alzheimers, Parkinsons and schizophrenia. Early work in the area was limited to the analysis of static brain networks obtained through averaging long-term functional connectivity, thus neglecting possible time-varying connections. There is growing evidence that functional networks dynamically reorganize and coordinate on millisecond scale for the execution of mental processes. Functional networks consist of distinct network states, where each state is defined as a period of time during which the network topology is quasi-stationary. For this reason, there has been an interest in characterizing the dynamics of functional networks using high temporal resolution electroencephalogram recordings.

In this thesis, dynamic functional connectivity networks are represented by multiway arrays, tensors, which are able to capture the complete topological structure of the networks. This thesis proposes new methods for both tracking the changes in these dynamic networks and characterizing or summarizing the network states. In order to achieve this goal, a Tucker decomposition based approach is introduced for detecting the change points for task-based electroencephalogram (EEG) functional connectivity networks through calculating the subspace distance between consecutive time steps. This is followed by a tensor-matrix projection based approach for summarizing multiple networks within a time interval.

Tensor based summarization approaches do not necessarily result in sparse network and succinct states. Moreover, subspace based summarizations tend to capture the background brain activity more than the low energy sparse activations. For this reason, we propose utilizing the sparse common component and innovations (SCCI) model which simultaneously finds the sparse common component of multiple signals. However, as the number of signals in the model increases, this becomes computationally prohibitive. In this thesis, a hierarchical algorithm to recover the common component in the SCCI model is proposed for large number of signals. The hierarchical recovery of SCCI model solves the time and memory limitations at the expense of a slight decrease in the accuracy. This hierarchical model is used to separate the common and innovation components of functional connectivity networks across time. The innovation components are tracked over time to detect the change points, and the common component of the detected network states are used to obtain the network summarization.

SCCI recovery algorithm finds the sparse representation of the common and innovation components of signals with respect to pre-determined dictionaries. However, input signals are not always well-represented by pre-determined dictionaries. In this thesis, a structured dictionary learning algorithm for SCCI model is developed. The proposed method is applied to EEG data collected during a study of error monitoring where two different types of brain responses are elicited in response to the stimulus. The learned dictionaries can discriminate between the response types and extract the error-related potentials (ERP) corresponding to the two responses.

In dedication to my mom, who dedicated herself to my education and never saw my graduation.

ACKNOWLEDGEMENTS

First I would like to thank my Ph.D. advisor, Prof. Selin Aviyente. It has been an honor for me to be her Ph.D. student. She has supported me in all aspects of the research and the education during my Ph.D. program. She taught me how to be a good researcher in this field. I appreciate all her contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

It is my honor to thank my Ph.D. guidance committee, Prof. Haydar Radha, Prof. Ramakrishna Mukkamala, and Dr. David Zhu for their unconditional support during my study. Special thanks goes to Prof. John Deller for being inspiration in many ways. He generously gave his time to offer me insightful advice toward improving my professional life.

I am grateful for having the support of my colleagues at the Signal and Image Processing Laboratory, Marisel Villafane–Delgado, Alp Ozdemir, Esra Al–sharo, Dr. Rodney Singleton, and Dr. Mahmood Al–Khassaweneh, in my professional and personal life.

I would like to express my gratitude to the technical and administrative staff of the college of engineering and the department of Electrical and Computer Engineering at Michigan State University for their assistance during my study, in particular, Dr. Katy Luchini Colbry, Meagan Kroll, Laurie Rashid, and Michelle Stewart.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by the National Science Foundation.

Most importantly, I would like to thank my family for all their love and encouragement, for my parents who raised me with a love of science and supported me in all my pursuits, and most of all for my loving, supportive, encouraging, and patient wife, Azin, whose faithful support during the final stages of this Ph.D. is so appreciated.

Arash Mahyari
Michigan State University
May 2017

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

Empirical studies from electromagnetic recordings and neuroimaging suggest that human cognition arises from transient synchronization between distant and specific neural populations [2, 3, 4]. The interactions between these functionally distinct neurocognitive networks, also known as functional connectivity (FC), have been studied in detail through neuronal time series recorded from different sensors, e.g. electroencephalogram (EEG) [5], or from brain voxels, e.g. functional magnetic resonance imaging (fMRI) [6, 7]. EEG measures electrical signals and perturbations of the extracranial magnetic fields, and have fairly good temporal but poor spatial resolutions. On the other hand, fMRI detects changes in regional brain activity by measuring blood oxygenation levels, and it has a high spatial but poor temporal resolution. Recently, methods from complex network theory have been used to describe the topological structure and attributes of functional connectivity networks [8, 9], where nodes correspond to distinct brain regions and edges to pairwise associations (or functional connectivity) between them [10]. Fig. 1.1 shows the schematic representation of building graphs from the signals recorded by EEG.

The pairwise associations are quantified through large-scale synchronization [11]. Types of indices used for quantifying large-scale synchrony can be broadly categorized into linear and nonlinear measures [12]. Linear correlation measures include temporal correlation, spectral coherence, directed transfer function [13], partial directed coherence [14], and Granger causality [15], and attempt to quantify the degree of synchronization. These methods are closely related to each other and rely on a multivariate autoregressive model for describing the dynamics of multichannel EEG signals. The magnitude of the transfer function between different channels quantifies the connectivity between the different cortical sites. These measures have limitations for the following reasons. First of all, they rely on parametric modeling of EEG signals and thus suffer from some of the common problems with parametric models such as determination of the order and robustness of the parameters to perturbations. Second, they assume stationarity of the underlying signals by defining the transfer function solely in the frequency domain and the linearity of the interactions between different channels. This linearity assumption restricts the types of interactions that can be quantified to amplitude effects and does not allow the separation of the effects of amplitude and phase. Nonlinear correlation measures, on the other hand, attempt to address this limitation of linear measures by quantifying temporal correlation through measures of phase synchrony [16, 17] and generalized synchronization measures, including information theoretic measures [18, 19]. A desirable phase synchrony measure should be able to separate the phase and amplitude effects from each other and take the nonstationary nature of brain activity into account. Two different measures for quantifying time-varying phase synchrony have been proposed to address these issues. The first method uses the Hilbert transform of the signal to get an analytic form of the signal and estimates instantaneous phase directly from its analytic form [20]. To estimate the instantaneous phase of a signal from its analytic form, one has to ensure that the signal is composed of a narrowband of frequencies. Thus, the Hilbert transform method of computing phase synchrony requires first bandpass filtering of the signal around a frequency of interest and then applying the Hilbert transform to obtain the instantaneous phase. This is an indirect way of deriving the frequency-dependent phase estimates and is prone to error especially in the case of noisy signals as noise tends to be wideband. The second approach computes a time-varying complex energy spectrum using either the continuous wavelet transform with a complex Morlet wavelet [21] or the short-time Fourier transform (STFT) [22]. It has been observed that the two approaches are similar in their results with the time-varying spectrum-based methods giving higher resolution phase synchrony estimates over time and frequency, especially at the low frequency range [16]. Although the wavelet- and STFT-based phase synchrony estimates address the issue of nonstationarity, they suffer from a number of drawbacks. In the case of the wavelet transform, the nonuniform resolution across time and frequency results

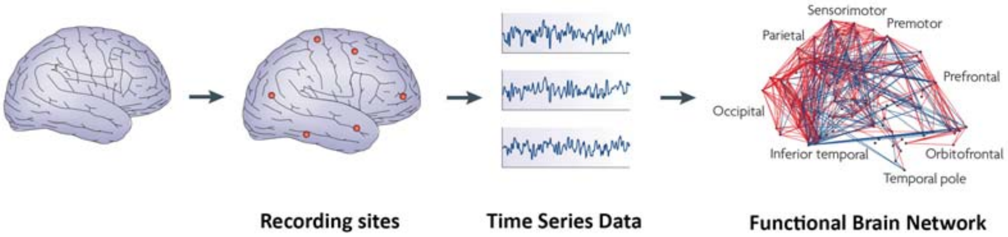


Figure 1.1 Schematic presentation of functional network construction.

in biased energy representations and corresponding phase estimates. In the case of STFT, there is a tradeoff between time and frequency resolution because of the window function. Phase-synchrony based on Reduced Interference Distribution (RID)-Rihaczek distribution, a bilinear time frequency distribution, was proposed to better track dynamic changes in the synchrony of neural signals [23]. This measure derives the signal energy distribution in time and frequency by application of the complex signal notation, and remove the crossterms for multicomponent signals by applying the Choi-Williams kernel function. In this thesis, we use RID-Rihaczek based phase synchrony to quantify the functional connectivity.

Once the functional connectivity graphs are constructed, various tools from graph theory have been adapted to the study of these networks. These methods can be categorized as local, intermediate and global scale measures. At the global scale, small-world networks have been used to model brain connectivity. Small-world networks demonstrate clustered interconnectivity within groups of nodes and a short path length between any two nodes [24]. Small-world model has been used to study the functional connectivity of healthy and disease populations from fMRI [25] and magnetoencephalogram (MEG) [26]. For example, in [27], the interfere of small-world architecture with brain pathology such as schizophrenia and Alzheimers diseases is shown. In [28] also, global and local efficiency measures of parallel information processing were used to show that the functional connectivity network is economical in the sense of supporting high efficiency for low cost. Degree correlation and hierarchy are used to find division-specific organization of anatomical connections between regions [29]. At the intermediate-scale level, interdensity, the ratio between the actual number of links in a set and the total number of possible links in the same set, is used frequently to determine the optimal partition, consisting of separate clusters [30]. A module is defined as a subset of units within a network such that connections between them are denser than connections with the rest of the network. Human brain has a modular organizations with the connector nodes appearing at the junctions between anatomically segregated cortices and the provincial hubs located within functionally specialized areas of cortex. At the local scale, measures such as centrality, clustering coefficient and path length have been computed. Centrality measures identify hub nodes in neural networks [31].

Until recently, most of the analysis has been carried out with the assumption that functional connectivity networks are static or the connections do not change with time. In static networks, topological features of functional connectivity networks are defined over long periods of time, neglecting possible instantaneous time-varying properties of the topologies. In this analysis, the connectivities over a time and frequency window of interest are averaged and are assumed to remain unaltered while recent studies have shown that the functional connectivity in brain networks may exhibit dynamic changes within short time scales due to task demands, learning and anesthesia [32, 33, 34, 35, 36]. Moreover, there is growing evidence suggesting that the emergence of a unified neural process is controlled by the continuous formation and destruction of functional connectivity over multiple time scales [37, 38]. Early work in the area of dynamic network connectivity focused on extending measures from static graph theory to dynamic graphs. Valencia *e*t al. [39] showed brain connectivity patterns vary with time and frequency while maintaining a small-world structure. In other work, the global and local efficiency measures are applied to graphs across time and some network metrics are obtained forming a time series [40]. Other methods include finding communities [41], and extracting features [40] from graphs at each time and tracking their time series over time. These methods do not evaluate the dynamic evolution of connectivity over time, and cannot provide a meaningful summarization of time-varying network topology.

Recent work in cognitive neuroscience, for example, now suggests the presence of two broad functional brain networks that directly reflect primary systems underlying goal-directed behavior: incentive salience and cognitive control. The incentive salience network's primary function is to scan the environment for the anticipation of reward and detection of potential danger, and centrally involves the anterior cingulate cortex (ACC), amygdala, and the major dopaminergic structures of the brain [42, 43, 44]. The salience network is functionally integrated with other brain structures that collectively constitute the cognitive control network, notably the lateral (L)PFC, orbitofrontal PFC, and inferior frontal gyrus [44, 45, 46]. After receiving input from the salience network, the control network becomes active to guide responses, often modifying prepotent behavior [47, 48]. This guidance entails the use of mental representations of potential future outcomes, including weighting the potential positive and negative outcomes of different response options [48]. Error-processing studied using the error-related negativity is understood to involve both salience and control processing, and will be the focus of an initial application of the proposed approach.

In recent years, two complementary approaches to dynamic functional connectivity (dFC) tracking have been proposed. In the first class of approaches, it is assumed that the network is in one particular state at each time point and the states fluctuate across time. For example, Allen et al. [49] proposed a data-driven approach based on k-means clustering to identify "FC-states" from resting state fMRI. Although clustering approaches provide a potentially powerful method for determining the spontaneous changes in resting state FC, they assume a distinct network state at each time point instead of a network state that is both contiguous in time and made up of multiple building blocks [50]. Similarly, in [51] fluctuations related to spatial components, obtained through independent vector analysis (IVA) over time, are investigated through clustering and Markov modeling. In another study [52], resting state EEG FC networks were grouped into coherent states through cross-correlation analysis and evolutionary clustering of FC edge time series. In the second class of approaches, Leonardi et al. [53] propose a principal component analysis (PCA) based approach to reveal characteristic fluctuations of whole-brain FC, where the eigenconnectivities that account for the most variation in the collection of dynamic FCNs across time and subjects are denoted as network states. The retained eigenconnectivities are treated as building blocks of dynamic FC and the subject-specific time-dependent weights that describe the contribution of each eigenconnectivity is obtained through orthogonal projection. Similarly, in [54] a PCA based approach is used to reconstruct the principal resting-state network dynamics from EEG data. The multivariate data describing time-frequency interdependencies between electrodes is analyzed through PCA to obtain low-dimensional orthogonal subspace of resting-state network dynamics. In previous work [55], a PCA based approach for tracking dFC in task-based EEG was introduced based on average dFC across subjects. In [50], two approaches to network state identification, i.e. clustering and PCA, have been compared for dynamic connectivity analysis of resting-state and task-based fMRI. It was shown that resting state dFC was better described by multiple FC patterns that overlap, whereas task-based dFC was better represented by sparse separated FC patterns.

While tracking the dynamic nature of FC, the existing techniques mostly assume that the timing of the various states is known, that at each time point the brain is at a discrete state, or that the current state is a weighted sum of FC-states [50]. Although these assumptions are valid for resting-state FC studies, in the case of task-based studies the FC-states are assumed to be quasi-stationary over a time interval [52, 56]. Therefore, an important extension to current approaches would be to introduce methods that can detect the time points where significant changes to network organization occur. To address this issue, Bassett et al. [57] explored the dynamic organizational changes of graphs during learning using predetermined data windows spanning multiple temporal scales. Cribben et al. [58] took a different approach and considered a data-driven technique for partitioning the time course into intervals based on the underlying FC patterns. This technique, called Dynamic Connectivity Regression (DCR), detects temporal change points in FC and estimates a connectivity graph for each time interval between pairs of change points. However, this approach relies on a greedy partitioning scheme for determining the change points and the corresponding FC states.

For the most part, existing works assume FC networks consist of different states at each time point while FC networks consist of multiple quasi-stationary time intervals. With this assumption, it is difficult to find a consistent pattern for EEG connectivity in particular during event-related potentials (ERP), electrophysiological responses that are commonly used for detecting the brain response to external stimuli, since it is assumed that FCs are different from one time sample to another. As a result, the integrated networks of brain regions corresponding to the task related activity, ERPs in particular, cannot be identified. So, there is a need for methods which assumes FC networks consists of FC states, or time intervals, and track the FC networks through identifying these FC states. Another drawback of existing methods is the vectorization of the FC matrices, which does not distinguish between nodes, time, and subject information during the analysis. Therefore, any analysis of FC networks in order to detect FC states and summarize them should be based on separate information obtained from different modes. Finally, these existing methods usually result in very dense summarizations of the time intervals since the background activity is not separated from the foreground networks before FC network tracking. So, FC state summarization should first extract the foreground network from the background activity to find the sparse FC state representation.

1.1 Overview of the Contributions

The contributions of this dissertation can be divided into three parts: change-point detection and network state representation of functional connectivity networks based on tensor analysis, hierarchical sparse common component and innovation (SCCI) recovery algorithm, and structured dictionary learning. In this thesis, FC networks are assumed to consist of multiple time intervals or FC states. The boundaries of these FC states are detected through change point detection. Moreover, the integrated functional connectivity network corresponding to different tasks are represented by network state representation. In order to obtain a sparse network state pattern which excludes the background activity, the hierarchical SCCI recovery method is used. Finally, to improve the accuracy of the SCCI recovery methods, the structured dictionary learning algorithm is proposed here which learns the dictionary from the training signals.

In Chapter 3, a tensor-based approach is introduced in order to identify the change points and summarize the detected time intervals of functional connectivity networks. The major contributions of this work can be summarized as follows:

1. The time intervals during which the functional connectivity network has a common quasi-stationary pattern across time and subjects are identified.

2. Functional connectivity networks are represented through tensors, which separates the information carried by nodes, subject and time modes. The change point detection finds the changes across the connectivity common to all subjects performing the same task.

3. The summarization through the tensor projection captures the variability common to all subjects across time.

4. The proposed algorithm is flexible for extensions to higher order representations such as time and frequency dependent network states across time.

Tensor-based approach does not necessarily separate the background activity from the foreground network. As a result, the analysis to detect change points and summarize the time intervals is based on both the background and foreground networks, which leads to false alarms in change point detection and a very dense summarization matrix. SCCI model separates the background activity from the foreground network so that the analysis is solely based on the foreground network. This reduces the number of false alarms in change point detection and finds sparse summarizations of the time intervals. In Chapter 4, a new hierarchical algorithm to recover the common and innovation parts in the the SCCI model more efficiently is proposed with the following contributions:

1. A hierarchical algorithm to recover the common sparse and innovation components in the SCCI model in a more computationally efficient manner has been developed. Bounds on the computational complexity and reconstruction error for this new algorithm are derived and compared to the original SCCI model recovery method.

2. A novel way of applying the hierarchical recovery method to separate the innovation components of the functional connectivity networks from the background network is proposed by assuming that background activity is the common component across time. The innovation components of the SCCI model are specific to each time sample and their support are tracked across time to identify the change in the subspace across time. Any change in the support of the innovation components are considered as change points.

3. The hierarchical recovery is also used to summarize the functional connectivity network within the detected time intervals. The functional connectivity networks within each time interval are considered to follow the SCCI model. The common component of the SCCI model represents the summarized functional connectivity network within the time interval. The performance of the proposed framework is evaluated to find the state representation of detected time intervals.

The SCCI and H-SCCI model recovery assumes pre-determined dictionaries for the common and innovation components. However, pre-determined dictionaries do not represent the signals and FC networks well since the signals may not be sparse with respect to predetermined dictionaries. Therefore, in Chapter 5, a structured dictionary learning approach is proposed to learn the common and innovation dictionaries simultaneously in SCCI model. The contributions of this chapter are as follows:

1. A structured dictionary learning algorithm is proposed to learn the common and innovation dictionaries simultaneously for SCCI model.

2. The structured dictionary learning algorithm is used to learn the common and innovation dictionaries of ERPs. The learned dictionaries are used to extract the ERPs from the EEG test signals of neighboring electrodes. The learned dictionaries represent ERPs better than pre-determined dictionaries, which leads to better ERP extraction from EEG signals.

CHAPTER 2

BACKGROUND

In this chapter, we will review the background on time-frequency phase synchrony measure used to construct the functional connectivity networks and tensor algebra.

2.1 Time-Frequency Phase Synchrony

Rihaczek distribution, a member of Cohen's class of time-frequency distribution, derives the signal energy in time and frequency by application of the complex signal notation unlike most members of Cohen's class which do not carry any phase information. Rihaczek distribution extends the total complex energy idea ($\int s_1(t)s_2^*(t)dt$, where $s_1(t)$ and $s_2(t)$ are two complex signals at the same frequency) to calculate the interaction energy of a signal within some frequency band centered at a frequency of interest with the signal itself at a given time restricted to an infinitesimal time interval, which leads to the time-frequency energy density function:

$$C_{Rihaczek}(t,\omega) = \frac{1}{\sqrt{2\pi}}s(t)S^*(\omega)\exp(-j\omega t), \tag{2.1}$$

where $C_{Rihaczek}(t,\omega)$ is the complex energy of a signal around time t and frequency ω. Rihaczek distribution can also be shown to be a bilinear, time and frequency shift invariant time-frequency distribution that satisfies the marginals, preserves the energy of the signal with strong time and frequency support properties, that is, the distribution is only nonzero when both the signal itself and its Fourier spectrum are nonzero.

However, the problem with Rihaczek distribution is that it includes the crossterms of multicomponent signals. Since the crossterms are located at the same time and frequency locations as the original signals, they cause biased energy and phase estimates. The kernel function Choi-Williams (CW) with $\phi(\theta,\tau) = \exp\left(\frac{-(\theta\tau)^2}{\sigma}\right)$ is used in RID-Rihaczek distribution in order to filter out the crossterms of Rihaczek distribution [59], as:

$$C_i(t,\omega) = \int\int \underbrace{\exp\left(-\frac{(\theta\tau)^2}{\sigma}\right)}_{\text{Choi-Williams kernel}} \underbrace{\exp(j\frac{\theta\tau}{2})}_{\text{Rihaczek kernel}} A_i(\theta,\tau)e^{-j(\theta t+\tau\omega)}d\tau d\theta, \tag{2.2}$$

where $A_i(\theta,\tau) = \int s_i(u+\frac{\tau}{2})s_i^*(u-\frac{\tau}{2})e^{j\theta u}du$ is the ambiguity function (correlation function) of the signal $s_i(t)$. The value of σ can be adjusted to achieve a desired trade-off between resolution and the amount of crossterms retained.

The phase-synchrony measure used in this thesis is based on RID-Rihaczek distribution. Synchrony measures the relation between the temporal structures of the signals regardless of signal amplitude. The amount of synchrony between two signals is usually quantified by estimating the instantaneous phase of the individual signals around the frequency of interest. The phase difference between the two signals (or electrodes) $s_i(t)$ and $s_j(t)$ for the kth trial is $\Phi^k_{(i,j)}(t,\omega) = |\Phi^k_i(t,\omega) - \Phi^k_j(t,\omega)|$, where $\Phi_i(t,\omega) = \arg\left[\frac{C_i(t,\omega)}{|C_i(t,\omega)|}\right]$.

However, the time-varying phase spectrum is not directly useful for measuring the synchrony between the signals. In order to quantify functional integration, the phase locking value (PLV) [21] is used to measure phase synchrony between electrodes i and j at time t and frequency ω, which is the averaged over L realizations/trials, as:

$$PLV_{(i,j)}(t,\omega) = \frac{1}{L}\left|\sum_{k=1}^{L}\exp\left(j\Phi^k_{(i,j)}(t,\omega)\right)\right|, \tag{2.3}$$

If the phase difference varies little across the trials, PLV is close to 1. PLV is a value between zero and one.

Once the pairwise synchrony values are computed at each time and frequency point, we can construct a time-varying connectivity graph $\{\mathbf{G}(t);\ \forall t \in \{t = 1, 2, \dots, T\}\}$ with

$$G_{(i,j)}(t) = \frac{1}{\Omega}\sum_{\omega=\omega_a}^{\omega_b} PLV_{(i,j)}(t,\omega), \tag{2.4}$$

where $G_{(i,j)}(t) \in [0,1]$ represents the connectivity strength between the nodes i and j within the frequency band of interest, $[\omega_a, \omega_b]$, and Ω is the number of frequency bins in that band. It is also possible to define connectivity graphs over both time and frequency by directly taking the PLV value as the entry in these graphs.

The adjacency matrix $\mathbf{G}_s(t)$ is built from EEG data set described in section 2.2 for each subject s at each time step t for each pair of electrodes i and j using Eq. 2.4. $G_{s,(i,j)}(t)$, $\forall i, j \in \{1, 2, \ldots, N\}$ indicates the strength of the connection between nodes i and j. The collection of S subjects at time t forms a 3-way tensor $\mathcal{G}(t) \in \mathbb{R}^{N \times N \times S}$, where $\mathbf{G}_s(t) = \mathcal{G}_{i_3=s}(t)$.

2.2 EEG Data

The EEG data analyzed in this thesis comes from a study containing the error-related negativity (ERN), which is a brain potential response that occurs following performance errors in a speeded reaction time task usually 25-75 ms after the response [60]. EEG data from 62-channels was collected in accordance with the 10/20 system on a Neuroscan Synamps2 system (Neuroscan, Inc.) at a sampling rate of 128 Hz. This study included 91 undergraduate students (34 male) from the University of Minnesota). Full methodological details of the recording are available in a previous paper [61]. The task was a common speeded-response letter (H/S) flanker, where error response-locked trials from each subject was utilized. 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 [62]. The CSD has fewer assumptions than many inverse transforms, attenuates volume conduction, and represents independent sources near the cortical surface [63].

The time-varying connectivity matrices in this study are constructed by computing the pairwise PLV over an average of 24 trials between 62 channels in the theta frequency band ((4,7) Hz) for all time (2 seconds corresponding to 256 time samples) and all subjects, resulting in $\mathbf{G}_s(t) \in \mathbb{R}^{62 \times 62}, t \in \{1, 2, \ldots, 256\}, s \in \{1, 2, \ldots, 91\}$.

2.3 Tensor Subspace Analysis

Tensors are the extensions of vectors and matrices to higher orders. Each dimension of a tensor is called a mode, or a way, and the number of variables in each mode determines the dimension of that mode. Let $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$ be a d-way tensor, where m_i is the dimension along the ith mode, and $x_{i_1,i_2,i_3,\ldots,i_d}$ denotes the $(i_1, i_2, i_3, \ldots, i_d)$th element of the tensor $\mathcal{X}$.

For the special case $d = 3$, x_{i_1,i_2,i_3} is the i_1th row, i_2th column, and i_3th tube of the 3-way tensor, $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ (Fig. 2.1). A subtensor is defined by fixing the indices along one or more modes. When two indices are fixed in two modes and the index varies in the other mode, the resulting vector is called a fiber. Fig. 2.1 displays column, row, and tube fibers of a 3-way tensor. Similarly, when an index is fixed in one mode and the indices vary in the two other modes, the partition is called a slice. When the i_1th row is fixed, the partition is a $m_2 \times m_3$ horizontal slice ($\mathbf{X}_{i_1..}$); if the i_2th column is fixed, the resulting partition is a $m_1 \times m_3$ vertical slice ($\mathbf{X}_{.i_2.}$); for the fixed i_3th tube, a $m_1 \times m_2$ frontal slice is obtained ($\mathbf{X}_{..i_3}$) [64]. Fig. 2.2 shows the horizontal, vertical, and frontal slices of a 3-way tensor.

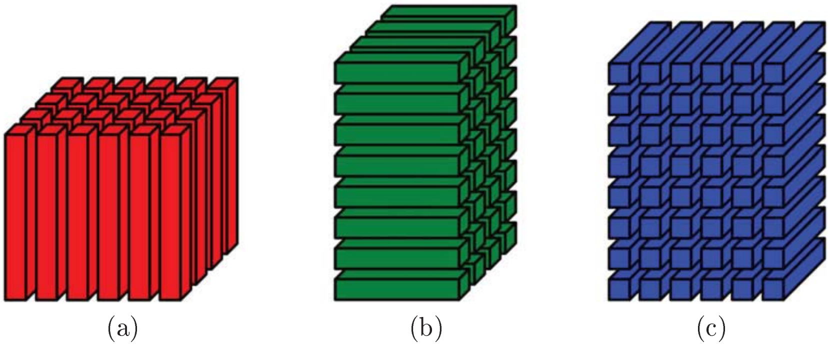


Figure 2.1 (a) Columns, (b) Rows, (c) Tubes.

A model to represent tensors consists of two parts, a structural part and a residual part. While the structural part tries to describe the underlying structure of the multiway tensor, the residual part represents the part of data which is not described by the structural part.

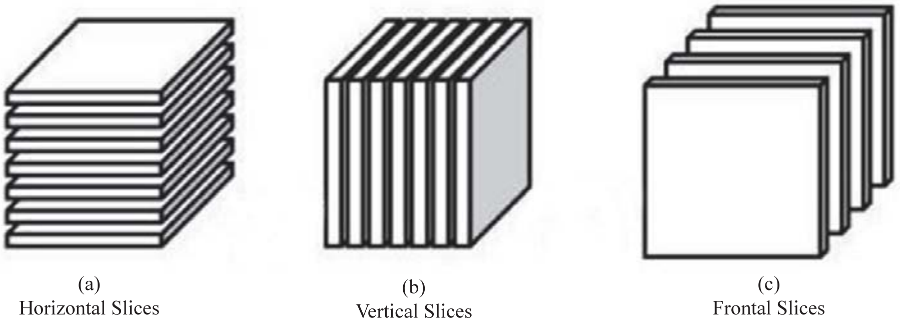


Figure 2.2 Horizontal, Vertical, and Frontal Slices of a 3-way tensor.

The residual part is analyzed for the goodness of fit. The sum of square of the residual describes the unexplained variation of the tensor in terms of least square.

Analysis methods available for 2-way tensors (matrices) need to be generalized based on the idea of factor models in order to analyze multiway tensors [64]. Linear data models such as Principal Component Analysis (PCA) [65] and Independent Component Analysis (ICA) [66] are widely used for decomposition of matrices. Depending on the criteria, different types of basis vectors are extracted and appropriate lower dimensional features are obtained through projection. Multiway data analysis extends these linear methods to capture multilinear structures and underlying correlations in higher-order datasets, also known as tensors. Several tensor decomposition methods such as Canonical Decomposition (CANDECOMP) or Parallel Factor analysis (PARAFAC), and Tucker decomposition have been proposed [64, 67, 68]. Before going on to the details of these decomposition methods, we need to provide the basic algebra for multiway tensors.

2.3.1 Basic Multiway Tensor Algebra

For the sake of simplicity, the basic tensor algebra concepts will be defined based on 3-way tensors. However, they can be generalized to higher dimensional tensors.

Unfolding, *a.k.a.* matricization or flattening, is the process of creating a single matrix by concatenating the slices of a tensor. $\mathbf{X}_{(n)}$ denotes the unfolded tensor $\mathcal{X}$ along the nth

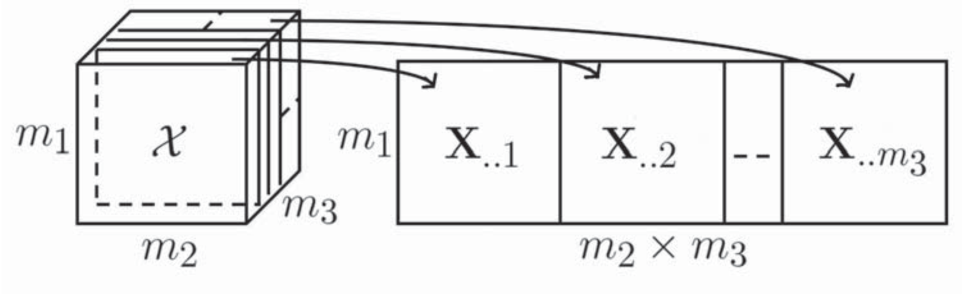


Figure 2.3 Unfolding a 3-way tensor in the first mode.

mode. Fig. 2.3 shows the unfolding process in the first mode for a 3-way tensor.

n–mode product: The n–mode product, $\times_n$, of a tensor, $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$, by a matrix $\mathbf{U} \in \mathbb{R}^{m_n \times J_n}$ is the multiplication of the unfolded tensor in the nth mode by the matrix $\mathbf{U}$, as:

$$\mathcal{X} \times_n \mathbf{U} = \mathbf{U}^{\dagger} \mathbf{X}_{(n)} = \sum_{i_n} x_{i_1, i_2, \ldots, i_n, \ldots, i_d} U_{j_n, i_n}, \tag{2.5}$$

where $\dagger$ is the matrix transpose operator.

Kronecker product: The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{m_1 \times m_2}$ and $\mathbf{B} \in \mathbb{R}^{m_3 \times m_4}$ is a $m_1 m_3 \times m_2 m_4$ matrix, defined as:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots & a_{1m_2}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots & a_{2m_2}\mathbf{B} \\ \vdots \vdots \ddots \vdots & & & \\ a_{m_1 1}\mathbf{B} & a_{m_1 1}\mathbf{B} & \ldots & a_{m_1 m_2}\mathbf{B} \end{bmatrix}. \tag{2.6}$$

Khatri–Rao product: The Khatri–Rao product of two matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m]$ is given as:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \ldots, \mathbf{a}_m \otimes \mathbf{b}_m]. \tag{2.7}$$

n–rank: The n–rank of the tensor $\mathcal{X}$ is the column rank of the matrix obtained by unfolding the tensor along its nth mode.

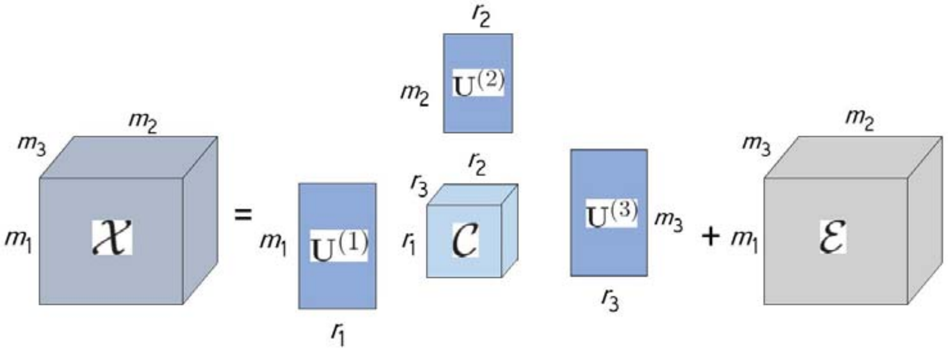


Figure 2.4 Tucker decomposition for a 3-way tensor.

2.3.2 Tucker Decomposition

The Tucker decomposition is a higher order generalization of Singular Value Decomposition (SVD) for tensors [64]. The Tucker decomposition of the d-way tensor $\mathcal{X}$ can be expressed as:

$$\begin{gathered}\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \ldots \times_d \mathbf{U}^{(d)} + \mathcal{E}, \\ \mathcal{X} = \textstyle\sum_{i_1,i_2,i_3,\ldots,i_d} \mathcal{C}_{i_1,i_2,i_3,\ldots,i_d} \left(\mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ \mathbf{u}_{i_3}^{(3)} \circ \ldots \circ \mathbf{u}_{i_d}^{(d)}\right) + \mathcal{E}_{i_1,i_2,i_3,\ldots,i_d},\end{gathered} \tag{2.8}$$

where $\mathcal{C} \in \mathbb{R}^{r_1 \times r_2 \times r_3 \times \ldots \times r_d}$ is the core tensor, $\mathcal{E} \in \mathbb{R}^{m_1 \times m_2 \times m_3 \times \ldots \times m_d}$ is the residual error, and $\times_k$ is the product of a tensor and a matrix along mode-k. $\mathbf{U}^{(1)} \in \mathbb{R}^{m_1 \times r_1}$, $\mathbf{U}^{(2)} \in \mathbb{R}^{m_2 \times r_2}$, $\ldots$, and $\mathbf{U}^{(d)} \in \mathbb{R}^{m_d \times r_d}$ are the matrix of the left singular vectors of the tensor unfolded along the ith mode, called component matrices, and $r_i \leq m_i$. Therefore, the kth singular vector along mode i, $\mathbf{u}_k^{(i)}$ is the kth column of the component matrix $\mathbf{U}^{(i)}$. Fig. 2.4 shows the Tucker decomposition for a 3-way tensor.

The tensor $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times m_3 \times \ldots \times m_d}$ is projected to another tensor $\mathcal{Y} \in \mathbb{R}^{r_1 \times r_2 \times r_3 \times \ldots \times r_d}$ using the projection matrices $\tilde{\mathbf{U}}^{(i)} \in \mathbb{R}^{m_i \times r_i}$ [69], as:

$$\mathcal{Y} = \mathcal{X} \times_1 \tilde{\mathbf{U}}^{(1)\dagger} \times_2 \tilde{\mathbf{U}}^{(2)\dagger} \times_3 \tilde{\mathbf{U}}^{(3)\dagger} \ldots \times_d \tilde{\mathbf{U}}^{(d)\dagger}, \tag{2.9}$$

where † is the matrix transpose operator, and the projection matrix $\tilde{\mathbf{U}}^{(i)}$ is formed by taking the first r_i columns of $\mathbf{U}^{(i)}$, where usually $r_i \leq m_i$.

Tucker decomposition is more flexible compared to other tensor decomposition methods such as PARAFAC thanks to its core array $\mathcal{C}$. The core array represents the interaction between factors from different modes [64]. While the Tucker decomposition cannot determine the component matrices uniquely, its component matrices are orthogonal. The orthogonality property of the Tucker decomposition makes it an appropriate decomposition for lower dimensional subspace projection.

2.3.2.1 Tucker-ALS

Various algorithms such as Alternating Least Squares [70], Simultaneous matrix diagonalizations [71], Simultaneous generalized Schur decomposition [72], etc. have been proposed to calculate these models. The most popular method for computing Tucker decomposition is Alternating Least Square (ALS) method. The extension of ALS for n-way tensors is called Higher-order Orthogonal Iteration (HOOI) [68]. The basic idea is to estimate one component matrix at a time using SVD of the unfolded tensor. For a 3-way tensor, the Tucker decomposition for an unfolded tensor in the first mode is written as $\mathbf{X}_{(1)} = \mathbf{U}^{(1)}\mathbf{C}_{(1)}(\mathbf{U}^{(2)} \otimes \mathbf{U}^{(3)})^{\dagger} + \mathbf{E}_{(1)}$. In order to calculate the Tucker decomposition, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ are randomly initialized. The left singular vector of $\mathbf{X}_{(1)}(\mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)})$ is the estimation of $\mathbf{U}^{(1)}$. Then, SVD of $\mathbf{X}_{(2)}(\mathbf{U}^{(3)} \otimes \mathbf{U}^{(1)})$ based on the estimated $\mathbf{U}^{(1)}$ is calculated to obtain $\mathbf{U}^{(2)}$. Finally, SVD of $\mathbf{X}_{(3)}(\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})$ based on estimated $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ yields an estimate for $\mathbf{U}^{(3)}$. This process is repeated until convergence.

2.3.2.2 Numerical Convex Hull based Model Selection

Tensor rank estimation problem can be addressed through model selection. The numerical convex hull algorithm has been proposed to find the best rank of the tensor along each mode [73]. It is a numerical procedure implementing the visual inspection of scree-like plots [74]. These hull solutions have the best goodness-of-fit/number-of-free-parameter balance.

Therefore, the best rank is the one whose measure of the goodness-of-fit is on or close to an elbow in the higher boundary of the convex hull of this scree-like plot.

The number of free parameters (*fp*) for Tucker model of a 3-way tensor with size (m_1, m_2, m_3) and rank (r_1, r_2, r_3) is $m_1 r_1 + m_2 r_2 + m_3 r_3 + r_1 r_2 r_3 - r_1^2 - r_2^2 - r_3^2$ [73]. The goodness-of-fit parameter is defined as:

$$f = \frac{\sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} c_{ijk}^2}{\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \sum_{k=1}^{m_3} x_{ijk}^2}. \tag{2.10}$$

To find the higher boundary of the convex hull, all possible values of goodness-of-fit and all possible number of free parameter are calculated. For each *fp*, only the largest goodness-of-fit value is retained. Then, all n retained goodness-of-fit values are sorted and labeled with s_i $(i = 1, 2, \ldots, n)$. Next, s_i is discarded if there exists s_j $(j < i)$ such that $f_j > f_i$. Starting from $i = 1$, discard s_{i+1} if its value is below the line connecting s_i to s_{i+2}. The remaining points gives the higher boundary of the convex hull of the goodness-of-fit versus the number of free parameters. The best model (rank) is the one maximizing the ratio:

$$\frac{\frac{f_i - f_{i-1}}{fp_i - fp_{i-1}}}{\frac{f_{i+1} - f_i}{fp_{i+1} - fp_i}}. \tag{2.11}$$

2.3.3 PARAFAC Decomposition

The d-way tensor $\mathcal{X} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$ is decomposed using PARAFAC with R factors or a linear combination of R rank-1 tensors as:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \ldots \circ \mathbf{a}_r^{(d)} + \mathcal{E} = \sum_{r=1}^{R} \lambda_r \left(\hat{\mathbf{a}}_r^{(1)} \circ \hat{\mathbf{a}}_r^{(2)} \circ \ldots \circ \hat{\mathbf{a}}_r^{(d)} \right) + \mathcal{E} \tag{2.12}$$

where $\mathbf{a}_r^{(i)} \in \mathbb{R}^{m_i \times 1}$ is the loading vector along the ith mode, and $\hat{\mathbf{a}}_r^{(i)} = \frac{\mathbf{a}_r^{(i)}}{\|\mathbf{a}^{(i)}{}_r\|_2}$ is the corresponding normalized loading vector. λ_r is the eigenvalue representing the weight of the

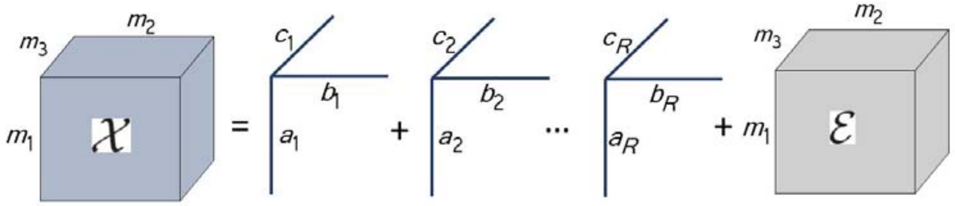


Figure 2.5 PARAFAC decomposition for a 3-way tensor.

r^{th} factor, and $\mathcal{E} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_d}$ is the residual. The loading matrices are defined as matrices whose columns are the loading vectors, $\mathbf{A}^{(i)} = [\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \ldots, \mathbf{a}_R^{(i)}]$. Fig. 2.5 shows PARAFAC decomposition for a special case $d = 3$. For a 3-way tensor, $d = 3$, three types of slices along three modes of a three-way tensor are defined as:

Horizontal Slices: The i_1^{th} horizontal slice (Fig. 2.2(a)) is described as:

$$\mathbf{X}_{i_1..} = \textstyle\sum_{r=1}^{R} a_r^{(1)}(i_1)(\mathbf{a}^{(2)}{}_r \circ \mathbf{a}^{(3)}{}_r). \tag{2.13}$$

Vertical Slices: The i_2^{th} vertical slice (Fig. 2.2(b)) is described as:

$$\mathbf{X}_{.i_2.} = \textstyle\sum_{r=1}^{R} a_r^{(2)}(i_2)(\mathbf{a}^{(1)}{}_r \circ \mathbf{a}^{(3)}{}_r). \tag{2.14}$$

Frontal Slices: The i_3^{th} frontal slice (Fig. 2.2(c)) is described as:

$$\mathbf{X}_{..i_3} = \textstyle\sum_{r=1}^{R} c_r^{(3)}(i_3)(\mathbf{a}^{(1)}{}_r \circ \mathbf{a}^{(2)}{}_r). \tag{2.15}$$

The advantage of parallel factor analysis (PARAFAC) decomposition is that it provides a unique model for the multiway array. This model imposes a restriction that the factors in different modes can only interact factorwise, which results in the same number of factors for each mode. It also does not have the rotational freedom problem. Therefore, it guarantees that the pure spectra can be obtained if the right number of components is chosen [75].

There are numerous methods to determine the appropriate number of factors, like core consistency [76], residual analysis [76], etc [77, 78]. However, these techniques cannot guarantee the optimal number of factors for real data [64].

2.3.3.1 PARAFAC-ALS

The unfolded tensor in the first mode for PARAFAC model is written as $\mathbf{X}_{(1)} = \mathbf{A}^{(1)} \left(\mathbf{A}^{(3)} \odot \mathbf{A}^{(2)}\right)^{\dagger} + \mathbf{E}_{(1)}$. In order to calculate the PARAFAC parameters, two matrices $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$ are randomly initialized. Then, the matrix $\mathbf{A}^{(1)}$ is estimated as: $\mathbf{A}^{(1)} = \mathbf{X}_{(1)}\mathbf{Z}\left(\mathbf{Z}^{\dagger}\mathbf{Z}\right)^{-1}$, where $\mathbf{Z} = \mathbf{A}^{(3)} \odot \mathbf{A}^{(2)}$. Using the estimated $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)} = \mathbf{X}_{(2)}\mathbf{Z}\left(\mathbf{Z}^{\dagger}\mathbf{Z}\right)^{-1}$, where $\mathbf{Z} = \mathbf{A}^{(3)} \odot \mathbf{A}^{(1)}$, is estimated. Based on previous estimated factors, $\mathbf{A}^{(3)}$ is estimated using $\mathbf{A}^{(3)} = \mathbf{X}_{(3)}\mathbf{Z}\left(\mathbf{Z}^{\dagger}\mathbf{Z}\right)^{-1}$, where $\mathbf{Z} = \mathbf{A}^{(2)} \odot \mathbf{A}^{(1)}$. This procedure is iteratively repeated until convergence.

CHAPTER 3

TENSOR-BASED FUNCTIONAL BRAIN CONNECTIVITY TRACKING

3.1 Introduction

In this chapter, we propose a two step approach to track whole brain dynamic connectivity. In the first step, we detect the change points in time where significant changes to the network structure occur. We quantify these changes through a low-rank tensor approximation to the FCNs constructed across subjects using subspace distance metrics. Once the change points are detected, the network state for the time intervals between change points is assumed to be stationary and summarized through tensor-matrix projections across subjects and time. The key contributions of the proposed work are three fold. First, unlike most of the current work we determine time intervals during which the FCN has a common quasi-stationary pattern across time and subjects. Second, most of the current work reduces the high dimensionality of the FCNs by vectorizing the connectivity matrices into long vectors before identifying the network states. This approach may lose some important relationships between the nodes and 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 approach is flexible for extensions to higher order representations such as time and frequency dependent network states across time.

3.2 Change Point Detection

The first step to tracking the dynamics of functional connectivity is to detect the boundaries of the of functional connectivity states. Let's consider the adjacency matrices of all subjects at time t to form a 3-way tensor $\mathcal{G}(t), t \in \{1, 2, \ldots, T\}$, then the sequence $\mathcal{G}(t)$ contains $\tau + 1$

functional connectivity states [52, 49, 79], with change points located at $\{t_1, t_2, \ldots, t_\tau\}$, where the location and the number of change points are not known *a priori*. Here, we propose two methods for determining the location of the change points: similarity-based change point detection and subspace distance based change point detection.

3.2.1 Change Point Detection Using Similarity Matrix

In this section, we introduce a change point detection algorithm based on calculating the similarity between graphs at different time steps. The time-varying functional connectivity graphs across subjects and time, $\mathcal{G}(t), t = 1, 2, \ldots, T$, can be considered as a 4-mode tensor $\mathcal{G} \in \mathbb{R}^{N \times N \times T \times S}$ constructed as channel × channel × time × subject, with N being the number of channels, T the total number of time points and S the number of subjects. The Tucker decomposition of this connectivity tensor yields:

$$\mathcal{G} = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)} + \mathcal{E}, \tag{3.1}$$

where $\mathcal{C} \in \mathbb{R}^{N \times N \times T \times S}$ is the core tensor, and $\mathcal{E} \in \mathbb{R}^{N \times N \times T \times S}$ is the residual error.

To obtain an approximation of $\mathcal{G}$, $\hat{\mathcal{G}} \in \mathbb{R}^{N \times N \times T \times S}$, we first consider the full Tucker decomposition with $\mathbf{U}^{(1)} \in \mathbb{R}^{N \times N}$, $\mathbf{U}^{(2)} \in \mathbb{R}^{N \times N}$, $\mathbf{U}^{(3)} \in \mathbb{R}^{T \times T}$, and $\mathbf{U}^{(4)} \in \mathbb{R}^{S \times S}$. The singular values along each mode are ordered by fixing the index of all of the other modes to 1. Since the first singular values along each mode represent the largest variance of the data along that mode, we choose the first singular value to order the remaining modes. To get the approximation tensor, the appropriate number of singular vectors along first and second modes $\hat{N}$ is defined as $\hat{N} = j_k$, where j_k is the highest index for which $\left|\mathcal{C}_{j_k,1,1,1}\right| \geq 0$. Similarly, the number of singular vectors along the fourth mode is $\hat{S} = s_k$, where s_k is the highest index for which $\left|\mathcal{C}_{1,1,1,s_k}\right| \geq 0$. The time mode is not projected to a lower dimensional space since all time points are necessary to identify the exact boundaries of the network states. The lower dimensional projection matrices are defined as: $\hat{\mathbf{U}}^{(1)} = [\mathbf{u}_1^{(1)} \mathbf{u}_2^{(1)} \ldots \mathbf{u}_{\hat{N}}^{(1)}]$,

$\hat{\mathbf{U}}^{(2)} = [\mathbf{u}_1^{(2)}\mathbf{u}_2^{(2)}...\mathbf{u}_{\hat{N}}^{(2)}]$, $\hat{\mathbf{U}}^{(4)} = [\mathbf{u}_1^{(4)}\mathbf{u}_2^{(4)}...\mathbf{u}_{\hat{S}}^{(4)}]$.

The reconstructed tensor $\hat{\mathcal{G}} \in \mathbb{R}^{N\times N\times T\times S}$ is obtained as:

$$\hat{\mathcal{G}} = \mathcal{G} \times_1 \left(\hat{\mathbf{U}}^{(1)}\hat{\mathbf{U}}^{(1)^\dagger}\right) \times_2 \left(\hat{\mathbf{U}}^{(2)}\hat{\mathbf{U}}^{(2)^\dagger}\right) \times_3 \left(\hat{\mathbf{U}}^{(3)}\hat{\mathbf{U}}^{(3)^\dagger}\right) \times_4 \left(\hat{\mathbf{U}}^{(4)}\hat{\mathbf{U}}^{(4)^\dagger}\right). \tag{3.2}$$

The 4-mode approximation tensor $\hat{\mathcal{G}}$ can be written as a sequence of 3-mode tensors $\hat{\mathcal{G}}(t); t \in \{1, 2, \dots, T\}$. To detect the boundaries of network states, we propose a new temporal clustering algorithm. Unlike regular data clustering, the proposed method considers both the similarity of the lower dimensional representation of the networks as well as their closeness in time. The similarity of two networks at time t_1 and t_2 is quantified through a cosine similarity metric between $\hat{\mathcal{G}}(t_1)$ and $\hat{\mathcal{G}}(t_2)$ as follows:

$$\Delta(t_1, t_2) = \frac{\left\langle \hat{\mathcal{G}}(t_1), \hat{\mathcal{G}}(t_2) \right\rangle}{\|\hat{\mathcal{G}}(t_1)\| \|\hat{\mathcal{G}}(t_2)\|}; t_1, t_2 \in \{1, 2, \dots, T\}, \tag{3.3}$$

where $\langle a, b\rangle$ is the inner product of a and b, and $\| a \|$ is the Frobenius Norm. Similarly, the temporal closeness between two graphs is quantified as $\Theta(t_1, t_2) = e^{\frac{-(t_1-t_2)^2}{2\sigma^2}}; t_1, t_2 \in \{1, 2, \dots, T\}$, where σ is a parameter which determines the weighting for different time separations, and depends on the sampling frequency.

The combined similarity matrix is defined as:

$$\Psi(t_1, t_2) = \lambda\Theta(t_1, t_2) + (1 - \lambda)\Delta(t_1, t_2); t_1, t_2 \in \{1, 2, \dots, T\}, \tag{3.4}$$

where $\lambda \in (0, 1)$ determines the trade-off between tensor similarity and time proximity. This similarity matrix is input to a standard spectral clustering algorithm combined with k-means to identify the boundaries of the network states [80].

3.2.2 Change Point Detection Using Subspace Distance

In the second approach to change point detection, in order to detect the boundaries of functional connectivity states $t_k, \forall k \in \{1, 2, \dots, \tau\}$, we calculate the distance between the lower dimensional subspaces that best approximate the connectivity graphs along the three modes of $\mathcal{G}(t)$ for $t = 1, 2, \ldots, T$. This approach is motivated by the fact that most real data lies in a low dimensional subpace or an union of subspaces [81].

In order to find the change points which are consistent among all subjects, we use a bootstrapping approach as follows [82]. $\bar{S}$ subjects are chosen randomly without replacement. The optimal low rank of the tensor $\mathcal{G}(t) \in \mathbb{R}^{N \times N \times \bar{S}}$, $(\hat{n}(t), \hat{n}(t), \hat{s}(t))$, for the selected subjects is obtained by the convex hull algorithm. The convex hull algorithm evaluates the goodness of fit between the original tensor and its low rank approximation for different combinations of ranks and selects the optimal value numerically [73]. Since the adjacency matrix $\mathbf{G}_s(t)$ is symmetric, the first and second modes are forced to have the same rank $\hat{n}(t)$ [83, 84].

Given the ranks of the tensor along each mode, $(\hat{n}(t), \hat{n}(t), \hat{s}(t))$, we used higher-order orthogonal iteration (HOOI) to compute a low-rank approximation of $\mathcal{G}(t)$, denoted by $\hat{\mathcal{G}}(t)$, with rank $(\hat{n}(t), \hat{n}(t), \hat{s}(t))$ [68]. The 3-way tensor $\hat{\mathcal{G}}(t)$ at time step t is decomposed using Tucker decomposition as follows:

$$\hat{\mathcal{G}}(t) = \mathcal{C}(t) \times_1 \mathbf{U}^{(1)}(t) \times_2 \mathbf{U}^{(2)}(t) \times_3 \mathbf{U}^{(3)}(t), \tag{3.5}$$

where $\mathcal{C}(t) \in \mathbb{R}^{\hat{n}(t) \times \hat{n}(t) \times \hat{s}(t)}$ is the core tensor, and $\mathbf{U}^{(1)} \in \mathbb{R}^{N \times \hat{n}(t)}$, $\mathbf{U}^{(2)} \in \mathbb{R}^{N \times \hat{n}(t)}$, and $\mathbf{U}^{(3)} \in \mathbb{R}^{\bar{S} \times \hat{s}(t)}$ are the matrix of the left singular vectors. The bases $\mathbf{u}_i^{(1)}(t)$, $i \in \{1, 2, \ldots, \hat{n}(t)\}$, $\mathbf{u}_j^{(2)}(t)$; $j \in \{1, 2, \ldots, \hat{n}(t)\}$, and $\mathbf{u}_k^{(3)}(t)$, $k \in \{1, 2, \ldots, \hat{s}(t)\}$ span the lower dimensional subspaces. Note that $\mathbf{U}^{(1)}(t) = \mathbf{U}^{(2)}(t)$ in the case of symmetric connectivity.

Once $\mathbf{U}^{(1)}(t)\, \forall t \in \{1, 2, \ldots, T\}$ are determined, the distance between subspaces of the first mode of $\mathcal{G}(t)$, $\forall t \in \{1, 2, \ldots, T\}$ need to be calculated as we are interested in detecting the changes along the connectivity mode. The subspace distance is calculated between consecutive time points using $\mathbf{U}^{(1)}(t)$. Since subspaces of the first mode of $\mathcal{G}(t)$ and $\mathcal{G}(t+1)$ may be of different dimensions $\hat{n}(t)$ and $\hat{n}(t+1)$, ordinary subspace distance measures such as Euclidean distance cannot be used. Thus, we quantify the subspace distance using the

Grassmann distance [85] between $(\mathcal{G}(t), \mathcal{G}(t+1))$, $\forall t \in \{1, 2, \ldots, T-1\}$ along the first mode, as follows:

$$d(t) = \frac{1}{\sqrt{r}} \sqrt{\sum_{k=1}^{r} \theta_k^2 \left(\mathbf{U}^{(1)}(t), \mathbf{U}^{(1)}(t+1)\right)}, \tag{3.6}$$

where $r = \min(\hat{n}(t), \hat{n}(t+1))$, and $\theta_k \left(\mathbf{U}^{(1)}(t), \mathbf{U}^{(1)}(t+1)\right)$ is the kth principal angle between subspaces of the first mode of $\mathcal{G}(t)$ and $\mathcal{G}(t+1)$ [85].

In order to decide whether $d(t)$ is a change point, the subspace distance, $d(t)$, needs to be converted into a binary value, $v(t)$, through thresholding, as follows:

$$\begin{gathered} v(t) = \begin{cases} 1 & \text{if } d(t) \geq \mu_d + 2\sigma_d \\ 0 & \text{if } d(t) < \mu_d + 2\sigma_d \end{cases}, \\ \mu_d = \frac{1}{T-1} \sum_{t=1}^{T} d(t), \\ \sigma_d = \sqrt{\frac{1}{T-2} \sum_{t=1}^{T} (d(t) - \mu_d)^2}. \end{gathered} \tag{3.7}$$

where the threshold is selected as $\mu + 2\sigma$ since that corresponds to the 95% significance level based on the observation that the distances are distributed as Gaussian. This process is repeated B times for different random samplings across the subject mode in order to find change points which are consistent among all different random samplings. Thus, the binary time series, $v(t)$, across B samplings are summed up. The final set of change points are the local maxima (peaks) of this sum. Fig. 3.1 illustrates the steps of change-point detection algorithm.

3.3 State Representation

After identifying the time boundaries of the functional connectivity states, the goal is to summarize the functional connectivity patterns across time and subjects within the time interval to derive the state connectome. Previously, this was commonly addressed by averaging the edges over the time interval [52, 86]. This method has the drawback of emphasizing all of the edges equally and resulting in very dense network representations.

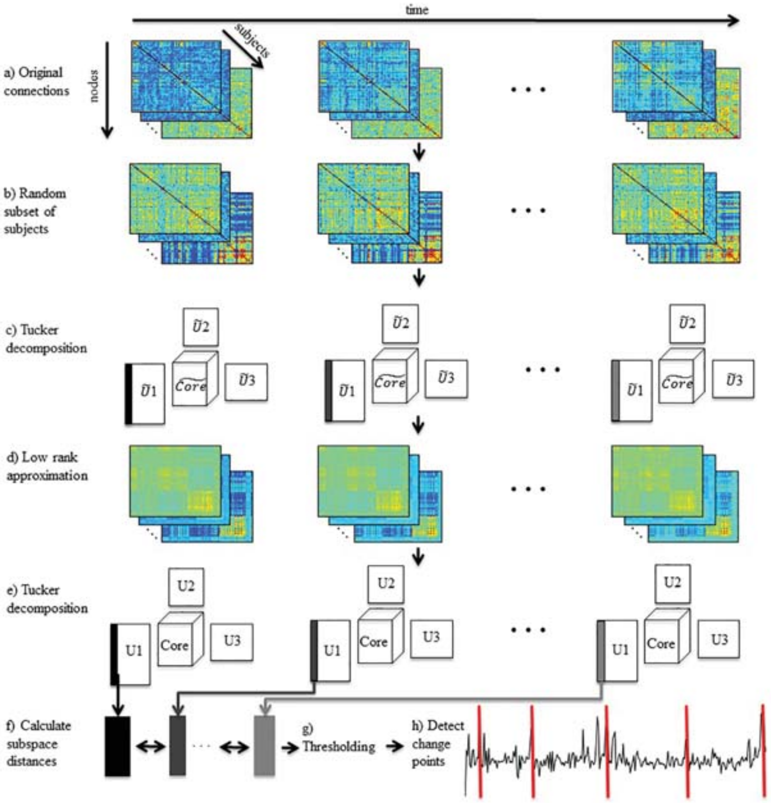


Figure 3.1 A flowchart of the change-point detection algorithm described in subsection 3.2.2, (a) The original FC tensors formed across all subjects at each time point, (b) A random subset of subjects is selected for bootstrapping, (c) Full Tucker decomposition of FC tensors corresponding to the selected subjects is performed to determine the best low-rank approximation along each mode, (d) Low-rank approximation of FC tensors using convex hull approximation, (e) The Tucker decomposition of the low-rank approximation obtained from (d), (f) Subspace distance calculation between consecutive time steps using the basis elements (columns of) U_1, (g) The subspace distances thresholding, (h) Change point detection.

The functional connectivity states are assumed to be quasi-stationary, which means that the corresponding connectivity patterns do not change significantly within the network states. Therefore, the entire network state can be represented by a single adjacency matrix demonstrating the most significant interactions among various brain regions.

In this section, two approaches based on the tensor-matrix projection are presented to summarize the functional connectivity states, projection on a single singular vector, and projection on a set of singular vectors.

3.3.1 Projection on a Single Singular Vector

For a given time interval (t_k, t_{k+1}) and the 3-mode tensor sequence corresponding to this interval $\mathcal{G}(t_k), \mathcal{G}(t_k+1), \ldots, \mathcal{G}(t_{k+1})$, the goal is to extract the topographic map $\Theta \in \mathbb{R}^{N \times N}$ which best represents that network state. The 3-mode tensors corresponding to the time interval (t_k, t_{k+1}) can be rewritten as a 4-mode tensor by taking the time modality into account $\mathcal{G}' \in \mathbb{R}^{N \times N \times (t_{k+1}-t_k+1) \times S}$, and decomposed using the full Tucker decomposition, $\mathbf{U'}^{(1)} \in \mathbb{R}^{N \times N}, \mathbf{U'}^{(2)} \in \mathbb{R}^{N \times N}, \mathbf{U'}^{(3)} \in \mathbb{R}^{(t_{k+1}-t_k+1) \times (t_{k+1}-t_k+1)}, \mathbf{U'}^{(4)} \in \mathbb{R}^{S \times S}$, similar to Equation 3.1.

In order to summarize the subject information to find a general unique model which fits all subjects, the 4-mode tensor $\mathcal{G}' \in \mathbb{R}^{N \times N \times (t_{k+1}-t_k+1) \times S}$ is projected by the singular vector $\mathbf{u}_l'^{(4)}$ corresponding to the l^{th} largest singular value in this mode. Likewise, to summarize the time information of the resulting 3-mode tensor, it is projected to the singular vector $\mathbf{u}_k'^{(3)}$ corresponding to the k^{th} largest singular value of the time mode, $\eta = \mathcal{G}' \times_3 \mathbf{u}_k'^{(3)} \times_4 \mathbf{u}_l'^{(4)}$. The values of k and l are usually equal to 1 but may change depending on the data.

3.3.2 Projection on a set of Singular Vectors

Since all subjects perform the same task at the same time, we assume that the connectivity networks across subjects are consistent for a given time interval. Given the time interval (t_k, t_{k+1}), all 3-way tensors $\mathcal{G}(t)$ in this time interval are individually fully decomposed using Tucker decomposition as :

$$\mathcal{G}(t) = \mathcal{C}(t) \times_1 \mathbf{U}^{(1)}(t) \times_2 \mathbf{U}^{(2)}(t) \times_3 \mathbf{U}^{(3)}(t), \tag{3.8}$$

where $\mathcal{C}(t)$ is the core tensor, and $\|\mathcal{C}_{i_l=1}(t)\|^2 \geq \|\mathcal{C}_{i_l=2}(t)\|^2 \geq \cdots \geq \|\mathcal{C}_{i_l=N}(t)\|^2$ for $l = 1$ and 2, and $\|\mathcal{C}_{i_3=1}(t)\|^2 \geq \|\mathcal{C}_{i_3=2}(t)\|^2 \geq \cdots \geq \|\mathcal{C}_{i_3=S}(t)\|^2$ are the singular values along the

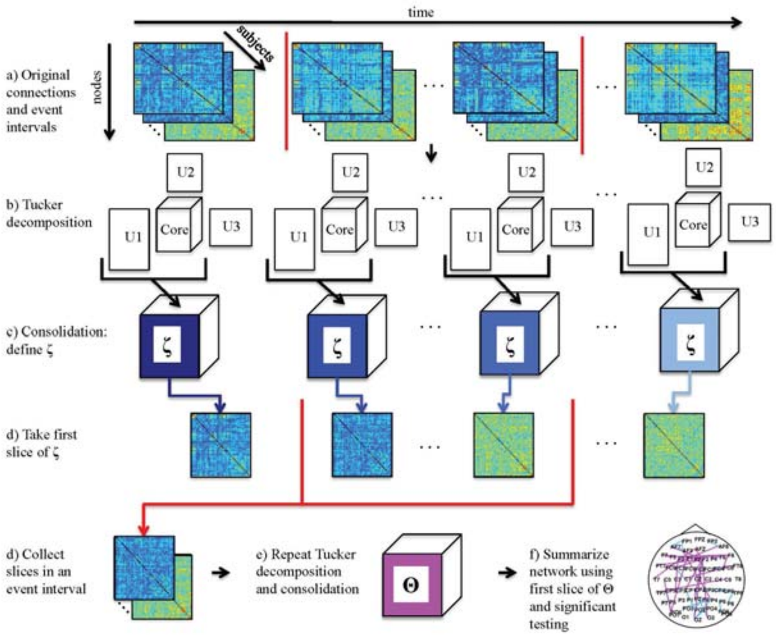

time
subjects
nodes
a) Original connections and event intervals
b) Tucker decomposition
U2
U1
Core
U3
c) Consolidation: define ζ
ζ
d) Take first slice of ζ
d) Collect slices in an event interval
e) Repeat Tucker decomposition and consolidation
Θ
f) Summarize network using first slice of Θ and significant testing

Figure 3.2 Functional Connectivity State Summarization algorithm flowchart described in subsection 3.3.2.

first, second, and third mode, respectively.

We define $\zeta(t) = \mathcal{C}(t) \times_1 \mathbf{U}^{(1)}(t) \times_2 \mathbf{U}^{(2)}(t)$, which leads to an alternative expression of Tucker decomposition as $\mathcal{G}(t) = \zeta(t) \times_3 \mathbf{U}^{(3)}(t)$. Due to the orthogonality of $\mathbf{U}^{(1)}(t)$ and $\mathbf{U}^{(2)}(t)$, we can conclude from $\|\mathcal{C}_{i_3=1}(t)\|^2 \geq \|\mathcal{C}_{i_3=2}(t)\|^2 \geq \cdots \geq \|\mathcal{C}_{i_3=S}(t)\|^2$ that $\|\zeta_{i_3=1}(t)\|^2 \geq \|\zeta_{i_3=2}(t)\|^2 \geq \cdots \geq \|\zeta_{i_3=S}(t)\|^2$ [87]. Thus, the subtensor

$$\theta(t) = \zeta_{i_3=1}(t) = \sum_{s=1}^{S} U_{s,1}^{(3)}(t)\mathbf{G}_s(t), \tag{3.9}$$

where $\theta(t) \in \mathbb{R}^{N\times N}$, is the factor that captures most of the energy of the activation patterns across subjects at time t. Similarly, $\zeta_{i_3=2}(t)$ captures the second largest amount of energy across all subjects at time t [87, 88]. The subtensor $\theta(t)$ is the common functional connectivity pattern across all subjects at the time step t.

3.3.2.1 Time Summarization

After the connectivity information across all subjects is summarized and represented by a single adjacency matrix, the network state $\theta(t)$, $\forall t \in \{t_k, t_k + 1, \cdots, t_{k+1}\}$ needs to be summarized across time mode within the detected functional connectivity state boundaries in order to derive the state connectome.

First, we construct a 3-way tensor $\Theta \in \mathbb{R}^{N \times N \times (t_{k+1} - t_k + 1)}$ from the set of matrices $\theta(t) \in \mathbb{R}^{N \times N}$, where the third mode is time. The tensor Θ is fully decomposed using Tucker decomposition as follows:

$$\Theta = \vartheta \times_1 \bar{\mathbf{U}}^{(1)} \times_2 \bar{\mathbf{U}}^{(2)} \times_3 \bar{\mathbf{U}}^{(3)} = \bar{\zeta} \times_3 \bar{\mathbf{U}}^{(3)}. \tag{3.10}$$

The subtensor $\eta = \bar{\zeta}_{i_3=1} = \sum_{t=t_k}^{t_{k+1}} \bar{U}_{t,1}^{(3)} \Theta_{i_3=t}$, where $\eta \in \mathbb{R}^{N \times N}$, captures the largest amount of energy across all time steps. Thus, it can be used as the common functional connectivity state across all subjects and time steps within the time interval (t_k, t_{k+1}).

3.3.3 Significance Testing

The subtensor η represents the summarization of functional connectivity across subjects and time within the time interval (t_k, t_{k+1}). In order to characterize the FC within a state more concisely, we need to determine the significant edges of η. Thus, a hypothesis testing on η obtained for each time interval is required in order to identify the significant edges. Since the histogram of the edge values in η appear to be bell shape and symmetric, we assume a Gaussian distribution for the edge values in η (see Fig. 3.10). This assumption can be validated using Kolmogorov–Smirnov test [89, 90], where the null hypothesis is that the edge values in η are drawn from a Gaussian distribution family. A z-test can then be used on the edges of η to determine the most significant edges. Fig. 3.2 shows an overview of the proposed summarization algorithm.

3.3.4 Computational Complexity

The computational complexity of the change point detection algorithm depends mostly on the complexity of HOOI since HOOI requires the decomposition of the tensor with various ranks in order to find the optimal low rank. Let's assume that HOOI decomposes the tensor with I different sets of ranks. The complexity of each decomposition depends on the selected rank. The full rank decomposition has the highest computational complexity, which is equal to $\mathcal{O}\left(\left(\frac{N+N+S}{3}\right)^4\right)$ [91]. Assuming full rank decomposition for all HOOI iterations (the worst case), the computational complexity of HOOI is at most equal to $\mathcal{O}\left(I\left(\frac{N+N+S}{3}\right)^4\right)$. Even though HOOI procedure is performed for every time step, HOOI procedure for all time steps can be computed in parallel since HOOI for each time step is independent of another. Similarly, the computational complexity of the full Tucker decomposition and the $(\hat{n}, \hat{n}, \hat{s})$-rank decompositions are $\mathcal{O}\left(\left(\frac{N+N+S}{3}\right)^4\right)$ and $\mathcal{O}\left(\left(\frac{\hat{n}+\hat{n}+\hat{s}}{3}\right)^4\right)$, respectively. Thus the total computational complexity of the change point detection algorithm step is $\mathcal{O}\left((I+1)\left(\frac{N+N+S}{3}\right)^4 + \left(\frac{\hat{n}+\hat{n}+\hat{s}}{3}\right)^4\right)$.

The computational complexity of FC state representation consists of two parts, computational complexity of summarization across subjects and time. Each summarization step includes the computational complexity of the Tucker decomposition and that of the projection step. The computational complexity of the multiplication of a matrix with size $d_1 \times d_2$ by a vector with size $d_2 \times 1$ is $\mathcal{O}\left(d_1 d_2\right)$ [92]. The computational complexity of Tucker decomposition across subjects at each time step is $\mathcal{O}\left(S\left(\frac{N+N+S}{3}\right)^4\right)$, and the computational complexity of the projection of subjects onto a single matrix at each time step is $\mathcal{O}\left(S^2 N^2\right)$. The total computational complexity for the subject summarization step is $\mathcal{O}\left(\left(\frac{N+N+S}{3}\right)^4 + S^2 N^2\right)$ since the summarization of all subjects are independent and can be done in parallel. Similarly, the total computational complexity of summarization across time is $\mathcal{O}\left(\left(\frac{N+N+\tau}{3}\right)^4 + \tau^2 N^2\right)$, where $\tau = t_{k+1} - t_k + 1$. Therefore, the total computational complexity of the summarization step is

$$\mathcal{O}\left(\left(\frac{N+N+S}{3}\right)^4+\left(\frac{N+N+\tau}{3}\right)^4+(\tau^2+S^2)N^2\right).$$

3.4 Experimental Results For Simulated Data

In this section, the results of the proposed method on simulated networks is provided to evaluate the performance of the proposed methods.

We generate a simulated dynamic network that resembles the data structure of our EEG FCNs to evaluate the effectiveness of the proposed approach. A tensor corresponding to a dynamic network with $N = 64$ nodes, $S = 32$ subjects, and $T = 128$ time steps is generated. The change points are fixed at $t_1 = 41$ and $t_2 = 81$.

Within the time interval $[1, 40]$, all graphs at each time point and across all subjects are generated with a two cluster structure. The first cluster consists of nodes $(1 - 48)$, while nodes $(49 - 64)$ belong to the second cluster. Intra-cluster edges have higher mean edge value and lower variance compared to inter-cluster edges. The weights of the intra-cluster edges are selected randomly from $\mathcal{N}(0.2, 0.01)$ while the weights of the inter-cluster edges are chosen randomly from $\mathcal{N}(0.1, 0.02)$. Out of 1128 edges within the first cluster, 500 significant edges are selected randomly using R-MAT algorithm [93]. In order for the selected edges to be significantly important, the weights are selected from $\mathcal{N}(\mu, 0.01)$, where $\mu \in (0.2, 0.6)$ and added to the intra-cluster edge values. Similarly, 50 out of 120 edges from the second cluster are chosen randomly using R-MAT algorithm, and their weights are determined by adding random values from $\mathcal{N}(\mu, 0.01)$ with the same μ value to the intra-cluster edges.

For the time interval $[41, 80]$, nodes $(1-16)$ at each time step and for all subjects belong to the first cluster while nodes $(17-64)$ belong to the second cluster. The weights of all intra-cluster edges are selected randomly from $\mathcal{N}(0.2, 0.01)$ while the weights of the inter-cluster edges are generated from $\mathcal{N}(0.1, 0.02)$. 50 significant edges within the first cluster and 500 significant edges within the second cluster are selected randomly using R-MAT algorithm.

The weights of the significant edges for all subjects at each time step are formed by selecting random values from the distribution $\mathcal{N}(\mu, 0.01)$, where $\mu \in (0.2, 0.6)$, and adding them to the intra-cluster edges. Graphs for all subjects at each time step within the time interval $[81, 128]$ are generated similar to the graphs within the time interval $[1, 40]$.

To evaluate the performance of the proposed algorithm under different conditions, 5 experiments with μ values in the range $(0.2, 0.6)$ are run. Each experiment is repeated 400 times. The signal to noise ratio (SNR) is calculated for different values of μ, where signal power is defined as the total power of the significant edges ($\mathcal{N}(\mu, 0.01)$), and the noise power is defined as the total power of inter and intra-cluster edges. Let $n_{s,(i,j)}(t)$ be the background connectivity strength of subject s between nodes i and j at time t, and $n_{s,(i,j)}(t) + f_{s,(i,j)}(t)$ be the connectivity of the subject s between nodes i and j at time t, where $f_{s,(i,j)}(t) \sim \mathcal{N}(\mu, 0.01)$. Then, SNR is defined as

$$SNR = \frac{\frac{1}{STN^2}\sum_{t=1}^{T}\sum_{s=1}^{S}\sum_{i,j=1}^{N} f_{s,(i,j)}^2(t)}{\frac{1}{STN^2}\sum_{t=1}^{T}\sum_{s=1}^{S}\sum_{i,j=1}^{N} n_{s,(i,j)}^2(t)},$$

where S is the total number of subjects, T is the total number of time steps, and N is the number nodes.

3.4.1 Simulated Network Change Point Detection Using Subspace Distance

The change point detection algorithm is run on all 400 simulations of this dynamic network model. The binary vector, $v(t)$, across 400 simulations are averaged for each μ. The peaks of the average subspace distance for each μ indicates the change points as seen in Fig. 3.3.

3.4.2 Simulated Network Functional Connectivity State Projection on a set of Singular Vectors

Each of the three time intervals that were detected by the subspace distance measure were summarized individually across 400 simulations. z-test with a p-value of 0.95 is performed on

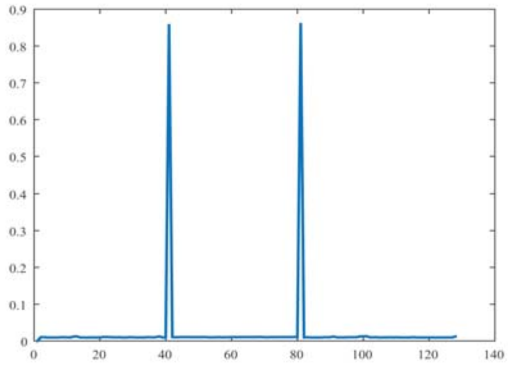


Figure 3.3 The average subspace distance of the simulated network for $\mu = 0.90$ with two peaks at $t = 41$ and $t = 81$.

Table 3.1 The error probability for simulated dynamic networks described in Section V-A

SNR	-10.63dB	-7.13dB	-4.66dB	-2.71dB	-1.11dB
Type I (Miss)	32.8%	17%	10.8%	10.2%	9.9%
Type II (False Alarm)	3.53%	2.41%	1.64%	1.46%	1.37%

the summarized graphs. Since the weights of the edges are not recovered during summarization, there is no ground-truth to compare η to. Therefore, we compare only the location of the detected significant edges within each time interval, and calculate the probability of type I (false positive) and II (false negative) errors averaged across each experimental condition, (μ, σ), i.e. different SNRs. The probability of type I refers to the number of those significant edges which are not detected to the total number of significant edges. The probability of type II is the number of detected edges which belong to the background to the total number of normal edges. Table 3.1 summarizes the results for each SNR value. It can be seen that as SNR increases, i.e. the strength of the significant edges, the probability of both type I and II errors decreases as expected since the significant edges are easier to detect with large values of SNR. In practice, most FCNs are expected to have a SNR in the range of 0dB and as such will result in Type I and II errors below 10%.

3.5 Experimental Results for Real Data

In this section, the results of the proposed method on EEG functional connectivity networks is provided to evaluate the performance of the proposed methods. The time-varying connectivity matrices for EEG data in this study are constructed by computing the pairwise PLV over an average of 24 trials between 62 channels in the theta frequency band for all time (2 seconds corresponding to 256 time samples) and all subjects using Eq. 2.3, resulting in $\mathcal{G}(t) \in \mathbb{R}^{62\times62\times91}$ for the error response.

3.5.1 Functional Connectivity Change Point Detection

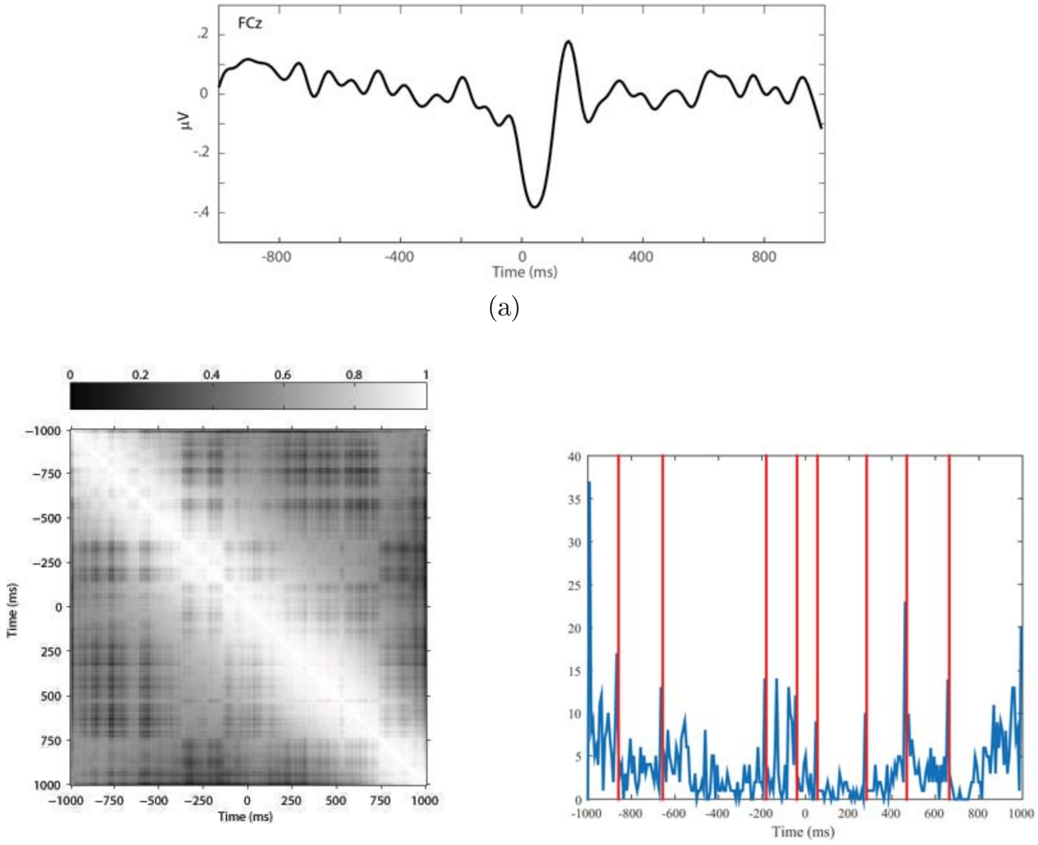
The two proposed change point detection algorithms are applied to $\mathcal{G}(t), t \in \{1, 2, \ldots, 256\}$.

3.5.1.1 Change Point Detection Using Similarity Matrix

In order to detect the change points, the time-varying graphs, $\mathbf{G}_s(t), t \in \{1, 2, \ldots, 256\}, s \in \{1, 2, \ldots, S\}$, for all subjects and all time will be treated as a 4-mode tensor $\mathcal{G}$, which is decomposed using Tucker decomposition (Eq. 3.1). The approximation tensors $\hat{\mathcal{G}}(t); t \in \{1, 2, \ldots, 256\}$ with $\hat{N} = 2, \hat{S} = 3$ are used to obtain the 256×256 similarity matrix Ψ. The matrix Ψ is computed with $\lambda = 0.4$ and $\sigma = 2500$ as shown in Fig. 3.4(b). The values of λ and σ are empirically chosen to obtain the best separation between clusters.

Once the matrix $\mathbf{\Psi}$ is obtained, the critical time points are detected using the spectral clustering with $K = 5$. The number of clusters K is selected based on the eigenspectrum of the similarity matrix [94]. The detected change points are at $\{-703, -375, -132, 188, 736\}$ ms.

The time interval $(-132, 188)$ms corresponds to ERN and Pre-ERN time intervals while the time interval $(188, 736)$ms is Pe. As it appears in Fig. 3.4(a), the positive peak of ERN in the waveform is close to 188ms, which is in agreement with the end point of the detected ERN time interval. This method requires λ and σ to be determined empirically.



(a)

(b)

(c)

Figure 3.4 (a)Waveform for error trials across all subjects, 0 ms marks the time of the button-press incorrect response, (b) Similarity matrix, Ψ, computed for 2 seconds of EEG data across 91 subjects in the theta frequency band, (c) The binary vector resulted from the subspace distance between the projection operators at consecutive time points, calculated using the Grassman distance, where red lines mark the identified change points.

3.5.1.2 Change Point Detection Using Subspace Distance

In order to detect the change points using the subspace distance approach, we selected $\bar{S} = 40$ subjects out of 91 subjects randomly without replacement to calculate the subspace distance between consecutive time points and detect change points. This procedure is repeated 106 times. Fig. 3.4 (c) shows number of change points at each time point across 106 random samplings. By choosing the peaks of this time series, change points are detected to be at $\{-859, -656, -180, -39, 55, 281, 469, 664\}$ms. As apparent in Fig. 3.4 (c), the $(-39, 55)$ms time window corresponds most directly to the ERN, while the two periods after are associated with post-ERN activity, including the Pe. Right before ERN time interval, $(-39, 55)$ms, the functional connectivity network is not stable due to the differences in response time across subjects. This leads to the observation of multiple peaks right before the 0 time point which we do not consider as valid change points. This method does not require an empirical selection of parameters compared to the method using the similarity matrix. However, it gives multiple change points close to each other right before ERN. The reason why several change points close to each others are detected is that the participants expect to receive the response right before ERN and their brains are preparing to react to the stimuli. Thus, the functional connectivity is transiting from the previous state to the new ERN state. Since these change points are due to the transition from one state to the next, there is no valid time intervals between them and they can be ignored.

In order to show the effectiveness of tensor-based methods compared to the traditional matrix-based methods, the upper triangle part of $\mathbf{G}_s(t)$, $s \in \{1, 2, \ldots, S\}$ is vectorized and concatenated across subjects to form a matrix. This process is repeated for all time steps so that T subject-connectivity matrices of size 1953×91 are obtained similar to [53]. SVD is performed on the matrices at each time step similar to the tensor approach without subject sampling. An optimal rank at each time point is determined using the convex hull algorithm. Given the rank at each time, the Grassmann distance between consecutive time steps is computed. Fig. 3.5 shows the Grassman distance between consecutive time steps based on SVD decomposition of the vectorized adjacency matrices, and based on the tensor projection method [1]. In the graph corresponding to SVD method, a few change points are obvious, among them, the one close to zero corresponds to the beginning of ERN. However, other change points such as the end of ERN are not detectable. Although the scales of SVD and tensor-based methods are different, it can be seen that the tensor decomposition is more sensitive to the changes in the network structure with more apparent change points before

[1] Note that in this experiment random sampling of subjects is not performed, and these are distances not the binary vectors. That is why the result given here for the tensor projection is different from Fig. 3.13

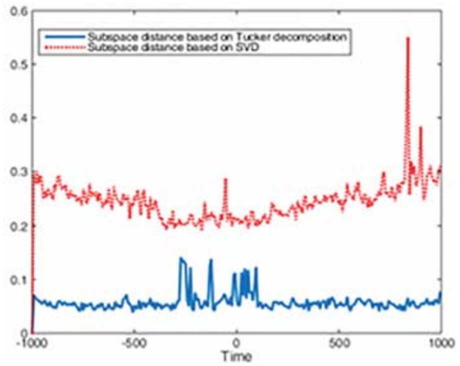


Figure 3.5 The Grassmann distance between consecutive time steps of the functional connectivity network calculated based on SVD and tensor projection.

and after the response. Both methods yield similar peaks around ERN which verifies the accuracy of the proposed tensor based method.

3.5.2 Functional Connectivity State Representation

Once the change points are determined, tensor projection along subject and time modes for all time intervals is performed to obtain the FC-states. However, two different change-point detection methods give different time intervals. In order to compare the proposed summarization methods, they are implemented on predetermined time intervals $(-100, 0)$ms, $(0, 100)$ms, and $(100, 400)$ms.

3.5.2.1 Projection on Single Singular Vectors

To obtain a single network representation using one singular vector, we first selected $k = l = 1$ the singular vectors corresponding to the highest singular values, and a Gaussian distribution is fitted to the projected connections, η to perform a z-test with $p = 0.95$. As it is shown in Fig. 3.7, most connections are between neighboring nodes since the projection to the subspace spanned by the largest singular value mostly contains edges between physically adjacent nodes. This is a side effect of volume conduction affecting PLV values and does not convey the actual long-range relationships we are interested in.

To get a better representation of the significant connections, we projected the 4-way tensor onto the singular vectors corresponding to $k = l = 2$. Gaussian distribution is fitted to the projected connectivity of three time intervals, $\mathcal{N}_1(-0.01, 0.11)$, $\mathcal{N}_2(-0.01, 0.13)$, and $\mathcal{N}_3(-0.08, 0.14)$. Fig. 3.6 shows the histogram and fitted Gaussian distribution for the time interval $(0, 100)$ms. Fig. 3.8 shows the significant edges of η with $p = 0.95$. As it appears, the result is improved significantly. Instead of detecting the connections between neighboring nodes, the long range connections are detected. In Fig. 3.8(b) ($(0, 100)$ms), there is increasing connectivity in medial-frontal regions, also the engagement of lateral-frontal regions is observable as hypothesized [95, 96, 55]. Connections are largest among medial and lateral prefrontal regions in the $(0, 100)$ms network state relative to the other network states, consistent with engagement of these regions during the ERN. The significant edges of the time interval $(-100, 0)$ms (Fig. 3.8(a)) shows the connectivity between parietal and medial prefrontal cortex regions in the background, and right prefrontal is significantly connected to the parietal region as well. The most significant edges of the time interval $(100, 400)$ms (Fig. 3.8(c)) are between left lateral prefrontal to the parietal region.

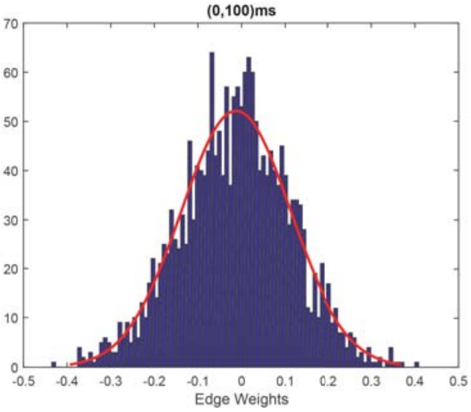
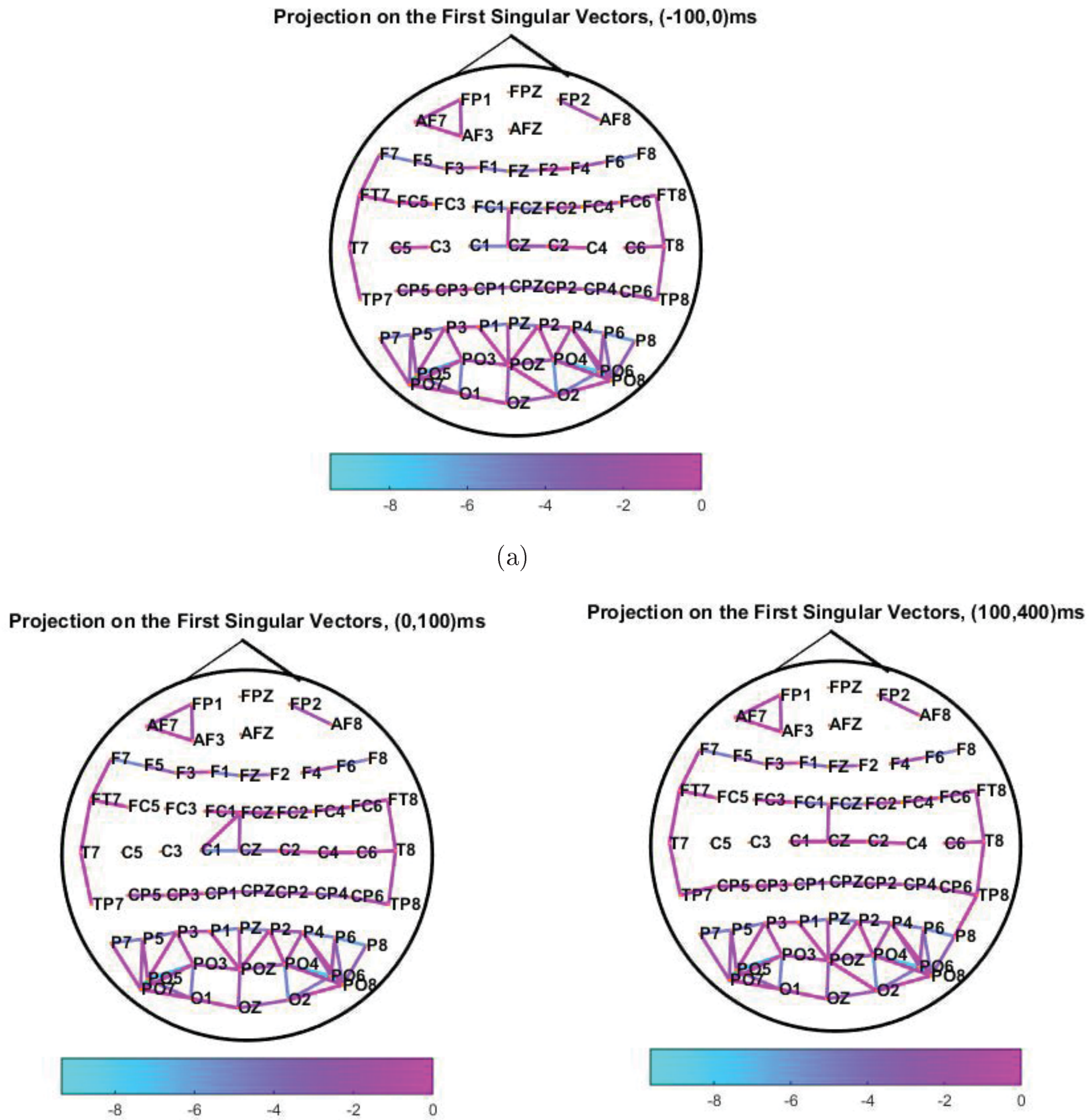


Figure 3.6 The histogram of the edges in network state η_2 corresponding to ERN time interval and the fitted Gaussian distribution $\mathcal{N}_2(-0.20, 0.09)$.

The projection onto the singular vectors $k = l = 3$ is represented in Fig. 3.9, which does not show that much difference from the case $k = l = 2$. Thus, since $k = l = 2$ corresponds to the second largest singular values, it is used in this analysis to filter out the effect of volume conduction while at the same time retaining the most significant activity.


Projection on the First Singular Vectors, (-100,0)ms

(a)

Projection on the First Singular Vectors, (0,100)ms

Projection on the First Singular Vectors, (100,400)ms

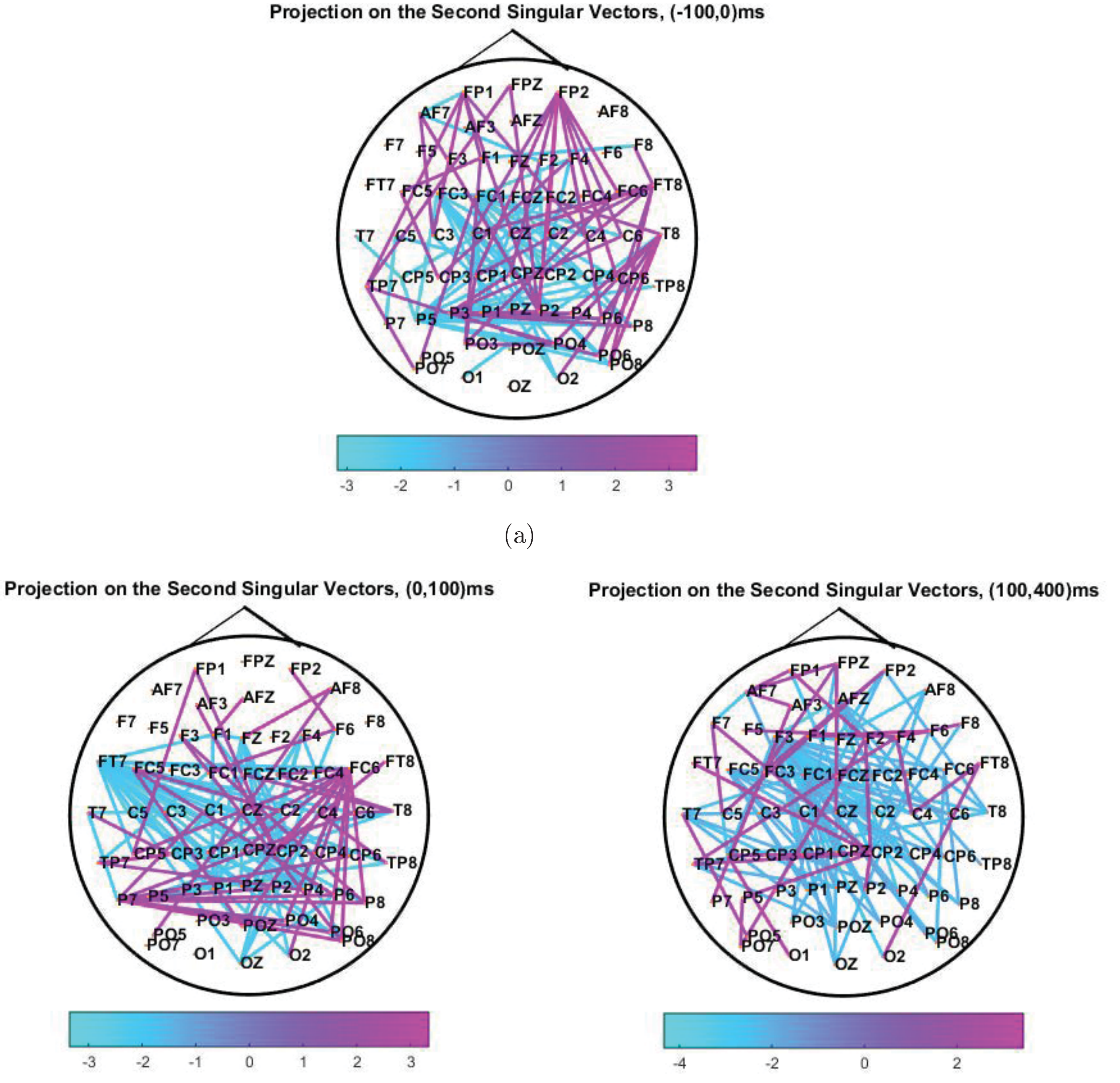
(b)

(c)

Figure 3.7 Projection on single singular vectors: the most significant edges of the projected FC on the singular vectors $k = l = 1$ with $p = 0.95$ for time intervals (a) $(-100, 0)$ms, (b) $(0, 100)$ms, (c) $(100, 400)$ms.

3.5.2.2 Projection on a set of Singular Vectors

In order to detect the significant edges within each interval using the projection on the set of singular vectors, the upper triangular part of η_i is converted to a vector with $\binom{63}{2} = 1953$


Projection on the Second Singular Vectors, (-100,0)ms
-3 -2 -1 0 1 2 3
Projection on the Second Singular Vectors, (0,100)ms
-3 -2 -1 0 1 2 3
Projection on the Second Singular Vectors, (100,400)ms
-4 -2 0 2

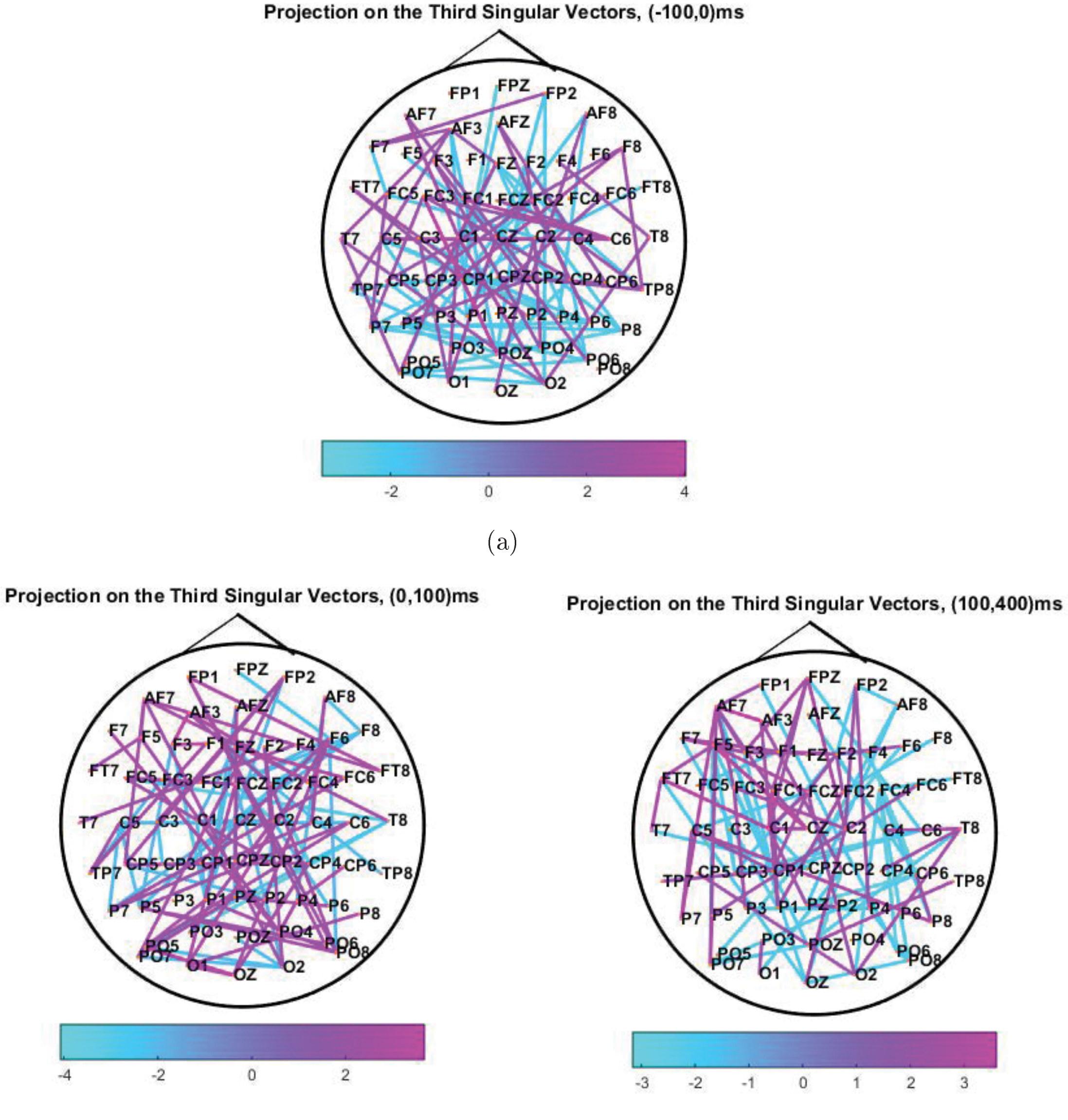
(a)

(b)

(c)

Figure 3.8 Projection on single singular vectors: the most significant edges of the projected FC on the singular vectors $k = l = 2$ with $p = 0.95$ for time intervals (a) $(-100, 0)$ms, (b) $(0, 100)$ms, (c) $(100, 400)$ms.

entries, and Gaussian distributions with different mean and variances are fitted to the vectors. $\mathcal{N}_1(0.22, 0.01)$, $\mathcal{N}_2(-0.20, 0.09)$, and $\mathcal{N}_3(-0.16, 0.09)$ are the Gaussian distributions fitted to (η_1, η_2, η_3), respectively. Fig. 3.10 shows the histogram of η_2 corresponding to ERN time interval and the fitted Gaussian distribution. Kolmogorov–Smirnov test is used to measure the goodness-of-fit for the assumed Gaussian distributions. Kolmogorov–Smirnov test with



(a)

(b)

(c)

Figure 3.9 Projection on single singular vectors: the most significant edges of the projected FC on the singular vectors $k = l = 3$ with $p = 0.95$ for time intervals (a) $(-100, 0)$ms, (b) $(0, 100)$ms, (c) $(100, 400)$ms.

$\alpha = 0.01$ does not reject the null hypothesis that the edges follow a Gaussian distribution.

The summarized functional connectivities $(\eta_1, \eta_2, \cdots, \eta_9)$ are normalized with respect to the estimated means and variances. z–test with $\alpha = 0.05$ is performed to determine the significant edges within the 9 time intervals. Fig. 3.11 shows the significant edges for three time intervals $(-100, 0)$ms, $(0, 100)$ms, $(100, 400)$ms.

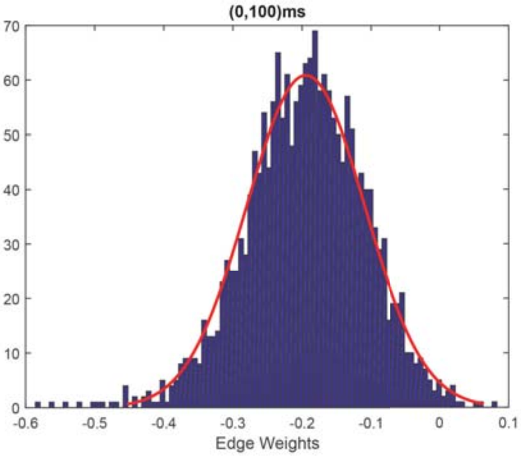
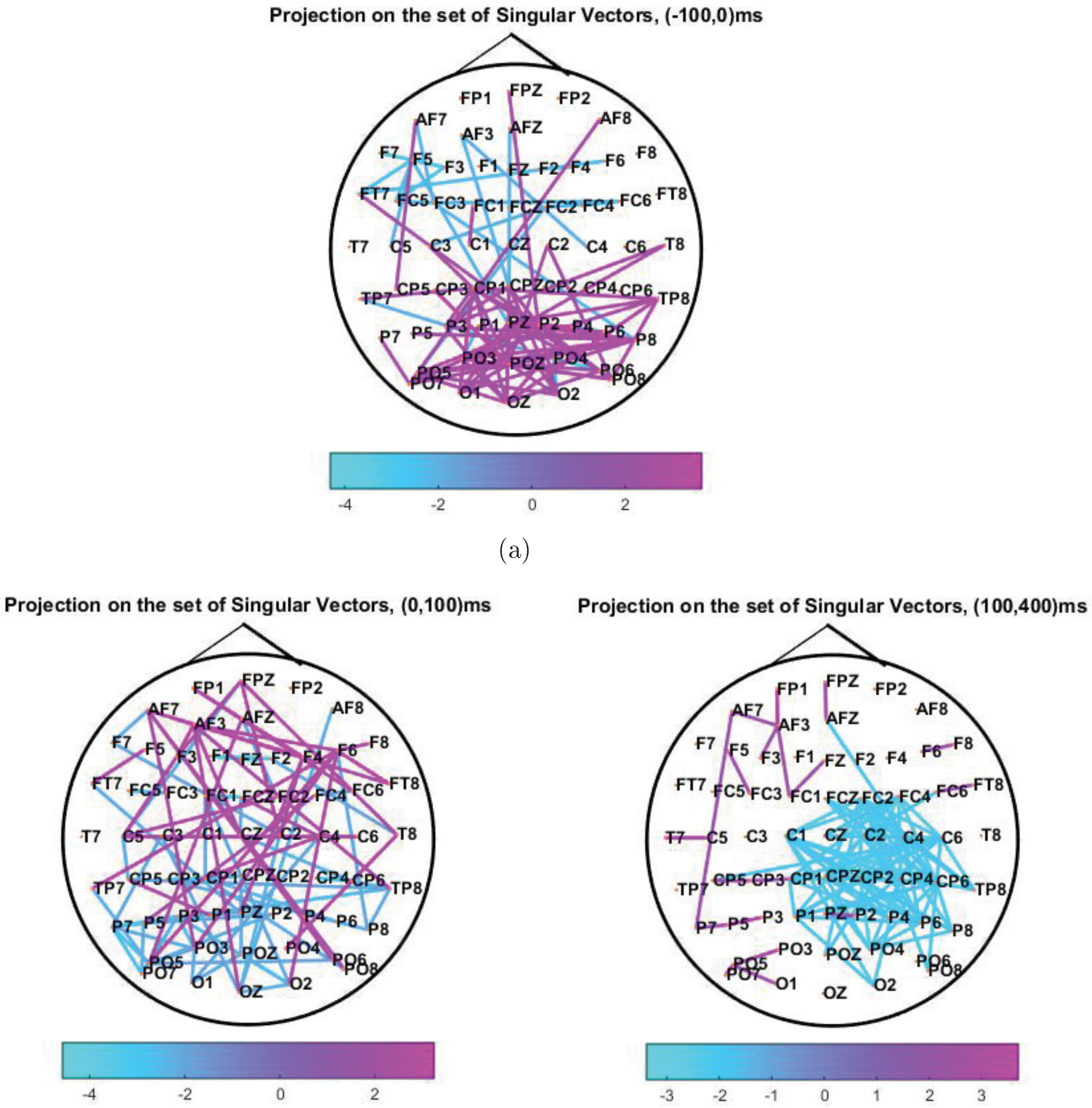


Figure 3.10 The histogram of the edges in network state η_2 corresponding to ERN time interval and the fitted Gaussian distribution $\mathcal{N}_2(-0.20, 0.09)$.

Importantly, because polarity is arbitrary for the connectivity state obtained through tensor projection, the significant tails of the distributions indicate inversely related network subparts. That is, for a given observed network state, the two tails index regions of relative increased versus decreased or inhibited connectivity during that state. As will be described next, two of the states appear to have meaningful activity only on one side. The observed pattern of significant functional connectivities are consistent with observed and hypothesized activity associated with the ERN.

First, in Fig. 3.11 (b) $(0, 100)$ms, increased connectivity is observed in medial-frontal regions, consistent with widely observed ERN amplitude measures. Apparent also is the engagement of lateral-frontal regions as hypothesized, and consistent with previous work [95, 96, 55]. Connections are largest among medial and lateral frontal regions in the $(0, 100)$ms network state relative to the other network states, consistent with engagement of these regions during the ERN. Interestingly, while some left-lateralized parietal regions are functionally connected to the frontal regions during the ERN window, a larger central and right lateralized parietal cluster is deactivated during increases in frontal activity. This is consistent with a functional state dominated by frontal and lateral-frontal regions. Next, the pattern for the network state prior to the ERN window (Fig. 3.11 (a)) contains a similar structure, with a difference being that, relative to the ERN window, the density of connections among frontal regions is decreased while connections among parietal regions is increased. This is consistent with the idea that frontal networks are in the process of becoming functionally integrated leading up to the ERN, where this type of functional integration is maximal. Finally, for the remaining window (Fig. 3.11 (c)) only one tail represents a densely connected state of functional connectivity (where the edges from the other tail are more distributed and disconnected, inconsistent with an integrated functional network and potentially spurious). In Fig. 3.11 (c) $(100, 400)$ms the primary network state continues this shift towards parietal regions. Broadly, the network states identified before, during, and after the ERN are consistent with hypothesized ERN-related network dynamics, focused on the formation of transient frontal networks during error processing.

Similar to the previous section, the result of the tensor projection method is compared with the result of SVD based method. However, since the change points obtained from the SVD approach are not exactly the same as the change points from the tensor approach, the summarized networks would correspond to different time intervals and the comparison would not be fair. Therefore, the ERN time interval obtained from the tensor–based method is used for comparison purposes. For the SVD approach, the matrices constructed by vectorizing and stacking the FC matrices of subjects at each time step are projected onto their first singular vectors to summarize the subject information. The resulting vectors at each time step are concatenated across time to form a new matrix. The new matrix is projected onto its first singular vector to summarize the whole information across time and subject. The two–sided top 5% of edges are shown in Fig. 3.12(a) and (b) for SVD and tensor-based methods, respectively. Substantial previous research, including several reports from our group, support connections between medial and lateral prefrontal cortex area (especially right-frontal). This pattern of connections is apparent for the tensor based approach Fig. 3.12(b), with significant connections centered on electrodes F6, F4. For the SVD based approach presented in Fig. 3.12(a), on the other hand, the pattern is inconsistent with previous reporting



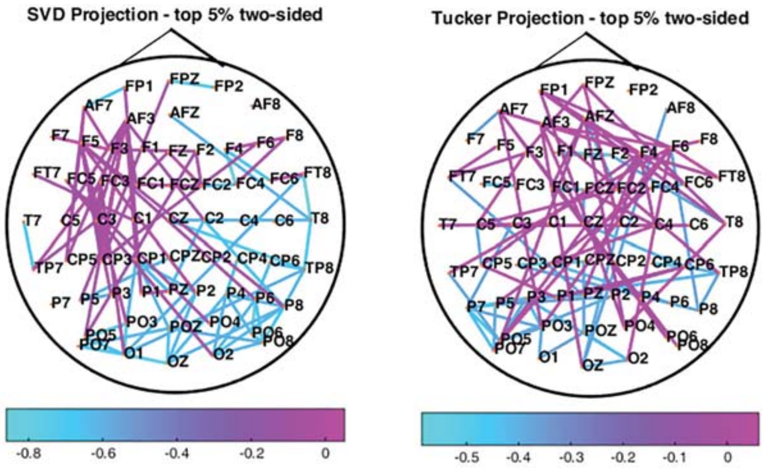
(a)

(b)

(c)

Figure 3.11 Projection on a set of singular vectors: the most significant edges of the projected FC on the set of singular vectors with $p = 0.95$ for time intervals (a) $(-100, 0)$ms, (b) $(0, 100)$ms, (c) $(100, 400)$ms.

about activity during the ERN. Most obviously, connections for the SVD are dominated by a substantially left-lateralized pattern of fronto-central connections.


SVD Projection - top 5% two-sided
Tucker Projection - top 5% two-sided
FP1 FPZ FP2 AF7 AF3 AFZ AF8 F7 F5 F3 F1 FZ F2 F4 F6 F8 FT7 FC5 FC3 FC1 FCZ FC2 FC4 FC6 FT8 T7 C5 C3 C1 CZ C2 C4 C6 T8 TP7 CP5 CP3 CP1 CPZ CP2 CP4 CP6 TP8 P7 P5 P3 P1 PZ P2 P4 P6 P8 PO7 PO5 PO3 POZ PO4 PO6 PO8 O1 OZ O2
-0.8
-0.6
-0.4
-0.2
0
-0.5
-0.4
-0.3
-0.2
-0.1
0

(a) (b)

Figure 3.12 (a) shows the top 5% of the projection result using SVD method, (b) shows the top 5% of the projection result using Tucker decomposition.

3.5.2.3 Functional Connectivity State Representation based on the Detected Change Points

In this section, we present the summarization results of the projection on a set of singular vectors method based on the real detected change points. Tensor projection along subject and time modes for all 9 time intervals is performed to obtain the FC-states $(\eta_1, \eta_2, \cdots, \eta_9)$. In order to detect the significant edges within each interval, the upper triangular part of η_i is converted to a vector with $\binom{63}{2} = 1953$ entries, and Gaussian distributions with different mean and variances are fitted to the vectors. $\mathcal{N}_1(0.22, 0.01)$, $\mathcal{N}_2(0.32, 0.01)$, $\mathcal{N}_3(0.07, 0.01)$, $\mathcal{N}_4(-0.44, 0.01)$, $\mathcal{N}_5(-0.08, 0.01)$, $\mathcal{N}_6(0.24, 0.01)$, $\mathcal{N}_7(-0.40, 0.02)$, $\mathcal{N}_8(-0.40, 0.01)$, and $\mathcal{N}_9$ $(-0.25, 0.02)$ are the Gaussian distributions fitted to $(\eta_1, \eta_2, \cdots, \eta_9)$, respectively. Fig. 3.10 shows the histogram of η_5 corresponding to ERN time interval and the fitted Gaussian distribution. Kolmogorov–Smirnov test is used to measure the goodness-of-fit for the assumed Gaussian distributions. Kolmogorov–Smirnov test with $\alpha = 0.01$ does not reject the null hypothesis that the edges follow a Gaussian distribution.

The summarized functional connectivities $(\eta_1, \eta_2, \cdots, \eta_9)$ are normalized with respect to the estimated means and variances. z–test with $\alpha = 0.05$ is performed to determine the significant edges within the 9 time intervals. Fig. 3.13 (c) shows the significant edges for four time intervals $(-180, -39)$ms, $(-39, 55)$ms, $(55, 281)$ms, $(281, 469)$ms.

3.6 Conclusions

In this chapter, two approaches to dynamic functional connectivity network identification and characterization based on tensor representation of the FCNs were introduced. The first approach identified network states through computing a similarity metric which takes both the similarity of the low-rank approximations to tensors and their proximity in time to obtain a partitioning of the dynamic networks into contiguous time intervals. The second approach introduces a framework for detecting change points at which the network connectivity patterns show consistent change across subjects through low-rank approximations to the tensors and computation of subspace distance between these low-rank approximations. After identifying the boundaries, the network states corresponding to the different time intervals are obtained. In the first approach, this summarization is achieved by projecting the tensor onto one singular vector along each mode, *i.e.* time and subject modes. The second approach extracts the network states through tensor projection onto a set of singular vectors across subject and time modes, which is similar to PCA and extracting eigenvectors with significant energy. The proposed methods are evaluated on EEG data collected during a study of cognitive control. The network states identified before, during, and after the ERN are consistent with hypothesized ERN-related network dynamics, focused on the formation of transient frontal networks during error processing followed by parietal networks. The summarization results of the first approach obtained by projecting on $k = l = 2$ are as good as (not as sparse, though,) the summarization results obtained from the second method. Since the second approach does not require the selection of a particular singular vector, it is preferred over the first method.

The proposed approach offers several improvements over existing approaches. First, the proposed approach does not assume a discrete state model across time and extracts quasi-stationary network states similar to EEG microstates [97]. Second, by using the tensor representation of FCNs the topological structure of the networks are preserved and flexibility for identifying network states across different variables such as frequency and stimulus type is introduced.

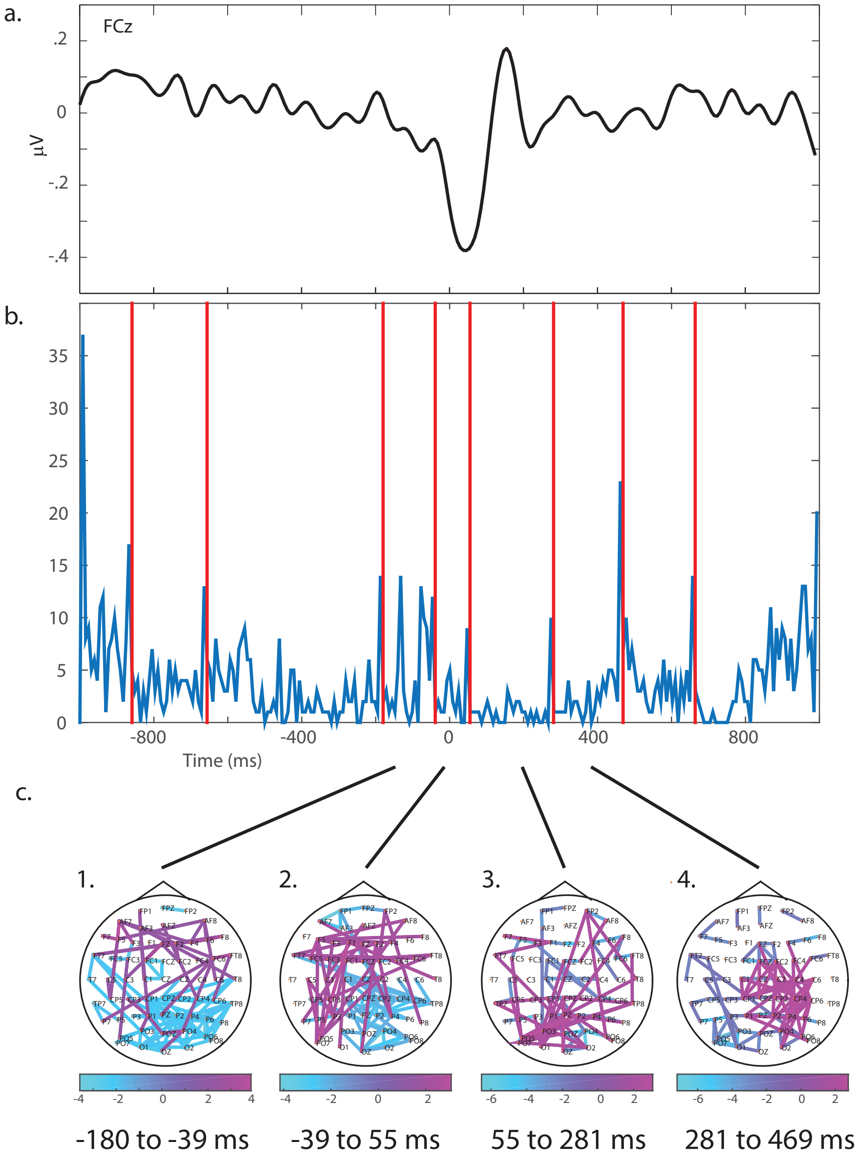


Figure 3.13 (a) Waveform for error trials across all subjects, 0 ms marks the time of the button-press incorrect response; (b) The binary vector resulted from the subspace distance between the projection operators at consecutive time points, calculated using the Grassman distance, where red lines mark the identified change points; (c) The most significant edges of η with $p = 0.95$ for time intervals 1. $(-180, -39)$ ms, 2. $(-39, 55)$ ms, 3. $(55, 281)$ ms, 4. $(281, 469)$ ms.

CHAPTER 4

HIERARCHICAL RECOVERY FOR SPARSE COMMON COMPONENT AND INNOVATION MODEL

4.1 Introduction

Sparse signal approximation refers to the representation of a signal as a linear combination of a small subset of elements of a dictionary [98]. Sparse signal approximation usually considers one signal at a time, not taking into account the correlation within a group of signals. Simultaneous sparse approximation, *a*.k.a. joint sparse recovery, on the other hand, finds sparse representations of multiple signals collected through sensors monitoring the same environment simultaneously using a common dictionary [99, 100, 101]. Joint sparse recovery has been used in many applications such as sensor networks [102], neuroelectromagnetic imaging [103, 104], source localization [105], and image restoration [106].

Different approaches to finding the common sparse representation among a set of signals have been proposed. Tropp *e*t al. [99] proposed a greedy algorithm, *i*.e. simultaneous orthogonal matching pursuit (S-OMP), which extends orthogonal matching pursuit to joint sparse recovery. In [100], a convex relaxation approach was used to find the joint sparse approximation of multiple signals. Blanchard *e*t al. [107] extended well-known sparse approximation methods, iterative hard thresholding, normalized iterative hard thresholding, hard thresholding pursuit, normalized hard thresholding pursuit, and Compressive Sampling Matching Pursuit, to the joint sparse recovery problem.

A closely related line of work was proposed in the compressive sensing community. Distributed compressive sensing (DCS) is an extension of compressive sensing to multiple observations problem [108, 109]. It simultaneously finds the sparse representation of a set of compressively sensed signals by assuming a SCCI model for signals. The SCCI model recovery is a suitable tool to extract common component of highly correlated signals in many applications such as video processing [110, 111, 112] and time-varying networks [113]. The SCCI model can be used to separate the foreground network from the background activity in functional connectivity networks because the background activity is the common component of FC networks across time and subjects. The SCCI model also can be used to extract the sparse common pattern of the quasi-stationary time intervals, representing FC-states.

However, a major problem with using SCCI model to extract the common component of a set of signals is that the size of the dictionary increases dramatically with the number of signals. For a set of J N-dimensional signals, the size of the required dictionary in SCCI recovery method to find the common component is $JN \times (J+1)N$, which demands huge memory and computational resources. In the joint sparse recovery literature, several methods for addressing this high computational complexity have been introduced. Lee *et al* [114] proposed orthogonal subspace matching pursuit (OSMP) for a new joint sparse recovery method, SA-MUSIC. OSMP is used in the the partial support recovery step of SA-MUSIC to provide a computationally efficient solution to joint sparse recovery with a performance guarantee. In [115], the performance of the joint sparse recovery of infinite measurement vectors (IMV), an infinite set of jointly sparse vectors, is improved. Instead of discretizing the entire infinite set of sparse vectors in IMV models, a reduced finite-dimensional problem is derived from which the common nonzero location set can be inferred exactly. In [116], $l_{1,2}$-norm penalty is used to enforce joint sparsity on the signals. The accelerated proximal gradient finds efficiently the joint sparse representation of the set of signals. Hyder and Mahata [117] proposed joint $l_{2,0}$ approximation algorithm (JLZA), an extension of the zero-norm approximation algorithm, to decrease the computational complexity while ensuring the robustness to the measurement noise. Even though these algorithms have improved the computational complexity of simultaneous sparse recovery problem, they all assume that the set of signals is sparse with respect to the same dictionary and the locations of the nonzero entries of the vector coefficient are the same among all signals while their values are different. However, SCCI model focuses on a more general problem with the assumption that the collection of signals have a common sparse component with respect to a basis while the innovation components are sparse with respect to another dictionary with the locations of the nonzero entries of the coefficient vector being different. Thus, these methods are not directly applicable to SCCI model.

In this chapter, we propose a hierarchical algorithm to recover the parts of the SCCI model more efficiently. The proposed algorithm, called Hierarchical SCCI Model Recovery (H-SCCI), partitions the set of signals into a small number of subsets, and finds the common component of each subset separately. The common components of the subsets are then used as a new set of signals to find the common component across all signals. The error bound and the computational complexity of SCCI model and its hierarchical implementation are derived and compared, showing how hierarchical implementation outperforms SCCI model recovery in terms of computational complexity. However, the error bound of H-SCCI is slightly larger than SCCI model recovery due to the aggregated approximation errors across the iterations. Although this method is focused on the two-stage implementation of H-SCCI, it can be easily extended to more stages.

In this chapter, H-SCCI model recovery is used to track dynamic functional connectivity brain networks over time. FC networks generally consist of a background activity, which is common across all time steps, and a foreground network which corresponds to the transient activations [118]. The background activity is similar to the default mode network (DMN) identified from resting state fMRI, and varies slowly across time such that it can be assumed to be sparse with respect to frequency domain bases. Thus, the background activity (background FC pattern) is separated from the foreground by applying H-SCCI model recovery with DCT as the basis of the common component. Since the innovation components are unique to each FC network, the dissimilarity of the innovation components of the consecutive FC networks is employed to detect the change points in the network structure. Once change points are detected, the time intervals can be summarized to obtain their common FC pattern through another stage of H-SCCI model recovery algorithm.

Similarly, the proposed algorithm is evaluated on the video background extraction problem which is of great importance in many automatic video content analysis applications such as surveillance video coding [119], motion detection [120], object tracking [121, 122], *e*tc. Video background extraction aims to separate the moving objects, *a*.k.a. foreground, from the static objects, *a*.k.a. background, in order to facilitate the tracking of moving objects. Even though the subsequent steps of the background extraction are of more interest, the accuracy of the overall system depends on the performance of background extraction. Since the background scene does not change noticeably over time, it has a sparse representation with respect to a frequency basis such as DCT. Thus, H-SCCI with discrete cosine transform (DCT) basis is used to extract the background scene.

4.2 SCCI Model Recovery

Distributed compressive sensing assumes that signals acquired across multiple sensors are sparse in a collection of bases, *i*.e. the set of signals is jointly sparse. Due to the inter-signal correlation, jointly sparse signals are usually assumed to be composed of a common sparse component which is shared by all signals, and an innovation component which is unique to each signal [123]. The encoding part of distributed compressive sensing is not different from compressive sensing in that each signal is separately projected onto some random, incoherent bases. However, the decoding is based on simultaneous sparse recovery of all signals, which can be used for various purposes including the common component extraction. In this thesis, we only focus on the sparse recovery part of SCCI model.

4.2.1 SCCI Model

Let's assume that the set of signals $\Lambda = \{\mathbf{x}_j \in \mathbb{R}^N; \forall j \in \{1, 2, \ldots, J\}\}$ are jointly sparse. It is assumed that there is an inter-signal correlation among the signals. The SCCI model [124, 109], which includes a common component $\mathbf{z}_c \in \mathbb{R}^N$ and an innovation component

$\mathbf{z}_j \in \mathbb{R}^N$, can be written as:

$$\mathbf{x}_j = \mathbf{z}_c + \mathbf{z}_j, \;\; j = 1, 2, \ldots, J. \tag{4.1}$$

The common component represents the inter-signal correlation among the signals while the innovation component is unique to each signal. The common and innovation components of the set of signals Λ are sparse with respect to two different sets of bases, ϕ_c and ϕ_j, respectively, as:

$$\begin{gathered} \mathbf{z}_c = \phi_c \theta_c, \\ \mathbf{z}_j = \phi_j \theta_j, \quad\; j \in \{1, 2, \ldots, J\}, \end{gathered} \tag{4.2}$$

where θ_c and θ_j are the coefficient vectors, and the bases ϕ_c and ϕ_j are orthogonal. Since the signal $\mathbf{x}_j$ is sparse, the coefficient vectors have a small number of nonzero entities, $\|\theta_c\|_0 = K_c \ll N$ and $\|\theta_j\|_0 = K_j \ll N$.

In order to recover the sparse representation of the set of signals Λ, all signals are stacked to form a single optimization problem. Eq. 4.3 shows the compact representation of all signals and their sparse representations in matrix format.

$$\begin{gathered} \mathbf{X} = \mathbf{\Phi\Theta}, \\ \mathbf{\Theta} = \begin{bmatrix} \theta_c \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_J \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_J \end{bmatrix}, \qquad \mathbf{\Phi} = \begin{bmatrix} \phi_c & \phi_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \phi_c & \mathbf{0} & \phi_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \phi_c & \mathbf{0} & \cdots & \mathbf{0} & \phi_J \end{bmatrix}, \end{gathered} \tag{4.3}$$

where $\mathbf{X} \in \mathbb{R}^{JN\times 1}$, $\mathbf{\Theta} \in \mathbb{R}^{JN\times 1}$, and $\mathbf{\Phi} \in \mathbb{R}^{JN\times (J+1)N}$.

The goal is to find the coefficient vector $\mathbf{\Theta}$ such that the error between the signal $\mathbf{X}$ and the sparse representation $\mathbf{\Phi\Theta}$ is minimized:

$$\hat{\mathbf{\Theta}} = \arg\min \|\mathbf{\Theta}\|_0 \text{ s.}t. \ \|\mathbf{X} - \mathbf{\Phi\Theta}\| \leq \epsilon. \tag{4.4}$$

where ϵ is the bound on the recovery error. The optimization problem in Eq. 4.5 is l_0-norm minimization problem which is NP-hard and non-convex. The l_0-norm minimization is usually relaxed to l_1-norm minimization problem. However, in order to make the optimization problem more general and flexible, $\lambda - \textit{weighted}\, l_1$-norm minimization problem is proposed [124, 109], as:

$$\hat{\mathbf{\Theta}} = \arg\min \lambda_c \|\theta_c\|_1 + \sum_{j=1}^{J} \lambda_j \|\theta_j\|_1 \ \text{ s.}t. \ \|\mathbf{X} - \mathbf{\Phi\Theta}\| \leq \epsilon, \tag{4.5}$$

where $\lambda_c, \lambda_j \geq 0$ are the weights of common and innovation components. In [125, 109], it is shown numerically that this modification improves the performance of the signal recovery. The weights, λ_js, for the innovation components are usually selected to be the same. Without loss of generality, λ_js are selected to be equal to one and the optimal value of λ_c is determined numerically [125]. The estimated coefficients, $\hat{\mathbf{\Theta}}$, are used for the sparse representation of the set of signals, or for recovering the common component $\mathbf{z}_c$.

4.3 Hierarchical SCCI Model Recovery

4.3.1 Hierarchical SCCI Model

The joint sparsity model assumed for the set of signals, $\Lambda = \{\mathbf{x}_j \in \mathbb{R}^N; \forall j \in \{1, 2, \ldots, J\}\}$, is SCCI, as in Eq. 4.1. In this thesis, in order to make the computation faster, we partition the set of signals, Λ, into Γ arbitrary, non-overlapping subsets, Λ_γ, such that:

$$\begin{gathered} \Lambda_\gamma \subset \Lambda \quad \gamma = 1, 2, \ldots, \Gamma, \\ \Lambda = \textstyle\bigcup_{\gamma=1}^{\Gamma} \Lambda_\gamma, \\ \Lambda_i \bigcap \Lambda_j = \emptyset \quad \text{for} \quad i \neq j. \end{gathered} \tag{4.6}$$

Once the sparse recovery procedure is explained, the way these subsets are selected will be clarified in Section 4.3.3.

All signals $\mathbf{x}_j \in \Lambda$ have a common component, $\mathbf{z}_c$, due to the inter-signal correlation. Consequently, all signals within each subset $\mathbf{x}_j \in \Lambda_\gamma \subset \Lambda$ also have the same common component, $\mathbf{z}_c$. In addition, the signals within each subset may have additional inter-signal correlation, which leads to the common component within the subset Λ_γ. Thus, this model includes a global common component $\mathbf{z}_c \in \mathbb{R}^N$, a within–subset common component $\mathbf{w}_{c,\gamma} \in \mathbb{R}^N$, and a within–subset innovation component $\mathbf{w}_j \in \mathbb{R}^N$, as:

$$\mathbf{x}_j = \mathbf{z}_c + \mathbf{w}_{c,\gamma} + \mathbf{w}_j \text{ for } \mathbf{x}_j \in \Lambda_\gamma. \tag{4.7}$$

$\mathbf{u}_\gamma = \mathbf{z}_c + \mathbf{w}_{c,\gamma}$ is defined as the common component of the signals within the subset $\mathbf{x}_j \in \Lambda_\gamma$ while $\mathbf{z}_c$ is the global common component. Similarly, $\mathbf{z}_j = \mathbf{w}_{c,\gamma} + \mathbf{w}_j$ is the global innovation component. Since we assume the same joint sparsity model as Eq. 4.1, the common and innovation components are sparse on two sets of bases, ϕ_c and ϕ_j, respectively, as:

$$\begin{aligned} \mathbf{u}_\gamma &= \phi_c \theta_{u_\gamma}; \quad \gamma = 1, 2, \ldots, \Gamma, \\ \mathbf{w}_j &= \phi_j \theta_{w_j} \;\; ; \quad \{j | \mathbf{x}_j \in \Lambda_\gamma\}, \end{aligned} \tag{4.8}$$

where θ_c and θ_j are the coefficient vectors, with $\|\theta_c\|_0 = K_{u_\gamma}$ and $\|\theta_{w_j}\|_0 = K_{w_j}$. Fig. 4.1 shows the flowchart of H-SCCI.

4.3.2 Signal Recovery

The goal is to extract the common component $\mathbf{z}_c$ when J is large. First, the common component of each subset Λ_γ is extracted similar to SCCI model recovery through l_1-norm optimization algorithm by searching for the components with the sparsest coefficient vectors and minimum error. All signals of the subset Λ_γ are stacked to form a vector as follows:

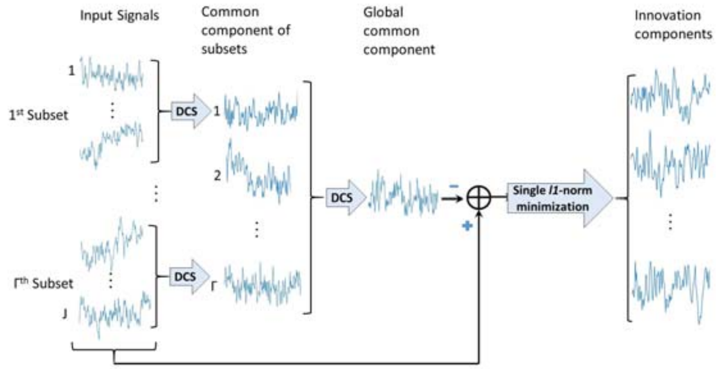


Figure 4.1 Flowchart of the proposed method: Hierarchical SCCI Model for two stages.

$$\mathbf{X}_\gamma = \mathbf{\Phi}_\gamma \mathbf{\Theta}_\gamma, \qquad \Lambda_\gamma = \{\mathbf{x}_{l_1}, \mathbf{x}_{l_2}, \ldots, \mathbf{x}_{l_{\bar{\gamma}}}\},$$

$$\mathbf{\Theta}_\gamma = \begin{bmatrix} \theta_{W_\gamma} \\ \theta_{w_{l_1}} \\ \theta_{w_{l_2}} \\ \vdots \\ \theta_{w_{l_{\bar{\gamma}}}} \end{bmatrix}, \qquad \mathbf{X}_\gamma = \begin{bmatrix} \mathbf{x}_{l_1} \\ \mathbf{x}_{l_2} \\ \vdots \\ \mathbf{x}_{l_{\bar{\gamma}}} \end{bmatrix}, \qquad \mathbf{\Phi}_\gamma = \begin{bmatrix} \phi_c & \phi_{l_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \phi_c & \mathbf{0} & \phi_{l_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \phi_c & \mathbf{0} & \cdots & \mathbf{0} & \phi_{l_{\bar{\gamma}}} \end{bmatrix}, \tag{4.9}$$

where $\{l_1, l_2, \ldots, l_{\bar{\gamma}}\}$ is the set of indices for the signals of the γth subset, and $\bar{\gamma}$ is the number of signals in the γth subset.

The common component for the γth subset, $\mathbf{u}_\gamma$, is estimated using $\hat{\mathbf{u}}_\gamma = \phi_c \hat{\theta}_{u_\gamma}$ through solving the following optimization equation:

$$\hat{\mathbf{\Theta}}_\gamma = \arg\min \|\mathbf{\Theta}_\gamma\|_1 \ \text{ s.}t. \ \|\mathbf{X}_\gamma - \mathbf{\Phi}_\gamma \mathbf{\Theta}_\gamma\| \leq \epsilon. \tag{4.10}$$

Once the common components of all subsets $\mathbf{u}_\gamma; \forall\gamma \in \{1, 2, \ldots, \Gamma\}$ are extracted, they are considered as a new set of jointly sparse signals with the SCCI model: $\hat{\mathbf{u}}_\gamma = \mathbf{z}_c + \mathbf{w}_{c,\gamma}$, where $\mathbf{z}_c$ is their common component. $\mathbf{z}_c$ is recovered through another l_1-norm optimization algorithm.

After recovering the common component $\mathbf{z}_c$, the sparse representation of the innovation components, $\mathbf{z}_j$, are estimated through a single signal recovery problem:

$$\hat{\theta}_j = \arg\min \|\theta_j\|_1 \text{ s.}t. \ \|(\mathbf{x}_j - \mathbf{z}_c) - \phi_j\theta_j\| \leq \epsilon. \tag{4.11}$$

The pseudocode of the hierarchical SCCI model recovery is presented in algorithm 1.

Algorithm 4.3.1 Hierarchical SCCI Model Recovery

```
Input: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J\}$, $\phi_c$, $\{\phi_1, \phi_2, \ldots, \phi_J\}$, and $\Gamma$.
Output: The common component $\mathbf{z}_c$, and the innovation components $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_J\}$.
Partition $J$ signals into $\Gamma$ non-overlapping groups.
for $\gamma$ = 1:$\Gamma$ do
    For each partition form the basis matrix ($\mathbf{\Phi} \in \mathbb{R}^{N\bar{\gamma}\times(\bar{\gamma}+1)N}$) and the signal vector ($\mathbf{X} \in \mathbb{R}^{\bar{\gamma}N\times 1}$) in Eq. 4.9 from the signals in the $\gamma$th subset.
    Solve $l1$-norm minimization in Eq. 4.10.
    Generate $\hat{\mathbf{u}}_\gamma = \mathbf{z}_c + \mathbf{w}_{c,\gamma} = \phi_c\theta_{W_\gamma}$ from the recovered coefficients $\hat{\mathbf{\Theta}}_\gamma$.
end for
Form $\mathbf{\Phi} \in \mathbb{R}^{N\Gamma\times(\Gamma+1)N}$ and $\mathbf{X} \in \mathbb{R}^{N\Gamma\times 1}$ from $\hat{\mathbf{u}}_\gamma \in \mathbb{R}^{N\times 1}$ for $\gamma = 1, 2, \ldots, \Gamma$.
Solve $l1$-norm minimization for the new set of signals to recover $\mathbf{z}_c$ from $\hat{\mathbf{u}}_\gamma$s.
Recover the innovation components of the signals using Eq. 4.11
```

4.3.3 Subset Selection

In order to implement the proposed algorithm, the set of J signals has to be split into Γ groups. Here, we do not assume any *a priori* information regarding grouping the signals. Indeed, there is no need for the set of signals to be partitioned in a specific manner. Random grouping of the set of signals into Γ subsets with equal size leads to the same results. We illustrate this through an example.

Assume we have a set of 4 jointly sparse signals, $\Lambda = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$. Our goal is to find the joint sparse representation of this set of signals. Assume there is an inter-signal correlation across the whole collection of signals, which is the global common component $\mathbf{z}_c$. In addition, assume that there is an additional within-subset common component $\mathbf{w}_{c,1}$, between $\mathbf{x}_1$ and $\mathbf{x}_2$ which does not exist between these two signals and the remaining signals, $\mathbf{x}_3$ and $\mathbf{x}_4$. Note that this common component may also be equal to zero. Similarly, there is another within-subset common component, $\mathbf{w}_{c,2}$, between $\mathbf{x}_3$ and $\mathbf{x}_4$. However, in practice, we assume that we do not have *a priori* information about the structure of the grouping and the inter-signal correlations. The set of signals can be written in terms of their components as follows:

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{z}_c + \mathbf{w}_{c,1} + \mathbf{w}_1,\\ \mathbf{x}_2 &= \mathbf{z}_c + \mathbf{w}_{c,1} + \mathbf{w}_2,\\ \mathbf{x}_3 &= \mathbf{z}_c + \mathbf{w}_{c,2} + \mathbf{w}_3,\\ \mathbf{x}_4 &= \mathbf{z}_c + \mathbf{w}_{c,2} + \mathbf{w}_4,\end{aligned} \tag{4.12}$$

where $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$ are the innovation components of the signals. Assuming that the common and innovation bases are orthogonal, Eq.4.12 can be rewritten as follows:

$$\begin{aligned}\mathbf{x}_1 &= \phi_c\theta_c + \phi_j\theta_{c,w_1} + \phi_j\theta_{w_1},\\ \mathbf{x}_2 &= \phi_c\theta_c + \phi_j\theta_{c,w_1} + \phi_j\theta_{w_2},\\ \mathbf{x}_3 &= \phi_c\theta_c + \phi_j\theta_{c,w_2} + \phi_j\theta_{w_3},\\ \mathbf{x}_4 &= \phi_c\theta_c + \phi_j\theta_{c,w_2} + \phi_j\theta_{w_4}.\end{aligned} \tag{4.13}$$

First, assume that we randomly partition these four signals into two subsets such that $\Lambda_1 = \{\mathbf{x}_1, \mathbf{x}_2\}$ and $\Lambda_2 = \{\mathbf{x}_3, \mathbf{x}_4\}$, the common components of the subsets will then be found through Algorithm 1 as

$$\mathbf{u}_1 = \mathbf{z}_c + \mathbf{w}_{c,1} = \phi_c \underbrace{\left(\theta_c + \phi_c^{-1}\phi_j\theta_{c,w_1}\right)}_{\theta_{\mathbf{u}_1}},$$

$$\mathbf{u}_2 = \mathbf{z}_c + \mathbf{w}_{c,2} = \phi_c \underbrace{\left(\theta_c + \phi_c^{-1}\phi_j\theta_{c,w_2}\right)}_{\theta_{\mathbf{u}_2}}.$$

By considering $\{\mathbf{u}_1, \mathbf{u}_2\}$ as a new set of jointly sparse signals, the common component of $\{\mathbf{u}_1, \mathbf{u}_2\}$ is extracted as $\mathbf{z}_c = \phi_c\theta_c$, which is the common component among all four signals.

Next, let's assume that the four signals are randomly categorized into two different subsets $\Lambda_1 = \{\mathbf{x}_1, \mathbf{x}_3\}$ and $\Lambda_2 = \{\mathbf{x}_2, \mathbf{x}_4\}$, then the common components of the subsets will be found as $\mathbf{u}_1 = \mathbf{z}_c = \phi_c\theta_c$ and $\mathbf{u}_2 = \mathbf{z}_c = \phi_c\theta_c$. If we construct a new subset of jointly sparse signals from $\mathbf{u}_1$ and $\mathbf{u}_2$, their common component is common among all four signals. Thus, no matter how the signals are categorized, the extracted common component is the true one, $\mathbf{z}_c$. If there is no inter-signal correlation within the subsets, i.e. $\mathbf{w}_{c,1} = \mathbf{w}_{c,2} = 0$, the common component of the subsets (regardless of the partitioning) are $\mathbf{u}_1 = \mathbf{z}_c$ and $\mathbf{u}_2 = \mathbf{z}_c$.

4.3.4 Computational Cost

The proposed algorithm is significantly faster than existing simultaneous sparse representation algorithms. However, the computational cost mainly depends upon the algorithm used in solving the optimization problem. The optimization problems can be solved through either the vector-matrix multiplication or fast operators [92].

The SCCI model recovery of J signals with N samples requires a $JN \times (J+1)N$ dictionary, $\widetilde{\Phi}$, to be multiplied by a $(J+1)N$-dimensional coefficient vector $\mathbf{\Theta}$. The cost of this vector-matrix multiplication is $\vartheta = \mathcal{O}\left((J+1)Nlog\left((J+1)N\right)\right)$. The reason hierarchical SCCI model recovery is able to reduce the computational cost significantly is that it splits the dictionary to Γ matrices so that the optimization algorithm looks for the sparse representation of signals over a smaller set of bases.

In the worst case scenario, SCCI model recovery method implemented using the l_1-norm minimization costs $I\frac{(2(J+1)N)^3}{3} + \mathcal{O}(J(J+1)N^2)$, where I is the number of iterations and assumed to be fixed for all cases [92]. The advantage of hierarchical SCCI model recovery method is that it can be implemented through multiple steps, and the J signals can be partitioned into Γ groups. In the following derivations, we will assume that one level of partitioning is adequate to compute the joint sparse recovery of signals, *i.e.* the common components of Γ groups, $\mathbf{z}_c + \mathbf{z}_\gamma$, are calculated separately in the first stage and the global common component, $\mathbf{z}_c$, is extracted from the Γ outcomes of the first stage. Each group contains $\frac{J}{\Gamma}$ signals in the first stage. Using the l_1-norm minimization, each group in the first stage requires $I\frac{\left(2(\frac{J}{\Gamma}+1)N\right)^3}{3} + \mathcal{O}(\frac{J}{\Gamma}(\frac{J}{\Gamma}+1)N^2)$ multiplications at most. These optimization problems are independent of each other and thus can be performed in parallel. In addition, the second stage cost is bounded by $I\frac{(2(\Gamma+1)N)^3}{3}+\mathcal{O}(\Gamma(\Gamma+1)N^2)$. Thus, the total computational cost required to recover the joint common component using Hierarchical SCCI model is upper bounded by

$$I\frac{(2(J+\Gamma)N)^3}{3\Gamma^3}+I\frac{(2(\Gamma+1)N)^3}{3}+\mathcal{O}\left(N^2\frac{J^2+\Gamma(\Gamma^3+\Gamma^2+J)}{\Gamma^2}\right). \tag{4.14}$$

For instance, let's suppose that we have $J=20$ jointly sparse signals with $N=256$. SCCI model recovery with l_1-norm minimization costs $I4.14\times 10^{11}+\mathcal{O}(2.75\times 10^7)$. If we divide the set of signals into $\Gamma=4$ groups, this cost becomes $I4.43\times 10^{10}+\mathcal{O}\left(3.28\times 10^6\right)$. The computational cost of regular implementation is $9.36+\mathcal{O}(8.4)$ times greater than the hierarchical-based implementation, which is more than 10 times slower.

Using OMP for the *l1*-norm optimization will result in a computational cost of $\frac{4(JN)^3}{3}+N^3J^2(J+1)+\mathcal{O}\left(N(J+1)\right)$ for SCCI model recovery [92]. The hierarchical SCCI model recovery based on OMP costs $\frac{4(\frac{J}{\Gamma}N)^3}{3}+N^3\frac{J^2}{\Gamma^2}(\frac{J}{\Gamma}+1)+\mathcal{O}\left(N(\frac{J}{\Gamma}+1)\right)$ in the first stage and $\frac{4(\Gamma N)^3}{3}+N^3\Gamma^2(\Gamma+1)+\mathcal{O}\left(N(\Gamma+1)\right)$ in the second stage. Thus, the total cost is $\frac{4J^3+\Gamma^5N^3}{3\Gamma^2}+\frac{N^3}{\Gamma^2}(J^3+J^2\Gamma+\Gamma^5+\Gamma^4)+\mathcal{O}\left(N(J+2\Gamma+1)\right)$. Table 4.1 shows the computational time for SCCI model recovery and H-SCCI model recovery for different values of N and J. When $J=8$ and there is only one subset $\Gamma=1$, and the computational time of SCCI recovery model and H-SCCI recovery model are very close. However, as J increases, the difference between computational time becomes more apparent.

Table 4.1 The computational cost of SCCI and H-SCCI recovery model using OMP for different values of J and N. The numbers in the numerator show the computational cost of SCCI model recovery and the numbers in the denominator show that of H-SCCI model recovery.

$\frac{J}{\Gamma}$ N	8/1	$\frac{16}{2}$	$\frac{32}{4}$	$\frac{64}{8}$	$\frac{128}{16}$
128	$\frac{2.64e^{9}+\mathcal{O}(1.15e^{3})}{1.21e^{9}+\mathcal{O}(1.41e^{3})}$	$\frac{2.06e^{10}+\mathcal{O}(2.18e^{3})}{3.66e^{8}+\mathcal{O}(1.66e^{3})}$	$\frac{1.62e^{11}+\mathcal{O}(4.22e^{3})}{3.13e^{8}+\mathcal{O}(2.18e^{3})}$	$\frac{1.29e^{12}+\mathcal{O}(8.32e^{3})}{1.60e^{9}+\mathcal{O}(3.20e^{3})}$	$\frac{1.03e^{13}+\mathcal{O}(1.65e^{4})}{1.20e^{10}+\mathcal{O}(5.25e^{3})}$
256	$\frac{2.11e^{10}+\mathcal{O}(2.30e^{3})}{9.70e^{9}+\mathcal{O}(2.82e^{3})}$	$\frac{1.65e^{11}+\mathcal{O}(4.35e^{3})}{2.93e^{9}+\mathcal{O}(3.33e^{3})}$	$\frac{1.30e^{12}+\mathcal{O}(8.45e^{3})}{2.51e^{9}+\mathcal{O}(4.35e^{3})}$	$\frac{1.03e^{13}+\mathcal{O}(1.67e^{4})}{1.28e^{10}+\mathcal{O}(6.40e^{3})}$	$\frac{8.24e^{13}+\mathcal{O}(3.30e^{4})}{9.60e^{10}+\mathcal{O}(1.05e^{4})}$
512	$\frac{1.69e^{11}+\mathcal{O}(4.61e^{3})}{7.76e^{10}+\mathcal{O}(5.63e^{3})}$	$\frac{1.32e^{12}+\mathcal{O}(8.70e^{3})}{2.34e^{10}+\mathcal{O}(6.66e^{3})}$	$\frac{1.04e^{13}+\mathcal{O}(1.69e^{4})}{2.00e^{10}+\mathcal{O}(8.70e^{3})}$	$\frac{8.27e^{13}+\mathcal{O}(3.33e^{4})}{1.02e^{11}+\mathcal{O}(1.28e^{4})}$	$\frac{6.59e^{14}+\mathcal{O}(6.60e^{4})}{7.68e^{11}+\mathcal{O}(2.10e^{4})}$
1024	$\frac{1.35e^{12}+\mathcal{O}(9.22e^{3})}{6.21e^{11}+\mathcal{O}(1.13e^{4})}$	$\frac{1.05e^{13}+\mathcal{O}(1.74e^{4})}{1.88e^{11}+\mathcal{O}(1.33e^{3})}$	$\frac{8.32e^{13}+\mathcal{O}(3.38e^{4})}{1.60e^{11}+\mathcal{O}(1.74e^{4})}$	$\frac{6.61e^{14}+\mathcal{O}(6.66e^{4})}{8.19e^{11}+\mathcal{O}(2.56e^{4})}$	$\frac{5.27e^{15}+\mathcal{O}(1.32e^{5})}{6.15e^{12}+\mathcal{O}(4.20e^{4})}$

4.3.5 Error Bound

Joint recovery of the sparse representation of a set of signals faster comes with the tradeoff of increased error. Since the problem of finding the sparse representation of the set of signals jointly is divided into Γ different joint sparse recovery problems, this increases the reconstruction error.

Assuming we want to recover $\mathbf{\Theta}$ from the observation vector $\mathbf{X}$, it is shown [126] that the reconstruction error from the l_1-norm optimization problem

$$\min \|\mathbf{\Theta}\|_1 \quad s.t. \quad \|\mathbf{X} - \Phi\mathbf{\Theta}\| \leq \epsilon$$

is bounded by:

$$\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2 \leq C_v \epsilon, \tag{4.15}$$

where the constant C_v depends on δ_{4v} the $4v$-restricted isometry constant of Φ.

However, most signals are not exactly sparse, rather approximately sparse. The coefficient vector $\mathbf{\Theta}$ of such signals can be truncated with the v largest coefficients, $\mathbf{\Theta}_v$. Then, the error bound for the approximately sparse signals follows:

$$\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2 \leq C_{1,v}\epsilon + C_{2,v}\frac{\|\mathbf{\Theta} - \mathbf{\Theta}_v\|_1}{\sqrt{v}}. \tag{4.16}$$

SCCI model recovers the common and innovation components jointly within one l_1-norm optimization problem, which means the vector $\mathbf{\Theta}$ includes the sparse coefficients of all signals. Thus, the error bound for SCCI model recovery can be computed in terms of its components, $\mathbf{z}_c$, $\mathbf{w}_{c,\gamma}$ and $\mathbf{w}_j$, using Eq. 4.16., as:

$$\begin{gathered}\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}_v\|_1 = \textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| \\ + \textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| + \sum_{j=1}^{J}\sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|,\end{gathered} \tag{4.17}$$

where we define the truncated vector $\mathbf{\Theta}_v$ as a vector with Jv nonzero entities. Each coefficient vector in $\hat{\mathbf{\Theta}}$ corresponding to J signals are truncated using the v largest values. The truncated vector $\mathbf{\Theta}_v$ is built from these J truncated coefficient vectors. The upper bound for SCCI recovery model is:

$$\begin{aligned}\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2 \leq C_{1,v}\epsilon + \frac{C_{2,v}}{\sqrt{v}} \Big(\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| \\ + \textstyle\sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| + \sum_{j=1}^{J} \sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|\Big).\end{aligned} \tag{4.18}$$

In order to calculate the upper error bound of the hierarchical SCCI model recovery, we partition the dataset into Γ subsets. Each subset requires a l_1-norm optimization problem, and the global common component is recovered through another l_1-norm optimization problem, which leads to a total of $\Gamma + 1$ optimization problems.

Theorem 1: Assume that $\mathbf{X} = \Phi\mathbf{\Theta}$ is a vector of J jointly sparse signals following the signal model in Eq. 4.7, where Φ is the dictionary built from ϕ_c, ψ_is, and ϕ_js. Φ has v-restricted isometry constant δ_v, and $\|\mathbf{X} - \Phi\mathbf{\Theta}\|_2 \leq \epsilon$. Then, the error made by the hierarchical SCCI model recovering the jointly sparse representation of J signals is bounded by:

$$\begin{aligned}\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2 \leq C_{1,v}\epsilon + \frac{C_{2,v}}{\sqrt{v}} \Big(2\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| \\ +2 \textstyle\sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| + \sum_{j=1}^{J} \sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|\Big),\end{aligned} \tag{4.19}$$

where $C_{1,v}$ and $C_{2,v}$ depend on δ_{4v} [126].

Proof:

Following the upper error bound provided by Eq. 4.16, the upper error bound for the γth subset is:

$$\begin{aligned}\|\hat{\mathbf{\Theta}}_\gamma - \mathbf{\Theta}_\gamma\|_2 \leq C_{1,v,\gamma}\epsilon + \frac{C_{2,v,\gamma}}{\sqrt{v}} \Big(\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| \\ +\textstyle\sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| + \sum_{j\in\gamma th \text{ subset}} \sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|\Big).\end{aligned} \tag{4.20}$$

The constants $C_{1,v,\gamma}$ and $C_{2,v,\gamma}$ depend on the v-restricted isometry constant, $\delta_{v,\gamma}$, of the matrix Φ_γ. Since Φ_γ is a submatrix of Φ, the relationship between the local isometry

$\delta_{v,\gamma}$ and the global isometry δ_v, the v-restricted isometry constant of the matrix Φ, is as follows [127]:

$$\delta_v = \sup_{\gamma} \ \delta_{v,\gamma}. \tag{4.21}$$

Therefore, we can conclude that:

$$\begin{aligned}\|\hat{\mathbf{\Theta}}_\gamma - \mathbf{\Theta}_\gamma\|_2 \leq C_{1,v}\epsilon + \frac{C_{2,v}}{\sqrt{v}} \Big(&\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| + \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| \\ &+ \textstyle\sum_{j\in\gamma th\ subset} \sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|\Big) \forall\gamma \in \{1, 2, \ldots, \Gamma\}.\end{aligned} \tag{4.22}$$

After calculating the sparse representation of Γ subsets, the last l_1-norm optimization step finds the common part among their outputs. The recovery error from this last step is calculated similarly using Eq. 4.16, as follows:

$$\|\hat{\mathbf{\Theta}}_c - \mathbf{\Theta}_c\|_2 \leq C_{1,v}\epsilon + \frac{C_{2,v}}{\sqrt{v}} \Big(\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| + \sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]|\Big). \tag{4.23}$$

Therefore, the total recovery error for the proposed algorithm is bounded by:

$$\begin{aligned}\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2 \leq C_{1,v}\epsilon + \frac{C_{2,v}}{\sqrt{v}} \Big(&2\textstyle\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| \\ +2 \textstyle\sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]| &+ \textstyle\sum_{j=1}^{J} \sum_{k=1}^{N} |w_j[k] - w_{j,v}[k]|\Big).\end{aligned} \tag{4.24}$$

□

Comparing the error bounds of SCCI and hierarchical SCCI model recovery in Eq. 4.18 and Eq. 4.19 imply that the upper error bound of hierarchical SCCI model recovery is larger than that of SCCI model by $\frac{C_{2,v}}{\sqrt{v}} \Big(\sum_{k=1}^{N} |z_c[k] - z_{c,v}[k]| + \sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{N} |w_{c,\gamma}[k] - w_{c,\gamma,v}[k]|\Big)$.

More recently, Needell *et.* al [128] proposed Regularized Orthogonal Matching Pursuit (ROMP) to find a sparse representation of signals with a tighter error bound. The error bound of SCCI model recovery when ROMP algorithm is used for sparse recovery is equal to:

$$\begin{gathered}\|\hat{\mathbf{\Theta}}-\mathbf{\Theta}\|_2 \leq C\sqrt{\log v}\left(\|\epsilon\|_2+\tfrac{1}{\sqrt{v}}\left(\textstyle\sum_{k=1}^{N}|z_c[k]-z_{c,v}[k]|\right.\right.\\ \left.\left.+\textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{N}|w_{c,\gamma}[k]-w_{c,\gamma,v}[k]|+\sum_{j=1}^{J}\sum_{k=1}^{N}|w_j[k]-w_{j,v}[k]|\right)\right),\end{gathered}\tag{4.25}$$

where C is a constant, and ϵ is the noise, which is zero in the noiseless situation.

Theorem 2: Assume that $\mathbf{X}=\Phi\mathbf{\Theta}$ is a vector of J jointly sparse signals following the signal model in Eq. 4.7, where Φ is the dictionary built from ϕ_c, ψ_is, and ϕ_js. Then, the error made by the hierarchical DCS recovering the jointly sparse representation of J signals using ROMP algorithm is bounded by:

$$\begin{gathered}\|\hat{\mathbf{\Theta}}-\mathbf{\Theta}\|_2 \leq C\sqrt{\log v}\left(\|\epsilon\|_2+\tfrac{1}{\sqrt{v}}\left(\textstyle\sum_{k=1}^{N}|z_c[k]-z_{c,v}[k]|\right.\right.\\ \left.\left.+\textstyle\sum_{k=1}^{N}|w_{c,\gamma}[k]-w_{c,\gamma,v}[k]|+\tfrac{1}{2}\sum_{j=1}^{J}\sum_{k=1}^{N}|w_j[k]-w_{j,v}[k]|\right)\right),\end{gathered}\tag{4.26}$$

where C is a constant, and ϵ is the noise, which is zero in the noiseless situation [128].

Proof:

The upper error bound for the γth subset is:

$$\begin{gathered}\|\hat{\mathbf{\Theta}}_\gamma-\mathbf{\Theta}_\gamma\|_2 \leq C\sqrt{\log v}\left(\|\epsilon\|_2+\tfrac{1}{\sqrt{v}}\left(\textstyle\sum_{k=1}^{N}|z_c[k]-z_{c,v}[k]|\right.\right.\\ \left.\left.+\textstyle\sum_{k=1}^{N}|w_{c,\gamma}[k]-w_{c,\gamma,v}[k]|+\sum_{j\in\gamma th\ subset}\sum_{k=1}^{N}|w_j[k]-w_{j,v}[k]|\right)\right).\end{gathered}\tag{4.27}$$

After calculating the sparse representation of Γ subsets, the last l_1-norm optimization step finds the common part among their outputs. The recovery error from this last step is calculated similarly as follows:

$$\begin{gathered}\|\hat{\mathbf{\Theta}}_c-\mathbf{\Theta}_c\|_2 \leq C\sqrt{\log v}\left(\|\epsilon\|_2+\tfrac{1}{\sqrt{v}}\left(\textstyle\sum_{k=1}^{N}|z_c[k]-z_{c,v}[k]|\right.\right.\\ \left.\left.+\textstyle\sum_{\gamma=1}^{\Gamma}\sum_{k=1}^{N}|w_{c,\gamma}[k]-w_{c,\gamma,v}[k]|\right)\right).\end{gathered}\tag{4.28}$$

The total recovery error of H-SCCI model recovery calculated using ROMP is bounded by:

$$\begin{gathered}\|\hat{\mathbf{\Theta}}-\mathbf{\Theta}\|_2 \leq C\sqrt{\log v}\left(\|\epsilon\|_2+\tfrac{1}{\sqrt{v}}\left(\textstyle\sum_{k=1}^{N}|z_c[k]-z_{c,v}[k]|\right.\right.\\ \left.\left.+\textstyle\sum_{k=1}^{N}|w_{c,\gamma}[k]-w_{c,\gamma,v}[k]|+\tfrac{1}{2}\sum_{j=1}^{J}\sum_{k=1}^{N}|w_j[k]-w_{j,v}[k]|\right)\right).\end{gathered}\tag{4.29}$$

□

4.3.6 Numerical Performance Evaluation

In this section, we present the numerical evaluation of the proposed method to show the improvement of Hierarchical SCCI model recovery with respect to SCCI model recovery. A real-world scenario introduced in [109] is used here to compare the computational cost and the reconstruction error of H-SCCI and SCCI model recovery, numerically.

In order to generate a set of sparse signals with inter-signal correlation, the common and innovation components of the joint sparsity model (Eq. 4.1) are created separately. To create the sparse common component, the location of the nonzero entries of the coefficient vector, θ_c, are selected randomly with the sparsity $K_c \sim Binomial(N, p_c = \frac{1}{5})$. The value of the selected nonzero coefficients are chosen from $N(\mu, \sigma)$. DCT bases is chosen as the common component bases, $\Phi_c \in \mathbb{R}^{N \times N}$. Then, the common component is obtained through $\mathbf{z}_c = \Phi_c \theta_c$. Similarly, the location of the nonzero entries of the coefficient vectors of the innovation components, $\theta_j, \forall j$, are chosen randomly with the sparsity $K_j \sim Binomial(N, p_j = \frac{1}{5})$. The value of nonzero coefficients are chosen from $N(\mu, \sigma)$. The $N \times N$ Haar basis is selected as the innovation component dictionary Φ_j. The jth signal in the collection is then constructed as $x_j = \Phi_c \theta_c + \Phi_j \theta_j$.

In the first experiment, the length of the signal, N, is kept constant and equal to 128 samples. In order to consider the effect of the number of signals, J, on the computational cost, the number of signals, J, is selected from $\{8, 16, 24, 32, 40, 48, 56, 64, 72, 80\}$. SCCI model is used to find the common and innovation components of J jointly distributed signals. In order to use the H-SCCI model, the number of subsets, Γ, is selected regardless of the value of J such that $\frac{J}{\Gamma} = 8$. The experiment is repeated 1000 times. The computational cost and error are calculated for both methods. Fig. 4.2 shows the mean +/- the standard deviation of the computational time required to calculate SCCI and H-SCCI models. Fig. 4.3 shows the mean +/- the standard deviation of mean squared error (MSE) of both SCCI and H-SCCI model recoveries. As it can be seen, the computational cost of SCCI model recovery grows exponentially as the number of signals increases. Yet, the H-sCCI model recovery requires less amount of time with smaller standard deviation. On the other hand, MSE of H-SCCI model recovery is becoming saturated, and in this experience in the worst case the MSE differences between two methods is 9×10^{-4}. Thus, if there is no restriction on the time to calculate the joint sparsity model and any memory limitations, SCCI model recovery works better since MSE is smaller. However, for larger values of J, the hierarchical implementation recovers the SCCI model much faster at the cost of slightly increased MSE.

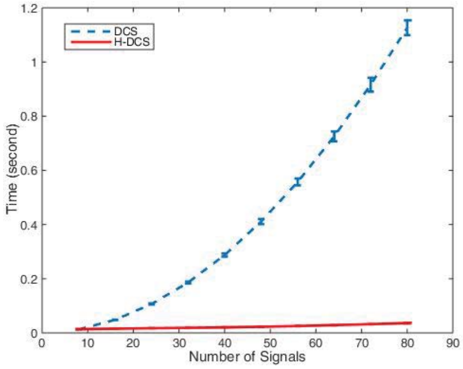


Figure 4.2 The mean +/- the standard deviation of the computational time required to calculate SCCI and H-SCCI models when the bases of the common components is DCT and the bases of the innovation components is Haar.

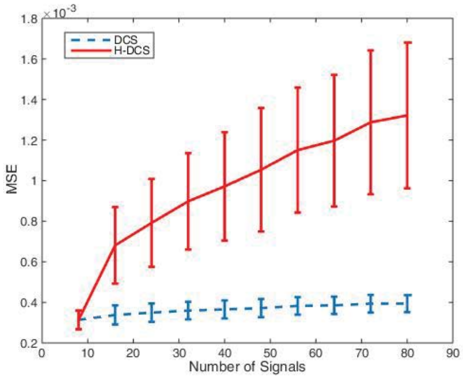


Figure 4.3 The mean +/- the standard deviation of MSE of both SCCI and H-SCCI model recovery when the bases of the common components is DCT and the bases of the innovation components is Haar.

In the second experiment, J jointly sparse signals are generated in the same way as described in the first experiment. However, the common and innovation bases are chosen to be identical and equal to $N \times N$ DCT. The computational cost and MSE are calculated for both SCCI and H-SCCI model recoveries. Fig. 4.4 shows the mean +/- the standard deviation of the computational time required to calculate SCCI and H-SCCI models. Fig. 4.5 shows the mean +/- the standard deviation of MSE of both algorithms.

Apparent from Fig. 4.2 and 4.4, the computational time of SCCI model recovery is larger on average in addition to its standard deviation in the second experiment compared to the first one. Similarly, the mean and standard deviation of MSE for SCCI model recovery decrease for the second experiment while they stay the same for H-SCCI model recovery. Therefore, in situations where the common and innovation components can be assumed to be sparse with respect to the same set of bases, the performance of H-SCCI model recovery improves more compared to SCCI model in terms of the computational time and reconstruction error.

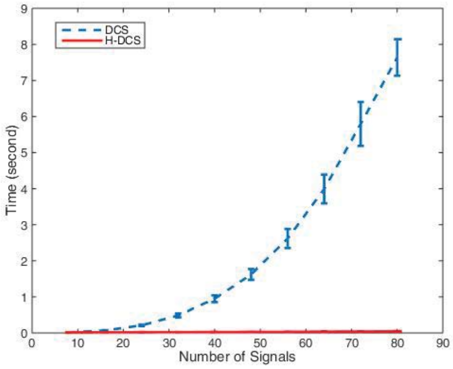


Figure 4.4 The mean +/- the standard deviation of the computational time required to calculate SCCI and H-SCCI models when the bases of the common components and the innovation components are identical (DCT).

In the third example, we investigate the effect of the noise on H-SCCI model recovery and compare the results with SCCI model recovery. J jointly sparse signals are generated in the same way as described in the first and second experiments. The common and innovation

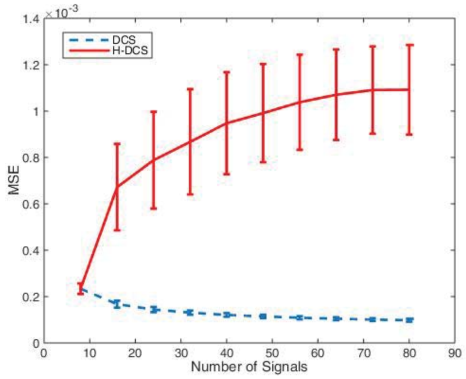


Figure 4.5 The mean +/- the standard deviation of MSE of both SCCI and H-SCCI model recoveries when the bases of the common components and the innovation components are identical (DCT).

bases are chosen to be identical and equal to $N \times N$ DCT. However, the number of signals is fixed and equal to $J = 16$ in this experiment so that the only variable is the signal-to-Noise ration (SNR). The experiment is performed for 10 different SNR values and repeated 1000 times for each SNR value. Fig. 4.6 and Fig. 4.7 show the computation time and MSE values of SCCI and H-SCCI model recoveries for different SNR values. H-SCCI model recovery outperforms SCCI model recovery in terms of computational time for all SNR values. On the other hand, MSE of SCCI model recovery is less than that of H-SCCI model recovery for larger values of SNR whereas MSE of SCCI model recovery becomes larger than that of H-SCCI model recovery with the decrease in the SNR value. When SNR values are large, SCCI model is able to recover the sparse representation of the original signals while H-SCCI model introduces some noise into the recovered signals. However, as SNR values decreases, H-SCCI model recovery outperforms SCCI model recovery since H-SCCI model first focuses on the recovery of the sparse representation of the common component. In the first level of the implementation, H-SCCI model recovery method filters out the innovation components and a part of the noise. Then, in the second stage, H-SCCI model filters out the remaining noise and innovation components. Once the common component is recovered, in the final stage, H-SCCI model recovers the innovation components by filtering out the noise through the sparse recovery. Thus, H-SCCI model consists of three levels of sparse approximation, each acting as a noise filtering by finding the low-dimensional representation of the signals.

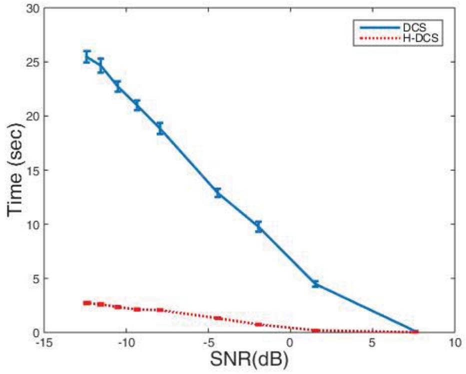


Figure 4.6 The mean +/- the standard deviation of the computational time required to calculate SCCI and H-SCCI models when the bases of the common components and the innovation components are identical (DCT) and the signals are contaminated by noisy.

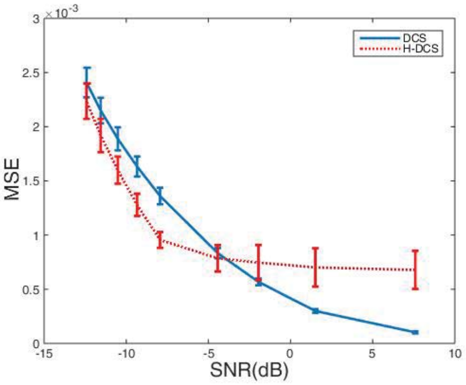


Figure 4.7 The mean +/- the standard deviation of MSE for both SCCI and H-SCCI model recoveries when the bases of the common components and the innovation components are identical (DCT) for noisy signals with varying SNR values.

4.4 Applications

In this section, the proposed algorithm is used for two different applications: functional connectivity network summarization, and video background extraction.

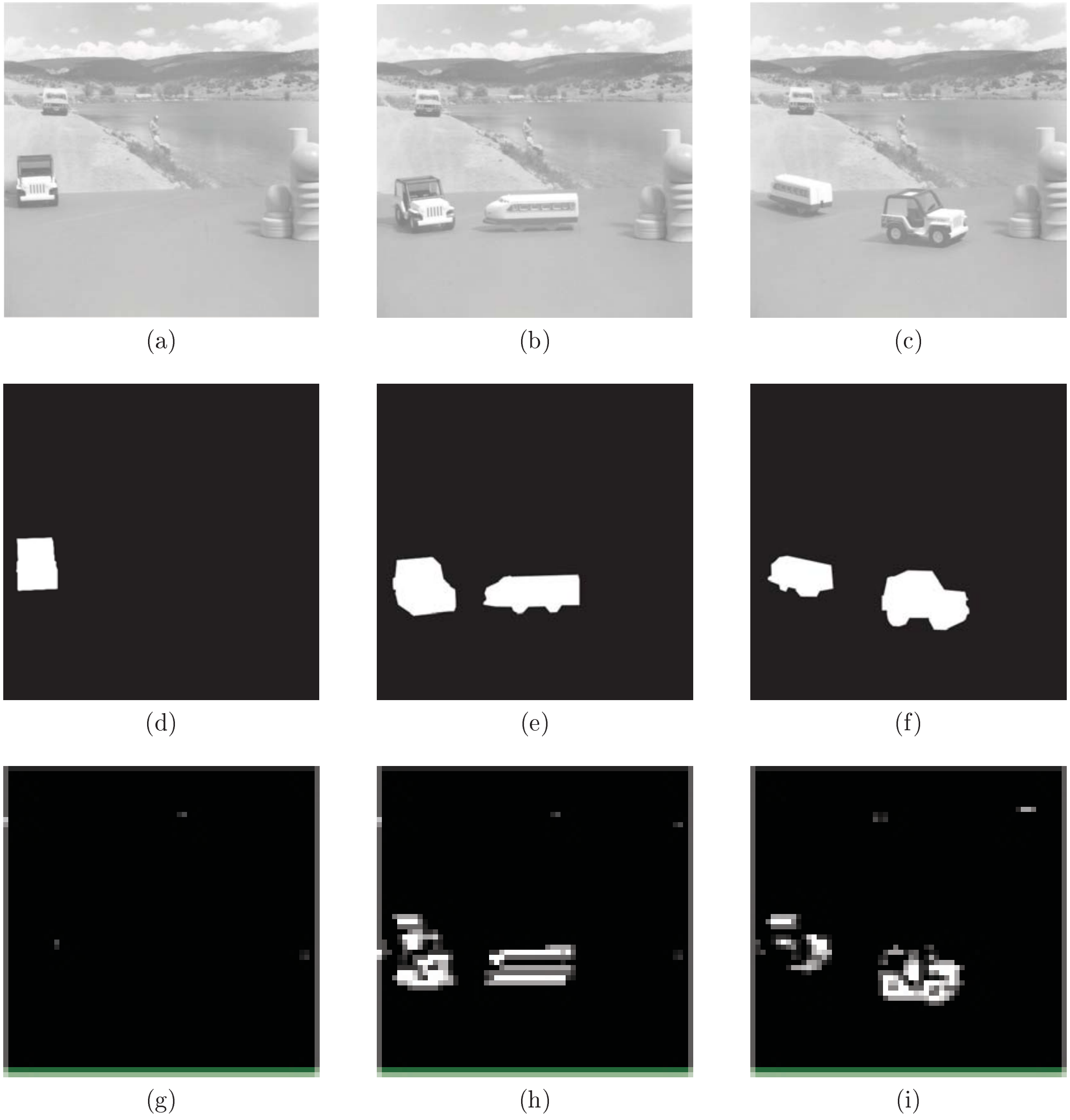
4.4.1 Video Background Extraction

Background extraction plays an important role in most computer vision and video processing applications such as surveillance video coding [119], motion detection, object tracking, *e*tc. Video background extraction aims to separate the moving objects, *a*.k.a. foreground, from the static objects, *a*.k.a. background, in order to facilitate analyzing the trajectory of moving objects. Even though the subsequent steps of the background extraction are of more interest, the accuracy of the overall system depends on the performance of the background extraction. Over the last few years, background extraction has been improved. Per-pixel Gaussian mixture model [129] combines the space of depth and luminance-invariant color to make a model of the background over time. The problem with the slow convergence of Gaussian mixture model is addressed by incorporating the incremental Expectation Maximization (EM) type of learning into a recursive filter [130]. The visual background extractor (ViBe) [1] uses a random policy to estimate the sample-based background.

Recently, SCCI model has been used to extract the background scene which is the common component of all frames in order to compress the video more efficiently [110]. Do *et al.* [111] designed a real-time video coding system based on distributed compressive sensing. Here, we use the H-SCCI model to extract the background scene from a sequence of images in order to find moving objects. The proposed H-SCCI model recovery algorithm overcomes restrictions such as time and memory, which helps to extract the background scene of a larger sequence of images.

4.4.1.1 Implementation

In the first experiment, a sequence of 9, 512 × 512-images of toy vehicles is chosen for background extraction purposes [131], where the images consist of a common background scene and moving objects. The images are downsampled to 80×80 due to computation time and memory limitations. All 9 images are vectorized, $\Lambda = \{\mathbf{x}_j \in \mathbb{R}^{6400}; \forall j \in \{1, 2, \ldots, 9\}\}$

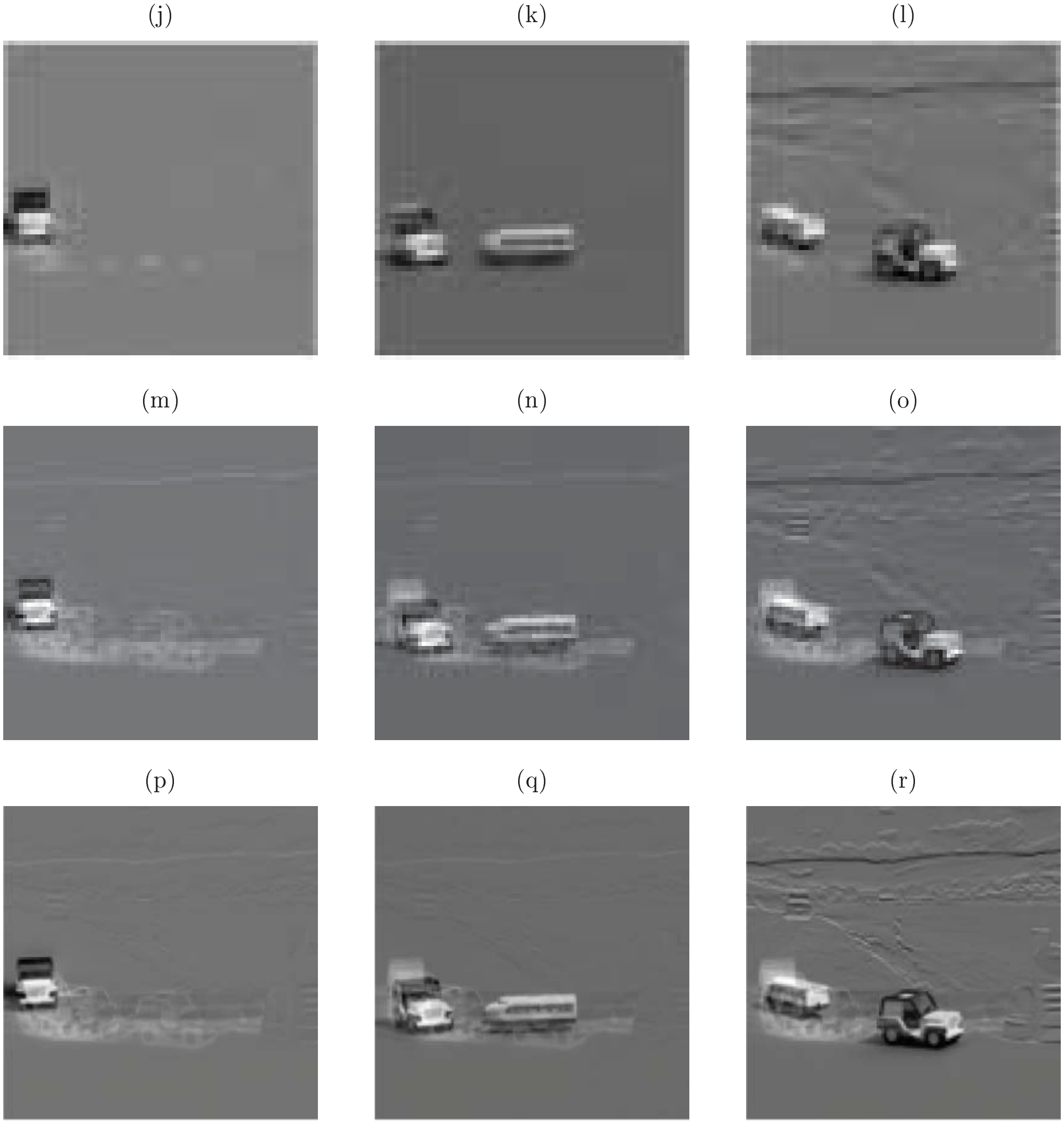

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.8 The toy vehicle experiment: first column shows three original frames of the sequence, the second column is the ground truth for the moving objects, the third column is the corresponding results obtained by ViBe [1] with frames resized to 80×80, the fourth column is the detected moving objects using SCCI model with frames resized to 80×80, the fifth column is the detected moving objects using H-SCCI model with frames resized to 80×80, the sixth column is the detected moving objects using H-SCCI model with frames resized to 128×128.

Figure 4.8 (cont'd)



(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

with $\mathbf{x}_j = \mathbf{b}_c + \mathbf{f}_j$ where $\mathbf{b}_c$ is the background scene and $\mathbf{f}_j$ is the moving object corresponding to the jth frame. The 6400×6400 identity matrix $\mathbf{I}_{6400}$ is used as the bases for the common and innovation components (background and foreground), $\phi_c = \phi_j = \mathbf{I}_{6400}$. The background of the set of images, Λ, is extracted first using SCCI model.

The same set of 80×80 images is then used to evaluate the performance of the H-SCCI model recovery algorithm. The set of 9 images are partitioned into $\Gamma = 3$ sets by placing the consecutive images into the same subset. The identity matrix is selected as the bases for the common and innovation components, $\phi_c = \phi_j = \mathbf{I}_{6400}$. The common component of each subset is extracted separately through the $l1$-norm optimization problem (Eq. 4.10), yielding $\mathbf{u}_\gamma, \gamma \in \{1, 2, 3\}$. We use the common components of these three subsets to extract the global common component, $\mathbf{b}_c$, solving another $l1$-norm optimization problem.

H-SCCI model recovery is also applied to the same sequence of images downsampled to 128×128. All images are vectorized to form the signal set $\Lambda = \{\mathbf{x}_j \in \mathbb{R}^{6400}; \forall j \in \{1, 2, \ldots, 9\}\}$. Every three consecutive images are grouped into one subset. Fig. 4.8 shows three frames from the original sequence and the foreground extracted by SCCI model and H-SCCI model for different image sizes. The SCCI model recovery is not capable of extracting the background from images with size 128×128 due to memory and computational time requirements. The downsampled background obtained from H-SCCI model is compared with the background extracted using SCCI model. Table 4.2 shows the computational time and peak signal to noise ration (PSNR) for this set of experiments. In order to calculate PSNR, the background extracted with SCCI model is considered as the groundtruth and compared with that extracted from H-SCCI model. However, in the case of 128×128 set of images, SCCI model recovery algorithm is not capable of extracting the background of the images due to memory and computational time restrictions. Thus, the downsampled background obtained from H-SCCI model is compared with the background extracted using SCCI model in order to calculate PSNR. By comparing the third and fourth rows of Fig. 4.8, we can see that there is some misidentification of the moving objects in the H-SCCI model, which can be explained through the larger error bound of H-SCCI model recovery. In the fourth row, there are some artifacts or distortion between the two vehicles, mostly due to the different location of the vehicles. These distortions show the movement of the vehicles in the sequence of the images. These distortions do not exist in the foreground detected using SCCI model. However, as the dimension of the images grows, the distortion from H-SCCI model decreases

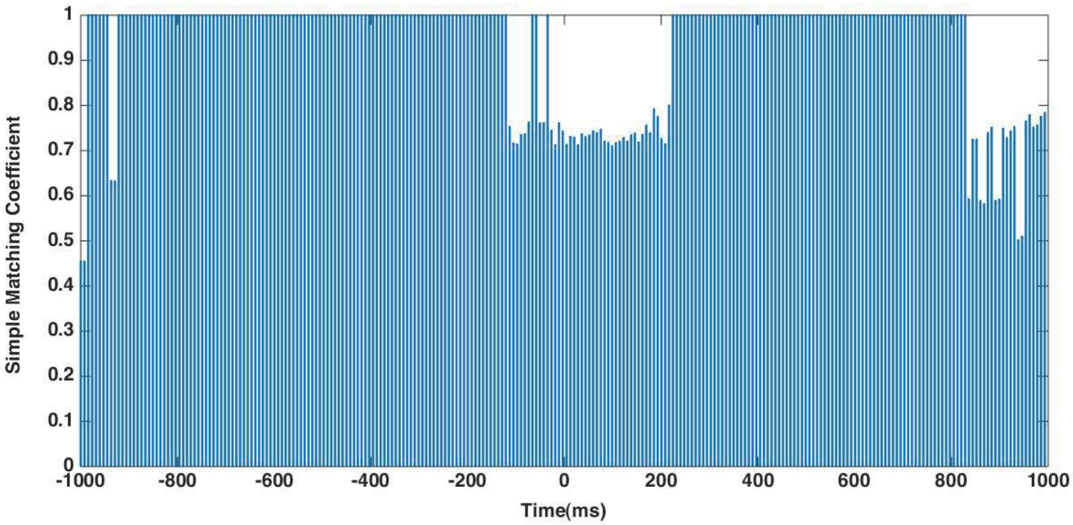


Figure 4.9 Simple Matching Coefficient: similarity between the support of the coefficients of the consecutive $\hat{\mathbf{g}}(t)$.

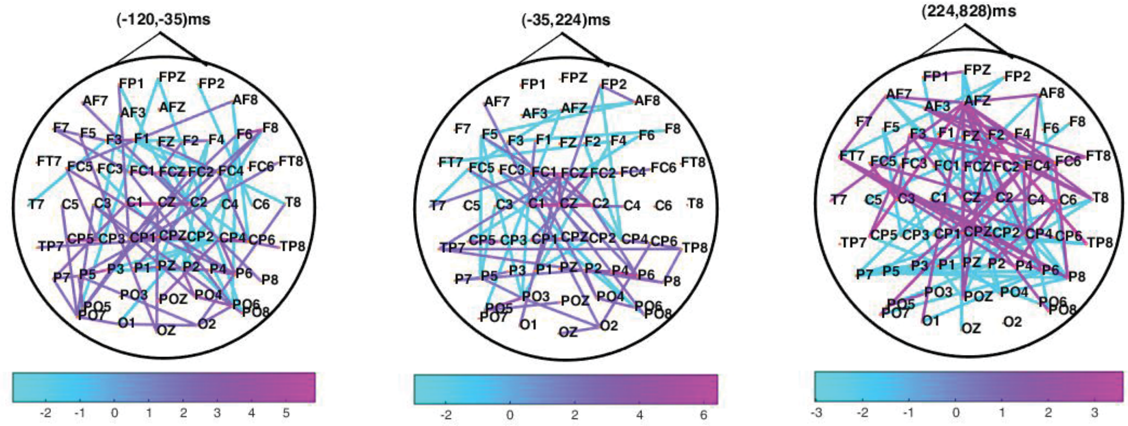
(the fifth row of Fig. 4.8) since there are a larger number of coefficients in $\boldsymbol{\Theta}$ which allows the optimization problem to find a better combination of coefficients to reduce the error. This explains the improvement in MSE and PSNR in Table 4.2. Note that the distortion can also be decreased through tightening the conditions of the optimization problem, which will slightly increase the convergence time.

Table 4.2 The computational cost and PSNR for the toy-vehicle experiment

Method	Computational Cost (min)		PSNR (dB)
	H-SCCI model	SCCI model	
80 × 80 frames	28	141	33.33
128 × 128 frames	76	-	36

4.4.2 Functional Brain connectivity Network State Representation

In this chapter, we propose to use H-SCCI model recovery to track dynamic FC networks constructed from multichannel EEG data collected during a study of Error-related negativity (ERN). The dynamic network is converted to signals or vectors at each time point and the



(a)

(b)

(c)

Figure 4.10 The significant edges for the time intervals: (a) $(-120, -35)$ms, (b) $(-35, 224)$ms, (c) $(224, 828)$ms, obtained through z-test with $p = 0.95$.

H-SCCI model framework is applied to this collection of signals to first identify change points in connectivity and then extract the common sparse components for network summarization.

4.4.2.1 Implementation

The dynamic connectivity networks across time are assumed to follow a time-varying signal plus background model:

$$\mathbf{G}(t) = \mathbf{B} + \hat{\mathbf{G}}(t), \tag{4.30}$$

where $\mathbf{B} \in \mathbb{R}^{62\times 62}$ is the background FC pattern common across all time steps, and $\hat{\mathbf{G}}(t) \in \mathbb{R}^{62\times 62}$ is the innovation component of the connectivity matrix at time t. H-SCCI model is used to find the background FC pattern, $\mathbf{B}$, and the innovation connectivity matrices, $\hat{\mathbf{G}}(t), t \in \{1, 2, \dots, 256\}$. First, the upper triangular part of the connectivity matrices are vectorized, as:

$$\mathbf{g}(t) = \mathbf{b} + \hat{\mathbf{g}}(t), \tag{4.31}$$

where $\mathbf{g}(t), \mathbf{b}, \hat{\mathbf{g}}(t) \in \mathbb{R}^{1953\times 1}$. The set of 256 signals, $\mathbf{g}(t), t \in \{1, 2, \ldots, 256\}$, is partitioned to $\Gamma = 10$ subsets, $\Lambda_1 = \{\mathbf{g}(1), \ldots, \mathbf{g}(25)\}$, $\Lambda_1 = \{\mathbf{g}(26), \ldots, \mathbf{g}(50)\}$, ..., and $\Lambda_{10} = \{\mathbf{g}(226), \ldots, \mathbf{g}(256)\}$. Discrete Cosine Transform (DCT) basis is used for the common component, $\phi_c \in \mathbb{R}^{1953\times 1953}$, while Haar wavelet basis is selected for the innovation component, $\phi_j \in \mathbb{R}^{1953\times 1953}$. The MATLAB package "*l1benchmark*" is used to solve the optimization problem in Eq. 4.10, where $\mathbf{z}_c = \mathbf{b}$, $\mathbf{x}_j = \mathbf{g}(t)$, and $\mathbf{z}_j = \hat{\mathbf{g}}(t)$. Once the background FC pattern is determined, Eq. 4.11 is used to obtain the innovation components, $\hat{\mathbf{g}}(t)$.

Change Point Detection: The innovation components, $\hat{\mathbf{g}}(t)$, are used to detect the change points. To achieve this goal, the support of $\hat{\mathbf{g}}(t)$, i.e. the indices of the non-zero coefficients with respect to the basis of representation, is tracked over time. The FCNs are assumed to be quasi-stationary within a short time interval [52, 56]. Therefore, the support of the innovation components' coefficients should not change much within each time interval. Any change in the support of the coefficients between consecutive time samples is considered as a change point. Let the vector $\tau(t) \in \mathbb{R}^{1953\times 1}$ be a binary vector such that the ith element $\tau_i(t) = 1$ if $\hat{\theta}_{j,i}(t)$ ($\hat{\theta}_{j,i}(t)$ is the ith element of the vector $\hat{\theta}_j$ in Eq. 4.11) is nonzero, and $\tau_i(t) = 0$ otherwise. In order to detect the change points, the similarity between the supports of the coefficients of the consecutive time samples is quantified using the Simple Matching Coefficient (SMC) [132]. SMC is a similarity measure between binary vectors, which takes into account both zero and one attributes. Fig. 4.9 shows the SMC similarity between consecutive $\tau(t)$s.

From the plot, we can identify several distinct time intervals: $(-1000, -921)$ms, $(-921, -120)$ ms, $(-120, -35)$ ms, $(-35, 224)$ ms, $(224, 828)$ ms, and $(828, 1000)$ ms. The time interval $(-35, 224)$ ms includes the ERN response. During error processing (ERN time interval), the brain is in the process of forming transient frontal networks, which causes the decrease in the similarity of the support of the coefficients of the innovation components of FCNs as the network is changing.

Network Summarization: Once the time intervals are determined, the FC topographic map of each time interval needs to be obtained. FC topographic map consists of the significant functional connections consistent across the detected time interval, (t_0, t_1). In order to get the FC topographic map, we apply H-SCCI model recovery algorithm to $\hat{\mathbf{g}}(t), t \in (t_0, t_1)$. Thus, the connectivity vectors are written as:

$$\hat{\mathbf{g}}(t) = \pi + \check{\mathbf{g}}(t), \ \ t \in (t_0, t_1), \tag{4.32}$$

where π is the common part of the time interval (t_0, t_1) we are interested in, and $\check{\mathbf{g}}(t)$ represents the innovation activity at time t.

The signal model assumed for the connectivity matrices within the time interval in Eq. 4.32 is the SCCI model (Eq. 4.3), where $\mathbf{z}_c = \pi$, $\mathbf{x}_j = \hat{\mathbf{g}}(t)$, and $\mathbf{z}_j = \check{\mathbf{g}}(t)$. The resulting optimization problem (Eq. 4.10) for $\hat{\mathbf{g}}(t), t \in (t_0, t_1)$ is solved, yielding the common pattern of FCs across time within the time interval, π. Since the background FC pattern is eliminated in the first step (see Eq. 4.31), π is the common FC pattern specific to the time interval. Since the distribution of the common FC patterns is Gaussian, z-test determines the most significant edges. Fig. 4.10 shows the most significant FCs for three time intervals, $(-120, -35)$ms, $(-35, 224)$ms, and $(224, 828)$ms, determined through z-test with $p = 0.95$. Fig. 4.10(b) corresponds to ERN time interval showing the significant connections between medial-prefrontal and medial-central regions engaging electrodes FC1, FCz, FC2, C1, Cz, and C2. This pattern is consistent with widely observed ERN amplitude measures [95, 96, 55]. There are also some connections between occipital-parietal and frontal regions. Fig. 4.10(a) corresponds to the pre-ERN time interval. In this state, the brain is preparing to respond to the external stimulus with connections distributed across the scalp. There are also some connections among FC1, FCz, and FC2 in addition to the connections among C1, Cz, and C2. Fig. 4.10(c) corresponds to the error-related positivity (Pe) and P300, showing strong connections between frontal and parietal regions. Broadly, the network states identified before, during, and after the ERN are consistent with hypothesized ERN-related network dynamics, focused on the formation of transient frontal networks during error processing.

4.5 Conclusions

Joint sparse recovery approximates a set of input signals simultaneously through linear combinations of dictionary elements. The joint sparse recovery algorithm uses the minimum number of bases while it keeps the error between the approximation and the input signals small. In this chapter, we proposed a hierarchical recovery of SCCI model to jointly recover the set of signals. The proposed method is evaluated on two different datasets, EEG functional connectivity networks and video background extraction. H-SCCI model is able to detect the change points and find the common connectivity pattern of each time interval. The results are consistent with the hypothesized ERN-related network dynamics, focused on the formation of transient frontal networks during error processing followed by parietal networks. The experimental results on video data illustrate the reduced computational cost with some increased distortion in the recovered foreground objects.

CHAPTER 5

STRUCTURED DICTIONARY LEARNING FOR SPARSE COMMON COMPONENT AND INNOVATION MODEL

5.1 Introduction

In SCCI, the sparsity of the two parts of the signals is usually described through predetermined dictionaries. However, as the literature in dictionary learning indicates, predetermined dictionaries are not necessarily optimal [133]. The well-known dictionary learning algorithms such as KSVD and method of optimal directions (MOD) [134] cannot be directly used as they do not guarantee the preservation of the special structure of the SCCI dictionary. Recently, Chen *et al.* [135] proposed Joint Orthogonal Matching Pursuit (JOMP) with dictionary update for SCCI. In each iteration, only one atom for representing either the common or the innovation components is selected based on the correlation between the residue and the atoms of the input dictionary. However, when combined with dictionary learning, this method only learns a single dictionary for the common and innovation components, where some atoms belong to the common component and others belong to the innovation components. As a result, when JOMP recovers the sparse representation of the test data with respect to this dictionary, it may select the atoms learned for the innovation component to represent the common component and vice versa.

In this chapter, we propose a two-stage dictionary learning algorithm which addresses these problems by learning two separate dictionaries for the common and innovation components simultaneously. In the proposed method, structured KSVD, after finding the coefficient vectors, first the common component dictionary is updated. Then, the innovation component dictionary is updated using the coefficient vector and the training data. The procedure to update the common and innovation dictionaries are similar to KSVD, where the updated atom is the singular vector of the error matrix. Unlike KSVD, the structured KSVD breaks down the main dictionary to the common and innovation dictionaries and learn these subdictionaries.

In this chapter, the structured KSVD is used to learn the common and innovation dictionaries for Error-related potentials (ERPs) from training data in order to extract ERP from the signals recorded by the neighboring electrodes because ERPs are usually similar across neighboring electrodes due to volume conduction. Therefore, ERP extraction from background EEG activity can be performed by extracting the common component of the set of neighboring electrodes using SCCI model. Previously, ERPs have been used extensively for the detection of neural diseases and the design of brain computer interface (BCI) [136, 137]. The difficulty with extracting ERPs is that their amplitude is small compared to the background activity. Different approaches such as averaging, independent component analysis (ICA), and wavelets [138] have been proposed [139]. Since ERPs are time-locked, it is assumed that averaging will add-up the effect of ERP while attenuating the background activity. ICA is used to separate the evoked potential from the background activity using the signals recorded by nearby electrodes [139] assuming that ERP and the background activity are statistically independent, which is not necessarily true. Wavelet based approaches, on the other hand, identify the coefficients associated with ERPs statistically [138]. This method requires selecting the value of the threshold, and assumes ERP is in a subspace of wavelet space. ERPs are also extracted [140] by principal component analysis with the assumption that after the averaging the variance of the background activity decreases while the variance of ERP has a direction which can be found using PCA. This assumption is true only if the background activity is actual noise whereas it is another brain event which is not of interest. The proposed algorithm is applied to ERN and the correct-related negativity (CRN) signals. ERN and CRN are ERPs that are commonly used to index cognitive control [?]. The distance between the common dictionaries of ERN and CRN shows that these two responses are sparse with respect to two different set of bases. Finally, the learned dictionaries are used to extract the two ERPs corresponding to different responses.

5.2 KSVD Algorithm

In traditional sparse signal recovery, predetermined basis sets such as Fourier, wavelets or their combinations are commonly used to find the sparse representation of signals. However, in many applications, the dictionary representing the sparse representation of the signal is not known. Dictionary learning (DL) has been developed to address this problem [141, 133]. Current dictionary learning algorithms such as KSVD [133] and MOD [134] focus on signals that have a common sparse representation, and not on SCCI model described in Chapter 4.

Dictionary learning algorithms try to find the sparse coefficient vectors $\mathbf{A} = [\alpha_1, \alpha_2, \ldots, \alpha_J]$ and the dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_L]$ for a set of training signals $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J]$ in the model $\mathbf{Y} = \mathbf{DA}$ such that:

$$\min_{\mathbf{A},\mathbf{D}} \ \{\|\mathbf{Y} - \mathbf{DA}\|_F^2\} \quad \text{s. t. } \forall i \ \|\alpha_i\|_0 \leq K \tag{5.1}$$

where K is the sparsity level of the coefficient vectors, $\mathbf{y}_j \in \mathbb{R}^N$, $\alpha_j \in \mathbb{R}^L$, $\mathbf{Y} \in \mathbb{R}^{N \times J}$, $\mathbf{D} \in \mathbb{R}^{N \times L}$, $\mathbf{A} \in \mathbb{R}^{L \times J}$, and $L \geq N$.

KSVD minimizes Eq. 5.1 iteratively by first finding the coefficient vectors, $\mathbf{A}$, while the dictionary, $\mathbf{D}$, is fixed and then updating $\mathbf{D}$. After initializing the dictionary $\mathbf{D}^{(0)}$, in the first stage, KSVD uses a pursuit algorithm to find the sparse coefficient vector at each iteration k, $\alpha_j^{(k)}$, for each signal $\mathbf{y}_j, j \in \{1, 2, \ldots, J\}$ through solving the following optimization problem:

$$\min_{\alpha_j^{(k)}} \ \|\alpha_j^{(k)}\|_0 \ \text{s. t. } \|\mathbf{y}_j - \mathbf{D}^{(k)}\alpha_j^{(k)}\|_2^2 \leq \epsilon \ \ \forall j \in \{1, 2, \ldots, J\}. \tag{5.2}$$

Once the sparse coefficients are computed, KSVD updates each column of $\mathbf{D}^{(k)}$ at a time, fixing all columns in $\mathbf{D}^{(k)}$ except $\mathbf{d}_l^{(k)}$, and updating the lth column vector as $\mathbf{d}_l^{(k+1)}$. In order to update the lth column $\mathbf{d}_l$, ω_l is defined as the group of indices pointing to the training signals $\mathbf{y}_j$s that use the atom $\mathbf{d}_l$, which is equivalent to the nonzero entries of α_T^l, where α_T^l is the lth row of $\mathbf{A}$, as: $\omega_l = \{j | 1 \leq j \leq J, \alpha_T^l(j) \neq 0\}$. The overall representation error matrix $\mathbf{E}_l$ is computed as:

$$\mathbf{E}_l = \mathbf{Y} - \sum_{f \neq l} \mathbf{d}_f \alpha_T^f. \tag{5.3}$$

The matrix $\mathbf{E}_l^R$ is obtained by selecting the columns of $\mathbf{E}_l$ with the columns identified in ω_l. After applying SVD to the matrix $\mathbf{E}_l^R = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T$, the updated dictionary column $\mathbf{d}_l^{k+1}$ is equal to the first column of $\mathbf{U}$ whereas the updated coefficient vector α_R^l is the first column of $\mathbf{V}$ multiplied by $\boldsymbol{\Delta}(1,1)$. This process is repeated until the stopping criteria are met.

5.3 Common And Innovation Dictionaries Learning

The challenge with using dictionary learning algorithms such as KSVD in DCS is that they cannot be applied directly to Eq. 4.5. The dictionary in these equations, $\boldsymbol{\Phi} \in \mathbb{R}^{JN \times (J+1)L}$, is constructed from different sets of basis, $\phi_c \in \mathbb{R}^{N \times L}$ and $\phi_j \in \mathbb{R}^{N \times L}$[1]. Thus, some entities of the dictionary $\boldsymbol{\Phi}$ are constrained to zero (as it is depicted in Eq. 4.3). However, no dictionary learning algorithm guarantees to preserve this structure of $\boldsymbol{\Phi}$ during the update step of the dictionary. Therefore, we propose a structured dictionary learning algorithm based on K-SVD to fulfill this constraint and learn the dictionary $\boldsymbol{\Phi}$ from the training samples. Since there is one common component, $\mathbf{z}_c$ among the training signals and the common dictionary has to be updated based on this common component, we randomly select J training signals from the whole set of training signals, learn the dictionaries, and repeat the process to update $\boldsymbol{\Phi}$.

After initializing the bases ϕ_c and ϕ_j and constructing $\boldsymbol{\Phi}^{(k)}$ at the kth iteration, the coefficient vector $\hat{\boldsymbol{\Theta}}^{(k)} \in \mathbb{R}^{(J+1)L \times 1}$ is estimated, as:

$$\hat{\boldsymbol{\Theta}}^{(k)} = \arg\min \|\boldsymbol{\Theta}^{(k)}\|_0 \text{ s.}t. \ \|\mathbf{X} - \boldsymbol{\Phi}^{(k)}\boldsymbol{\Theta}^{(k)}\|_2^2 \leq \epsilon. \tag{5.4}$$

where $\mathbf{X} \in \mathbb{R}^{JN \times 1}$ is the input set of training signals as in Eq. 4.3.

[1] Note that we assume one unique ϕ_j for all J innovation components

After estimating the coefficient vector, dictionaries ϕ_c and ϕ_j are updated separately similar to KSVD. In order to update $\phi_c^{(k)}$, $\phi_j^{(k)}$ is assumed to be fixed and $\phi_c^{(k)} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_L]$. The vector of input signals $\mathbf{X} \in \mathbb{R}^{NJ\times 1}$ is rearranged as a matrix $\mathbf{Y} = [\mathbf{x}_1, \ldots, \mathbf{x}_J] \in \mathbb{R}^{N\times J}$, and the coefficient vector $\hat{\theta}_c \in \mathbb{R}^{L\times 1}$ for the common part is concatenated to form the matrix $\mathbf{A}_c^{(k)} = [\hat{\theta}_c, \ldots, \hat{\theta}_c] \in \mathbb{R}^{L\times J}$. Since all training signals have the same common component, they all either use the lth atom or not. Thus, $\omega_{c,l}$ is defined as a row vector of ones with size J, $\mathbf{1} \in \mathbb{R}^{1\times J}$ if the common component includes the lth atom of the common dictionary, or a row vector of zeros with size J, $\mathbf{0} \in \mathbb{R}^{1\times J}$. The error corresponding to the lth atom, $\mathbf{E}_{c,l} \in \mathbb{R}^{N\times J}$, is defined as:

$$\mathbf{E}_{c,l} = \mathbf{Y} - \sum_{f\neq l} \mathbf{d}_f \theta_{c,T}^{f}. \tag{5.5}$$

The matrix $\mathbf{E}_{c,l}^{R}$ is obtained by selecting the columns of $\mathbf{E}_{c,l}$ with the columns identified in $\omega_{c,l}$. $\mathbf{d}_l^{(k+1)}$ is the first eigenvector of $\mathbf{E}_{c,l}^{R} = \mathbf{U}_c \mathbf{\Delta}_c \mathbf{V}_c^{T}$ while its corresponding coefficient vectors are updated by replacing them with the first column of $\mathbf{V}_c$ multiplied by $\Delta_c(1,1)$.

Once all the columns of $\phi_c^{(k+1)}$ are updated, the second dictionary $\phi_j^{(k)}$ is updated in a similar manner assuming $\phi_c^{(k+1)}$ is fixed. The dictionary $\phi_j^{(k)} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_L]$, and the vector of coefficients $\{\hat{\theta}_j^{(k)}\}_{j=1}^{J}$ from $\mathbf{\Theta}^{(k)}$ is rearranged to $\mathbf{A}_j^{(k)} = [\hat{\theta}_1, \ldots, \hat{\theta}_J] \in \mathbb{R}^{L\times J}$. The set $\omega_{j,l} = \{i | 1 \leq i \leq J, \theta_{j,T}^{l}(i) \neq 0\}$ is formed to indicate the signals that use the lth atom of the innovation dictionary. The error matrix $\mathbf{E}_{j,l}$ is then computed based on the signals identified in $\omega_{j,l}$, as:

$$\mathbf{E}_{j,l} = \mathbf{Y} - \sum_{f=1, f\neq l}^{N} \mathbf{c}_f \theta_{j,T}^{f}, \tag{5.6}$$

where $\theta_{j,T}^{f}$ is the fth row of $\mathbf{A}$. The matrix $\mathbf{E}_{j,l}^{R}$ is obtained by selecting the columns of $\mathbf{E}_{j,l}$ with the columns identified in $\omega_{j,l}$. The first eigenvector of $\mathbf{E}_{j,l}^{R}$ is considered as $\mathbf{c}_l^{(k+1)}$.

Once $\phi_c^{(k+1)}$ and $\phi_j^{(k+1)}$ are updated completely, the main dictionary $\mathbf{\Phi}^{(k+1)}$ is constructed again and the coefficient vector $\hat{\mathbf{\Theta}}^{(k+1)}$ is updated using Eq. 5.4. This process is repeated until convergence. The pseudocode of the structured dictionary learning for SCCI is presented in algorithm 1.

5.4 Experimental Results

5.4.1 Synthetic Data

The proposed structured dictionary learning is first evaluated on a set of synthesized signals. Two random matrices $\phi_c \in \mathbb{R}^{32\times 32}$ and $\phi_j \in \mathbb{R}^{32\times 32}$ are generated with i.i.d. uniformly distributed entries from $U(0,1)$ and $U(5,6)$, respectively. Each column is normalized to a unit l_2-norm. 150 signals $\{\mathbf{x}_i\}_{i=1}^{150}$, where $\mathbf{x}_i \in \mathbb{R}^{32\times 1}$, are produced by adding the common component $\mathbf{z}_c \in \mathbb{R}^{32\times 1}$ and the innovation components $\mathbf{z}_i \in \mathbb{R}^{32\times 1}$. The common component $\mathbf{z}_c$ is a linear combination of K_c atoms of ϕ_j, with uniformly distributed i.i.d. coefficients in random and independent locations. Similarly, the innovation components $\mathbf{z}_j$ are different linear combinations of K_j atoms of ϕ_j.

In order to apply KSVD to synthetic signals, the dictionaries ϕ_c and ϕ_j are initialized with the training data. After constructing $\mathbf{\Phi}$ and $\mathbf{X}$, the coefficient vector $\hat{\mathbf{\Theta}}$ is found using OMP with $K_c + (K_j \times 150)$ coefficients, where $K = K_c = K_j$ varying from 2 to 5. The maximum number of iterations was set to 20. The computed dictionaries $\hat{\phi}_c$ and $\hat{\phi}_j$ are compared separately against the generating dictionaries ϕ_c and ϕ_j. This comparison is done by sweeping through the columns of the generating dictionary ϕ_c (or ϕ_j) and finding the closest column (in l_2 distance) in the computed dictionary $\hat{\phi}_c$ (or $\hat{\phi}_j$). The distance is measured via $1 - |\mathbf{d}_i^T \hat{\mathbf{d}}_i|$, where $\mathbf{d}_i$ is an atom of ϕ_c (or ϕ_j) and $\hat{\mathbf{d}}_i$ is the corresponding atom in the learned dictionary. A distance less than 0.1 for ϕ_c and less than 0.05 for ϕ_j are considered a success. The simulations are repeated 500 times, and the number of successes in each simulation is computed. Fig. 5.1 displays the percentage of successfully recovered atoms for the common and innovation dictionaries. Overall, the percentage of recovered atoms of the common and innovation dictionaries are more than 90%, and as the sparsity level increases, the accuracy increases. However, the number of the recovered atoms of the common dictionary is less than that of the innovation dictionary since there are J different innovation coefficients $\{\theta_j\}_{j=1}^{J}$ to be used in updating the innovation dictionary while there

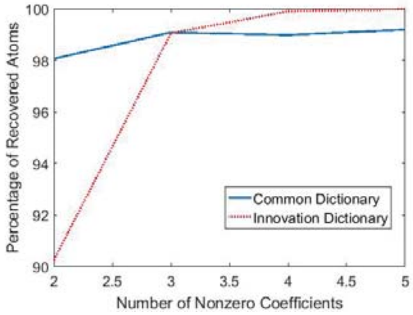


Figure 5.1 Percentage of successfully recovered atoms for the common and innovation dictionaries versus the number of sparse coefficients.

is only one common coefficient to be used in updating the common dictionary. Thus, the atoms of the common dictionary are updated based on only one common coefficient vector.

Algorithm 5.4.1 Structured Dictionary Learning for SCCI

```
Input: Initialize $\phi_c^{(0)}$ and $\phi_j^{(0)}$. Choose $\epsilon$. Set $k = 0$.
Output: The common component dictionary $\phi_c$, and the innovation component dictionary $\phi_j$.

while The convergence criteria is not met do
    Form the basis matrix ($\mathbf{\Phi}^{(k)} \in \mathbb{R}^{NJ\times(J+1)L}$) from $\phi_c^{(k)}$ and $\phi_j^{(k)}$. Form the signal vector ($\mathbf{X} \in \mathbb{R}^{JN\times 1}$) in Eq. 4.3 from the training signals.
    Solve $l1$-norm minimization in Eq. 5.4.
    for $l = 1{:}L$ do
        Compute the error matrix $\mathbf{E}_{c,l}$ using Eq. 5.5.
        Update $\mathbf{d}_l^{k+1}$ using the singular vector of $\mathbf{E}_{c,l}^R$.
    end for
    for $l = 1{:}L$ do
        Compute the error matrix $\mathbf{E}_{j,l}$ using Eq. 5.6.
        Update $\mathbf{c}_l^{k+1}$ using the singular vector of $\mathbf{E}_{j,l}^R$.
    end for
    k = k+1
end while
```

5.4.2 EEG Data

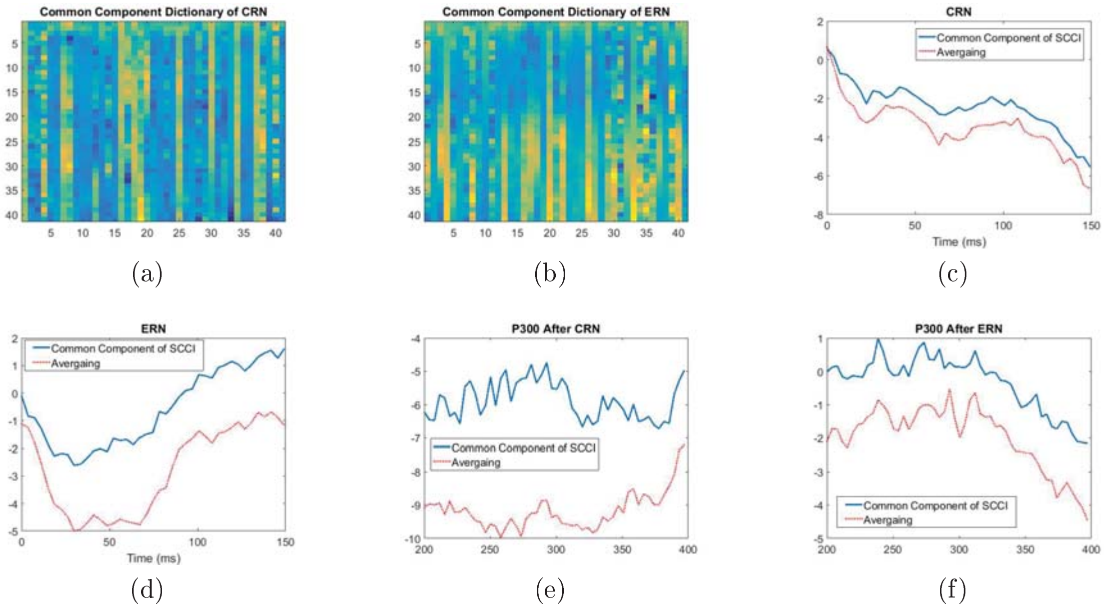
The proposed framework is applied to a set of EEG signals containing ERN and the CRN response, which are ERPs that are commonly used to index cognitive control [**?**]. The ERN is a brain potential response that occurs following performance errors in a speeded reaction time task usually 25-75 ms after the response [61]. In this study, we focus on the EEG recordings for 16 female participants. A speeded-response flanker task was employed, and response-locked averages were computed for each subject [142]. The brain activity is recorded during the error and correct processing using electroencephalogram with sampling rate equal to 256Hz with 1 second before the response and 2 seconds after the response. For each subject, EEG data over multiple trials (repetition of the same visual stimulus) is recorded (126 trials for ERN, and 1289 trials for CRN). The signals from 62-channels were collected in accordance with the 10/20 system on a Neuroscan Synamps2 system (Neuroscan, Inc.).

In this thesis, we assume that the electrodes that are close to each other will follow SCCI model due to volume conduction. Since we know most of the ERN and CRN activities to be localized around frontal-central brain regions [143], we select a time frame of $(0, 150)$ms corresponding to the ERP from electrodes F2, F4, F3, F5, and FCZ. We randomly select 10 trials from each electrode for CRN and average them to obtain the training dataset, and use another 10 and average them to get the test dataset. Similarly, the trials of each electrode of ERN are divided into two sets of training and test dataset. The number of training and test signals after averaging over trails for 16 subjects and 5 electrodes are 80. The training signals are used in the structured KSVD to learn the common and innovation dictionaries for ERN and CRN, separately. Fig. 5.2 (a) and (b) show the learned common component dictionaries for CRN and ERN, respectively. In order to evaluate how similar the learned dictionaries for CRN and ERN are, the distance $1 - |\mathbf{d}_i^T \hat{\mathbf{d}}_i|$ is computed. 12 atoms out of 41 atoms of the common components dictionary of ERN are close to that of CRN, which indicates the need to learn the dictionaries for ERN and CRN separately. The learned dictionaries are used to find the common component of the test CRN and ERN data. Fig. 5.2 (c) and (d) show the common component of CRN and ERN, respectively. The error processing waveform has a negative peak right after the response and becomes positive around 100ms after the response. However, the waveform from the correct response does not show the same negative peak (Fig. 5.2 (c)). Even though error and correct response waveforms are different, standard joint sparse representation methods use the same fixed dictionary such as discrete cosine transform (DCT) to extract them from the electrodes. We also compared the recovered ERPs to the ones obtained by averaging (Fig. 5.2 (c) and (d)). The recovered ERN using using averaging remains negative after 100ms (Fig. 5.2 (d)), and detecting the ERN boundaries becomes difficult.

Similarly, the same electrodes are used to extract P300, which starts around 200ms after the response and ends around 400ms after the response. P300 after CRN and ERN are expected to be different. Thus, the common and innovation dictionaries for P300 of CRN and ERN are learned from the set of training signals, separately. The learned dictionaries are used to extract the common component of the test data. Fig. 5.2 (e) and (f) show P300 extracted from the test data using the learned dictionaries. In comparison to averaging, DL can extract positive peaks in P300 after ERN which is aligned with literature and referred to as P3e [144].

5.5 Conclusion

In this chapter, we proposed to use the sparse common component and innovations model to extract ERPs from EEG recordings of closely located electrodes. Since responses to different stimuli have different characteristics, predetermined dictionaries are not suitable to represent and distinguish between them. We introduced a structured KSVD algorithm to simultaneously learn the common and innovation dictionaries. The proposed structured dictionary learning algorithm is used to learn the dictionaries representing the common component of ERN and CRN. The comparison of the learned dictionaries shows the differences between the dictionaries needed to represent ERN and CRN. Finally, the learned dictionaries are used to extract the ERPs corresponding to different responses. The results are consistent with previous findings about the waveforms of ERN and CRN.



(a) (b) (c)

(d) (e) (f)

Figure 5.2 (a) The dictionary learned to represent the common component of CRN, (b) The dictionary learned to represent the common component of ERN, (c) The common component of the CRN of the test data recovered using the learned dictionary and averaging, (d) The common component of the ERN of the test data recovered using the learned dictionary and averaging, (e) The common component of P300 after CRN of the test data is recovered using the learned dictionary and averaging, (f) The common component of P300 after ERN of the test data is recovered using the learned dictionary and averaging.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

With the advances in high resolution neuroimaging, there has been a growing interest in the study of functional brain connectivity. Most of the early studies of functional connectivity brain networks have focused on the computation of graph theoretic indices for static networks, i.e. long-time averages of connectivity networks. However, functional connectivity is a dynamic process and the construction and reorganization of the networks is key to understanding human cognition. In this thesis, methods to track the functional connectivity networks over time through detecting the quasi-stationary time intervals, called FC states, and summarizing the FC states have been developed.

In Chapter 3, tensor theory was used to represent dynamic functional connectivity networks across subjects. This representation offers several advantages over existing methods such as preserving the network structure and accounting for across subject variability. It also separates the information across different modes such as the connectivity and the time and subject modes from each other. The change points were detected by tracking the changes in the connectivity subspace of the consecutive time samples. The time intervals were then summarized through tensor-matrix projection to find the common pattern across subjects and time. This framework can be easily extended to higher order modes by considering the changes in connectivity across frequency or experimental conditions. By taking the frequency information into account, we may be able to identify change points that are sensitive to different frequency bands and further decompose the different network states into their frequency components. This formulation will follow the framework introduced in Chapter 3 with the addition of an extra mode thus increasing the computational complexity.

Chapter 4 introduced a new hierarchical implementation method of SCCI recover to obtain the common and innovation sparse components of a group of signals in a computationally efficient manner. The hierarchical SCCI recovery algorithm was shown to recover the common component much faster than the existing methods. The computational cost and error bound derived for the proposed method indicate the time efficiency of the proposed algorithm despite the higher error bound. The H-SCCI model recovery was used to extract the foreground network from the background activity. Since the innovation component uniquely defines each time step, the support of the innovation components were used to detect the boundaries of time intervals. By tracking the support of innovation components over time, change points were defined as the time points when the support of innovation components change. After identifying the change points, it was assumed that FC networks within each time interval follows the SCCI model. Thus, H-SCCI model recovery was used to extract the sparse common pattern for each time interval.

SCCI and H-SCCI model recovery algorithms introduced in Chapter 4 employed pre-determined dictionaries for the common and innovation component recovery. However, most signals are not sparse with respect to pre-determined dictionaries. Therefore, in Chapter 5, a structured KSVD algorithm was introduced to simultaneously learn the common and innovation dictionaries. The proposed structured dictionary learning algorithm was used to learn the dictionaries representing the common component of ERN and CRN in order to extract ERN and CRN from EEG recordings of closely located electrodes. Future work will consider the extension of the structured KSVD algorithm to hierarchical implementation of SCCI recovery. The hierarchical implementation of SCCI recovery consists of multiple stages of the common and innovation components recovery. In Chapter 4, the same pre-determined common and innovation dictionaries were used at all stages to recover the common component. However, the signals at different stages are different and may not be sparse with respect to the same set of dictionaries. The structured dictionary learning algorithm will be used to learn these dictionaries for each stage specifically. The learned dictionaries will increase the performance and accuracy of the hierarchical SCCI recovery. The structured KSVD can also be used to extract the common component of the functional connectivity networks within a time interval using H-SCCI model recovery as described in Chapter 4.

APPENDIX

APPENDIX

PRINCIPAL ANGLE

Let $\mathbf{U}^{(1)}(t) = [\mathbf{u}_1^{(1)}(t)\,\mathbf{u}_2^{(1)}(t)\ \ \ldots\ \mathbf{u}_{\hat{n}(t)}^{(1)}(t)]$ and $\mathbf{U}^{(1)}(t+1) = [\mathbf{u}_1^{(1)}(t+1)\,\mathbf{u}_2^{(1)}(t+1)\ \ \ldots\ \mathbf{u}_{\hat{n}(t+1)}^{(1)}\ (t+1)]$ span the subspaces of $\mathcal{G}(t)$ and $\mathcal{G}(t+1)$ along the first mode. Also, define $r = \min\ (\hat{n}(t),\ \ \hat{n}(t+1))$. Then, the kth principle vectors $\mathbf{p}_k \in \mathbf{U}^{(1)}(t)$ and $\mathbf{q}_k \in \mathbf{U}^{(1)}(t+1)$ for $k = 1, 2, \ldots, r$ are defined as a solution to the following optimization problem:

$$\begin{aligned} &\mathrm{m}aximize\ \mathbf{p}_k^{\dagger}\mathbf{q}_k,\\ &\ subject\ to\\ &\qquad\qquad \mathbf{p}_k \in \mathbf{U}^{(1)}(t), \mathbf{q}_k \in \mathbf{U}^{(1)}(t+1),\\ &\qquad\qquad \mathbf{p}_k^{\dagger}\mathbf{u}_1^{(1)}(t) = \cdots = \mathbf{p}_k^{\dagger}\mathbf{u}_{r-1}^{(1)}(t) = 0, \|\mathbf{p}_k\| = 1,\\ &\qquad\qquad \mathbf{q}_k^{\dagger}\mathbf{u}_1^{(1)}(t+1) = \cdots = \mathbf{q}_k^{\dagger}\mathbf{u}_{r-1}^{(1)}(t+1) = 0, \|\mathbf{q}_k\| = 1. \end{aligned} \tag{1}$$

The kth principal angle is defined as $cos(\theta_k) = p_k^{\dagger}q_k; k = 1, 2, \ldots, r$[85].

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] O. Barnich and M. Van Droogenbroeck, “Vibe: a powerful random technique to estimate the background in video sequences,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 945–948.

[2] M. Steriade, P. Gloor, R.R. Llinas, F.H. Lopes Da Silva, and M. Mesulam, “Basic mechanisms of cerebral rhythmic activities,” *Electroencephalography and clinical neurophysiology*, vol. 76, no. 6, pp. 481–508, 1990.

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

[4] F. Varela, J. 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.

[5] F. Babiloni, F. Cincotti, C. Babiloni, F. Carducci, D. Mattia, L. Astolfi, A. Basilisco, P.M. Rossini, L. Ding, Y. Ni, J. Chengg, K. Christineg, J. Sweeneyg, and B. He, “Estimation of the cortical functional connectivity with the multimodal integration of high-resolution eeg and fmri data by directed transfer function,” *Neuroimage*, vol. 24, no. 1, pp. 118–131, 2005.

[6] A. Di Martino, A. Scheres, D.S. Margulies, A. Kelly, L.Q. Uddin, Z. Shehzad, B. Biswal, J.R. Walters, F. X. Castellanos, and M.P. Milham, “Functional connectivity of human striatum: a resting state fmri study,” *Cerebral cortex*, vol. 18, no. 12, pp. 2735–2747, 2008.

[7] M.P. Van Den Heuvel and H.E. H. Pol, “Exploring the brain network: a review on resting-state fmri functional connectivity,” *European Neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, 2010.

[8] Ed Bullmore and Olaf Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

[9] M. P. van den Heuvel, C.J. Stam, M. Boersma, and H.E.H. Pol, “Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain,” *Neuroimage*, vol. 43, no. 3, pp. 528–539, 2008.

[10] C.J. Stam and J.C. Reijneveld, “Graph theoretical analysis of complex networks in the brain,” *Nonlinear biomedical physics*, vol. 1, no. 1, pp. 3, 2007.

[11] J. Lachaux, E. Rodriguez, J. Martinerie, and F.J. Varela, “Measuring phase synchrony in brain signals,” *Human brain mapping*, vol. 8, no. 4, pp. 194–208, 1999.

[12] E. Pereda, R. Q. Quiroga, and J. Bhattacharya, “Nonlinear multivariate analysis of neurophysiological signals,” *Progress in neurobiology*, vol. 77, no. 1, pp. 1–37, 2005.

[13] M.J. Kaminski and K.J. Blinowska, “A new method of the description of the information flow in the brain structures,” *Biological cybernetics*, vol. 65, no. 3, pp. 203–210, 1991.

[14] L.A. Baccalá and K. Sameshima, “Partial directed coherence: a new concept in neural structure determination,” *Biological cybernetics*, vol. 84, no. 6, pp. 463–474, 2001.

[15] C.W.J. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[16] 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.

[17] A.R. Damasio, “Synchronous activation in multiple cortical regions: a mechanism for recall,” 1990.

[18] H. Hinrichs, HJ Heinze, and MA Schoenfeld, “Causal visual interactions as revealed by an information theoretic measure and fMRI,” *Neuroimage*, vol. 31, no. 3, pp. 1051–1060, 2006.

[19] CJ Stam, “Nonlinear dynamical analysis of EEG and MEG: review of an emerging field,” *Clinical Neurophysiology*, vol. 116, no. 10, pp. 2266–2301, 2005.

[20] P. Tass, M.G. 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, pp. 3291, 1998.

[21] J. 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.

[22] D. Li and R. Jung, “Quantifying coevolution of nonstationary biomedical signals using time-varying phase spectra,” *Annals of biomedical engineering*, vol. 28, no. 9, pp. 1101–1115, 2000.

[23] 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.

[24] D.J. Watts and S.H. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[25] 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, pp. e1000100, 2008.

[26] L. Douw, MM Schoonheim, D. Landi, ML Van Der Meer, JJG Geurts, JC Reijneveld, M. Klein, and CJ Stam, “Cognition is related to resting-state small-world network topology: an magnetoencephalographic study,” *Neuroscience*, vol. 175, pp. 169–177, 2011.

[27] CJ Stam, T. Montez, BF Jones, S. Rombouts, Y. Van Der Made, YAL Pijnenburg, and P. Scheltens, “Disturbed fluctuations of resting state EEG synchronization in Alzheimer’s disease,” *Clinical Neurophysiology*, vol. 116, no. 3, pp. 708–715, 2005.

[28] S. Achard and E. Bullmore, “Efficiency and cost of economical brain functional networks,” *PLoS computational biology*, vol. 3, no. 2, pp. e17, 2007.

[29] D.S. Bassett, E. Bullmore, B.A. Verchinski, V.S. Mattay, D.R. Weinberger, and A. Meyer-Lindenberg, “Hierarchical organization of human cortical networks in health and schizophrenia,” *The Journal of Neuroscience*, vol. 28, no. 37, pp. 9239, 2008.

[30] F.D.V. Fallani, F. Pichiorri, G. Morone, M. Molinari, F. Babiloni, F. Cincotti, and D. Mattia, “Multiscale topological properties of functional brain networks during motor imagery after stroke,” *Neuroimage*, vol. 83, pp. 438–449, 2013.

[31] O. Sporns, C.J. Honey, and R. Kötter, “Identification and classification of hubs in brain networks,” *PLoS One*, vol. 2, no. 10, pp. e1049, 2007.

[32] 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.

[33] 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.

[34] 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.

[35] P. Boveroux, A. Vanhaudenhuyse, M. 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, pp. 1038–53, 2010.

[36] J. Schrouff, V. Perlbarg, M. Boly, G. Marrelec, P. Boveroux, A. Vanhaudenhuyse, M. 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.

[37] 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.

[38] 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.

[39] 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, pp. 050905, 2008.

[40] N. Santoro, W. Quattrociocchi, P. Flocchini, A. Casteigts, and F. Amblard, “Time-varying graphs and social network analysis: Temporal indicators and metrics,” *Arxiv preprint arXiv:1102.0629*, 2011.

[41] P.J. Mucha, T. Richardson, K. Macon, M.A. Porter, and J.P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[42] P.W. Kalivas and N.D. Volkow, “The neural basis of addiction: a pathology of motivation and choice,” *American Journal of Psychiatry*, 2014.

[43] K.C. Berridge and T. E. Robinson, “What is the role of dopamine in reward: hedonic impact, reward learning, or incentive salience?,” *Brain Research Reviews*, vol. 28, no. 3, pp. 309–369, 1998.

[44] W.W. Seeley, V. Menon, A.F. Schatzberg, J. Keller, G. H. Glover, H. Kenna, A.L. Reiss, and M.D. Greicius, “Dissociable intrinsic connectivity networks for salience processing and executive control,” *The Journal of neuroscience*, vol. 27, no. 9, pp. 2349–2356, 2007.

[45] E.K. Miller and J.D. Cohen, “An integrative theory of prefrontal cortex function,” *Annual review of neuroscience*, vol. 24, no. 1, pp. 167–202, 2001.

[46] E.K Miller, “The prefontral cortex and cognitive control,” *Nature reviews neuroscience*, vol. 1, no. 1, pp. 59–65, 2000.

[47] R.Z. Goldstein and N.D. Volkow, “Drug addiction and its underlying neurobiological basis: neuroimaging evidence for the involvement of the frontal cortex,” *American Journal of Psychiatry*, vol. 159, no. 10, pp. 1642–1652, 2002.

[48] A. Bechara, “Decision making, impulse control and loss of willpower to resist drugs: a neurocognitive perspective,” *Nature neuroscience*, vol. 8, no. 11, pp. 1458–1463, 2005.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] N. Leonardi, J. Richiardi, M. Gschwind, S. Simioni, J. 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.

[54] 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.

[55] 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.

[56] M.J. Lowe, E.B. Beall, K.E. Sakaie, K.A. Koenig, L. Stone, R.A. Marrie, and M.D. Phillips, "Resting state sensorimotor functional connectivity in multiple sclerosis inversely correlates with transcallosal motor pathway transverse diffusivity," .

[57] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton, "Dynamic reconfiguration of human brain networks during learning," *Proceedings of the National Academy of Sciences*, vol. 108, no. 18, pp. 7641–7646, 2011.

[58] 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.

[59] L. Cohen, *Time-frequency analysis*, vol. 778, Prentice hall, 1995.

[60] 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.

[61] 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.

[62] J. Kayser and C.E. Tenke, "Principal components analysis of laplacian waveforms as a generic method for identifying ERP generator patterns: I. evaluation with auditory oddball tasks," *Clinical Neurophysiology*, vol. 117, no. 2, pp. 348–368, 2006.

[63] 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.

[64] 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.

[65] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2005.

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

[67] 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.

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

[69] H. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.

[70] Age Smilde, Rasmus Bro, and Paul Geladi, *Multi-way analysis: applications in the chemical sciences*, Wiley. com, 2005.

[71] Lieven De Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 642–666, 2006.

[72] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "Computation of the canonical decomposition by means of a simultaneous generalized schur decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 295–327, 2004.

[73] E. Ceulemans and H.A. Kiers, "Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 1, pp. 133–150, 2006.

[74] E.B. Fowlkes, A.E. Freeny, and J.M. Landwehr, "Evaluating logistic models for large contingency tables," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 611–622, 1988.

[75] Rasmus Bro, "Parafac. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[76] Rasmus Bro and Henk AL 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.

[77] Morten Mørup and Lars Kai Hansen, "Automatic relevance determination for multiway models," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 352–363, 2009.

[78] 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.

[79] M. Amft, D. Bzdok, A. R. Laird, P.T. Fox, L. Schilbach, and S.B. Eickhoff, "Definition and characterization of an extended social-affective default network," *Brain Structure and Function*, pp. 1–19, 2014.

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

[81] D.M. Zoltowski and S. Aviyente, "Low-rank tensor decomposition based dynamic network tracking," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*. IEEE, 2014, pp. 468–472.

[82] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, vol. 14, pp. 1137–1145.

[83] L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Army Science Conference*, 2010, pp. 77–79.

[84] T. Idé and H. Kashima, "Eigenspace-based anomaly detection in computer systems," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 440–449.

[85] K. Ye and L. Lim, "Distance between subspaces of different dimensions," *arXiv preprint arXiv:1407.0900*, 2014.

[86] A. G. Mahyari and S. Aviyente, "Two-dimensional svd for event detection in dynamic functional brain networks," in *Proceedings of IEEE Global Conference on Signal and Information Processing*, 2013.

[87] M. G. Thomason and J. Gregor, "Higher order singular value decomposition of tensors for fusion of registered images," *Journal of Electronic Imaging*, vol. 20, no. 1, pp. 013023–013023, 2011.

[88] Q. Zhang, Y. Wang, M. D. Levine, X. Yuan, and L. Wang, "Multisensor video fusion based on higher order singular value decomposition," *Journal of Personal and Ubiquitous Computing*, 2014.

[89] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[90] F.J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[91] 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.

[92] D.L. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 1094–1121, 2012.

[93] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-mat: A recursive model for graph mining," in *In SDM*, 2004.

[94] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[95] 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.

[96] M. Bolanos, 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.

[97] 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.

[98] S. Pazos, M. Hurtado, and C. Muravchik, "On sparse methods for array signal processing in the presence of interference," *Antennas and Wireless Propagation Letters, IEEE*, vol. 14, pp. 1165–1168, 2015.

[99] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[100] J. A. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[101] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal processing*, vol. 91, no. 7, pp. 1505–1526, 2011.

[102] Z. Luo, M. Gaspar, J. Liu, and A. Swami, "Distributed signal processing in sensor networks," *Signal Processing Magazine, IEEE*, vol. 23, no. 4, pp. 14–15, 2006.

[103] C. Phillips, J. Mattout, M. D. Rugg, P. Maquet, and K. J. Friston, "An empirical bayesian solution to the source reconstruction problem in eeg," *NeuroImage*, vol. 24, no. 4, pp. 997–1011, 2005.

[104] I. F. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm," *Electroencephalography and clinical Neurophysiology*, vol. 95, no. 4, pp. 231–251, 1995.

[105] E. Ollila, "Nonparametric simultaneous sparse recovery: an application to source localization," *arXiv preprint arXiv:1502.02441*, 2015.

[106] M. Fornasier and H. Rauhut, "Recovery algorithms for vector-valued data with joint sparsity constraints," *SIAM Journal on Numerical Analysis*, vol. 46, no. 2, pp. 577–613, 2008.

[107] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy algorithms for joint sparse recovery," *Signal Processing, IEEE Transactions on*, vol. 62, no. 7, pp. 1694–1704, 2014.

[108] D. Baron, M. B. Wakin, M.F. Duarte, S. Sarvotham, and R.G. Baraniuk, "Distributed compressed sensing," 2005.

[109] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R.G. Baraniuk, "Distributed compressive sensing," *arXiv preprint arXiv:0901.3403*, 2009.

[110] L. Kang and C. Lu, "Distributed compressive video sensing," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 1169–1172.

[111] T.T. Do, Y. Chen, D.T. Nguyen, N. Nguyen, L. Gan, and T.D. Tran, "Distributed compressed video sensing," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1393–1396.

[112] Y. Baig, E. Lai, and A. Punchihewa, "Distributed video coding based on compressed sensing," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 325–330.

[113] S. Patterson, Y.C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *Signal Processing, IEEE Transactions on*, vol. 62, no. 19, pp. 4931–4946, 2014.

[114] K. Lee, Y. Bresler, and M. Junge, "Subspace methods for joint sparse recovery," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3613–3641, 2012.

[115] M. Mishali and Y.C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4692–4702, 2008.

[116] X. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, 2012.

[117] M. Hyder and K. Mahata, "A robust algorithm for joint-sparse recovery," *IEEE Signal Processing Letters*, vol. 16, no. 12, pp. 1091–1094, 2009.

[118] K. Marcotte, V. Perlbarg, G. Marrelec, H. Benali, and A.I. Ansaldo, "Default-mode network functional connectivity in aphasia: therapy-induced neuroplasticity," *Brain and language*, vol. 124, no. 1, pp. 45–55, 2013.

[119] X. Zhang, T. Huang, Y. Tian, and W. Gao, "Background-modeling-based adaptive prediction for surveillance video coding," *Image Processing, IEEE Transactions on*, vol. 23, no. 2, pp. 769–784, 2014.

[120] A. Manzanera and J.C. Richefeu, "A new motion detection algorithm based on σ–δ background estimation," *Pattern Recognition Letters*, vol. 28, no. 3, pp. 320–328, 2007.

[121] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[122] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13, 2006.

[123] M. F. Duarte, S. Sarvotham, D. Baron, M.B. Wakin, and R.G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Asilomar Conf. Signals, Sys., Comput*, 2005, pp. 1537–1541.

[124] D. Baron, M.F. Duarte, S. Sarvotham, M. B. Wakin, and R.G. Baraniuk, "An information-theoretic approach to distributed compressed sensing," in *Proc. 45rd Conference on Communication, Control, and Computing*, 2005.

[125] D. Baron, M. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "Distributed compressive sensing," *Tech. Rep. TREE0612*, 2006, http://dsp.rice.edu/cs/ .

[126] E.J. Candes, J.K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[127] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *Information Theory, IEEE Transactions on*, vol. 54, no. 5, pp. 2210–2219, 2008.

[128] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 310–316, 2010.

[129] M. Harville, G. Gordon, and J. Woodfill, "Adaptive video background modeling using color and depth," in *Image Processing, 2001. Proceedings. 2001 International Conference on.* IEEE, 2001, vol. 3, pp. 90–93.

[130] D. Lee, "Effective gaussian mixture learning for video background subtraction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 827–832, 2005.

[131] "The usc-sipi image database: Image sequence," http://sipi.usc.edu/database/ .

[132] J.C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, pp. 857–871, 1971.

[133] M. Aharon, M.and Elad and A. Bruckstein, "Ksvd: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

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

[135] W. Chen, I.J Wassell, and M.RD Rodrigues, "Dictionary design for distributed compressive sensing," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 95–99, 2015.

[136] O.I. Khan, F. Farooq, F. Akram, M. Choi, S.M. Han, and T. Kim, "Robust extraction of p300 using constrained ica for bci applications," *Medical & biological engineering & computing*, vol. 50, no. 3, pp. 231–241, 2012.

[137] G. Schalk, D.J McFarland, T. Hinterberger, N. Birbaumer, and J.R Wolpaw, "Bci2000: a general-purpose brain-computer interface (bci) system," *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[138] M. Fatourechi, S.G Mason, G. E Birch, and R.K Ward, "A wavelet-based approach for the extraction of event related potentials from eeg," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*. IEEE, 2004, vol. 2, pp. ii–737.

[139] H. Serby, E. Yom-Tov, and G.F Inbar, "An improved p300-based brain-computer interface," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 13, no. 1, pp. 89–98, 2005.

[140] A. Ivannikov, T. Karkkainen, T. Ristaniemi, and H. Lyytinen, "Extraction of erp from eeg data," in *2007 9th International Symposium on Signal Processing and Its Applications*, 2007.

[141] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[142] B.A Eriksen and C.W Eriksen, "Effects of noise letters upon the identification of a target letter in a nonsearch task," *Perception & psychophysics*, vol. 16, no. 1, pp. 143–149, 1974.

[143] A.G. Mahyari, D. Zoltowski, E. Bernat, and S. Aviyente, "A tensor decomposition based approach for detecting dynamic network states from EEG," 2016.

[144] D.S. Ruchkin, S. Sutton, and D. Mahaffey, "Functional differences between members of the p300 complex: P3e and p3b," *Psychophysiology*, vol. 24, no. 1, pp. 87–103, 1987.