Novel Statistical Models for Complex Data Structures

by

Shuai Huang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2012 by the
Graduate Supervisory Committee:

Jing Li, Chair
Ronald Askin
George Runger
Jieping Ye

ARIZONA STATE UNIVERSITY

August 2012

ABSTRACT

Rapid advance in sensor and information technology has resulted in both spatially and temporally data-rich environment, which creates a pressing need for us to develop novel statistical methods and the associated computational tools to extract intelligent knowledge and informative patterns from these massive datasets. The statistical challenges for addressing these massive datasets lay in their complex structures, such as high-dimensionality, hierarchy, multi-modality, heterogeneity and data uncertainty. Besides the statistical challenges, the associated computational approaches are also considered essential in achieving efficiency, effectiveness, as well as the numerical stability in practice. On the other hand, some recent developments in statistics and machine learning, such as sparse learning, transfer learning, and some traditional methodologies which still hold potential, such as multi-level models, all shed lights on addressing these complex datasets in a statistically powerful and computationally efficient way. In this dissertation, we identify four kinds of general complex datasets, including "high-dimensional datasets", "hierarchically-structured datasets", "multi-modality datasets" and "data uncertainties", which are ubiquitous in many domains, such as biology, medicine, neuroscience, health care delivery, manufacturing, etc. We depict the development of novel statistical models to analyze complex datasets which fall under these four categories, and we show how these models can be applied to some real-world applications, such as Alzheimer's disease research, nursing care process, and manufacturing.

DEDICATION

To my family

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my advisor, Professor. Jing Li, for her generous support, endless patience and encouragement, and insightful guidance throughout my research. She brought me into this scholar community, offered me great research opportunities and resources. Without her, this dissertation would not have become possible.

I also would like to thank my other dissertation committee members: Professor Jieping Ye provided me precious help and insights into machine learning research, who has been acting as my second "advisor" in computer science; and Professor Ronald Askin and Professor George Runger, who provided valuable comments and suggestions for both my dissertation and my future career. My gratitude also goes to Dr. Kewei Chen, who provided me wonderful opportunities to work with him on Alzheimer's disease research using Neuroimaging data, and learn how to collaborate with health professionals and neuroimaging experts. I also would like to thank Professor Rong Pan, who helped me a lot on reliability research and always gave me precious career advice. I also would like to extend my gratitude to Professor Teresa Wu, Professor Esma Gel, Professor Muhong Zhang, for their kind support and encouragement during these 5 years.

I would like to thank all the people who have helped me at ASU. Special thanks to Houtao Deng, Liangjie Xue, for their generous help given to me many times.

Last but not least, I would like to thank my family, my parents and especially my wife, Qi, for her love and understanding. I also would like to thank my dog, Bobo, for his loyal company in this desert.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Motivation

Recent rapid developments of sensor and information technologies have resulted in a spatially and temporally data-rich environment. With massive datasets readily available, there is a pressing need to extract intelligent knowledge and informative patterns in order to accomplish various decision-making goals. The statistical challenges for analyzing these massive datasets lay in their complex structures, such as high-dimensionality, hierarchy, multi-modality, heterogeneity, and uncertainty. In addition to the statistical challenges, there are computational challenges in achieving efficiency, effectiveness, as well as numerical stability in practice. In what follows, some real-world examples are shown to illustrate the aforementioned complex data structures and the challenges in modeling and analyzing the data.

1.1.1 High-dimensional datasets

High-dimensional datasets are ubiquitous in many applications, such as the output of the second-generation sequencing machines in genomics, and images in fields ranging from physics to neuroscience to medicine. A consequence is a big challenge for statisticians to extract informative patterns and intelligent knowledge from them, e.g., to identify a few genes or proteins out of a huge list, which may be active in a particular metabolic or disease process.

1.1.2 Hierarchically-structured datasets

Hierarchically-structured datasets are reflections of the hierarchical nature of many complex systems and organizations. For example, to investigate the nursing care process in a hospital setting, information can be collected on multiple levels, including the individual nurse level, unit level, and hospital level. How to link these multi-level

information sources and fuse them for understanding and improve the nursing care process is a challenge.

1.1.3 Data uncertainty

The "data uncertainty" that is focused here does not refer to the inherent randomness of a process or system or sampling uncertainty of statistical models. Rather, the data uncertainty refers to the uncertainty introduced into the data collection mechanisms, for reasons such as poor calibration on sensors, sensor malfunction, and human errors. For example, distributed sensor networks (DSN) have been widely equipped in many manufacturing processes, automatically collecting data on various process variables and quality attributes. Due to the large scale and mixed types of these sensors, as well as the real-time uninterruptable data collection mechanism, effective calibration and timely maintenance is commonly unavailable. Thus, sensor errors and noises are inevitable, which result in corrupted measurements on the process. How to effectively utilize these corrupted sensor data for process monitoring, fault detection, and root cause diagnosis is not well addressed in the statistical process control literature.

1.1.4 Multi-modality

Technological platforms have been advanced to such a stage where multiple aspects of the same process or system can be measured collectively. For example, both MRI and PET can be used to measure the brain structure and activity, respectively. These "multi-modality" datasets need to be analyzed in an integrated way, e.g., an integration of MRI and PET imaging data may achieve higher statistical power in detecting subtle disease-related patterns in early stages of many progressive diseases, which is an essential task for achieving effective evidence-based medical diagnosis and prevention.

1.2 State of the art

This dissertation focuses on following complex data structures, including high-dimensionality, hierarchy, multi-modality, and uncertainty. Correspondingly, I will review several existing research areas that handle such data structures.

1.2.1 Sparse learning

The essential idea of sparse learning is to encourage parsimony in statistical modeling, which can actually be traced back to some ancient scientific principles, such as "Occam's razor". The parsimony of statistical models implies a preference over the models with a smaller number of free parameters, among all the models that can sufficiently capture the complexity and uncertainty of the datasets. Some direct benefits of such a parsimonious statistical model include enhanced interpretability and stability of the models. A classic example is the ridge regression, which employs a L2-norm regularized least square formulation to encourage sparsity of the regression parameters, as shown below:

$$\widehat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\},$$

where $\mathbf{Y}$ is a $n \times 1$ vector, which records the responses of n samples; $\mathbf{X}$ is a $n \times p$ design matrix, with p being the number of predictors; $\boldsymbol{\beta}$ is the regression parameter vector with length p; the square of the L2-norm of any vector $\boldsymbol{a}$, $\|\boldsymbol{a}\|_2^2$, is a sum of the squares of the elements in $\boldsymbol{a}$. λ is the penalty parameter, which controls the shrinkage effect, i.e., larger λ, more penalty is imposed on the magnitudes of $\widehat{\boldsymbol{\beta}}$.

Ridge regression is commonly known as being capable to effectively shrink many unreliable or redundant regression parameters in $\widehat{\boldsymbol{\beta}}$ toward zero, thus achieving more stability than ordinary least-square regression models. However, since the objective function of ridge regression is a smooth function without any singularity in the parameter space for any $\boldsymbol{\beta}_i = 0$, there is actually no parameter in $\widehat{\boldsymbol{\beta}}$ that will be exactly zero. This implies that ridge regression is not a truly sparse model, since all the variables are always

kept in the regression model, no matter how large the penalty parameter λ is used. On the other hand, in many applications, such as cancer research, there is a hypothesis that only a few genes or proteins are correlated with the disease processes, although we can obtain the gene expressions of thousands of genes simultaneously. Therefore, ridge regression is less effective in these applications, and statistical models that can achieve exact sparsity in $\widehat{\boldsymbol{\beta}}$ are required.

LASSO is such an approach which can effectively achieve exact sparsity in $\widehat{\boldsymbol{\beta}}$. The formulation of LASSO is similar to ridge, except the L1-norm is used to replace the L2-norm:

$$\widehat{\boldsymbol{\beta}} = \arg\min\{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\},$$

where $\|\boldsymbol{\beta}\|_1$ is a sum of the absolute values of the elements in $\boldsymbol{\beta}$. As the L1-norm introduces singularity into the objective function for any $\boldsymbol{\beta}_i = 0$, LASSO can set many unreliable or redundant parameters to be exactly zero, thus achieving model selection. The extra cost of LASSO is an efficient computational algorithm, since LASSO has no closed-form solution. By drawing on recent developments on optimization theories especially on convex optimization, a number of efficient algorithms have been developed to solve LASSO. Following this line, a number of novel norms have been developed to address various kinds of complex datasets, with the goals of making use of some "structural information" about the structure of these complex datasets, such as group LASSO and fused LASSO. The idea of LASSO has also been extended to other statistical models, such as sparse PCA, graphical models and mixed effect models. It has been found in statistics and machine learning communities that sparse learning is an effective and promising technique to address high-dimensional datasets. In this dissertation, several new high-dimensional statistical models anchored with sparse learning will be presented in Chapters 2 and 3.

1.2.2 Transfer learning

Transfer learning is a generalization of the traditional statistical learning paradigm, which is capable to address multiple related datasets jointly. The "multiple related datasets" refer to specific kinds of hierarchically-structured datasets. For instance, considering the brain activity measurements on different stages of a brain disease, each stage has a dataset and a corresponding brain connectivity network model which produces this dataset. Not like the traditional statistical estimation paradigm, which estimates the brain connectivity network for each stage in isolation, transfer learning aims to infer these multiple brain connectivity networks from those datasets jointly, enabling the knowledge learning from one dataset to be transferred to another, by exploring the similarity between the brain connectivity networks. This transferability is particularly advantageous when the sample size is small for all the stages, since it enables the use of the dataset for other stages to help the learning task of each individual stage. In this dissertation, a specific transfer learning methodology is developed in chapter 4 which can learn multiple Gaussian graphical models jointly.

1.2.3 Multi-level models

Multi-level models, also known as hierarchical linear models, mixed models, and random coefficient models, are statistical models which can decompose the variation of an outcome of interest into different levels. This can be illustrated by the "radon example", which is a risk factor for causing lung cancer in high concentration. The distribution of radon levels in U.S. homes varies greatly. In order to identify the regions with high risks, the environmental protection agency collected radon measurements in a random sample of over 80,000 homes from 3000 countries. As the data is structured hierarchically as homes within countries, a multi-level model for this radon data analysis is:

$$y_{ij} \sim N\left(\alpha_j + \beta x_{ij}, \sigma_y^2\right), \text{ for } i = 1,2,\dots,n_j; j = 1,2,\dots,J.$$

$$\alpha_j \sim N\left(r_0 + r_1 u_j, \sigma_\alpha^2\right), \text{ for } j = 1,2, \dots, J.$$

where y_{ij} is the logarithm of the radon measurement in house i within country j, x_{ij} is an indicator for whether the measurement was taken in a basement, and u_j is the log uranium level in country j. In this way, σ_y^2 captures the "within-country" variations, and σ_α^2 captures variations between countries.

As multi-level models are advantageous to decompose the variation of the outcome of interest into different levels and identify significant correlations within the same level and across different levels, they have been widely adopted in many disciplines, such as social science, education and biology, and has recently been borrowed by the machine learning community to build more powerful machine learning algorithms. In this dissertation, a novel multi-level model is developed to address a complex dataset collected from a health care delivery problem, i.e., the nursing care coordination process.

1.3 Research objectives

The objectives of this research are:

1) Develop novel statistical models and their associated computational algorithms for analyzing complex datasets, such as high-dimensional datasets, hierarchically-structured datasets, multi-modality datasets, and datasets with data uncertainty, by drawing on recent theoretical developments in statistics and machine learning, such as sparse learning, transfer learning, and multi-level models.
2) Apply these novel statistical models for knowledge discovery and decision making from real-world datasets, including biomedical, healthcare and manufacturing applications.

1.4 Organization of the dissertation

This dissertation is presented in a multiple manuscript format. Each of the chapters, 2 to 7, is written as an individual research paper, including an abstract, a main body, and references. The relationships among these chapters are depicted in Figure 1-1.

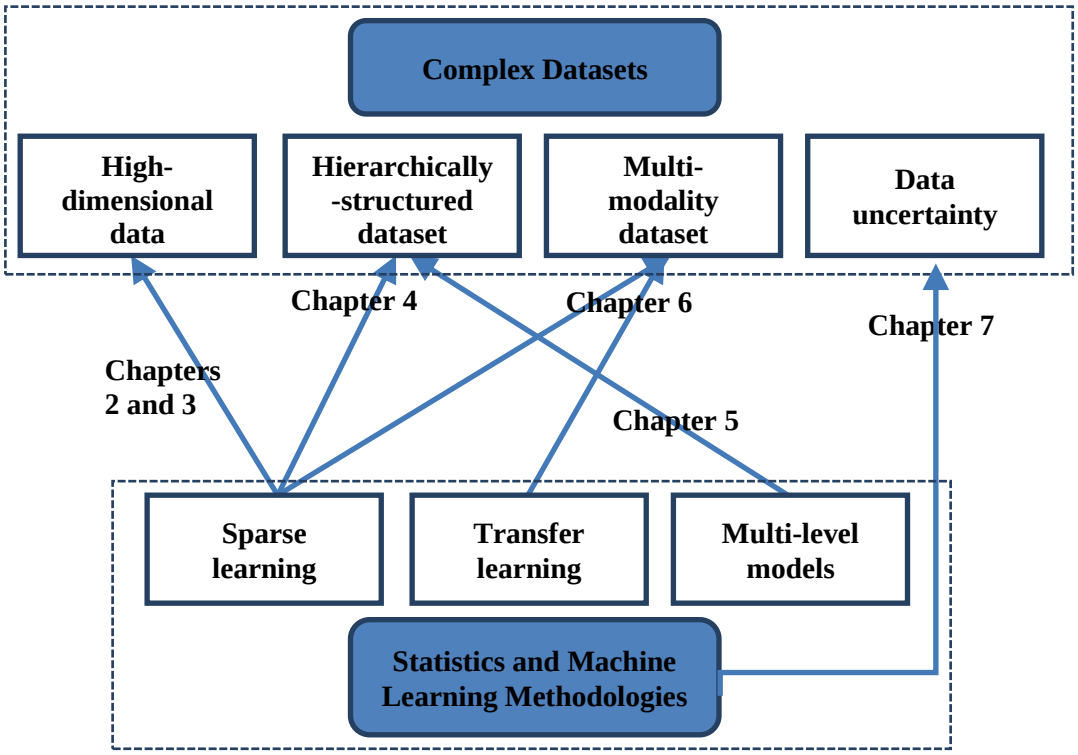


Fig. 1-1: Relationships among the chapters in this dissertation

Chapter 2 [Huang, et al., 2009, 2010] presents a novel high-dimensional statistical model, called sparse inverse covariance estimation (SICE), which is useful for analyzing the interactions between a large number of variables from measurement data, i.e., inferring the brain connectivity networks of tens or hundreds of brain regions, or inferring gene regulatory networks of hundreds or thousands of genes. A monotone property of SICE is also proved which provides a way for estimating the strength of interactions within the networks. The SICE, along with the strength estimation, has been applied to analyze a

PET scan data, downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) website, based on which novel knowledge and new insights into the Alzheimer's disease has been discovered.

Chapter 3 [Huang, et al., 2011] presents s new high-dimensional statistical method, called sparse Bayesian network (SBN), which is capable to learn the structure of a large Bayesian network from a high-dimensional dataset. SBN employs a novel formulation involving one L1-norm penalty term to impose sparsity and another penalty term to ensure that the learned BN is a directed acyclic graph – a required property of BN. Both theoretical analysis and extensive experiments are presented, which demonstrated that SBN is superior over the competing algorithms on a wide spectrum of benchmark BN structures under various sample sizes. An application of SBN on the Alzheimer's disease research is also investigated.

Chapter 4 [Huang, et al., in press] depicts the development of a transfer learning approach for estimating multiple Gaussian graphical models (GGM) jointly, from multiple related datasets. Anchored by the Bayesian hierarchical methodology, the relatedness between multiple GGMs are represented by a common Wishart distribution, and an EM algorithm is derived to estimate these GGMs with the presence of unknown hyper parameters of this Wishart distribution. Sparse learning is also employed to ensure its applicability on high-dimensional datasets. The developed transfer learning method is applied to a real-world dataset, a fMRI imaging datasets for two groups of people, which shows the transfer learning method is superior over the traditional statistical learning paradigm.

Chapter 5 [Huang, et al., in press] depicts the development of a multi-level statistical model for analyzing hierarchically-structured datasets. This model development is motivated by a health care delivery problem, i.e., the modeling and analysis of nursing

care process, where information is collected on multiple levels, e.g., on individual nurse level, unit level, hospital level, etc. Not like traditional hierarchically-structured data, which can be well addressed by existing multi-level models, whose response variables are measured on individual level (the lowest level of the dataset), here, limited by the inherent difficulty to measure individuals' contributions on an unit's overall quality, the response variables are measured only on unit-level. With the goals of fusing all the levels' information for modeling and improving the nursing care process, a multi-level latent response linear regression model is developed and applied on a real dataset collected from 38 units within 4 hospitals, which helps discovery of some informative relationships between some nursing activities, unit infrastructure and patient falls.

Chapter 6 presents a statistical method, called, sparse composite linear discriminant analysis (SCLDA), to identify a few variables (out of a huge list) which are predictive for classification, by fusing multi-modality data. SCLDA employs a novel parameterization that decomposes each LDA parameter into a product of a common parameter shared by all the modalities and a parameter specific to each modality, which enables joint analysis of all the modalities and borrowing strength from one another. By employing some optimization reformulation and the DC algorithm, we derived an efficient and easily interpretable algorithm to estimate the free parameters of SCLDA. An application of SCLDA on a dataset with two modalities, the Magnetic Resonance Imaging (MRI) and Positron Emission Tomography of 116 subjects, is also presented.

Chapter 7 depicts the development of a regression-based process monitoring method with consideration of measurement errors. As most existing process monitoring methods are based on a common assumption that the measured values of variables are the true values, with limited consideration of various types of measurement errors embedded in the data, those methods are less effective when applied in real-world applications, where

measurement errors are inevitable. On the other hand, research on measurement errors has been conducted from a pure theoretical statistics point of view, without any linking of the modeling and analysis of measurement errors with monitoring and fault detection. Motivated by such a lack of methodology, a method for multivariate process monitoring and fault detection considering four types of major measurement errors, including sensor bias, sensitivity, noise and dependency of the relationship between a variable and its measured value on some other variables, has been developed. This method is applicable to processes where the natural ordering of the variables is known, and processes where the causal relationships among variables are known and can be described by a Bayesian network. This method is demonstrated in two industrial processes.

Chapter 2

LEARNING BRAIN CONNECTIVITY OF ALZHEIMER'S DISEASE FROM NEUROIMAGING DATA

Abstract

Recent advances in neuroimaging techniques provide great potentials for effective diagnosis of Alzheimer's disease (AD), the most common form of dementia. Previous studies have shown that AD is closely related to alternation in the functional brain network, i.e., the functional connectivity among different brain regions. In this paper, we consider the problem of learning functional brain connectivity from neuroimaging, which holds great promise for identifying image-based markers used to distinguish Normal Controls (NC), patients with Mild Cognitive Impairment (MCI), and patients with AD. More specifically, we study sparse inverse covariance estimation (SICE), also known as exploratory Gaussian graphical models, for brain connectivity modeling. We prove a monotone property of the SICE algorithm, which is a very important property for helping identify the difference between AD, MCI, and NC. We apply the proposed algorithm to the neuroimaging PET data of 42 AD, 116 MCI, and 67 NC subjects. The experimental results reveal several interesting connectivity patterns consistent with literature findings.

2.1 Introduction

Alzheimer's disease (AD) is a fatal, neurodegenerative disorder characterized by progressive impairment of memory and other cognitive functions. It is the most common form of dementia and currently affects over five million Americans; this number will grow to as many as 14 million by year 2050. The current knowledge about the cause of AD is very limited; clinical diagnosis is imprecise with definite

diagnosis only possible by autopsy; also, there is currently no cure for AD, while most drugs only alleviate the symptoms.

To tackle these challenging issues, the rapidly advancing neuroimaging techniques provide great potentials. These techniques, such as MRI, PET, and fMRI, produce data (images) of brain structure and function, making it possible to identify the difference between AD and normal brains. Recent studies have demonstrated that neuroimaging data provide more sensitive and consistent measures of AD onset and progression than conventional clinical assessment and neuropsychological tests [1].

Recent studies have found that AD is closely related to alternation in the functional brain network, i.e., the functional connectivity among different brain regions [2]-[3]. Specifically, it has been shown that functional connectivity substantially decreases between the hippocampus and other regions of AD brains [3]-[4]. Also, some studies have found increased connectivity between the regions in the frontal lobe [6]-[7].

Learning functional brain connectivity from neuroimaging data holds great promise for identifying image-based markers used to distinguish between AD, MCI (Mild Cognitive Impairment), and normal aging. Note that MCI is a transition stage from normal aging to AD. Understanding and precise diagnosis of MCI have significant clinical value since it can serve as an early warning sign of AD. Despite all these, existing research in functional brain connectivity modeling suffers from limitations:

A large body of functional connectivity modeling has been based on correlation analysis [2]-[3], [5]. However, correlation only captures pairwise information and fails to provide a complete account for the interaction of many (more than two) brain regions. Other multivariate statistical methods have also been used, such as Principle Component Analysis (PCA) [8], PCA-based Scaled Subprofile Model [9], Independent Component Analysis [10]-[11], and Partial Least Squares [12]-[13],

which group brain regions into latent components. The brain regions within each component are believed to have strong connectivity, while the connectivity between components is weak. One major drawback of these methods is that the latent components may not correspond to any biological entities, causing difficulty in interpretation.

In addition, graphical models have been used to study brain connectivity, such as structural equation models [14]-[15], dynamic causal models [16], and Granger causality. However, most of these approaches are confirmative, rather than exploratory, in the sense that they require a prior model of brain connectivity to begin with. Also, these approaches are usually applied to a small number of pre-selected brain regions (less than 20 in most cases). These limitations make them inadequate for studying AD brain connectivity, because there is little prior knowledge about which regions should be involved and how they are connected. This makes exploratory models highly desirable.

In this paper, we study sparse inverse covariance estimation (SICE), also known as exploratory Gaussian graphical models, for brain connectivity modeling. Inverse covariance matrix has a clear interpretation that the off-diagonal elements correspond to partial correlations, i.e., the correlation between each pair of brain regions given all other regions. This provides a much better model for brain connectivity than simple correlation analysis which models each pair of regions without considering other regions. Also, imposing sparsity on the inverse covariance estimation ensures a reliable brain connectivity to be modeled with limited sample size, which is usually the case in AD studies since clinical samples are difficult to obtain. From a domain perspective, imposing sparsity is also valid because neurological findings have demonstrated that a brain region usually only directly

interacts with a few other brain regions in neurological processes [2]-[3]. Various algorithms for achieving SICE have been developed in recent year [17]-[22]. In addition, SICE has been used in various applications, including evaluating patterns of association among variables [23], exploration of genetic networks [21], senator voting records analysis [17], hyperspectral image classification [24], and speech recognition [25]. However, SICE has been barely used in brain connectivity modeling, especially for AD studies.

In this paper, we prove a monotone property of the proposed SICE algorithm, which is a very important property for helping identify the difference between AD, MCI, and NC. We apply SICE to the neuroimaging PET data of 42 AD, 116 MCI, and 67 NC subjects enrolled in the ANDI (Alzheimer's Disease Neuroimaging Initiative) project. The experimental results reveal several interesting connectivity patterns consistent with literature findings, and also some new patterns that can help the knowledge discovery of AD.

2.2 The SICE algorithm and monotone property

An inverse covariance matrix can be represented graphically. If used to represent brain connectivity, the nodes are activated brain regions; existence of an arc between two nodes means that the two brain regions are closely related in the brain's functional process.

Let $\{X_1, \cdots, X_p\}$ be all the brain regions under study. $\{X_1, \cdots, X_p\}$ follows a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the inverse covariance matrix. Suppose we have n samples (e.g., n subjects with AD) for these brain regions. Note that we will only illustrate here the SICE for AD, whereas the SICE for MCI and NC can be achieved in a similar way.

We can formulate the SICE into an optimization problem, i.e.,

$$\widehat{\boldsymbol{\Theta}} = \text{argmax}_{\boldsymbol{\Theta}\succ\mathbf{0}}\, log\big(det(\boldsymbol{\Theta})\big) - tr(\mathbf{S}\boldsymbol{\Theta}) - \lambda\|vec(\boldsymbol{\Theta})\|_1 \qquad (1)$$

where $\mathbf{S}$ is the sample covariance matrix; $det(\cdot)$, $tr(\cdot)$, and $\|vec(\cdot)\|_1$ denote the determinant, trace, and sum of the absolute values of all elements of a matrix, respectively. The part "$log\big(det(\boldsymbol{\Theta})\big) - tr(\mathbf{S}\boldsymbol{\Theta})$" in (1) is the log-likelihood, whereas the part "$\|vec(\boldsymbol{\Theta})\|_1$" represents the "sparsity" of the inverse covariance matrix $\boldsymbol{\Theta}$. (1) aims to achieve a tradeoff between the likelihood fit of the inverse covariance estimate and the sparsity. The tradeoff is controlled by λ, called the regularization parameter; larger λ will result in more sparse estimate for $\boldsymbol{\Theta}$. The formulation in (1) follows the same line of the L_1-norm regularization [26]-[27], which has been introduced into the least squares formulation to achieve model sparsity and the resulting model is called Lasso [27]. Next, we show that with λ going from small to large, the resulting brain connectivity models have a monotone property. Before introducing the monotone property, the following definitions are needed.

Definition: In the graphical representation of the inverse covariance, if node X_i is connected to X_j by an arc, then X_i is called a "neighbor" of X_j. If X_i is connected to X_k though some chain of arcs, then X_i is called a "connectivity component" of X_k.

Intuitively, being neighbors means that two nodes (i.e., brain regions) are directly connected, whereas being connectivity components means that two brain regions are indirectly connected, i.e., the connection is mediated through other regions. In other words, not being connectivity components (i.e., two nodes completely separated in the graph) means that the two corresponding brain regions are completely independent of each other. Connectivity components have the following monotone property:

Monotone property of the SICE: Let $\mathbf{C}_k(\lambda_1)$ and $\mathbf{C}_k(\lambda_2)$ be the sets of all the connectivity components of X_k with $\lambda = \lambda_1$ and $\lambda = \lambda_2$, respectively. If $\lambda_1 < \lambda_2$, then $\mathbf{C}_k(\lambda_1) \supseteq \mathbf{C}_k(\lambda_2)$.

Proof of the monotone property can be found in the appendix. This monotone property can be used to identify how strongly connected each node (brain region) X_k to its connectivity components. For example, assuming that $\mathbf{C}_k(\lambda_1) = \{X_i, X_j\}$ and $\mathbf{C}_k(\lambda_2) = \{X_i\}$, this means that X_i is more strongly connected to X_k than X_j. Thus, by changing λ from small to large, we can obtain an order for the strength of connection between pairs of brain regions. As will be shown in Section 2-3, this order is different between AD, MCI, and NC.

2.3 Application in brain connectivity modeling of AD

2.3.1 Data acquisition and preprocessing

We apply SICE on FDG-PET images for 49 AD, 116 MCI, and 67 NC subjects downloaded from the ADNI website. We apply Automated Anatomical Labeling (AAL) [28] to extract data from each of the 116 anatomical volumes of interest (AVOI), and derived average of each AVOI for every subject. The AVOI represent different regions of the whole brain.

2.3.2 Brain connectivity modeling by SICE

42 AVOI are selected for brain connectivity modeling, as they are considered to be potentially related to AD. These regions distribute in the frontal, parietal, Occipital, and temporal lobes. Pease see Table 2-1 for names of the AVOI and which lobe each of them belongs to. The number before each AVOI is used to index the node in the connectivity models.

Using the algorithm proposed in Section 2-2, one connectivity model can be learned for AD, one for MCI, and one for NC, for a given λ. With different λ's, the resulting

connectivity models hold a monotone property, which can help obtain an order for the strength of connection between brain regions. To show the order clearly, we develop a tree-like plot in Fig. 2-1, which is for the AD group. To generate this plot, we start λ at a very small value (i.e., the right-most of the horizontal axis), which results in a fully-connected connectivity model. A fully-connected connectivity model is one that contains no region disconnected with the rest of the brain. Then, we decrease λ by small steps and record the order of the regions disconnected with the rest of the brain regions.

Table 2-1: Names of the AVOI for connectivity modeling ("L" means that the brain region is located at the left hemisphere; "R" means right hemisphere.)

Frontal lobe		Parietal lobe		Occipital lobe		Temporal lobe	
1	Frontal_Sup_L	13	Parietal_Sup_L	21	Occipital_Sup_L	27	Temporal_Sup_L
2	Frontal_Sup_R	14	Parietal_Sup_R	22	Occipital_Sup_R	28	Temporal_Sup_R
3	Frontal_Mid_L	15	Parietal_Inf_L	23	Occipital_Mid_L	29	Temporal_Pole_Sup_L
4	Frontal_Mid_R	16	Parietal_Inf_R	24	Occipital_Mid_R	30	Temporal_Pole_Sup_R
5	Frontal_Sup_Medial_L	17	Precuneus_L	25	Occipital_Inf_L	31	Temporal_Mid_L
6	Frontal_Sup_Medial_R	18	Precuneus_R	26	Occipital_Inf_R	32	Temporal_Mid_R
7	Frontal_Mid_Orb_L	19	Cingulum_Post_L			33	Temporal_Pole_Mid_L
8	Frontal_Mid_Orb_R	20	Cingulum_Post_R			34	Temporal_Pole_Mid_R
9	Rectus_L					35	Temporal_Inf_L 8301
10	Rectus_R					36	Temporal_Inf_R 8302
11	Cingulum_Ant_L					37	Fusiform_L
12	Cingulum_Ant_R					38	Fusiform_R
						39	Hippocampus_L
						40	Hippocampus_R
						41	ParaHippocampal_L
						42	ParaHippocampal_R

For example, in Fig. 2-1, as λ decreases below λ_1 (but still above λ_2), region "Tempora_Sup_L" is the first one becoming disconnected from the rest of the brain. As λ decreases below λ_2 (but still above λ_3), the rest of the brain further divides into three disconnected clusters, including the cluster of "Cingulum_Post_R" and "Cingulum_Post_L", the cluster of "Fusiform_R" up to "Hippocampus_L", and the cluster of the other regions. As λ continuously decreases, each current cluster will split into smaller clusters; eventually, when λ reaches a very large value, there will be no arc in the IC model, i.e., each region is now a cluster of itself and the split will stop. The

sequence of the splitting gives an order for the strength of connection between brain regions. Specifically, the earlier (i.e., smaller λ) a region or a cluster of regions becomes disconnected from the rest of the brain, the weaker it is connected with the rest of the brain. For example, in Fig. 2-1, it can be known that "Tempora_Sup_L" may be the weakest region in the brain network of AD; the second weakest ones are the cluster of "Cingulum_Post_R" and "Cingulum_Post_L", and the cluster of "Fusiform_R" up to "Hippocampus_L". It is very interesting to see that the weakest and second weakest brain regions in the brain network include "Cingulum_Post_R" and "Cingulum_Post_L" as well as regions all in the temporal lobe, all of which have been found to be affected by AD early and severely [3]-[5].

Next, to facilitate the comparison between AD and NC, a tree-like plot is also constructed for NC, as shown in Fig. 2-2. By comparing the plots for AD and NC, we can observe the following two distinct phenomena: First, in AD, between-lobe connectivity tends to be weaker than within-lobe connectivity. This can be seen from Fig. 2-1 which shows a clear pattern that the lobes become disconnected with each other before the regions within each lobe become disconnected with each other, as λ goes from small to large. This pattern does not show in Fig. 2-2 for NC. Second, the same brain regions in the left and right hemisphere are connected much weaker in AD than in NC. This can be seen from Fig. 2-2 for NC, in which the same brain regions in the left and right hemisphere are still connected even at a very large λ for NC. However, this pattern does not show in Fig. 2-1 for AD.

Furthermore, a tree-like plot is also constructed for MCI (Fig. 2-3), and compared with the plots for AD and NC. In terms of the two phenomena discussed previously, MCI shows similar patterns to AD, but these patterns are not as distinct from NC as AD. Specifically, in terms of the first phenomenon, MCI also shows weaker between-lobe

connectivity than within-lobe connectivity, which is similar to AD. However, the degree of weakerness is not as distinctive as AD. For example, a few regions in the temporal lobe of MCI, including "Temporal_Mid_R" and "Temporal_Sup_R", appear to be more strongly connected with the occipital lobe than with other regions in the temporal lobe. In terms of the second phenomenon, MCI also shows weaker between-hemisphere connectivity in the same brain region than NC. However, the degree of weakerness is not as distinctive as AD. For example, several left-right pairs of the same brain regions are still connected even at a very large λ, such as "Rectus_R" and "Rectus_L", "Frontal_Mid_Orb_R" and "Frontal_Mid_Orb _L", "Parietal_Sup_R" and "Parietal_Sup_L", as well as "Precuneus_R" and "Precuneus_L". All above findings are consistent with the knowledge that MCI is a transition stage between normal aging and AD.

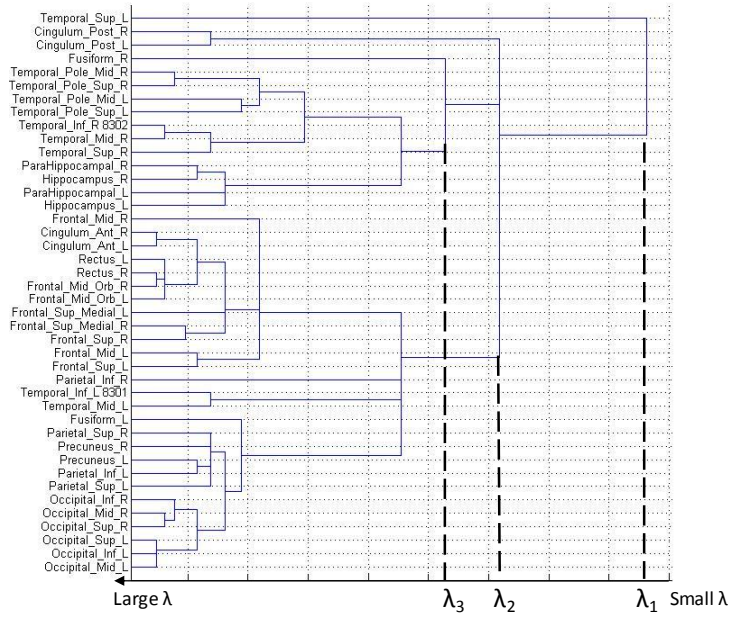


Fig 2-1: Order for the strength of connection between brain regions of AD

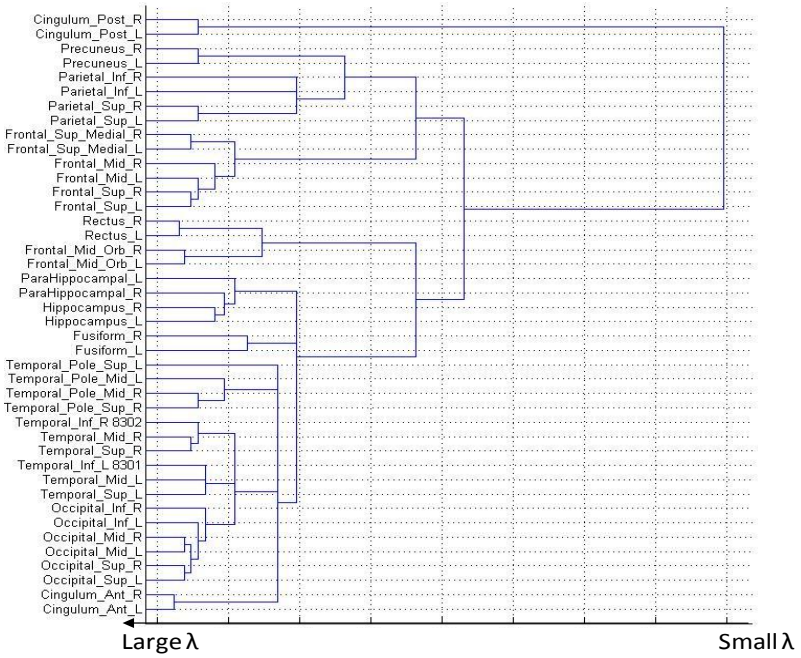


Fig 2-2: Order for the strength of connection between brain regions of NC

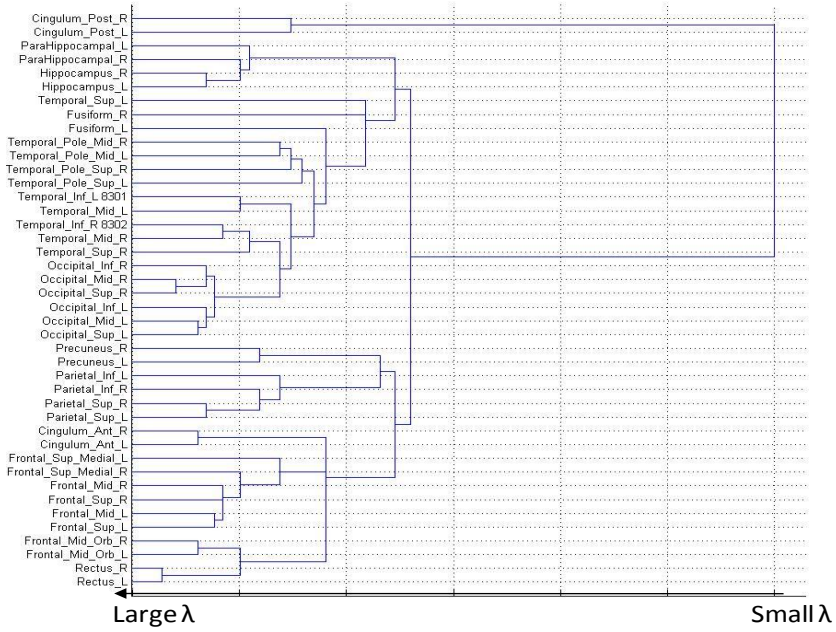


Fig 2-3: Order for the strength of connection between brain regions of MCI

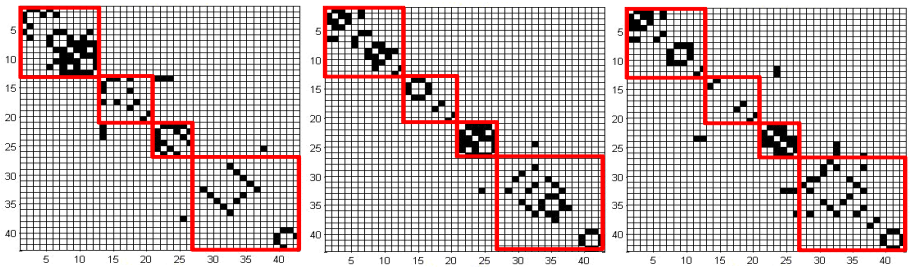
Furthermore, we would like to compare how within-lobe and between-lobe connectivity is different across AD, MCI, and NC. To achieve this, we first learn one connectivity model for AD, one for MCI, and one for NC. We adjust the λ in the learning of each

model such that the three models, corresponding to AD, MCI, and NC, respectively, will have the same total number of arcs. This is to "normalize" the models, so that the comparison will be more focused on how the arcs distribute differently across different models. By selecting different values for the total number of arcs, we can obtain models representing the brain connectivity at different levels of strength. Specifically, given a small value for the total number of arcs, only strong arcs will show up in the resulting connectivity model, so the model is a model of strong brain connectivity; when increasing the total number of arcs, mild arcs will also show up in the resulting connectivity model, so the model is a model of mild and strong brain connectivity.

For example, Fig. 2-4 shows the connectivity models for AD, MCI, and NC with the total number of arcs equal to 50 (Fig. 2-4(a)), 120 (Fig. 2-4(b)), and 180 (Fig. 2-4(c)). In this paper, we use a "matrix" representation for the SICE of a connectivity model. In the matrix, each row represents one node and each column also represents one node. Please see Table 2-1 for the correspondence between the numbering of the nodes and the brain region each number represents. The matrix contains black and white cells: a black cell at the i-th row, j-th column of the matrix represents existence of an arc between nodes X_i and X_j in the SICE-based connectivity model, whereas a white cell represents absence of an arc. According to this definition, the total number of black cells in the matrix is equal to twice the total number of arcs in the SICE-based connectivity model. Moreover, on each matrix, four red cubes are used to highlight the brain regions in each of the four lobes; that is, from top-left to bottom-right, the red cubes highlight the frontal, parietal, occipital, and temporal lobes, respectively. The black cells inside each red cube reflect within-lobe connectivity, whereas the black cells outside the cubes reflect between-lobe connectivity.

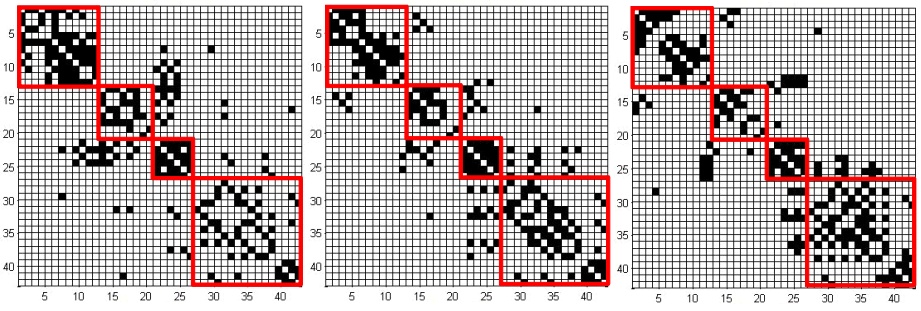
While the connectivity models in Fig. 2-4 clearly show some connectivity difference between AD, MCI, and NC, it is highly desirable to test if the observed difference is statistically significant. Therefore, we further perform a hypothesis testing and the results are summarized in Table 2-2. Specifically, a P-value is recorded in the sub-table if it is smaller than 0.1, such a P-value is further highlighted if it is even smaller than 0.05; a "---" indicates that the corresponding test is not significant (P-value>0.1). Inspection of the results in Fig. 2-4 and Table 2-2 reveals the following interesting observations:

Within-lobe connectivity: The temporal lobe of AD has significantly less connectivity than NC. This is true across different strength levels (e.g., strong, mild, and weak) of the connectivity; in other words, even the connectivity between some strongly-connected brain regions in the temporal lobe may be disrupted by AD. In particular, it is clearly from Fig. 2-4(b) that the regions "Hippocampus" and "ParaHippocampal" (numbered by 39-42, located at the right-bottom corner of Fig. 2-4(b)) are much more separated from other regions in AD than in NC. The decrease in connectivity in the temporal lobe of AD, especially between the Hippocampus and other regions, has been extensively reported in the literature [3]-[5]. Furthermore, the temporal lobe of MCI does not show a significant decrease in connectivity, compared with NC. This may be because MCI does not disrupt the temporal lobe as badly as AD.



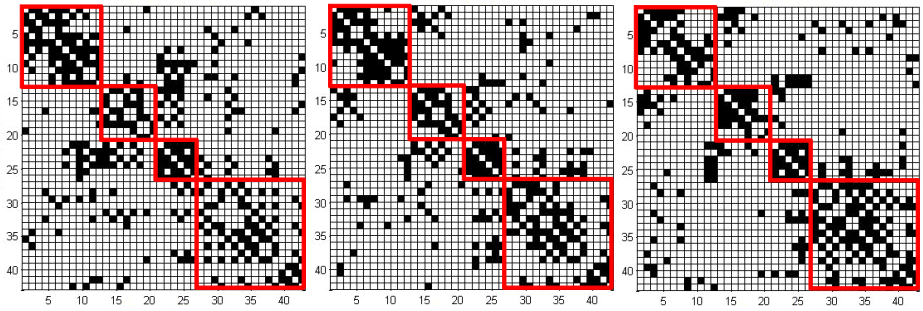
AD MCI NC

Fig 2-4(a): SICE-based brain connectivity models (total number of arcs equal to 50)



AD MCI NC

Fig 2-4(b): SICE-based brain connectivity models (total number of arcs equal to 120)



AD MCI NC

Fig 2-4(c): SICE-based brain connectivity models (total number of arcs equal to 180)

The frontal lobe of AD has significantly more connectivity than NC, which is true across different strength levels of the connectivity. This has been interpreted as compensatory reallocation or recruitment of cognitive resources [6]-[7]. Because the regions in the frontal lobe are typically affected later in the course of AD (our data are early AD), the increased connectivity in the frontal lobe may help preserve some cognitive functions in AD patients. Furthermore, the frontal lobe of MCI does not show a significant increase in connectivity, compared with NC. This indicates that the compensatory effect in MCI brain may not be as strong as that in AD brains.

Table 2-2: P-values from the statistical significance test of connectivity difference between AD, MCI, and NC

(a) Total number of arcs = 50 (b) Total number of arcs = 120 (c) Total number of arcs = 180

AD vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	0.053	---	---	---
Parietal		---	0.083	---
Occipital			---	---
Temporal				0.091

AD vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	0.033	---	---	---
Parietal		---	0.007	0.095
Occipital			---	0.018
Temporal				0.039

AD vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	0.067	0.058	0.096	---
Parietal		---	0.019	0.011
Occipital			---	---
Temporal				0.017

AD vs. MCI	Frontal	Parietal	Occipital	Temporal
Frontal	0.085	---	---	---
Parietal		---	0.088	---
Occipital			0.054	---
Temporal				0.093

AD vs. MCI	Frontal	Parietal	Occipital	Temporal
Frontal	---	0.055	0.022	---
Parietal		---	---	---
Occipital			---	0.019
Temporal				---

AD vs. MCI	Frontal	Parietal	Occipital	Temporal
Frontal	---	0.063	0.004	---
Parietal		---	---	---
Occipital			---	0.058
Temporal				---

MCI vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	---	---	---	---
Parietal		---	---	---
Occipital			---	---
Temporal				---

MCI vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	---	---	0.024	---
Parietal		---	0.052	---
Occipital			---	---
Temporal				---

MCI vs. NC	Frontal	Parietal	Occipital	Temporal
Frontal	---	---	0.061	---
Parietal		---	0.0504	---
Occipital			---	0.041
Temporal				---

There is no significant difference between AD, MCI, and NC in terms of the connectivity within the parietal lobe and within the occipital lobe. Another interesting finding is that all the P-values in the third sub-table of Table 2-2(a) are insignificant. This implies that distribution of the strong connectivity within and between lobes for MCI is very similar to NC; in other words, MCI has not been able to disrupt the strong connectivity among brain regions (it disrupts some mild and weak connectivity though).

Between-lobe connectivity: In general, human brains tend to have less between-lobe connectivity than within-lobe connectivity. A majority of the strong connectivity occurs within lobes, but rarely between lobes. These can be clearly seen from Fig 2-4 (especially Fig. 2-4(a)) in which there are much more black cells along the diagonal direction than the off-diagonal direction, regardless of AD, MCI, and NC.

The connectivity between the parietal and occipital lobes of AD is significantly more than NC which is true especially for mild and weak connectivity. The increased connectivity between the parietal and occipital lobes of AD has been previously reported in [3]. It is also interpreted as a compensatory effect in [6]-[7]. Furthermore, MCI also shows increased connectivity between the parietal and occipital lobes, compared with NC, but the increase is not as significant as AD.

While the connectivity between the frontal and occipital lobes shows little difference between AD and NC, such connectivity for MCI shows a significant decrease especially for mild and weak connectivity. Also, AD may have less temporal-occipital connectivity, less frontal-parietal connectivity, but more parietal-temporal connectivity than NC.

Between-hemisphere connectivity: Recall that we have observed from the tree-like plots in Figs. 2-3 and 2-4 that the same brain regions in the left and right hemisphere are connected much weaker in AD than in NC. It is desirable to test if this observed difference is statistically significant. To achieve this, we test the statistical significance of the difference between AD, MCI, and NC, in term of the number of connected same-region left-right pairs. Results show that when the total number of arcs in the connectivity models is equal to 120 or 90, none of the tests is significant. However, when the total number of arcs is equal to 50, the P-values of the tests for "AD vs. NC", "AD vs. MCI", and "MCI vs. NC" are 0.009, 0.004, and 0.315, respectively. We further perform tests for the total number of arcs equal to 30 and find the P-values to be 0. 0055, 0.053, and 0.158, respectively. These results indicate that AD disrupts the strong connectivity between the same regions of the left and right hemispheres, whereas this disruption is not significant in MCI.

2.4 Conclusion

In the paper, we applied SICE to model functional brain connectivity of AD, MCI, and NC based on PET neuroimaging data. Our findings were consistent with the previous literature and also showed some new aspects that may suggest further investigation in brain connectivity research in AD.

Appendix I: The proposed SICE algorithm

This section details our approach for estimating sparse inverse covariance matrix from data, which can be achieved through solving for the optimization problem in (1). Our

approach is based on the block coordinate descent (BCD) algorithm, but with an extended capacity of allowing for prior domain knowledge to be incorporated into the problem solving process.

The basic idea of the BCD algorithm is to update each column (or row) of $\mathbf{\Theta}$ iteratively while fixing all other columns (or rows), until convergence. Because the BCD algorithm works by iterations, we will only illustrate the steps in one iteration and other iterations work in a similar way. At a certain iteration, we first need to partition the current $\mathbf{\Theta}$ as follows. Let $\mathbf{\Theta}_{\backslash j \backslash j}$ be the matrix produced by removing row j and column j from $\mathbf{\Theta}$, θ_{jj} be the element at row j and column j of $\mathbf{\Theta}$, and $\mathbf{\Theta}_j$ be the column j of $\mathbf{\Theta}$ with θ_{jj} removed. Then, $\mathbf{\Theta}$ can be partitioned as $\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Theta}_{\backslash j \backslash j} & \mathbf{\Theta}_j \\ \mathbf{\Theta}_j^T & \theta_{jj} \end{bmatrix}$, and correspondingly $\mathbf{S}$ can be partitioned as $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\backslash j \backslash j} & \mathbf{S}_j \\ \mathbf{S}_j^T & s_{jj} \end{bmatrix}$. Next, we want to update $\mathbf{\Theta}_j$ and θ_{jj} while holding other elements in $\mathbf{\Theta}$ constant. To do this, let f represent the objective function in (1), i.e., $f = log|\mathbf{\Theta}| - tr(\mathbf{S\Theta}) - \lambda\|\mathbf{\Theta}\|_1$; take the partial derivatives of f with respect to $\mathbf{\Theta}_j$ and θ_{jj}, respectively; and then make the partial derivatives to be zero, i.e.,

$$\frac{\partial f}{\partial \mathbf{\Theta}_j} = -\frac{2}{\theta_{jj} - \mathbf{\Theta}_j^T \mathbf{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{\Theta}_j} \mathbf{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{\Theta}_j - \mathbf{S}_j - \lambda\, SGN\left(\mathbf{\Theta}_j\right) = 0, \qquad \text{(A-1)}$$

$$\frac{\partial f}{\partial \theta_{jj}} = \frac{1}{\theta_{jj} - \mathbf{\Theta}_j^T \mathbf{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{\Theta}_j} - s_{jj} - \lambda = 0, \qquad \text{(A-2)}$$

where $SGN\left(\mathbf{\Theta}_j\right)$ denotes the partial derivative of $\|\mathbf{\Theta}\|_1$ with respect to $\mathbf{\Theta}_j$. It is difficult to solve for $\mathbf{\Theta}_j$ and θ_{jj} from (A-1) and (A-2) directly. Therefore, we adopt the following strategies.

Letting $\mathbf{a} = -\frac{\mathbf{\Theta}_j}{\theta_{jj} - \mathbf{\Theta}_j^T \mathbf{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{\Theta}_j}$, then (A-1) and (A-2) become

$$2\mathbf{\Theta}_{\backslash j \backslash j}^{-1}\mathbf{a} - \mathbf{S}_j + \lambda\, SGN(\mathbf{a}) = 0 \qquad \text{(A-3)}$$

$$\mathbf{a} = -\left(s_{jj} + \lambda\right)\mathbf{\Theta}_j. \qquad \text{(A-4)}$$

It is clear that (A-3) is also the result of making the partial derivative of g with respect to $\mathbf{a}$ to be zero in the following optimization problem:

$$\min_{\mathbf{a}} g = \mathbf{a}^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{a} - \mathbf{S}_j^T \mathbf{a} + \lambda \|\mathbf{a}\|_1, \quad \text{(A-5)}$$

which is equivalent to the following min-max problem:

$$\max_{\boldsymbol{\kappa}} \min_{\mathbf{a}} g = 2\left(-\frac{1}{2}\boldsymbol{\kappa}^T \Theta_{\backslash j \backslash j} \boldsymbol{\kappa} + \boldsymbol{\kappa}^{\mathrm{T}} \mathbf{a}\right) - \mathbf{S}_j^T \mathbf{a} + \lambda \|\mathbf{a}\|_1. \quad \text{(A-6)}$$

This min-max problem can be solved by the prox method.

After $\mathbf{a}$ and $\boldsymbol{\kappa}$ are obtained, (A-4) can be used to find $\boldsymbol{\Theta}_j$, i.e., $\boldsymbol{\Theta}_j = -\frac{\mathbf{a}}{s_{jj}+\lambda}$. Furthermore, based on (A-2), θ_{jj} can be obtained, i.e., $\theta_{jj} = \frac{(-\mathbf{a}^T \boldsymbol{\kappa} + \mathbf{1})}{s_{jj}+\lambda}$.

Furthermore, suppose that some prior domain knowledge is available, e.g., nodes X_i and X_j are disconnected in the IC model, which means that $\theta_{ij} = 0$ in $\boldsymbol{\Theta}$. Then, we can forces the corresponding entry in $\mathbf{a}$ to be zero in each iteration. As a result, we can re-formulate (A-5) as follows:

$$\min_{\mathbf{a}} g = \mathbf{a}^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{a} - \mathbf{S}_j^T \mathbf{a} + \lambda \|\mathbf{a}\|_1$$

$$s.t. \, \mathbf{a}_i = \mathbf{0} \quad if \quad i \in \mathbf{V}$$

where $\mathbf{V}$ is the set of indices (based on prior domain knowledge) corresponding to zero entries in $\mathbf{a}$. Note that this problem is also strictly convex and can be solved efficiently. □

Appendix II: Proof of the monotone property of the SICE algorithm

A sufficient and necessary condition of the monotone property is as follow:

Theorem 1: Let $\left\{\mathbf{C}_1^{\lambda_1}, \ldots, \mathbf{C}_{L_1}^{\lambda_1}\right\}$ and $\left\{\mathbf{C}_1^{\lambda_2}, \ldots, \mathbf{C}_{L_2}^{\lambda_2}\right\}$ denote the clusters of nodes in the SICE-based graphical models, with λ equal to λ_1 and λ_2 ($\lambda_1 < \lambda_2$), respectively. Then, for any $C_i^{\lambda_2}$, $i \in \{1,2,\ldots,L_2\}$, there must exist a $C_j^{\lambda_1}$, $j \in \{1,2,\ldots,L_1\}$ such that $C_i^{\lambda_2} \subseteq C_j^{\lambda_1}$.

This section proves the monotone property by proving that Theorem 1 is true.

(1) can be equivalently written as

$$\widehat{\boldsymbol{\Sigma}} \;=\; \text{argmin}\; log\, det(\boldsymbol{\Sigma}) + tr(\mathbf{S}\boldsymbol{\Sigma}^{-1}) + \lambda\|\boldsymbol{\Sigma}^{-1}\|_1 . \tag{B-1}$$

It is known from [17] that the solution, $\widehat{\boldsymbol{\Sigma}}$, is unique with a fixed positive λ, and $\widehat{\boldsymbol{\Sigma}}$ must satisfy the equations in (B-2):

$$\begin{aligned} &(\mathbf{S})_{kl} - (\boldsymbol{\Sigma})_{kl} = -\lambda, \quad \text{for } \left(\boldsymbol{\Sigma}^{-1}\right)_{kl} > 0; \\ &(\mathbf{S})_{kl} - (\boldsymbol{\Sigma})_{kl} = \lambda, \quad \text{for } \left(\boldsymbol{\Sigma}^{-1}\right)_{kl} < 0; \\ &|(\mathbf{S})_{kl} - (\boldsymbol{\Sigma})_{kl}| \le \lambda, \quad \text{for } \left(\boldsymbol{\Sigma}^{-1}\right)_{kl} = 0; \end{aligned} \tag{B-2}$$

where $(\cdot)_{kl}$ denotes the element at the k-th row, l-th column of a matrix.

When $\lambda = \lambda_1$, denote the solution to (B-1) by $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$. Furthermore, we can rearrange the rows and columns of $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$, such that $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$ becomes a block diagonal matrix and each sub-matrix along the main diagonal of the rearranged $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$ correspond to a cluster of nodes in the SICE-based graphical model. Denote the sub-matrices by $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}_{\mathbf{C}_j^{\lambda_1}}, j = 1, \dots, L_1$. Recall that $\mathbf{C}_j^{\lambda_1}$is the j-th cluster of nodes in the graphical model. As a result, $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$ can be written as:

$$\widehat{\boldsymbol{\Sigma}}^{\lambda_1} = \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}^{\lambda_1}_{\mathbf{C}_1^{\lambda_1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \widehat{\boldsymbol{\Sigma}}^{\lambda_1}_{\mathbf{C}_2^{\lambda_1}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \widehat{\boldsymbol{\Sigma}}^{\lambda_1}_{\mathbf{C}_{L_1}^{\lambda_1}} \end{bmatrix}. \tag{B-3}$$

A sufficient condition for Theorem 1 being true is that the solution to (B-1) when $\lambda = \lambda_2$, denoted by $\widehat{\boldsymbol{\Sigma}}^{\lambda_2}$, must share the same structure as (B-3), i.e., $\widehat{\boldsymbol{\Sigma}}^{\lambda_2}$ can be written as:

$$\widehat{\boldsymbol{\Sigma}}^{\lambda_2} = \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}^{\lambda_2}_{\mathbf{C}_1^{\lambda_1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \widehat{\boldsymbol{\Sigma}}^{\lambda_2}_{\mathbf{C}_2^{\lambda_1}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \widehat{\boldsymbol{\Sigma}}^{\lambda_2}_{\mathbf{C}_{L_1}^{\lambda_1}} \end{bmatrix}. \tag{B-4}$$

To prove this sufficient condition, our strategy will include two steps: step one aims to find a matrix having the same structure as $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$; step two aims to prove that this matrix is a solution to (B-2) with $\lambda = \lambda_2$.

Step One:

The rows and columns of the sample covariance matrix, **S**, can be rearranged in the same way as $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$, i.e.,

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathbf{C}_1^{\lambda_1}} & \cdots & \cdots & \cdots \\ \cdots & \mathbf{S}_{\mathbf{C}_2^{\lambda_1}} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \mathbf{S}_{\mathbf{C}_{L_1}^{\lambda_1}} \end{bmatrix}. \tag{B-5}$$

Next, one optimization problem can be formulated corresponding to one sub-matrix $\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}$, $j = 1, \ldots, L_1$, i.e.,

$$\widehat{\boldsymbol{\Upsilon}}_{\mathbf{C}_j^{\lambda_1}}^{\lambda_2} = \operatorname{argmin} log\ det\left(\boldsymbol{\Upsilon}_{\mathbf{C}_j^{\lambda_1}}^{\lambda_2}\right) + tr\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\left(\boldsymbol{\Upsilon}_{\mathbf{C}_j^{\lambda_1}}^{\lambda_2}\right)^{-1}\right) + \lambda_2 \left\|\left(\boldsymbol{\Upsilon}_{\mathbf{C}_j^{\lambda_1}}^{\lambda_2}\right)^{-1}\right\|_1, \tag{B-6}$$

Furthermore, the solutions to (B-6), i.e., $\widehat{\boldsymbol{\Upsilon}}_{\mathbf{C}_j^{\lambda_1}}^{\lambda_2}$, $j = 1, \ldots, L_1$, can be put together and form a big matrix $\widehat{\boldsymbol{\Upsilon}}^{\lambda_2}$, i.e.,

$$\widehat{\boldsymbol{\Upsilon}}^{\lambda_2} = \begin{bmatrix} \widehat{\boldsymbol{\Upsilon}}_{\mathbf{C}_1^{\lambda_1}}^{\lambda_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \widehat{\boldsymbol{\Upsilon}}_{\mathbf{C}_2^{\lambda_1}}^{\lambda_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \widehat{\boldsymbol{\Upsilon}}_{\mathbf{C}_{L_1}^{\lambda_1}}^{\lambda_2} \end{bmatrix}. \tag{B-7}$$

It is obvious that $\widehat{\boldsymbol{\Upsilon}}^{\lambda_2}$ has the same structure as $\widehat{\boldsymbol{\Sigma}}^{\lambda_1}$.

Step Two:

This step aims to prove that the $\widehat{\mathbf{\Upsilon}}^{\lambda_2}$ in (B-7) satisfies (B-2) with $\lambda = \lambda_2$. To prove this, we need to prove that (i) the elements in $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$, $j = 1, \dots, L_1$, satisfy (B-2), and that (ii) the elements not in $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$, all of which are equal to zero, also satisfy (B-2).

(i) Suppose that $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}$ is an element in $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$, $j \in \{1, \dots, L_1\}$; more specifically, suppose that $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}$ is the element at the h-th row, s-th column of $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$, i.e.,

$$\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} = \left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}. \tag{B-8}$$

Because $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$ is the solution to the optimization in (B-6), it must satisfy (B-9):

$$\begin{aligned}
&\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} - \left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} = -\lambda_2, \quad \text{for } \left(\left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)^{-1}\right)_{hs} > 0; \\
&\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} - \left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} = \lambda_2, \quad \text{for} \left(\left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)^{-1}\right)_{hs} < 0; \\
&\left|\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} - \left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs}\right| \le \lambda_2, \quad \text{for } \left(\left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)^{-1}\right)_{hs} = 0;
\end{aligned} \tag{B-9}$$

It is easy to know that $\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs}$ is in fact the element at the k-th row, l-th column of $\mathbf{S}$, i.e.,

$$\left(\mathbf{S}_{\mathbf{C}_j^{\lambda_1}}\right)_{hs} = (\mathbf{S})_{kl}; \tag{B-10}$$

and $\left(\left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)^{-1}\right)_{hs}$ is the element at the k-th row, l-th column of $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)^{-1}$, i.e.,

$$\left(\left(\mathbf{\Upsilon}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}\right)^{-1}\right)_{hs} = \left(\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)^{-1}\right)_{kl}. \tag{B-11}$$

Inserting (B-8), (B-10), and (B-11) into (B-9) results in (B-2) with $\lambda = \lambda_2$.

(ii) Suppose that $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}$ is an element not in $\widehat{\mathbf{\Upsilon}}^{\lambda_2}_{\mathbf{C}_j^{\lambda_1}}$, $j = 1, \ldots, L_1$, i.e., $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl} = 0$. Furthermore, it can be known that $\left(\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)^{-1}\right)_{kl} = 0$, because $\widehat{\mathbf{\Upsilon}}^{\lambda_2}$ is a block diagonal matrix. Since $\left(\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)^{-1}\right)_{kl} = 0$, to prove that $\left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}$ satisfies (B-2) with $\lambda = \lambda_2$ is to prove that $\left|(\mathbf{S})_{kl} - \left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}\right| \leq \lambda_2$. It can be derive that $\left|(\mathbf{S})_{kl} - \left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}\right| = |(\mathbf{S})_{kl}| = \left|(\mathbf{S})_{kl} - \left(\widehat{\mathbf{\Sigma}}^{\lambda_1}\right)_{kl}\right| \leq \lambda_1$, where the second equality holds because $\left(\widehat{\mathbf{\Sigma}}^{\lambda_1}\right)_{kl} = 0$, and the "$\leq$" holds due to the last equation in (B-1) with $\lambda = \lambda_1$. Also, it has been known that $\lambda_1 \leq \lambda_2$. Therefore, $\left|(\mathbf{S})_{kl} - \left(\widehat{\mathbf{\Upsilon}}^{\lambda_2}\right)_{kl}\right| \leq \lambda_1 \leq \lambda_2$. □

References

[1] S.Molchan. (2005) The Alzheimer's disease neuroimaging initiative. Business Briefing: US Neurology Review, pp.30-32, 2005.

[2] C.J.Stam, B.F. Jones, G.Nolte, M. Breakspear, and P. Scheltens. (2007) Small-world networks and functional connectivity in Alzheimer's disease. Cerebral Corter 17:92-99.

[3] K. Supekar, V. Menon, D. Rubin, M. Musen, M.D. Greicius. (2008) Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer's Disease. PLoS Comput Biol 4(6) 1-11.

[4] K. Wang, M. Liang, L. Wang, L. Tian, X. Zhang, K. Li and T. Jiang. (2007) Altered Functional Connectivity in Early Alzheimer's Disease: A Resting-State fMRI Study, Human Brain Mapping 28, 967-978.

[5] N.P. Azari, S.I. Rapoport, C.L. Grady, M.B. Schapiro, J.A. Salerno, A.

Gonzales-Aviles. (1992) Patterns of interregional correlations of cerebral glucose metabolic rates in patients with dementia of the Alzheimer type. Neurodegeneration 1: 101–111.

[6] R.L. Gould, B.Arroyo, R,G. Brown, A.M. Owen, E.T. Bullmore and R.J. Howard. (2006) Brain Mechanisms of Successful Compensation during Learning in Alzheimer Disease, Neurology 67, 1011-1017.

[7] Y. Stern. (2006) Cognitive Reserve and Alzheimer Disease, Alzheimer Disease Associated Disorder 20, 69-74.

[8] Friston, K.J. (1994) Functional and effective connectivity: A synthesis. Human Brain Mapping 2, 56-78.

[9] Alexander, G., Moeller, J. (1994) Application of the Scaled Subprofile model: a statistical approach to the analysis of functional patterns in neuropsychiatric disorders: A principal component approach to modeling regional patterns of brain function in disease. Human Brain Mapping, 79-94.

[10] Calhoun, V.D., Adali, T., Pearlson, G.D., Pekar, J.J. (2001) Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms. Hum.Brain Mapp. 13, 43-53.

[11] Calhoun, V.D., Adali, T., Pekar, J.J., Pearlson, G.D. (2003) Latency (in)sensitive ICA. Group independent component analysis of fMRI data in the temporal frequency domain. Neuroimage. 20, 1661-1669.

[12] McIntosh, A.R., Bookstein, F.L., Haxby, J.V., Grady, C.L. (1996) Spatial pattern analysis of functional brain images using partial least squares. Neuroimage. 3, 143-157.

[13] Worsley, K.J., Poline, J.B., Friston, K.J., Evans, A.C. (1997) Characterizing the response of PET and fMRI data using multivariate linear models. Neuroimage. 6, 305-319.

[14] Bullmore, E., Horwitz, B., Honey, G., Brammer, M., Williams, S., Sharma, T. (2000) How good is good enough in path analysis of fMRI data? NeuroImage 11, 289–301.

[15] McIntosh, A.R., Grady, C.L., Ungerieider, L.G., Haxby, J.V., Rapoport, S.I., Horwitz, B. (1994) Network analysis of cortical visual pathways mapped with PET. J. Neurosci. 14 (2), 655–666.

[16] Friston, KJ, Harrison, L, Penny, W. (2003) Dynamic causal modelling. Neuroimage 19, 1273-1302.

[17] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. (2008) Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. Journal of Machine Learning Researc 9:485-516.

[18] J. Dahl, L. Vandenberghe, and V. Roycowdhury. (2008) Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods Software 23(4):501-520.

[19] J. Friedman, T.astie, and R. Tibsirani. (2007) Spares inverse covariance estimation with the graphical lasso, Biostatistics 8(1):1-10.

[20] J.Z. Huang, N. Liu, M. Pourahmadi, and L. Liu. (2006) Covariance matrix selection and estimation via penalized normal likelihood. Biometrika, 93(1):85-98.

[21] H. Li and J. Gui. (2005) Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. Biostatistics 7(2):302-317.

[22] Y. Lin. (2007) Model selection and estimation in the gaussian graphical model. Biometrika 94(1)19-35, 2007.

[23] A. Dobra, C. Hans, B. Jones, J.R. Nevins, G. Yao, and M. West. (2004) Sparse graphical models for exploring gene expression data. Journal of Multivariate Analysis 90(1):196-212.

[24] A. Berge, A.C. Jensen, and A.H.S. Solberg. (2007) Sparse inverse covariance estimates for hyperspectral image classification, Geoscience and Remote Sensing, IEEE Transactions on, 45(5):1399-1407.

[25] J.A. Bilmes. (2000) Factored sparse inverse covariance matrices. In ICASSP:1009-1012.

[26] D.L. Donoho. (2006) For most large underdetermined systems of linear equations, the minimal l1-orm near-solution approximates the sparest near-solution. Communications on Pure and Applied Mathematics 59(7):907-934.

[27] R. Tibshirani. (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B 58(1):267-288.

[28] N. Tzourio-Mazoyer and et al. (2002) Automated anatomical labelling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single subject brain. Neuroimage 15:273-289.

Chapter 3

A SPARSE STRUCTURE LEARNING ALGORITHM FOR GAUSSIAN BAYESIAN NETWORK IDENTIFICATION FROM HIGH-DIMENSIONAL DATA

Abstract

Structure learning of Bayesian networks (BNs) is an important topic in machine learning. Driven by modern applications in genetics and brain sciences, accurate and efficient learning of large-scale BN structures from high-dimensional data becomes a challenging problem. To tackle this challenge, we propose a Sparse Bayesian Network (SBN) structure learning algorithm that employs a novel formulation involving one L1-norm penalty term to impose sparsity and another penalty term to ensure that the learned BN is a directed acyclic graph—a required property of BNs. Through both theoretical analysis and extensive experiments on eleven moderate and large benchmark networks with various sample sizes, we show that SBN leads to improved learning accuracy, scalability, and efficiency as compared with ten existing popular BN learning algorithms. We apply SBN to a real-world application of brain connectivity modeling for Alzheimer's disease (AD) and reveal findings that could lead to advancements in AD research.

3.1 Introduction

A Bayesian network (BN) is a graphical model for representing the probabilistic relationships among variables. BNs have been widely used in the fields of genetics [1], [2], ecology [3], [4], social sciences [5], medical sciences [6], brain sciences [7], [8], and manufacturing [9]. A BN consists of two components: the structure, which is a Directed Acyclic Graph (DAG) for representing the dependency and independency among variables, and a set of parameters for representing the quantitative information of the

dependency. Accordingly, learning a BN from data includes structure learning and parameter learning. This paper focuses on structure learning.

One type of structure learning method is constraint-based. Constraint-based methods [10], [11], [12], [13], [14] use conditional independence tests to identify the dependent and independent relationships among variables. A major weakness of these methods is that too many tests may have to be performed with each test being built upon the results of another, leading to escalated errors in the BN structure identification.

Another type of structure learning method is score-based, in which a "score" is defined for each possible BN structure and then a search algorithm is used to find the structure with the highest score. Various score functions have been proposed, including those based on the Bayesian method [15], [16], [17], [18], [19], minimum description length [20], [21], [22], [23], and entropy [10], [24]. Furthermore, once a score function is specified, a search method is needed to find the structure with the highest score. Because the number of possible structures grows exponentially with respect to the number of variables, an exhaustive search over all possible structures may be computationally too expensive or unfeasible. Therefore, various inexact search methods have been proposed, such as heuristic search techniques [15], [24], [25], [26], genetic algorithms [28], [29], simulated annealing [30].]. Sampling methods such as Markov Chain Monte Carlo (MCMC) [18], [24] have also been utilized to travel through the DAG space. These methods usually find a BN structure that is a local optimum, and have been less effective in high-dimensional DAG spaces. In addition, some work has been done to combine score-based methods with constraint-based methods [31]. Then there is the recently developed novel additive noise model [32], which differs from both constraint-based and score-based methods and has the advantage of learning nonlinear interactions for non-gaussian BNs.

Driven by modern applications in brain sciences and genetics, there has been a great need of algorithms capable of learning large BN structures with high accuracy and efficiency from limited samples. For example, BNs provide an effective tool for identifying how different brain regions interact with each other in task performance, skill learning, and disease processes from neuroimaging data [7], [8]. A typical neuroimaging dataset includes hundreds of variables (brain regions) while the sample size (number of experimental subjects) is usually in tens. Also, BNs are very useful for modeling the interacting patterns between genes from microarray gene expression data, which measures thousands of genes with sample size being no more than a few hundred [1], [2].

For the purpose of learning a large BN with small sample sizes, a useful strategy is to impose a "sparsity" constraint of some kind. Many real-world networks are indeed sparse, such as the gene association networks [1], [33] and brain connectivity networks [34]. When learning the structure of these networks, a sparsity constraint helps prevent overfitting and improves computational efficiency. For example, the Sparse Candidate (SC) algorithm [35], one of the first large-scale BN structure learning algorithms, achieves sparsity by assuming that the maximum number of parents for each node is limited to a small constant. One major problem with SC is that the user has to guess the maximum number of parents. Also, it is usually unrealistic to assume that all the nodes have the same maximum number of parents. The L1MB-DAG algorithm [36] does not require a prior specification on the maximum number of parents. Instead, it uses LASSO to select a small set of potential parents for each variable. LASSO is known for sparse variable selection [37].

In addition to the sparsity consideration, recently developed BN structure learning methods ususally consist of two stages: Stage 1 is to identify the potential parents of each variable; Stage 2 applies some search methods to identify the parents out of the potential

parent set. The advantage of the two-stage approach is improved efficiency, as Stage 2 is a local search over a possibly small set of potential parents for each variable identified by Stage 1, rather than a global search over all the variables. The two-stage approach has been popularly adopted by many existing algorithms, including the SC and the L1MB-DAG algorithms, mentioned previously, as well as the Hill-Climbing (MMHC) [38], the Grow-Shrink [39], the TC and the TC-bw [40] algorithms. The difference between these algorithms primarily lies in how they identify the potential parent set in Stage 1. For example, L1MB-DAG uses LASSO, MMHC uses the G2 statistic, and TC and TC-bw use a t-test. An apparent weakness of the two-stage approach is that if a true parent is missed in Stage 1, it will never be recovered in Stage 2. Another weakness of the existing algorithms is that the computational efficiency is still too low for learning large BNs. For example, it may take hours or days to learn a BN with 500 nodes.

In this paper, we propose a new sparse Gaussian BN structure learning algorithm, called SBN. It is a one-stage approach that identifies the parents of all variables directly, thus having a low risk of missing parents (i.e., a high accuracy in BN structure identification) compared with many existing algorithms that employ the two-stage approach. Specifically, in development of the SBN, we propose a novel formulation with one L1-norm penalty term to impose sparsity and another penalty term to ensure that the learned BN is a Directed Acyclic Graph (DAG)—a required property of BN. The theoretical property about how to select the regularization parameter associated with the second penalty term is discussed. Under this formulation, we propose to use the Block Coordinate Descent (BCD) and shooting algorithms to estimate the BN structure. Further, our theoretical analysis indicates that the computational complexity of SBN is linear in the sample size and quadratic in the number of variables. This characteristic makes SBN

more scalable and efficient than most existing algorithms, and thus well suited for large-scale BN structure learning from high-dimensional datasets.

In addition, we perform theoretical analysis to show why the two-stage approach popularly adopted in the existing literature has a high risk of misidentifying the true parents and how the proposed SBN overcomes this deficiency. Also, extensive experiments on synthetic data are performed to compare SBN and the existing algorithms in terms of the learning accuracy, scalability, and efficiency. Finally, we apply SBN to a real-world application of brain connectivity modeling for Alzheimer's disease (AD). In particular, SBN is applied to the neuroimaging PDG-PET data of 42 AD patients and 67 matching normal control (NC) subjects in order to identify the brain connectivity model for each of the two study groups. A connectivity model represented by a BN reveals the directional effects of one brain region over another—called the effective connectivity. Effective connectivity has been much less studied in the AD literature, as most existing work focuses on functional connectivity, i.e., the correlations among brain regions. In this sense, the application of SBN to AD has the advantage over undirected graphical models of providing new insights into the mechanisms/pathways that distinct brain regions communicate with each other. In this application, the effective connectivity model of AD identified by SBN is compared in many different ways with that of NC, including the connectivity at the global scale, intra-/inter-lobe and inter-hemisphere connectivity distribution, and the connectivity associated with specific brain regions. The findings are consistent with known pathology and the clinical progression in AD.

The rest of the paper is organized as follows: Section 3-2 introduces the key definitions and concepts of BN. Section 3-3 presents the development of SBN. Section 3-4 performs a theoretical analysis on the competitive advantage of SBN over the existing algorithms that employ the two-stage approach. Section 3-5 presents the results of the experiments

on synthetic data. Section 3-6 presents the application of SBN to brain connectivity modeling of AD. Section 3-7 is the conclusion.

3.2 Bayesian network: key definitions and concepts

In this section, we give a brief introduction to the key definitions and concepts of BNs that are relevant to this paper:

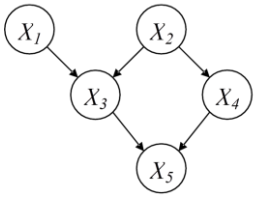


Fig. 3-1. A Bayesian Network structure (DAG).

A BN is composed by a structure and a set of parameters. The structure (Fig. 3-1) is a DAG that consists of p nodes $[X_1, \dots, X_p]$ and directed arcs between some nodes; no cycle is allowed in a DAG. Each node represents a random variable. In this paper, we will use nodes and variables interchangeably. The directed arcs encode the dependent and independent relationships among the variables. If there is a directed arc from X_i to X_j, X_i is called a *parent* of X_j and X_j is called a *child* of X_i. Two nodes are called *spouses* of each other if they share a common child. If there is a *directed path* from X_i to X_j, i.e., $X_i \to \cdots \to X_j$, X_i is called an *ancestor* of X_j. A directed arc is also a directed path and a parent is also an ancestor according to this definition. The *Markov Blanket (MB)* of X_j is a set of variables and given this set of variables, X_j will be independent of all other variables. The MB consists of the parents, children, and spouses of X_j.

In this paper, we will adopt the following notations with respect to a BN structure: we denote the structure by a $p \times p$ matrix $\mathbf{G}$, with entry $\mathbf{G}_{ij} = 1$ representing a directed arc from X_i to X_j and $\mathbf{G}_{ij} = 0$ otherwise. The set of parents of a node X_i is denoted by

$\mathbf{PA}(X_i)$. In addition, we define a $p \times p$ matrix, $\mathbf{P}$, which records all the directed paths in the structure, i.e., if there is a directed path from X_i to X_j, entry $\mathbf{P}_{ij} = 1$; otherwise, $\mathbf{P}_{ij} = 0$.

In addition to the structure, another important component of a BN is the parameters. The parameters are the conditional probability distribution of each node given its parents. Specifically, when the nodes follow a multivariate normal distribution, a regression-type parameterization can be adopted, i.e., $X_i = \boldsymbol{\beta}_i{}^T \mathbf{PA}(X_i) + \varepsilon_i$ with $\varepsilon_i \sim N(0, \sigma_i{}^2)$ and $\boldsymbol{\beta}_i$ being a vector of regression coefficients. Without loss of generality, we assume in this paper that the nodes are standardized, i.e., each with a zero mean and unit variance. Then, the parameters of a BN are $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p]$.

3.3 The proposed sparse BN structure learning algorithm – SBN

One of the challenging issues in BN structure learning is to ensure that the learned structure must be a DAG, i.e., no cycle is present. To achieve this, we first identify a sufficient and necessary condition for a DAG:

Lemma 1. *A sufficient and necessary condition for a DAG is* $\beta_{ji} \times \mathbf{P}_{ij} = 0$ *for every pair of nodes* X_i *and* X_j.

Proof. To prove the necessary condition, suppose that a BN structure, $\mathbf{G}$, is a DAG. Let's assume that $\beta_{ji} \times \mathbf{P}_{ij} \neq 0$ for a pair of nodes X_i and X_j. Then, there exists a directed path from X_j to X_i and a directed path from X_i to X_j, i.e., there is a cycle in $\mathbf{G}$, which is a contradiction to our presumption that $\mathbf{G}$ is a DAG. To prove the sufficient condition, suppose that $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for every pair of nodes X_i and X_j. If $\mathbf{G}$ is not a DAG, i.e., there is a cycle, it means that there exist two variables, X_i and X_j, with a directed arc from X_j to X_i ($\beta_{ji} \neq 0$) and a directed path from X_i to X_j ($\mathbf{P}_{ij} = 1$). This is a contradiction to our presumption that $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for every pair of nodes X_i and X_j. □

Based on Lemma 1, we further present our formulation for sparse BN structure learning. It is an optimization problem with the objective function and constraints given by:

$$\widehat{\mathbf{B}} = min_{\mathbf{B}} \sum_{i=1}^{p} \begin{Bmatrix} \left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)\left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)^T \\ +\lambda_1 \|\boldsymbol{\beta}_i\|_1 \end{Bmatrix} \quad (1)$$

$$s.t. \quad \beta_{ji} \times \mathbf{P}_{ij} = 0, \ i,j = 1, \dots, p, i \neq j.$$

According to the definition of $\mathbf{P}$, $\mathbf{P}$ is a function of $\mathbf{B}$. So the constraints in (1) are functions of $\mathbf{B}$. The notations in (1) are explained as follows: $\boldsymbol{x}_i = [x_{i1}, \dots, x_{in}]$ denote the sample vector for X_i, where n is the sample size. $\mathbf{PA}(\boldsymbol{x}_i)$ denotes the sample matrix for variables in $\mathbf{PA}(X_i)$. The first term in the objective function, $\sum_{i=1}^{p}\left\{\left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)\left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)^T\right\}$, is a profile likelihood to measure the model fit. In the second term, $\|\boldsymbol{\beta}_i\|_1$ is the sum of the absolute values of the elements in $\boldsymbol{\beta}_i$ and thus is the so-called L1-norm penalty [37]. The regularization parameter, λ_1, controls the number of non-zero elements in the solution to $\boldsymbol{\beta}_i$, $\widehat{\boldsymbol{\beta}}_i$; the larger the λ_1, the fewer nonzero elements in $\widehat{\boldsymbol{\beta}}_i$. Because fewer nonzero elements in $\widehat{\boldsymbol{\beta}}_i$ correspond to fewer arcs in the learned BN structure, a larger λ_1 results in a sparser structure. In addition, the constraints are to assure that the learned BN is a DAG (see Lemma 1 and Theorem 1 below).

Solving the constrained optimization in (1) is difficult. Therefore, the penalty method [42] is employed to transform it into an unconstrained optimization problem, through adding an extra L1-norm penalty into the objective function, i.e.,

$$\widehat{\mathbf{B}}_{ap} = \min_{\mathbf{B}} \sum_{i=1}^{p} f_i(\boldsymbol{\beta}_i) = min_{\mathbf{B}} \sum_{i=1}^{p} \begin{Bmatrix} \left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)\left(\boldsymbol{x}_i - \boldsymbol{\beta}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)^T \\ +\lambda_1 \|\boldsymbol{\beta}_i\|_1 + \ \lambda_2 \sum_{j \in PA(X_i)} |\beta_{ji} \times \mathbf{P}_{ij}| \end{Bmatrix}, \quad (2)$$

where $j \Subset \mathbf{PA}(X_i)$ denotes that the variable indexed by j, i.e., X_j is a parent of X_i. Here, $\lambda_2 \sum_{j \in PA(X_i)} |\beta_{ji} \times \mathbf{P}_{ij}|$ is to push $\beta_{ji} \times \mathbf{P}_{ij}$ to become zero. Under some mild

conditions [42], there exists a λ_2^* such that for all $\lambda_2 \geq \lambda_2^*$, $\widehat{\mathbf{B}}_{ap}$ is also a minimizer for (1). Theorem 1 gives a practical estimation for λ_2^*.

Theorem 1. *Any* $\lambda_2 > (n-1)^2\, p/\lambda_1 - \lambda_1$ *will guarantee* $\widehat{\mathbf{B}}_{ap}$ *to be a DAG.*

Proof. To prove this, we first need to prove that, with a certain value of λ_1 and any value of λ_2, $\widehat{\mathbf{B}}_{\text{ap}}$ is bounded, i.e., $\lambda_1\|\widehat{\boldsymbol{\beta}}_i\|_1 \leq \left(\boldsymbol{x}_i - \widehat{\boldsymbol{\beta}}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\beta}}_i{}^T \mathbf{PA}(\boldsymbol{x}_i)\right)^T + \lambda_1\|\widehat{\boldsymbol{\beta}}_i\|_1 + \lambda_2 \sum_{j\in\mathbf{PA}(X_i)}|\hat{\beta}_{ji} \times \mathbf{P}_{ij}| \leq \boldsymbol{x}_i\boldsymbol{x}_i^T = n-1$, for each $\widehat{\boldsymbol{\beta}}_i$. The second inequality holds because $\boldsymbol{x}_i\boldsymbol{x}_i^T$ is the value of the left-hand side of the inequality when $\boldsymbol{\beta}_i = 0$, which is obviously larger than that when $\boldsymbol{\beta}_i = \widehat{\boldsymbol{\beta}}_i$. The last equality holds because we have standardized all the variables. Thus, we know that $\max_{k\in\mathbf{PA}(X_i)}|\hat{\beta}_{ki}| \leq (n-1)/\lambda_1$. Now, we use proof-by-contradiction to show that, with any $\lambda_2 > (n-1)^2\, p/\lambda_1 - \lambda_1$, we will get a DAG. Suppose that such a λ_2 doesn't guarantee a DAG. Then, there must be at least a pair of variables X_i and X_j with $\beta_{ji} \times \mathbf{P}_{ij} \neq 0$, which is $\beta_{ji} \neq 0$ and $\mathbf{P}_{ij} = 1$. Based on the first order optimality condition, $\beta_{ji} \neq 0$ i.f.f. $\left|\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\beta}}_{i/j}^T \mathbf{PA}_{/j}(\boldsymbol{x}_i)\right)\boldsymbol{x}_j^T\right| - \left(\lambda_1 + \lambda_2|\mathbf{P}_{ij}|\right) > 0$. Here, $\widehat{\boldsymbol{\beta}}_{i/j}^T$ denotes the elements in $\widehat{\boldsymbol{\beta}}_i$ without $\hat{\beta}_{ji}$ and $\mathbf{PA}_{/j}(\boldsymbol{x}_i)$ is defined similiarly. However, $\left|\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\beta}}_{i/j}^T \mathbf{PA}_{/j}(\boldsymbol{x}_i)\right)\boldsymbol{x}_j^T\right| \leq |\boldsymbol{x}_i\boldsymbol{x}_j^T| + \sum_{k\in\mathbf{PA}/j(\boldsymbol{x}_i)}|\hat{\beta}_{ki}\boldsymbol{x}_k\boldsymbol{x}_j^T| < (n-1)i_k \max_{k\in\mathbf{PA}(X_i)}\hat{\beta}_{ki} < (n-1)^2\, p/\lambda_1$, resulting in $\left|\left(\boldsymbol{x}_i - \widehat{\boldsymbol{\beta}}_{i/j}^T \mathbf{PA}_{/j}(\boldsymbol{x}_i)\right)\boldsymbol{x}_j^T\right| - \left(\lambda_1 + \lambda_2|\mathbf{P}_{ij}|\right) < 0$. □

Theorem 1 implies that if we specify any $\lambda_2 > (n-1)^2\, p/\lambda_1 - \lambda_1$, we will get a minimizer of (1) through solving (2). However, in practice, directly solving (2) by specifying a large λ_2 may converge slowly. This is because the unconstrained problem in (2) may be ill-conditioned with a too large value for λ_2 [42]. To avoid this situation, the "warm start" method [42] can be used, which works in the following way: first, it specifies a series of

values for λ_2, i.e., $\lambda_2^0 < \lambda_2^1 < \lambda_2^2 < \cdots < \lambda_2^M$, with a small λ_2^0 and $\lambda_2^M > (n-1)^2\, p/\lambda_1 - \lambda_1$; next, it optimizes (2) with $\lambda_2 = \lambda_2^0$ to get a minimizer $\widehat{\mathbf{B}}_{ap}^0$, using an arbitrary initial value; then, it optimizes (2) with $\lambda_2 = \lambda_2^1$, using $\widehat{\mathbf{B}}_{ap}^0$ as an initial value; this process iterates, until it optimizes (2) with $\lambda_2 = \lambda_2^M$. With the last minimizer as the initial value for the next optimization problem, this method can be quite efficient.

Given λ_1 and λ_2, the BCD algorithm [43] can be employed to solve (2). The BCD algorithm updates each $\boldsymbol{\beta_i}$ iteratively, assuming that all other parameters are fixed. In our situation, this is equivalent to optimizing $f_i(\boldsymbol{\beta_i})$ in (2) iteratively and the algorithm will terminate when some convergence conditions are satisfied. We remark that $f_i(\boldsymbol{\beta_i})$, after some transformation, is similar to LASSO [37], i.e.,

$$f_i(\boldsymbol{\beta_i}) = \left(\boldsymbol{x}_i - \boldsymbol{\beta_i}^T \mathbf{PA}(\boldsymbol{x_i})\right)\left(\boldsymbol{x}_i - \boldsymbol{\beta_i}^T \mathbf{PA}(\boldsymbol{x_i})\right)^T + \sum_{j \in \mathbf{PA}(X_i)} \left(\lambda_1 + \lambda_2 \left|\mathbf{P}_{ij}\right|\right)\left|\beta_{ji}\right|. \quad (3)$$

As a result, the shooting algorithm [44] for LASSO may be used to optimize $f_i(\boldsymbol{\beta_i})$ in each iteration. Note that at each iteration for optimizing $f_i(\boldsymbol{\beta_i})$, we also need to calculate $\mathbf{P}_{ij}$ for $j \Subset \mathbf{PA}(X_i)$. This can be done by a Breadth-first search on $\mathbf{G}$ with X_i being the root node [45]. A more detailed description of the BCD algorithm and the shooting algorithm used to solve (2) is given in Figs. 3-2 and 3-3, respectively.

Finally, we want to mention that the L2-norm penalty, $\lambda_2 \sum_{j \in PA(X_i)} \left(\beta_{ji} \times \mathbf{P}_{ij}\right)^2$, might also be used in (2). The advantage is that it is a differentiable function of β_{ji}. Also, as shown in [42], $\beta_{ji} \times \mathbf{P}_{ij} \to 0$ when $\lambda_2 \to \infty$. However, the weakness of the L2-norm penalty, compared with L1-norm penalty, is that there is no guarantee that a finite λ_2 exists to assure $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for all pairs of X_i and X_j.

Time complexity analysis: Each iteration of the BCD algorithm consists of two operations: a shooting algorithm and a Breadth-first search on $\mathbf{G}$. These two operations cost $O(pn)$ [46] and $O(p + |\mathbf{G}|)$, respectively. Here $|\mathbf{G}|$ is the number of nonzero

elements in $\mathbf{G}$. If $\mathbf{G}$ is sparse, i.e., $|\mathbf{G}| = Cp$ with a small constant C, then $O(p + |\mathbf{G}|) = O(p)$. Thus, the computational cost at each iteration is only $O(pn)$. Furthermore, each sweep through all columns of $\mathbf{B}$ costs $O(p^2 n)$. Our simulation study shows that it usually takes no more than 5 sweeps to converge.

3.4 Some theoretical analysis on the competitive advantage of the proposed SBN algorithm

Simulation studies in Section 3-5 will show that SBN is more accurate than various existing algorithms that employ a two-stage approach. This section aims to provide some theoretical insights about why the existing algorithms are less accurate. Please note that although a comprehensive analysis of this kind on all types of BNs and all two-stage algorithms is the most desirable, it is also very challenging, if not impossible, and beyond the scope of this paper. Therefore, in this section, we focus on some specific types of BNs and one popular two-stage algorithm, so as to provide some supporting evidences for the proposed SBN in addition to the results of the simulation studies in Section 3-5.

Input: sample matrix, **X**; number of variable, p; regularization parameters, $\{\lambda_i\}_{i=1,2}$; initial $\mathbf{B}^0$; stopping criterion, ϵ.

Initialize:

Let $converge = false$;

Let $t = 0$;

Repeat

For $i = 1,2, \dots, p$

A Breadth-first search on **G** with X_i being

the root node to calculate $\mathbf{P}_{ij}$ for $j = 1, \dots, p$.

Use the shooting algorithm in Fig. 3-3 to

Optimize $f_i(\boldsymbol{\beta}_i)$ and get $\boldsymbol{\beta}_i^{t+1}$;

End for

If $\left\|\mathbf{B}^{t+1} - \mathbf{B}^{t}\right\|_2 \leq \epsilon$ **then**

$converge = true$;

Else

$converge = false$;

Fig. 3-2. The BCD algorithm used for solving (2)

Input: sample vector $\boldsymbol{x}_i$; sample matrix $\mathbf{PA}(\boldsymbol{x}_i)$; regularization parameters, $\{\lambda_i\}_{i=1,2}$; initial $\boldsymbol{\beta}_i^0$; stopping criterion, ϵ.

Initialize:

Let converge = false;

Let $t = 0$;

Repeat

For $j = 1,2,\dots,p$

$$\beta_{ji}^{t+1} = \left(\left|\frac{\left(\boldsymbol{x}_i - \boldsymbol{\beta}_{i/j}^{t\,T}\mathbf{PA}_{/j}(\boldsymbol{x}_i)\right)\boldsymbol{x}_j^T}{\boldsymbol{x}_i\boldsymbol{x}_i^T}\right| - \frac{(\lambda_1+\lambda_2|\mathbf{P}_{ij}|)}{\boldsymbol{x}_i\boldsymbol{x}_i^T}\right)_+ sign\left(\frac{\left(\boldsymbol{x}_i - \boldsymbol{\beta}_{i/j}^{t\,T}\mathbf{PA}_{/j}(\boldsymbol{x}_i)\right)\boldsymbol{x}_j^T}{\boldsymbol{x}_i\boldsymbol{x}_i^T}\right)$$

;

End for

If $\left\|\boldsymbol{\beta}_i^{t+1} - \boldsymbol{\beta}_i^t\right\|_2 \le \epsilon$ **then**

$converge = true$;

Else

Fig. 3-3. The shooting algorithm used for solving (3)

Recall that Stage 1 of the two-stage approach is to identify the potential parents of each X_i. The existing algorithms achieve this goal by identifying the MB of X_i. A typical method is variable selection based on regressions, i.e., to build a regression of X_i on all other variables and consider the variables selected to be the MB. One difference between various algorithms is the type of regression used and the method used for variable selection. For example, the TC algorithm [40] uses ordinary regression and a t-test for variable selection; the L1MB-DAG algorithm [36] uses LASSO.

However, in the regression of X_i, not only will the coefficients for the variables not in the MB be small (theoretically zero due to the definition of MB), the coefficients for the parents may also be very small due to the correlation between the parents and the children. As a result, some parents may not be selected in the variable selection, i.e., they will be missed in Stage 1 of the two-stage approach, leading to greater BN learning errors.

In contrast, SBN may not suffer from this problem, because it is a one-stage approach that identifies the parents directly.

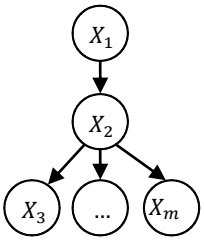


Fig. 3-4. A general tree

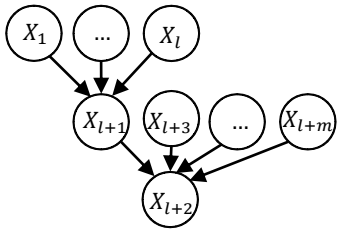


Fig. 3-5. A general inverse tree

To further illustrate this point, we analyze one two-stage algorithm, the TC algorithm. TC does variable selection using a t-test. To determine whether a variable should be selected, a t-test uses the statistic $\hat{\beta}/se(\hat{\beta})$, where $\hat{\beta}$ is the least-square estimate for the regression coefficient of this variable and $se(\hat{\beta})$ is the standard error. The larger the value of $\hat{\beta}/se(\hat{\beta})$, the higher the chance that the variable will be selected. Theorems 2 and 3 below show that even though the value of $\hat{\beta}/se(\hat{\beta})$ corresponding to a parent of X_i is large in the true BN, its value may decrease drastically in the regression of X_i on all other variables. Theorem 2 focuses on a specific type of BN, a general tree, in which all variables have one common ancestor and there is at most one directed path between two variables; Theorem 3 focuses on a general inverse tree, which becomes a general tree if reversing all the arcs. Proof of Theorem 2 can be found in Appendix A; proof of Theorem 3 can be found in the Supplemental Material.

Theorem 2. *Consider a general tree with* m *variables, whose structure and parameters are given by* $X_1 = e_1$, $X_2 = \beta_{12}X_1 + e_2$, $X_i = \beta_{2i}X_2 + e_i$, $i = 3,4,\dots,m$ *(Fig. 3-4). All the variables have unit variance. Let* $\hat{\beta}_{12}$ *denote the least-square estimate for* β_{12} *in regression* $X_2 = \beta_{12}X_1 + e_2$. *Let* $\hat{\beta}_{12}^{MB}$ *denote the least-square estimate for* β_{12}^{MB} *in regression* $X_2 = \beta_{12}^{MB}X_1 + \beta_{23}^{MB}X_3 + \cdots + \beta_{2m}^{MB}X_m + e_2^{MB}$ *(i.e., a regression that regresses* X_2 *on all other variables in the general tree).Then, the following relations*

hold:

$$\left|\hat{\beta}_{12}^{MB}\right| = \left|\hat{\beta}_{12}\right| \times \left|\frac{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)}{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=3}^{m}\left[\hat{\beta}_{2i}^2(1-\hat{\beta}_{12}^2)\prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right]}\right| < \left|\hat{\beta}_{12}\right|,$$

$$\left|\frac{\hat{\beta}_{12}^{MB}}{se\left(\hat{\beta}_{12}^{MB}\right)}\right| = \left|\frac{\hat{\beta}_{12}}{se\left(\hat{\beta}_{12}\right)}\right| \times \sqrt{\frac{\prod_{i=3}^{m}\left(1-\hat{\beta}_{2i}^2\right)}{\prod_{i=4}^{m}\left(1-\hat{\beta}_{2i}^2\right)+\sum_{i=4}^{m}\left[\hat{\beta}_{2i}^2\prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right]}}$$

$$< \left|\frac{\hat{\beta}_{12}}{se(\hat{\beta}_{12})}\right|$$

where $\hat{\beta}_{ij}$ denotes the least-square estimate for a regression coefficient β_{ij} and $se(\hat{\beta}_{ij})$ denotes the standard error for $\hat{\beta}_{ij}$.

Theorem 3. *Consider a general inverse tree with $m+l+2$ variables, whose structure and parameters are given by $X_k = e_k$, $k = 1,2,\dots,l,l+3,\dots,l+m$, $X_{l+1} = \sum_{k=1}^{l}\beta_{k,l+1}X_k + e_{l+1}$, $X_{l+2} = \beta_{l+1,l+2}X_{l+1} + \sum_{i=3}^{m}\beta_{l+i,l+2}X_{l+i} + e_{l+2}$ (Fig. 3-5). All the variables have unit variance. Let $\hat{\beta}_{k,l+1}$ denote the least-square estimate for $\beta_{k,l+1}$ in regression $X_{l+1} = \sum_{k=1}^{l}\beta_{k,l+1}X_k + e_{l+1}$, $k = 1,2,\dots,l$. Let $\hat{\beta}_{k,l+1}^{MB}$ denote the least-square estimate for $\beta_{k,l+1}^{MB}$ in regression $X_{l+1} = \sum_{k=1}^{l}\beta_{k,l+1}^{MB}X_k + \beta_{l+1,l+2}^{MB}X_{l+2} + \sum_{i=3}^{m}\beta_{l+i,l+2}^{MB}X_{l+i} + e_{l+1}^{MB}$ (i.e., a regression that regresses X_{l+1} on all other variables in the general inverse tree). Then, the following relations hold:*

$$\left|\hat{\beta}_{k,l+1}^{MB}\right| = \left|\hat{\beta}_{k,l+1}\right| \times \left|\frac{1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2}{1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2\sum_{i=1}^{l}\hat{\beta}_{i,l+1}^2}\right| < \left|\hat{\beta}_{k,l+1}\right|,$$

$$\left|\frac{\hat{\beta}_{k,l+1}^{MB}}{se(\hat{\beta}_{k,l+1}^{MB})}\right| = \left|\frac{\hat{\beta}_{k,l+1}}{se(\hat{\beta}_{k,l+1})}\right| \times \left|\frac{1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2}{1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2\sum_{i=1}^{l}\hat{\beta}_{i,l+1}^2}\right|$$

$$\times \sqrt{\frac{(1-\sum_{i=1}^{l}\hat{\beta}_{i,l+1}^2)(1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2\sum_{i=1}^{l}\hat{\beta}_{i,l+1}^2)}{(1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2\sum_{i=1,i\neq k}^{l}\hat{\beta}_{i,l+1}^2)}}$$

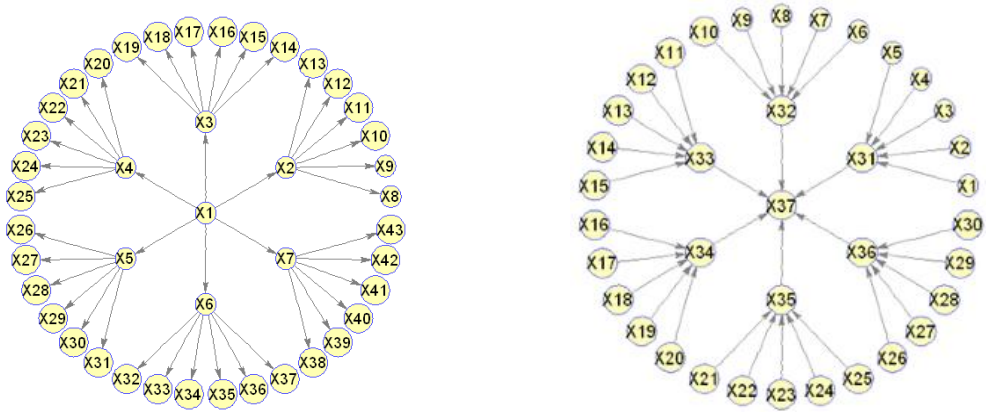
$$\times \sqrt{\frac{1-\sum_{i=1}^{l}\hat{\beta}_{i,l+1}^2(1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2)}{1-\sum_{i=1,i\neq k}^{l}\hat{\beta}_{i,l+1}^2(1-\sum_{i=1}^{m}\hat{\beta}_{l+2+i,l+2}^2-\hat{\beta}_{l+1,l+2}^2)}} < \left|\frac{\hat{\beta}_{k,l+1}}{se(\hat{\beta}_{k,l+1})}\right|.$$

Here we use two examples to illustrate the Theorems. Consider a general tree with m=8 (see Fig. 3-4 to recall the definition for m) and least-square estimates for the parameters being $\hat{\beta}_{12} = 0.3$ and $\hat{\beta}_{2i} = 0.8, i = 3, \dots, 8$. Then, using the formula for $\hat{\beta}_{12}^{MB}$ in Theorem 2, we can get $|\hat{\beta}_{12}^{MB}| = |\hat{\beta}_{12}| \times 0.093 < |\hat{\beta}_{12}|$. Using the formula for $\hat{\beta}_{12}^{MB}/se(\hat{\beta}_{12}^{MB})$, we can get $|\hat{\beta}_{12}^{MB}/se(\hat{\beta}_{12}^{MB})| = |\hat{\beta}_{12}/se(\hat{\beta}_{12})| \times 0.29 < |\hat{\beta}_{12}/se(\hat{\beta}_{12})|$. Consider a general inverse tree with $l = 5$ and $m = 0$ (see Fig. 3-5 to recall definitions for l and m) and least-square estimates for the parameters being $[\hat{\beta}_{16}, \cdots, \hat{\beta}_{56}] = [0.24, 0.325, 0.256, 0.304, 0.216]$ and $\hat{\beta}_{67} = 0.38$. Then, using the formula for $\hat{\beta}_{k,l+1}^{MB}$ (i.e., $\hat{\beta}_{k,6}^{MB}, k = 1, \dots, 5$) in Theorem 3, we can get $|\hat{\beta}_{16}^{MB}| = |\hat{\beta}_{16}| \times 0.15 < |\hat{\beta}_{16}|$, $|\hat{\beta}_{26}^{MB}| = |\hat{\beta}_{26}| \times 0.163 < |\hat{\beta}_{26}|$, $|\hat{\beta}_{36}^{MB}| = |\hat{\beta}_{36}| \times 0.48 < |\hat{\beta}_{36}|$, $|\hat{\beta}_{46}^{MB}| = |\hat{\beta}_{46}| \times 0.148 < |\hat{\beta}_{46}|$, and $|\hat{\beta}_{56}^{MB}| = |\hat{\beta}_{56}| \times 0.148 < |\hat{\beta}_{56}|$. Using the formula for $\hat{\beta}_{k,l+1}^{MB}/se(\hat{\beta}_{k,l+1}^{MB})$, we can verify $|\hat{\beta}_{k,l+1}^{MB}/se(\hat{\beta}_{k,l+1}^{MB})| < |\hat{\beta}_{k,l+1}/se(\hat{\beta}_{k,l+1})|$.

Note that the theoretical study in this section focuses on Stage 1 of the two-stage approach. It would also be interesting to analyze Stage 2, e.g., to find out the relative significance of the coefficients for variables in the MB and identify under what conditions the true parents may be missed. We plan to conduct such analysis in the future.

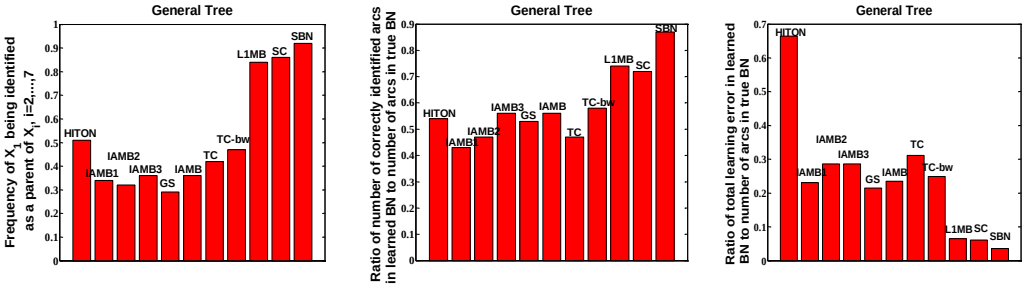
3.5 Simulation study on synthetic data

We perform five simulations. The first two show that, on a general tree and a general inverse tree, the existing algorithms based on the two-stage approach may miss some true parents with a high probability, while SBN performs well. The third simulation is to compare the structure learning accuracy of SBN with other competing algorithms using some benchmark networks. The fourth and fifth simulations are to investigate the scalability and efficiency of SBN and compare it with other competing algorithms. The code is available at http://www.public.asu.edu/~shuang31/codes/SBN.rar.



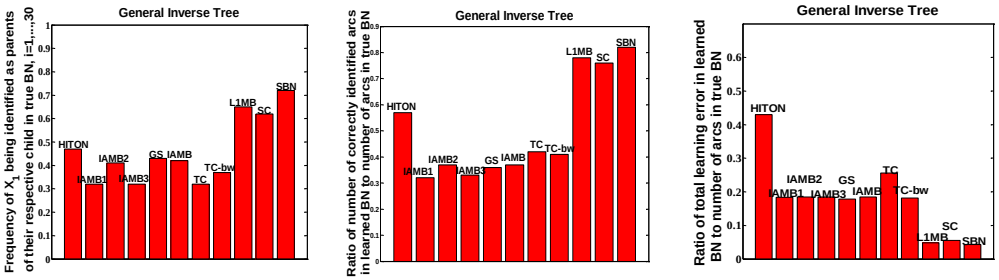
(a) (b)

Fig. 3-6. (a) General tree used in the simulation study in Section 5.1; (b) General inverse tree used in the simulation study in Section 3-5-2 (regression coefficients of arcs generated from $\pm Uniform(0.5,1)$)



(a) (b) (c)

Fig. 3-7. (a) Frequency of X_1 being identified as a parent of X_i, $i = 2, \cdots, 7$; (b) ratio of number of correctly identified arcs in learned BN to number of arcs in true BN; (c) ratio of total learning error in learned BN (false positives plus false negatives) to number of arcs in true BN



(a) (b) (c)

Fig. 3-8. (a) Frequency of X_i being identified as parents of their respective child in true BN, $i = 1, \cdots, 30$; (b) ratio of number of correctly identified arcs in learned BN to number of arcs in true BN; (c) ratio of total learning error in learned BN (false positives plus false negatives) to number of arcs in true BN

3.5.1 Learning accuracy for general tree

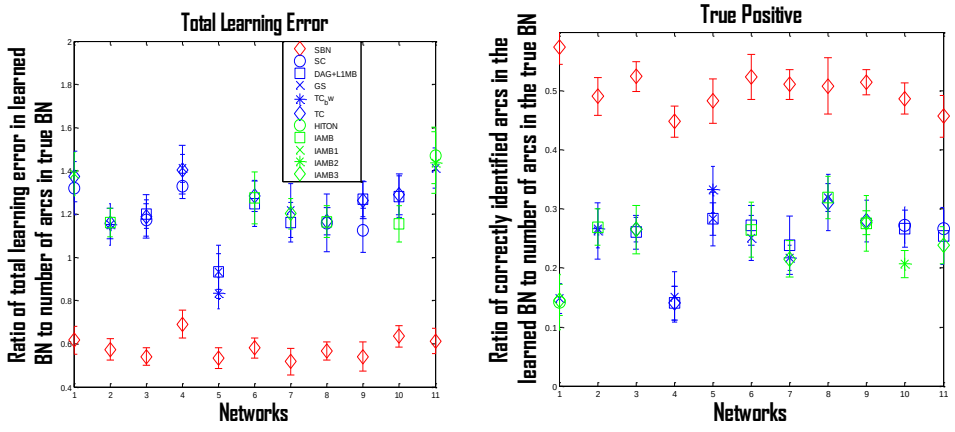
We select 10 existing algorithms in our study: HITON-PC [47], IAMB and three of its variants [48], GS [39], SC [35], TC and its advanced version TC-bw [40], and L1MB-DAG [36]. We focus on the general tree shown in Fig. 3-6 (a), in which the regression coefficient of each arc is randomly generated from $\pm Uniform(0.5,1)$. We simulate data from this general tree with a sample size of 200.

We apply the selected existing algorithms on the simulated data; the parameters of each algorithm are selected in the way that has been suggested in the respective paper. Specifically, HITON-PC is applied with a significance level of 5% used in the G^2 test of statistical independence and degrees of freedom set according to reference 14 cited in the paper of HITON-PC [47]. IAMB and its variants are applied with the significant level set to be 5%. GS is applied using the default value of 0.05 in its algorithm. SC is applied using the Bayesian scoring heuristic and the maximum number of parents chosen for the SC algorithm to be 5 and 10 (the one with better performance is kept and its corresponding result is presented). TC and TC-bw are applied by setting parameter $\alpha = 2/(p(p-1))$ as suggested and adopted in the paper [40]. There is no free parameter in L1MB-DAG.

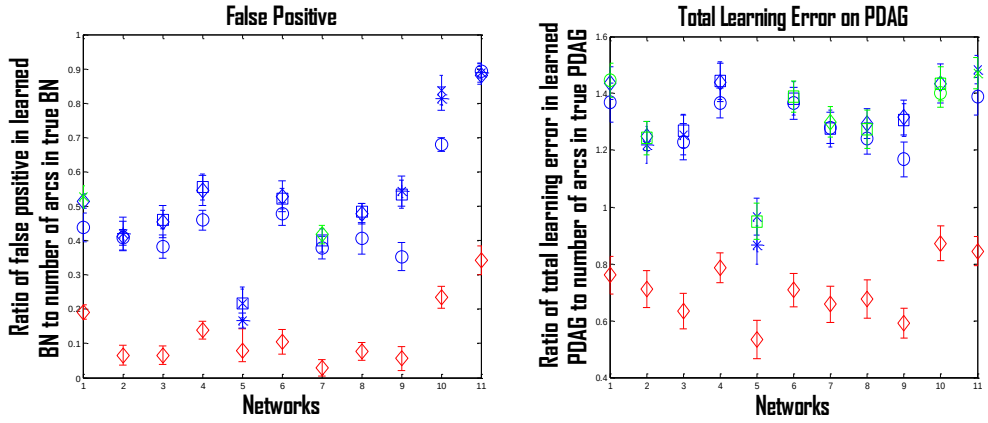
In applying the proposed SBN, λ_1 is selected by BIC (i.e., a step search is employed to find the λ_1 that produces the minimum BIC value). Following Theorem 1, λ_2 is set to be $10[(n-1)^2 p/\lambda_1 - \lambda_1]$ which empirically guarantees a DAG to be learned. The initial

value of SBN is the output of L1MB which uses LASSO in Stage 1 to identify the MB for each variable. We treat the identified MB by LIMB as parents and use the resulting BN as the initial value for SBN.

The results averaged over 100 repetitions are shown in Figs. 3-7 (a)–(c). The X-axis records the 10 selected algorithms and the proposed SBN (the last one). The Y-axis of each figure in (a)–(c) is a different performance measure, i.e., the frequency for X_1 being identified as a parent of X_i, $i = 2, \cdots, 7$, in (a), the ratio of the number of correctly identified arcs in the learned BN to the number of arcs in the true BN in (b), and the ratio of the total learning error in the learned BN (false positives plus false negatives) to the number or arcs in the true BN in (c). Note that Fig. 3-7(a) focuses on the arcs between X_1 and X_i, $i = 2, \cdots, 7$, in order to demonstrate Theorem 2 (i.e., because the MB of X_i includes not only parent X_1 but also six children, the coefficient of the arc between parent X_1 and X_i may be underestimated so that X_1 may not be included in the MB identified in Stage 1 of the competing algorithms). The observation from Fig. 3-7(a) is consistent with this theoretical explanation, which shows that the competing algorithms do not perform as well as SBN. Figs. 3-7(b) and (c) are performance measures defined on all arcs. They also show SBN's better performance.

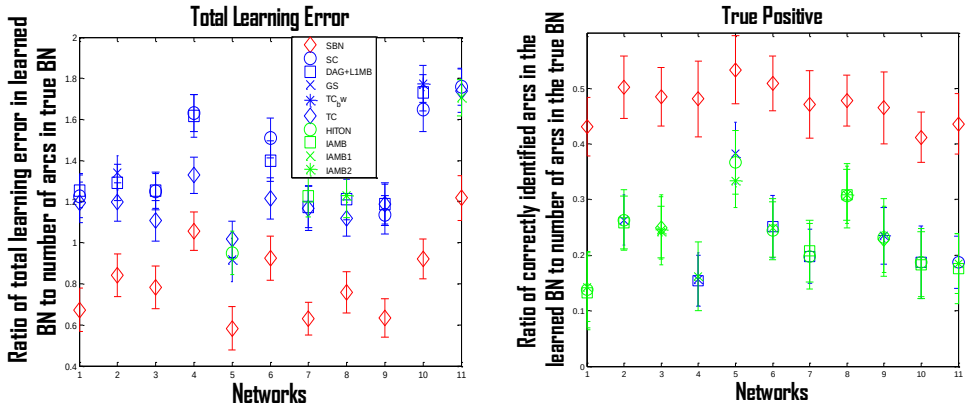


(a) (b)

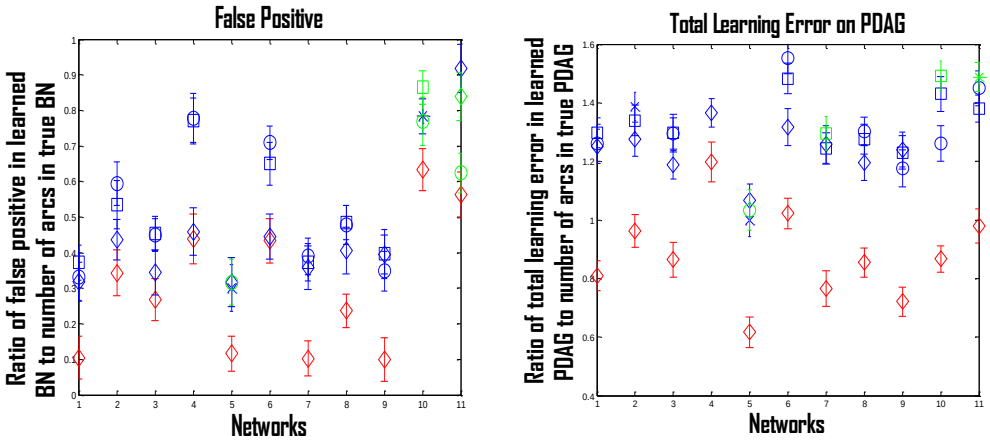


(c) (d)

Fig. 3-9: (a) Ratio of total learning error in the learned BN (false positives plus false negatives) to the number of arcs in the true BN, for the 10 competing algorithms and SBN, on 11 benchmark networks; (b) ratio of the correctly identified arcs in the learned BN to the number of arcs in the true BN; (c) ratio of the false positive in the learned BN to the number of arcs in the true BN. (d) ratio of the total learning error in the learned PDAG to the number of arcs in the true PDAG. The learned BN and PDAG in (a) – (d) are based on a simulation dataset of sample size 1000. Error bars represent three standard derivations.



(a) (b)



(c) (d)

Fig. 3-10: (a) Ratio of total learning error in the learned BN (false positives plus false negatives) to the number of arcs in the true BN, for the 10 competing algorithms and SBN, on 11 benchmark networks; (b) ratio of the correctly identified arcs in the learned BN to the number of arcs in the true BN; (c) ratio of the false positive in the learned BN to the number of arcs in the true BN. (d) ratio of the total learning error in the learned PDAG to the number of arcs in the true PDAG. The learned BN and PDAG in (a) – (d) are based on a simulation dataset of sample size 100. Error bars represent three standard derivations.

3.5.2 Learning accuracy for general inverse tree

We focus on the general inverse tree in Fig. 3-6 (b), in which the regression coefficient of each arc is randomly generated from $\pm Uniform(0.5,1)$. We simulate data from this general tree with a sample size of 200.

We apply the 10 selected existing algorithms and SBN on the simulated data in the same way as that in Section 3-5-1. The results of 100 repetitions are shown in Figs. 3-8 (a)–(c), which can be read in a similar way to Fig. 3-7. Note that Fig. 3-8(a) focuses on the arcs between X_i, $i = 1, \cdots, 30$, and their respective children, in order to demonstrate Theorem 3. Figs. 3-8(a)–(c) show that SBN performs better.

3.5.3 Learning accuracy for benchmark networks

We select 7 moderately large networks from the Bayesian Network Repository (BNR) [49]. These networks are selected based on the consideration that they provide a range of small-to-moderately-large networks with the number of nodes ranging from 7 to 61, they are sparse, and they were also used in [36], which is a competing algorithm of ours. We also use the tiling technique [50] to produce two large BNs, Alarm2 and Hailfinder2. Two other networks with specific structures, Factor and Chain [51], are also considered. The numbers of nodes and arcs in each of the 11 networks are shown in Table 3-1.

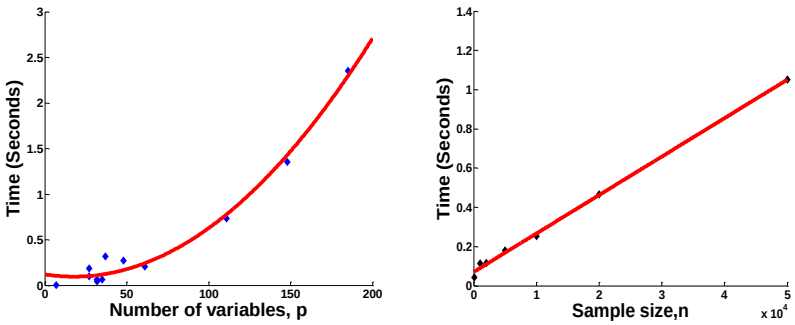
TABLE 3-1
BENCHMARK NETWORKS

	Networks	Number of nodes	Number of arcs
1	Factor	27	68
2	Alarm (BNR)	37	46
3	Barley (BNR)	48	84
4	Carpo (BNR)	61	74
5	Chain	7	6
6	Hailfinder (BNR)	56	66
7	Insurance (BNR)	27	52
8	Mildew (BNR)	35	46
9	Water (BNR)	32	66
10	Alarm 2	296	410
11	Haifinder 2	280	390

To specify the parameters of a network, i.e., to specify the regression coefficients of each variable on its parents, we randomly sample from $\pm Uniform(0.5, 1)$. Then, we simulate data for each network with a sample size 1000, and apply the 10 competing algorithms and SBN to learn the BN structure. The results over 100 repetitions are shown in Fig. 3-9(a), in which the X-axis records the 11 networks and the Y-axis records the ratio of the total learning error in the learned BN (false positives plus false negatives) to the number of arcs in the true BN. This figure deserves more explanation: we found it hard to show

all 10 competing algorithms, i.e., they become indistinguishable. Thus, for each benchmark network (i.e., a tick on the X axis), we only show the three competing algorithms with the best performance. For example, for network "Carpo" (4^{th} tick on the X axis) in Fig. 3-9(a), the top three competing algorithms shown are GS, TC, and SC. Figs. 3-9 (b)-(d) are comparison plots in terms of other criteria. Specifically, Fig. 3-9 (b) plots the ratio of the correctly identified arcs in the learned BN to the number of arcs in the true BN. Fig. 3-9 (c) plots the ratio of the falsely identified arcs in the learned BN to the number of arcs in the true BN. Fig. 3-9 (d) is similar to (a) but for PDAG (partially directed acyclic graph). Given a BN (a learned one or true one), the corresponding PDAG can be obtained by the method proposed in [13]. A PDAG is a collection of statistically equivalent BN structures, i.e., these structures all represent the same set of dependent and independent relationships so they are statistically indistinguishable. The PDAG of a BN can be constructed by replacing a directed arc between X_i and X_j in the BN with an undirected one, if some statistically equivalent BN structures have $X_i \rightarrow X_j$ and others have $X_i \leftarrow X_j$. A PDAG is very useful when making a causal interpretation, i.e., we may interpret the directed arcs in the PDAG as representing the direction of direct causal influence. Figs. 3-9 (a)–(d) show that SBN performs much better than all the competing algorithms in BN- and PDAG-identification.

Furthermore, we would like to compare SBN with the competing algorithms under small sample sizes. We decrease the sample size to 100 and repeat the above procedure. The results are shown in Figs. 3-10 (a)–(d). It can be seen that SBN still performs much better than all the competing algorithms in BN- and PDAG-identification even for small sample sizes.



(a) (b)

Fig. 3-11. Scalability of SBN with respect to (a) the number of variables, p; (b) the sample size, n.

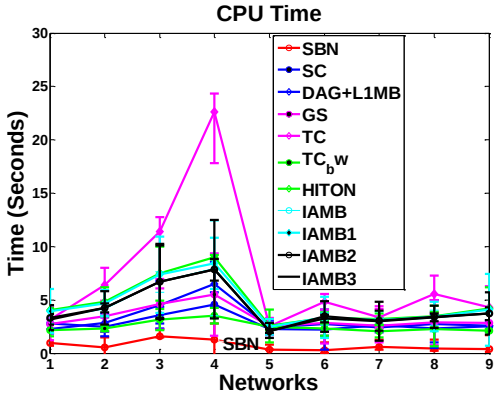


Fig. 3-12. Comparison of SBN with competing algorithms on CPU time in structure learning. Y-axis is the CPU time for each sweep through all the columns of **B**, on a computer with Intel Core 2, 2.2 G Hz, 4G memory. X-axis is the first nine networks in Table 3-1.

TABLE 3-2
COMPARISON OF SBN WITH COMPETING ALGORITHMS ON THE CPU TIME IN STRUCTURE LEARNING OF TWO LARGE NETWORKS (STANDARD DERIVATION IS SHOWN IN THE BRACKET)

Algorithms	Alarm 2	Haifinder 2
SBN	***67.1 (13.4)***	***78.8 (19.5)***
SC	958 (73.6)	987 (83.2)
L1MB-DAG	11715 (1034.8)	13521 (2543.3)
GS	1071 (142.4)	1204 (98.5)
TC-bw	35981 (2578.3)	41214 (5435.3)
TC	445 (89.3)	496 (67.9)
HITON	10324 (3390.7)	13913 (2482.1)
IAMB	6423 (894.1)	8060 (1427.4)
IAMB1	6416 (987.6)	8148 (1075.6)
IAMB2	6411 (1293.2)	7994 (919.1)

IAMB3	6415 (1508)	7998 (1793.7)

3.5.4 Scalability

We study two aspects of scalability for SBN: the scalability with respect to the number of variables in a BN, p, and the scalability with respect to the sample size, n. We use the CPU time for each sweep through all the columns of **B** as the parameter for measurement. Specifically, we fix n =1000, and vary p by using the 11 benchmark networks.. Also, we fix p=37 (the Alarm network). The results over 100 repetitions are shown in Fig. 3-11 (a) and (b), respectively. It can be seen that the times are linear in n and quadratic in p, which confirms our theoretical time complexity analysis in Section 3-3.

3.5.5 Efficiency

We further compare the complete CPU time of SBN with other competing algorithms, in structure learning of the 11 benchmark networks. The results of 100 repetitions are shown in Table 3-2 (the two large networks, Alarm 2 and Haifinder 2) and Fig. 3-12 (the other networks). It can be seen that SBN is the fastest algorithm in structure learning of all the benchmark networks. This is expected since the fastest algorithms among the 10 competing algorithms, i.e., GS and TC, have a time complexity $O(p^3n)$, while SBN only costs $O(p^2n)$ (i.e., each sweep of SBN costs $O(p^2n)$ and our simulation study shows that SBN usually takes no more than 5 sweeps to converge). Note that in many applications, there is no prior knowledge which can be used to identify a good initial value for B, in which cases SBN usually need a good initialization learnt from other algorithms, such as L1MB or SC. Thus, in these applications, it is reasonable to consider the computational complexity of SBN as a sum of the computational complexity of both SBN and L1MB.

3.6 Brain connectivity modeling by SBN

FDG-PET images of 49 AD and 67 matching normal control (NC) subjects are downloaded from the Alzheimer's Disease Neuroimaing Initiative website ((www.loni.ucla.edu/ADNI). Demographic information and MMSE scores of the subjects are given in Table 3-3.

We apply Automated Anatomical Labeling [52] to segment each image into 116 anatomical volumes of interest (AVOIs) and then select 42 AVOIs that are considered to be potentially relevant to AD based on the literature. Each AVOI becomes a region/variable/node in SBN. Please see Table 3-4 for the name of each AVOI brain region. These regions distributed in the four lobes of the brain, i.e., the frontal, parietal, occipital, and temporal lobes. The measurement data of each region, according to the mechanism of FDG-PET, is the regional average FDG binding counts, representing the degree of glucose metabolism.

We apply SBN to learn a BN for AD and another one for NC, to represent their respective brain connectivity models. Note that because BNs are directed graphical models, a connectivity model learned by SBN reveals the directional effects of one brain region over another—called the effective connectivity of the brain [59]. Effective connectivity has been much less studied in the AD literature, while most existing work focuses on the functional connectivity, i.e., the correlations among brain regions. Studies on effective connectivity can greatly complement the existing functional connectivity studies by providing insight into how the correlations are mediated, which may further lead to an understanding of the mechanism underlying the communication among distinct brain regions. In this sense, SBN has the advantage over undirected graphical models of discovering new knowledge about AD.

TABLE 3-3

Demographic Information and MMSE

	NC	AD	*P*-VALUE
Age (mean ± SD)	76.0±4.69	75.3±6.85	0.53
Gender (Male/Female)	43/24	27/22	0.77
Years of education (mean ± SD)	15.9±3.24	14.7±3.02	0.01
Baseline MMSE	29.0±1.18	23.6±1.93	<0.001

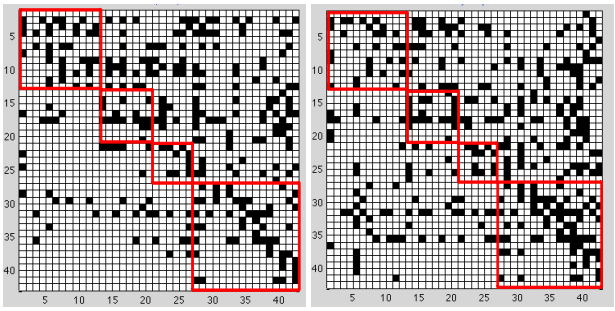
TABLE 3-4

NAMES OF THE AVOI FOR BRAIN CONNECTIVITY MODELING (L = LEFT HEMISPHERE, R=RIGHT HEMISPHERE)

Frontal lobe		Parietal lobe		Occipital lobe		Temporal lobe	
1	Frontal_Sup_L	13	Parietal_Sup_L	21	Occipital_Sup_L	27	Temporal_Sup_L
2	Frontal_Sup_R	14	Parietal_Sup_R	22	Occipital_Sup_R	28	Temporal_Sup_R
3	Frontal_Mid_L	15	Parietal_Inf_L	23	Occipital_Mid_L	29	Temporal_Pole_Sup_L
4	Frontal_Mid_R	16	Parietal_Inf_R	24	Occipital_Mid_R	30	Temporal_Pole_Sup_R
5	Frontal_Sup_Medial_L	17	Precuneus_L	25	Occipital_Inf_L	31	Temporal_Mid_L
6	Frontal_Sup_Medial_R	18	Precuneus_R	26	Occipital_Inf_R	32	Temporal_Mid_R
7	Frontal_Mid_Orb_L	19	Cingulum_Post_L			33	Temporal_Pole_Mid_L
8	Frontal_Mid_Orb_R	20	Cingulum_Post_R			34	Temporal_Pole_Mid_R
9	Rectus_L					35	Temporal_Inf_L 8301
10	Rectus_R					36	Temporal_Inf_R 8302
11	Cingulum_Ant_L					37	Fusiform_L
12	Cingulum_Ant_R					38	Fusiform_R
						39	Hippocampus_L
						40	Hippocampus_R
						41	ParaHippocampal_L
						42	ParaHippocampal_R

In the learning of an AD (or NC) effective connectivity model, the value for λ_1 needs to be selected. In this paper, we adopt two criteria in selecting λ_1: one is to minimize the prediction error of the model and the other is to minimize the BIC. Both criteria have been popularly adopted in sparse learning [20], [21], [22], [37]. The two criteria lead to similar findings from the effective connectivity models, so only the results based on the minimum prediction error are shown in this section and the results based on BIC are included in Supplemental Material. For a given λ_1 value, the prediction error of the corresponding BN is computed as follows: First, a regression is fit for each node using the parents as predictors, and the regression coefficients are estimated by MLE. Then, the mean square error between the true and predicted values of each node is computed based on leave-one-out cross validation. Finally, the mean square errors of all the nodes are

summed to represent the prediction error of the BN. The λ_1 value that leads to the minimum prediction error is selected; with this λ_1, SBN is applied to learn a BN brain connectivity model. Fig. 3-13 shows the connectivity models for AD and NC. Each model is represented by a "matrix." Each row/column is one AVOI, X_j. A black cell at the i-th row and j-th column of the matrix represents that X_i is a parent of X_j. On each matrix, four red cubes are used to highlight the four lobes, i.e., the frontal, parietal, occipital, and temporal lobes, from top-left to bottom-right. The black cells inside each red cube reflect intra-lobe effective connectivity, whereas the black cells outside the cubes reflect inter-lobe effective connectivity.



(a) (b)

Fig. 3-13. Brain effective connectivity models by SBN. (a) AD; (b) NC.

The following interesting observations can be drawn from the connectivity models:

Global-scale effective connectivity:

The total number of arcs in a BN connectivity model— equal to the number of black cells in a matrix plot in Fig. 3-13—represents the amount of effective connectivity (i.e., the amount of directional information flow) in the whole brain. This number is 285 and 329 for AD and NC, respectively. In other words, AD has 13.4% less amount of effective connectivity than NC. Loss of connectivity in AD has been widely reported in the literature [60], [68], [69], [70].

Intra-/inter-lobe effective connectivity distribution:

Aside from having different amounts of effective connectivity at the global scale, AD may also have a different distribution pattern of connectivity across the brain from NC. Therefore, we count the number of arcs in each of the four lobes and between each pair of lobes in the AD and NC effective connectivity models. The results are summarized in Table 3-5. It can be seen that the temporal lobe of AD has 22.9% less amount of effective connectivity than NC. The decrease in connectivity in the temporal lobe of AD has been extensively reported in the literature [53], [54], [55]. The interpretation may be that AD is featured by dramatic cognitive decline and the temporal lobe is responsible for delivering memory and other cognitive functions. As a result, the temporal lobe is affected early and severely by AD, and the connectivity network in this lobe is severely disrupted. On the other hand, the frontal lobe of AD has 27.6% more amount of connectivity than NC. This observation has been interpreted as compensatory reallocation or recruitment of cognitive resources [56], [53], [57]. Because the regions in the frontal lobe are typically affected later in the course of AD (our data uses mild to moderate AD), the increased connectivity in the frontal lobe may help preserve some cognitive functions in AD patients. In addition, AD shows a decrease in the amount of connectivity in the parietal lobe, which has also been reported to be affected by AD. There is no significant difference between AD and NC in the occipital lobe. This observation is reasonable because the occipital lobe is primarily involved in the brain's visual function, which is not affected by AD.

TABLE 3-5

INTRA – AND INTER- LOBE EFFECTIVE CONNECTIVITY AMOUNTS

(A) AD

	Frontal	Parietal	Occipital	Temporal
Frontal	37	28	18	43
Parietal		16	14	42
Occipital			10	23
Temporal				54

(B) NC

	Frontal	Parietal	Occipital	Temporal
Frontal	29	32	12	61
Parietal		20	16	42
Occipital			11	36
Temporal				70

In addition to generating the connectivity models of AD and NC based on the minimum prediction error and minimum BIC criteria, we also generate the connectivity models by making the total numbers of arcs the same for AD and NC. We choose to do this to factor out the connectivity difference between AD and NC that is due to the difference at the global scale so that the remaining difference will reflect their difference in connectivity distribution. Specifically, the connectivity models with the total number of arcs equal to 120, 80, and 60 are generated (see Supplemental Material), which show similar intra- and inter-lobe effective connectivity distribution patterns to those discussed previously.

Direction of local effective connectivity:

As mentioned previously, one advantage of BNs over undirected graphical models in brain connectivity modeling is that the directed arcs in a BN reflect the directional effect of one region over another, i.e., the effective connectivity. Specifically, if there is a directed arc from brain regions X_i to X_j, it indicates that X_i takes a dominant role in the communication with X_j. The connectivity modes in Fig. 3-13 reveal a number of interesting findings in this regard:

(i) There are substantially fewer black cells in the area defined by rows 27–42 and columns 1–26 in AD than NC. Recall that rows 27–42 correspond to regions in the temporal lobe. Thus, this pattern indicates a substantial reduction in arcs pointing from temporal regions to the other regions in the AD brain, i.e., temporal regions lose their dominating roles in communicating information with the other regions as a result of AD. The loss is the most severe in the communication from temporal to frontal regions.

(ii) Rows 31 and 35, corresponding to brain regions "Temporal_Mid_L" and "Temporal_Inf_L", respectively, are among the rows with the largest number black cells in NC, i.e., these two regions take a significantly dominant role in communicating with

other regions in normal brains. However, the dominancy of the two regions is substantially reduced by 34.8% and 36.8%, respectively, in AD. A possible interpretation is that these are neocortical regions associated with amyloid deposition and early FDG hypometabolism in AD [60], [61], [62], [63], [64], [65].

(iii) Columns 39 and 40 correspond to regions "Hippocampus_L" and "Hippocampus_R," respectively. There are a total of 33 black cells in these two columns in NC, i.e., 33 other regions dominantly communicate information with the hippocampus. However, this number reduces to 22 (33.3% reduction) in AD. The reduction is more severe in Hippocampus_L—actually a 50% reduction. The hippocampus is well known to play a prominent role in making new memories and recalling. It has been widely reported that the hippocampus is affected early in the course of AD, leading to memory loss—the most common symptom of AD.

(iv) There are a total of 93 arcs pointing from the left to the right hemispheres of the brain in NC; this number reduces to 71 (23.7% reduction) in AD. The number of arcs from the right to the left hemispheres in AD is close to that in NC. This provides evidence that AD may be associated with inter-hemispheric disconnection and the disconnection is mostly unilateral, which has also been reported by some other papers [66], [67].

Note that all the above findings also hold for the PDAGs that are derived from the DAGs in Fig. 3-13. Please see Supplemental Material for the PDAGs.

3.7 Conclusion

In this paper, we proposed a BN structure learning algorithm, SBN, for learning large-scale BN structures from high-dimensional data. SBN adopted a novel formulation that involves one L1-norm penalty term to impose sparsity on the learning and another penalty to ensure the learned BN to be a DAG. We studied the theoretical property of the

formulation and identified a finite value for the regularization parameter of the second penalty; this value ensures that the learned BN is a DAG. Under this formulation, we further proposed use of the BCD and shooting algorithms to estimate the BN structure.

Our theoretical analysis on the time complexity of SBN showed that it is linear in the sample size and quadratic in the number of variables. This makes SBN more scalable and efficient than most existing algorithms, and thus makes it well suited for large-scale BN structure learning from high-dimensional datasets. In addition, we performed theoretical analysis on the competitive advantage of SBN over the existing algorithms in terms of learning accuracy. Our analysis showed that the existing algorithms employ a two-stage approach in BN structure identification, and thus having a high risk of misidentifying parents of each variable, whereas SBN does not suffer from this problem.

Our experiments on 11 moderate to large benchmark networks showed that SBN outperforms 10 competing algorithms in all metrics defined for measuring the learning accuracy and under various sample sizes. Also, SBN outperforms the 10 competing algorithms in scalability and efficiency.

We applied SBN to identify the effective brain connectivity model of AD from neuroimaging PDG-PET data. Compared with a brain connectivity model of NC, we found that AD had significantly reduced amounts of effective connectivity in key pathological regions. This is consistent with known pathology and the clinical progression in AD. Clinically, our findings may be useful for monitoring disease progress, evaluating treatment effects (both symptomatic and disease modifying), and enabling early detection of network disconnection in prodromal AD.

In future work, we will investigate how to measure statistical significance of the DAG identified by our algorithm. Potential methods include bootstrap [71], permutation tests [72], and stability selection [73]. This study is also important from the medical point of

view as it will help verify the significance of the identified brain connectivity loss based on the DAG. Also, although this paper focuses on structure learning of Gaussian BNs, the same formulation may be adopted for discrete BNs, which will be interesting to explore. In addition, we will investigate the behavior of SBN on Markov equivalent class. Our empirical observation has shown that the objective function of SBN is not Markov equivalent, i.e., SBN attributes different scores to BNs that are Markov equivalent. More in-depth theoretical analysis will be performed in future research.

Appendix

Based on Wright's second decomposition rule [58], the sample covariance matrix of all the variables, denoted by T_m, can be represented as a function of the least-square estimates for the parameters of the BN, $\hat{\beta}_{12}, \hat{\beta}_{23}, \hat{\beta}_{24}, \cdots, \hat{\beta}_{2m}$, i.e.,

$$T_m = \begin{bmatrix} 1 & \hat{\beta}_{12} & \hat{\beta}_{23} & \hat{\beta}_{24} & \cdots & \hat{\beta}_{2m} \\ \hat{\beta}_{12} & 1 & \hat{\beta}_{12}\hat{\beta}_{23} & \hat{\beta}_{12}\hat{\beta}_{24} & \cdots & \hat{\beta}_{12}\hat{\beta}_{2m} \\ \hat{\beta}_{23} & \hat{\beta}_{12}\hat{\beta}_{23} & 1 & \hat{\beta}_{23}\hat{\beta}_{24} & \cdots & \hat{\beta}_{2m}\hat{\beta}_{23} \\ \hat{\beta}_{24} & \hat{\beta}_{12}\hat{\beta}_{24} & \hat{\beta}_{23}\hat{\beta}_{24} & 1 & \cdots & \hat{\beta}_{2m}\hat{\beta}_{24} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{\beta}_{2m} & \hat{\beta}_{12}\hat{\beta}_{2m} & \hat{\beta}_{2m}\hat{\beta}_{23} & \hat{\beta}_{24}\hat{\beta}_{2m} & \cdots & 1 \end{bmatrix} \quad \text{(A-1)}$$

Now, consider the regression of X_2 on all other variables, i.e., $X_2 = \beta_{12}^{MB} X_1 + \beta_{23}^{MB} X_3 + \cdots + \beta_{2m}^{MB} X_m + e_2^{MB}$. According to the least square criterion, the regression coefficients,

$$\left[\hat{\beta}_{12}^{MB}, \hat{\beta}_{23}^{MB}, \cdots, \hat{\beta}_{2m}^{MB}\right]^T = C_{m-1}^{-1}\left[\hat{\beta}_{12}, \hat{\beta}_{23}, \cdots, \hat{\beta}_{2m}\right]^T, \quad \text{(A-2)}$$

where C_{m-1} is a sub-matrix of T_m by deleting the 1st column and 1st row from T_m. Denote C_{m-1}^{-1} by $A_{m-1} = (a_{ij})$. Then,

$$\hat{\beta}_{12}^{MB} = a_{11}\hat{\beta}_{12} + a_{12}\hat{\beta}_{23} + \cdots + a_{1m-1}\hat{\beta}_{2m}. \quad \text{(A-3)}$$

Our final objective is to express $\hat{\beta}_{12}^{MB}$ by the parameters of the BN. This can be achieved if we can express $a_{11}, a_{12}, \cdots, a_{1\,m-1}$ by the parameters of the BN, which is the goal of the following derivation.

It is known that

$$a_{1j} = (-1)^{1+j} \frac{\det(C_{1j,m-2})}{\det(C_{m-1})}, \tag{A-4}$$

where$C_{1j,m-2}$ is a matrix by deleting the 1st row and the jth column from C_{m-1}. So, the problem becomes calculation of $\det(C_{m-1})$ and $\det(C_{1j,m-2})$.

(i) Calculation of $\det(C_{m-1})$:

We first show the result:

$$\det(C_{m-1}) = \prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2) + \sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2j}^2)\right) \tag{A-5}$$

Note that $\prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2j}^2) = 1$ while $m = 3$ and $i = 3$.

Next, we will use the induction method in 1)-2) below to prove (A-5):

1) When $m = 3$, it is easy to see that (A-5) holds.

2) Assume that (A-5) holds for $m-2$, i.e.,

$$\det(C_{m-2}) = \prod_{i=3}^{m-1}(1-\hat{\beta}_{2i}^2) + \sum_{i=3}^{m-1}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m-1}(1-\hat{\beta}_{2j}^2)\right).$$

Then we need prove that (A-5) holds for $m-1$. Based on the definition of determinant,

$$\det(C_{m-1}) = \sum_{j=1}^{m-1}(-1)^{1+j} c_{1j} \det(C_{1j,m-2}), \tag{A-6}$$

Where c_{1j} is the entry at the 1st row and jth column of C_{m-1}. Now we need to derive $\det(C_{1j,m-2})$(only results are shown below due to page limits):

When $j = 1$, $\det(C_{1j,m-2}) = \prod_{i=4}^{m}(1-\hat{\beta}_{2i}^2) + \sum_{i=4}^{m}\left(\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2i}^2)\right)$. (A-7)

When $j \neq 1$, $\det(C_{1j,m-2}) = (-1)^{1+j+1}\hat{\beta}_{12}\hat{\beta}_{2j+2}\prod_{k=3,k\neq j}^{m}(1-\hat{\beta}_{2k}^2)$. (A-8)

Then, insert (A-7), (A-8), and $c_{11} = 1$, $c_{12} = \hat{\beta}_{12}\hat{\beta}_{23}$, …,$c_{1m-1} = \hat{\beta}_{12}\hat{\beta}_{2m}$ into (A-6):

$\det(C_{m-1}) =$

$\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2) + \sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2j}^2)\right) - \hat{\beta}_{12}^2 \sum_{i=3}^{m}\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2j}^2) = \prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2) + \sum_{i=3}^{m}\left[(1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}(1-\hat{\beta}_{2j}^2)\right]$.

This completes the proof for (A-5).

(ii) Calculation of $\det(C_{1j,m-2})$: $\det(C_{1j,m-2})$ has been obtained by (A-7) and (A-8).

Inserting (A-5), (A-7), and (A-8) into (A-4), we get:

$$a_{11} = \frac{\prod_{i=4}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=4}^{m}\left(\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)}{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)},$$

Furthermore,$a_{11} + a_{12}\hat{\beta}_{12}\hat{\beta}_{23} + a_{13}\hat{\beta}_{12}\hat{\beta}_{24} + \cdots + a_{1m-1}\hat{\beta}_{12}\hat{\beta}_{2m} = 1$. Inserting this into (A-3), we get$\hat{\beta}_{12}^{MB} = a_{11}\left(1-\left(1-\hat{\beta}_{12}^2\right)\right)/\hat{\beta}_{12}$. Plugging in the a_{11} above, we can get:

$$\hat{\beta}_{12}^{MB} = \hat{\beta}_{12}\frac{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)}{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)}, \qquad \text{(A-9)}$$

Obviously, the faction at the right-hand side is between 0 and 1. Therefore,$\left|\hat{\beta}_{12}^{MB}\right| < \left|\hat{\beta}_{12}\right|$.

Next we derive the formula for $\hat{\beta}_{12}^{MB}/se(\hat{\beta}_{12}^{MB})$. It is known that $se^2\left(\hat{\beta}_{12}^{MB}\right) = a_{11}/[(n-1)(T_m^{-1})_{11}]$. Since $(T_m^{-1})_{11} = \det(C_{m-1})/\det(T_m) = \det(C_{m-1})/\left[\left(1-\hat{\beta}_{12}^2\right)\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)\right]$and $\det(C_{m-1})$ is given in (A-5), we can get:

$$se^2\left(\hat{\beta}_{12}^{MB}\right) =$$

$$\frac{1}{n-1}\frac{(1-\hat{\beta}_{12}^2)\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)}{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)} \times \frac{\prod_{i=4}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=4}^{m}\left(\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)}{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=3}^{m}\left((1-\hat{\beta}_{12}^2)\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)}$$

(A-10)

Also, $se^2\left(\hat{\beta}_{12}\right) = \left(1-\hat{\beta}_{12}^2\right)/(n-1)$. Putting this together with (A-9) and (A-10), we can get:

$$\frac{\hat{\beta}_{12}^{MB}}{se(\hat{\beta}_{12}^{MB})} = \frac{\hat{\beta}_{12}}{se(\hat{\beta}_{12})}\sqrt{\frac{\prod_{i=3}^{m}(1-\hat{\beta}_{2i}^2)}{\prod_{i=4}^{m}(1-\hat{\beta}_{2i}^2)+\sum_{i=4}^{m}\left(\hat{\beta}_{2i}^2 \prod_{j=3,j\neq i}^{m}\left(1-\hat{\beta}_{2j}^2\right)\right)}},$$

It is obvious that the part under the root is less than one. Therefore, $\left|\hat{\beta}_{12}^{MB}/se(\hat{\beta}_{12}^{MB})\right| < \left|\hat{\beta}_{12}/se(\hat{\beta}_{12})\right|$.

Reference

[1] Friedman, N., Linial, M., Nachman, I. and Pe'er, D. Using Bayesian networks to analyze expression data. Journal of Computational Biology, 7, 601–620, 2000.

[2] Rodin, A. S. and Boerwinkle, E. Mining genetic epidemiology data with Bayesian networks I: Bayesian networks and example application (plasma apoE levels). Bioinformatics, 21(15), 3273–3278, 2005.

[3] Marcot, B.G., Holthausen, R.S., Raphael, M.G., Rowland, M. and Wisdom,M. Using Bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. Forest Ecology and Management, 153(1–3), 29–42, 2001.

[4] Borsuk, M.E., Stow, C.A. andReckhow, K.H. A Bayesian network of eutrophication models for synthesis, prediction, and uncertainty analysis. Ecological Modelling, 173, 219–239, 2004.

[5] Dai, H., Korb, K.B., Wallace, C.S. and Wu, X. A study of casual discovery with weak links and small samples, in Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufman, San Francisco, CA, pp. 1304–1309, 1997.

[6] Mani, S. and Cooper, G.F. A study in casual discovery from population-based infant birth and death records, in Proceedings of theAMIAAnnual Fall Symposium, Hanley and Belfus, Philadelphia, PA, pp. 315–319, 1999.

[7] Rajapakse, J.C., Zhou, J. Learning effective brain connectivity with dynamic Bayesian networks. NeuroImage 37, 749–760, 2007.

[8] Li, J.N., Wang, Z.J., Palmer, S.J., McKeown, M.J. Dynamic Bayesian network modeling of fMRI: A comparison of group-analysis methods, NeuroImage 37, 749–760. 2008.

[9] Li, J., and Shi, J. Knowledge Discovery from Observational Data for Process Control through Causal Bayesian Networks. IIE Transactions, 39(6), 681-690, 2007.

[10] De Campos, L. Independency Relationships and Learning Algorithms for Singly Connected Networks. J. Experimental and Theoretical Artificial Intelligence, vol. 10, pp. 511-549, 1998.

[11] De Campos, L. and Huete, J. A New Approach for Learning Belief Networks Using Independence Criteria. Int'l J. Approximate Reasoning, vol. 24, pp. 11-37, 2000.

[12] Pearl, J. and Verma, T. Equivalence and Synthesis of Causal Models. Proc. Sixth Conf. Uncertainty in Artificial Intelligence, 1990.

[13] Spirtes, P., Glymour, C. and Scheines, R. Causation, Prediction and Search. Lecture Notes in Statistics 81, Springer, 1993.

[14] Meek, C. Causal Inference and Causal Explanation with Background Knowledge. Proc. 11th Conf. Uncertainty in Artificial Intelligence, 1995.

[15] Cooper, G. and Herskovits, E. A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning, vol. 9, pp. 309-347, 1992.

[16] Heckerman, D., Geiger, D. and Chickering, D. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning, vol. 20, 197-243, 1995.

[17] Buntine, W. A Guide to the Literature on Learning Probabilistic Networks from Data. IEEE Trans. Knowledge and Data Eng., vol. 8, 195-210, 1996.

[18] Friedman, N. and Koller, D. Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. Machine Learning, vol. 50, 95-125, 2003.

[19] Heckerman, D. A Tutorial on Learning Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, 1996.

[20] Lam, W. and Bacchus, F. Learning Bayesian Belief Networks. An Approach Based on the MDL Principle. Computational Intelligence, vol. 10, pp. 269-293, 1994.

[21] Suzuki, J. A Construction of Bayesian Networks from Databases Based on an MDL Principle. Proc. Ninth Conf. Uncertainty in Artificial Intelligence, pp. 266-273, 1993.

[22] Bouckaert, R. Belief Networks Construction Using the Minimum Description Length Principle. Lecture Notes in Computer Science 747, pp. 41-48, 1993.

[23] Friedman, N. and Goldszmidt, M. Learning Bayesian Networks with Local Structure. Proc. 12th Conf. Uncertainty in Artificial Intelligence, 1996.

[24] Chow, C. and Liu, C. Approximating Discrete Probability Distributions with Dependence Trees. IEEE Trans. Information Theory, vol. 14, pp. 462-467, 1968.

[25] Chickering, D. Optimal Structure Identification with Greedy Search. J. Machine Learning Research, vol. 3, pp. 507-554, 2002.

[26] Acid, S. and De Campos, J. Searching for Bayesian Network Structures in the Space of Restricted Acyclic Partially Directed Graphs. J. Artificial Intelligence Research, vol. 18, pp. 445-490, 2003.

[27] Castelo, R. and Kocka, T. On Inclusion-Driven Learning of Bayesian Networks. J. Machine Learning Research, vol. 4, pp. 527-574, 2003.

[28] Larranaga, R., Kuijpers, C., Murga, R. and Yurramendi, Y. Learning Bayesian Network Structures by Searching for the Best Ordering with Genetic Algorithms. IEEE Trans. Systems, Man, and Cybernetics, vol. 26, pp. 487-493, 1996.

[29] Larranaga, P., Poza, M., Yurramendi, Y., Murga, R. and Kuijpers, C. Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, pp. 912-926, 1996.

[30] Chickering, D., Geiger, D. and Heckerman, D. Learning Bayesian Networks: Search Methods and Experimental Results. Preliminary Papers Fifth Int'l Workshop Artificial Intelligence and Statistics, 1995.

[31] Chen, X.W., Anantha, G. and Lin, X.T. Improving Bayesian Network Structure Learning with Mutual Information-Based Node Ordering in the K2 Algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, pp. 628-640, 2008.

[32] Hoyer, P.O., Janzing, D., Mooij, J.M., Peters, J. and Scholkopf, B. Nonlinear Causal Discovery with Additive Noise Models. NIPS 21, 2009.

[33] Peng, J., Wang, P., Zhou, N. and Zhu, J. Partial correlation estimation by joint sparse regression models., Journal of the American Statistical Association **104**:735-746, 2009.

[34] Sporns, O., Chialvo, D. R., Kaiser, M., and Hilgetag, C. C. Organization, Development and Function of Complex Brain Networks. Trends Cogn. Sci, 8, 418–425, 2004.

[35] Friedman, N.; Nachman, I. and Pe'er, D. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. UAI, 1999.

[36] Schmidt, M.; Niculescu-Mizil, A. and Murphy, K. Learning Graphical model structures using L1-Regularization paths. AAAI 2007.

[37] Tibshirani, R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society Series B, 58(1):267–288, 1996.

[38] Tsamardinos, I.; Brown, L.E., and Aliferis, C.F. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning, **65**(1), 31-78, 2006.

[39] Margaritis, D. and Thrun, S. Bayesian network induction via local neighborhoods. NIPS 12, 1999

[40] Pellet, J.P. and Elisseeff, A. Using Markov Blankets for Causal Structure Learning. Journal of Machine Learning Research 9, 1295-1342, 2008.

[41] Estrada, E. and Naomichi, H. Communicability in Complex Networks. Phys. Rev. E 77 036111, 2008.

[42] Luus, R. and Wyrwicz, R. Use of Penalty Functions in Direct Search Optimization. Hung. J. Ind. Chem. 24, 273-278, 1996.

[43] BERTSEKAS, D. P. Nonlinear Programming, 2nd Edition, Athena Scientific, Belmont, 1999.

[44] Fu, W. Penalized Regressions: The Bridge vs the Lasso, Journal of Computational and Graphical Statistics, 7 (3), 397–416, 1998.

[45] Cormen, T.H., Leiserson, C.E., Rivest, R.L and Stein, C. Introduction to Algorithms. 3rd edition, MIT Press.

[46] Friedman, J.;,Hastie, T., Hofling, H. and Tibshirani, R. Pathwise Coordinate Optimization. The Annals of Applied Statistics, 2007 (1), no.2, p.302-332, 2007.

[47] Aliferis, C. F., Tsamardinos, I. and Statnikov, A. HITON, a Novel Markov Blanket Algorithm for Optimal Variable Selection. AMIA, 2003.

[48] Tsamardinos, I. and Aliferis, C. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. Artificial Intelligence and Statistics, 2003.

[49] Bayesian Network Repository: http://www.cs.huji.ac.il/labs/compbio/Repository.

[50] Tsamardinos, I., Statnikov, A., Brown, L. E., and Aliferis, C. F. Generating Realistic Large Bayesian Networks by Tiling. In The 19th International FLAIRS Conference, 2006.

[51] Mackey, D. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

[52] Tzourio-Mazoyer, N., Automated Anatomical Labelling of Activations in SPM using a Macroscopic Anatomical Parcellation of the MNI MRI Single Subject Brain. Neuroimage,15:273-289, 2002.

[53] Supekar, K., Menon, V., Rubin, D., Musen, M., Greicius, M.D. Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer's Disease. PLoS Comput Biol 4(6) 1-11, 2008.

[54] Azari, N.P., Rapoport, S.I., Grady, C.L., Schapiro, M.B., Salerno, J.A. and Gonzales-Aviles, A. Patterns of Interregional Correlations of Cerebral Glucose Metabolic Rates in Patients with Dementia of the Alzheimer Type. Neurodegeneration 1: 101–111, 1992.

[55] Wang, K., Liang, M., Wang, L., Tian, L., Zhang, X. and Jiang, T. Altered Functional Connectivity in Early Alzheimer's Disease: A Resting-State fMRI Study. Human Brain Mapping 28, 967-978, 2007.

[56] Gould, R.L., Arroyo, B., Brown, R,G., Owen, A.M. and Howard, R.J. Brain Mechanisms of Successful Compensation during Learning in Alzheimer Disease. Neurology 67, 1011-1017, 2006.

[57] Stern, Y. Cognitive Reserve and Alzheimer Disease. Alzheimer Disease Associated Disorder 20, 69-74, 2006.

[58] Korb, K.B. and Nicholson, A.E. Bayesian Artificial Intelligence. Chapman & Hall/CRC, London, UK, 2003.

[59] Friston, K.J. Functional and Effective Connectivity in Neuroimaging: A Synthesis. Human Brain Mapping 2, 56-78, 1994.

[60] Greicius, M.D., Srivastava, G., Reiss, A.L. and Menon, V. Default-mode Network Activity Distinguishes AD from Healthy Aging: Evidence from Functional MRI. Proc. Natl. Acad. Sci. 101, 4637–4642, 2004.

[61] Alexander, G.E., Chen, K., Pietrini, P., Rapoport, S.I., Reiman, E.M. Longitudinal PET Evaluation of Cerebral Metabolic Decline in Dementia: A Potential Outcome Measure in

Alzheimer's Disease Treatment Studies. Am.J.Psychiatry 159, 738-745, 2002.

[62] Braak, H., Braak, E. Evolution of the Neuropathology of Alzheimer's Disease. Acta Neurol Scand Suppl 165, 3-12, 1996.

[63] Braak, H., Braak, E., Bohl, J. Staging of Alzheimer-related Cortical Destruction. Eur Neurol 33, 403-408, 1993.

[64] Ikonomovic, M.D., Klunk, W.E., Abrahamson, E.E., Mathis, C.A., Price, J.C., Tsopelas, N.D., Lopresti, B.J., Ziolko, S., Bi, W., Paljug, W.R., Debnath, M.L., Hope, C.E., Isanski, B.A., Hamilton, R.L., DeKosky, S.T.. Post-mortem Correlates of in vivo PiB-PET Amyloid Imaging in a Typical Case of Alzheimer's Disease. Brain 131, 1630-1645, 2008.

[65] Klunk, W.E., Engler, H., Nordberg, A., Wang, Y., Blomqvist, G., Holt, D.P., Bergstrom, M., Savitcheva, I., Huang, G.F., Estrada, S., Ausen, M.L., Barletta, J., Price, J.C., Sandell, J., Lopresti, B.J., Wall, A., Koivisto, P., Antoni, G., Mathis, C.A. and Langstrom, B.. Imaging Brain Amyloid in Alzheimer's Disease with Pittsburgh Compound-B. Ann Neurol. 55, 306-319, 2004.

[66] Reuter-Lorenz, P.A. and Mikels, J.A. A Split-Brain Model of Alzheimer's Disease? Behavioral Evidence for Comparable Intra and Interhemispheric Decline. Neuropscyhologia 43, 1307-1317, 2005.

[67] Lipton, A.M., Benavides, R., Hynan, L.S., Bonte, F.J., Harris, T.S., White, C.L. 3rd, Bigio, E.H. Lateralization on Neuroimaging does not Differentiate Frontotemporal Lobar Degeneration from Alzheimer's Dsease. Dement Geriatr Cogn Disord 17(4), 324-327, 2004.

[68] Hedden, T., Van Dijk, K.R., et al Disruption of Functional Connectivity in Clinically Normal Older Adults Harboring Amyloid Burden. J. Neurosci. 29, 12686–12694, 2009.

[69] Andrews-Hanna, J.R., Snyder, A.Z., et al. Disruption of Large-Scale Brain Systems in Advanced Aging,.Neuron 56, 924–935, 2007.

[70] Wu, X., Li, R., Fleisher, A.S., Reiman, E.M., Chen, K. and Yao, L.. Altered Default Mode Network Connectivity in AD -- A Resting Functional MRI and Bayesian Network Study. Human Brain Mapping, in press.

[71] Efron, B. and Ribshirani, R.J. An Introduction to the Bootstrap. CRC Press, 1994.

Good, P. Permutation, Parametric and Bootstrap Tests of Hypotheses. 3rd ed., Springer, 2005.

Meinshausen, N. and Buehlmann, P. Stability Selection. Journal of the Royal Statistical Society, Series B, vol.72, 417-473, 2010.

[72] Good, P. Permutation, Parametric and Bootstrap Tests of Hypotheses. 3rd ed., Springer, 2005.

[73] Meinshausen, N. and Buehlmann, P. Stability Selection. Journal of the Royal Statistical Society, Series B, vol.72, 417-473, 2010.

Chapter 4

A TRANSFER LEARNING APPROACH FOR NETWORK MODELING

Abstract

Networks models have been widely used in many domains to characterize the interacting relationship between physical entities. A typical problem faced is to identify the networks of multiple related tasks that share some similarities. In this case, a transfer learning approach that can leverage the knowledge gained during the modeling of one task to help better model another task is highly desirable. In this paper, we propose a transfer learning approach, which adopts a Bayesian hierarchical model framework to characterize task relatedness and additionally uses the L_1-regularization to ensure robust learning of the networks with limited sample sizes. A method based on the Expectation-Maximization (EM) algorithm is further developed to learn the networks from data. Simulation studies are performed, which demonstrate the superiority of the proposed transfer learning approach over single task learning that learns the network of each task in isolation. The proposed approach is also applied to identification of brain connectivity networks of Alzheimer's disease (AD) from functional magnetic resonance image (fMRI) data. The findings are consistent with the AD literature.

4.1 Introduction

Network models have been extensively used in many domains to characterize the interacting relationship between physical entities. For example, they have been used to model how different genes interact in a biological process and the resulting networks are called gene association networks (Friedman et al., 2000). They have been used to model how different brain regions interact to jointly deliver a brain function such as cognition and emotion, and the resulting networks are called brain connectivity networks (Huang et al., 2010). They have also been used to model the relationship between process and

product quality variables for quality control of manufacturing processes (Li and Shi, 2007). With the advancement of sensing technologies, network models can be learned from the rich amounts of sensing data, such as gene micro-arrays, brain images, and production data for the aforementioned networks, respectively. Note that the network models focused on in this paper are also called graphical models. Learning graphical models from data has been a popular research area in statistics and machine learning.

Existing research in graphical models focuses on learning a network/graphical model for a single task. However, many real-world problems involve learning of network models for multiple related tasks (i.e., one model for each task). For example, there may be a group of Alzheimer's disease (AD) patients for each of whom we want to learn a brain connectivity network based on his/her functional magnetic resonance image (fMRI) data. The purpose of such a study may be to identify brain connectivity patterns common to AD patients, which have the potential for being used as AD biomarkers to help clinical diagnosis. Here, each patient is a task; these tasks/patients are related in the sense that they have the same disease and thus their respective brain connectivity networks may share some similarities. Because of the similarities, the networks of the AD patients should be learned jointly, rather than independently, to leverage the knowledge gained in the network modeling of one patient to help better model another patient. This kind of joint learning is called "transfer learning" in this paper. Transfer learning is especially useful when the data of each task has a low sample size, such as the fMRI data of each AD patient. In this case, transfer learning allows for use of data of other related tasks, in an appropriate way, to compensate for the sample shortage in each task.

Transfer learning is useful not only in the aforementioned multi-subject studies, but also in multi-time longitudinal studies. For example, it may be of interest to learn brain connectivity networks for several longitudinal time points of an AD patient to track

the disease progression. Despite difference, these networks should share some similarities because they all correspond to the same patient and disease progression is a continuous process. As a result, transfer learning can be used to learn these networks jointly to enable knowledge transfer between them.

In addition to brain connectivity networks, transfer learning may be useful in other applications. For example, it may be used for modeling gene association networks of patients with the same type of cancer or gene association networks at longitudinal time points of a cancer patient. As another example, transfer learning may be used for modeling process-quality interactions of several products belonging to the same product family or process-quality interactions of different generations of a product.

Transfer learning is a natural skill of human beings. For example, we may find that learning to recognize apples may help recognize pears; learning to play an electronic organ may help learn a piano. Transfer learning in statistics and machine learning has focused on predictive models such as regressions and neural networks (Bakker and Heskes, 2003; Baxter, 2000; Caruana, 1997; Lawrence and Platt, 2004; Zhang et al., 2006). One major difference of these existing works is that they characterize the relatedness of the tasks in different ways. Thrun and O'Sullivan (1996) proposed use of a distance metric to evaluate the relatedness between tasks. In the setting of neural networks, task relatedness was reflected by shared hidden nodes between tasks (Caruana, 1997). The recent work by Zhang et al. (2006) assumed that task parameters are generated from independent sources which account for the relatedness of the tasks. Several studies have adopted the Bayesian hierarchical modeling framework and taken the relatedness into account by placing a common prior on model parameters of the tasks (Lawrence and Platt, 2004; Yu et al. 2005; Xue et al., 2007). Despite the popularity of

transfer learning in predictive models, limited work has been done on transfer learning of graphical/network models. This paper intends to bridge this gap.

In this paper, we propose a transfer learning approach for network modeling of multiple related tasks. We focus on one particular type of network model called Gaussian Graphical Model (GGM). A GGM consists of nodes that are random variables following a multivariate normal distribution and undirected arcs that indicate non-zero partial correlations between variables. Various methods have been developed for learning a GGM from data, which are also known as methods for inverse covariance (IC) estimation, because the undirected arcs in a GGM correspond to nonzero entries in the IC matrix of the data. These methods are reviewed as follows. Note that transfer learning is not considered in these existing methods.

One class of methods for GGM learning is based on regression. For example, a variable-by-variable approach for neighborhood selection via the lasso regression was developed by Meinshausen and Buhlmann (2006). A joint sparse regression model, which simultaneously performs neighborhood selection for all variables, was developed by Schafer and Strimmer (2005). A sparse regression technique called SPACE, which is particularly useful in identifying hubs in gene association networks, was developed by Peng et al. (2009). Another class of methods employs the maximum likelihood framework. A penalized maximum likelihood approach that performs model selection and estimation simultaneously was proposed by Yuan and Lin (2007). Further, efficient algorithms were proposed by Friedman et al. (2007) and Levina et al. (2008) to implement the penalized maximum likelihood approach by Yuan and Lin (2007), which are applicable to high-dimensional problems. Some other methods were proposed, such as a method based on threshold gradient descent regularization developed by Li and Gui (2006) and a method for overcoming the ill-conditioned problem of the sample

covariance matrix by Schafer and Strimmer (2005). In addition, there are methods dealing with the situations when variables have a natural ordering (Bickel and Levina, 2008; Levina et al, 2008).

Different from these existing methods, the approach we propose enables transfer learning in learning of GGMs for multiple related tasks. Specifically, we adopt the Bayesian hierarchical modeling (BHM) framework in our problem formulation to characterize the relatedness of tasks. We further add L_1-regularization to our problem formulation. L_1-regularization has been well-known to be able to discourage truly zero parameters or small-valued parameters from showing up in the learned model. This is especially advantageous when the model is high-dimensional and the sample size is limited, in which case conventional statistical estimation methods without regularization, such as the Maximum Likelihood Estimation (MLE), may generate unreliable estimates. L_1-regularization has been adopted in regressions (Tibshirani, 1996) and graphical models without transfer learning (Friedman et al., 2007), and have demonstrated effectiveness. Under the proposed problem formulation, we further develop a method based on the Expectation-Maximization (EM) algorithm (Dellaert, 2002) to solve the problem, i.e., to jointly learn GGMs for multiple related tasks with transfer learning enabled. Furthermore, we conduct simulation studies to compare performance of the proposed transfer learning approach with single task learning that learns GGMs for each task in isolation. Finally, we apply the proposed approach to a real-world application of brain connectivity network modeling for AD based on fMRI data. 15 AD patients are considered as related tasks and the purpose is to identify common patterns shared by their brain connectivity networks, in contrast with the normal brain connectivity networks of 16 matched normal controls (NCs).

4.2 Introduction to GGM

A GGM consists of nodes, $\mathbf{X} = \{X_1, \cdots, X_p\}$, and undirected arcs. The nodes are random variables following a multivariate normal distribution, i.e., $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let $\boldsymbol{\Theta}$ be the inverse covariance (IC) matrix of the distribution, i.e., $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$. There is an arc between nodes X_i and X_j if and only if the entry at the i-th row and j-th column of $\boldsymbol{\Theta}$ is nonzero. Please see Fig. 4-1 for an example of a GGM and the corresponding IC matrix.

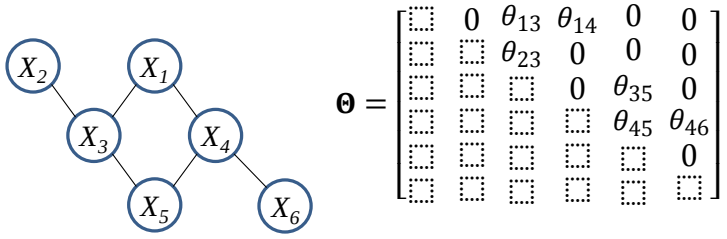


Fig. 4-1 A GGM and the corresponding IC matrix ($\theta_{ij} \neq 0$; only entries at the upper triangle are shown because the matrix is symmetric and the diagonal entries are not used in the GGM)

Given data on the nodes, the GGM can be learned by estimating the IC matrix. It is straightforward to derive the log-likelihood of $\boldsymbol{\Theta}$, which is $log|\boldsymbol{\Theta}| - tr(\mathbf{S}\boldsymbol{\Theta})$, where $|\cdot|$ and $tr(\cdot)$ denote the determinant and trace of a matrix, respectively, and $\mathbf{S}$ is the sample covariance matrix. By maximizing this log-likelihood, we can obtain the MLE for $\boldsymbol{\Theta}$, which is $\widehat{\boldsymbol{\Theta}}^{MLE} = \mathbf{S}^{-1}$. However, for large-scale GGMs with limited sample sizes, the MLE may be quite unreliable in the sense that many zero entries in $\boldsymbol{\Theta}$ may be nonzero in $\widehat{\boldsymbol{\Theta}}^{MLE}$, leading to a densely connected GGM that is hard to interpret. In the extreme case when the sample size is less than the number of nodes, $\mathbf{S}$ is not invertible. To tackle this deficiency of the MLE, a well-known strategy is to maximize the L_1-regularized log-likelihood function, i.e.,

$$\widehat{\boldsymbol{\Theta}} = \operatorname{argmax}_{\boldsymbol{\Theta} > \mathbf{0}} log|\boldsymbol{\Theta}| - tr(\mathbf{S}\boldsymbol{\Theta}) - \lambda\|\boldsymbol{\Theta}\|_1, \quad (1)$$

where $\|\boldsymbol{\Theta}\|_1$ is the so-called L_1-norm of $\boldsymbol{\Theta}$, i.e., the sum of the absolute values of the entries in $\boldsymbol{\Theta}$. λ is the regularization parameter; the larger the λ, the more zero entries the $\widehat{\boldsymbol{\Theta}}$ will have, i.e., the $\widehat{\boldsymbol{\Theta}}$ will be sparser. λ can be specified by the user or cross-validation. (1) is best known as the method of graphical LASSO (Friedman et al., 2007), which can be efficiently solved by the Block Coordinate Descent (BCD) algorithm (Sun et al., 2009).

4.3 Problem formulation and transfer learning in GGM

Assume that there are m related tasks. To estimate the IC matrix for each task, $\boldsymbol{\Theta}_i$, the graphical LASSO in (1) may be used, i.e.,

$$\widehat{\boldsymbol{\Theta}}_i \quad = \quad \text{argmax} \quad _{\boldsymbol{\Theta}_i \succ \mathbf{0}} \, log|\boldsymbol{\Theta}_i| - tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1 , \qquad (2)$$

where $\mathbf{S}_i$ is the sample covariance matrix for task i, $i = 1, \cdots, m$. This method treats the tasks as independent and does not exploit their relatedness. Alternatively, we propose to consider the relatedness by assuming that the $\boldsymbol{\Theta}_i$'s are "samples" drawn from the same probability distribution, i.e., we adopt the BHM framework to characterize the task relatedness (Fig. 4-2). This same probability distribution is chosen to be a Wishart distribution, i.e., $\boldsymbol{\Theta}_i \sim Wishart_v(\boldsymbol{\Theta}^h)$, where $\boldsymbol{\Theta}^h$ is a $p \times p$ positive definite matrix (called the scale matrix) and $v > p - 1$ (called the degrees of freedom). This choice is based on the following considerations: (i) The $\boldsymbol{\Theta}_i$'s are symmetric, positive-definite matrices and the Wishart distribution is developed for matrices of such characteristics. (ii) In Bayesian inference, the Wishart distribution has been commonly used as the prior distribution of the IC matrix of a multivariate normal distribution. (iii) The degrees of freedom, v, of the Wishart distribution is nicely interpretable in our problem, as will be shown in Section 4-4-3. (iv) The scale matrix, $\boldsymbol{\Theta}^h$, of the Wishart distribution depicts how the tasks are related. Specifically, because

$$\text{E}(\boldsymbol{\Theta}_i) = v\boldsymbol{\Theta}^h, \qquad (3)$$

the tasks are related in the sense that their respective IC matrices share the same prior mean.

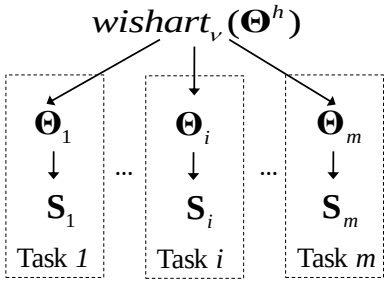


Fig. 4-2 A BHM framework for characterizing task relatedness

To enable transfer learning, the $\boldsymbol{\Theta}_i$'s should be estimated by utilizing not only the data, i.e., the $\mathbf{S}_i$'s, but also the parameters of the common Wishart distribution, $\boldsymbol{\Theta}^h$ and v. The challenge here is that $\boldsymbol{\Theta}^h$ and v are unknown. v may be specified by the user or identified from data by cross-validation. This strategy, however, does not work for $\boldsymbol{\Theta}^h$ because $\boldsymbol{\Theta}^h$ is a high-dimensional matrix. Therefore, we propose to integrate out $\boldsymbol{\Theta}^h$ during the estimation of the $\boldsymbol{\Theta}_i$'s, i.e., we aim to find an estimate, $\widehat{\boldsymbol{\Theta}}_i$, for each $\boldsymbol{\Theta}_i$, that maximizes the logarithm of the posterior probability of $\boldsymbol{\Theta}_i$'s given the data, marginalizing over $\boldsymbol{\Theta}^h$. This can be written as:

$$\{\widehat{\boldsymbol{\Theta}}_i\}_{i=1,2,\dots,m} \operatorname{argmax}_{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}} \int_{\boldsymbol{\Theta}^h} \log P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h | \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) d\boldsymbol{\Theta}^h. \quad (4)$$

Equivalently, we can maximize the logarithm of the joint probability which is proportional to the posterior probability in (4), i.e.,

$$\{\widehat{\boldsymbol{\Theta}}_i\}_{i=1,2,\dots,m} = \operatorname{argmax}_{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}} \int_{\boldsymbol{\Theta}^h} \log P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) d\boldsymbol{\Theta}^h. \quad (5)$$

In (5), the $P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right)$ can be written as

$P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) =$

$P\left(\{\mathbf{S}_i\}_{i=1,2,\dots,m} | \{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h\right) P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m} | \boldsymbol{\Theta}^h\right) P\left(\boldsymbol{\Theta}^h\right)$. According to the structure in Fig. 4-2, the $\boldsymbol{\Theta}_i$'s are conditionally independent given $\boldsymbol{\Theta}^h$, so $P\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m} | \boldsymbol{\Theta}^h\right) =$

$\prod_{i=1}^{m} P(\boldsymbol{\Theta}_i|\boldsymbol{\Theta}^h)$. Each $\mathbf{S}_i$ is independent of $\boldsymbol{\Theta}^h$ given $\boldsymbol{\Theta}_i$, so $P(\{\mathbf{S}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h) = \prod_{i=1}^{m} P(\mathbf{S}_i|\boldsymbol{\Theta}_i)$. Therefore, (5) can be further written as:

$$\{\widehat{\boldsymbol{\Theta}}_i\}_{i=1,2,\dots,m} = \underset{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}}{\operatorname{argmax}} \int_{\boldsymbol{\Theta}^h} log\left(P(\boldsymbol{\Theta}^h)\prod_{i=1}^{m} P(\boldsymbol{\Theta}_i|\boldsymbol{\Theta}^h)\prod_{i=1}^{m} P(\mathbf{S}_i|\boldsymbol{\Theta}_i)\right) d\boldsymbol{\Theta}^h, \quad (6)$$

where $P(\boldsymbol{\Theta}_i|\boldsymbol{\Theta}^h) \propto |\boldsymbol{\Theta}^h|^{-v/2}|\boldsymbol{\Theta}_i|^{(v-p-1)/2}\ e^{-tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right)/2}$ and $P(\mathbf{S}_i|\boldsymbol{\Theta}_i) \propto |\boldsymbol{\Theta}_i|^{n_i/2}\ e^{-n_i tr(\mathbf{S}_i\boldsymbol{\Theta}_i)/2}$. Then, (6) becomes:

$$\{\widehat{\boldsymbol{\Theta}}_i\}_{i=1,2,\dots,m} =$$

$$\operatorname{argmax}_{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}} \int_{\boldsymbol{\Theta}^h} \sum_{i=1}^{m} \left\{\frac{v-p-1+n_i}{2} log|\boldsymbol{\Theta}_i| - \frac{1}{2} tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right) - \frac{n_i}{2} tr(\mathbf{S}_i\boldsymbol{\Theta}_i)\right\} d\boldsymbol{\Theta}^h \quad (7)$$

Moreover, due to the same consideration as graphical LASSO, we add L_1-regularization to (7), which gives the final objective function, i.e.,

$$\{\widehat{\boldsymbol{\Theta}}_i\}_{i=1,2,\dots,m} = \operatorname{argmax}_{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}} \int_{\boldsymbol{\Theta}^h} \sum_{i=1}^{m} \left\{\frac{v-p-1+n_i}{2} log|\boldsymbol{\Theta}_i| - \frac{1}{2} tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right) - \frac{n_i}{2} tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1\right\} d\boldsymbol{\Theta}^h, \quad (8)$$

4.4 Problem solving by an EM algorithm

It is difficult to solve (8) directly, since it involves an integral over a high-dimensional matrix $\boldsymbol{\Theta}^h$. We propose a method based on the EM algorithm. Adopting the EM algorithm is a natural choice because we want to integrate out $\boldsymbol{\Theta}_h$ in (8), i.e., treat $\boldsymbol{\Theta}_h$ as latent, and the EM algorithm is a well-known approach for model estimation with latent variables. In this section, we first introduce the general EM algorithm, and then describe the proposed method.

4.4.1 Introduction to general EM algorithm

Consider a probabilistic model that depends on some observed variables, **Y**, and some unobserved latent variables, **Z**. Let **Ω** denote the parameters of the model that need to be

estimated. One way of estimation is to find estimates for $\mathbf{\Omega}$ that maximize the posterior probability of $\mathbf{\Omega}$ given $\mathbf{Y}$ and $\mathbf{Z}$. However, as $\mathbf{Z}$ is latent, the estimation cannot be done directly, but through an iterative algorithm that alternates between an E (expectation) step and an M (maximization) step. Specifically, the E step is to calculate the expectation of the logarithm of the joint distribution of $\mathbf{\Omega}$, $\mathbf{Z}$, and $\mathbf{Y}$ (which is proportional to the posterior probability of $\mathbf{\Omega}$ given $\mathbf{Y}$ and $\mathbf{Z}$), with respect to the conditional distribution of $\mathbf{Z}$ given $\mathbf{Y}$ under the current (i.e., the t-th iteration) estimates for the parameters $\mathbf{\Omega}^t$. Denote this expectation by $Q(\mathbf{\Omega}|\mathbf{\Omega}^t)$, i.e.,

$$\textbf{E step: } Q(\mathbf{\Omega}|\mathbf{\Omega}^t) = E_{\mathbf{Z}|\mathbf{Y},\mathbf{\Omega}^t}\{L(\mathbf{\Omega},\mathbf{Z},\mathbf{Y})\} = \int_{\mathbf{Z}} L(\mathbf{\Omega},\mathbf{Z},\mathbf{Y})P(\mathbf{Z}|\mathbf{Y},\mathbf{\Omega}^t)\,d\mathbf{Z}$$

where, $L(\mathbf{\Omega},\mathbf{Z},\mathbf{Y}) = logP(\mathbf{\Omega},\mathbf{Z},\mathbf{Y})$. Then, the M step is to find the parameters that maximize $Q(\mathbf{\Omega}|\mathbf{\Omega}^t)$, i.e.,

$$\textbf{M step: } \mathbf{\Omega}^{t+1} = \underset{\mathbf{\Omega}}{\text{argmax}} Q(\mathbf{\Omega}|\mathbf{\Omega}^t)$$

These parameters $\mathbf{\Omega}^{t+1}$ are then used for the next E step. Such iterations have been proven to converge (Wu, 1983).

4.4.2 Solving the transfer learning formulation under EM framework

To fit the transfer learning formulation in (8) into the EM framework, we consider $\{\mathbf{\Theta}_i\}_{i=1,2,\dots,m}$, $\mathbf{\Theta}^h$, and $\{\mathbf{S}_i\}_{i=1,2,\dots,m}$ to be the parameters to be estimated, the latent and the observed variables (i.e., $\mathbf{\Omega}$, $\mathbf{Z}$, and $\mathbf{Y}$), respectively. Then, the E step is to calculate $\int_{\mathbf{\Theta}^h} L\left(\{\mathbf{\Theta}_i\}_{i=1,2,\dots,m}, \mathbf{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) P\left(\mathbf{\Theta}^h|\{\mathbf{S}_i\}_{i=1,2,\dots,m}, \{\mathbf{\Theta}_i^t\}_{i=1,2,\dots,m}\right) d\mathbf{\Theta}^h$. According to Fig. 4-2, $\mathbf{\Theta}^h$ is independent of $\{\mathbf{S}_i\}_{i=1,2,\dots,m}$ given $\{\mathbf{\Theta}_i^t\}_{i=1,2,\dots,m}$, i.e., $P\left(\mathbf{\Theta}^h|\{\mathbf{S}_i\}_{i=1,2,\dots,m}, \{\mathbf{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = P\left(\mathbf{\Theta}^h|\{\mathbf{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$. Also, according to (8),

$L\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) = \sum_{i=1}^{m}\left\{\frac{v-p-1+n_i}{2} log|\boldsymbol{\Theta}_i| - \frac{1}{2} tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right) - \frac{n_i}{2} tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1\right\}.$ **(9)**

Then, the E and M steps are:

E step: $Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) =$

$$\int_{\boldsymbol{\Theta}^h} L\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}, \boldsymbol{\Theta}^h, \{\mathbf{S}_i\}_{i=1,2,\dots,m}\right) P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) d\boldsymbol{\Theta}^h \quad (10)$$

where $L(\cdot)$ is given in (9).

M step: $\{\boldsymbol{\Theta}_i^{t+1}\}_{i=1,2,\dots,m} = \underset{\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}}{\text{argmax}}\ Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$

To conduct the E and M steps, we further decompose them into sub-steps:

Conducting the E step:

We decompose the E step into two sub-steps: finding the parametric form of the distribution $P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$, and then transforming the $Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$ into a form that facilitates the maximization in the M step. Results of these two sub-steps are summarized in Propositions 1 and 2, respectively (proofs are given in the Appendix).

Proposition 1: The probability distribution of $\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}$ is an Inverse-Wishart distribution with scale matrix $\sum_{i=1}^{m}\boldsymbol{\Theta}_i^t$ and degrees of freedom $mv-p-1$.

Proposition 2: The $Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$ can be decomposed into a sum of m terms, with the i^{th} term involving only $\boldsymbol{\Theta}_i$ not $\boldsymbol{\Theta}_j$ ($j \neq i$), i.e., $\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = \sum_{i=1}^{m}\left\{\frac{v-p-1+n_i}{2} log|\boldsymbol{\Theta}_i| - \frac{n_i}{2} tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1 - \frac{1}{2} tr\left((mv-p-1)\left(\sum_{i=1}^{m}\boldsymbol{\Theta}_i^t\right)^{-1}\boldsymbol{\Theta}_i\right)\right\}$.

Conducting the M step:

We also decompose the M step into two sub-steps. First, as a result of Proposition 2, the maximization in the M step can be carried out by solving m smaller-scale maximization problems, i.e.,

Corollary 2.1: To find the $\{\mathbf{\Theta}_i^{t+1}\}_{i=1,2,\dots,m}$ can be achieved by solving m optimization problems, i.e.,

$$\mathbf{\Theta}_i^{t+1} = \underset{\mathbf{\Theta}_i}{\operatorname{argmax}}\left\{\frac{v-p-1+n_i}{2} log|\mathbf{\Theta}_i| - \frac{n_i}{2} tr(\mathbf{S}_i\mathbf{\Theta}_i) - \lambda_i\|\mathbf{\Theta}_i\|_1 - \frac{1}{2} tr\left((mv-p-1)\left(\sum_{i=1}^{m}\mathbf{\Theta}_i^t\right)^{-1}\mathbf{\Theta}_i\right)\right\}, \quad (11)$$

Next, we need to develop an efficient algorithm to solve each optimization in (11). Through some algebra, it can be found that solving (11) is equivalent to solving (12),

$$\mathbf{\Theta}_i^{t+1} = \underset{\mathbf{\Theta}_i}{\operatorname{argmax}}\left\{log|\mathbf{\Theta}_i| - tr\left(\mathbf{S}_i^t\mathbf{\Theta}_i\right) - \lambda'_i\|\mathbf{\Theta}_i\|_1\right\}, \quad (12)$$

where $\lambda'_i = \frac{2}{v-p-1+n_i}\lambda_i$, and

$$\mathbf{S}_i^t = \frac{n_i\mathbf{S}_i+(mv-p-1)\left(\sum_{k=1}^{m}\mathbf{\Theta}_k^t\right)^{-1}}{v-p-1+n_i}. \quad (13)$$

(12) has a similar form to the graphical LASSO in (2). The difference is that the graphical LASSO has $\mathbf{S}_i$ instead of $\mathbf{S}_i^t$, so it does not enable transfer learning. To solve (12), we adopt the efficient BCD algorithm (Sun et al., 2009), which was originally developed for graphical LASSO, to our problem and prove the convergence (see Appendix).

To conclude this section, we give the practical steps for solving the transfer learning formulation in GGM learning in Fig. 4-3.

Input: sample covariance matrixes, $\{\mathbf{S}_i\}_{i=1,2,\ldots,m}$; sample size for each task, n_i; number of variable, p; number of tasks, m; regularization parameters, $\{\lambda_i\}_{i=1,2,\ldots,m}$; degrees of freedom in the common prior Wishart distribution, v; initial $\{\mathbf{\Theta}_i^0 \succ 0\}_{i=1,2,\ldots,m}$; stopping criterion, ϵ.

Initialize:

Compute $\{\mathbf{S}_i^0\}_{i=1,2,\ldots,m}$ by (13);

Let $converge = false$;

Let $t = 0$;

Repeat

Use the BCD algorithm to solve (12) and get $\{\mathbf{\Theta}_i^{t+1}\}_{i=1,2,\ldots,m}$;

Compute $\{\mathbf{S}_i^{t+1}\}_{i=1,2,\ldots,m}$ by (13);

If $\sum_{i=1}^{m}\|\mathbf{\Theta}_i^{t+1} - \mathbf{\Theta}_i^{t}\|_2 \leq \epsilon$ **then**

$converge = true$;

Else

$converge = false$;

End if

Let $t = t + 1$;

Until $converge = true$

Output: $\{\mathbf{\Theta}_i^{t+1}\}_{i=1,2,\ldots,m}$ as the estimated IC matrix (i.e., GGM) for each task.

Fig. 4-3 Steps for solving the transfer learning formulation in GGM learning

4.4.3 Analysis of transfer learning

Here, we would like to discuss some intuition and rationale behind the proposed transfer learning approach. Because of (3), i.e., $\mathrm{E}(\mathbf{\Theta}_k) = v\mathbf{\Theta}^h$, the $\sum_{k=1}^{m}\mathbf{\Theta}_k^t$ in (13) can be considered as the estimate for $mv\mathbf{\Theta}^h$ obtained at iteration t of the EM algorithm, i.e.,

$$\sum_{k=1}^{m}\mathbf{\Theta}_k^t = mv\mathbf{\Theta}^{h,t} \tag{14}$$

Inserting (14) into (13),

$$\mathbf{S}_i^t = \frac{n_i\mathbf{S}_i + \frac{mv-p-1}{mv}\left(\mathbf{\Theta}^{h,t}\right)^{-1}}{v-p-1+n_i}. \tag{15}$$

(15) indicates that $\mathbf{S}_i^t$ is a combination of two information sources: $\mathbf{S}_i$, which is specific to task i, and $\mathbf{\Theta}^{h,t}$, which is common to all the tasks. According to (14), $\mathbf{\Theta}^{h,t} = \sum_{k=1}^{m}\mathbf{\Theta}_k^t/mv$, i.e., $\mathbf{\Theta}^{h,t}$ utilizes the information in all the tasks. Therefore, $\mathbf{S}_i^t$ embraces

not only the information specific to task i, but also information in other tasks through $\mathbf{\Theta}^{h,t}$. Because $\mathbf{S}_i^t$ is used in (12) for learning the IC matrix of task i, the learning makes use of the information in all the task; or in other words, the information in other tasks is "transferred" into the learning of task i. This is the fundamental difference between the proposed approach and the graphical LASSO, in which learning of the IC matrix of task i only utilizes $\mathbf{S}_i$.

An interesting observation on (15) is that, for a given dataset (i.e., $n_i, p,$ and m are fixed), v determines the relative weights of the two information sources. The larger the v, the more the $\mathbf{\Theta}^{h,t}$ will be utilized in obtaining $\mathbf{S}_i^t$ and further in learning the IC matrix of task i. In other words, by specifying a larger v, we want more information in other tasks to be transferred to the learning of task i. However, the value of v can be hardly known as *a priori* in practical data analysis. We propose the use of cross-validation for selecting an appropriate v, which will be discussed in the next section.

Also, (15) reveals how transfer learning is affected by $n_i, p,$ and m. Specifically, with other parameters fixed, more information in task i will be utilized when task i has more samples, i.e., smaller n_i. Furthermore, when more tasks are learned together (bigger m), $\mathbf{\Theta}^{h,t}$ will be weighted heavier relative to $\mathbf{S}_i$. This makes sense because more tasks will help obtain a more reliable estimate for $\mathbf{\Theta}^{h,t}$. In addition, when the dimensionality of each task is higher (bigger p), $\mathbf{\Theta}^{h,t}$ will be weighted less. This also makes sense because $\mathbf{\Theta}^{h,t} = \sum_{k=1}^{m} \mathbf{\Theta}_k^t / mv$ and $\mathbf{\Theta}_k^t$'s are learned from data; in general, statistical learning suffers more as the dimensionality of problems increases. By weighting $\mathbf{\Theta}^{h,t}$ less, the proposed transfer learning approach provides a robust mechanism to tackle high dimensionality.

4.4.4 Selection of parameters v and λ_i

To apply the proposed algorithm in Fig. 4-3 to solve (8), v and λ_i ($i = 1, \ldots, m$) need to be specified. This can be accomplished by cross-validation. For example, if a V-fold cross-validation is used, the following steps can be performed. First, the dataset of each task is partitioned into V subsamples. $V - 1$ subsamples are used for learning an IC matrix for each task by applying the proposed algorithm at a given v and λ. Here, we assume the same λ for all the tasks for simplicity; making the λ_i's different is just a straightforward extension. Next, based on the learned IC matrix of each task, the likelihood of the data in the remaining one subsample in this task is computed. The likelihood values corresponding to all the tasks are summed together and we obtain an "overall" likelihood for the given v and λ. This process is repeated V times for the same v and λ, and each time a different subsample is used for computing the overall likelihood. Then, the average overall likelihood is computed over the V repetitions. This whole procedure is performed on different combinations of values for v and λ (i.e., a grid search on v and λ). The final v and λ selected are the ones maximizing the average overall likelihood. Note that this proposed cross-validation procedure is similar to the likelihood-based cross-validation, which has been commonly used in probability density function estimation.

4.5 Simulation study

The simulation study consists of the following steps:

(i) Construct the common prior mean matrix, $\mathbf{\Theta}_h$. Because $\mathrm{E}(\mathbf{\Theta}_i) = v\mathbf{\Theta}^h$, we first need to construct $\mathbf{\Theta}^h$. Specifically, the initial value for $\mathbf{\Theta}^h$, $\tilde{\mathbf{\Theta}}^h$, is generated by

$$\tilde{\tau}_{jk}^h = \begin{Bmatrix} 1 & j = k \\ 0 & j \neq k, j \nleftrightarrow k \\ \sim Uniform(D) & j \neq k, j \leftrightarrow k \end{Bmatrix},$$

where $\tilde{\tau}_{jk}^{h}$ denotes the entry at the j-th row and k-th column of $\widetilde{\boldsymbol{\Theta}}^{h}$; $j \leftrightarrow k$ means that there is an arc between nodes j and k, and $j \nleftrightarrow k$ means otherwise; $D = [-1, -0.5] \cup [0.5,1]$. $\widetilde{\boldsymbol{\Theta}}^{h}$ is then rescaled to ensure the positive definiteness. The rescaling includes first summing the absolute values of the off-diagonal entries for each row, then dividing each off-diagonal entry by 1.5 fold of the sum, and finally averaging the resulting matrix with its transpose to ensure the symmetry. This rescaling process was suggested by Peng et al. (2009). The rescaled matrix is $\boldsymbol{\Theta}^{h}$. Furthermore, let $\boldsymbol{\Theta}^{h} = \boldsymbol{\Theta}^{h}/v$.

(ii) Construct the true IC matrices of m related tasks, $\boldsymbol{\Theta}_i$, $i = 1, \cdots, m$. To generate a $\boldsymbol{\Theta}_i$, the following substeps are performed: (ii.1) Randomly modify $(100 - s)\%$ of the nonzero entries in $\boldsymbol{\Theta}^{h}$ to be zero. (ii.2) Randomly modify the same number of zero entries in $\boldsymbol{\Theta}^{h}$ to be nonzero. This is to ensure that each $\boldsymbol{\Theta}_i$ has the same sparsity as $\boldsymbol{\Theta}^{h}$. Each of these nonzero entries is sampled from $Uniform(D)$. (ii.3) For the remaining unmodified nonzero entries in $\boldsymbol{\Theta}^{h}$, resample their values from $Uniform(D)$ and add the resulting value of each nonzero entry with the entry in $\boldsymbol{\Theta}^{h}$ at the same row and column. This is to ensure $\mathrm{E}(\boldsymbol{\Theta}_i) = v\boldsymbol{\Theta}^{h}$. An example for this step is given in Fig. 4-4.

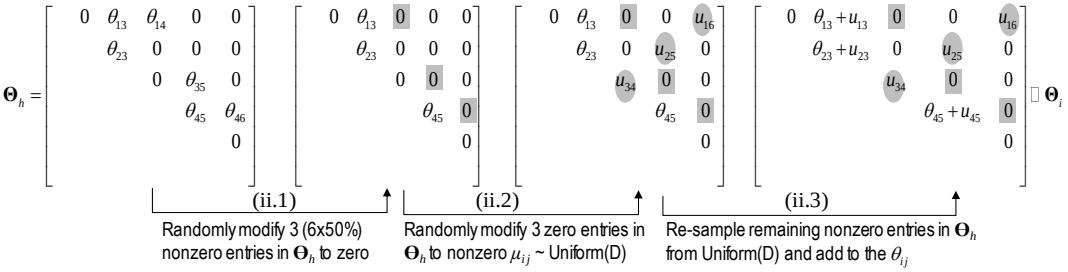


Fig. 4-4 Substeps for constructing the IC matrix of each task, $\boldsymbol{\Theta}_i$, from $\boldsymbol{\Theta}^{h}$(s%=50%).

It can be seen that $s\%$ reflects how much the IC matrix of each task, $\boldsymbol{\Theta}_i$, is related to $\boldsymbol{\Theta}^{h}$. The higher the $s\%$, the more the relatedness. Considering the extreme case when $s\% = 100\%$, the $\boldsymbol{\Theta}_i$ constructed following substeps (ii.1)-(ii.3) will have the same

positions of nonzero entries as $\boldsymbol{\Theta}^h$, so the GGM corresponding to $\boldsymbol{\Theta}_i$ will look the same as that corresponding to $\boldsymbol{\Theta}^h$. When $s\% = 0$, the $\boldsymbol{\Theta}_i$ will have completely different positions of nonzero entries from $\boldsymbol{\Theta}^h$, so their GGMs will not share even a single arc. Through relating to $\boldsymbol{\Theta}^h$, the IC matrices of all the tasks are related, so $s\%$ may be considered as an indirect measure of task relatedness. It makes sense to specify the relatedness of $\boldsymbol{\Theta}_i$ to $\boldsymbol{\Theta}^h$, rather than specifying the relatedness between the $\boldsymbol{\Theta}_i$'s directly, because transfer learning between tasks is through $\boldsymbol{\Theta}^h$, the common prior shared by all the tasks.

(iii) Generate simulation data for each task. A dataset consisting of n independently and identically distributed (IID) observations is generated from a multivariate normal distribution with mean zero and IC matrix $\boldsymbol{\Theta}_i$ for task i, $i = 1, \cdots, m$.

(iv) Estimate the IC matrix of each task. We apply the transfer learning approach to the data and estimate the IC matrix of each task.

(v) Measure the performance. In order to apply the transfer learning approach, we need to choose values for λ and υ. To assess the overall performance of transfer learning across different choices for λ and υ, ROC curves are employed. An ROC curve plots the numbers of true positives vs. false positives over all possible choices for λ and υ. Here, we count one true positive if a nonzero entry in the true IC matrix is also nonzero in the estimated IC matrix; we count one false positive if a zero entry in the true IC matrix is nonzero in the estimated IC matrix. Thus, the number of true positives (Y axis of the ROC curve) and the number of false positives (X axis of the ROC curve) reflects the power of algorithm in detecting non-zero entries and the false alarm error, respectively.

Note that execution of steps (iii)-(iv) once will generate one ROC curve for each task. To reduce sampling variation, the two steps can be repeated for N times and a mean ROC curve can be generated.

In this section, we compare the performance of the proposed transfer learning approach with the single task learning approach based on graphical LASSO, i.e., (2). Following a similar procedure to steps (i)-(v), a mean ROC curve can be obtained for each task for single task learning. Based on the definition of ROC curves, a learning approach is better if its ROC curve is closer to the upper left corner of the plot.

Figs. 4-5 to 4-7 compare transfer learning and single task learning based on the mean ROC curves for the first task (other tasks show similar patterns and are not shown here due to space limits). The comparison is across various parameter settings, including the number of variables ($p = 50,100,200$), the number of related tasks ($m = 2,5,10$), and task relatedness ($s\% = 90\%, 50\%, 0\%$). Small sample sizes are assumed for each task, so the sample size n is set to be equal to p. Also, the number of non-zero entries in $\mathbf{\Theta}^h$ is set to be equal to p, such that the GGM in each task is sparse.

The following observations can be obtained:

- The advantage of transfer learning over single task learning is more significant when the tasks are more related. When the tasks are little related (e.g., Fig. 4-7), the necessity of using transfer learning is not obvious.
- The advantage of transfer learning over single task learning is more significant when there are more related tasks.
- The performances of both transfer learning and single task learning degrade as the network becomes larger (i.e., larger p). However, the performance of transfer learning may be improved by having more related tasks to compensate for sample shortage, whereas more related tasks do not help

single task learning.

Computational efficiency: The proposed transfer learning algorithm is very fast. This is because the efficient BCD algorithm is used in the M step of the proposed EM algorithm. Also, the proposed EM algorithm usually takes only 3~6 iterations to converge. Table 4-1 shows the CPU time of MTL of the simulation studies.

Table 4-1: CPU time (in seconds) of the simulation studies of MTL

	p=50	p=100	p=200
m=2	6.34	25.13	102.35
m=5	15.98	65.57	225.85
m=10	36.12	140.76	562.74

For a comparison, Table 4-2 shows the corresponding CPU time of STL of the simulation studies.

Table 4-2: CPU time (in seconds) of the simulation studies of STL

	p=50	p=100	p=200
m=2	2.87	9.34	24.98
m=5	5.21	29.88	73.47
m=10	11.35	52.19	101.53

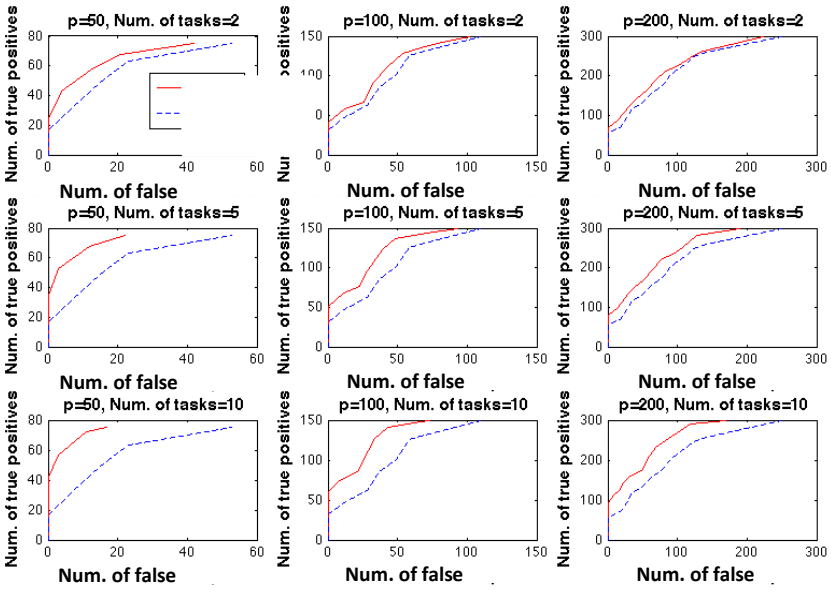


Fig. 4-5 Mean ROC curves for transfer learning (red solid curve) and single task learning (blue dash curve) with task relatedness $s\% = 90\%$

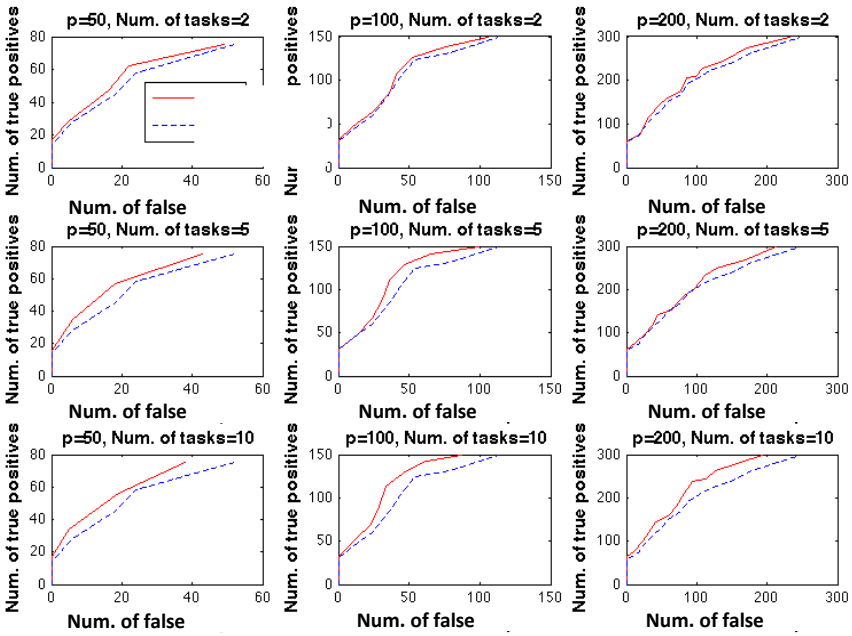


Fig. 4-6 Mean ROC curves for transfer learning (red solid curve) and single task learning (blue dash curve) with task relatedness $s\% = 50\%$

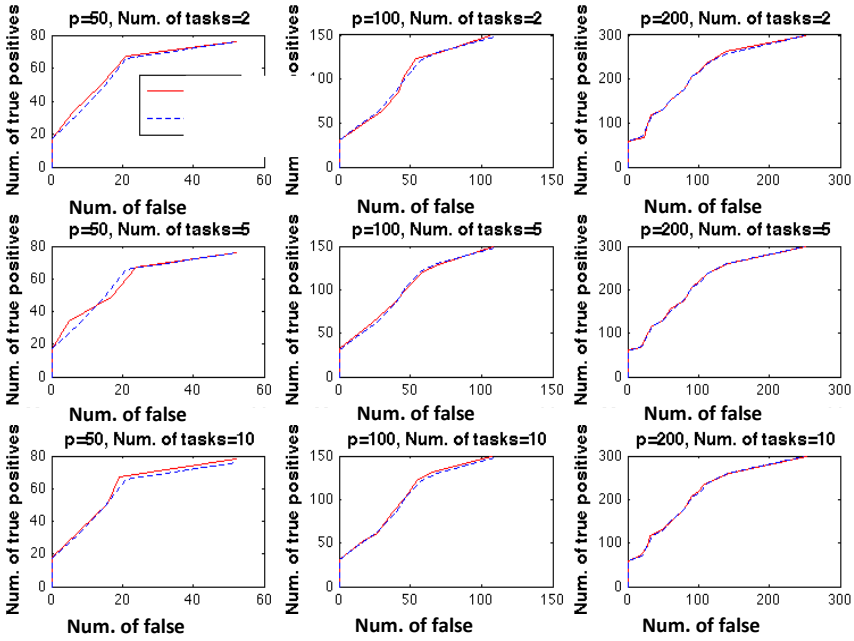


Fig. 4-7 Mean ROC curves for transfer learning (red solid curve) and single task learning (blue dash curve) with task relatedness $s\% = 0\%$

4.6 Application in brain connectivity network modeling of AD

The simulation study in Section 4-5 has demonstrated effectiveness of the proposed transfer learning approach. In this section we explore the application to brain connectivity modeling of AD. AD is a fatal, neurodegenerative disorder currently affecting over five million Americans. The existing knowledge about AD is very limited and clinical diagnosis is imprecise. Recent studies have found that AD is closely related to alteration in the brain connectivity network (Stam et al., 2007; Supekar et al., 2008). Identification of brain connectivity patterns common to AD patients provides the potential for identifying AD biomarkers to help clinical diagnosis. To explore this goal, we apply the proposed transfer learning approach to the fMRI data of 15 AD patients, which are considered as 15 related tasks, and learn a brain connectivity network for each patient. To provide a contrast for the AD connectivity networks, the networks of 16 NCs, who are considered as another set of related tasks, are also learned by the proposed transfer learning approach.

The selection criteria for the 15 AD patients and 16 NCs are as follows. The AD patients are aged between 53~79, right-handed, free of other diseases such as stroke and focal pathology, and with MMSE scores between 0-20. MMSE is a clinical instrument for cognitive assessment; the lower the score the more severe the dementia. The NCs are selected to purposely match the AD patients in terms of other selection criteria except MMSE scores, i.e., they are in the same age cohort as the AD patients, right-handed, free of other diseases, but with MMSE scores between 27-30 (i.e., they do not have dementia).

The fMRI data of each of the 31 subjects at his/her resting state was obtained using the 3-Tesla Siemens whole-body MRI system at Tiantan Hospital in Beijing, China. We apply the following preprocessing steps to the data. First, we apply the Automated Anatomical Labeling (AAL) technique (Tzourio-Mazoyer, 2002) to segment the whole

brain of each subject into 116 regions. 42 regions, whose names are given in Table 4-2, are selected as they have been considered to be potentially related to AD in the literature. These regions distribute in the four major neocortical lobes of the brain, i.e., the frontal, parietal, occipital, and temporal lobes. Next, within each selected region, the voxel-wise fMRI time courses are averaged into one regional average time course. Then, the first five data points for each regional average time course are discarded. Finally, each regional average time course is detrended. After the preprocessing, the dataset of each subject takes the form of a 245×42 matrix (245 sampling points in each regional average fMRI time course and 42 selected brain regions).

Table 4-2: Names of the brain regions selected for brain connectivity network modeling ("L" means that the brain region is located at the left hemisphere; "R" means right hemisphere.)

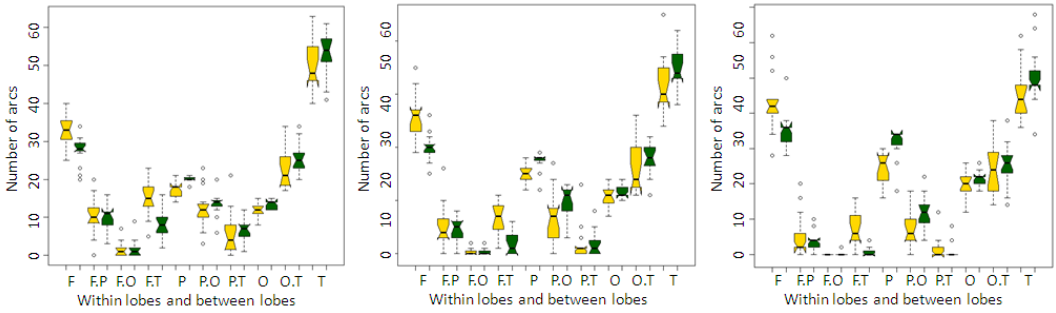
Prefrontal lobe		Parietal lobe		Occipital lobe		Temporal lobe	
1	Frontal_Sup_L	13	Parietal_Sup_L	21	Occipital_Sup_L	27	Temporal_Sup_L
2	Frontal_Sup_R	14	Parietal_Sup_R	22	Occipital_Sup_R	28	Temporal_Sup_R
3	Frontal_Mid_L	15	Parietal_Inf_L	23	Occipital_Mid_L	29	Temporal_Pole_Sup_L
4	Frontal_Mid_R	16	Parietal_Inf_R	24	Occipital_Mid_R	30	Temporal_Pole_Sup_R
5	Frontal_Sup_Medial_L	17	Precuneus_L	25	Occipital_Inf_L	31	Temporal_Mid_L
6	Frontal_Sup_Medial_R	18	Precuneus_R	26	Occipital_Inf_R	32	Temporal_Mid_R
7	Frontal_Mid_Orb_L	19	Cingulum_Post_L			33	Temporal_Pole_Mid_L
8	Frontal_Mid_Orb_R	20	Cingulum_Post_R			34	Temporal_Pole_Mid_R
9	Rectus_L					35	Temporal_Inf_L 8301
10	Rectus_R					36	Temporal_Inf_R 8302
11	Cingulum_Ant_L					37	Fusiform_L
12	Cingulum_Ant_R					38	Fusiform_R
						39	Hippocampus_L
						40	Hippocampus_R
						41	ParaHippocampal_L
						42	ParaHippocampal_R

The datasets obtained through the aforementioned steps can be reasonably considered to follow normal distributions, because each brain region includes at least several hundreds of voxels and the measurement data for each region is a regional average over the belonging voxels and thus the Central Limit Theorem applies. Also, the

normality assumption is common for fMRI studies (Valdes-Sosa et al., 2005 and Worsley, et al., 1997). We apply the transfer learning approach to the datasets of the 15 AD patients, which produces 15 GGMs. Similarly, GGMs are learned for the 16 NCs. Note that in applying the transfer learning approach, the values for the regularization parameters, λ_i's, need to be selected. In this paper, we focus on comparing AD and NCs in terms of the distribution/organization of the connectivity in the brain, which has been less studied in the literature, but not in terms of the global scale of the connectivity, which has been studied substantially. To achieve this, we must factor out the connectivity difference between AD and NCs that is due to their difference at the global scale, so that the remaining difference will reflect their difference in the connectivity distribution/organization. A common strategy is to control the total number of arcs in the AD and NCs connectivity networks to be the same, which has been adopted by a number of other studies (Supekar et al., 2008; Stam et al., 2007). We also adopt this strategy; specifically, we adjust the λ_i's in estimating the 15 AD connectivity networks and those in estimating the 16 NCs connectivity networks, such that all these networks have the same total number of arcs. Also, by selecting different values for the total number of arcs, we can obtain connectivity networks at different strength levels. Specifically, given a small value for the total number of arcs, only strong arcs will show up in the resulting connectivity networks; when increasing the total number of arcs, mild (or even weak) arcs will also show up in the resulting connectivity networks.

Comparison of AD and NCs in terms of connectivity distribution/organization can be achieved by comparing them in terms of the numbers of arcs within each lobe as well as between each pair of lobes. To be able to assess the statistical significance of the comparison, the 15 (16) GGMs are treated as "samples" of the AD (NCs) brain connectivity network. Based on these samples, we generate boxplots for the numbers of

arcs within and between lobes for AD and NCs in Fig. 4-8. Furthermore, statistical significance of the observed difference between AD and NCs in the boxplots is assessed by hypothesis testing. The P-values of the hypothesis testing are given in Table 4-3.



(a) Total num. of arcs=180 (b) Total num. of arcs=120 (c) Total num. of arcs=90

Fig. 4-8 Boxplots for numbers of arcs within and between lobes in learned connectivity networks by transfer learning (F: frontal, P: parietal, O: occipital, T: temporal; yellow: AD, green: NCs)

Table 4-3 P-values for hypothesis testing on AD vs. NCs comparison based on learned connectivity networks by transfer learning

(a) Total num. of arcs=180

	Frontal	Parietal	Occipital	Temporal
Frontal	0.0009	0.7604	1	0.0001
Parietal		0.0002	0.5257	0.4753
Occipital			0.0125	0.0829
Temporal				0.2143

(b) Total num. of arcs=120

	Frontal	Parietal	Occipital	Temporal
Frontal	0.0001	0.546	0.6229	0.0000
Parietal		0.0003	0.0466	0.9005
Occipital			0.1153	0.2314
Temporal				0.0657

(c) Total num. of arcs=90

	Frontal	Parietal	Occipital	Temporal
Frontal	0.0033	0.2924	0.3343	0.0005
Parietal		0.0002	0.1331	0.6983
Occipital			0.1158	0.621
Temporal				0.1282

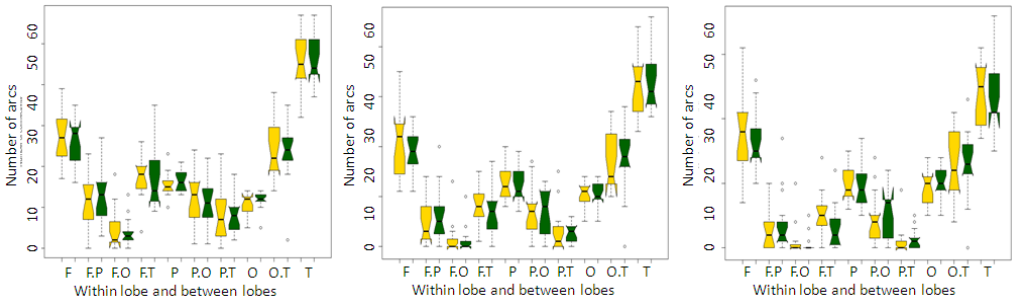
The following interesting findings are obtained:

- The parietal lobe of AD has significantly less connectivity than NCs. Loss of connectivity in the parietal lobe of AD has been reported by a number of other studies in the literature (Langbaum, et al., 2009; Chen, et al., 2010).
- The regions well-known to be attacked by AD the first and severely are "Hippocampus_L&R" in the temporal lobe, which play an important role in memory. We perform hypothesis testing on the difference between AD and NCs in terms of the number of arcs between "Hippocampus_L&R" and other

regions in the temporal lobe, and find the P-values to be 0.0526, 0.0111, and 0.0089 for total numbers of arcs equal to 180, 120, and 90, respectively. This indicates significant loss of connectivity between "Hippocampus_L&R" and other regions in AD, which is consistent with quite a few other studies in the literature (Supekar et al., 2008; Wang et al., 2007).

- The frontal lobe of AD has significantly more connectivity than NCs. This is consistent with the previous literature and has been interpreted as compensatory reallocation or recruitment of cognitive resources (Gould et al., 2006; Stern, 2006). Because regions in the frontal lobe are typically less affected by AD, increase of connectivity in the frontal lobe may help preserve some cognitive functions in AD patients. Same explanation may be applied to the increase of connectivity between the frontal and temporal lobes in AD.
- All the above findings are consistent across different total numbers of arcs in the connectivity networks.

For comparison purposes, we also apply graphical LASSO to the same datasets. Please note that graphical LASSO is a single task learning approach which learns each task independently. Boxplots and P-value tables similar to those in Fig. 4-8 and Table 4-3 are generated, as shown in Fig. 4-9 and Table 4-4. It can be seen that the single task learning fails to identify any significant difference between AD and NCs. The reason, as revealed by the boxplots, is that there is large variability in the 15 (16) brain connectivity networks of AD (NCs). This is because the single task learning is not able to make use of task relatedness to compensate for sample size shortage. As a result, the learned connectivity networks may not be reliable.



(a) Total num. of arcs=180 (b) Total num. of arcs=120 (c) Total num. of arcs=90

Fig. 4-9 Boxplots for numbers of arcs within and between lobes in learned connectivity networks by single task learning (F: frontal, P: parietal, O: occipital, T: temporal; yellow: AD, green: NCs)

Table 4-4 P-values for hypothesis testing on AD vs. NCs comparison based on learned connectivity networks by single task learning

(a) Total num. of arcs=180

	Frontal	Parietal	Occipital	Temporal
Frontal	0.4264	0.5497	0.644	0.9579
Parietal		0.3953	0.6733	0.9006
Occipital			0.2215	1
Temporal				0.874

(b) Total num. of arcs=120

	Frontal	Parietal	Occipital	Temporal
Frontal	0.4276	0.5481	0.7669	0.3731
Parietal		0.9224	0.9041	0.8764
Occipital			0.66	0.7984
Temporal				0.8111

(c) Total num. of arcs=90

	Frontal	Parietal	Occipital	Temporal
Frontal	Frontal	0.6949	0.4827	0.5587
Parietal	Parietal		1	0.5018
Occipital	Occipital			0.4005
Temporal	Temporal			

4.7 Conclusion

This paper proposed a transfer learning approach for jointly learning GGMs of multiple related tasks. The proposed approach adopted the BHM framework in the problem formulation and considered IC matrices of the tasks to be samples drawn from the same Wishart distribution. An L_1-regularization was further added to the problem formulation to impose sparsity on the GGMs estimation. Under this problem formulation, a method based on the EM algorithm was further developed to learn GGMs from data. Simulation studies showed that the transfer learning approach performs better than single task learning. This advantage is more substantial when there are more related tasks or when the tasks are more related to each other. Also, the proposed transfer learning approach was applied to the fMRI data of 15 AD patients and 16 NCs for brain connectivity

network identification. Comparison between the connectivity networks of AD and NCs revealed that AD is associated with decreased connectivity in the parietal lobe and between hippocampus and other regions in the temporal lobe, and increased connectivity in the frontal lobe and between the frontal and temporal lobes. All these findings are consistent with AD pathology and existing findings in the literature.

Transfer learning of other types of network models is also of great interest, such as directed models (also known as Bayesian networks) and models of non-Gaussian variables. For example, Bayesian networks have been used to characterize the directional effect of one brain region on another. In addition, from practical data analysis point of view, it is of interest to examine if the identified patterns will still be present given small perturbation to the data. Specifically, we may create some perturbed datasets out of the original dataset by resampling or adding noisy to the original data measurement, and then apply the proposed approach to the perturbed datasets and look for patterns that consistently occur across these datasets. This is also a way of providing assurance for the robustness of the results against sampling variability. We will investigate these methodological and practical issues in future work.

Appendix

I. Proof of Proposition 1:

Using the Bayes' rule,

$$P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}\right) = P\left(\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}|\boldsymbol{\Theta}^h\right)P(\boldsymbol{\Theta}^h)/P\left(\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}\right) \quad .$$

Furthermore, because $P\left(\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}|\boldsymbol{\Theta}^h\right) = \prod_{i=1}^m P\left(\boldsymbol{\Theta}_i^t|\boldsymbol{\Theta}^h\right)$ and $P(\boldsymbol{\Theta}^h)$ is a constant,

$$P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}\right) \propto \prod_{i=1}^m P\left(\boldsymbol{\Theta}_i^t|\boldsymbol{\Theta}^h\right)/P\left(\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\ldots,m}\right) \quad \text{(A-1)}$$

So, the first step is to derive $P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right)$. To do this, we condition $P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right)$ on $\boldsymbol{\Theta}^h$, i.e.,

$$P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right) = \int_{\boldsymbol{\Theta}^h} P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\middle|\boldsymbol{\Theta}^h\right) P(\boldsymbol{\Theta}^h)\, d\boldsymbol{\Theta}^h \propto \int_{\boldsymbol{\Theta}^h} \prod_{i=1}^m P\left(\boldsymbol{\Theta}_i^t\middle|\boldsymbol{\Theta}^h\right) d\boldsymbol{\Theta}^h \tag{A-2}$$

Because $\boldsymbol{\Theta}_i^t \sim Wishart_v\left(\boldsymbol{\Theta}^h\right)$,

$$P\left(\boldsymbol{\Theta}_i^t\middle|\boldsymbol{\Theta}^h\right) \propto \left|\boldsymbol{\Theta}^h\right|^{-v/2}\left|\boldsymbol{\Theta}_i^t\right|^{(v-p-1)/2} \; e^{-tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i^t\right)/2}, \tag{A-3}$$

Inserting (A-3) into (A-2),

$$P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right) \propto \int_{\boldsymbol{\Theta}^h} \prod_{i=1}^m \left\{\left|\boldsymbol{\Theta}^h\right|^{-v/2}\left|\boldsymbol{\Theta}_i^t\right|^{(v-p-1)/2} \; e^{-tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i^t\right)/2}\right\} d\,\boldsymbol{\Theta}^h$$

$$= \frac{\prod_{i=1}^m \left|\boldsymbol{\Theta}_i^t\right|^{\frac{v-p-1}{2}}}{\left|\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}}} \int_{\boldsymbol{\Theta}^h} \left|\boldsymbol{\Theta}^h\right|^{-\frac{(mv-p-1)+p+1}{2}} \left|\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}} e^{-\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right)} d\,\boldsymbol{\Theta}^h\,. \tag{A-4}$$

Note that the function inside the integral in (A-4) happens to be proportional to the density function of an inverse-Wishart distribution with degrees of freedom $(mv-p-1)$ and scale matrix $\sum_{i=1}^m \boldsymbol{\Theta}_i^t$. So, the integral in (A-4) is actually a constant. Then, (A-4) can be simplified into

$$P\left(\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right) \propto \frac{\prod_{i=1}^m \left|\boldsymbol{\Theta}_i^t\right|^{\frac{v-p-1}{2}}}{\left|\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}}} \tag{A-5}$$

Inserting (A-5) and (A-3) into (A-1),

$$P\left(\boldsymbol{\Theta}^h\middle|\left\{\boldsymbol{\Theta}_i^t\right\}_{i=1,2,\dots,m}\right) \propto \frac{\left|\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}}}{\prod_{i=1}^m \left|\boldsymbol{\Theta}_i^t\right|^{\frac{v-p-1}{2}}} \prod_{i=1}^m \left\{\left|\boldsymbol{\Theta}^h\right|^{-v/2}\left|\boldsymbol{\Theta}_i^t\right|^{(v-p-1)/2} \; e^{-tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i^t\right)/2}\right\}$$

$$= \left|\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}} \left|\boldsymbol{\Theta}^h\right|^{-\frac{mv}{2}} \; e^{-\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\sum_{i=1}^m \boldsymbol{\Theta}_i^t\right)}, \tag{A-6}$$

which happens to be proportional to the density function of an inverse-Wishart distribution with degrees of freedom $(mv-p-1)$ and scale matrix $\sum_{i=1}^m \boldsymbol{\Theta}_i^t$. □

II. Proof of Proposition 2:

Inserting (9) into (10), $Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = \int_{\boldsymbol{\Theta}^h}\sum_{i=1}^m\left\{\frac{v-p-1+n_i}{2}log|\boldsymbol{\Theta}_i| - \frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right) - \frac{n_i}{2}tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1\right\}P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)d\boldsymbol{\Theta}^h$. Some terms on the right-hand side of this equation do not include $\boldsymbol{\Theta}^h$, so they can be moved outside the integral, i.e.,

$$Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = \sum_{i=1}^m\left\{\frac{v-p-1+n_i}{2}log|\boldsymbol{\Theta}_i| - \frac{n_i}{2}tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1\right\} - \int_{\boldsymbol{\Theta}^h}\sum_{i=1}^m\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right)P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)d\boldsymbol{\Theta}^h \quad \text{(A-7)}$$

It can be seen from (A-7) that the key in getting $Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)$ is the find the integral $\int_{\boldsymbol{\Theta}^h}\sum_{i=1}^m\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right)P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)d\boldsymbol{\Theta}^h$. Because $tr(\cdot)$ is a linear operator, this integral becomes

$$\int_{\boldsymbol{\Theta}^h}\sum_{i=1}^m\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\boldsymbol{\Theta}_i\right)P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)d\boldsymbol{\Theta}^h = \sum_{i=1}^m\frac{1}{2}tr(\mathbf{D}\boldsymbol{\Theta}_i), \quad \text{(A-8)}$$

where $\mathbf{D} = \int_{\boldsymbol{\Theta}^h}\boldsymbol{\Theta}^{h^{-1}}P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right)d\boldsymbol{\Theta}^h$. Now we need to derive **D**. Specifically, according to Proposition 1, $P\left(\boldsymbol{\Theta}^h|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = \frac{\left|\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}}|\boldsymbol{\Theta}^h|^{-\frac{mv}{2}}\ e^{-\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right)}}{2^{\frac{(mv-p-1)p}{2}}\Gamma_p\left(\frac{mv-p-1}{2}\right)}$. Also, $d\boldsymbol{\Theta}^h = \left|\boldsymbol{\Theta}^h\right|^{p+1}d\boldsymbol{\Theta}^{h^{-1}}$. Therefore, **D** becomes

$$\mathbf{D} = \int_{\boldsymbol{\Theta}^{h^{-1}}}\boldsymbol{\Theta}^{h^{-1}}\frac{\left|\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right|^{\frac{(mv-p-1)}{2}}\left|\boldsymbol{\Theta}^{h^{-1}}\right|^{\frac{mv-p-1-p-1}{2}}\ e^{-\frac{1}{2}tr\left(\boldsymbol{\Theta}^{h^{-1}}\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right)}}{2^{\frac{(mv-p-1)p}{2}}\Gamma_p\left(\frac{mv-p-1}{2}\right)}d\boldsymbol{\Theta}^{h^{-1}}, \quad \text{(A-9)}$$

which happens to be the mean of a Wishart distribution for $\boldsymbol{\Theta}^{h^{-1}}$ with degrees of freedom equal to $mv-p-1$ and scale matrix $\left(\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right)^{-1}$, i.e.,

$$\mathbf{D} = (mv-p-1)\left(\sum_{i=1}^m\boldsymbol{\Theta}_i^t\right)^{-1} \quad \text{(A-10)}$$

Furthermore, inserting (A-10) into (A-8) and then into (A-7),

$$Q\left(\{\boldsymbol{\Theta}_i\}_{i=1,2,\dots,m}|\{\boldsymbol{\Theta}_i^t\}_{i=1,2,\dots,m}\right) = \sum_{i=1}^{m}\left\{\frac{v-p-1+n_i}{2} log|\boldsymbol{\Theta}_i| - \frac{n_i}{2} tr(\mathbf{S}_i\boldsymbol{\Theta}_i) - \lambda_i\|\boldsymbol{\Theta}_i\|_1 - \frac{1}{2} tr\left((mv-p-1)\left(\sum_{i=1}^{m}\boldsymbol{\Theta}_i^t\right)^{-1}\boldsymbol{\Theta}_i\right)\right\}. \quad \square$$

III. BCD algorithm for solving the optimization in (12) and proof of its convergence

The basic idea of the BCD algorithm is to update each column (or row) of $\boldsymbol{\Theta}_i$ iteratively while fixing all other columns (or rows), until convergence. For notation simplicity, we change the notations of (12) in the following way: We drop the subscript "i" which is the index for the tasks, since we will apply the BCD algorithm to all the tasks in the same way. Also, we drop the superscripts "t" and "$t+1$" which represent the t-th iteration of the EM algorithm, because we will apply the BCD algorithm in every iteration of the EM algorithm. Therefore, (12) becomes

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\{log|\boldsymbol{\Theta}| - tr(\mathbf{S}\boldsymbol{\Theta}) - \lambda' \|\boldsymbol{\Theta}\|_1\}, \qquad \text{(A-11)}$$

where $\boldsymbol{\Theta}^*$ denotes the optimal solution to $\boldsymbol{\Theta}$. In what follows, we will show how to apply the BCD algorithm to solve for (A-11).

Specifically, because the BCD algorithm works by iterations, we will only illustrate the steps in one iteration and other iterations work in a similar way. At a certain iteration, we first need to partition the current $\boldsymbol{\Theta}$ as follows. Let $\boldsymbol{\Theta}_{\backslash j\backslash j}$ be the matrix produced by removing row j and column j from $\boldsymbol{\Theta}$, θ_{jj} be the element at row j and column j of $\boldsymbol{\Theta}$, and $\boldsymbol{\Theta}_j$ be the column j of $\boldsymbol{\Theta}$ with θ_{jj} removed. Then, $\boldsymbol{\Theta}$ can be partitioned as $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_{\backslash j\backslash j} & \boldsymbol{\Theta}_j \\ \boldsymbol{\Theta}_j^T & \theta_{jj} \end{bmatrix}$, and correspondingly $\mathbf{S}$ can be partitioned as $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\backslash j\backslash j} & \mathbf{S}_j \\ \mathbf{S}_j^T & s_{jj} \end{bmatrix}$. Next, we want to update $\boldsymbol{\Theta}_j$ and θ_{jj} while holding other elements in $\boldsymbol{\Theta}$ constant. To do this, let f represent the objective function in (A-7), i.e., $f = log|\boldsymbol{\Theta}| - tr(\mathbf{S}\boldsymbol{\Theta}) - \lambda'\|\boldsymbol{\Theta}\|_1$; take

the partial derivatives of f with respect to $\boldsymbol{\Theta}_j$ and θ_{jj}, respectively; and then make the partial derivatives to be zero, i.e.,

$$\frac{\partial f}{\partial \boldsymbol{\Theta}_j} = -\frac{2}{\theta_{jj} - \boldsymbol{\Theta}_j^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \boldsymbol{\Theta}_j} \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \boldsymbol{\Theta}_j - \mathbf{S}_j - \lambda' SGN(\boldsymbol{\Theta}_j) = 0, \text{ and} \quad \text{(A-12)}$$

$$\frac{\partial f}{\partial \theta_{jj}} = \frac{1}{\theta_{jj} - \boldsymbol{\Theta}_j^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \boldsymbol{\Theta}_j} - s_{jj} - \lambda' = 0, \quad \text{(A-13)}$$

where $SGN(\boldsymbol{\Theta}_j)$ denotes the partial derivative of $\|\boldsymbol{\Theta}\|_1$ with respect to $\boldsymbol{\Theta}_j$. It is difficult to solve for $\boldsymbol{\Theta}_j$ and θ_{jj} from (A-12) and (A-13) directly. Therefore, we adopt the following strategies.

Letting $\mathbf{a} = -\frac{\boldsymbol{\Theta}_j}{\theta_{jj} - \boldsymbol{\Theta}_j^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \boldsymbol{\Theta}_j}$, then (A-12) and (A-13) become

$$2\boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{a} - \mathbf{S}_j + \lambda' SGN(\mathbf{a}) = 0 \quad \text{(A-14)}$$

$$\mathbf{a} = -(s_{jj} + \lambda')\boldsymbol{\Theta}_j. \quad \text{(A-15)}$$

It is clear that (A-14) is also the result of making the partial derivative of g with respect to $\mathbf{a}$ to be zero in the following optimization problem:

$$\min_{\mathbf{a}} g = \mathbf{a}^T \boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \mathbf{a} - \mathbf{S}_j^T \mathbf{a} + \lambda' \|\mathbf{a}\|_1, \quad \text{(A-16)}$$

which is equivalent to the following min-max problem:

$$\max_{\boldsymbol{\kappa}} \min_{\mathbf{a}} g = 2\left(-\frac{1}{2}\boldsymbol{\kappa}^T \Theta_{\backslash j \backslash j} \boldsymbol{\kappa} + \boldsymbol{\kappa}^{\mathrm{T}} \mathbf{a}\right) - \mathbf{S}_j^T \mathbf{a} + \lambda' \|\mathbf{a}\|_1. \quad \text{(A-17)}$$

This min-max problem can be solved by the prox method proposed by (Nemirovski, 2005).

After $\mathbf{a}$ and $\boldsymbol{\kappa}$ are obtained, (A-15) can be used to find $\boldsymbol{\Theta}_j$, i.e., $\boldsymbol{\Theta}_j = -\frac{\mathbf{a}}{s_{jj} + \lambda'}$. Furthermore, based on (A-13), θ_{jj} can be obtained, i.e., $\theta_{jj} = \frac{(-\mathbf{a}^T \boldsymbol{\kappa} + \mathbf{1})}{s_{jj} + \lambda'}$.

Proof of convergence: According to (Tseng, 2001), the BCD algorithm converges if and only if (A-16) has a unique solution of $\mathbf{a}$ at each iteration. The unique solution is guaranteed if the optimization problem in (A-16) is strictly convex. The strict convexity is true if $\boldsymbol{\Theta}_{\backslash j \backslash j}$ is positive definite, denoted by $\boldsymbol{\Theta}_{\backslash j \backslash j} \succ 0$. $\boldsymbol{\Theta}_{\backslash j \backslash j} \succ 0$ if $\boldsymbol{\Theta} \succ 0$. Therefore, the key to prove the convergence of the BCD algorithm is to prove $\boldsymbol{\Theta} \succ 0$. Recall that the BCD algorithm works by iterations and (A-16) needs to be solved at each iteration. As a

result, we need to prove $\boldsymbol{\Theta} \succ 0$ at each iteration. To achieve this, we use mathematical induction, which includes a basis step and an inductive step:

Let ${}^{j}\boldsymbol{\Theta}$ be the $\boldsymbol{\Theta}$ obtained at the j-th iteration of the BCD algorithm.

Basis step: Because ${}^{0}\boldsymbol{\Theta}$ can be chosen by the user, ${}^{0}\boldsymbol{\Theta}$ is guaranteed to satisfy ${}^{0}\boldsymbol{\Theta} \succ 0$.

Inductive step: Assuming ${}^{j-1}\boldsymbol{\Theta} \succ 0$, we need to prove ${}^{j}\boldsymbol{\Theta} \succ 0$, which is equivalent to prove $\left| {}^{j}\boldsymbol{\Theta} \right| > 0$. Because $\left| {}^{j}\boldsymbol{\Theta} \right| = \left| {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j} \right| \left({}^{j}\theta_{jj} - {}^{j}\boldsymbol{\Theta}_{j}^{T} \; {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \; {}^{j}\boldsymbol{\Theta}_{j} \right)$, we need to prove $\left| {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j} \right| > 0$ and $\left({}^{j}\theta_{jj} - {}^{j}\boldsymbol{\Theta}_{j}^{T} \; {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \; {}^{j}\boldsymbol{\Theta}_{j} \right) > 0$. $\left| {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j} \right| > 0$ is true since we have assumed ${}^{j-1}\boldsymbol{\Theta} \succ 0$. Based on (A-10), ${}^{j}\theta_{jj} - {}^{j}\boldsymbol{\Theta}_{j}^{T} \; {}^{j-1}\boldsymbol{\Theta}_{\backslash j \backslash j}^{-1} \; {}^{j}\boldsymbol{\Theta}_{j} = 1/\left(s_{jj} + \lambda\right) > 0$. As a result, $\left| {}^{j}\boldsymbol{\Theta} \right| > 0$ and ${}^{j}\boldsymbol{\Theta} \succ 0$. □

Reference

[1] Bakker, B.; and Heskes, T. "Task Clustering and Gating for Bayesian Multitask Learning". Journal of Machine Learning Research, 4:83–99, 2003.

[2] Baxter, J. "A Model of Inductive Bias Learning". Journal of Artificial Intelligence Research, 2000.

[3] Bickel, P. J.; and Levina, E. ''Regularized Estimation of Large Covariance Matrices,'' Annals of Statistics, 36, 199–227, 2008.

[4] Caruana, R. "Multitask learning". Machine Learning, 28:41–75, 1997.

[5] Chen, K., Langbaum, J.B.S., Fleisher, A.S., Ayutyanont, N., Reschke, C., Lee, W., Liu, X., Bandy, D., Alexander, G.E., Thompson, P.M., Foster, N.L., Harvey, D.J., de Leon, M.J., Koeppe, R.A., Jagust, W.J., Weimer, M.W., Reiman, E.M., and the ADNI. "Twelve-month metabolic declines in probable Alzheimer's disease and amnestic mild cognitive impairment assessed using an empirically pre-defined statistical region-of-interest: Findings from the Alzheimer's Disease Neuroimaging Initiative," NeuroImage, 51, 645-664, 2010.

[6] Dellaert, F. "The Expectation Maximization Algorithm", Technical Report. GVU Center; College of Computing; Georgia Tech, GIT-GVU-02-20, 2002.

[7] Friedman, N.; Linial, M.; Nachman, I.; and Pe'er, D. "Using Bayesian Networks to Analyze Expression Data". Journal of Computational Biology, 7, 601–620, 2000.

[8] Friedman, J.; Hastie, T.; and Tibshirani, R. "Sparse Inverse Covariance Estimation with the Graphical Lasso". Biostatistics, 8(1):1–10, 2007.

[9] Gould, R.L.; Arroyo, B.; Brown, R,G.; Owen, A.M.; Bullmore E.T.; and Howard, R.J. "Brain Mechanisms of Successful Compensation during Learning in Alzheimer Disease", Neurology 67, 1011-1017, 2006.

[10] Huang, S.; Li, J., et al. "Learning Brain Connectivity of Alzheimer's Disease by Sparse Inverse Covariance Estimation," NeuroImage, 50, 935-949, 2010.

[11] Lawrence, N.D.; and Platt, J.C. "Learning to Learn with the Informative Vector Machine". In Proceedings of the 21st International Conference on Machine Learning, 2004.

[12] Langbaum, J.B.S., Chen, K., Lee, W., Reschke, C., Bandy, D., Fleisher, A.S., Alexander, G.E., Foster, N.L., Weiner, M.W., Koeppe, R.A., Jagust, W.J., Reiman, E.M., and the ADNI. "Categorical and correlational analyses of baseline fluorodeoxyglucose positron emission tomography images from the Alzheimer's Disease Neuroimaging Initiative (ADNI)." NeuroImage, 45, 1107-1116, 2009.

[13] Levina, E.; Rothman, A. J.; and Zhu, J. ''Sparse Estimation of Large CovarianceMatrices via a Nested Lasso Penalty,'' Annals of Applied Statistics, 2, 245–263, 2008.

[14] Li, H.; and Gui, J. ''Gradient Directed Regularization for Sparse Gaussian Concentration Graphs, with Applications to Inference of Genetic Networks,'' Biostatistics, 7, 302–317, 2006.

[15] Li, J., and Shi, J. "Knowledge Discovery from Observational Data for Process Control through Causal Bayesian Networks," IIE Transactions, 39(6), 681-690, 2007.

[16] Meinshausen, N.; and Buhlmann, P. ''High Dimensional Graphs and Variable Selection with the Lasso,'' Annals of Statistics, 34, 1436–1462, 2006.

[17] Nemirovski, A. "Prox-method with Rate of Convergence o(1/t) for Variational Inequalities with lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems". SIAM Journal on Optimization, 15(1):229–251, 2005.

[18] Peng, J.; Wang, P.; Zhou, N.F.; and Zhu, J. "Partial Correlation Estimation by Joint Sparse Regression Models", Journal of the American Statistical Association, 104(486), 735-746, 2009.

[19] Ravikumar, P., Raskutti, G., Wainwright, M. J. and Yu, B. "Model selection in Gaussian graphical models: High-dimensional consistency of l1-regularized MLE." Advances in Neural Information Processing Systems (NIPS) 21, 2008.

[20] Schafer, J., and Strimmer, K. ''A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics,'' Statistical Applications in Genetics and Molecular Biology, 4 (1), Article 32, 2005.

[21] Stam, C.J.; Jones, B.F.; Nolte, G.; Breakspear, M.; and Scheltens, P. "Small-World Networks and Functional Connectivity in Alzheimer's Disease". Cerebral

Corter 17:92-99, 2007.

[22] Stern,Y. “Cognitive Reserve and Alzheimer Disease”, Alzheimer Disease Associated Disorder 20, 69-74, 2006.

[23] Supekar, K.; Menon, V.; Rubin, D.; Musen, M.; and Greicius, M.D. “Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer's Disease”. PLoS Comput Biol 4(6) 1-11, 2008.

[24] Sun, L.; Patel, R.; Liu, J.; Chen, K.; Wu, T.; Li, J.; Reiman, E.; and Ye, J. “Mining Brain Region Connectivity for Alzheimer’s Disease Study via Sparse Inverse Covariance Estimation”. Proceedings of Knowledge Discovery and Data Mining Conference (KDD), 2009.

[25] Thrun, S. and O'Sullivan, J.. “Discovering structure in multiple learning tasks: The TC algorithm.” In International Conference on Machine Learning, 489-497, 1996.

[26] Tibshirani, R. “Regression Shrinkage and Selection via the Lasso”. Journal of the Royal Statistical Society Series B, 58(1):267–288, 1996.

[27] Tseng, P. “Convergence of Block Coordinate Descent Method for Nondifferentiable Maximation”. J. Opt. Theory and Applications, 109(3):474–494, 2001.

[28] Tzourio-Mazoyer, N. and et al. “Automated Anatomical Labelling of Activations in SPM using a Macroscopic Anatomical Parcellation of the MNI MRI Single Subject Brain”. Neuroimage,15:273-289, 2002.

[29] Valdés-Sosa, P., Sánchez-Bornot, J., Lage-Castellanos, A., Vega-Hernández, M., Bosch-Bayard, J., Melie-García, L. and Canales-Rodríguez, E. “Estimating brain functional connectivity with sparse multivariate autoregression.” Philos. Trans. Roy. Soc. B: Biological Sciences 360 969–981, 2005.

[30] Wang, K.; Liang, M.; Wang, L., Tian, L.; Zhang, X.; Li, K.; and Jiang, T. “Altered Functional Connectivity in Early Alzheimer’s Disease: A Resting-State fMRI Study”. Human Brain Mapping 28, 967-978, 2007.

[31] Worsley, K.J., Poline, J.B., Friston, K.J., and Evans, A.C. “Characterizing the response of PET and fMRI data using multivariate linear models.” NeuroImage , 6 (4) 305 – 319, 1997.

[32] Wu, C.F.J. “On the Convergence Properties of the EM Algorithm”, The Annals of Statistics, 11, 95-103, 1983.

[33] Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research, 8, 35–63, 2007

[34] Yuan, M.; and Lin, Y. ‘‘Model Selection and Estimation in the Gaussian

Graphical Model,'' Biometrika, 94 (1), 19–35, 2007.

[35] Zhang, J.; Ghahramani, Z.; and Yang, Y. "Learning Multiple Related Tasks using Latent Independent Component Analysis". In Y. Weiss, B. Sch¨olkopf, and J. Platt, editors, Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA, 2006.

Chapter 5

MULTI-DATA FUSION FOR ENTERPRISE QUALITY IMPROVEMENT BY A MULTILEVEL LATENT RESPONSE MODEL

Abstract

Quality improvement of an enterprise needs a model to link multiple data sources, including the independent and interdependent activities of individuals in the enterprise, enterprise infrastructure, climate, and administration strategies, as well as the quality outcomes of the enterprise. This is a challenging problem because the data are at two levels, i.e., the individual and enterprise levels, and each individual's contribution to the enterprise quality outcome is usually not explicitly known. These challenges make general regression analysis and conventional multilevel models fall short. We propose a new multilevel model that treats individuals' contributions to the enterprise quality outcomes as latent variables. Under this new formulation, an algorithm is further developed to estimate the model parameters, which integrates the Fisher-scoring algorithm and generalized least squares estimation. Extensive simulation studies are performed, which demonstrate the superiority of the proposed model over the competing approach in terms of the statistical properties in parameter estimation. The proposed model is applied to a real-world application of nursing quality improvement, and helps identify key nursing activities and unit (a hospital unit is an enterprise in this context) quality-improving measures that help reduce patient falls.

5.1 Introduction

An enterprise consists of individuals. Examples of an enterprise include a company or a company division, such as the technology division and marketing division, consisting of belonging employees; a military team consisting of soldiers on a specific mission; and a hospital or a hospital unit, such as the emergency unit and surgical unit,

consisting of performing medical professionals. Quality improvement for an enterprise is different from quality improvement for an individual in terms of the following aspects:

1) Quality of an enterprise is affected by the independent and interdependent activities of the belonging individuals.
2) Quality of an enterprise is also affected by factors at the enterprise level such as the infrastructure, climate, and administration strategies.
3) Each individual contributes a certain portion to the quality of the enterprise; however, this individual contribution is usually immeasurable.

For example, consider a hospital unit as an enterprise. While the quality of a hospital unit has many dimensions, here we focus on one dimension – the quality of nursing in patient care, which can be measured by the numbers of falls and medication errors in the unit during a certain time period as well as patient satisfaction regarding the care provided by the unit. Because our focus is the quality of nursing, we consider the belonging individuals of the unit to be nurses, although a hospital unit also includes other medical professionals. It is not difficult to see that the aforementioned three aspects in the quality improvement of an enterprise fit perfectly in the context of nursing: 1) The quality of nursing in a unit is affected by the independent nursing activities of individual nurses as well as their interdependent activities in coordinating the patient care [1]. 2) The quality of nursing in a unit is also affected by unit infrastructure, climate, and administration strategies. 3) Each nurse contributes a certain portion to the nursing quality of the unit. However, this contribution is immeasurable because delivery of patient care requires team work and it is difficult to explicitly attribute the quality outcome of the unit to each individual nurse.

For quality improvement of an enterprise, three data sources may be utilized and they are at two levels. Specifically, at the enterprise level, there are two data sources on

quality outcomes and quality-affecting factors, such as enterprise infrastructure, climate, and administration strategies, respectively. At the individual level, there is one data source on individual independent and interdependent activities. To accomplish the goal of quality improvement of an enterprise, a statistical model is needed to link these multiple data sources together. The model should be able to address the questions of how individual activities affect the quality of the enterprise, how factors at the enterprise level affect the quality, and how individuals' activities/performance interact with the enterprise-level factors so as to jointly affect the enterprise quality. Answers to these questions are keys to formulating action plans for quality improvement of the enterprise.

To address the above questions, regression models provide a potential tool. Compared with many "black-box" methods which focus primarily on best prediction of the outcome variable, regression models are a "white-box" method offering better interpretability by explicitly revealing what variables lead to the best prediction for the outcome. This is especially important when the final objective is to formulate an action plan. Furthermore, parametric regression models allow for statistical inference in variable or model selection, which offers a rigorous way for generalizing the model estimated from a specific dataset to the population. Third, multilevel regression, as a specific type of regression models, was developed to handle multi-level data sources.

In the "language" of regression, the quality outcomes are called responses; the enterprise-level quality-affecting factors and the individual-level activities variables can both be called predictors. In general regression analysis, all the responses and predictors should be at the same level. Therefore, to be able to apply general regression analysis to our data, an intuitive approach is to transform the data sources to a single level. Specifically, since out of the three data sources, only the predictors of individual activities are at the individual level, we might consider aggregating these predictors to the

enterprise level by using some pre-defined summary statistics, such as the sample mean and variance of the data of all the individuals in each enterprise. Then, general regression analysis can be applied at the enterprise level. One drawback of this approach is loss of information, since the pre-defined summary statistics may not be able to capture the full spectrum of behaviors of the individuals in each enterprise. Another more severe drawback is aggregation bias [3,4], i.e., a predictor at different levels may have different effects on the response. For example, in the nursing context, there is an important predictor called "exchanging", which is originally defined for each individual nurse (i.e., an individual-level predictor) to characterize activities of exchanging information with others regarding patient care. If "exchanging" were aggregated to the unit-level, then the resulting variable, i.e., the average level of information exchange of a unit, is a proxy measure of the unit's normative environment/facility in promoting information exchange and thus may have an effect on the nursing quality of the unit different from the effect of individual nurses' activities in exchanging information.

When predictors are at different levels, multilevel regression provides a more appropriate tool than general regression analysis. Multilevel regression has been discussed in diverse literatures under a variety of titles. For example, they are referred to as multilevel regression or multilevel linear models in sociological research [5,6], as mixed-effects models in biometric research [7-9], as random efficient models in econometrics [10,11], and as covariance components models in statistics [12]. A multilevel regression model allows the inclusion of predictors at two (e.g., individual and enterprise levels) or more levels. The basic idea is to build a separate regression model for each enterprise by linking the individual-level predictors with the response, and then model the variation among enterprises by considering the regression coefficients as multivariate responses explained by enterprise-level predictors. However, existing

multilevel models are not directly applicable to our problem because they require that the response variable be at the individual-level.

To the best of our knowledge, there has been a lack of an effective model to link enterprise-level quality responses with enterprise- and individual- level predictors for quality improvement of the enterprise. This research aims to bridge this gap by proposing a new multilevel regression model with enterprise-level responses, which treats individuals' contributions to the responses as latent variables. Furthermore, the proposed model is applied to a real-world application of nursing quality improvement. Most existing research in nursing quality is either qualitative, or quantitative but only utilizing the unit-level information [13-21], because the activities of individual nurses have long been considered to be non-quantifiable [1]. This application analyzes the data collected by two co-authors (Lamb and Schmitt) in a Robert Wood Johnson Foundation (RWJF) sponsored project, in which the research team designed the *first* instrument to measure nurses' independent activities and interdependent activities in coordinating patient care using a comprehensive collection of variables. The data also include measurements of unit-level quality-affecting factors and quality outcomes. By linking these multiple data sources together, the findings may have a profound impact on nursing quality improvement.

The remainder of the paper is organized as follows: Section 5-2 presents the new multilevel model development; Section 5-3 performs simulation studies to assess the model performance; Section 5-4 presents the findings in applying the proposed model to nursing quality improvement; Section 5-5 concludes the paper.

5.2 Proposed model – multilevel regression with enterprise-level response

5.2.1 Model formulation

Let i be the index for enterprises and m be the total number of enterprises, i.e., $i = 1, \cdots, m$. Let j be the index for individuals and n_i be the total number of individuals in enterprise i, i.e., $j = 1, \cdots, n_i$. Let y_i denote the response (quality outcome) of enterprise i. Let $\tilde{y}_{ij}$ denote the contribution of individual j to y_i and $\tilde{y}_{ij}$ is latent. In this paper, we focus on the situation when the contributions of individuals to the enterprise-level response are additive, i.e., i.e., $y_i = \sum_{j=1}^{n_i} \tilde{y}_{ij}$. Please note that although model formulation (this section) and estimation and inference (next section) are discussed for this simple additive relationship between $\tilde{y}_{ij}, j = 1, \cdots, n_i,$ and $y_i,$ they can be readily extended to address two other situations: one situation is that y_i is a weighted sum of the $\tilde{y}_{ij}, j = 1, \cdots, n_i,$ with known weights; the other situation is that y_i is a general function of $\tilde{y}_{ij}, j = 1, \cdots, n_i,$ i.e., $y_i = f(\tilde{y}_{ij}, j = 1,2, \dots, n_i)$, with the function form, f, known and can be reasonably approximated by a linear function through the Taylor expansion. In this paper, we focus on $y_i = \sum_{j=1}^{n_i} \tilde{y}_{ij}$ for better presentation and clarity and also considering that this relationship is appropriate for the real-world application in Section 5-4.

Also, we focus on a single response; a multi-response model is just a straightforward extension to the single-response model we propose. Furthermore, assume that there are a total of Q individual-level predictors and P enterprise-level predictors. Let x_{qij} denote the measurement on the q-th individual-level predictor for individual j in enterprise i. Let s_{pi} be the measurement on the p-th enterprise-level predictor for enterprise i.

The model formulation consists of two stages: At Stage-1, each individual's latent response is linked to a set of individual-level predictors. At Stage-2, the regression coefficients in the Stage-1 model for each enterprise are response variables that are hypothesized to be explained by enterprise-level predictors. Specifically,

The Stage-1 model is to regress the latent response on the individual-level predictors, i.e.,

$$\tilde{y}_{ij} = \beta_{0i} + \beta_{1i}x_{1ij} + \cdots + \beta_{Qi}x_{Qij} + \varepsilon_{ij}, \quad (1)$$

where $\varepsilon_{ij} \sim N(0, \sigma^2)$. This is to assume that the residual errors, ε_{ij}, $i = 1, \cdots, m$, $j = 1, \cdots, n_i$, are normal and IID (independent and identically distributed) across different individuals and enterprises. Note that the model in (1) keeps i (index for enterprises) in the subscript for the model coefficients, $\beta_{0i}, \cdots, \beta_{Qi}$. This is to acknowledge the uniqueness of each enterprise, i.e., we consider that the effect of individual-level predictors on the quality response may vary across enterprises.

The Stage-2 model is to regress the regression coefficients in (1) on the enterprise-level predictors:

$$\beta_{qi} = \gamma_{q0} + \gamma_{q1}s_{1i} + \cdots + \gamma_{qP}s_{Pi} + u_{qi}, \ q = 0,1, \cdots, Q, \quad (2)$$

where $u_{qi} \sim N(\mathbf{0}, \tau_{qq})$ and $cov(u_{qi}, u_{q'i}) = \tau_{qq'}$ for $q \neq q'$. Note that the distribution parameters, τ_{qq} and $\tau_{qq'}$, do not have enterprise index "i" in their subscripts. This is to assume that the random effects of different enterprises, i.e., u_{qi}, $i = 1, \cdots, m$, are IID across the enterprises. This model aims to characterize how enterprise-level predictors may modify the effect that the individual-level predictors have on the response.

By inserting (2) into (1), a combined model can be obtained:

$$\tilde{y}_{ij} = \gamma_{00} + \sum_{q=1}^{Q} \gamma_{q0}x_{qij} + \sum_{p=1}^{P} \gamma_{0p}s_{pi}$$

$$+ \sum_{p=1}^{P}\sum_{q=1}^{Q} \gamma_{qp}s_{pi}\, x_{qij} + u_{0i} + \sum_{q=1}^{Q} u_{qi}x_{qij} + \varepsilon_{ij}. \quad (3)$$

γ_{00} is the grand mean; γ_{q0} is the fixed main effect from an individual-level predictor; γ_{0p} is the fixed main effect from an enterprise-level predictor; γ_{qp} is the fixed interaction effect between an individual-level predictor and an enterprise-level predictor; u_{0i} and u_{qj} are random effects; ε_{ij} is the residual error. The two-stage model formulation can be more clearly depicted by Fig. 5-1.

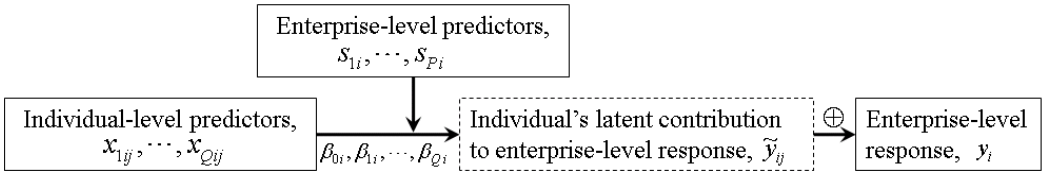


Fig. 5-1 Proposed multilevel model with enterprise-level response

5.2.2 Model estimation

To make the discussion in this section easier, we adopt an equivalent but more succinct representation for the combined model in (3), i.e.,

$$\tilde{y}_{ij} = \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij} + \mathbf{u}_i{}^T \mathbf{x}_{ij} + \varepsilon_{ij}, \tag{4}$$

where $\mathbf{x}_{ij}$ is a vector of measurements on the Q individual-level predictors for individual j in enterprise i, i.e., $\mathbf{x}_{ij} = [1, x_{1ij}, \cdots, x_{Qij}]^T$; $\mathbf{u}_i$ is a vector of the random effects, i.e., $\mathbf{u}_i = [u_{0i}, u_{1i}, \cdots, u_{Qi}]^T$; $\mathbf{s}_i$ is a vector of measurements on the P enterprise-level predictors for enterprise i, i.e., $\mathbf{s}_i = [1, s_{1i}, \cdots, s_{Pi}]^T$; $\boldsymbol{\Upsilon}$ is a $(1+P) \times (1+Q)$ matrix of the fixed effects, i.e.,

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \gamma_{00} & \gamma_{10} & \cdots & \gamma_{Q0} \\ \gamma_{01} & \gamma_{11} & \cdots & \gamma_{Q1} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{0P} & \gamma_{1P} & \cdots & \gamma_{QP} \end{bmatrix}.$$

Furthermore, let $\boldsymbol{\Gamma}$ denote the covariance matrix of $\mathbf{u}_i$.

According to the model in (4), the parameters to be estimated are $\boldsymbol{\Psi} = \{\boldsymbol{\Upsilon}, \boldsymbol{\Gamma}, \sigma^2\}$. The MLE (maximum likelihood estimation) method can be employed. Specifically, it can be derived from (4) that $\tilde{y}_{ij}$ follows a normal distribution, i.e., $\tilde{y}_{ij} \sim N(\mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}, \mathbf{x}_{ij}^T \boldsymbol{\Gamma} \mathbf{x}_{ij} +$

σ^2). Furthermore, based on the relationship $y_i = \sum_{j=1}^{n_i} \tilde{y}_{ij}$, the distribution of y_i can be derived, i.e.,

$$y_i \sim N\left(\sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}, \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \boldsymbol{\Gamma} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right). \quad (5)$$

Because the y_i, $i = 1, \cdots, m$, are observable, the log-likelihood function of $\boldsymbol{\Psi}$ can be built upon the y_i's, i.e.,

$$l(\boldsymbol{\Psi}) \propto \sum_{i=1}^{m} \left\{ -\frac{1}{2} \log\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \boldsymbol{\Gamma} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right) - \frac{1}{2} \frac{\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)^2}{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \boldsymbol{\Gamma} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2} \right\}, \quad (6)$$

It is difficult to find close-form expression for the maximizer of the log-likelihood in (6). Therefore, some iterative algorithm needs to be adopted. We adopt the Fisher-scoring (FS) algorithm [22]. The FS algorithm begins with user-specified initial values for the parameters, i.e., $\boldsymbol{\Psi}^{(0)} = \left\{\boldsymbol{\Upsilon}^{(0)}, \boldsymbol{\Gamma}^{(0)}, \sigma^{2(0)}\right\}$ in our case; and then continuously updates the parameters by $\boldsymbol{\Psi}^{(\mathrm{k}+1)} = \boldsymbol{\Psi}^{(\mathrm{k})} + \alpha^{(k+1)} \mathbf{p}_k$ until some convergence condition is met, e.g., $\left\|\boldsymbol{\Psi}^{(\mathrm{k}+1)} - \boldsymbol{\Psi}^{(\mathrm{k})}\right\|_2 < \epsilon$, where $\epsilon = 10^{-4}$ is a common choice. Clearly, the key in adapting the FS algorithm to our problem setting is to obtain $\mathbf{p}_k$ and $\alpha^{(k+1)}$, called the step direction and step size, respectively. In what follows, we will first discuss how to obtain $\mathbf{p}_k$ and then discuss how to select $\alpha^{(k+1)}$.

The step direction, $\mathbf{p}_k$, can be expressed as $\mathbf{p}_k = -\mathbf{H}^{-1}\mathbf{F}$, where $\mathbf{F}$ is the gradient $\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}}$ evaluated at $\boldsymbol{\Psi}^{(\mathrm{k})}$ and $\mathbf{H}$ is the expectation of the matrix $\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}}$, $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}}\right)$, evaluated at $\boldsymbol{\Psi}^{(\mathrm{k})}$. The derivation of $\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}}$ and $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}}\right)$ can be found in Appendix. One potential limitation in use of $\mathbf{p}_k = -\mathbf{H}^{-1}\mathbf{F}$ for computing $\mathbf{p}_k$ is that we need to calculate the inverse of matrix $\mathbf{H}$, which maybe computationally inefficient if the dimension of $\boldsymbol{\Psi}$ is large. To overcome this limitation, we propose the following strategy: Note that $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \sigma^2}\right) = 0$ and $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \boldsymbol{\Gamma}}\right) = 0$, i.e., the update of $\boldsymbol{\Upsilon}$ is independent of $\boldsymbol{\Gamma}$ and σ^2 in

each iteration of the FS algorithm. This inspires us another way to update $\boldsymbol{\Upsilon}$. Specifically, we can rewrite (5) as

$$y_i = vec\left(\mathbf{s}_i{}^T \otimes \sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) vec(\boldsymbol{\Upsilon})^T + \xi_i \ , \tag{7}$$

where $\xi_i \sim N\left(0, \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \boldsymbol{\Gamma} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i\sigma^2\right)$, and $vec(\cdot)$ is an operator that converts a matrix into a row vector by concatenating the columns of the matrix. Furthermore, Let $\mathbf{V}$ be an $m \times m$ diagonal matrix with diagonal entry $\mathbf{V}_{ii} = \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \boldsymbol{\Gamma} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i\sigma^2$. Let $\mathbf{Z}$ be an $m \times (1+P)(1+Q)$ matrix with the i-th row being $vec\left(\mathbf{s}_i{}^T \otimes \sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)$. Let $\mathbf{Y} = [y_1, \cdots, y_m]^T$. Then, we can use the generalized least squares method to estimate the $vec(\boldsymbol{\Upsilon})^T$ in (7), i.e.,

$$vec\left(\hat{\boldsymbol{\Upsilon}}\right)^T = (\mathbf{Z}^T\mathbf{V}\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{V}\mathbf{Y}. \tag{8}$$

Incorporating this new strategy into the proposed FS algorithm results in the following new algorithm: At the k-the iteration, the FS algorithm is used to obtain estimates for $\boldsymbol{\Gamma}$ and σ^2, i.e., $\boldsymbol{\Gamma}^{(\mathrm{k})}$ and $\sigma^{(\mathrm{k})}$, which are used to compute $\mathbf{V}^{(\mathrm{k})}$; then, the generalized least squares method is used to obtain an estimate for $vec(\boldsymbol{\Upsilon})^T$, i.e., $vec\left(\boldsymbol{\Upsilon}^{(\mathrm{k})}\right)^T = \left(\mathbf{Z}^T\mathbf{V}^{(\mathrm{k})}\mathbf{Z}\right)^{-1}\mathbf{Z}^T\mathbf{V}^{(\mathrm{k})}\mathbf{Y}$. A complete description of this new algorithm is given in Fig. 5-2.

Next, we discuss how to select the step size, $\alpha^{(k+1)}$. The selection of $\alpha^{(k+1)}$ is a classic but challenging problem in optimization. In our case, the approach introduced in [10] is recommended. Specifically, the FS algorithm can start with $\alpha^{(k+1)} = 1$. If $l\left(\boldsymbol{\Psi}^{(\mathrm{k}+1)}\right) > l\left(\boldsymbol{\Psi}^{(\mathrm{k})}\right)$, then accept $\alpha^{(k+1)} = 1$; otherwise, make $\alpha^{(k+1)} = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$, until $l\left(\boldsymbol{\Psi}^{(\mathrm{k}+1)}\right) > l\left(\boldsymbol{\Psi}^{(\mathrm{k})}\right)$. This approach works empirically well. However, a potential drawback is that sometimes it may be impossible to find a positive $\alpha^{(k+1)}$ that makes $l\left(\boldsymbol{\Psi}^{(\mathrm{k}+1)}\right) > l\left(\boldsymbol{\Psi}^{(\mathrm{k})}\right)$, so the algorithm will break down. A major reason is that the

iterations in the FS algorithm may be far away from the optimal solution. To avoid this, the initial values, $\boldsymbol{\Psi}^{(0)} = \left\{\boldsymbol{\Upsilon}^{(0)}, \boldsymbol{\Gamma}^{(0)}, {\sigma^2}^{(0)}\right\}$, should be well chosen.

Input: data for individual- and enterprise-level predictors and enterprise-level response, i.e., $\mathbf{x}_{ij}$, $\mathbf{s}_i$, and y_i; initial values for the parameters, $\boldsymbol{\Psi}^{(0)}$; stopping criterion, ϵ. Denote $\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}} = \{\boldsymbol{\Gamma}, \sigma^2\}$.

Initialize:
Let $converge = false$;
Let $k = 0$;
Repeat
Calculate $\mathbf{F} = \frac{\partial l\left(\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}\right)}{\partial \boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}}$ and $\mathbf{H} = E\left(\frac{\partial l\left(\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}\right)^2}{\partial \boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}} \partial \boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}}\right)$ by (A-1) and (A-2), respectively, both evaluated at $\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}^{(k)}$;
Let $\mathbf{p}_k = -\mathbf{H}^{-1}\mathbf{F}$ and calculate $\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}^{(k+1)} = \boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}^{(k)} + \alpha^{(k+1)}\mathbf{p}_k$;
Calculate $\boldsymbol{\Upsilon}^{(k+1)}$ by (8).

If $\left\|\boldsymbol{\Psi}^{k+1} - \boldsymbol{\Psi}^{k}\right\|_2 \leq \epsilon$ **then**
$converge = true$;
Else
$converge = false$;
End if

Let $k = k + 1$;

Until $converge = true$
Output: $\boldsymbol{\Psi}^{k+1} = \left\{\boldsymbol{\Psi}_{/\boldsymbol{\Upsilon}}^{(k+1)}, \boldsymbol{\Upsilon}^{(k+1)}\right\}$.

Fig. 5-2: The proposed algorithm for estimating the model parameters, $\boldsymbol{\Psi} = \{\boldsymbol{\Upsilon}, \boldsymbol{\Gamma}, \sigma^2\}$, of the proposed multilevel model with enterprise-level response

To select good initial values, we adapt the method developed in [24] for conventional multilevel regression with individual-level responses, to our problem that considers the individual-level responses to be latent. Specifically, our method includes two steps. The first step is to select $\boldsymbol{\Upsilon}^{(0)}$ and ${\sigma^2}^{(0)}$. For this purpose, we ignore the

random effects, so (5) becomes $y_i \sim N\left(\sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}, n_i\sigma^2\right)$. Estimates for $\boldsymbol{\Upsilon}$ and σ^2 can be easily obtained by MLE; these estimates are used as $\boldsymbol{\Upsilon}^{(0)}$ and $\sigma^{2(0)}$, respectively.

The second step is to select $\boldsymbol{\Gamma}^{(0)}$. Specifically, let $r_i = y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon}^{(0)} \mathbf{x}_{ij}$. Then, according to (4), we can make

$$r_i = \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \mathbf{u}_i + \varepsilon_i, \tag{9}$$

where $\varepsilon_i = \sum_{j=1}^{n_i} \varepsilon_{ij}$. Based on (11), the least squares estimate for $\mathbf{u}_i$ can be obtained, i.e.,

$$\hat{\mathbf{u}}_i = \left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) r_i. \tag{10}$$

Inserting (9) into (10), $\hat{\mathbf{u}}_i = \mathbf{u}_i + \left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\varepsilon_i$. This further leads to $\sum_{i=1}^{m} \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i{}^T = \sum_{i=1}^{m} \mathbf{u}_i \mathbf{u}_i{}^T + \varepsilon_i^2 \sum_{i=1}^{m}\left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1} +$

$\sum_{i=1}^{m}\left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\varepsilon_i \mathbf{u}_i{}^T +$

$\sum_{i=1}^{m} \varepsilon_i \mathbf{u}_i \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}$. Because ε_i is independent of any element in $\mathbf{u}_i$, the last two terms are negligible. Therefore,

$\frac{1}{m}\sum_{i=1}^{m} \mathbf{u}_i \mathbf{u}_i{}^T = \frac{1}{m}\left\{\sum_{i=1}^{m} \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i{}^T - \varepsilon_i^2 \sum_{i=1}^{m}\left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}\right\} \approx \frac{1}{m}\left\{\sum_{i=1}^{m} \hat{\mathbf{u}}_i \hat{\mathbf{u}}_i{}^T -\right.$

$\left. n_i \sigma^{(0)^2} \sum_{i=1}^{m}\left[\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\right]^{-1}\right\} \cdot \frac{1}{m}\sum_{i=1}^{m} \mathbf{u}_i \mathbf{u}_i{}^T$ is used as $\boldsymbol{\Gamma}^{(0)}$.

5.2.3 Model inference

After model parameters have been estimated, the next step may be to perform hypothesis testing to check the statistical significance of the parameters. Three types of hypotheses usually need to be tested in our case: tests for fixed effects, tests for random effects, and tests for model comparison.

Specifically, to test a fixed effect, γ_{qp}, i.e., $H_0: \gamma_{qp} = 0$ vs. $H_1: \gamma_{qp} \neq 0$, the test statistic, $t = \hat{\gamma}_{qp}/\sqrt{var(\hat{\gamma}_{qp})}$, can be used, $q = 0,1,\ldots,Q$, $p = 0,1,\ldots,P$. $\hat{\gamma}_{qp}$ is an MLE

estimate for γ_{qp} and $var(\hat{\gamma}_{qp})$ can be asymptotically approximated by the corresponding element in matrix $\mathbf{H}^{-1}$. Recall that both $\hat{\gamma}_{qp}$ and $\mathbf{H}$ have been obtained from the model estimation method proposed in Section 5-2-2. This test statistic is asymptotically standard-normal; this property can be used to calculate the P-value of the test. Furthermore, to test the random effects is to test the covariance matrix of the random effects, $\mathbf{\Gamma}$, i.e., $H_0: \mathbf{\Gamma} = \mathbf{\Gamma}_0$ vs. $H_1: \mathbf{\Gamma} = \mathbf{\Gamma}_1$. For example, to test if a random effect u_{qi} exists, $q = 0,1, \dots, Q$, we can set $\mathbf{\Gamma}_0$ to be a reduced form of $\mathbf{\Gamma}_1$ by making the q-th row and column of $\mathbf{\Gamma}_1$ to be null. Denote the maximum log-likelihood values under H_0 and H_1 by l_0 and l_1. Then, the test statistic is $-2(l_0 - l_1)$, which has an approximate χ^2 distribution with d degrees of freedom, where d is the difference between the number of unique parameters in $\mathbf{\Gamma}_1$ and that in $\mathbf{\Gamma}_0$. This is a typical likelihood-ratio test (LRT), which can also be used to simultaneously test multiple fixed and random effects, i.e., LRT can be used in model comparison.

5.3 Simulation studies

An intuitive, competing approach to the proposed model is to aggregate the individual-level predictors to the enterprise level by using some summary statistics and then perform general regression analysis at the enterprise level. Limitations of this aggregate model have been conceptually discussed in Section 5-1. In this section, we will use simulation data to compare the performance of the aggregate model and the proposed model in terms of the statistical properties in fixed and random effect estimation.

To generate the simulation data, the following true model is used. Consider that an individual-level response, y_{ij}, is linked to one individual-level predictor, x_{ij}, by a regression $y_{ij} = \beta_i x_{ij} + \varepsilon_{ij}$; and β_i is linked to one enterprise-level predictor, s_i, by a regression $\beta_i = \gamma s_i + u_i$. These two regressions can be combined into one, i.e., $y_{ij} = \gamma s_i x_{ij} + u_i x_{ij} + \varepsilon_{ij}$. The parameters of the true model include the fixed effect, γ,

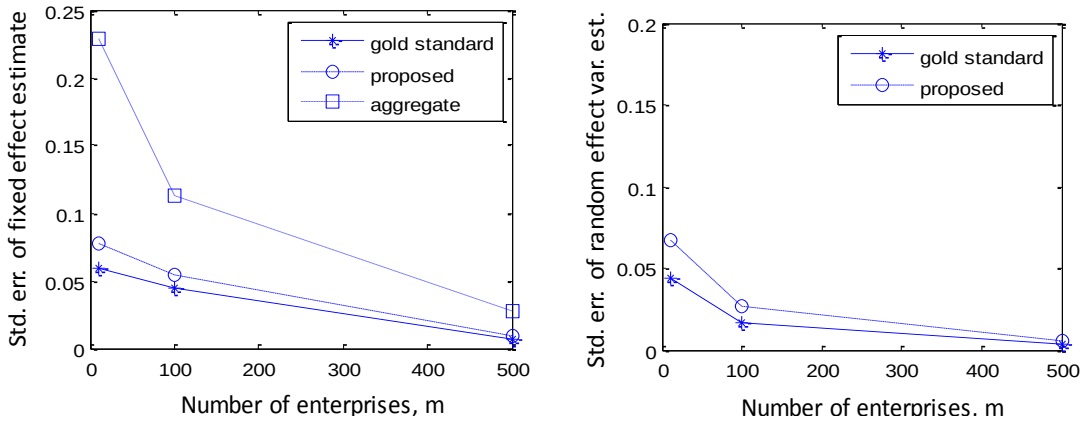
variance of the random effect, $\tau = var(u_i)$, and variance of the residual error, $\sigma^2 = var(\varepsilon_{ij})$. γ, τ, and σ^2 are all assumed to be one. Furthermore, data are generated from the true model for each unit. Specifically, for unit i, the following steps are performed: (i) draw two sample from $N(0,1)$ and make them the values for u_i and s_i, respectively, and compute β_i; (ii) draw n_i samples from $N(0,1)$ for $\varepsilon_{i1}, \cdots, \varepsilon_{in_i}$ and another n_i samples from $N(0,1)$ for $x_{i1}, \cdots, x_{in_i}$, and compute $y_{i1}, \cdots, y_{in_i}$; (iii) compute $y_i = \sum_{j=1}^{n_i} y_{ij}$.

Based on the data, y_i, x_{ij}, and s_i, $j = 1, \cdots, n_i$, $i = 1, \cdots, m$, the proposed model can be applied to estimate the true model parameters γ, τ, and σ^2. Alternatively, the aggregate model can also be applied, which builds an ordinary regression of y_i on predictors s_i, $x_i = \sum_{j=1}^{n_i} x_{ij}$, and $s_i x_i$. In this aggregate model, γ is estimated by the coefficient of predictor $s_i x_i$. The aggregate model is not able to separate between- and within-enterprise variations by providing separate estimates for τ, and σ^2. Instead, it estimates the overall variation by the residual variance. In addition, as data on the individual-level response, y_{ij}, is available in simulation, conventional multilevel regression is also applied to estimate γ, τ, and σ^2. These estimates can be used as the "gold standard" to assess the impact of treating the individual-level response as latent by the proposed model. The results of comparison between the three models are presented as follows:

Fixed effect estimation:

For each model, the average estimate for γ over 100 repetitions of the simulation is obtained. Further, the deviation of the average estimate from the true $\gamma = 1$ is computed to assess the bias in the estimation. The deviations for the gold-standard, proposed, and aggregate models are 0.028, 0.021, 0.023, respectively, when $m = 10$ (number of enterprises) and $n_i = n = 50$ (enterprise sample size). The deviations are

small, which is also true for other m and n values. This implies that the proposed model gives unbiased estimators for fixed effects. In fact, this property of the proposed model can be theoretically proved. Specifically, according to (10), the estimates for the fixed effect are $vec(\widehat{\mathbf{Y}})^T = (\mathbf{Z}^T\mathbf{V}\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{V}\mathbf{Y}$. Then, $E\left(vec(\widehat{\mathbf{Y}})^T\right) = \{(\mathbf{Z}^T\mathbf{V}\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{V}\}E(\mathbf{Y}) = \{(\mathbf{Z}^T\mathbf{V}\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{V}\}\mathbf{Z}vec(\mathbf{Y})^T = vec(\mathbf{Y})^T$, where the second "=" follows from (8). The gold-standard model also gives unbiased estimators for fixed effects, which is a well-known property for conventional multilevel regression. In the aggregate model, the data on the response variable, i.e., the y_i's, are independent but non-identically distributed, because $var(y_i) = x_i^2\tau + n\sigma^2$. Even though the data are not IID, OLS (Ordinary Least Squares) estimation can still give unbiased estimators for the regression coefficients [23], so the estimates for fixed effects by the aggregate model are also unbiased.

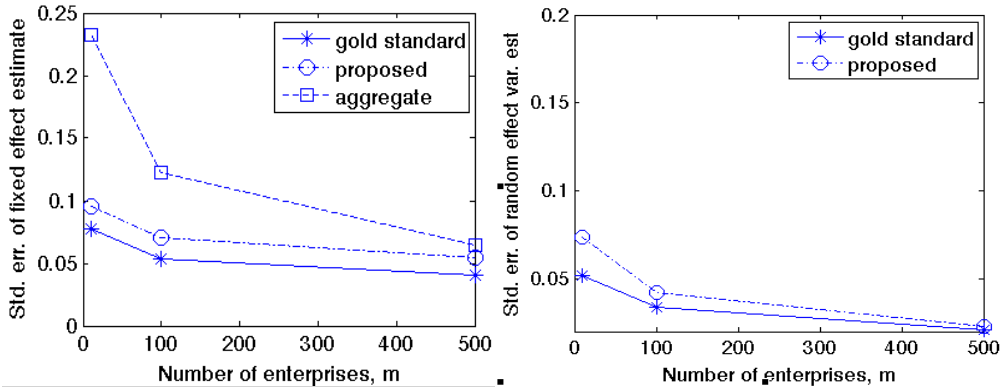


(a) (b)

Fig. 5-3(a) Standard error of the estimate for the fixed effect, γ, vs. number of enterprises, m; (b) Standard error of the estimate for the random effect variance, τ, vs. number of enterprises, m. In both figures, enterprise sample size is fixed to be $n = 50$.

Furthermore, we compare the three models in terms of the standard error of the fixed effect estimate. Fig. 5-3(a) shows the standard error averaged over 100 repetitions of the simulation by each model (y-axis), with respect to the number of enterprises, m (x-

axis). The enterprise sample size, n, is fixed to be 50. It can be seen that the standard error by the proposed model is very close to that by the gold-standard model, whereas that by the aggregate model is much larger. A large standard error leads to the risk of miss-detecting significant fixed effects. Also, Fig. 5-3(a) shows that increasing the number of enterprise, m, can significantly reduce the standard errors for all three models. Furthermore, we vary the enterprise sample size by generating the sample size of each enterprise, n_i, from a uniform distribution on interval [1,10]. The simulation is repeated and the results are shown in Fig. 5-4. Comparing Fig. 5-4 with Fig. 5-3, it can be seen that smaller and unbalanced enterprise sample sizes increase the standard errors but only slightly. In other words, the enterprise sample size influences the standard errors much less than the number of enterprises. This observation is consistent with existing knowledge about multi-level regression [2, 23].



(a) (b)

Fig. 5-4(a) Standard error of the estimate for the fixed effect, γ, vs. number of enterprises, m; (b) Standard error of the estimate for the random effect variance, τ, vs. number of enterprises, m. In both figures, the sample size of each enterprise is sampled from $uniform[1,10]$.

Random effect estimation:

The aggregate model does not include the random effect. Therefore, the comparison is between the proposed and gold-standard models. For each model, the deviation of the average estimate for τ from the true $\tau = 1$ is computed to assess the bias in the estimation. The deviations for the gold-standard and proposed models are -0.015 and -0.024, respectively, when $m = 100$ and $n = 50$. The magnitudes of these deviations become smaller when m increases. This empirically implies that both models might give unbiased estimators for τ. Furthermore, we compare the two models in terms of the standard error of the estimate for τ. The result is given in Fig. 5-3(b). Fig. 5-3(b) shows that the standard error by the proposed model is very close to that by the gold-standard model. Increasing the number of enterprise, m, can significantly reduce the standard errors for both models, whereas increasing n does not have this effect (results not shown here).

Residual error estimation:

For each model, the deviation of the average estimate for σ from the true $\sigma = 1$ is computed to assess the bias in the estimation. The deviations for the gold-standard, proposed, and aggregate models are -0.003, -0.177, 13.394, respectively, when $m = 100$ and $n = 50$. The magnitudes of these deviations have little change when m and n increase. This implies that the proposed model may be biased in estimating the residual error σ; fortunately, the magnitude of the bias is small. In contrast, the aggregate model over-estimates σ with a large bias. The large bias is due to the fact that the aggregate model cannot separate within- and between-enterprise variation, so that the estimate for the residual error is a combination of the two variation sources. Furthermore, we compare the three models in terms of the standard error of the estimate for σ. Same phenomena/trends are observed as the standard error of the fixed effect estimate.

Furthermore, we perform simulation studies with more individual- and enterprise-level predictors. Specifically, in the first set of simulations, we include Q individual-level predictors ($Q > 1$), while keeping the number of enterprise-level predictors to be one. Therefore, the true model used to generate the simulation data is $y_{ij} = \sum_{q=1}^{Q} \gamma_q s_i x_{qij} + \sum_{q=1}^{Q} u_{qi} x_{qij} + \varepsilon_{ij}$, where $\gamma_q = 1$, $\tau_{qq} = var(u_{qi}) = 1$, and $\sigma^2 = var(\varepsilon_{ij})$=1. s_i and x_{qij} are sampled from the $N(0,1)$ distribution. Based on the simulation data, the proposed, aggregate, and gold-standard models are applied to estimate the Q fixed effects, and variances of the Q random effects (the aggregate model cannot estimate these), and the residual variance. Because all three models have been theoretically proven to give an unbiased estimator for each fixed effect, the simulation result for assessing the bias of each model in fixed effect estimation is not shown here. The models are compared in terms of the standard errors of the fixed effects estimates, the biases and standard errors of the random effect variance estimates, and the bias and standard error of the residual variance estimate. Considering the space limits, instead of showing the standard error of each fixed effect estimate, we show the average over the standard errors of the estimates for the Q fixed effects. Similar consideration applies to the random effects, i.e., we show the average over the biases/standard errors of the estimates for the Q random effects' variances. The results for $Q = 8$ are shown in the "$Q = 8, P = 1$" column of each sub-table in Table 5-1.

In the second set of simulations, we include *P* enterprise-level predictors ($P > 1$), while keeping the number of individual-level predictors to be one. Therefore, the true model used to generate the simulation data is $y_{ij} = \sum_{p=1}^{P} \gamma_p s_{pi} x_{ij} + u_i x_{ij} + \varepsilon_{ij}$, where $\gamma_p = 1$, $\tau = var(u_i) = 1$, and $\sigma^2 = var(\varepsilon_{ij})$ =1. s_{pi} and x_{ij} are sampled from the $N(0,1)$ distribution. Based on the simulation data, the three models are applied. The

results for $P = 8$ shown in the "$Q = 1, P = 8$" column of each sub-table in Table 5-1. Note that because there are eight fixed effects, the average over the standard errors of the estimates for these fixed effects is shown in Table 5-1(a) considering the space limits. In addition, the results for one individual-level predictor and one enterprise-level predictor, which have been discussed previously, are copied here for purpose of comparison.

Table 5-1: Comparison between proposed, gold standard, and aggregate models in terms of the statistical properties in model estimation ($m = 100, n = 50$)

(a) Std. err. of fixed effect estimate

	Q=1, P=1	Q=8, P=1	Q=1, P=8
Gold standard	0.045	0.328	0.304
Proposed	0.054	0.554	0.613
Aggregate	0.113	1.012	0.719

(b) Std. err. of random effect variance estimate

	Q=1, P=1	Q=8, P=1	Q=1, P=8
Gold standard	0.017	0.614	0.086
Proposed	0.027	1.245	0.257
Aggregate	N/A	N/A	N/A

(c) Bias of random effect variance estimate

	Q=1, P=1	Q=8, P=1	Q=1, P=8
Gold standard	-0.015	0.044	0.064
Proposed	-0.024	0.214	0.092
Aggregate	N/A	N/A	N/A

(d) Std. err. of residual std. deviation (σ) estimate

	Q=1, P=1	Q=8, P=1	Q=1, P=8
Gold standard	0.00001	0.005	0.002
Proposed	0.002	0.281	0.047
Aggregate	3	26.82	20.368

(e) Bias of residual std. deviation (σ) estimate

	Q=1, P=1	Q=8, P=1	Q=1, P=8
Gold standard	0.003	0.308	0.144
Proposed	0.177	3.228	0.131
Aggregate	13.394	22.174	20.598

The following observations can be drawn from all the simulations performed:

- The proposed model is consistently better than the aggregate model in terms of all the statistical properties chosen for comparison and regardless of the number of predictors.
- The proposed model performs close to the gold-standard model, especially when the number of individual-level predictors, Q, is small. This is because Q

represents the number of random effects; a large Q increases the variation sources, introducing more uncertainty in the model estimation.

- With a fixed number of enterprises, i.e., $m = 100$ in Table 5-2, the less parameters to be estimated, the better the estimation. This is true for all three models. Furthermore, the enterprise sample size, n, influences on the model performance much less than the number of enterprises.

5.4 Application in nursing quality improvement

5.4.1 Data collection and selection of predictors and response variable

The case study in this section uses the data collected by two co-authors in a RWJF-sponsored project. Data collection is mainly in the format of surveys handed out to 614 nurses in 32 hospital units (human subject approval has been obtained). Using the "language" of this paper, nurses are the "individuals" and units are the "enterprises".

Selection of individual-level predictors:

The survey includes 11 questions designed to measure independent nursing activities and interdependent activities in coordinating patient care. Examples of the 11 questions include "I organize the supplies that I need to be able to keep the care of my patients on track", "I initiate actions to get my nursing team members to do what is needed to keep my patients on their plan of care," and "I communicate information to my interdisciplinary team members they need to know to carry out their patient care activities or to make changes in their plan of care". Each question includes four aspects for capturing the nurses' perception of (i) the amount of time spent on the activity in a usual shift, (ii) the priority placed on the activity for a usual shift, (iii) the amount of time spent on the activity in the last shift worked, and (iv) the amount of time spent on this activity compared to the perception of amount of time needed in the last shift worked. In the survey, the nurses were asked to respond to the four aspects for each question. The

response to each aspect is on a 1-5 numerical scale, where 1 to 5 represents the least amount of time spend (for aspects (i), (iii), and (iv))/the lowest priority (for aspects (ii)) to the most amount of time spend/the highest priority. In the data analysis described in this section, we take the average over the responses to the four aspects and consider the average as the response to each question. In this way, the response to each question is a combined measure for the amount of time spent and priority of the activity this question corresponds to.

The 11 questions were designed to be indicators for six underlying constructs, including organizing one's own activities and resources, checking patient progress and response, doing the work of others to keep care on track, assisting each other's work, mobilizing people and resources, and exchanging information with team members. The six constructs are called "organizing", "checking", "backfilling", "assisting", "mobilizing", and "exchanging", or denoted by "o", "c", "b", "a", "m", and "e" in short in this section. Note that constructs "o", "c", and "b" correspond to the independent nursing activities, whereas "a", "m", and "e" correspond to the interdependent nursing activities, i.e., activities that need coordination between nurses and with other health care professionals.

To verify the designed/hypothetical correspondence between the 11 questions and the six constructs, we perform factor analysis with a rotation method called procrustes [25]. This method intends to identify factors underlying the questions, such that the correspondence between the factors and the questions maximally overlaps with the designed/hypothetical correspondence between the six constructs and the questions. Six factors are identified and their correspondence with the 11 questions is almost the same as the designed/hypothetical correspondence, except that "a" is identified to correspond to questions Q5 and Q6, while it is designed to correspond to only Q5.

Question Q6 is “I communicate information to my nursing team members that they need to know to carry out their patient care activities or to make changes in the plan of care”. It is reasonable to believe that this question is an indicator for both “e” and “a”. Therefore, we use the six factors identified as individual-level predictors in our model.

Selection of unit-level predictors:

The survey also includes 30 questions capturing the nurses’ perceptions about the infrastructure, climate, and administration strategies in their respective units. Examples of the 30 questions include “our information technology helps me to find the information I need quickly”, “physicians respond quickly when we call them for a change in an order or change in patient status”, and “the physical layout of the unit allows us to get the supplies we need easily”. The response to each question is on a 1-5 numerical scale, where 1 to 5 represents “strongly disagree” to “strongly agree”. Note that although these questions ask for individual nurses’ responses, it is more appropriate to include them as unit-level predictors rather than individual-level predictors. Therefore, we take the average over the responses from all the nurses in each unit for a question. Furthermore, we perform principal component analysis (PCA) [26] on the 30 unit-level predictors and keep the first principle component (PC) as the final unit-level predictor included in our model. This has two purposes: (i) reduce the number of unit-level predictors; (ii) the first PC is a linear combination of the 30 unit-level predictors, thus serving as an overall measure for the extent to which each unit has characteristics that facilitate nursing quality improvement, hypothetically.

Selection of response variable/quality outcome:

Unit-level data for measuring the quality of nursing in each unit were collected separately from the survey. We include one quality measure, the total number of falls per

1000 patient days, or "falls" in short, as the response variable in our model. A summary of the predictors and response variable that have been selected is given in Table 5-2.

Table 5-2: Description of the predictors and response variable included in the model

	Description	Notes
Individual-level predictors		
o	A factor measuring the level of nursing activity in organizing one's own activities and resources.	"o", "c", "b", "a", "m", and "e" are all standardized variables. Each is standardized by the respective global (i.e., across all units) mean and global standard deviation. Larger values in a predictor indicate higher levels of activity this predictor describes. "Level" is a combined measure for the time spent in this activity and the priority of this activity.
c	A factor measuring the level of nursing activity in checking patient progress and response.	
b	A factor measuring the level of nursing activity in doing the work of others to keep care on track.	
a	A factor measuring the level of nursing activity in assisting each other's work.	
m	A factor measuring the level of nursing activity in mobilizing people and resources.	
e	A factor measuring the level of nursing activity in exchanging information with team members.	
Unit-level predictors		
s	A composite measure based on 30 variables measuring unit infrastructure, climate, and administration strategies	Larger values in "s" indicate that the unit has characteristics greater facilitate nursing quality improvement, hypothetically.
Unit-level Response variable		
falls	Number of falls per 1000 patient days	Smaller values indicate better nursing quality.

5.4.2 Modeling and results

We generate a scatterplot for each predictor in Table 5-2 with respect to the response. For an individual-level predictor, we plot the unit average because the response

is at the unit-level. The scatterplots can be found in Supplementary Material, which show linear trends. This confirms the validity of using a regression model for this particular application. Furthermore, we apply the proposed multi-level model to the data. Stage-1 model includes all six individual-level predictors, i.e., $\tilde{y}_{ij} = \beta_{0i} + \beta_{1i}(o)_{ij} + \beta_{2i}(c)_{ij} + \beta_{3i}(b)_{ij} + \beta_{4i}(a)_{ij} + \beta_{5i}(m)_{ij} + \beta_{6i}(e)_{ij} + \varepsilon_{ij}$. Stage-2 model regresses each Stage-1 model coefficient on the unit-level predictor, i.e., $\beta_{qi} = \gamma_{q0} + \gamma_{q1}s_i + u_{qi},\ q = 0,1,\cdots,6.$ The combined model is

$$\tilde{y}_{ij} = \gamma_{00} + \gamma_{10}(o)_{ij} + \gamma_{20}(c)_{ij} + \gamma_{30}(b)_{ij} + \gamma_{40}(a)_{ij} + \gamma_{50}(m)_{ij} + \gamma_{60}(e)_{ij} + \gamma_{01}s_i + \gamma_{11}(s\_o)_{ij} + \gamma_{21}(s\_c)_{ij} + \gamma_{31}(s\_b)_{ij} + \gamma_{41}(s\_a)_{ij} + \gamma_{51}(s\_m)_{ij}\gamma_{61}(s\_e)_{ij} + u_{0i} + u_{1i}(o)_{ij} + u_{2i}(c)_{ij} + u_{3i}(b)_{ij} + u_{4i}(a)_{ij} + u_{5i}(m)_{ij} + u_{6i}(e)_{ij} + \varepsilon_{ij}, \quad (13)$$

where $(s\_o)_{ij} = s_i \times (o)_{ij}$ and the coefficient γ_{11} reflects the interaction effect between s and o. $\tilde{y}_{ij}$ is the latent contribution of nurse j in unit i to the total number of falls in this unit. $y_i = \sum_{j=1}^{n_i} \tilde{y}_{ij}$ is the total number of falls in unit i, and is observable.

By applying the model estimation method proposed in Section 5-2, we obtain estimates for the fixed effects denoted by the "γ" coefficients in (13), and an estimate for the covariance matrix **Γ** of the random effects denoted by the "u" coefficients. Here, considering the sample size limitation, we assume that **Γ** is diagonal. Therefore, the proposed method actually gives an estimate for the variance/standard deviation of each random effect. In addition, the proposed method gives an estimate for σ^2 that is the variance of the residual ε_{ij}. These estimates are shown in Table 5-2.

Table 5-3: Estimated effects of individual-level and unit-level predictors on the number of falls

Fixed effect	**Estimate**	**se**	**P_value**
Intercept, γ_{00}	3.436	0.187	0.000
o, γ_{10}	-0.094	0.182	0.610
c, γ_{20}	-0.079	0.181	0.670
b, γ_{30}	0.162	0.222	0.475
a, γ_{40}	-0.509	0.225	0.036
m, γ_{50}	-0.218	0.269	0.427
e, γ_{60}	-0.165	0.327	0.620
s, γ_{01}	-0.262	0.146	0.088
s_o, γ_{11}	0.181	0.154	0.255
s_c, γ_{21}	0.127	0.140	0.377
s_b, γ_{31}	0.031	0.144	0.835
s_a, γ_{41}	-0.246	0.181	0.189
s_m, γ_{51}	0.000	0.116	1.000
s_e, γ_{61}	-0.079	0.303	0.798

	Intercept, std dev of u_{0i}	o, std dev of u_{1i}	c, std dev of u_{2i}	b, std dev of u_{3i}	a, std dev of u_{4i}	m, std dev of u_{5i}	e, std dev of u_{6i}
Random effect	1.260e-6	1.472e-6	6.247e-7	0.425	3.555e-6	1.338e-6	0.917
$\boldsymbol{\sigma} =$	1.957e-05						

It can be seen that the model in Table 5-3 (called the full model hereafter) has many fixed effects with large P-values and random effects having small standard deviations. This implies that the model may be simplified. We perform model selection using the LRT suggested in Section 5-2-3. The selected model further goes through model adequacy checks and there is no apparent violation of the model assumptions

(please see Supplement Material for details). However, unit 26 is found to be an outlier. After removing unit 26, the final model is obtained, as shown in Table 5-4. The R^2 value for this model is 0.94, showing a good fit.

Table 5-4: Estimated effects of individual-level and unit-level predictors on the number of falls in the final model

Fixed effect	**Estimate**	**se**	**P_value**
Intercept, γ_{00}	3.538	0.129	0.000
a, γ_{40}	-0.402	0.185	0.038
s, γ_{01}	-0.241	0.076	0.004
s_a, γ_{41}	-0.395	0.081	0.000

	Intercept, std dev of u_{0i}	e, std dev of u_{6i}
Random effect	0.238	0.869
$\boldsymbol{\sigma} =$	0.354	

To facilitate the interpretation of the final model, we write out the Stage-1 and Stage-2 models corresponding to the final model, i.e., $\tilde{y}_{ij} = \beta_{0i} + \beta_{4i}(a)_{ij} + \beta_{6i}(e)_{ij} + \varepsilon_{ij}$ (Stage-1); $\beta_{0i} = \gamma_{00} + \gamma_{01}s_i + u_{0i}$, $\beta_{4i} = \gamma_{40} + \gamma_{41}s_i$, $\beta_{6i} = u_{6i}$ (Stage-2). Some interesting conclusions can be drawn:

- In general, because the final model only includes the individual-level predictors, “a” and “e”, it indicates that other nursing activities, “o”, “c”, “b”, and “m”, do not have a significant impact on the number of falls. Note that “a” and “e” are both interdependent nursing activities in coordinating patient care. This indicates that the coordination between nurses and with other health care professionals may be more important for reducing “falls”, compared with independent nursing activities.

- β_{0i} is the mean number of falls in unit i when the levels of nursing activities "a" and "e" are equal to their respective global means. β_{0i} is affected by "s" according to the Stage-2 model; also, the coefficient for "s", γ_{01}, is negative. This implies that a unit with a high level of "s" will have less mean number of falls than a unit with a low level of "s", even though the nurses in the two units have the same levels of "assisting" and "exchanging" activities. Recall that "s" is a composite measure for the extent to which the unit has characteristics that facilitate nursing quality improvement, hypothetically. Our finding confirms this hypothesis. In addition, β_{0i} consists of a random effect, u_{0i}, whose variance is significant. This implies that hospital units vary in their mean number of falls even after controlling for "s". Furthermore, we can estimate the average of the mean number of falls across the population of units with the same "s" by $\gamma_{00} + \gamma_{01}s$. For example, if we focus on the population of units with $s = 0.0317$ (this number is the average value of "s" over the 32 units in the data), then on average these units will have a mean number of falls equal to $\gamma_{00} + 0.0317\gamma_{01} = 3.53$.
- β_{4i} reflects the strength of association between a nurse's "assisting" activity in unit i and the number of falls. β_{4i} is affected by "s" according to the Stage-2 model; also, the coefficient for "s", γ_{41}, is negative. This implies that in a unit with a high level of "s", increasing the "assisting" activity of nurses will reduce the number of falls more than in a unit with a low level of "s". In addition, β_{4i} does not include a random effect. This implies that after controlling for "s", hospital units behave similarly in terms of the strength of association between nurses' "assisting" activities and the number of falls, i.e., little variability in the strength of association remains to be explained.

Furthermore, $\gamma_{40} + \gamma_{41} s$ can be used to estimate the average strength of association between nurses' "assisting" activities and the number of falls across the population of units with the same "s". Note that because $\gamma_{40} < 0$, $\gamma_{41} < 0$, and $s > 0$, the strength of association is always negative, implying that nurses' "assisting" activities will reduce the number of falls regardless of which unit the nurses belong to.

- β_{6i} reflects the strength of association between a nurse's "exchanging" activity in unit i and the number of falls. β_{6i} includes a random effect, implying that hospital units vary in terms of the strength of association between nurses' "exchanging" activities and the number of falls. However, this variability cannot be accounted for by "s". Furthermore, as β_{6i} does not include any fixed effect, it implies that on average there is little association between nurses' "exchanging" activities and the number of falls.

5.5 Conclusion

This paper proposed a multilevel model to link individual- and enterprise-level predictors with an enterprise-level quality outcome, for enterprise quality improvement. Unlike conventional multilevel regression which requires the outcome be at the individual level, the proposed model treats each individual's contribution to the enterprise quality outcome as a latent variable. An algorithm was proposed to estimate the model parameters, which integrates the FS algorithm and generalized least squares estimation. Simulation studies were conducted to assess the performance of the proposed model, in comparison with the aggregate model which aggregates the individual-level predictors to the enterprise level and the gold-standard model which assumes that each individual's contribution to the enterprise quality outcome is explicitly known. These studies showed that the proposed model performs close to the gold-standard model in terms of the biases

and standard errors of the estimates for the fixed effects, variances of the random effects, and residual variance. In contrast, the aggregate model leads to much larger standard errors of the estimates for the fixed effects, and much larger bias and standard error of the residual variance estimate; also, the aggregate model cannot separate the within- and between-enterprise variations by providing separate estimates for the random effects variances and residual variance.

The proposed model was applied to a real-world application of nursing quality improvement. Our finding showed that the interdependent nursing activities in coordinating patient care, especially the "assisting" and "exchanging" activities, have significant impact on reducing patient falls. Also, our finding confirmed that the unit infrastructure, climate, and administration strategies that are hypothesized to improve nursing quality do help significantly reduce falls. In addition, the "assisting" activity of each nurse and the quality-improving infrastructure, climate, and administration strategies of the nurse's unit promote each other in reducing falls.

Finally, we point out several future research directions. Multiple quality outcomes, such as falls, medication errors, and patient satisfaction, may be considered altogether, leading to a multilevel model with multiple latent responses to be developed. Also, a generalized multilevel model may be more appropriate considering that the response variables may not all be strictly normal. Furthermore, robust model estimation with a large number of predictors and limited sample sizes may be studied. In addition to the methodological development, it would be of interest to investigate the difference between hospital units in terms of how the individual- and unit-level predictors affect nursing quality outcomes and formulate specific quality improvement plans for each unit. Also, instead of using a composite measure as the unit-level predictor, it may reveal more

insights to include each specific quality-assuring measure as a predictor in order to assess the effectiveness of each measure.

Appendix: Derivation of $\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}}$ and $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}}\right)$ in the FS algorithm

Through some matrix algebra, we derive the first derivatives, $\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}}$, as follows:

$$\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Upsilon}} = \sum_{i=1}^{m} \frac{\left(\sum_{j=1}^{n_i} \mathbf{s}_i \mathbf{x}_{ij}^T\right)\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)}{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2},$$

$$\frac{\partial l(\boldsymbol{\Psi})}{\partial \sigma^2} = -\frac{1}{2}\sum_{i=1}^{m} \frac{n_i}{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2} + \frac{1}{2}\sum_{i=1}^{m} \left\{ \frac{n_i \left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)^2}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2} \right\},$$

$$\frac{\partial l(\boldsymbol{\Psi})}{\partial \boldsymbol{\Gamma}} =$$

$$-\frac{1}{2}\sum_{i=1}^{m} \left\{ \frac{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)}{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2} - \frac{\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)^2}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2} \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \right\}, \tag{A-1}$$

Based on the first derivatives, the second derivatives can be further obtained:

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \boldsymbol{\Upsilon}} = -\sum_{i=1}^{m} \frac{1}{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2} \left(\sum_{j=1}^{n_i} \mathbf{s}_i \mathbf{x}_{ij}^T\right) \otimes \left(\sum_{j=1}^{n_i} \mathbf{s}_i \mathbf{x}_{ij}^T\right),$$

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \sigma^2} = -\sum_{i=1}^{m} n_i \frac{\left(\sum_{j=1}^{n_i} \mathbf{s}_i \mathbf{x}_{ij}^T\right)\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2},$$

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \boldsymbol{\Gamma}} = -\sum_{i=1}^{m} \left\{ \frac{\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2} \left(\sum_{j=1}^{n_i} \mathbf{s}_i \mathbf{x}_{ij}^T\right) \otimes \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right) \right\},$$

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \sigma^2 \partial \sigma^2} = \frac{1}{2}\sum_{i=1}^{m} \frac{n_i^2}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2} - \sum_{i=1}^{m} \left\{ n_i^2 \frac{\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)^2}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^3} \right\},$$

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \sigma^2 \partial \boldsymbol{\Gamma}} = \frac{1}{2}\sum_{i=1}^{m} n_i^2 \left\{ \frac{\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^2} - \frac{2\left(y_i - \sum_{j=1}^{n_i} \mathbf{s}_i{}^T \boldsymbol{\Upsilon} \mathbf{x}_{ij}\right)^2 \left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)}{\left(\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i} \mathbf{x}_{ij}\right) + n_i \sigma^2\right)^3} \right\},$$

$$\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Gamma} \partial \boldsymbol{\Gamma}} =$$

$$\frac{1}{2}\sum_{i=1}^{m}\left\{\frac{1}{\left(\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2\right)^2}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\otimes\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)-\right.$$

$$\left.2\frac{\left(y_i-\sum_{j=1}^{n_i}\mathbf{s}_i{}^T\boldsymbol{\Upsilon}\mathbf{x}_{ij}\right)^2}{\left(\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2\right)^3}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\otimes\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\right\},$$

where $\otimes$ is the kronecker operator [23]. Next, we calculate $E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Psi} \partial \boldsymbol{\Psi}}\right)$ with respect to y_i, $i = 1, \cdots, m$, which gives:

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \boldsymbol{\Upsilon}}\right) = -\sum_{i=1}^{m}\frac{1}{\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2}\left(\sum_{j=1}^{n_i}\mathbf{s}_i\mathbf{x}_{ij}^T\right)\otimes\left(\sum_{j=1}^{n_i}\mathbf{s}_i\mathbf{x}_{ij}^T\right),$$

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \sigma^2}\right) = 0,$$

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Upsilon} \partial \boldsymbol{\Gamma}}\right) = 0,$$

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \sigma^2 \partial \sigma^2}\right) = -\frac{1}{2}\sum_{i=1}^{m}\frac{n_i^2}{\left(\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2\right)^2},$$

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \sigma^2 \partial \boldsymbol{\Gamma}}\right) = -\frac{1}{2}\sum_{i=1}^{m}n_i^2\frac{\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)}{\left(\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2\right)^2},$$

$$E\left(\frac{\partial l(\boldsymbol{\Psi})^2}{\partial \boldsymbol{\Gamma} \partial \boldsymbol{\Gamma}}\right) =$$

$$-\frac{1}{2}\sum_{i=1}^{m}\left\{\frac{1}{\left(\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\boldsymbol{\Gamma}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)+n_i\sigma^2\right)^2}\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\otimes\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}\right)\left(\sum_{j=1}^{n_i}\mathbf{x}_{ij}^T\right)\right\}.$$

(A-2)

Reference

[1] Lamb, GS, Schmitt, MH, Edwards, P, Sainfort, F., Duva, I., Higgins, M. Measuring Staff Nurse Care Coordination in the Hospital. Invited presentation, National State of the Science Congress on Nursing Research. Washington, DC, 2008.

[2] Draper, N.R., Smith, H. Applied Regression Analysis, 3rd Edition. Wiley Press, 1998.

[3] Goldstein, H., Multilevel statistical models, 2nd ed, New York: John Wiley, 1995.

[4] Robinson, W.S. Ecological Correlations and the Behavior of Individuals. American Sociological Review, 15, 351-357, 1950.

[5] Goldstein, H., Multilevel statistical models, 2nd ed, New York: John Wiley, 1995.

[6] Mason, W.M., Wong, G.M. and Entwistle, B., Contextual analysis through the multilevel linear model. In S.Leinhardt (Ed), Sociological methodology, pp. 72-103, San Francisco: Jossey-Bass, 1983.

[7] Elston, R.C. and Grizzle, J.E., Estimation of time response curves and their confidence bands, Biometrics, 18, 148-159, 1962.

[8] Laird, N.M. and Ware, H., Random-effects models for longtitudinal data, Biometrics, 38, 963-974, 1982.

[9] Singer, J.D., Using SAS PROC MIXED to fit multilevel models, hierarchical models: Issues and methods, Journal of Educational and Behavioral Statistics, 23(4), 323-355, 1998.

[10] Rosenberg, B. Linear regression with randomly dispersed parameters, Biometrika, 60, 61-75, 1973.

[11] Longford, N., Random coefficient models, Oxford: Clarendon, 1993.

[12] Dempster, A.P., Rubin, D.B. and Tsutakawa, R.K., Estimation in covariance components models, Journal of the American Statistical Association, 76, 341-353, 1981.

[13] Aiken, L.H., Clarke, S.P., Sloane, D.M., Lake, E.T., & Cheney, T. Effects of hospital care environment on patient mortality and nurse outcomes. The Journal of Nursing Administration, 38, 223-229, 2008.

[14] Aiken, L.H., Clarke, S.P., Sloane, D.M., Sochalski, J.A., & Silber, J.H. Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. Journal of the American Medical Association, 288(16), 1987-1993, 2002.

[15] Aiken, L.H., Smith, H.L., & Lake, E.T. Lower Medicare mortality among a set of hospitals known for good nursing care. Medical Care, 32, 771-787, 1994.

[16] Friese, C.R., Lake, E.T., Aiken, L.H., Silber, J., & Sochalski, J.A. Hospital nurse practice environments and outcomes for surgical oncology patients. Health Services Research, 43(4), 1145-63, 2008.

[17] Institute of Medicine, Committee on Quality of Health Care in America. Page, A.. (Ed.). Keeping patients safe: Transforming the work environment of nurses. Washington, DC: National Academies Press, 2004.

[18] Kazanjian, A., Green, C., Wong, J., & Reid, R. Effects of the hospital nursing environment of patient mortality: a systematic review. Journal of Health Services Research and Policy, 10(2), 111-117, 2005.

[19] Lake, E.T. & Friese, C.R. Variations in nursing practice environments: relation to staffing and hospital characteristics. Nursing Research, 55(1), 1-9, 2006.

[20] Laschinger, H.K.S. & Leiter, M.P. The impact of nursing work environments on patient safety outcomes. Journal of Nursing Administration, 36(5), 259-267, 2006.

[21] Mark, B.A., Hughes, L.C., Belyea, M., Bacon, C.T., Chang, Y. & Jones, C.B. Exploring organizational context and structure as predictors of medication errors and falls. Journal of Patient Safety, 4(2), 66-77, 2008.

[22] Longford, N. A fast scoring algorithm for maximum likelihood estimation in unbalanced models with nested random effects, Biometrika, 74(4), 817-827, 1987.

[23] Demidenko, E., Mixed models, theory and applications, New York: John Wiley, 2004.

[24] Sun, Y., Zhang, W.Y. and Tong, H., Estimation of the covariance matrix of random effects in longitudinal studies, The Annals of Statistics, 35(6), 2795-2814, 2007.

[25] Harman, H.H. Modern factor analysis, 3rd Edition, The University of Chicago Press, 1976.

[26] Jolliffe, I.T. Principal component analysis, 2nd Edition, Springer, 2001.

Chapter 6

SPARSE COMPOSITE LINEAR DISCRIMINATION ANALYSIS FOR MULTI-MODALITY NEUROIMAGING DATA FUSION

Abstract

Various imaging modalities have been used for diagnosis of brain disorders such as the Alzheimer's disease (AD). Because different modalities provide complementary measures for the same disease process, fusion of multi-modality data may increase the statistical power in identification of disease-related brain regions. We propose a sparse composite linear discriminant analysis (SCLDA) model for identification of disease-related brain regions from multi-modality neuroimaging data. SCLDA uses a novel formulation that decomposes each LDA parameter into a product of a common parameter shared by all the modalities and a parameter specific to each modality, which enables joint modeling of all the modalities and borrowing of strength from one another. We prove that this formulation is equivalent to a penalized likelihood with non-convex regularization, which can be solved by the DC (difference of convex functions) programming. We perform extensive simulations to show that SCLDA performs better than existing competing algorithms on feature selection. We apply SCLDA to the Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) images of 49 early AD patients and 67 normal controls (NC). Our study identifies disease-related brain regions consistent with findings in the AD literature.

6.1 Introduction

With the rapid advancement of neuroimaging techniques, various imaging modalities have become available. Multi-modality imaging data are especially useful in clinical diagnosis of brain disorders, which provide unique and often complementary characterization of the underlying disease process. For example, in diagnosis of the

Alzheimer's disease (AD), two commonly used imaging modalities are Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET). MRI is a typical structural neuroimaging technique, which allows for visualization of brain anatomy. PET is a typical functional neuroimaging technique, which measures brain activities. The extensive use of MRI and PET in AD diagnosis is supported by the finding that AD is associated with both structural and functional alterations in the brain [2] [3] [4] [5] [6] [11] [12] [13] 14].

An interesting observation obtained from past MRI and PET studies of AD is that there is overlap between the disease-related brain regions detected by MRI and those by PET, such as regions in the hippocampus area and the mesia temporal lobe [16] [17]. This is not surprising since MRI and PET are two complementary measures for the same disease process, i.e., it starts mainly in the hippocampus and entorhinal cortex, and subsequently spreads throughout the temporal and orbiogrontal cortex, poseterior cingulate, and association cortex generally [7]. However, most existing studies only exploited structural and functional alterations in separation, which ignore the potential interaction between them. The fusion of MRI and PET imaging modalities will increase the statistical power in identification of disease-related brain regions, especially at the early stage of disease development, when the disease-related regions are most likely to be weak-effect regions that are difficult to be detected from MRI or PET alone. Once a good set of disease-related brain regions is identified, they can be further used to build an effective classifier (i.e., a biomarker from the clinical perspective) to enable disease diagnose with high sensitivity and specificity. The ability for diagnosing a disease at the early developmental stage is of great clinical value in terms of maximizing the effect of treatment and improving patients' quality of life.

The idea of multi-modality data fusion has been exploited before. For example, a number of models have been proposed to combine electroencephalography (EEG) and functional MRI (fMRI), including parallel EEG-fMRI independent component analysis [18] [19], EEG-informed fMRI analysis [18] [20], and variational Bayesian methods [18] [21]. The purpose of these studies is to combine EEG, which has high temporal resolution but low spatial resolution, and fMRI, which has low temporal resolution but high spatial resolution, so as to obtain an accurate picture for the whole brain with both high spatial and high temporal resolutions[18] [19] [20] [21]. This purpose is different from ours, which focuses on disease-related brain region identification and subsequent classification. On the other hand, there are some existing studies that include either MRI (or perfusion MRI) and PET data for classification [15] [22] [23] [24] [25]. However, these studies do not make use of the fact that MRI and PET measure the same underlying disease process from two complementary perspectives (i.e., structural and functional perspectives), so that the analysis of one imaging modality can borrow strength from the other.

In this paper, we focus on the problem of identifying disease-related brain regions from multi-modality data. This is actually a variable selection problem. Because neuroimaging datasets are typically featured by a high dimensionality and a small sample size (i.e., large p, small n), regularization techniques are needed for effective variable selection, such as the L1-regularization technique [25] [26] [27] [28] [29] [30] and the L2/L1-regularization technique [31]. In particular, L2/L1-regularization has been used for variable selection jointly on multiple related datasets, also known as multitask feature selection [31], which has a similar nature to our problem. Note that both L1- and L2/L1-regularizations are convex regularizations, which have gained them popularity in the literature. On the other hand, there is increasing evidence that these convex regularizations tend to produce too severely shrunken parameter estimates [33] [34] [35]

[39]. As a result, these convex regularizations may have a risk of not being able to identify the weak-effect disease-related brain regions, which unfortunately may make up a large portion of the disease-related brain regions at the early stage of disease development. Also, convex regularizations tend to select many irrelevant variables to compensate for the overly severe shrinkage in the parameters of the relevant variables [34]. Considering these limitations of convex regularizations, we study non-convex regularizations [33] [34] [35] [39], which have the advantage of producing mildly or slightly shrunken parameter estimates so as to be able to preserve weak-effect disease-related brain regions and enjoy the advantage of avoiding selecting many disease-irrelevant regions.

Specifically in this paper, we propose a sparse composite linear discriminant analysis model, called SCLDA, for identification of disease-related brain regions from multi-modality data. Please note that although we have been using MRI and PET to discuss the context of this study, the proposed SCLDA is general to any number of imaging modalities. Also, SCLDA can be used to for multi-class discrimination.

The contributions of our paper include:

- **Formulation:** We propose a novel formulation that decomposes each LDA parameter into a product of a common parameter shared by all the data sources and a parameter specific to each data source. This formulation enables joint modeling of all the data sources and borrowing of strength from one another. We further prove that this formulation is equivalent to a penalized likelihood with non-convex regularization.
- **Algorithm**: We show that the proposed non-convex optimization can be solved by the DC (difference of convex functions) programming [39]. More importantly, we show that in using the DC programming, the property of the non-convex regularization in terms of preserving weak-effect features can be nicely revealed.

- **Application**: We apply the proposed SCLDA to the PET and MRI data of early AD patients and normal controls (NC). Our study identifies disease-related brain regions that are consistent with the findings in the AD literature. AD vs. NC classification based on SCLDA-identified regions achieves high accuracy, which makes the proposed method a potential tool to complement the existing tools in clinical diagnosis of early AD. In contrast, the convex-regularization-based multitask feature selection method [31] identifies more irrelevant brain regions and yields a lower classification accuracy.

The rest of the paper is organized as follows: Section 6-2 reviews LDA and some of its variants. Section 6-3 presents the formulation of SCLDA, as well as the DC programming used to solve the optimization problem associated with the SCLDA. Section 6-4 presents the results of experiments on synthetic data. Section 6-5 presents an application of SCLDA for identifying disease-related brain regions by fusing PET and MRI. Section 6-6 is the conclusion.

6.2 Review of LDA and its variants

Denote $\boldsymbol{Z} = \{Z_1, Z_2, \dots, Z_p\}^T$ as the variables and assume there are J classes. Denote N_j as the sample size of class j and $N = \sum_{j=1}^{J} N_j$ is the total sample size. Let $\mathbf{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}^T$ be the $N \times p$ sample matrix, where $\boldsymbol{z}_i$ is the i^{th} sample and $g(i)$ is its associated class index. Let $\widehat{\boldsymbol{\mu}}_j = \frac{1}{N_j}\sum_{i=1, g(i)=j}^{N} \boldsymbol{z}_i$ be the sample mean of class j, $\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_i$ be the overall sample mean, $\mathbf{T} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}})^T$ be the total normalized sum of squares and products (SSQP), $\mathbf{W}_j = \frac{1}{N_j}\sum_{i=1, g(i)=j}^{N}\big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_j\big)\big(\boldsymbol{z}_i - \widehat{\boldsymbol{\mu}}_j\big)^T$ be the normalized class SSQP of class j, and $\mathbf{W} = \frac{1}{N}\sum_{j=1}^{J} N_j \mathbf{W}_j$ be the overall normalized class SSQP.

The objective of LDA is to seek for a $p \times r$ linear transformation matrix, $\boldsymbol{\theta}_r$, $r = J - 1$, such that $\boldsymbol{\theta}_r^T Z$ retains the maximum amount of class discrimination information in Z. To achieve this objective, one approach is to find the $\boldsymbol{\theta}_r$ that maximizes the between-class variance of $\boldsymbol{\theta}_r^T Z$, which can be measured by $\text{tr}(\boldsymbol{\theta}_r^T \mathbf{T} \boldsymbol{\theta}_r) - \text{tr}(\boldsymbol{\theta}_r^T \mathbf{W} \boldsymbol{\theta}_r)$, while minimizing the within-class variance of $\boldsymbol{\theta}_r^T Z$, which can be measured by $\text{tr}(\boldsymbol{\theta}_r^T \mathbf{W} \boldsymbol{\theta}_r)$. Here tr() is the matrix trace operator. This is equivalent to solving the following optimization problem:

$$\widehat{\boldsymbol{\theta}}_r = \text{argmax}_{\boldsymbol{\theta}_r} \frac{\text{tr}(\boldsymbol{\theta}_r^T \mathbf{T} \boldsymbol{\theta}_r)}{\text{tr}(\boldsymbol{\theta}_r^T \mathbf{W} \boldsymbol{\theta}_r)}. \quad (1)$$

Note that $\widehat{\boldsymbol{\theta}}_r$ corresponds to the right eigenvector of $\mathbf{W}^{-1}\mathbf{T}$.

Another approach used to find the $\boldsymbol{\theta}_r$ is to use maximum likelihood estimation for Gaussian populations that have different means and a common covariance matrix. Specifically in [36], this approach was developed by assuming that the class distributions are Gaussian with a common covariance matrix, and their mean differences lie in a r-dimensional subspace of the p-dimensional original variable space. Hastie [37] further generalized this approach by assuming that each class distribution is a mixture of Gaussians, which has more flexibility than LDA. However, both approaches assume a common covariance matrix for all the classes, which is too strict in many practical applications, especially in high-dimensional problems where the covariance matrices of different classes tend to be different. Consequently, the linear transformation explored by LDA may not be effective.

In [38], a heterogeneous LDA (HLDA) is developed to relax this assumption. The HLDA aims to find a $p \times p$ linear transformation matrix, $\boldsymbol{\theta}$, in which only the first q columns ($\boldsymbol{\theta}_q$) contain discrimination information and the remaining $p - q$ columns ($\boldsymbol{\theta}_{p-q}$) contain no discrimination information. For Gaussian models, assuming lack of

discrimination information is equivalent to assuming that the means and the covariance matrices of the class distributions are the same for all classes, in the $p-q$ dimensional subspace. Following this, the log-likelihood function of $\boldsymbol{\theta}$ can be written as follows [38]:

$$l(\boldsymbol{\theta}|\mathbf{Z}) = -\frac{N}{2}\log\left|\boldsymbol{\theta}_{p-q}^{T}\mathbf{T}\boldsymbol{\theta}_{p-q}\right| - \sum_{j=1}^{J}\frac{N_j}{2}\log\left|\boldsymbol{\theta}_{q}^{T}\mathbf{W}_j\boldsymbol{\theta}_{q}\right| + N\log|\boldsymbol{\theta}|, \qquad (2)$$

Here $|\mathbf{A}|$ denotes the determinant of matrix $\mathbf{A}$. There is no closed-form solution for $\boldsymbol{\theta}$. As a result, numeric methods are needed to derive the maximum likelihood estimate for $\boldsymbol{\theta}$. q can be specified by users or some model selection criteria such as Akaike's Information Criterion (AIC). It is worth mentioning that the LDA in the form of (1) is a special case of the HLDA [38].

6.3 The proposed SCLDA

6.3.1 The formulation of SCLDA

For the same set of physical variables such as brain regions, $\boldsymbol{Z} = \{Z_1, Z_2, \ldots, Z_p\}^T$, there may be multiple data sources each capturing a different aspect of the set of physical variables. For example, MRI data contains volumetric information of each brain region and PET data measures regional activities. Let $\boldsymbol{z}^{(m)}$ denote the m-th data source for $\boldsymbol{Z}$, $\mathrm{m} = 1,2,\ldots,\mathrm{M}$. $\boldsymbol{z}^{(m)}$ is a $N \times p$ sample matrix.

For each data source, $\boldsymbol{z}^{(m)}$, there is a linear transformation matrix $\boldsymbol{\theta}^{(m)}$, which retains the maximum amount of class discrimination information. A naive way for estimating $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(M)}\}$ is to separately estimate each $\boldsymbol{\theta}^{(m)}$ based on $\boldsymbol{z}^{(m)}$. Apparently, this approach does not take advantage of the fact that all the data sources measure the same set of physical variables (e.g., the same set of brain regions), so they may consist of complementary information. Also, when the sample size of each data source is small, this approach may lead to unreliable estimates for the $\boldsymbol{\theta}^{(m)}$'s.

To tackle these problems, we propose a composite parameterization following the line as [40]. Specifically, let $\theta_{k,l}^{(m)}$ be the element at the *k*-th row and *l*-th column of $\boldsymbol{\theta}^{(m)}$. We treat $\left\{\theta_{k,l}^{(1)}, \theta_{k,l}^{(2)}, \ldots, \theta_{k,l}^{(M)}\right\}$ as an interrelated group and parameterize each $\theta_{k,l}^{(m)}$ as $\theta_{k,l}^{(m)} = \delta_k \gamma_{k,l}^{(m)}$, for $1 \le k \le p$, $1 \le l \le p$ and $1 \le m \le M$. In order to assure identifiability, we restrict each $\delta_k \ge 0$. Here, δ_k represents the common information shared by all the data sources about variable k, while $\gamma_{k,l}^{(m)}$ represents the specific information only captured by the m^{th} data source. For example, in "diseased" vs. "normal" discrimination, if $\delta_k = 0$, it means that all the data sources indicate that variable/region k is not a disease-related brain region. Specifically, if there are two data sources, MRI and PET, $\delta_k = 0$ implies that the disease is irrelevant to structural or functional alteration in region k. $\delta_k \neq 0$ implies that region k may be a disease-related brain region and this assertion is supported by the m^{th} data source if $\gamma_{k,l}^{(m)} \neq 0$. Specifically, if the m^{th} data source is MRI (or PET), $\gamma_{k,l}^{(m)} \neq 0$ implies that the disease is relevant to structural (or functional) alteration in region k.

The log-likelihood function of $\boldsymbol{\Theta}$ is:

$$l_0\left(\boldsymbol{\Theta} \middle| \left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(M)}\right\}\right) =$$

$$\sum_{m=1}^{M} \left\{ -\frac{N^{(m)}}{2} \log \left| {\boldsymbol{\theta}_{p-q}^{(m)}}^{T} \mathbf{T}^{(m)} \boldsymbol{\theta}_{p-q}^{(m)} \right| - \sum_{j=1}^{J} \frac{N_j^{(m)}}{2} \log \left| \boldsymbol{\theta}_q^{(m)} \mathbf{W}_j^{(m)} \boldsymbol{\theta}_q^{(m)} \right| + \right.$$

$$\left. N^{(m)} \log \left| \boldsymbol{\theta}^{(m)} \right| \right\},$$

which follows the same line as (2). However, our formulation includes the following constraints on $\boldsymbol{\Theta}$:

$$\theta_{k,l}^{(m)} = \delta_k \gamma_{k,l}^{(m)}, \delta_k \ge 0, 1 \le k, l \le p, 1 \le m \le M. \quad (3)$$

Let $\boldsymbol{\Gamma} = \left\{\gamma_{k,l}^{(m)}, 1 \le k \le p,\ 1 \le l \le p, 1 \le m \le M\right\}$ and $\boldsymbol{\Psi} = \{\delta_k, 1 \le k \le p\}$. An intuitive choice for estimation of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$ is to maximize the $l_0\left(\boldsymbol{\Theta}|\left\{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \dots, \mathbf{Z}^{(M)}\right\}\right)$ subject to the constraints in (3). However, it can be anticipated that no element in the estimated $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$ will be exactly zero, resulting in a model which is not interpretable, i.e., poor identification of disease-related regions. Thus, we encourage the estimation of $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$ to be sparse, by imposing the L1-penalty on $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$. By doing so, we obtain the following optimization problem for the proposed SCLDA:

$$\widehat{\boldsymbol{\Theta}} = \text{argmin}_{\boldsymbol{\Theta}}\, l_1\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right),$$

$$= \text{argmin}_{\boldsymbol{\Theta}}\left\{-l_0\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right) + \lambda_1 \textstyle\sum_k \delta_k + \lambda_2 \sum_{k,l,m} \gamma_{k,l}^{(m)}\right\},$$

subject to

$$\theta_{k,l}^{(m)} = \delta_k \gamma_{k,l}^{(m)},\ \delta_k \ge 0,\ 1 \le k, l \le p,\ 1 \le m \le M. \tag{4}$$

Here, λ_1 and λ_2 control the degrees of sparsity of $\boldsymbol{\Psi}$ and $\boldsymbol{\Gamma}$, respectively. Tuning of two regularization parameters is difficult. Fortunately, we prove the following Theorem which indicates that formulation (4) is equivalent to a simpler optimization problem involving only one regularization parameter.

Theorem 1. *The optimization problem (4) is equivalent to the following optimization problem:*

$$\widetilde{\boldsymbol{\Theta}} = \text{argmin}_{\boldsymbol{\Theta}}\, l_2\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right),$$

$$= \text{argmin}_{\boldsymbol{\Theta}}\left\{\begin{matrix} -l_0\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right) \\ + \lambda \sum_k \sqrt{\sum_{l=1}^{q} \sum_{m=1}^{M} \left|\theta_{k,l}^{(m)}\right|} \end{matrix}\right\}, \tag{5}$$

with $\lambda = 2\sqrt{\lambda_1 \lambda_2}$, *i.e.,* $\hat{\theta}_{k,l}^{(m)} = \tilde{\theta}_{k,l}^{(m)}$.

The proof can be found in the Appendix.

6.3.2 Use DC programming to solve (5)

The optimization problem (5) is a non-convex optimization problem that is difficult to solve. We address this problem by using an iterative two-stage procedure known as the Difference of Convex functions (DC) programming [39]. The basic idea behind the DC programming is that, at the first stage of every iteration, a surrogate convex objective function is proposed to bound the non-convex objective function at the current solution; then, at the second stage, a new solution is obtained by maximizing this surrogate convex objective function. This process iterates until a certain convergence rule is met. It is worth mentioning that the DC programming shares the same spirit as the Expectation-Maximization (EM) algorithm or Minorization-Maximization (MM) algorithm that has been widely used in statistics and machine learning.

We adopt the DC algorithm to solve (5) following [39], in which the DC programming is to solve a least-square problem with non-convex penalty terms. In our problem, the essential task is to find a decomposition of $\sum_k \sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|}$ as a sum of two convex functions. As suggested in [39], the decomposition, $\sqrt{\theta_{k,l}^{(m)}} = \left|\theta_{k,l}^{(m)}\right| - \left(\left|\theta_{k,l}^{(m)}\right| - \sqrt{\theta_{k,l}^{(m)}}\right)$, can be used for the non-convex penalty term $\sqrt{\theta_{k,l}^{(m)}}$. This inspires us to use

$$\sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|} = \sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right| - \left(\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right| - \sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|}\right),$$

as a decomposition for $\sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|}$. Based on the theory developed in [39], this decomposition results in the following objective function

$$\tilde{\boldsymbol{\Theta}}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{\Theta}} l_3\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right) =$$

$$\operatorname{argmin}_{\boldsymbol{\Theta}} \left\{-l_0\left(\boldsymbol{\Theta}|\left\{\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \dots, \boldsymbol{z}^{(M)}\right\}\right) + \sum_k \lambda_k^{(t+1)} \sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|\right\}, \quad (6)$$

as the surrogate convex objective function, where

$$\lambda_k^{(t+1)} = \lambda - h'(z)\big|_{z=\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)(t)}\right|}, \; h(z) = \lambda\left(z - \sqrt{z}\right),$$

$\widetilde{\boldsymbol{\Theta}}^{(t+1)}$ is the solution at iteration $t+1$ and $\theta_{k,l}^{(m)(t)}$ is a corresponding element. It is shown that this decomposition produces a surrogate convex objective function with L1-penalty, which can be solved by many existing efficient algorithms developed for LASSO-type of problems. A complete procedure for the DC programming is depicted in Figure 6-1.

Initialize: Let $t = 0$;
Repeat
Calculate

$$\lambda_k^{(t)} = \lambda - h'(z)\big|_{z=\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)(t)}\right|};$$

for $1 \le k, l \le p, \; 1 \le m \le M$;
Solve (6) and get $\widetilde{\boldsymbol{\Theta}}^{(t)}$;
Let $t = t + 1$;
Until converge

Fig. 6-1. The DC programming for solving (5)

The optimization problem (6) is a standard L1-regularization type of problem, whose objective function is a sum of a smooth likelihood/lease-square error function and an L1-penalty on the parameters. This problem can be solved by many efficient numeric algorithms in the literature [25, 26]. In our case, the two-metric method is employed [25].

It has been pointed out that the DC programming for solving many non-convex regularization problems can be closely linked to the adaptive LASSO formulation [39]. To illustrate this point in our case, we note that

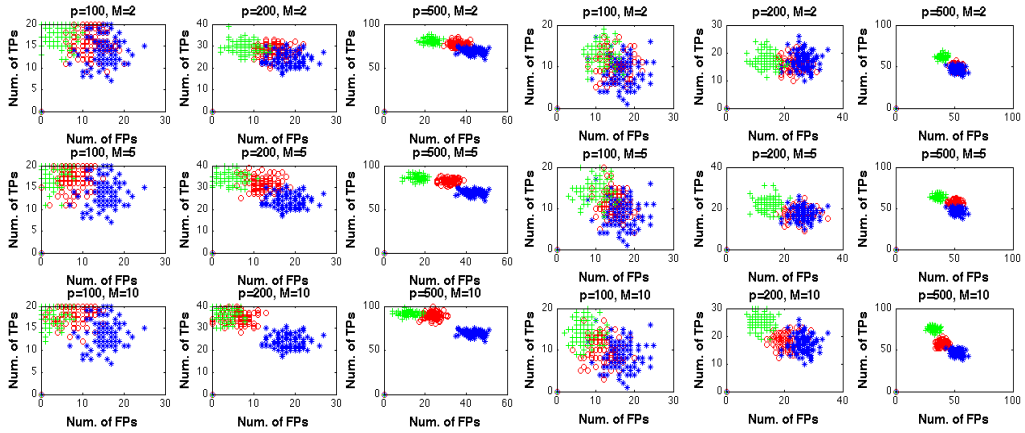
$$h'(z)|_{z=\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|} = \lambda - \frac{\lambda}{2\left(\sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)}\right|} + \sigma\right)}$$

is in our DC programming, where σ is a user-specified number for numerical stability consideration. This produces a new regularization parameter for iteration $t+1$,

$$\lambda_k^{(t+1)} = \frac{\lambda}{2\left(\sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)^{(t)}}\right|} + \epsilon\right)},$$

which is inversely proportional to the magnitude of $\sqrt{\sum_{l=1}^{q}\sum_{m=1}^{M}\left|\theta_{k,l}^{(m)^{(t)}}\right|}$.

This implies that, at each iteration, the DC programming essentially reweights the regularization parameters of each L1-regularization type of problem in (6). Specifically, at iteration $t+1$, the new regularization parameters associated with the zero $\theta_{k,l}^{(m)}$'s identified at the previous iteration will increase drastically, while the new regularization parameters associated with the non-zero $\theta_{k,l}^{(m)}$'s identified at the previous iteration will decrease proportionally to the sum of the absolute magnitudes of these $\theta_{k,l}^{(m)}$'s. which belong to the same variable. In this manner, the shrinkage effect imposed on the non-zero $\theta_{k,l}^{(m)}$'s by the L1-regularization is effectively alleviated. This explains why the proposed SCLDA has the capability of preserving weak-effect features.



(a) (b)

Fig. 6-2. Average numbers of TPs vs. FPs for proposed SCLDA (green symbols "+"), SLDA (blue symbols "*"), and MSLDA (red symbols "o") with (a) $s\% = 90\%, n/p = 1$; (b) $s\% = 70\%, n/p = 1$.

6.3.3 Selection of λ and q

Recall that λ is the regularization parameter in (5); q is the number of columns in $\boldsymbol{\theta}^{(M)}$ that contains discrimination information. We first discuss the optimal selection of λ and q for the case when there is only one data source. This optimal selection can be obtained by maximizing Akaike's Information Criteria (AIC) [46]:

$$AIC = s(\lambda, q) - \frac{d}{n_{avg}},$$

where $s(\lambda, q)$ is the cross validation classification accuracy associated with λ and q, d is the number of nonzero parameters in $\widetilde{\boldsymbol{\Theta}}$, and n_{avg} is the average number of observations per class. For the general case where there are M data sources, the optimal selection of λ and q can be obtained by maximizing the average AIC, which is:

$$AIC_{avg} = \sum_{i=1}^{M} AIC_i / M.$$

where AIC_i is the AIC value for the i^{th} data source. Both the simulation studies in section 6-4 and real application in section 6-5 reveal that this criterion can produce accurate and meaningful model selection results.

6.4 Simulation studies

In this section, we conduct experiments to compare the performance of the proposed SCLDA with sparse LDA (SLDA) [42] and multitask feature selection [31]. Specifically, as we focus on LDA, we use the multitask feature selection method developed in [31] on LDA, denoted as MSLDA. Both SLDA and MSLDA adopt convex regularizations. Specifically, SLDA selects features from one single data source with L1-regularization; MSLDA selects features from multiple data sources with L2/L1 regularization.

We evaluate the performances of these three methods across various parameters settings, including the number of variables, p, the number of features, l, the number of data sources, M, sample size, n, and the degree of overlapping of the features across different data sources, $s\%$ (the larger the $s\%$, the more shared features among the datasets). Definition of $s\%$ can be found in the following simulation procedure. For a given combination of values of these parameters, simulation data can be generated in the following way: First, we generate a $1 \times q$ vector, $\boldsymbol{\beta}$, with l elements being randomly selected to be 1 and the other elements being zero. Second, to generate a feature vector for dataset i, $\boldsymbol{\beta}_i$, we randomly change $(100 - s)\%$ of the non-zero elements of $\boldsymbol{\beta}$ to be zero, and, at the same time, we randomly change the same number of zero elements to be 1. The nonzero elements of $\boldsymbol{\beta}_i$ correspond to the features of dataset i. In this manner, the larger the $s\%$, the more overlapping of the features across the data sources. We further randomly pick up half of the nonzero elements of each $\boldsymbol{\beta}_i$ and modify them in such a way: its new value is sampled from either the uniform distribution $U(0,0.5)$ or the uniform distribution $U(-0.5,0)$. This is to mimic the true situation in the real application in Section 6-5, where a large portion of weak-effect features are present. After that, we use the resulting $\boldsymbol{\beta}_i$ and $-\boldsymbol{\beta}_i$ as the mean vectors of two classes of dataset i (for simplicity, we only investigate 2-class problems for the simulation). For each data set, the

covariance matrix is generated from a Wishart distribution with the degree of freedom being n and the scale matrix being an Identity matrix. Each covariance matrix is further diagonalized to make the diagonal elements being 0.2 (the corresponding signal-to-noise ratio is about 5). With these mean vectors and covariance matrices, we generate the dataset i by sampling one sub-dataset for each class from its corresponding multivariate Gaussian distribution, and then combining these two sub-datasets as dataset i. For each specification of the parameters settings, M datasets can be generated following the simulation procedure. We apply the proposed SCLDA to the M datasets, and identify one feature vector $\widehat{\boldsymbol{\theta}}^{(i)}$ for each dataset, with λ and q chosen by the method described in Section 6-3-3. The result can be described by the number of true positives (TPs) and the number of false positives (FPs). Here, true positives are the non-zero elements in the learned feature vector $\widehat{\boldsymbol{\theta}}^{(i)}$ that are also non-zero in the $\boldsymbol{\beta}_i$; false positives are the non-zero elements in $\widehat{\boldsymbol{\theta}}^{(i)}$ that are actually zero in $\boldsymbol{\beta}_i$. As there are M pairs of TPs and FPs for M datasets, the average TP over the M datasets and the average FP over the M datasets are used as the performance measures. This procedure (i.e., from data simulation, to SCLDA, to TPs and FPs generation) can be repeated for 100 times, and 100 pairs of average TP and average FP are collected for SCLDA. In a similar way, we can obtain 100 pairs of average TP and average FP for both SLDA and MSLDA.

Fig. 6-2 (a) and (b) show comparison between SCLDA, SLDA and MSLDA by scattering the average TP against the average FP for each method. Each point corresponds to one of the 100 repetitions. The comparison is across various parameter settings, including the number of variables ($p = 100, 200, 500$), the number of data sources ($M = 2, 5, 10$), and the degree of overlapping of the features across different data sources ($s\% = 90\%, 70\%$). Additionally, n/p is kept constant, i.e., $n/p = 1$. A general observation is that SCLDA is better than SLDA and MSLDA across all the parameter

settings. Some specific trends can be summarized as follows: (i) Both SCLDA and MSLDA outperform SLDA in terms of TPs; SCLDA further outperforms MSLDA in terms of FPs. (ii) In Fig. 6-2 (a), rows correspond to different numbers of data sources, i.e., $M = 2, 5, 10$, respectively. It is clear that the advantage of SCLDA over both SLDA and MSLDA is more significant when there are more data sources. Also, MSLDA performs consistently better than SLDA. Similar phenomena are shown in Fig. 6-2 (b). This demonstrates that in analyzing each data source, both SCLDA and MSLDA are able to make use of the information contained in other data sources. SCLDA can use this information more efficiently, as SCLDA can produce less shrunken parameter estimates than MSLDA and thus SCLDA is able to preserve weak-effect features. (iii) Comparing Fig. 6-2 (a) and (b), it can be seen that the advantage of SCLDA or MSLDA over SLDA is more significant as the data sources have higher degree of overlapping in their features. Finally, although not presented here, our simulation shows that the three methods perform similarly when $s\% = 40\%$ or less.

Furthermore, we conduct an experiment to compare SCLDA, SLDA, and MSLDA under different sample sizes. As shown in Fig. 6-3, SCLDA performs significantly better when sample sizes are small, which confirms that SCLDA is statistically more efficient.

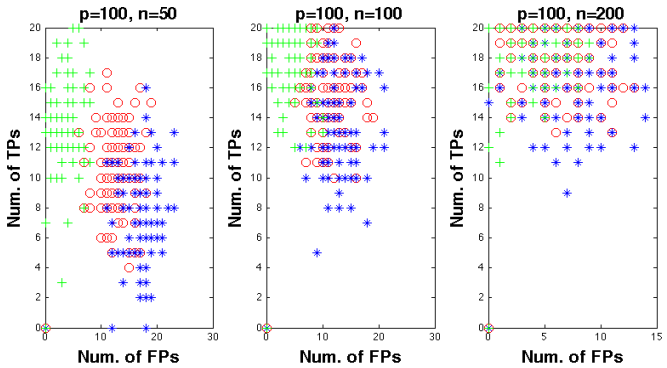


Fig. 6-3. Average numbers of TPs vs. FPs for proposed SCLDA (green symbols "+"), SLDA (blue symbols "*"). and MSLDA (red symbols "o") with $s\% = 90\%, M = 2$.

6.5 Applications

6.5.1 Data preprocessing

Our study includes 49 early AD patients and 67 age-matched normal controls (NC), with each subject of AD or NC being scanned by both PDG-PET (a type of PET imaging) and MRI. The FDG-PET and MRI images can be downloaded from the database created by the Alzheimer's Disease Neuroimaging Initiative (www.adni-info.org). In what follows, we outline the data preprocessing steps.

Each image is spatially normalized to the Montreal Neurological Institute (MNI) template using the affine transformation and subsequent non-linear warpping algorithm [43] implemented in the SPM MATLAB toolbox. This is to ensure that each voxel is located in the same anatomical region for all subjects, so that spatial locations can be reported and interpreted in a consistent manner. Once all the images in the MNI template, we further apply the Automated Anatomical Labeling (AAL) technique [43] to segment the whole brain of each subject into 116 brain regions. The 90 regions that belong to the cerebral cortex are selected for the later analysis, as the other regions not included in the cerebral cortex are rarely considered related to AD in the literature. The measurement of each region in the PDG-PET data is regional metabolism of glucose; the measurement of each region in the MRI data is the structural volume of the region.

6.5.2 Disease-related brain regions

SCLDA is applied to the preprocessed PET and MRI data of AD and NC. 26 disease-related brain regions are identified from PET and 21 from MRI (see Table 6-1 for their names). The maps of the disease-related brain regions identified from PET and MRI are highlighted in Fig. 6-4 (a) and (b), respectively, with different colors given to neighboring regions in order to distinguish them. Each figure is a set of horizontal cut

away slices of the brain as seen from the top, which aims to provide a full view of locations of the regions.

TABLE 6-1

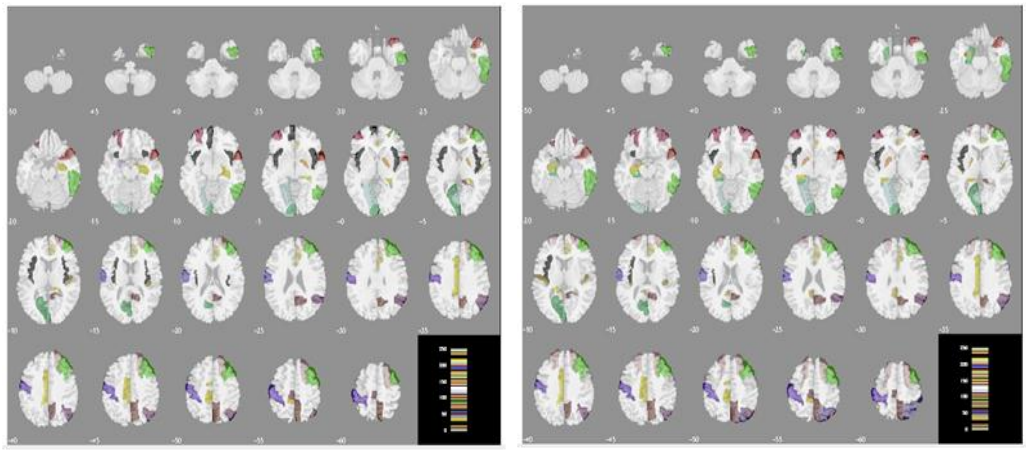
EXPLANATORY POWER OF FUNCTIONAL AND STRUCTURAL MEASUREMENTS FOR SEVERITY OF COGNITIVE IMPAIRMENT

Brain regions	**PET**		**MRI**		**Brain regions**	**PET**		**MRI**	
	R^2	p-value	R^2	p-value		R^2	p-value	R^2	p-value
Precentral_L	0.003	0.503	0.027	0.077	Amygdala_L	0.090	0.001	0.313	$<10^{-4}$
Precentral_R	0.044	0.022	~	~	Calcarine_L	0.038	0.034	0.028	0.070
Frontal_Sup_L	0.051	0.013	0.047	0.018	Lingual_L	0.066	0.005	0.044	0.023
Frontal_Sup_R	0.044	0.023	~	~	Postcentral_L	0.038	0.035	0.026	0.081
Frontal_Mid_R	0.056	0.010	0.072	0.003	Parietal_Sup_R	0.001	0.677	~	~
Frontal_M_O_L	0.036	0.040	0.086	0.001	Angular_R	0.173	$<10^{-4}$	0.063	0.006
Frontal_M_O_R	0.019	0.138	0.126	0.000	Precuneus_R	0.063	0.006	0.025	0.084
Insula_L	0.016	0.171	0.163	$<10^{-4}$	Paracentr_Lobu_L	0.035	0.043	0.000	0.769
Insula_R	~	~	0.125	0.000	Pallidum_L	0.082	0.001	~	~
Cingulum_A_R	0.004	0.497	0.082	0.001	Pallidum_R	~	~	0.020	0.122
Cingulum_Mid_L	0.001	0.733	0.040	0.030	Heschl_L	0.001	0.640	~	~
Cingulum_Post_L	0.184	$<10^{-4}$	~	~	Heschl_R	0.000	0.744	0.111	0.000
Hippocampus_L	0.158	$<10^{-4}$	~	~	Temporal_P_S_R	0.008	0.336	0.071	0.003
Hippocampus_R	~	~	0.242	$<10^{-4}$	Temporal_Inf_R	0.187	$<10^{-4}$	0.147	$<10^{-4}$
ParaHippocamp_L	0.206	$<10^{-4}$	~	~	All regions	0.702	$<10^{-4}$	0.497	$<10^{-4}$

Notation "~" means this region is not identified from PET (or MRI) as a disease-related region by SCLDA

One major observation is that the identified disease-related brain regions from MRI are in the hippocampus, parahippocampus, temporal lobe, frontal lobe, and precuneus, which is consistent with the existing literature that reports structural atrophy in these brain areas [3] [4] [5] [6] [12] [13] [14]. The identified disease-related brain regions from PET are in the temporal, frontal, and parietal lobes, which is consistent with many functional neuroimaging studies that report reduced functional activities in these areas [8] [9] [10] [12] [13] [14]. Many of these identified disease-related regions can be explained in terms of the AD pathology. For example, hippocampus is a region affected by AD the earliest

and severely [6]. Also, because regions in the temporal lobe are essential for memory, damage on these regions by AD can explain the memory loss which is a major clinical symptom of AD. The consistency of our findings with the AD literature supports effectiveness of the proposed SCLDA.



(a) (b)

Fig. 6-4. (a) Locations of disease-related brain regions identified from MRI; (b) locations of disease-related brain regions identified from FDG-PET.

Another finding is that there is a large overlap between the identified disease-related regions from PET and those from MRI, which implies strong interaction between functional and structural alterations in these regions. Although well-accepted biological mechanisms underlying this interaction are still not very clear, there are several explanations existing in the literature [14] [45]. For example, one explanation is that both functional and structural alterations could be the consequence of dendritic arborizations, which results from intracellular accumulation of PHFtau and further leads to neuron death and grey matter loss.

6.5.3 Classification accuracy

While SCLDA is a method for variable selection, the selected variables can be input into a classifier for classification. Specifically, the selected disease-related brain regions from

PET and those from MRI can be input into a classifier for discrimination between AD and NC. The classification accuracy, in turn, can help evaluate the performance of SCLDA in identification of disease-related brain regions. For the purpose of comparison, MSLDA is also used to identify disease-related brain regions from PET and MRI, which are further input into the same classifier as that following SCLDA.

More specifically, the data is randomly divided into a training set (with 90% of the entire data) and a test set (with 10% of the entire data). Each approach (MSLDA or SCLDA) is applied to the training set to identify disease-related brain regions. Afterwards, a linear SVM (Support Vector Machine) classifier is applied to the identified regions and the classification accuracy is computed on the test set. This procedure is repeated for 100 times. On average, SCLDA selects 26 and 21 disease-related brain regions from PET and MRI, respectively, whereas MSLDA selects 45 and 38 regions from PET and MRI, respectively. The classification accuracies of SVM corresponding to SCLDA and MSLDA are shown in Fig. 6-5. A two-sample t-test shows that SCLDA has a significantly higher accuracy than MSLDA with p-value < 0.05. This result also con firms that, compared with SCLDA, MSLDA identifies a much larger number of disease-related brain regions which may contain some regions that are indeed disease-irrelevant, so that inclusion of them may deteriorate the classification accuracy.

6.5.4 Relationship between PET and MRI measurements, and severity of cognitive impairment

In addition to classification, it is also of interest to further verify relevance of the identified disease-related regions with AD in an alternative way. One approach is to investigate the degree to which these disease-related regions are relevant to cognitive impairment that can be measured by the Alzheimer's disease assessment scale – cognitive subscale (ADAS-cog). ADAS measures severity of the most important symptoms of AD,

and its subscale, ADAS-cog, is the most popular cognitive testing instrument used in clinic trails. ADAS-cog consists of 11 items measuring disturbances of memory, language, praxis, attention, and other cognitive abilities that are often affected by AD. As the total score of these 11 items provides an overall assessment of cognitive impairment, we regress this ADAS-cog total score (the response) against the PET or MRI measurement of each identified brain region, using a simple regression. The regression results are listed in Table 6-1.

It is not surprising to find that some regions in the hippocampus area and temporal lobes are among the best predictors, as these regions are extensively reported in the literature as the most severely affected by AD [3] [4] [5] [6]. Also, it is found that most of these brain regions are weak-effect predictors, as most of them can only explain a small portion of the variability in the ADAS-cog total score, i.e., many R^2 values in Table 6-1 are less than 10%. However, although the effects are weak, most of them are significant, i.e., most of the p-values in Table 6-1 are smaller than 0.05. Furthermore, it is worth noting that 70.22% variability in ADAS-cog can be explained by taking all the 26 brain regions identified from PET as predictors in a multiple regression model, and 49.72% variability can be explained by taking all the 21 brain regions from MRI as predictors in a multiple regression model. All this findings imply that the disease-related brain regions are indeed weak-effect features if considered individually, but jointly they can play a strong role for characterizing AD. This verifies the suitability of the proposed SCLDA for AD studies, as SCLDA can preserve weak-effect features.

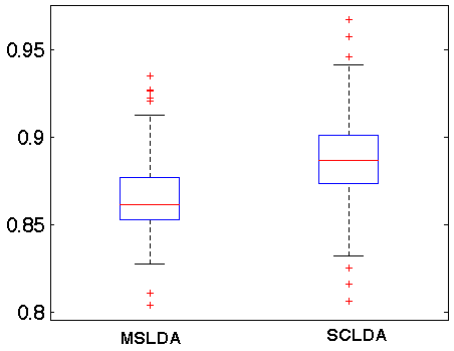


Fig. 6-5. Boxplots of classification accuracies for MSLDA and SCLDA (100 repetitions in cross-validation)

6.6 Conclusion

In the paper, we proposed a SCLDA model for fusing multi-modality neuroimaging data. In the proposed SCLDA formulation, each LDA parameter was decomposed into a common parameter shared by all the imaging modalities, multiplied by a parameter specific to each modality. We further demonstrated that this formulation is equivalent to a penalized likelihood with non-convex regularization, which can be solved by the DC programming. Simulation studies were performed, showing that SCLDA performs better than two completing methods, SLDA and MSLDA, both of which adopt convex regularizations. Finally, we applied SCLDA to the PET and MRI data of early AD patients and normal controls. Early AD is a stage at which the disease-related brain regions are most likely to be weak-effect regions that are difficult to be detected from MRI or PET alone. Our result showed that by exploiting the interaction between MRI and PET and borrowing strength from each other, SCLDA was able to identify disease-related brain regions that are consistent with the AD literature. Classification based on the brain regions identified by SCLDA also shows higher accuracy than those identified by MSLDA. Potential future work includes investigation of statistical significance of the identified features. Some potential methods include bootstrap, permutation tests, and

stability selection. Also, we will investigate asymptotic properties of SCLDA, to see if SCLDA is asymptotically consistent and with what rate, the consistency can be achieved.

Appendix

The proof of Theorem 1 is shown here.

Proof: Recall that $\boldsymbol{\Theta}$ consists of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Psi}$. Our basic idea in proving Theorem 1 is the following: we first prove that any local optima of $l_1\left(\boldsymbol{\Gamma}, \boldsymbol{\Psi} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$, denoted by $\hat{\boldsymbol{\Gamma}}$ and $\hat{\boldsymbol{\Psi}}$, corresponds to a local optima of

$l_1'\left(\boldsymbol{\Gamma}, \boldsymbol{\Psi} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) =$

$\sum_{m=1}^{M}\left\{\frac{N^{(m)}}{2}\log\left|{\boldsymbol{\theta}_{p-q}^{(m)}}^{T}\mathbf{T}^{(m)}\boldsymbol{\theta}_{p-q}^{(m)}\right| + \sum_{j=1}^{J}\frac{N_j}{2}\log \boldsymbol{\theta}_q^{(m)}\mathbf{W}_j^{(m)}\boldsymbol{\theta}_q^{(m)} - N^{(m)}\log\left|\boldsymbol{\theta}^{(m)}\right|\right\} +$

$\sum_k \delta_k + \eta\sum_k\sum_{l=1}^{q}\sum_{m=1}^{M}\gamma_{k,l}^{(m)}$,

which is denoted $\tilde{\boldsymbol{\Gamma}}, \tilde{\boldsymbol{\Psi}}$. To show this, we insert $\tilde{\boldsymbol{\Gamma}}, \tilde{\boldsymbol{\Psi}}$ into l_1', which is

$l_1'\left(\tilde{\boldsymbol{\Gamma}}, \tilde{\boldsymbol{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$,

$= \sum_{m=1}^{M}\left\{\frac{N^{(m)}}{2}\log\left|{\tilde{\boldsymbol{\theta}}_{p-q}^{(m)}}^{T}\mathbf{T}^{(m)}\tilde{\boldsymbol{\theta}}_{p-q}^{(m)}\right| + \sum_{j=1}^{J}\frac{N_j}{2}\log \tilde{\boldsymbol{\theta}}_q^{(m)}\mathbf{W}_j^{(m)}\tilde{\boldsymbol{\theta}}_q^{(m)} - N^{(m)}\log\left|\tilde{\boldsymbol{\theta}}^{(m)}\right|\right\} +$

$\sum_k \tilde{\delta}_k + \eta\sum_k\sum_{l=1}^{q}\sum_{m=1}^{M}\tilde{\gamma}_{k,l}^{(m)}$,

which can be further written as

$l_1'\left(\tilde{\boldsymbol{\Gamma}}, \tilde{\boldsymbol{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) =$

$\sum_{m=1}^{M}\left\{\frac{N^{(m)}}{2}\log\left|{\tilde{\boldsymbol{\theta}}_{p-q}^{(m)}}^{T}\mathbf{T}^{(m)}\tilde{\boldsymbol{\theta}}_{p-q}^{(m)}\right| + \sum_{j=1}^{J}\frac{N_j}{2}\log \tilde{\boldsymbol{\theta}}_q^{(m)}\mathbf{W}_j^{(m)}\tilde{\boldsymbol{\theta}}_q^{(m)} - N^{(m)}\log\left|\tilde{\boldsymbol{\theta}}^{(m)}\right|\right\} +$

$\lambda_1\sum_k\frac{\tilde{\delta}_k}{\lambda_1} + \lambda_2\sum_k\sum_{l=1}^{q}\sum_{m=1}^{M}\lambda_1\tilde{\gamma}_{k,l}^{(m)}$,

$= l_1\left(\lambda_1\tilde{\boldsymbol{\Gamma}}, \frac{\tilde{\boldsymbol{\Psi}}}{\lambda_1} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) \geq l_1\left(\hat{\boldsymbol{\Gamma}}, \hat{\boldsymbol{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$. The last inequality holds since $\hat{\boldsymbol{\Gamma}}$ and $\hat{\boldsymbol{\Psi}}$ is a local optima of l_1. Similarly, we insert $\hat{\boldsymbol{\Gamma}}, \hat{\boldsymbol{\Psi}}$ into l_1 and we can prove

$l_1\left(\hat{\mathbf{\Gamma}}, \widehat{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) = l_1'\left(\frac{\hat{\mathbf{\Gamma}}}{\lambda_1}, \lambda_1 \widehat{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) \geq$

$l_1'\left(\tilde{\mathbf{\Gamma}}, \widetilde{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$,

Therefore, we have

$l_1\left(\hat{\mathbf{\Gamma}}, \widehat{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) = l_1'\left(\frac{\hat{\mathbf{\Gamma}}}{\lambda_1}, \lambda_1 \widehat{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) =$

$l_1'\left(\tilde{\mathbf{\Gamma}}, \widetilde{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$.

This has an implication that $\left\{\frac{\hat{\mathbf{\Gamma}}}{\lambda_1}, \lambda_1 \widehat{\mathbf{\Psi}}\right\}$ is also a local optimizer of $l_1'\left(\mathbf{\Gamma}, \mathbf{\Psi} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$ and $\tilde{\delta}_k \tilde{\gamma}_{k,l}^{(m)} = \hat{\delta}_k \hat{\gamma}_{k,l}^{(m)}$, $\eta = \lambda_1 \lambda_2$.

Now we demonstrate that any local optima of $l_1'\left(\mathbf{\Gamma}, \mathbf{\Psi} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$ corresponds to a local optima of $l_2\left(\mathbf{\Theta} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$, $\widehat{\mathbf{\Theta}}$. Using the same idea above, we can obtain that

$l_1'\left(\tilde{\mathbf{\Gamma}}, \widetilde{\mathbf{\Psi}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$,

$\geq \sum_{m=1}^{M} \left\{ \frac{N^{(m)}}{2} \log \left| \tilde{\boldsymbol{\theta}}_{p-q}^{(m)\,T} \mathbf{T}^{(m)} \tilde{\boldsymbol{\theta}}_{p-q}^{(m)} \right| + \sum_{j=1}^{J} \frac{N_j}{2} \log \tilde{\boldsymbol{\theta}}_q^{(m)} \mathbf{W}_j^{(m)} \tilde{\boldsymbol{\theta}}_q^{(m)} - N^{(m)} \log \left| \tilde{\boldsymbol{\theta}}^{(m)} \right| \right\} +$

$2 \sum_k \sqrt{\eta \tilde{\delta}_k \sum_{l=1}^{q} \sum_{m=1}^{M} \left| \tilde{\gamma}_{k,l}^{(m)} \right|}$,

$= l_2\left(\widetilde{\mathbf{\Theta}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right) \geq l_2\left(\widehat{\mathbf{\Theta}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$.

On the other hand, let $\widehat{\mathbf{\Psi}} = \left\{ \hat{\delta}_k = \sqrt{\eta \sum_{l=1}^{q} \sum_{m=1}^{M} \left| \hat{\theta}_{k,l}^{(m)} \right|}, 1 \leq k \leq p \right\}$ and $\hat{\mathbf{\Gamma}} = \left\{ \hat{\gamma}_{k,l}^{(m)} = \frac{\hat{\theta}_{k,l}^{(m)}}{\hat{\delta}_k}, 1 \leq k \leq p,\ 1 \leq l \leq p, 1 \leq m \leq M \right\}$, then we can obtain

$l_2\left(\widehat{\mathbf{\Theta}} | \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\right\}\right)$,

$= \sum_{m=1}^{M} \left\{ \frac{N^{(m)}}{2} \log \left| \hat{\boldsymbol{\theta}}_{p-q}^{(m)\,T} \mathbf{T}^{(m)} \hat{\boldsymbol{\theta}}_{p-q}^{(m)} \right| + \sum_{j=1}^{J} \frac{N_j}{2} \log \hat{\boldsymbol{\theta}}_q^{(m)} \mathbf{W}_j^{(m)} \hat{\boldsymbol{\theta}}_q^{(m)} - N^{(m)} \log \left| \hat{\boldsymbol{\theta}}^{(m)} \right| \right\} +$

$\sum_k \sqrt{\eta \sum_{l=1}^{q} \sum_{m=1}^{M} \left| \hat{\theta}_{k,l}^{(m)} \right|} + \sqrt{\eta} \sum_k \sum_{l=1}^{q} \sum_{m=1}^{M} \frac{\left| \hat{\theta}_{k,l}^{(m)} \right|}{\sqrt{\sum_{l=1}^{q} \sum_{m=1}^{M} \left| \hat{\theta}_{k,l}^{(m)} \right|}}$,

$= \sum_{m=1}^{M} \left\{ \frac{N^{(m)}}{2} \log \left| \hat{\boldsymbol{\theta}}_{p-q}^{(m)\,T} \mathbf{T}^{(m)} \hat{\boldsymbol{\theta}}_{p-q}^{(m)} \right| + \sum_{j=1}^{J} \frac{N_j}{2} \log \hat{\boldsymbol{\theta}}_{q}^{(m)} \mathbf{W}_j^{(m)} \hat{\boldsymbol{\theta}}_{q}^{(m)} - N^{(m)} \log \left| \hat{\boldsymbol{\theta}}^{(m)} \right| \right\} +$

$\sum_k \hat{\delta}_k + \lambda \sum_k \sum_{l=1}^{q} \sum_{m=1}^{M} \hat{\gamma}_{k,l}^{(m)},$

$= l_1'\left(\widehat{\boldsymbol{\Theta}} | \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\}\right) \geq l_1'\left(\widetilde{\boldsymbol{\Theta}} | \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\}\right).$

Thus, we have

$l_2\left(\hat{\boldsymbol{\Gamma}}, \widehat{\boldsymbol{\Psi}} | \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\}\right) = l_1'\left(\hat{\boldsymbol{\Gamma}}, \widehat{\boldsymbol{\Psi}} | \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\}\right) =$

$l_1'\left(\tilde{\boldsymbol{\Gamma}}, \widetilde{\boldsymbol{\Psi}} | \{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(M)}\}\right),$

which implies that $\{\hat{\boldsymbol{\Gamma}}, \widehat{\boldsymbol{\Psi}}\}$ is also a local optimizer of $l_2\left(\boldsymbol{\Gamma}, \boldsymbol{\Psi} | \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(M)}\}\right)$,

with $\lambda = 2\sqrt{\eta} = 2\sqrt{\lambda_1 \lambda_2}$ and $\tilde{\delta}_k \tilde{\gamma}_{k,l}^{(m)} = \hat{\theta}_{k,l}^{(m)}$.

Reference

[1] L. deToledo-Morrell, T. R. Stoub, M. Bulgakova, "MRI-derived entorhinal volume is a good predictor of conversion from MCI to AD," *Neurobiol. Aging, vol.* 25, pp. 1197–1203, 2004.

[2] J. H. Morra, Z. Tu, "Validation of automated hippocampal segmentation method," NeuroImage, vol. 43, 59–68, 2008.

[3] Morra, J.H., Tu, Z. 2009a. Automated 3D mapping of hippocampal atrophy. *Hum. Brain Map*. 30, 2766–2788.

[4] Morra, J.H., Tu, Z. 2009b. Automated mapping of hippocampal atrophy in 1-year repeat MRI data. *NeuroImage* 45, 213-221.

[5] Schroeter, M.L., Stein, T. 2009. Neural correlates of AD and MCI. *NeuroImage* 47, 1196–1206.

[6] Braak, H., Braak, E. 1991. Neuropathological stageing of Alzheimer-related changes. *Acta Neuro*. 82, 239–259.

[7] Bradley, K.M., O'Sullivan. 2002. Cerebral perfusion SPET correlated with Braak pathological stage in AD. *Brain* 125, 1772–1781.

[8] Keilp, J.G., Alexander, G.E. 1996. Inferior parietal perfusion, lateralization, and neuropsychological dysfunction in AD. *Brain Cogn.* 32, 365–383.

[9] Schroeter, M.L., Stein, T. 2009. Neural correlates of AD and MCI. *NeuroImage* 47, 1196–1206.

[10] Asllani, I., Habeck, C. 2008. Multivariate and univariate analysis of continuous arterial spin labeling perfusion MRI in AD. J. *Cereb. Blood Flow Metab*. 28, 725–736.

[11] Du,A.T., Jahng, G.H. 2006. Hypoperfusion in frontotemporal dementia and AD. *Neurology* 67, 1215–1220.

[12] Ishii, K., Kitagaki, H. 1996. Decreased medial temporal oxygen metabolism in AD. *J. Nucl. Med*. 37, 1159–1165.

[13] Johnson, N.A., Jahng, G.H. 2005. Pattern of cerebral hypoperfusion in AD. *Radiology* 234, 851–859.

[14] Wolf, H., Jelic, V. 2003. A critical discussion of the role of neuroimaging in MCI. *Acta Neuroal*: 107 (4), 52-76.

[15] Tosun, D., Mojabi, P. 2010. Joint analysis of structural and perfusion MRI for cognitive assessment and classification of AD and normal aging. *NeuroImage* 52, 186-197.

[16] Alsop, D., Casement, M. 2008. Hippocampal hyperperfusion in Alzheimer's disease. *NeuroImage* 42, 1267–1274.

[17] Mosconi, L., Tsui, W.-H. 2005. Reduced hippocampal metabolism in MCI and AD. *Neurology* 64, 1860–1867.

[18] Mulert, C., Lemieux, L. 2010. *EEG-fMRI: physiological basis, technique and applications*. Springer.

[19] Xu, L., Qiu, C., Xu, P. and Yao, D. 2010. A parallel framework for simultaneous EEG/fMRI analysis: methodology and simulation. *NeuroImage*, 52(3), 1123-1134.

[20] Philiastides, M. and Sajda, P. 2007. EEG-informed fMRI reveals spatiotemporal characteristics of perceptual decision making. *Journal of Neuroscience*, 27(48), 13082-13091.

[21] Daunizeau, J., Grova, C. 2007. Symmetrical event-related EEG/fMRI information fusion. *NeuroImage* 36, 69-87.

[22] Jagust, W. 2006. PET and MRI in the diagnosis and prediction of dementia. *Alzheimer's Dement* 2, 36-42.

[23] Kawachi, T., Ishii, K. and Sakamoto, S. 2006. Comparison of the diagnostic performance of FDG-PET and VBM. *Eur.J.Nucl.Med.Mol.Imaging* 33, 801-809.

[24] Matsunari, I., Samuraki, M. 2007. Comparison of 18F-FDG PET and optimized voxel-based morphometry for detection of AD. *J.Nucl.Med* 48, 1961-1970.

[25] Schmidt, M., Fung, G. and Rosales, R. 2007. Fast optimization methods for L1-regularization: a comparative study and 2 new approaches. *ECML* 2007.

[26] Liu, J., Ji, S. and Ye, J. 2009. *SLEP: sparse learning with efficient projections,* Arizona state university.

[27] Tibshirani, R. 1996. Regression Shrinkage and Selection via the Lasso, *JRSS, Series B*, 58(1):267–288.

[28] Friedman, J., Hastie, T. and Tibshirani, R. 2007. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 8(1):1–10.

[29] Zou, H., Hastie, T. and Tibshirani, R. 2006. Sparse PCA, *J. of Comp. and Graphical Statistics*, 15(2), 262-286.

[30] Qiao, Z., Zhou, L and Huang, J. 2006. Sparse LDA with applications to high dimensional low sample size data. *IAENG applied mathematics*, 39(1).

[31] Argyriou, A., Evgeniou, T. and Pontil, M. 2008. Convex multi-task feature learning. Machine Learning 73(3):243– 272.

[32] Huang, S., Li, J., et al. 2010. Learning Brain Connectivity of AD by Sparse Inverse Covariance Estimation, *NeuroImage*, 50, 935-949.

[33] Candes, E., Wakin, M. and Boyd, S. 2008. Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier analysis and applications*, 14(5), 877-905.

[34] Mazumder, R.; Friedman, J. 2009. SparseNet: Coordinate Descent with Non-Convex Penalties. Manuscript.

[35] Zhang, T. 2008. Multi-stage Convex Relaxation for Learning with Sparse Regularization. *NIPS* 2008.

[36] Campbell, N. 1984. Canonical variate analysis ageneral formulation. *Australian Jour of Stat* 26, 86–96.

[37] Hastie, T. and Tibshirani, R. 1994. *Discriminant analysis by gaussian mixtures*. Technical report. AT&T Bell Lab.

[38] Kumar, N. and Andreou, G. 1998. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communication*, 26 (4), 283-297.

[39] Gasso, G., Rakotomamonjy, A. and Canu, S. 2009. Recovering sparse signals with non-convex penalties and DC programming. *IEEE Trans. Signal Processing* 57(12), 4686-4698.

[40] Guo, J., Levina, E., Michailidis, G. and Zhu, J. 2011. Joint estimation of multiple graphical models. *Biometrika* 98 (1), 1-15.

[41] Bertsekas, D. 1982. Projected newton methods for optimization problems with simple constraints. *SIAM J. Control Optim* 20, 221-246.

[42] Clemmensen, L., Hastie, T., Witten, D. and Ersboll, B. 2011. Sparse Discriminant Analysis. Technometrics (in press)

[43] Friston, K.J., Ashburner, J. 1995. Spatial registration and normalization of images. *HBM* 2, 89–165.

[44] Tzourio-Mazoyer, N., et al., 2002. Automated anatomical labelling of activations in SPM. *NeuroImage* 15, 273–289.

[45] Bidzan, L. 2005. Vascular factors in dementia. *Psychiatr. Pol*. 39, 977-986.

[46] Chiang, L., Russell, E.R. 2001. *Fault detection and diagnosis in industrial systems*. Springer.

Chapter 7

REGRESSION-BASED PROCESS MONITORING WITH CONSIDERATION OF MEASUREMENT ERRORS

Abstract

Multivariate process monitoring and fault detection is an important problem in quality improvement. Most existing methods are based on a common assumption that the measured values of variables are the true values, with limited consideration on various types of measurement errors embedded in data. On the other hand, research on measurement errors has been conducted from a pure theoretical statistics point of view, without linking the modeling and analysis of measurement errors with monitoring and fault detection objectives. This paper proposes a method for multivariate process monitoring and fault detection considering four types of major measurement errors, including sensor bias, sensitivity, noise, and dependency of the relationship between a variable and its measured value on some other variables. This method includes design of new control charts based on data with measurement errors, and identification of the maximum allowable measurement errors to fulfill certain fault detectability requirements. This method is applicable to processes where a nature order of the variables is known, such as the cascade or multistage processes, and processes where the causal relationships among variables are known and can be described by a Bayesian network. The method is demonstrated in two industrial processes.

7.1 Introduction

Rapid advances in sensors and distributed sensing technologies have resulted in data-rich environments, creating unprecedented opportunities for quality improvement in many domains [8]. To make full use of the data for quality improvement, various statistical and data mining methods have been developed for process monitoring and fault

detection. Most of these methods rely on a common assumption that the measured values of variables are the true values, with limited consideration of various types of measurement errors embedded in the data. As a result, quite a few theoretically sound methods may be found less effective when applied to real-world applications, in which the data are far less perfect than they have been assumed to be.

While measurement errors may hinder quality improvement objectives to be fully realized in many domains, very limited studies have been conducted on how to take the measurement errors into consideration in developing methods in process monitoring and fault detection. Research on measurement errors has been focused on estimating various types of measurement errors from data, and identifying variable relationships based on statistical modeling and inference [2, 3, 4, 6, 7, 9, 26, 31]. This type of research usually studies measurement errors from a pure theoretical statistics point of view, without linking the modeling and analysis of measurement errors with quality improvement objectives. On the other hand, research in quality engineering has resulted in abundant methods for process monitoring and fault detection [1, 10, 11, 14, 20, 23, 24, 27, 28, 29, 32, 33]. However, few of these methods have been able to take measurement errors into consideration. Research has been conducted for quality improvement of automotive assembly processes, in which the impact of measurement errors on fixture failure diagnosis has been discussed [5]. However, this research only addresses specific processes, in which CMM (Coordinate Measuring Machine) or OCMM (Optical Coordinate Measuring Machine) are used to measure dimensional features of parts. As a result, only one type of measurement errors, i.e., sensor noise, is investigated, which limits its application to general processes and domains in which the types of measurement errors can be far more complex than just sensor noise.

In this paper, we adopt a general formulation to incorporate four major types of measurement errors. This formulation is described as follows: given a system of p variables $\{X_1, \cdots, X_p\}$, the relationship between a variable X_i and its measured value $\tilde{X}_i$ can be expressed by

$$\tilde{X}_j = b_j + s_j X_j + \mathbf{c}_j^T \mathbf{V}_j + \varepsilon_j. \quad (1)$$

(1) embraces four major types of measurement errors, namely,

- b_j is the measurement error caused by sensor setup/calibration bias, or drifting when sensors are used in harsh environments.
- s_j represents measurement sensitivity, $s_j > 1$ and $s_j < 1$ reflect the "scale-up" and "scale-down" errors commonly exiting in many types of sensors.
- $\mathbf{c}_j$ considers that the relationship between $\tilde{X}_j$ and X_j also depends on other variables $\mathbf{V}_j$ (i.e., covariates), where $\mathbf{V}_j \subset \{X_1, \cdots, X_p\}$ and $X_j \notin \mathbf{V}_j$ [4]. For example, when imaging sensors are used to detect product defects in hot rolling processes [17, 19], the measured defect features, i.e., $\tilde{X}_j$, may be corrupted by measurement errors. And these measurement errors may have different distributions with respect to different textures of the product materials and different rolling temperatures, i.e., $\mathbf{V}_j$, because these two factors impact performance of the imaging sensors.
- $\varepsilon_j \sim N\left(0, var(\varepsilon_j)\right)$ accounts for sensor noise; in other words, precision of the sensor is reflected by $var(\varepsilon_j)$.

In addition, we focus on studying the impact of measurement errors on regression-based methods in process monitoring and fault detection. Regression-based methods refer to those using regressions to build a model for quantitatively describing the interacting relationships among the variables in a process, called a "process model"; and

further conducting monitoring and fault detection based on the process model. For example, some researchers proposed to regress each variable in a process on all other variables [10, 23, 24], resulting in a process model that uses a number of regressions (equal to the number of variables) to reveal the correlation structure among variables. Some other researchers proposed to integrate regression analysis with various types of domain knowledge. For instance, when a natural order of variables is known, such as the variables in a cascade or multistage process, each variable can be regressed on its upstream variables, leading to a process model that facilitates more effective monitoring and fault detection [11, 28, 29, 32, 33]. Also, when the causal influences among variable are known and represented by a Bayesian network [13, 16, 18, 21], Li et al. [20] proposed to regress each variable on all its parent variables (i.e., the direct causes), leading to further improvement in diagnostic accuracy and reduction in computational complexity. However, despite the popularity of regression-based methods in process monitoring and fault detection, the impact of measurement errors on these methods has been little discussed.

In this paper, we focus on the regression-based methods for processes where a natural order of variables is known (i.e., cascade or multistage processes) and processes that can be described by a Bayesian network. We adopt a unified representation for these two types of processes, based on which we further develop methods for monitoring and fault detection considering the four types of measurement errors defined in (1).

The rest of the paper is organized as follows: Section 7-2 presents the unified representation for the two types of processes, and the monitoring and fault detection steps in general regression-based methods (i.e., methods when there are no measurement errors in the data); Section 7-3 proposes a new method for monitoring and fault detection considering the four types of measurement errors defined in (1); Section 7-4 studies how

to identify the maximum allowable measurement errors under given fault detectability requirements; Section 7-5 shows two examples in real industrial processes; Section 7-6 gives the conclusion.

Before going into these sections, some general notations are introduced here. A quantity with an overhead "~" implies that this quantify is measured with errors. For example, the true value and measured value of a variable in the system are denoted by X_j and $\tilde{X}_j$, respectively. A quantify with an overhead " – " implies that this quantify is a sample average. A quantify with an overhead "^" implies that this quantify is an estimator. For example, $\hat{E}({}^{r}e_i)$ is an estimator for $E({}^{r}e_i)$. The overheads "~" and "^" may be used together. For example, $\hat{\tilde{E}}({}^{r}e_i)$ represents an estimator for $E({}^{r}e_i)$ and is measured with errors. In addition, a letter may have a right subscript and left superscripts. The right subscript represents numbering and the left superscripts represent layers (the concept of "layer" will be introduced in Section 7-2). For example, ${}^{r}X_i$ denotes the i-th variable on layer_r.

7.2 Monitoring and fault detection in general regression-based methods under a unified process representation

In this section, we introduce the general framework of monitoring and fault detection when there is no measurement error in the data (i.e., what traditional regression-based methods assume). Then, in the next section (Section 7-3), we develop a new method to integrate measurement errors into this framework.

This paper focuses on cascade or multistage processes, and processes that can be represented by a Bayesian network. These two types of processes are introduced as follows: In a cascade or multistage process, variables have a natural ordering in which if any variable undergoes a parameter shift (e.g., a mean shift), it may affect some or all the variables following it, but none of the variables preceding it in this ordering [11]. As a

result, each variable can be considered to belong to one and only one "layer". Specifically, a variable belonging to layer_1 (or called a layer_1 variable, in short) is one that has no upstream variables; a layer_r variable is one that has layer_1 through layer_($r-1$) variables as its upstream variables. Under the concept of layer, each variable in the system, $X_j, j \in \{1, \cdots, p\}$, can also be denoted by ${}^{r}X_i$ (i.e., the i-th variable belonging to layer_r), $i \in \{1, \cdots, m_r\}$, $r \in \{1, \cdots, R\}$, where m_r is the total number of variables belonging to layer_r and R is the total number of layers. For example, if a system has four variable, $\{X_1, X_2, X_3, X_4\}$; X_1 happens before X_2 and X_3; X_2 and X_3 happen simultaneously, and both happen before X_4. Then, there are three layers in the system, i.e., $R = 3$. X_1 is a layer_1 variable, so it can be denoted by ${}^{1}X_1$; X_2 and X_3 are layer_2 variables, so they can be denoted by ${}^{2}X_1$ and ${}^{2}X_2$, respectively; X_4 is a layer_3 variable, so it can be denoted by ${}^{3}X_1$.

Furthermore, the regression-based process models in a cascade or multistage process, which regress each variable on its upstream variables, can be expresses as

$$ {}^{r}X_i = \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T \; {}^{k}\mathbf{X} + \; {}^{r}e_i, \tag{2} $$

Here, ${}^{k}\mathbf{X} = \left\{ {}^{k}X_1, \cdots, {}^{k}X_{m_k} \right\}^T$ denotes the set of variables belonging to layer_k; ${}^{r,k}\boldsymbol{\beta}_i$ and ${}^{r}e_i$ are regression coefficients and residual error, respectively. Also, without loss of generality, we assume in this paper that all variables have zero means.

A process may be represented by a Bayesian network, if the causal relationships among variables are known. A Bayesian network has two components, structure and parameters [15]. The structure of a Bayesian network is a directed acyclic graph (DAG), i.e., a set of variables, $\{X_1, \ldots, X_p\}$, connected by directed arcs (see Fig. 7-7-1 for an example). A directed graph is acyclic if there is no directed path $X_i \to \cdots \to X_j$ such that $X_i = X_j$. If there is a directed arc from X_i to X_j, i.e., $X_i \to X_j$, then X_i is a

direct cause (called a parent) of X_j, where "direct" means that the causal influence from X_i to X_j is not mediated through other variables. The parameters of a Bayesian network, when all variables follow normal distributions, can be the regression coefficients by regressing each variable on its parents. The structure and parameters of a Bayesian network can be obtained by domain knowledge or by statistical learning algorithms [12, 18, 30].

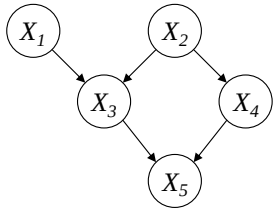


Fig. 7-1 An example of Bayesian network structure

Note that because a variable's parents must be its upstream variables, (2) can also be used to represent the regression-based process models in processes in which a Bayesian network is available, as long as the regression coefficients for ${}^{r}X_i$'s non-parents in ${}^{k}\mathbf{X}$ are set to be zero. Therefore, in this paper, we adopt (2) as a unified representation for the regression-based process models in cascade or multistage processes, as well as processes in which a Bayesian network is available.

Furthermore, given the process model, a regression-based monitoring and fault detection method usually involves two steps:

Step-one is performed based on a preliminary dataset containing data collected when the process runs under normal (i.e., no-fault) conditions. This dataset is used to estimate the regression coefficients ${}^{r,k}\boldsymbol{\beta}_i$ in (2), i.e.,

$$\begin{bmatrix} {}^{r,1}\hat{\boldsymbol{\beta}}_i \\ \vdots \\ {}^{r,r-1}\hat{\boldsymbol{\beta}}_i \end{bmatrix} = \widehat{\mathbf{cov}}^{-1}\left(\begin{bmatrix} {}^{1}\mathbf{X} \\ \vdots \\ {}^{r-1}\mathbf{X} \end{bmatrix}, \begin{bmatrix} {}^{1}\mathbf{X} \\ \vdots \\ {}^{r-1}\mathbf{X} \end{bmatrix}\right) \widehat{\mathbf{cov}}\left(\begin{bmatrix} {}^{1}\mathbf{X} \\ \vdots \\ {}^{r-1}\mathbf{X} \end{bmatrix}, {}^{r}X_i\right),$$

where $\widehat{\mathbf{cov}}(\cdot,\cdot)$ denotes the sample covariance matrix between two vectors, or a column vector of sample covariances between a vector and a scalar. Assuming that the sample size of the preliminary dataset is sufficiently large, $\widehat{\mathbf{cov}}(\cdot,\cdot)$ may be treated as its population counterpart $\mathbf{cov}(\cdot,\cdot)$, and consequently ${}^{r,k}\widehat{\boldsymbol{\beta}}_i$ may be treated as ${}^{r,k}\boldsymbol{\beta}_i$.

Step-two is performed to monitor and diagnose the process based on production data. Considering potential process faults to be mean shifts, a variable ${}^{r}X_i$ is considered to have experienced a mean shift if $E({}^{r}e_i) = {}^{r}\delta_i \neq 0$. In order to monitor $E({}^{r}e_i)$, a control chart may be built on [11]

$$\hat{E}({}^{r}e_i) = {}^{r}\overline{x}_i - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T \, {}^{k}\overline{\mathbf{x}}, \tag{3}$$

where ${}^{r}\overline{x}_i$ and ${}^{k}\overline{\mathbf{x}}$ are sample averages. It is known that $\hat{E}({}^{r}e_i)$ is an unbiased estimator for $E({}^{r}e_i)$, i.e., $E\left(\hat{E}({}^{r}e_i)\right) = E({}^{r}e_i)$, and $var\left(\hat{E}({}^{r}e_i)\right) = \frac{1}{n}\left(var({}^{r}X_i) - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T \, \mathbf{cov}({}^{k}\mathbf{X}, {}^{r}X_i)\right)$, where n is the sample size.

For example, if Shewhart control charts [25] are used, the control limits are

$$UCL = z_{\alpha/2}\sqrt{var\left(\hat{E}({}^{r}e_i)\right)}, \; CL = 0, \text{ and } LCL = -z_{\alpha/2}\sqrt{var\left(\hat{E}({}^{r}e_i)\right)},$$

where $z_{\alpha/2}$ is the upper-$\alpha/2$ percentage point of the standard normal distribution. Finally in step-two, Average Run Lengths (i.e., the average number of points it takes the control chart to generate an out-of-control signal) must be computed in order to evaluate the control chart performance. Specifically, the average run length when process is in control is $ARL_0 = 1/\alpha$; the average run length when there is a mean shift in ${}^{r}e_i$, i.e., $E({}^{r}e_i) = {}^{r}\Delta_i \neq 0$, is

$$ARL_1\left({}^r\Delta_i\right) =$$

$$1/\left(1-\Phi\left(z_{\alpha/2}-{}^r\Delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)+\Phi\left(-z_{\alpha/2}-{}^r\Delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)\right), \tag{4}$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution. Let ${}^r\delta_i$ be the magnitude of the mean shift ${}^r\Delta_i$, i.e., ${}^r\delta_i > 0$. For a positive mean shift ${}^r\Delta_i = {}^r\delta_i$, the second $\Phi(\cdot)$ in (4) will become very small. Thus, (4) becomes

$$ARL_1\left({}^r\delta_i\right) \approx 1/\left(1-\Phi\left(z_{\alpha/2}-{}^r\delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)\right). \tag{5}$$

For a negative mean shift ${}^r\Delta_i = -{}^r\delta_i$, (4) becomes

$$ARL_1\left({}^r\delta_i\right) =$$

$$1/\left(1-\Phi\left(z_{\alpha/2}+{}^r\delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)+\Phi\left(-z_{\alpha/2}+{}^r\delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)\right) \approx$$

$$1/\Phi\left(-z_{\alpha/2}+{}^r\delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right) = 1/\left(1-\Phi\left(z_{\alpha/2}-{}^r\delta_i/\sqrt{var\left(\hat{E}\left({}^re_i\right)\right)}\right)\right),$$

which is the same as (5). In other words, (5) holds for both positive and negative mean shifts.

Other types of control charts can be designed in a similar way. For example, to detect small mean shifts, EWMA control charts may be adopted [25]. In order to monitor $E\left({}^re_i\right)$, the EWMA control charts are built on $\hat{C}_t = \lambda\left\{\hat{E}\left({}^re_i\right)\right\}_t + (1-\lambda)\hat{C}_{t-1}$, where $\left\{\hat{E}\left({}^re_i\right)\right\}_t$ is the $\hat{E}\left({}^re_i\right)$ in (3) at time t and $\hat{C}_0 = 0$. The steady-state control limits for the EWMA charts are:

$$UCL = L\sqrt{\frac{\lambda}{2-\lambda}var\left(\hat{E}\left({}^re_i\right)\right)},\ CL = 0, \text{ and } LCL = -L\sqrt{\frac{\lambda}{2-\lambda}var\left(\hat{E}\left({}^re_i\right)\right)},$$

where L and λ are chosen to achieve some desired Average Run Lengths performance. The correspondence between Average Run Lengths and the values of L and λ can be found in [22].

7.3 Development of regression-based monitoring and fault detection method considering measurement errors

In this section, we adopt the two-step procedure in Section 7-2, but consider that the variables are measured with errors. Under the concept of "layer" defined in Section 7-2, an alternative representation for the relationship between a variable ${}^{r}X_i$ and its measured value ${}^{r}\tilde{X}_i$, to the representation in (1), is

$$ {}^{r}\tilde{X}_i = {}^{r}b_i + {}^{r}s_i\, {}^{r}X_i + \sum_{k=1}^{r-1} {}^{r,k}\mathbf{c}_i^T\, {}^{k}\mathbf{X} + {}^{r}\varepsilon_i. \quad (6) $$

Note that the summation in (6) is taken over only the layers before layer_r, because the relationship between ${}^{r}X_i$ and its measured value ${}^{r}\tilde{X}_i$ cannot be affected by variables happening later than ${}^{r}X_i$. Also, some coefficients in ${}^{r,k}\mathbf{c}_i$ may be zero.

In Step-one, we assume that a preliminary dataset is collected from carefully calibrated sensors whose measurement errors are negligible. As a result, the covariance matrix of variables, as well as ${}^{r,k}\boldsymbol{\beta}_i$, can be obtained.

Because the sensors used for collecting the preliminary dataset are free of measurement errors, purchasing, installing, calibrating, and maintaining the sensors usually incur significant costs. Thus, these sensors are not affordable to be used for monitoring and diagnosing continuous production in a long run. To monitor and diagnose continuous production, less costly sensors may be adopted, which, however, may generate data with substantial measurement errors.

Therefore, in Step-two, we consider that data are collected from sensors with non-negligible measurement errors, i.e., data on ${}^{r}\tilde{X}_i$, are available but data on ${}^{r}X_i$ are

not. As a result, instead of building control charts on the $\hat{E}({}^{r}e_i)$ in (3), another unbiased estimator for $E({}^{r}e_i)$ must be identified.

Obviously, the $\hat{\hat{E}}\,({}^{r}e_i)$ defined in (6) must be an unbiased estimator for $E({}^{r}e_i)$:

$$\hat{\hat{E}}\,({}^{r}e_i) = \hat{\hat{E}}({}^{r}X_i) - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T\, \hat{\hat{E}}\big({}^{k}\mathbf{X}\big), \tag{7}$$

where $\hat{\hat{E}}({}^{r}X_i)$ and $\hat{\hat{E}}\big({}^{k}\mathbf{X}\big)$ are unbiased estimators for $E({}^{r}X_i)$ and $E\big({}^{k}\mathbf{X}\big)$, respectively. In other words, $\hat{\hat{E}}\,({}^{r}e_i)$ can be found as long as $\hat{\hat{E}}({}^{r}X_i)$, $i = 1, \cdots, m_r$, $r = 1, \cdots, R$ can be obtained. When there are measurement errors, $\hat{\hat{E}}({}^{r}X_i)$ must be defined based on the data for ${}^{r}\tilde{X}_i$. Specifically, we found that $\hat{\hat{E}}({}^{r}X_i)$ can be defined based on a recursive equation given in (8):

$$\hat{\hat{E}}({}^{r}X_i) = \begin{cases} {}^{r}\overline{\tilde{y}}_i & r = 1 \\ {}^{r}\overline{\tilde{y}}_i - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{d}_i^T\, \hat{\hat{E}}\big({}^{k}\mathbf{X}\big) & r = 2, \cdots, R \end{cases}, \tag{8}$$

where ${}^{r}\overline{\tilde{y}}_i = \big({}^{r}\overline{\tilde{x}}_i - {}^{r}b_i\big)/\,{}^{r}s_i$, and ${}^{r}\overline{\tilde{x}}_i$ is the sample average for ${}^{r}\tilde{X}_i$; ${}^{r,k}\mathbf{d}_i = {}^{r,k}\mathbf{c}_i/\,{}^{r}s_i$. It is easy to prove that the $\hat{\hat{E}}({}^{r}X_i)$ in (8) is an unbiased estimator for $E({}^{r}X_i)$, so the proof is skipped. By inserting (8) into (7), an unbiased estimator for $E({}^{r}e_i)$, based on the data for ${}^{r}\tilde{X}_i$, can be obtained, i.e.,

$$\hat{\hat{E}}({}^{r}e_i) = \begin{cases} {}^{r}\overline{\tilde{y}}_i & r = 1 \\ {}^{r}\overline{\tilde{y}}_i - \sum_{k=1}^{r-1}\big({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\big)\, \hat{\hat{E}}\big({}^{k}\mathbf{X}\big) & r = 2, \cdots, R \end{cases}, \tag{9}$$

Furthermore, it can be derived that the variance for the $\hat{\hat{E}}({}^{r}e_i)$ in (9) is (see Appendix I for proof):

$$var\left(\hat{\hat{E}}({}^{r}e_i)\right) = \frac{1}{n}\Big(var({}^{r}X_i) - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T\, \mathbf{cov}\big({}^{k}\mathbf{X},\, {}^{r}X_i\big) + var({}^{r}\varepsilon_i)/\,{}^{r}s_i^2 +$$

$$\sum_{k=1}^{r-1}\sum_{j=1}^{m_k} {}^{r,k}a_{ij}^2\, var\big({}^{k}\varepsilon_j\big)/\,{}^{k}s_j^2\Big), \tag{10}$$

where ${}^{r,k}a_{ij}$ is the j-th element in vector ${}^{r,k}\mathbf{a}_i^T$, ${}^{r,k}\mathbf{a}_i^T = {}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T - \sum_{h=k+1}^{r-1}\left({}^{r,h}\mathbf{d}_i^T + {}^{r,h}\boldsymbol{\beta}_i^T\right){}^{h,k}\mathbf{W}^T$; ${}^{h,k}\mathbf{W} = \left\{{}^{h,k}\mathbf{w}_1, \cdots, {}^{h,k}\mathbf{w}_{m_h}\right\}$, and ${}^{h,k}\mathbf{w}_i^T$, $i \in \{1, \cdots, m_h\}$, is a recursive function defined as:

$${}^{h,k}\mathbf{w}_i^T = \begin{cases} {}^{h,k}\mathbf{d}_i^T & k = h-1 \\ {}^{h,k}\mathbf{d}_i^T - \sum_{t=k+1}^{h-1} {}^{h,t}\mathbf{d}_i^T \; {}^{t,k}\mathbf{W}^T & k = h-2, \cdots, 1 \end{cases}, \tag{11}$$

Finally, if Shewhart control charts are used, the control limits for $\hat{\bar{E}}({}^{r}e_i)$ are

$$\widetilde{UCL} = z_{\alpha/2}\sqrt{var\left(\hat{\bar{E}}({}^{r}e_i)\right)},\ \widetilde{CL} = 0, \text{ and } \widetilde{LCL} = -z_{\alpha/2}\sqrt{var\left(\hat{\bar{E}}({}^{r}e_i)\right)}, \tag{12}$$

and the average run lengths are $\widetilde{ARL}_0 = 1/\alpha$, and

$$\widetilde{ARL}_1({}^{r}\delta_i) \approx 1\Big/\left(1 - \phi\left(z_{\alpha/2} - {}^{r}\delta_i\Big/\sqrt{var\left(\hat{\bar{E}}({}^{r}e_i)\right)}\right)\right). \tag{13}$$

Here, to obtain the $\widetilde{ARL}_1({}^{r}\delta_i)$ in (12), we followed a procedure similar to that used to obtain the $ARL_1({}^{r}\delta_i)$ in (5).

Other types of control charts can be developed in a similar way. For example, to detect small mean shifts, EWMA control charts may be adopted. Specifically, the points on the EWMA control charts are

$$\hat{\bar{C}}_t = \lambda\left\{\hat{\bar{E}}({}^{r}e_i)\right\}_t + (1-\lambda)\hat{\bar{C}}_{t-1}, \tag{14}$$

where $\left\{\hat{\bar{E}}({}^{r}e_i)\right\}_t$ is the $\hat{\bar{E}}({}^{r}e_i)$ in (9) at time t and $\hat{\bar{C}}_0 = 0$. The steady-state control limits for the EWMA charts are:

$$UCL = L\sqrt{\frac{\lambda}{2-\lambda}var\left(\hat{\bar{E}}({}^{r}e_i)\right)}, \quad CL = 0, \quad \text{and}$$

$$LCL = -L\sqrt{\frac{\lambda}{2-\lambda}var\left(\hat{\bar{E}}({}^{r}e_i)\right)}, \tag{15}$$

where L and λ are chosen to achieve some desired Average Run Lengths performance. The correspondence between Average Run Lengths and the values of L and λ can be found in [22].

7.4 Identification of maximum allowable measurement errors under given fault detectability requirements

Because timely detection of mean shifts are always required, there should be an upper bound for $\widetilde{ARL}_1({}^r\delta_i)$, denoted by $\widetilde{ARL}_{1U}({}^r\delta_i)$ which is set according to specific domain standards. In other words, the mean shift ${}^r\delta_i$ is considered to be *detectable* only if it can be detected within a required time period, i.e.,

$$\widetilde{ARL}_1({}^r\delta_i) \le \widetilde{ARL}_{1U}({}^r\delta_i). \tag{16}$$

According to (13), (16) holds if and only if $1/\left(1-\phi\left(z_{\alpha/2}-{}^r\delta_i/\sqrt{var\left(\hat{\tilde{E}}({}^re_i)\right)}\right)\right) \le \widetilde{ARL}_{1U}({}^r\delta_i)$, i.e.,

$$var\left(\hat{\tilde{E}}({}^re_i)\right) \le \left(\frac{{}^r\delta_i}{z_{\alpha/2}-\phi^{-1}\left(1-1/\widetilde{ARL}_{1U}({}^r\delta_i)\right)}\right)^2. \tag{17}$$

By inserting (10) into (17), (18) can be obtained:

$$\frac{var({}^r\varepsilon_i)}{{}^rs_i^2}+\sum_{k=1}^{r-1}\sum_{j=1}^{m_k}{}^{r,k}a_{ij}^2\frac{var({}^k\varepsilon_j)}{{}^ks_j^2} \le$$

$$n\left(\frac{{}^r\delta_i}{z_{\alpha/2}-\phi^{-1}\left(1-1/\widetilde{ARL}_{1U}({}^r\delta_i)\right)}\right)^2-\left(var({}^rX_i)-\sum_{k=1}^{r-1}{}^{r,k}\boldsymbol{\beta}_i^T\,\mathbf{cov}({}^k\mathbf{X},\,{}^rX_i)\right), \tag{18}$$

where the left-hand side is a function of the measurement errors. In other words, the measurement errors must be confined by (18) in order for the mean shift in ${}^rX_i$, i.e., ${}^r\delta_i$, to be detectable.

Consider a special case in which only sensor noise exist, i.e., ${}^rs_i = 1$ and ${}^{r,k}\mathbf{c}_i = \mathbf{0}$ in (6). Then (18) becomes

$$var(\,{}^{r}\varepsilon_i) + \sum_{k=1}^{r-1}\sum_{j=1}^{m_k} {}^{r,k}\beta_{ij}^2\, var\left({}^{k}\varepsilon_j\right) \;\le$$

$$n\left(\frac{{}^{r}\delta_i}{z_{\alpha/2}-\phi^{-1}\left(1-1/\widehat{ARL}_{1U}\left({}^{r}\delta_i\right)\right)}\right)^2 - \left(var\left({}^{r}X_i\right) - \textstyle\sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T\, \mathbf{cov}\left({}^{k}\mathbf{X},\ {}^{r}X_i\right)\right), \qquad (19)$$

where ${}^{r,k}\beta_{ij}$ is the j-th element in vector ${}^{r,k}\boldsymbol{\beta}_i^T$. The right-hand side of (19) gives the maximum level of sensor noise allowed, in order for the mean shift in ${}^{r}X_i$, i.e., ${}^{r}\delta_i$, to be detectable.

7.5 Examples

Two examples will be shown in this section: Example I aims to demonstrate the proposed method in monitoring and diagnosing a cotton spinning process – a cascade process; Example II aims to identify the maximum allowable measurement errors for a hot forming process – a process that can be represented by a Bayesian network.

7.5.1 Example I: A cotton spinning process

Description on the process, process models, and measurement errors

This example targets a cotton spinning process, a cascade process [11] in which the variables consist of X_1 (fiber fineness), X_2 (fiber length), X_3 (fiber strength), and X_4 (skein strength). This process has three layers: X_1 and X_2 belong to layer 1, X_3 belongs to layer 2, and X_4 and belongs to layer 3. Therefore, X_1, X_2, X_3, and X_4 can also be denoted as ${}^{1}X_1$, ${}^{1}X_2$, ${}^{2}X_1$, and ${}^{3}X_1$, respectively.

Furthermore, according to (2), the regression-based process models are

$$\begin{aligned} {}^{1}X_1 &= {}^{1}e_1 \\ {}^{1}X_2 &= {}^{1}e_2 \\ {}^{2}X_1 &= {}^{2,1}\beta_{11}\,{}^{1}X_1 + {}^{2,1}\beta_{12}\,{}^{1}X_2 + {}^{2}e_1 \\ {}^{3}X_1 &= {}^{3,1}\beta_{11}\,{}^{1}X_1 + {}^{3,1}\beta_{12}\,{}^{1}X_2 + {}^{3,2}\beta_{11}\,{}^{2}X_1 + {}^{3}e_1 \end{aligned} \qquad (20)$$

where all variables follow the standard normal distribution and the regression coefficients are ${}^{2,1}\beta_{11} = -0.16,\ {}^{2,1}\beta_{12} = 0,\ {}^{3,1}\beta_{11} = -0.343,\ {}^{3,1}\beta_{12} = 0.606,$ and ${}^{3,2}\beta_{11} = 0.352$ [11].

In this example, we assume the following measurement errors:

- ${}^{r}b_i = b$, i.e., the same setup/calibration bias for all sensors.
- ${}^{r}s_i = s$, i.e., the same measurement sensitivity for all sensors.
- $var({}^{r}\varepsilon_i) = \sigma_\varepsilon^2$, i.e., the same precision for all sensors.
- ${}^{2,1}c_{11} = {}^{2,1}c_{12} = {}^{3,1}c_{11} = {}^{3,1}c_{12} = {}^{3,2}c_{11} = c$, i.e., the relationship between a variable and its measured value depends on all variables at preceding layers, and the same strength of the dependency is assumed across all variables.

According to the measurement errors above, the relationship between a variable and its measured value becomes

$$\begin{aligned} {}^{1}\tilde{X}_1 &= b + s\ {}^{1}X_1 + {}^{1}\varepsilon_1 \\ {}^{1}\tilde{X}_2 &= b + s\ {}^{1}X_2 + {}^{1}\varepsilon_2 \\ {}^{2}\tilde{X}_1 &= b + s\ {}^{2}X_1 + c\ {}^{1}X_1 + c\ {}^{1}X_2 + {}^{2}\varepsilon_1 \\ {}^{3}\tilde{X}_1 &= b + s\ {}^{3}X_1 + c\ {}^{1}X_1 + c\ {}^{1}X_2 + c\ {}^{2}X_1 + {}^{3}\varepsilon_1 \end{aligned} \tag{21}$$

and $var({}^{1}\varepsilon_1) = var({}^{1}\varepsilon_2) = var({}^{2}\varepsilon_1) = var({}^{3}\varepsilon_1) = \sigma_\varepsilon^2$.

Process monitoring and detection of large mean shifts

To demonstrate the effectiveness of the proposed method in detecting large mean shifts, $\widetilde{ARL}_1({}^{r}\delta_i)$ may be computed using (13). It can be seen from (13) that the detectability (measured by $\widetilde{ARL}_1({}^{r}\delta_i)$) of a control chart is positively affected by ${}^{r}\delta_i / \sqrt{var\left(\hat{\tilde{E}}({}^{r}e_i)\right)}$, i.e., the larger the ${}^{r}\delta_i / \sqrt{var\left(\hat{\tilde{E}}({}^{r}e_i)\right)}$, the higher the detectability. To demonstrate the detectability of the proposed method, we introduce a mean shift of one standard deviation into each variable, and calculate the magnitude of

the induced mean shift in every ${}^{r}\delta_i/\sqrt{var\left(\hat{\bar{E}}\left({}^{r}e_i\right)\right)}$. The results are shown in Table 7-7-1, with $n = 1$, $b/s = 1$, $c/s = 1$, and $\sigma_\varepsilon/s = 1$. It can be seen that the detectability of the proposed method for variables on earlier layers (e.g., ${}^{1}X_1$ and ${}^{1}X_2$) is better than that for variables on later layers (e.g., ${}^{2}X_1$ and ${}^{3}X_1$). This is because variables on later layers are not only subject to measurement errors associated with these variables themselves, but also subject to measurement errors associated with variables at earlier layers as these measurement errors will propagate downstream.

Table 7-1 Mean shift induced in ${}^{r}\delta_i/\sqrt{var\left(\hat{\bar{E}}\left({}^{r}e_i\right)\right)}$ by shifting each variable

	$\frac{{}^{1}\delta_1}{\sqrt{var\left(\hat{\bar{E}}\left({}^{1}e_1\right)\right)}}$	$\frac{{}^{1}\delta_2}{\sqrt{var\left(\hat{\bar{E}}\left({}^{1}e_2\right)\right)}}$	$\frac{{}^{2}\delta_1}{\sqrt{var\left(\hat{\bar{E}}\left({}^{2}e_1\right)\right)}}$	$\frac{{}^{3}\delta_1}{\sqrt{var\left(\hat{\bar{E}}\left({}^{3}e_1\right)\right)}}$
${}^{1}\delta_1 = 1$	0.71	0	0	0
${}^{1}\delta_2 = 1$	0	0.71	0	0
${}^{2}\delta_1 = 1$	0	0	0.52	0
${}^{3}\delta_1 = 1$	0	0	0	0.51

For purpose of comparison, we develop a similar table to Table 7-1, i.e., Table 7-2, by applying the traditional regression-based method on the data with measurement errors. This method ignores the measurement errors and uses the measured values of each variable as the true values. Therefore, the charting statistic in each control chart is ${}^{1}\tilde{e}_1 = {}^{1}\bar{\tilde{x}}_1$, ${}^{1}\tilde{e}_2 = {}^{1}\bar{\tilde{x}}_2$, ${}^{2}\tilde{e}_1 = {}^{2}\bar{\tilde{x}}_1 - \left({}^{2,1}\beta_{11}\,{}^{1}\bar{\tilde{x}}_1 + {}^{2,1}\beta_{12}\,{}^{1}\bar{\tilde{x}}_2\right)$, and ${}^{3}\tilde{e}_1 = {}^{3}\bar{\tilde{x}}_1 - \left({}^{3,1}\beta_{11}\,{}^{1}\bar{\tilde{x}}_1 + {}^{3,1}\beta_{12}\,{}^{1}\bar{\tilde{x}}_2 + {}^{3,2}\beta_{11}\,{}^{2}\bar{\tilde{x}}_1\right)$, respectively. The results are shown in Table 7-7-2. Table 7-2 clearly shows that a shift in a variable may create shifts in variables other than itself. This implies that the traditional method may potentially generate too many false alarms, while the proposed method doe not suffer from this problem.

Table 7-2 Mean shift induced in $E(\,^{r}\tilde{e}_i)/\sqrt{var(\,^{r}\tilde{e}_i)}$ by shifting each variable

	$\frac{E(\,^{1}\tilde{e}_1)}{\sqrt{var(\,^{1}\tilde{e}_1)}}$	$\frac{E(\,^{1}\tilde{e}_2)}{\sqrt{var(\,^{1}\tilde{e}_2)}}$	$\frac{E(\,^{2}\tilde{e}_1)}{\sqrt{var(\,^{2}\tilde{e}_1)}}$	$\frac{E(\,^{3}\tilde{e}_1)}{\sqrt{var(\,^{3}\tilde{e}_1)}}$
$^{1}\delta_1 = 1$	0.71	0	0.5	0.26
$^{1}\delta_2 = 1$	0	0.71	0.5	0.34
$^{2}\delta_1 = 1$	0	0	0.5	0.53
$^{3}\delta_1 = 1$	0	0	0	0.53

Furthermore, we compute the Average Run Lengths of the proposed method, under different potential fault scenarios that may occur in the cotton spinning process. Note that because the cotton spinning process has four variables, there are 15 potential fault scenarios, including four single-fault scenarios (i.e., only one variable has a mean shift) and 11 multiple-fault scenarios (i.e., more than one variable has mean shifts). The results of the Average Run Lengths computation are summarized in Table 7-3. In the table, each row corresponds to a potential fault scenario; s is used to denote the value of the mean shift magnitude. Column "$ARL(\,^{r}\delta_i\,)$" records the Average Run Length for the mean shift $^{r}\delta_i$ to be detected by the corresponding control chart; $ARL(\,^{r}\delta_i\,)$ is indeed ARL_0 if $^{r}\delta_i = 0$, and is $ARL_1(\,^{r}\delta_i\,)$ otherwise. When compute the $ARL(\,^{r}\delta_i\,)$, the Bonferroni method is used to set the Type-I error of each individual control chart to be $0.05/4$ in order to control the overall system-level Type-I error to be no more than 0.05. Moreover, for each fault scenario, all the $ARL(\,^{r}\delta_i\,)$ corresponding to $^{r}\delta_i = 0$ are averaged and recorded in column "Ave_ARL_0"; all the $ARL(\,^{r}\delta_i\,)$ corresponding to $^{r}\delta_i = s$ are averaged and recorded in column "Ave_ARL_1" Columns Ave_ARL_0 and Ave_ARL_1 can be used to evaluate performance of the proposed method with respect to each fault scenario. Finally, the numbers in column Ave_ARL_0 are averaged, so are the numbers in column Ave_ARL_1; the results are recorded in the last row, which can be used

to evaluate the overall performance of the proposed method. Note that because this section focuses on applying the proposed method for large mean shifts detection and also due to space limit, only mean shifts with magnitude equal to 2 or 3 are shown in the table. Application of the proposed method for small mean shifts detection will be discussed in the next section.

Table 7-3 Fault scenarios and Average Run Lengths performance of the proposed method

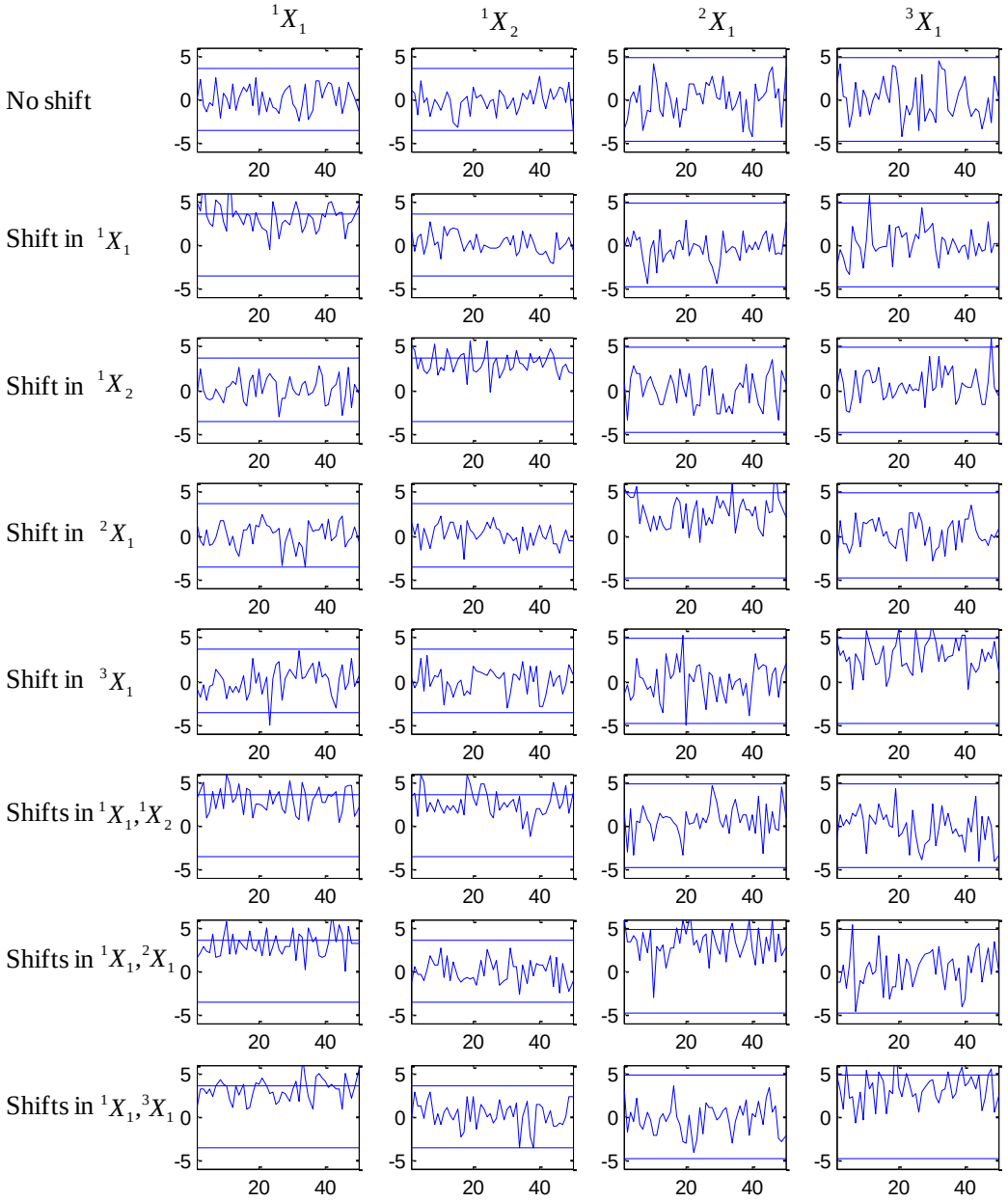
Fault Scenarios				Mean Shift Magnitude s=2						Mean Shift Magnitude s=3					
$^1\delta_1$	$^1\delta_2$	$^2\delta_1$	$^3\delta_1$	ARL($^1\delta_1$)	ARL($^1\delta_2$)	ARL($^2\delta_1$)	ARL($^3\delta_1$)	Ave_ARL$_0$	Ave_ARL$_1$	ARL($^1\delta_1$)	ARL($^1\delta_2$)	ARL($^2\delta_1$)	ARL($^3\delta_1$)	Ave_ARL$_0$	Ave_ARL$_1$
s	0	0	0	7.2	160.0	160.0	160.0	160.0	7.2	2.8	160.0	160.0	160.0	160.0	2.8
0	s	0	0	160.0	7.2	160.0	160.0	160.0	7.2	160.0	2.8	160.0	160.0	160.0	2.8
0	0	s	0	160.0	160.0	13.7	160.0	160.0	13.7	160.0	160.0	5.7	160.0	160.0	5.7
0	0	0	s	160.0	160.0	160.0	13.9	160.0	13.9	160.0	160.0	160.0	5.8	160.0	5.8
s	s	0	0	7.2	7.2	160.0	160.0	160.0	7.2	2.8	2.8	160.0	160.0	160.0	2.8
s	0	s	0	7.2	160.0	13.7	160.0	160.0	10.5	2.8	160.0	5.7	160.0	160.0	4.3
s	0	0	s	7.2	160.0	160.0	13.9	160.0	10.5	2.8	160.0	160.0	5.8	160.0	4.3
0	s	s	0	160.0	7.2	13.7	160.0	160.0	10.5	160.0	2.8	5.7	160.0	160.0	4.3
0	s	0	s	160.0	7.2	160.0	13.9	160.0	10.5	160.0	2.8	160.0	5.8	160.0	4.3
0	0	s	s	160.0	160.0	13.7	13.9	160.0	13.8	160.0	160.0	5.7	5.8	160.0	5.8
s	s	s	0	7.2	7.2	13.7	160.0	160.0	9.4	2.8	2.8	5.7	160.0	160.0	3.8
s	s	0	s	7.2	7.2	160.0	13.9	160.0	9.4	2.8	2.8	160.0	5.8	160.0	3.8
s	0	s	s	7.2	160.0	13.7	13.9	160.0	11.6	2.8	160.0	5.7	5.8	160.0	4.8
0	s	s	s	160.0	7.2	13.7	13.9	160.0	11.6	160.0	2.8	5.7	5.8	160.0	4.8
s	s	s	s	7.2	7.2	13.7	13.9	----	10.5	2.8	2.8	5.7	5.8	----	4.3
								160.0	**10.5**					**160.0**	**4.3**

For purpose of comparison, we develop a similar table to Table 7-3, i.e., Table 7-4, by applying the traditional regression-based method which ignores the measurement errors and uses the measured values of each variable as the true values. It can be seen from Tables 7-3 and 7-4 that the ARL_0 performance of the proposed method is better than the traditional methods for all the fault scenarios; the ARL_1 performance of the two methods are close.

Table 7-4 Fault scenarios and Average Run Lengths performance of the traditional method

Fault Scenarios				Mean Shift Magnitude s=2						Mean Shift Magnitude s=3					
$^1\delta_1$	$^1\delta_2$	$^2\delta_1$	$^3\delta_1$	ARL($^1\delta_1$)	ARL($^1\delta_2$)	ARL($^2\delta_1$)	ARL($^3\delta_1$)	Ave_ARL$_0$	Ave_ARL$_1$	ARL($^1\delta_1$)	ARL($^1\delta_2$)	ARL($^2\delta_1$)	ARL($^3\delta_1$)	Ave_ARL$_0$	Ave_ARL$_1$
s	0	0	0	7.2	160.0	14.9	42.2	72.4	7.2	2.8	160.0	6.3	23.7	63.3	2.8
0	s	0	0	160.0	7.2	14.9	28.7	67.9	7.2	160.0	2.8	6.3	14.2	60.2	2.8
0	0	s	0	160.0	160.0	14.9	13.4	111.1	14.9	160.0	160.0	6.3	5.6	108.5	6.3
0	0	0	s	160.0	160.0	160.0	13.4	160.0	13.4	160.0	160.0	160.0	5.6	160.0	5.6
s	s	0	0	7.2	7.2	3.2	10.3	6.8	7.2	2.8	2.8	1.4	4.1	2.8	2.8
s	0	s	0	7.2	160.0	3.2	5.7	82.8	5.2	2.8	160.0	1.4	2.3	81.1	2.1
s	0	0	s	7.2	160.0	14.9	5.7	87.5	6.4	2.8	160.0	6.3	2.3	83.1	2.5
0	s	s	0	160.0	7.2	3.2	4.5	82.2	5.2	160.0	2.8	1.4	1.8	80.9	2.1
0	s	0	s	160.0	7.2	14.9	4.5	87.5	5.8	160.0	2.8	6.3	1.8	83.1	2.3
0	0	s	s	160.0	160.0	14.9	2.9	160.0	8.9	160.0	160.0	6.3	1.3	160.0	3.8
s	s	s	0	7.2	7.2	1.4	2.5	2.5	5.3	2.8	2.8	1.0	1.2	1.2	2.2
s	s	0	s	7.2	7.2	3.2	2.5	3.2	5.6	2.8	2.8	1.4	1.2	1.4	2.3
s	0	s	s	7.2	160.0	3.2	1.8	160.0	4.1	2.8	160.0	1.4	1.1	160.0	1.8
0	s	s	s	160.0	7.2	3.2	1.6	160.0	4.0	160.0	2.8	1.4	1.0	160.0	1.8
s	s	s	s	7.2	7.2	1.4	1.3	----	4.3	2.8	2.8	1.0	1.0	----	1.9
								88.8	**7.0**					**86.1**	**2.9**

Finally, to illustrate the practical charting of the proposed method, data for the cotton spinning process are simulated. The simulation consists of 15 fault scenarios and one no-fault (i.e., normal) scenario. Under each scenario, the mean shift magnitude is set to be three and data are simulated for 50 time periods after the shift(s). Then, a Shewhart control chart is built for each variable based on the proposed method. The control charts are shown in Fig. 7-2. It can be clearly seen that control charts for variables having mean shifts general out-of-control signals, while control charts for variables not having mean shifts stay within control limits.


$^1X_1$
$^1X_2$
$^2X_1$
$^3X_1$
No shift
Shift in $^1X_1$
Shift in $^1X_2$
Shift in $^2X_1$
Shift in $^3X_1$
Shifts in $^1X_1$,$^1X_2$
Shifts in $^1X_1$,$^2X_1$
Shifts in $^1X_1$,$^3X_1$
5
0
-5
20
40

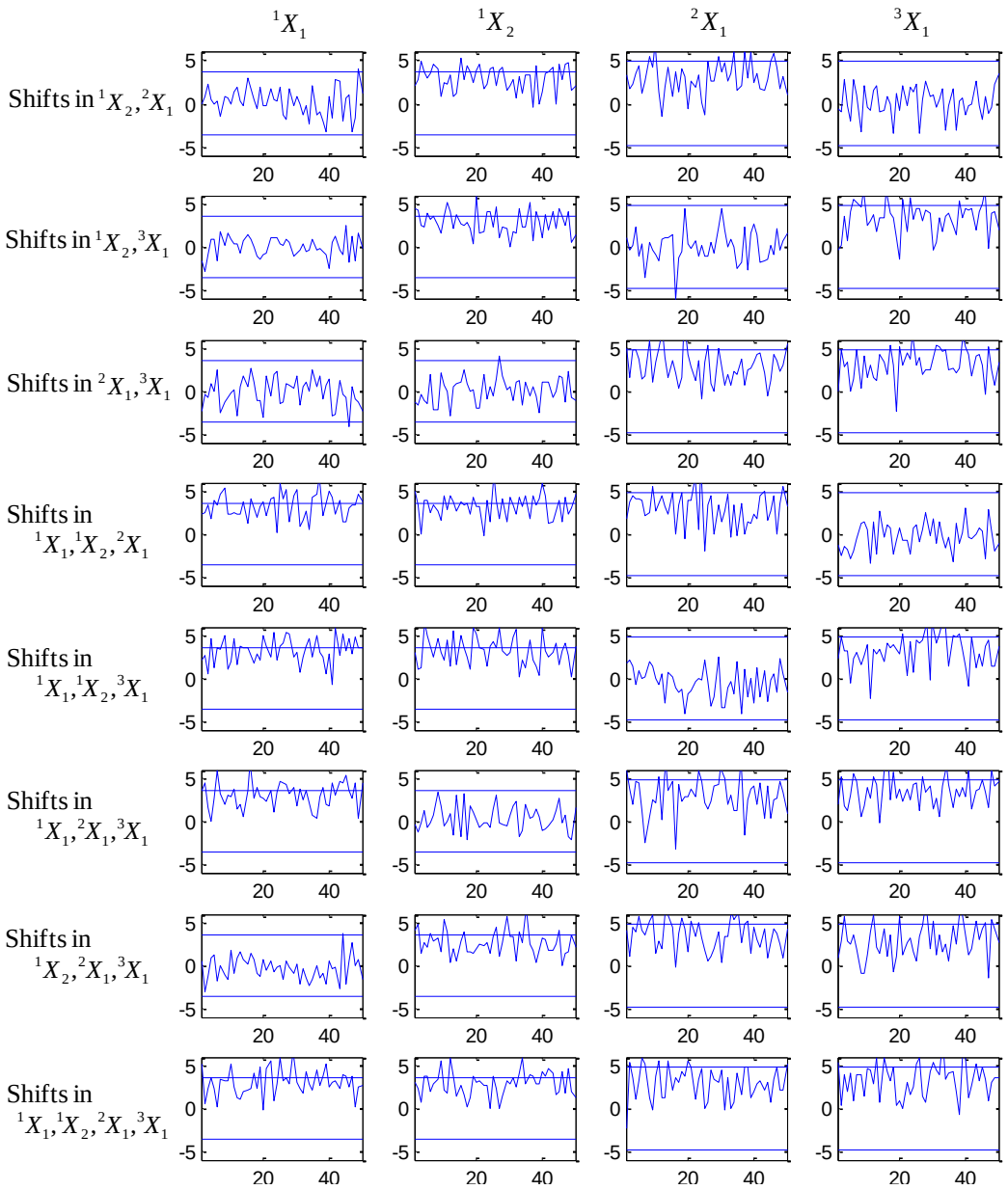
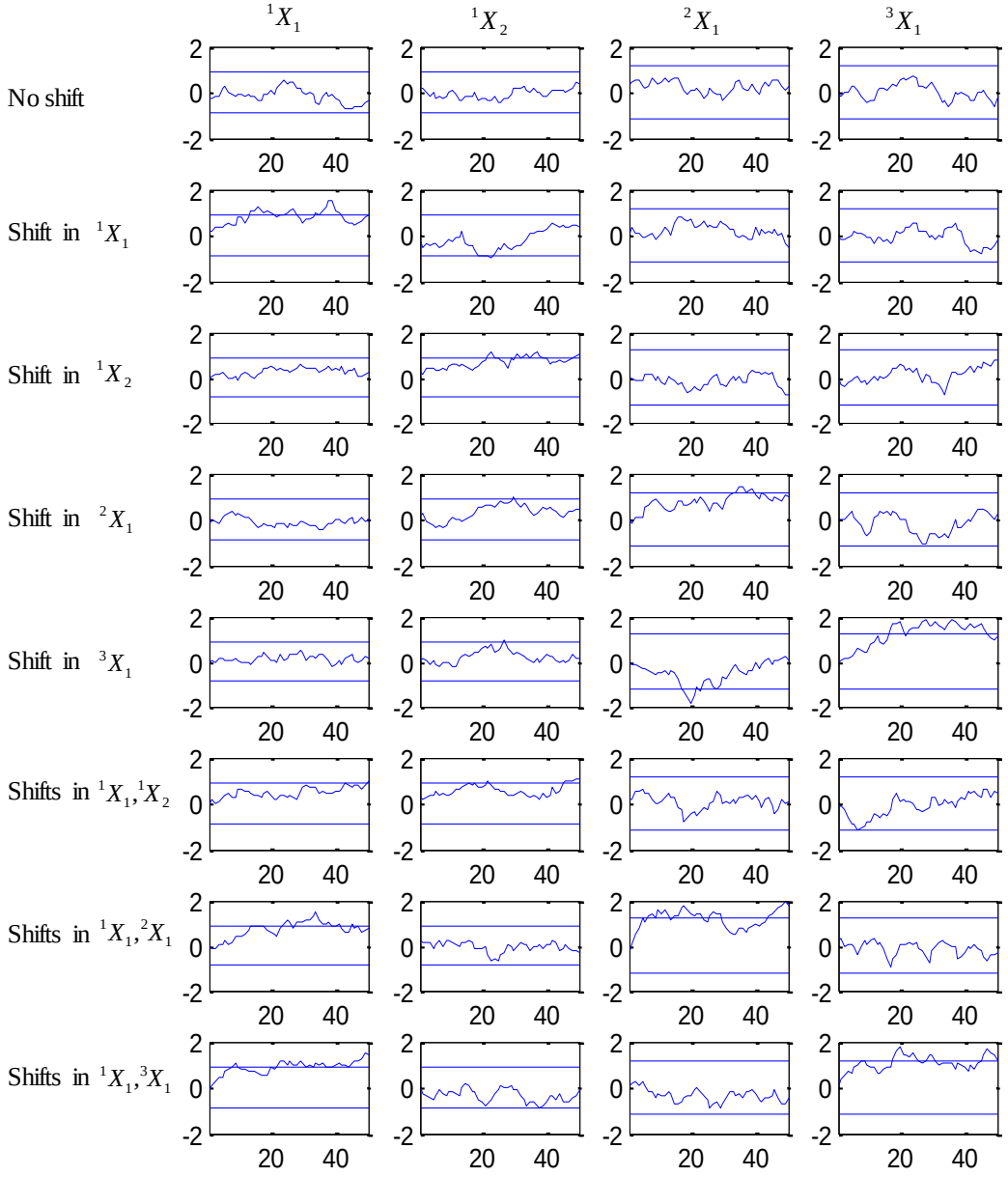


Fig. 7-2 Shewhart Control charts for 15 fault scenarios and one no-fault scenario of the cotton spinning process based on the proposed method


$^{1}X_{1}$
$^{1}X_{2}$
$^{2}X_{1}$
$^{3}X_{1}$
No shift
Shift in $^{1}X_{1}$
Shift in $^{1}X_{2}$
Shift in $^{2}X_{1}$
Shift in $^{3}X_{1}$
Shifts in $^{1}X_{1}, ^{1}X_{2}$
Shifts in $^{1}X_{1}, ^{2}X_{1}$
Shifts in $^{1}X_{1}, ^{3}X_{1}$
2
0
-2
20
40

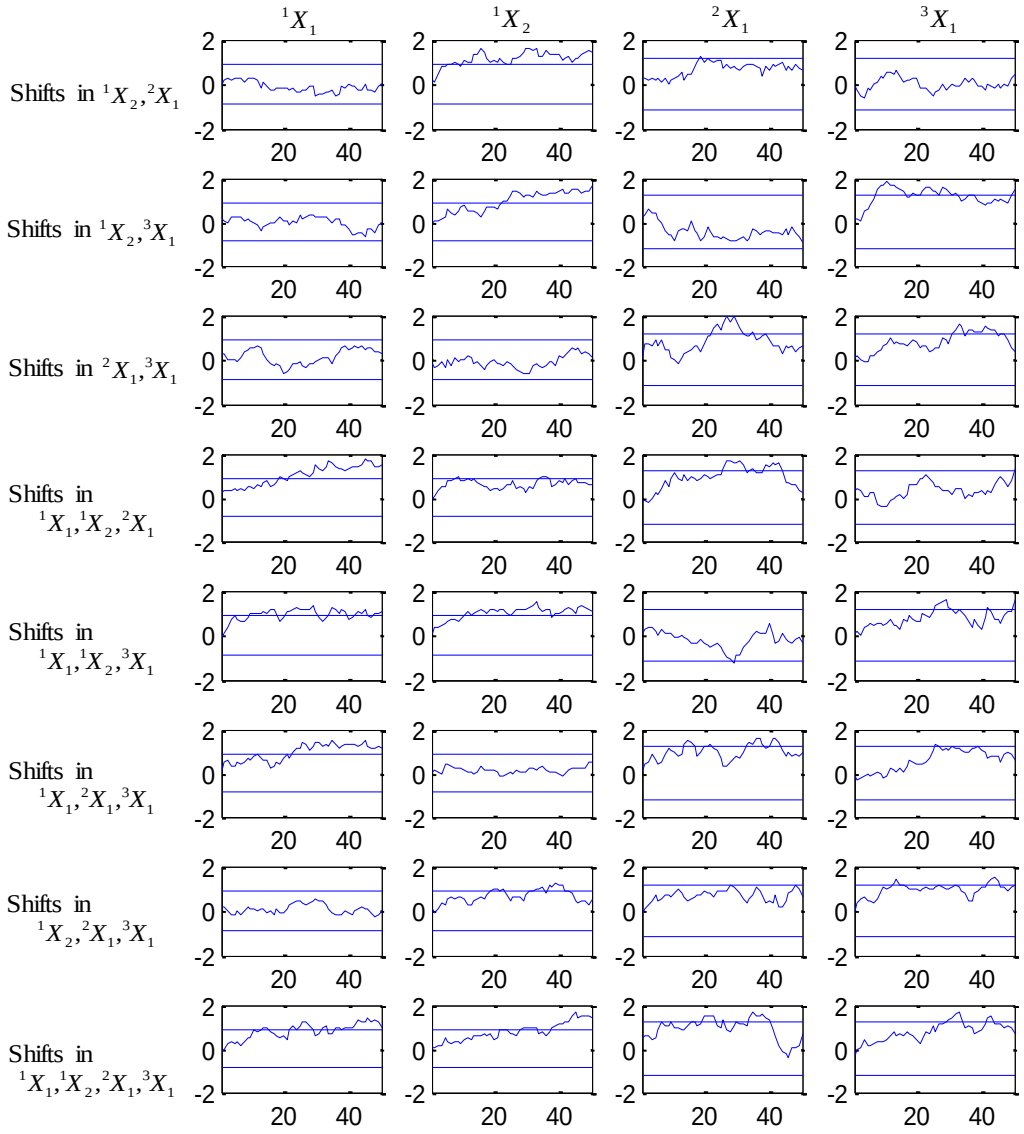


Fig. 7-3 EWMA Control charts for 15 fault scenarios and one no-fault scenario of the cotton spinning process based on the proposed method

To detect small mean shifts, the proposed method can be used in conjunction with the EWMA control charts, which has been discussed at the end of Section 7-3. To demonstrate the procedure, data for the cotton spinning process are simulated. The simulation consists of 15 faulty scenarios and one no-fault scenario. Under each scenario, the mean shift magnitude is set to be 1 and the data are simulated for 50 time periods after the shift(s). Then, for each time period $t \in \{1, \cdots, 50\}$, (14) is used to compute the

points on the EWMA control charts, with $\lambda = 0.1$. The control limits for the EWMA charts are computed using (15), with $L = 2.7$. The values for λ and L are selected to satisfied certain requirements on the Average Run Lengths [22]. The control charts are shown in Fig. 7-3. It can be seen that control charts for variables having mean shifts generate out-of-control signals, while control charts for variables not having mean shifts stay within control limits.

7.5.2 Example II: a hot forming process

This example targets a hot forming process consisting of five variables: X_1 (blank holding force, or BHF), X_2 (temperature), X_3 (tension in workpiece), X_4 (material flow stress), and X_5 (final dimension of workpiece). A 2-D physical illustration of this process is given in Fig. 7-7-4. A Bayesian network of this process has been identified by Li, et al. [20], as shown in Fig. 7-7-5 in which the variables X_1, X_2, X_3, X_4, and X_5 are denoted by ${}^1X_1$, ${}^1X_2$, ${}^2X_1$, ${}^2X_2$, and ${}^3X_1$, respectively, according to the layers they belong to. The regression-based process models (i.e., regressing each variable on its parents) are:

$$
\begin{aligned}
{}^1X_1 &= {}^1e_1, \\
{}^1X_2 &= {}^1e_2, \\
{}^2X_1 &= 0.325\ {}^1X_1 - 0.493\ {}^1X_2 + {}^2e_1, \\
{}^2X_2 &= -0.688\ {}^1X_2 + {}^2e_2 \\
{}^3X_1 &= 0.574\ {}^2X_1 + 0.335\ {}^2X_2 + {}^3e_1,
\end{aligned}
$$

and the variables follow the standard normal distribution.

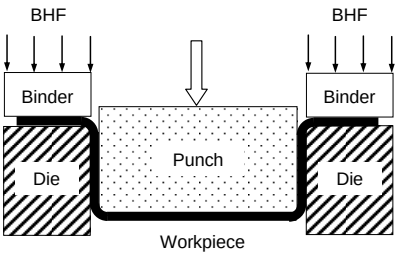


Fig. 7-4 2-D illustration of the hot forming process

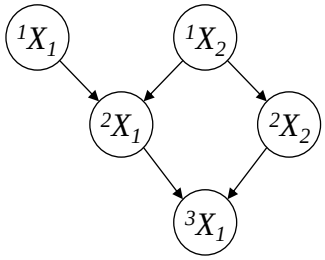


Fig. 7-5 Bayesian network of the hot forming process

Sensor noise is a major type of measurement errors in this process. Based on Section 7-4, the maximum level of sensor noise allowed, under a certain fault detectability requirement, can be identified. Specifically, under the requirement that mean shifts of three standard deviations in each variable must be detected within five samples, i.e., $\widetilde{ARL}_{1U}(3) = 5$, (19) can be used to obtain constraints on the variances of the sensor noise, i.e.,

$$\begin{cases} var(\,^{1}\varepsilon_1) \le 0.73 \\ var(\,^{1}\varepsilon_2) \le 0.73 \\ var(\,^{2}\varepsilon_1) + 0.325^2 var(\,^{1}\varepsilon_1) + 0.493^2 var(\,^{1}\varepsilon_2) \le 1.078 \\ var(\,^{2}\varepsilon_2) + 0.688^2 var(\,^{1}\varepsilon_2) \le 1.203 \\ var(\,^{3}\varepsilon_1) + 0.574^2 var(\,^{2}\varepsilon_1) + 0.335^2 var(\,^{2}\varepsilon_2) \le 1.302 \end{cases}, \qquad (22)$$

where the sample size n is set to be 1, and Bonferroni method is used to ensure the system-level Type-I error to be no more than 0.05. Furthermore, considering a special case of homogeneous sensor noise, i.e., $var(\,^{r}\varepsilon_i) = \sigma_\varepsilon^2$, (22) reduces to $\sigma_\varepsilon^2 \le 0.73$.

To verify if $\sigma_\varepsilon^2 \le 0.73$ can satisfy the detectability requirement $\widetilde{ARL}_{1U}(3) = 5$, we make σ_ε^2 to take values incrementally between 0 and 0.73. Then, for each value of σ_ε^2, we apply (13) to compute the average run lengths in detecting mean shifts of three standard deviations, i.e., $\widetilde{ARL}_1(\,^{1}\delta_1 = 3)$, $\widetilde{ARL}_1(\,^{1}\delta_2 = 3)$, $\widetilde{ARL}_1(\,^{2}\delta_1 = 3)$, $\widetilde{ARL}_1(\,^{2}\delta_2 = 3)$, and $\widetilde{ARL}_1(\,^{3}\delta_1 = 3)$ for control charts on $\hat{\tilde{E}}(\,^{1}e_1)$, $\hat{\tilde{E}}(\,^{1}e_2)$,

$\hat{\tilde{E}}({}^{2}e_1)$, $\hat{\tilde{E}}({}^{2}e_2)$, and $\hat{\tilde{E}}({}^{3}e_1)$, respectively. The results are shown in Fig. 7-6, which clearly indicate that $\widetilde{ARL}_1$ (${}^{r}\delta_i = 3$) < 5, i.e., the detectability requirement is satisfied.

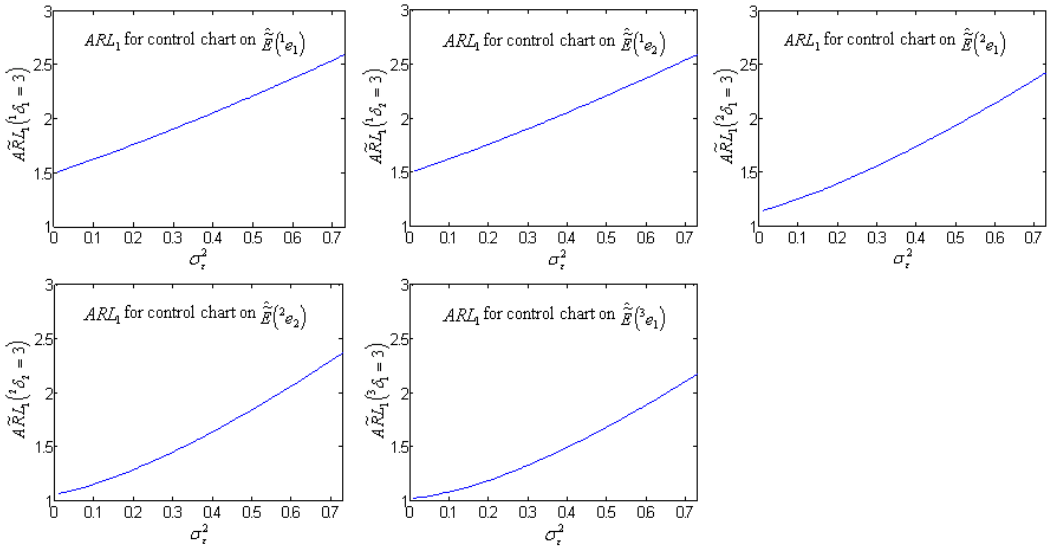


Fig. 7-6 Average run length in detecting mean shifts of three standard deviations with respect to different levels of sensor noise

In addition to sensor noise, another type of measurement errors commonly existing in this hot forming process is that the measurement errors of the sensors for ${}^{2}X_1$, ${}^{2}X_2$, and ${}^{3}X_1$ are all affected by temperature, i.e., ${}^{1}X_2$. Therefore, the relationship between each variable and its measured value is:

$$
\begin{aligned}
{}^{1}\tilde{X}_1 &= {}^{1}X_1 + {}^{1}\varepsilon_1, \\
{}^{1}\tilde{X}_2 &= {}^{1}X_2 + {}^{1}\varepsilon_2, \\
{}^{2}\tilde{X}_1 &= {}^{2}X_1 + {}^{2,1}c_{12}\,{}^{1}X_2 + {}^{2}\varepsilon_1, \\
{}^{2}\tilde{X}_2 &= {}^{2}X_2 + {}^{2,1}c_{22}\,{}^{1}X_2 + {}^{2}\varepsilon_2, \\
{}^{3}\tilde{X}_1 &= {}^{3}X_1 + {}^{3,1}c_{12}\,{}^{1}X_2 + {}^{3}\varepsilon_1.
\end{aligned}
$$

Furthermore, (18) can be applied to obtain the constrains on the measurement errors σ_ε^2, ${}^{2,1}c_{12}$, ${}^{2,1}c_{22}$, ${}^{3,1}c_{12}$, in order to achieve the detectability requirement $\widetilde{ARL}_{1U}(3) = 5$. The constraints are:

$$\begin{cases} \sigma_\varepsilon^2 \le 0.73 \\ \sigma_\varepsilon^2 \left(1.106 + \left({}^{2,1}c_{12} - 0.493\right)^2\right) \le 1.078 \\ \sigma_\varepsilon^2 \left(1 + \left({}^{2,1}c_{22} - 0.688\right)^2\right) \le 1.203 \\ \sigma_\varepsilon^2 \left(1.442 + \left({}^{3,1}c_{12} - 0.574\,{}^{2,1}c_{12} - 0.335\,{}^{2,1}c_{22}\right)^2\right) \le 1.302 \end{cases} . \quad (23)$$

To verify if the constraints in (23) can fulfill the detectability requirement, a similar graph to Fig. 7-6 can be constructed, with the x-axis being different combinations of values for σ_ε^2, ${}^{2,1}c_{12}$, ${}^{2,1}c_{22}$, ${}^{3,1}c_{12}$ that satisfy the constraints. Due to page limit, the graph is not shown here, but it confirms that the detectability requirement is fulfilled.

7.6 Conclusion

This paper proposed a regression-based method for multivariate process monitoring and fault detection considering four types of major measurement errors in the data. On the one hand, given that values of measurement errors may be known *a priori*, we developed control charts for mean shift detection based on the measurement data. On the other hand, if measurement errors are not known, we developed procedures to identify the maximum allowable measurement errors in order to satisfy certain detectability requirements. The proposed method is applicable to cascade or multistage processes in which variables have a natural ordering, and processes in which the causal relationships among variables can be known and represented by a Bayesian network.

The proposed method was applied to a cotton spinning cascade process, in which the method were compared with the traditional method that ignores the measurement errors and uses the measured values of each variable as the true values. The comparison showed that the proposed method significantly reduced false alarm rates in mean shift

detection. Also, the proposed method was applied to a hot forming process in which a Bayesian network is available. It successfully identified the maximum levels of measurement errors which allow the mean shifts to be detected within a given required time period. Future research may include developing methods for monitoring process variability considering measurement errors.

Appendix I Proof of the variance for $\hat{\bar{E}}({}^{r}e_i)$ in (9)

To prove (9), we first prove (A-1):

$$\hat{\bar{E}}({}^{r}X_i) = {}^{r}\bar{x}_i + {}^{r}\bar{\xi}_i - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{w}_i^T \, {}^{k}\bar{\boldsymbol{\xi}} \qquad \text{(A-1)}$$

where ${}^{r}\bar{\xi}_i = {}^{r}\bar{\varepsilon}_i / {}^{r}s_i$, ${}^{r}\bar{\varepsilon}_i$ is the sample average for ${}^{r}\varepsilon_i$, ${}^{k}\bar{\boldsymbol{\xi}} = \left\{ {}^{k}\bar{\xi}_1, \cdots, {}^{k}\bar{\xi}_{m_k} \right\}^T$; ${}^{r,k}\mathbf{w}_i^T$ is a recursive function defined in (10), i.e.,

$${}^{r,k}\mathbf{w}_i^T = \begin{cases} {}^{r,k}\mathbf{d}_i^T & k = r-1 \\ {}^{r,k}\mathbf{d}_i^T - \sum_{h=k+1}^{r-1} {}^{r,h}\mathbf{d}_i^T \, {}^{h,k}\mathbf{W}^T & k = r-2, \cdots, 1 \end{cases},$$

and ${}^{h,k}\mathbf{W} = \left\{ {}^{h,k}\mathbf{w}_1, \cdots, {}^{h,k}\mathbf{w}_{m_h} \right\}$.

(A-1) can be proved by mathematical induction, as follows:

Step 1(check if (A-1) holds for $r = 1$):

According to (7),

$$\hat{\bar{E}}\left({}^{1}X_i\right) = {}^{1}\bar{\tilde{y}}_i = \left({}^{1}\bar{\tilde{x}}_i - {}^{1}b_i \right) / {}^{1}s_i = \left({}^{1}b_i + {}^{1}s_i \, {}^{1}\bar{x}_i + {}^{1}\bar{\varepsilon}_i - {}^{1}b_i \right) / {}^{1}s_i = {}^{1}\bar{x}_i + {}^{1}\bar{\xi}_i.$$

Therefore, (A-1) holds for $r = 1$.

Step 2(Assuming that (A-1) holds for $r = 1, \cdots, l$, check if (A-1) holds for $r = l + 1$):

According to (7),

$$\hat{\bar{E}}\left({}^{l+1}X_i\right) = {}^{l+1}\bar{\tilde{y}}_i - \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \, \hat{\bar{E}}({}^{r}\mathbf{X}). \qquad \text{(A-2)}$$

Since we have assumed that (A-1) holds for $r = 1, \cdots, l$, (A-1) can be inserted into (A-2), i.e.,

$$\hat{\bar{E}}\left({}^{l+1}X_i\right) = {}^{l+1}\bar{\tilde{y}}_i - \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \left({}^{r}\bar{\mathbf{x}} + {}^{r}\bar{\boldsymbol{\xi}} - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right),$$

where ${}^{r}\bar{\mathbf{x}} = \left\{ {}^{r}\bar{x}_1, \cdots, {}^{r}\bar{x}_{m_r} \right\}^T$. Furthermore,

$$\hat{\bar{E}}\left({}^{l+1}X_i\right) = \left({}^{l+1}\bar{\tilde{x}}_i - {}^{l+1}b_i\right) / {}^{l+1}s_i - \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \left({}^{r}\bar{\mathbf{x}} + {}^{r}\bar{\boldsymbol{\xi}} - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i + \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \; {}^{r}\bar{\mathbf{x}} - \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \left({}^{r}\bar{\mathbf{x}} + {}^{r}\bar{\boldsymbol{\xi}} - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \left({}^{r}\bar{\boldsymbol{\xi}} - \sum_{k=1}^{r-1} {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \left(\sum_{r=1}^{l} {}^{l+1,r}\mathbf{d}_i^T \; {}^{r}\bar{\boldsymbol{\xi}} - \sum_{r=1}^{l} \sum_{k=1}^{r-1} {}^{l+1,r}\mathbf{d}_i^T \; {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \left(\sum_{k=1}^{l} {}^{l+1,k}\mathbf{d}_i^T \; {}^{k}\bar{\boldsymbol{\xi}} - \sum_{k=1}^{l-1} \sum_{r=k+1}^{l} {}^{l+1,r}\mathbf{d}_i^T \; {}^{r,k}\mathbf{W}^T \; {}^{k}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \left(\sum_{k=1}^{l-1} \left({}^{l+1,k}\mathbf{d}_i^T - \sum_{r=k+1}^{l} {}^{l+1,r}\mathbf{d}_i^T \; {}^{r,k}\mathbf{W}^T \right) {}^{k}\bar{\boldsymbol{\xi}} + {}^{l+1,l}\mathbf{d}_i^T \; {}^{l}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \left(\sum_{k=1}^{l-1} {}^{l+1,k}\mathbf{w}_i^T \; {}^{k}\bar{\boldsymbol{\xi}} + {}^{l+1,l}\mathbf{w}_i^T \; {}^{l}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{l+1}\bar{x}_i + {}^{l+1}\bar{\xi}_i - \sum_{k=1}^{l} {}^{l+1,k}\mathbf{w}_i^T \; {}^{k}\bar{\boldsymbol{\xi}}$$

Therefore, given that (A-1) holds for $r = 1, \cdots, l$, (A-1) holds for $r = l + 1$. This completes the proof of (A-1).

Next, insert (A-1) into (8),

$$\hat{\bar{E}}\left({}^{r}e_i \right) = {}^{r}\bar{\tilde{y}}_i - \sum_{k=1}^{r-1} \left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T \right) \left({}^{k}\bar{\mathbf{x}} + {}^{k}\bar{\boldsymbol{\xi}} - \sum_{h=1}^{k-1} {}^{k,h}\mathbf{W}^T \; {}^{h}\bar{\boldsymbol{\xi}} \right)$$

$$= {}^{r}\overline{x}_i + {}^{r}\bar{\xi}_i + \sum_{k=1}^{r-1} {}^{r,k}\mathbf{d}_i^T \, {}^{k}\overline{\mathbf{x}} - \sum_{k=1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right)\left({}^{k}\overline{\boldsymbol{\xi}} + {}^{k}\bar{\boldsymbol{\xi}} - \sum_{h=1}^{k-1} {}^{k,h}\mathbf{W}^T \, {}^{h}\bar{\boldsymbol{\xi}}\right)$$

$$= {}^{r}\overline{x}_i + {}^{r}\bar{\xi}_i - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T \, {}^{k}\overline{\mathbf{x}} - \sum_{k=1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right)\left({}^{k}\overline{\boldsymbol{\xi}} - \sum_{h=1}^{k-1} {}^{k,h}\mathbf{W}^T \, {}^{h}\bar{\boldsymbol{\xi}}\right)$$

$$= {}^{r}\overline{e}_i + {}^{r}\bar{\xi}_i - \sum_{k=1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right)\left({}^{k}\overline{\boldsymbol{\xi}} - \sum_{h=1}^{k-1} {}^{k,h}\mathbf{W}^T \, {}^{h}\bar{\boldsymbol{\xi}}\right)$$

$$= {}^{r}\overline{e}_i + {}^{r}\bar{\xi}_i - \left(\sum_{k=1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right) {}^{k}\overline{\boldsymbol{\xi}} - \sum_{k=1}^{r-1}\sum_{h=1}^{k-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right) {}^{k,h}\mathbf{W}^T \, {}^{h}\bar{\boldsymbol{\xi}}\right)$$

$$= {}^{r}\overline{e}_i + {}^{r}\bar{\xi}_i - \left(\sum_{h=1}^{r-1}\left({}^{r,h}\mathbf{d}_i^T + {}^{r,h}\boldsymbol{\beta}_i^T\right) {}^{h}\overline{\boldsymbol{\xi}} - \sum_{h=1}^{r-2}\sum_{k=h+1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right) {}^{k,h}\mathbf{W}^T \, {}^{h}\bar{\boldsymbol{\xi}}\right)$$

$$= {}^{r}\overline{e}_i + {}^{r}\bar{\xi}_i - \sum_{h=1}^{r-1}\left({}^{r,h}\mathbf{d}_i^T + {}^{r,h}\boldsymbol{\beta}_i^T - \sum_{k=h+1}^{r-1}\left({}^{r,k}\mathbf{d}_i^T + {}^{r,k}\boldsymbol{\beta}_i^T\right) {}^{k,h}\mathbf{W}^T\right) {}^{h}\bar{\boldsymbol{\xi}}$$

$$= {}^{r}\overline{e}_i + {}^{r}\bar{\xi}_i - \sum_{h=1}^{r-1} {}^{r,h}\mathbf{a}_i^T \, {}^{h}\bar{\boldsymbol{\xi}}$$

where ${}^{r}\overline{e}_i$ is the sample average for ${}^{r}e_i$. Therefore, the variance of $\hat{\bar{E}}({}^{r}e_i)$ is

$$var\left(\hat{\bar{E}}({}^{r}e_i)\right) = \frac{1}{n}\left(var({}^{r}e_i) + var({}^{r}\varepsilon_i)/\,{}^{r}s_i^2 + \sum_{h=1}^{r-1}\sum_{j=1}^{m_h} {}^{r,h}a_{ij}^2 \, var\left({}^{h}\varepsilon_j\right)/\,{}^{h}s_j^2\right)$$

$$=$$

$$\frac{1}{n}\left(var({}^{r}X_i) - \sum_{k=1}^{r-1} {}^{r,k}\boldsymbol{\beta}_i^T \, \mathbf{cov}\left({}^{k}\mathbf{X},\ {}^{r}X_i\right) + var({}^{r}\varepsilon_i)/\,{}^{r}s_i^2 + \right.$$

$$\left.\sum_{h=1}^{r-1}\sum_{j=1}^{m_h} {}^{r,h}a_{ij}^2 \, var\left({}^{h}\varepsilon_j\right)/\,{}^{h}s_j^2\right). \quad \Delta$$

Reference

[1] Barton, R. R. and Gonzalez-Barreto, D. R. (1996) "Process-oriented Basis Representations for Multivariate Process Diagnostics." Quality Engineering 9, pp. 107-118.

[2] Carroll, R. J. and Stefanski, L. A. (1990) "Approximate quasilikelihood estimation in models with surrogate predictors." J. Am. Statist. Ass., 85, pp. 652–663.

[3] Carroll, R.J., and Stefanski, L.A., (1997) "Asymptotic theory for the Simex. estimator in measurement error models", Advances in. Statistical Decision Theory and Applications, pp. 151-164.

[4] Carroll, R. J., Stefanski, L. A., and Ruppert, D. (2006). Measurement Error in Nonlinear Models, Second Edition. London: Chapman and Hall.

[5] Ceglarek, D. and Shi, J., (1999) "Fixture Failure Diagnosis for Sheet Metal Assembly with Consideration of Measurement Noise." ASME Transactions, Journal of Manufacturing Science and Engineering. 121, pp771-777.

[6] Cook, J.R. and Stefanski, L.A. (1994) "Simulation-extrapolation estimation in parametric measurement error models." J. Am. Stat. Assoc. 89, pp. 1314–1328.

[7] Devanarayan, V., and Stefanski, L.A. (2002) "Empirical simulation extrapolation for measurement error models with replicate measurements." Stat. Probab. Lett. 59, pp. 219-225.

[8] Ding, Y., E. A. Elsayed, et al. (2006). "Distributed sensing for quality and productivity improvements." IEEE Transactions on Automation Science and Engineering 3(4), pp. 344-358.

[9] Gleser, L.J. (1990) "Improvement of the naive approach to estimation in nonlinear error- in-variables regression models." In Statistical Analysis of Measurement Error Models and applications. Ameri math society, providence

[10] Hawkins, D.M. (1991). "Multivariate Quality Control Based on Regression-Adjusted Variables," Technometrics 33, pp. 61-75.

[11] Hawkins, D.M. (1993). "Regression Adjustment for Variables in Multivariate Quality Control," Journal of Quality Technology 25, pp. 170-182.

[12] Heckerman, D., (1999), "A Tutorial on Learning with Bayesian Networks," Learning in Graphical Models, pp. 301–354.

[13] Heckerman, D., Mamdani, A., and Wellman, M.P. (1995), "Real-World Applications of Bayesian Networks," Communications of ACM, 38(3), pp. 24-26.

[14] Jin, R., Li, J., and Shi, J., (2007), "Quality Prediction and Control in Rolling Processes using Logistic Regression," Transactions of NAMRI/SME, 35, pp. 113-120.

[15] Korb, K.B. and Nicholson, A.E. (2003). Bayesian Artificial Intelligence. London, UK: Chapman and Hall.

[16] Lerner, U. (2002). Hybrid Bayesian Networks for Reasoning about Complex Systems, Ph.D. Thesis, Stanford University.

[17] Li, J., Gutchess, D., Shi, J., and Chang, S. (2003) "Real-Time Surface Defect Detection in Hot Rolling Process," Proceedings of the Iron and Steel Exposition and 2003 AISE Annual Convention, Pittsburgh, PA.

[18] Li, J., and Shi, J., (2007), "Knowledge Discovery from Observational Data for Process Control using Causal Bayesian Networks," IIE Transactions, 39 (6), pp. 681 – 690.

[19] Li, J., Shi, J., and Chang, T.S., (2007) "On-line Seam Detection in Rolling Processes using Snake Projection and Discrete Wavelet Transform," ASME (American Society of

Mechanical Engineers) Transactions, Journal of Manufacturing Science and Engineering, 129(5), pp. 926-933.

[20] Li, J., Jin, J., and Shi, J., (2008) "Causation-based T^2 Decomposition for Multivariate Process Monitoring and Diagnosis," Journal of Quality Technology, 40(1), pp. 46-58.

[21] Li, J., Shi, J., and Satz, D., (2008) "Modeling and Analysis of Disease and Risk Factors through Learning Bayesian Network from Observational Data," Quality and Reliability Engineering International, 24(3), 291-302.

[22] Lucas, J.M., and Saccucci, M.S., (1990) "Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements," Technometrics, 32(1), 1-29.

[23] Mason, R. L., Tracy, N. D., and Young, J. C. (1995). "Decomposition of T^2 for Multivariate Control Chart Interpretation." Journal of Quality Technology 27, pp. 99-108.

[24] Mason, R. L., Tracy, N. D., and Young, J. C. (1997). "A Practical Approach for Interpreting Multivariate T^2 Control Chart Signals." Journal of Quality Technology 29, pp. 396-406.

[25] Montgomery, D.C. (2001). Introduction to Statistical Quality Control. New York: John Wiley.

[26] Pierce, D.A. and Kellerer, A.M. (2005) "Adjusting for covariate errors with nonparametric assessment of the true covariate distribution." Biometrika 91, pp. 863-876.

[27] Runger G. C., Barton, R. R., Castillo, E. D., and Woodall, W. H. (2007). "Optimal Monitoring of Multivariate Data for Fault Patterns." Journal of Quality Technology 39 (2), pp. 159-172.

[28] Shu, L., Tsung, F. and Tsui, K.-L., (2005), "Effects of Estimation Errors on the Performance of Cause-Selecting Charts", IIE Transactions, 37, pp. 559-567.

[29] Shu, L, Tsung, F, Tsui, K.L. (2004) "Run-length performance of regression control charts with estimated parameters." Journal of Quality Technology 36, pp. 280–292.

[30] Spirtes, P., Glymour, C., and Scheines, R., (2000). Causation, Prediction, and Search. The MIT Press: Cambridge, MA.

[31] Wang, N., Carroll, R. J.&Liang, K. Y. (1996). "Quasilikelihood estimation in measurement error models with correlated replicates". Biometrics 52, pp. 401–411.

[32] Wade, M.R., Woodall, W.H. (1993), "A review and analysis of cause-selecting control charts", Journal of Quality Technology, 25 pp.161-169.

[33] Zhang, G. (1984), "A new type of control charts - cause-selecting control charts and a diagnosis theory with control charts", Proceedings of World Quality Congress '84, pp.175-85.

Chapter 8

CONCLUSIONS AND FUTURE RESEARCH

8.1 Summary and original contributions

This dissertation contributes to generic methodology development for analyzing some general complex datasets that are ubiquitous in biomedical research, healthcare and manufacturing. It also contributes to domain knowledge discovery for Alzheimer's disease research, nursing care process modeling and quality control in manufacturing.

1) *For high-dimensional datasets*, new methods for high dimensional Gaussian graphical models (SICE, chapter 2) and Bayesian networks structure learning (SBN, chapter 3) are developed. These methods are built upon the sparse learning methodology, through development of novel penalty formulations, e.g., the one used in SBN to penalize the violation of the DAG assumption of a BN, and development of efficient computational algorithms. The SICE and SBN can be widely used in many applications, such as biology, medicine, finance, health care and manufacturing. By applying SICE and SBN on Neuroimaging datasets collected for Alzheimer's disease research, novel knowledge of AD is revealed that may potentially lead to better early diagnosis and treatment effect evaluation.
2) *For hierarchically-structured datasets*, two novel models are developed. One is the transfer learning approach for network modeling of multiple related datasets where there is a common group structure shared by these datasets. By drawing on Bayesian hierarchical methodology, this transfer learning approach can transfer the knowledge learned from one dataset to the network modeling for another dataset, by exploiting the shared group structure. Although this transfer learning approach is developed in a network modeling context, its principal formulation can be readily extended to some other kinds of statistical models, which can be considered as a generic contribution to

the methodology development for analyzing this kind of hierarchical-structured dataset. Another one is a multi-level latent response linear regression model, which is capable to analyze the datasets that has a multi-level data structure. For example, in a nursing care process research, data is collected on multiple levels, including the individual nurse level, unit level and hospital level. Not like the multi-level data structure commonly assumed in traditional multi-level models, where the response variables are measured on individual level, in many applications, the response variables can only be measured on the organization level. The developed model contributes to mitigate the challenge. In addition to the methodology development, this model is used to analyze a real-world dataset collected for a nursing care process, which sheds light on understanding the nursing care process, identifying the major determinants of the nursing care quality, and potentially leading to nursing care process improvement.

3) *For multi-modality data fusion*: a novel model, called sparse composite linear discrimination analysis (SCLDA), is developed to identify those variables which are predictive to the outcome of interest, e.g., the disease onset of a person, from multi-modality data sources. SCLDA is particularly useful for identifying those variables with weak-effect if considered in isolation, i.e., weak predictive power, but strong-effect if considered jointly. This advantage is achieved by employing a composite parameterization, which decomposes any parameter of a LDA into a product of a shared parameter across all the modalities, and a private parameter for each modality. By this composite parameterization, the fragmented weak effects across different modalities are effectively unified, therefore the statistical power of SCLDA on identifying those weak-effect variables is increased.

4) *For datasets corrupted with measurement errors*, a regression-based process monitoring method with consideration of measurement errors is developed. As most existing process monitoring methods are based on a common assumption that the measured values of variables are the true values, with limited consideration of the various types of measurement errors embedded in the data, those methods are less effective when applied in real-world applications, where measurement errors are inevitable. On the other hand, research on measurement errors has been conducted from a pure theoretical statistics point of view, without any linking of the modeling and analysis of measurement errors with monitoring and fault detection. Thus, this novel method fills in this gap, which is capable to perform multivariate process monitoring and fault detection considering four types of major measurement errors, including sensor bias, sensitivity, noise and dependency of the relationship between a variable and its measured value on some other variables.

8.2 Future research

As mentioned in chapter 1, complex datasets are often context-dependent. To make full use of these datasets, the modeling and analysis of these complex datasets needs special statistical consideration that can effectively exploit the special structures of these datasets. In what follows I would like to present several challenging topics:

1) *Electronic medical records (EMRs):* many healthcare organizations around the world are deploying EMRs. At the population level, EMRs can be used for public health surveillances, management of healthcare services. At the individual level, ERMs can be used for personalized medicine, early detection of disease onset, and prevention. EMRs can also be used for research, for instance, be used for identifying the linkage between genotypes with phenotypes. However, the current statistical methodology for effectively exploring EMRs is very limited and more advanced statistical models

are needed to make full use of EMRs, and link EMRs with other data modalities, such as public surveillance data, healthcare service provider's records, etc.

2) *Biotechnologies:* Biotechnologies have fueled a boom in molecular biology and medicine research. Nowadays it is not a difficulty to collect data from multiple biological entities on a genomic scale, e.g., genes, proteins, lipids and sugars, and on multiple levels, e.g., from molecular level to population level variables, yielding massive high-throughput "omic" data, such as genomic, epigenomic, proteomic and metabolomics data. With the goals of understanding the underlying biological mechanisms, novel statistical models and their associated computational tools are urgently needed.

3) *Integration of physical models:* Physical models, often in the form of ordinary differential equations or partial differential equations, are another kind of powerful computational tools for complex systems modeling. There are at least two reasons to believe that an integration of physical models with pure data-driven statistical models will be advantageous in many applications: 1) pure data-driven statistical models maybe less effective when data is scarce relative to the complexity of the model, or when the model is expected for extrapolate the regions where no data has been collected; 2) physical models need a sufficient knowledge about the structure of the systems, i.e., the interactions between the system entities, and also needs to be well parameterized and calibrated. Thus, if we can integrate physical models with statistical models, it is reasonable to expect that this integration will borrow strengths from both sides to overcome the deficiencies for one another, and produce a more powerful model to analyze the complex datasets.

REFERENCES

[1] Acid, S. and De Campos, J. Searching for Bayesian Network Structures in the Space of Restricted Acyclic Partially Directed Graphs. J. Artificial Intelligence Research, vol. 18, pp. 445-490, 2003..

[2] Alexander, G., Moeller, J. (1994) Application of the Scaled Subprofile model: a statistical approach to the analysis of functional patterns in neuropsychiatric disorders: A principal component approach to modeling regional patterns of brain function in disease. Human Brain Mapping, 79-94.

[3] Alexander, G.E., Chen, K., Pietrini, P., Rapoport, S.I., Reiman, E.M. Longitudinal PET Evaluation of Cerebral Metabolic Decline in Dementia: A Potential Outcome Measure in Alzheimer's Disease Treatment Studies. Am.J.Psychiatry 159, 738-745, 2002.

[4] Aliferis, C. F., Tsamardinos, I. and Statnikov, A. HITON, a Novel Markov Blanket Algorithm for Optimal Variable Selection. AMIA, 2003.

[5] Alsop, D., Casement, M. 2008. Hippocampal hyperperfusion in Alzheimer's disease. *NeuroImage* 42, 1267–1274.

[6] Aiken, L.H., Smith, H.L., & Lake, E.T. Lower Medicare mortality among a set of hospitals known for good nursing care. Medical Care, 32, 771-787, 1994.

[7] Aiken, L.H., Clarke, S.P., Sloane, D.M., Sochalski, J.A., & Silber, J.H. Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. Journal of the American Medical Association, 288(16), 1987-1993, 2002.

[8] Aiken, L.H., Clarke, S.P., Sloane, D.M., Lake, E.T., & Cheney, T. Effects of hospital care environment on patient mortality and nurse outcomes. The Journal of Nursing Administration, 38, 223-229, 2008.

[9] Andrews-Hanna, J.R., Snyder, A.Z., et al. Disruption of Large-Scale Brain Systems in Advanced Aging,.Neuron 56, 924–935, 2007.

[10] Argyriou, A., Evgeniou, T. and Pontil, M. 2008. Convex multi-task feature learning. Machine Learning 73(3):243– 272.

[11] Asllani, I., Habeck, C. 2008. Multivariate and univariate analysis of continuous arterial spin labeling perfusion MRI in AD. J. *Cereb. Blood Flow Metab*. 28, 725–736.

[12] Azari, N.P., Rapoport, S.I., Grady, C.L., Schapiro, M.B., Salerno, J.A. and Gonzales-Aviles, A. Patterns of Interregional Correlations of Cerebral Glucose Metabolic Rates in Patients with Dementia of the Alzheimer Type. Neurodegeneration 1: 101–111, 1992.

[13] Banerjee, O., L. El Ghaoui, and A. d'Aspremont. (2008) Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. Journal of Machine Learning Researc 9:485-516.

[14] Bakker, B.; and Heskes, T. “Task Clustering and Gating for Bayesian Multitask Learning”. Journal of Machine Learning Research, 4:83–99, 2003.

[15] Barton, R. R. and Gonzalez-Barreto, D. R. (1996) “Process-oriented Basis Representations for Multivariate Process Diagnostics.” Quality Engineering 9, pp. 107-118.

[16] Baxter, J. “A Model of Inductive Bias Learning”. Journal of Artificial Intelligence Research, 2000.

[17] Bayesian Network Repository: http://www.cs.huji.ac.il/labs/compbio/Repository.

[18] Berge, A., A.C. Jensen, and A.H.S. Solberg. (2007) Sparse inverse covariance estimates for hyperspectral image classification, Geoscience and Remote Sensing, IEEE Transactions on, 45(5):1399-1407.

[19] Bertsekas, D. 1982. Projected newton methods for optimization problems with simple constraints. *SIAM J. Control Optim* 20, 221-246.

[20] Bertsekas, D. Nonlinear Programming, 2nd Edition, Athena Scientific, Belmont, 1999.

[21] Bickel, P. J.; and Levina, E. ‘‘Regularized Estimation of Large Covariance Matrices,’’ Annals of Statistics, 36, 199–227, 2008.

[22] Bidzan, L. 2005. Vascular factors in dementia. *Psychiatr. Pol*. 39, 977-986.

[23] Bilmes. J.A. (2000) Factored sparse inverse covariance matrices. In ICASSP:1009-1012.

[24] Borsuk, M.E., Stow, C.A. andReckhow, K.H. A Bayesian network of eutrophication models for synthesis, prediction, and uncertainty analysis. Ecological Modelling, 173, 219–239, 2004.

[25] Bouckaert, R. Belief Networks Construction Using the Minimum Description Length Principle. Lecture Notes in Computer Science 747, pp. 41-48, 1993.

[26] Braak, H., Braak, E. 1991. Neuropathological stageing of Alzheimer-related changes. *Acta Neuro*. 82, 239–259.

[27] Braak, H., Braak, E., Bohl, J. Staging of Alzheimer-related Cortical Destruction. Eur Neurol 33, 403-408, 1993.

[28] Braak, H., Braak, E. Evolution of the Neuropathology of Alzheimer's Disease. Acta Neurol Scand Suppl 165, 3-12, 1996.

[29] Bradley, K.M., O'Sullivan. 2002. Cerebral perfusion SPET correlated with Braak pathological stage in AD. *Brain* 125, 1772–1781.

[30] Bullmore, E., Horwitz, B., Honey, G., Brammer, M., Williams, S., Sharma, T. (2000) How good is good enough in path analysis of fMRI data? NeuroImage 11, 289–301.

[31] Buntine, W. A Guide to the Literature on Learning Probabilistic Networks from Data. IEEE Trans. Knowledge and Data Eng., vol. 8, 195-210, 1996.

[32] Calhoun, V.D., Adali, T., Pearlson, G.D., Pekar, J.J. (2001) Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms. Hum.Brain Mapp. 13, 43-53.

[33] Calhoun, V.D., Adali, T., Pekar, J.J., Pearlson, G.D. (2003) Latency (in)sensitive ICA. Group independent component analysis of fMRI data in the temporal frequency domain. Neuroimage. 20, 1661-1669.

[34] Campbell, N. 1984. Canonical variate analysis ageneral formulation. *Australian Jour of Stat* 26, 86–96.

[35] Candes, E., Wakin, M. and Boyd, S. 2008. Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier analysis and applications*, 14(5), 877-905.

[36] Carroll, R. J. and Stefanski, L. A. (1990) "Approximate quasilikelihood estimation in models with surrogate predictors." J. Am. Statist. Ass., 85, pp. 652–663.

[37] Carroll, R.J., and Stefanski, L.A., (1997) "Asymptotic theory for the Simex. estimator in measurement error models", Advances in. Statistical Decision Theory and Applications, pp. 151-164.

[38] Carroll, R. J., Stefanski, L. A., and Ruppert, D. (2006). Measurement Error in Nonlinear Models, Second Edition. London: Chapman and Hall.

[39] Caruana, R. "Multitask learning". Machine Learning, 28:41–75, 1997.

[40] Castelo, R. and Kocka, T. On Inclusion-Driven Learning of Bayesian Networks. J. Machine Learning Research, vol. 4, pp. 527-574, 2003.

[41] Ceglarek, D. and Shi, J., (1999) "Fixture Failure Diagnosis for Sheet Metal Assembly with Consideration of Measurement Noise." ASME Transactions, Journal of Manufacturing Science and Engineering. 121, pp771-777.

[42] Chen, X.W., Anantha, G. and Lin, X.T. Improving Bayesian Network Structure Learning with Mutual Information-Based Node Ordering in the K2 Algorithm. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, pp. 628-640, 2008.

[43] Chen, K., Langbaum, J.B.S., Fleisher, A.S., Ayutyanont, N., Reschke, C., Lee, W., Liu, X., Bandy, D., Alexander, G.E., Thompson, P.M., Foster, N.L., Harvey, D.J., de Leon, M.J., Koeppe, R.A., Jagust, W.J., Weimer, M.W., Reiman, E.M., and the ADNI. "Twelve-month metabolic declines in probable Alzheimer's

disease and amnestic mild cognitive impairment assessed using an empirically pre-defined statistical region-of-interest: Findings from the Alzheimer's Disease Neuroimaging Initiative," NeuroImage, 51, 645-664, 2010.

[44] Chiang, L., Russell, E.R. 2001. *Fault detection and diagnosis in industrial systems*. Springer.

[45] Chickering, D., Geiger, D. and Heckerman, D. Learning Bayesian Networks: Search Methods and Experimental Results. Preliminary Papers Fifth Int'l Workshop Artificial Intelligence and Statistics, 1995.

[46] Chickering, D. Optimal Structure Identification with Greedy Search. J. Machine Learning Research, vol. 3, pp. 507-554, 2002.

[47] Chow, C. and Liu, C. Approximating Discrete Probability Distributions with Dependence Trees. IEEE Trans. Information Theory, vol. 14, pp. 462-467, 1968.

[48] Clemmensen, L., Hastie, T., Witten, D. and Ersboll, B. 2011. Sparse Discriminant Analysis. Technometrics (in press)

[49] Cook, J.R. and Stefanski, L.A. (1994) "Simulation-extrapolation estimation in parametric measurement error models." J. Am. Stat. Assoc. 89, pp. 1314–1328.

[50] Cooper, G. and Herskovits, E. A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning, vol. 9, pp. 309-347, 1992.

[51] Cormen, T.H., Leiserson, C.E., Rivest, R.L and Stein, C. Introduction to Algorithms. 3rd edition, MIT Press.

[52] Dahl, J., L. Vandenberghe, and V. Roycowdhury. (2008) Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods Software 23(4):501-520.

[53] Dai, H., Korb, K.B., Wallace, C.S. and Wu, X. A study of casual discovery with weak links and small samples, in Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI), Morgan Kaufman, San Francisco, CA, pp. 1304–1309, 1997.

[54] Daunizeau, J., Grova, C. 2007. Symmetrical event-related EEG/fMRI information fusion. *NeuroImage* 36, 69-87.

[55] De Campos, L. Independency Relationships and Learning Algorithms for Singly Connected Networks. J. Experimental and Theoretical Artificial Intelligence, vol. 10, pp. 511-549, 1998.

[56] De Campos, L. and Huete, J. A New Approach for Learning Belief Networks Using Independence Criteria. Int'l J. Approximate Reasoning, vol. 24, pp. 11-37, 2000.

[57] Dellaert, F. "The Expectation Maximization Algorithm", Technical

Report. GVU Center; College of Computing; Georgia Tech, GIT-GVU-02-20, 2002.

[58] Demidenko, E., Mixed models, theory and applications, New York: John Wiley, 2004.

[59] Dempster, A.P., Rubin, D.B. and Tsutakawa, R.K., Estimation in covariance components models, Journal of the American Statistical Association, 76, 341-353, 1981.

[60] Devanarayan, V., and Stefanski, L.A. (2002) "Empirical simulation extrapolation for measurement error models with replicate measurements." Stat. Probab. Lett. 59, pp. 219-225.

[61] Ding, Y., E. A. Elsayed, et al. (2006). "Distributed sensing for quality and productivity improvements." IEEE Transactions on Automation Science and Engineering 3(4), pp. 344-358.

[62] Dobra, A., C. Hans, B. Jones, J.R. Nevins, G. Yao, and M. West. (2004) Sparse graphical models for exploring gene expression data. Journal of Multivariate Analysis 90(1):196-212.

[63] Donoho. D.L. (2006) For most large underdetermined systems of linear equations, the minimal l1-orm near-solution approximates the sparest near-solution. Communications on Pure and Applied Mathematics 59(7):907-934.

[64] Draper, N.R., Smith, H. Applied Regression Analysis, 3rd Edition. Wiley Press, 1998.

[65] Du,A.T., Jahng, G.H. 2006. Hypoperfusion in frontotemporal dementia and AD. *Neurology* 67, 1215–1220.

[66] Efron, B. and Ribshirani, R.J. An Introduction to the Bootstrap. CRC Press, 1994.

[67] Elston, R.C. and Grizzle, J.E., Estimation of time response curves and their confidence bands, Biometrics, 18, 148-159, 1962.

[68] Estrada, E. and Naomichi, H. Communicability in Complex Networks. Phys. Rev. E 77 036111, 2008.

[69] Friedman, N. and Goldszmidt, M. Learning Bayesian Networks with Local Structure. Proc. 12th Conf. Uncertainty in Artificial Intelligence, 1996.

[70] Friedman, N.; Nachman, I. and Pe'er, D. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. UAI, 1999.

[71] Friedman, N.; Linial, M.; Nachman, I.; and Pe'er, D. "Using Bayesian Networks to Analyze Expression Data". Journal of Computational Biology, 7, 601–620, 2000.

[72] Friedman, N. and Koller, D. Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. Machine Learning, vol. 50, 95-125, 2003.

[73] Friedman, J. ,Hastie, T., Hofling, H. and Tibshirani, R. Pathwise Coordinate Optimization. The Annals of Applied Statistics, 2007 (1), no.2, p.302-332, 2007.

[74] Friedman, J., J. Hastie, and R. Tibsirani. (2007) Spares inverse covariance estimation with the graphical lasso, Biostatistics 8(1):1-10.

[75] Friese, C.R., Lake, E.T., Aiken, L.H., Silber, J., & Sochalski, J.A. Hospital nurse practice environments and outcomes for surgical oncology patients. Health Services Research, 43(4), 1145-63, 2008.

[76] Friston, K.J. (1994) Functional and effective connectivity: A synthesis. Human Brain Mapping 2, 56-78.

[77] Friston, K.J., Ashburner, J. 1995. Spatial registration and normalization of images. *HBM* 2, 89–165.

[78] Friston, K.J., Harrison, L, Penny, W. (2003) Dynamic causal modeling. Neuroimage 19, 1273-1302.

[79] Fu, W. Penalized Regressions: The Bridge vs the Lasso, Journal of Computational and Graphical Statistics, 7 (3), 397–416, 1998.

[80] Gasso, G., Rakotomamonjy, A. and Canu, S. 2009. Recovering sparse signals with non-convex penalties and DC programming. *IEEE Trans. Signal Processing* 57(12), 4686-4698.

[81] Gleser, L.J. (1990) "Improvement of the naive approach to estimation in nonlinear error- in-variables regression models." In Statistical Analysis of Measurement Error Models and applications. Ameri math society, providence

[82] Goldstein, H., Multilevel statistical models, 2nd ed, New York: John Wiley, 1995.

[83] Gonzales-Aviles. (1992) Patterns of interregional correlations of cerebral glucose metabolic rates in patients with dementia of the Alzheimer type. Neurodegeneration 1: 101–111.

[84] Good, P. Permutation, Parametric and Bootstrap Tests of Hypotheses. 3rd ed., Springer, 2005.

[85] Gould, R.L., B.Arroyo, R,G. Brown, A.M. Owen, E.T. Bullmore and R.J. Howard. (2006) Brain Mechanisms of Successful Compensation during Learning in Alzheimer Disease, Neurology 67, 1011-1017.

[86] Greicius, M.D., Srivastava, G., Reiss, A.L. and Menon, V. Default-mode Network Activity Distinguishes AD from Healthy Aging: Evidence from Functional MRI. Proc. Natl. Acad. Sci. 101, 4637–4642, 2004.

[87] Guo, J., Levina, E., Michailidis, G. and Zhu, J. 2011. Joint estimation of multiple graphical models. *Biometrika* 98 (1), 1-15.

[88] Harman, H.H. Modern factor analysis, 3rd Edition, The University of Chicago Press, 1976.

[89] Hastie, T. and Tibshirani, R. 1994. *Discriminant analysis by gaussian mixtures*. Technical report. AT&T Bell Lab.

[90] Hawkins, D.M. (1991). "Multivariate Quality Control Based on Regression-Adjusted Variables," Technometrics 33, pp. 61-75.

[91] Hawkins, D.M. (1993). "Regression Adjustment for Variables in Multivariate Quality Control," Journal of Quality Technology 25, pp. 170-182.

[92] Heckerman, D., Geiger, D. and Chickering, D. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning, vol. 20, 197-243, 1995.

[93] Heckerman, D., Mamdani, A., and Wellman, M.P. (1995), "Real-World Applications of Bayesian Networks," Communications of ACM, 38(3), pp. 24-26.

[94] Heckerman, D., (1999), "A Tutorial on Learning with Bayesian Networks," Learning in Graphical Models, pp. 301–354.

[95] Hedden, T., Van Dijk, K.R., et al Disruption of Functional Connectivity in Clinically Normal Older Adults Harboring Amyloid Burden. J. Neurosci. 29, 12686–12694, 2009.

[96] Hoyer, P.O., Janzing, D., Mooij, J.M., Peters, J. and Scholkopf, B. Nonlinear Causal Discovery with Additive Noise Models. NIPS 21, 2009.

[97] Huang, J.Z., N. Liu, M. Pourahmadi, and L. Liu. (2006) Covariance matrix selection and estimation via penalized normal likelihood. Biometrika, 93(1):85-98.

[98] Huang, S., Li, J., et al. 2010. Learning Brain Connectivity of AD by Sparse Inverse Covariance Estimation, *NeuroImage*, 50, 935-949.

[99] Ikonomovic, M.D., Klunk, W.E., Abrahamson, E.E., Mathis, C.A., Price, J.C., Tsopelas, N.D., Lopresti, B.J., Ziolko, S., Bi, W., Paljug, W.R., Debnath, M.L., Hope, C.E., Isanski, B.A., Hamilton, R.L., DeKosky, S.T.. Post-mortem Correlates of in vivo PiB-PET Amyloid Imaging in a Typical Case of Alzheimer's Disease. Brain 131, 1630-1645, 2008.

[100] Institute of Medicine, Committee on Quality of Health Care in America. Page, A.. (Ed.). Keeping patients safe: Transforming the work environment of nurses. Washington, DC: National Academies Press, 2004.

[101] Ishii, K., Kitagaki, H. 1996. Decreased medial temporal oxygen metabolism in AD. *J. Nucl. Med*. 37, 1159–1165.

[102] Jagust, W. 2006. PET and MRI in the diagnosis and prediction of dementia. *Alzheimer's Dement* 2, 36-42.

[103] Jin, R., Li, J., and Shi, J., (2007), "Quality Prediction and Control in Rolling Processes using Logistic Regression," Transactions of NAMRI/SME, 35, pp. 113-120.

[104] Johnson, N.A., Jahng, G.H. 2005. Pattern of cerebral hypoperfusion in AD. *Radiology* 234, 851–859.

[105] Jolliffe, I.T. Principal component analysis, 2nd Edition, Springer, 2001.

[106] Kawachi, T., Ishii, K. and Sakamoto, S. 2006. Comparison of the diagnostic performance of FDG-PET and VBM. *Eur.J.Nucl.Med.Mol.Imaging* 33, 801-809.

[107] Kazanjian, A., Green, C., Wong, J., & Reid, R. Effects of the hospital nursing environment of patient mortality: a systematic review. Journal of Health Services Research and Policy, 10(2), 111-117, 2005.

[108] Keilp, J.G., Alexander, G.E. 1996. Inferior parietal perfusion, lateralization, and neuropsychological dysfunction in AD. *Brain Cogn.* 32, 365–383.

[109] Klunk, W.E., Engler, H., Nordberg, A., Wang, Y., Blomqvist, G., Holt, D.P., Bergstrom, M., Savitcheva, I., Huang, G.F., Estrada, S., Ausen, B., Debnath, M.L., Barletta, J., Price, J.C., Sandell, J., Lopresti, B.J., Wall, A., Koivisto, P., Antoni, G., Mathis, C.A. and Langstrom, B.. Imaging Brain Amyloid in Alzheimer's Disease with Pittsburgh Compound-B. Ann Neurol. 55, 306-319, 2004.

[110] Korb, K.B. and Nicholson, A.E. (2003). Bayesian Artificial Intelligence. London, UK: Chapman and Hall.

[111] Kumar, N. and Andreou, G. 1998. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech Communication*, 26 (4), 283-297.

[112] Lake, E.T. & Friese, C.R. Variations in nursing practice environments: relation to staffing and hospital characteristics. Nursing Research, 55(1), 1-9, 2006.

[113] Laird, N.M. and Ware, H., Random-effects models for longtitudinal data, Biometrics, 38, 963-974, 1982.

[114] Lam, W. and Bacchus, F. Learning Bayesian Belief Networks. An Approach Based on the MDL Principle. Computational Intelligence, vol. 10, pp. 269-293, 1994.

[115] Lamb, GS, Schmitt, MH, Edwards, P, Sainfort, F., Duva, I., Higgins, M. Measuring Staff Nurse Care Coordination in the Hospital. Invited presentation, National State of the Science Congress on Nursing Research. Washington, DC, 2008.

[116] Langbaum, J.B.S., Chen, K., Lee, W., Reschke, C., Bandy, D., Fleisher, A.S., Alexander, G.E., Foster, N.L., Weiner, M.W., Koeppe, R.A., Jagust, W.J., Reiman, E.M., and the ADNI. "Categorical and correlational analyses of baseline fluorodeoxyglucose positron emission tomography images from the Alzheimer's Disease Neuroimaging Initiative (ADNI)." NeuroImage, 45, 1107-1116, 2009.

[117] Larranaga, R., Kuijpers, C., Murga, R. and Yurramendi, Y. Learning Bayesian Network Structures by Searching for the Best Ordering with Genetic Algorithms. IEEE Trans. Systems, Man, and Cybernetics, vol. 26, pp. 487-493, 1996.

[118] Larranaga, P., Poza, M., Yurramendi, Y., Murga, R. and Kuijpers, C. Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, pp. 912-926, 1996.

[119] Laschinger, H.K.S. & Leiter, M.P. The impact of nursing work environments on patient safety outcomes. Journal of Nursing Administration, 36(5), 259-267, 2006.

[120] Lawrence, N.D.; and Platt, J.C. "Learning to Learn with the Informative Vector Machine". In Proceedings of the 21st International Conference on Machine Learning, 2004.

[121] L. deToledo-Morrell, T. R. Stoub, M. Bulgakova, "MRI-derived entorhinal volume is a good predictor of conversion from MCI to AD," *Neurobiol. Aging, vol.* 25, pp. 1197–1203, 2004.

[122] Lerner, U. (2002). Hybrid Bayesian Networks for Reasoning about Complex Systems, Ph.D. Thesis, Stanford University.

[123] Levina, E.; Rothman, A. J.; and Zhu, J. ''Sparse Estimation of Large CovarianceMatrices via a Nested Lasso Penalty,'' Annals of Applied Statistics, 2, 245–263, 2008.

[124] Li, H.; and Gui, J. ''Gradient Directed Regularization for Sparse Gaussian Concentration Graphs, with Applications to Inference of Genetic Networks,'' Biostatistics, 7, 302–317, 2006.

[125] Li, J., Gutchess, D., Shi, J., and Chang, S. (2003) "Real-Time Surface Defect Detection in Hot Rolling Process," Proceedings of the Iron and Steel Exposition and 2003 AISE Annual Convention, Pittsburgh, PA.

[126] Li, J., and Shi, J., (2007), "Knowledge Discovery from Observational Data for Process Control using Causal Bayesian Networks," IIE Transactions, 39 (6), pp. 681 – 690.

[127] Li, J., Shi, J., and Chang, T.S., (2007) "On-line Seam Detection in Rolling Processes using Snake Projection and Discrete Wavelet Transform," ASME (American Society of Mechanical Engineers) Transactions, Journal of Manufacturing Science and Engineering, 129(5), pp. 926-933.

[128] Li, J., Jin, J., and Shi, J., (2008) "Causation-based T^2 Decomposition for Multivariate Process Monitoring and Diagnosis," Journal of Quality Technology, 40(1), pp. 46-58.

[129] Li, J., Shi, J., and Satz, D., (2008) "Modeling and Analysis of Disease and Risk Factors through Learning Bayesian Network from Observational Data," Quality and Reliability Engineering International, 24(3), 291-302.

[130] Li, J.N., Wang, Z.J., Palmer, S.J., McKeown, M.J. Dynamic Bayesian network modeling of fMRI: A comparison of group-analysis methods, NeuroImage 37, 749–760. 2008.

[131] Li, H., and J. Gui. (2005) Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. Biostatistics 7(2):302-317.

[132] Lin. Y., (2007) Model selection and estimation in the gaussian graphical model. Biometrika 94(1)19-35, 2007.

[133] Lipton, A.M., Benavides, R., Hynan, L.S., Bonte, F.J., Harris, T.S., White, C.L. 3rd, Bigio, E.H. Lateralization on Neuroimaging does not Differentiate Frontotemporal Lobar Degeneration from Alzheimer's Dsease. Dement Geriatr Cogn Disord 17(4), 324-327, 2004.

[134] Liu, J., Ji, S. and Ye, J. 2009. *SLEP: sparse learning with efficient projections*, Arizona state university.

[135] Longford, N. A fast scoring algorithm for maximum likelihood estimation in unbalanced models with nested random effects, Biometrika, 74(4), 817-827, 1987.

[136] Longford, N., Random coefficient models, Oxford: Clarendon, 1993.

[137] Lucas, J.M., and Saccucci, M.S., (1990) "Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements," Technometrics, 32(1), 1-29.

[138] Luus, R. and Wyrwicz, R. Use of Penalty Functions in Direct Search Optimization. Hung. J. Ind. Chem. 24, 273-278, 1996.

[139] Mackey, D. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

[140] Mani, S. and Cooper, G.F. A study in casual discovery from population-based infant birth and death records, in Proceedings of theAMIAAnnual Fall Symposium, Hanley and Belfus, Philadelphia, PA, pp. 315–319, 1999.

[141] Marcot, B.G., Holthausen, R.S., Raphael, M.G., Rowland, M. and Wisdom,M. Using Bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. Forest Ecology and Management, 153(1–3), 29–42, 2001.

[142] Margaritis, D. and Thrun, S. Bayesian network induction via local neighborhoods. NIPS 12, 1999.

[143] Mark, B.A., Hughes, L.C., Belyea, M., Bacon, C.T., Chang, Y. & Jones, C.B. Exploring organizational context and structure as predictors of medication errors and falls. Journal of Patient Safety, 4(2), 66-77, 2008.

[144] Mason, R. L., Tracy, N. D., and Young, J. C. (1995). "Decomposition of T^2 for Multivariate Control Chart Interpretation." Journal of Quality Technology 27, pp. 99-108.

[145] Mason, W.M., Wong, G.M. and Entwistle, B., Contextual analysis through the multilevel linear model. In S.Leinhardt (Ed), Sociological methodology, pp. 72-103, San Francisco: Jossey-Bass, 1983.

[146] Mason, R. L., Tracy, N. D., and Young, J. C. (1997). "A Practical Approach for Interpreting Multivariate T^2 Control Chart Signals." Journal of Quality Technology 29, pp. 396-406.

[147] Matsunari, I., Samuraki, M. 2007. Comparison of 18F-FDG PET and optimized voxel-based morphometry for detection of AD. *J.Nucl.Med* 48, 1961-1970.

[148] Mazumder, R.; Friedman, J. 2009. SparseNet: Coordinate Descent with Non-Convex Penalties. Manuscript.

[149] McIntosh, A.R., Grady, C.L., Ungerieider, L.G., Haxby, J.V., Rapoport, S.I., Horwitz, B. (1994) Network analysis of cortical visual pathways mapped with PET. J. Neurosci. 14 (2), 655–666.

[150] McIntosh, A.R., Bookstein, F.L., Haxby, J.V., Grady, C.L. (1996) Spatial pattern analysis of functional brain images using partial least squares. Neuroimage. 3, 143-157.

[151] Meek, C. Causal Inference and Causal Explanation with Background Knowledge. Proc. 11th Conf. Uncertainty in Artificial Intelligence, 1995.

[152] Meinshausen, N.; and Buhlmann, P. ‘‘High Dimensional Graphs and Variable Selection with the Lasso,’’ Annals of Statistics, 34, 1436–1462, 2006.

[153] Meinshausen, N. and Buehlmann, P. Stability Selection. Journal of the Royal Statistical Society, Series B, vol.72, 417-473, 2010.

[154] Molchan. S. (2005) The Alzheimer's disease neuroimaging initiative. Business Briefing: US Neurology Review, pp.30-32, 2005.

[155] Mosconi, L., Tsui, W.-H. 2005. Reduced hippocampal metabolism in MCI and AD. *Neurology* 64, 1860–1867.

[156] Morra, J. H., Z. Tu, “Validation of automated hippocampal segmentation method,” NeuroImage, vol. 43, 59–68, 2008.

[157] Morra, J.H., Tu, Z. 2009a. Automated 3D mapping of hippocampal atrophy. *Hum. Brain Map*. 30, 2766–2788.

[158] Morra, J.H., Tu, Z. 2009b. Automated mapping of hippocampal atrophy in 1-year repeat MRI data. *NeuroImage* 45, 213-221.

[159] Montgomery, D.C. (2001). Introduction to Statistical Quality Control. New York: John Wiley.

[160] Mulert, C., Lemieux, L. 2010. *EEG-fMRI: physiological basis, technique and applications*. Springer.

[161] Nemirovski, A. “Prox-method with Rate of Convergence o(1/t) for Variational Inequalities with lipschitz Continuous Monotone Operators and Smooth Convex-Concave Saddle Point Problems”. SIAM Journal on Optimization, 15(1):229–251, 2005.

[162] Pearl, J. and Verma, T. Equivalence and Synthesis of Causal Models. Proc. Sixth Conf. Uncertainty in Artificial Intelligence, 1990.

[163] Pellet, J.P. and Elisseeff, A. Using Markov Blankets for Causal Structure Learning. Journal of Machine Learning Research 9, 1295-1342, 2008.

[164] Peng, J.; Wang, P.; Zhou, N.F.; and Zhu, J. “Partial Correlation Estimation by Joint Sparse Regression Models”, Journal of the American Statistical Association, 104(486), 735-746, 2009.

[165] Philiastides, M. and Sajda, P. 2007. EEG-informed fMRI reveals spatiotemporal characteristics of perceptual decision making. *Journal of Neuroscience*, 27(48), 13082-13091.

[166] Pierce, D.A. and Kellerer, A.M. (2005) “Adjusting for covariate errors with nonparametric assessment of the true covariate distribution.” Biometrika 91, pp. 863-876.

[167] Qiao, Z., Zhou, L and Huang, J. 2006. Sparse LDA with applications to high dimensional low sample size data. *IAENG applied mathematics*, 39(1).

[168] Rajapakse, J.C., Zhou, J. Learning effective brain connectivity with dynamic Bayesian networks. NeuroImage 37, 749–760, 2007.

[169] Ravikumar, P., Raskutti, G., Wainwright, M. J. and Yu, B. "Model selection in Gaussian graphical models: High-dimensional consistency of l1-regularized MLE." Advances in Neural Information Processing Systems (NIPS) 21, 2008.

[170] Reuter-Lorenz, P.A. and Mikels, J.A. A Split-Brain Model of Alzheimer's Disease? Behavioral Evidence for Comparable Intra and Interhemispheric Decline. Neuropscyhologia 43, 1307-1317, 2005.

[171] Robinson, W.S. Ecological Correlations and the Behavior of Individuals. American Sociological Review, 15, 351-357, 1950.

[172] Rodin, A. S. and Boerwinkle, E. Mining genetic epidemiology data with Bayesian networks I: Bayesian networks and example application (plasma apoE levels). Bioinformatics, 21(15), 3273–3278, 2005.

[173] Rosenberg, B. Linear regression with randomly dispersed parameters, Biometrika, 60, 61-75, 1973.

[174] Runger G. C., Barton, R. R., Castillo, E. D., and Woodall, W. H. (2007). "Optimal Monitoring of Multivariate Data for Fault Patterns." Journal of Quality Technology 39 (2), pp. 159-172.

[175] Schafer, J., and Strimmer, K. ''A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics,'' Statistical Applications in Genetics and Molecular Biology, 4 (1), Article 32, 2005.

[176] Schmidt, M., Fung, G. and Rosales, R. 2007. Fast optimization methods for L1-regularization: a comparative study and 2 new approaches. *ECML* 2007.

[177] Schmidt, M.; Niculescu-Mizil, A. and Murphy, K. Learning Graphical model structures using L1-Regularization paths. AAAI 2007.

[178] Schroeter, M.L., Stein, T. 2009. Neural correlates of AD and MCI. *NeuroImage* 47, 1196–1206.

[179] Shu, L., Tsung, F. and Tsui, K.-L., (2005), "Effects of Estimation Errors on the Performance of Cause-Selecting Charts", IIE Transactions, 37, pp. 559-567.

[180] Shu, L, Tsung, F, Tsui, K.L. (2004) "Run-length performance of regression control charts with estimated parameters." Journal of Quality Technology 36, pp. 280–292.

[181] Singer, J.D., Using SAS PROC MIXED to fit multilevel models, hierarchical models: Issues and methods, Journal of Educational and Behavioral Statistics, 23(4), 323-355, 1998.

[182] Spirtes, P., Glymour, C. and Scheines, R. Causation, Prediction and Search. Lecture Notes in Statistics 81, Springer, 1993.

[183] Spirtes, P., Glymour, C., and Scheines, R., (2000). Causation, Prediction, and Search. The MIT Press: Cambridge, MA.

[184] Sporns, O., Chialvo, D. R., Kaiser, M., and Hilgetag, C. C. Organization, Development and Function of Complex Brain Networks. Trends Cogn. Sci, 8, 418–425, 2004.

[185] Stam, C.J., Jones, B.F., Nolte, G., Breakspear, M. and Scheltens, P. (2007) Small-world networks and functional connectivity in Alzheimer's disease. Cerebral Corter 17:92-99.

[186] Stern, Y. (2006) Cognitive Reserve and Alzheimer Disease, Alzheimer Disease Associated Disorder 20, 69-74.

[187] Sun, L.; Patel, R.; Liu, J.; Chen, K.; Wu, T.; Li, J.; Reiman, E.; and Ye, J. "Mining Brain Region Connectivity for Alzheimer's Disease Study via Sparse Inverse Covariance Estimation". Proceedings of Knowledge Discovery and Data Mining Conference (KDD), 2009.

[188] Sun, Y., Zhang, W.Y. and Tong, H., Estimation of the covariance matrix of random effects in longitudinal studies, The Annals of Statistics, 35(6), 2795-2814, 2007.

[189] Supekar, K., Menon, V., Rubin, D., Musen, M., Greicius, M.D. (2008) Network Analysis of Intrinsic Functional Brain Connectivity in Alzheimer's Disease. PLoS Comput Biol 4(6) 1-11.

[190] Suzuki, J. A Construction of Bayesian Networks from Databases Based on an MDL Principle. Proc. Ninth Conf. Uncertainty in Artificial Intelligence, pp. 266-273, 1993.

[191] Thrun, S. and O'Sullivan, J.. "Discovering structure in multiple learning tasks: The TC algorithm." In International Conference on Machine Learning, 489-497, 1996.

[192] Tibshirani. R. (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B 58(1):267-288.

[193] Tosun, D., Mojabi, P. 2010. Joint analysis of structural and perfusion MRI for cognitive assessment and classification of AD and normal aging. *NeuroImage* 52, 186-197.

[194] Tsamardinos, I. and Aliferis, C. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. Artificial Intelligence and Statistics, 2003.

[195] Tsamardinos, I.; Brown, L.E., and Aliferis, C.F. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning, **65**(1), 31-78, 2006.

[196] Tsamardinos, I., Statnikov, A., Brown, L. E., and Aliferis, C. F. Generating Realistic Large Bayesian Networks by Tiling. In The 19th International FLAIRS Conference, 2006.

[197] Tseng, P. "Convergence of Block Coordinate Descent Method for Nondifferentiable Maximation". J. Opt. Theory and Applications, 109(3):474–494, 2001.

[198] Tzourio-Mazoyer N. and et al. (2002) Automated anatomical labelling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single subject brain. Neuroimage 15:273-289.

[199] Valdés-Sosa, P., Sánchez-Bornot, J., Lage-Castellanos, A., Vega-Hernández, M., Bosch-Bayard, J., Melie-García, L. and Canales-Rodríguez, E. "Estimating brain functional connectivity with sparse multivariate autoregression." Philos. Trans. Roy. Soc. B: Biological Sciences 360 969–981, 2005.

[200] Wade, M.R., Woodall, W.H. (1993), "A review and analysis of cause-selecting control charts", Journal of Quality Technology, 25 pp.161-169.

[201] Wang, N., Carroll, R. J.&Liang, K. Y. (1996). "Quasilikelihood estimation in measurement error models with correlated replicates". Biometrics 52, pp. 401–411.

[202] Wang, K., M. Liang, L. Wang, L. Tian, X. Zhang, K. Li and T. Jiang. (2007) Altered Functional Connectivity in Early Alzheimer's Disease: A Resting-State fMRI Study, Human Brain Mapping 28, 967-978.

[203] Wolf, H., Jelic, V. 2003. A critical discussion of the role of neuroimaging in MCI. *Acta Neuroal*: 107 (4), 52-76.

[204] Wu, C.F.J. "On the Convergence Properties of the EM Algorithm", The Annals of Statistics, 11, 95-103, 1983.

[205] Wu, X., Li, R., Fleisher, A.S., Reiman, E.M., Chen, K. and Yao, L.. Altered Default Mode Network Connectivity in AD -- A Resting Functional MRI and Bayesian Network Study. Human Brain Mapping, in press.

[206] Worsley, K.J., Poline, J.B., Friston, K.J., Evans, A.C. (1997) Characterizing the response of PET and fMRI data using multivariate linear models. Neuroimage. 6, 305-319.

[207] Xu, L., Qiu, C., Xu, P. and Yao, D. 2010. A parallel framework for simultaneous EEG/fMRI analysis: methodology and simulation. *NeuroImage*, 52(3), 1123-1134.

[208] Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research, 8, 35–63, 2007.

[209] Yuan, M.; and Lin, Y. ‘‘Model Selection and Estimation in the Gaussian Graphical Model,’’ Biometrika, 94 (1), 19–35, 2007.

[210] Zhang, G. (1984), “A new type of control charts - cause-selecting control charts and a diagnosis theory with control charts”, Proceedings of World Quality Congress ’84, pp.175-85.

[211] Zhang, J.; Ghahramani, Z.; and Yang, Y. “Learning Multiple Related Tasks using Latent Independent Component Analysis”. In Y. Weiss, B. Sch ̈olkopf, and J. Platt, editors, Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA, 2006.

[212] Zhang, T. 2008. Multi-stage Convex Relaxation for Learning with Sparse Regularization. *NIPS* 2008.

[213] Zou, H., Hastie, T. and Tibshirani, R. 2006. Sparse PCA, *J. of Comp. and Graphical Statistics*, 15(2), 262-286.