MULTIMODAL LEARNING AND ITS APPLICATION TO MODELING ALZHEIMER'S DISEASE

By

Qi Wang

A DISSERTATION

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

Computer Science — Doctor of Philosophy

2020

ABSTRACT

MULTIMODAL LEARNING AND ITS APPLICATION TO MODELING ALZHEIMER'S DISEASE

By

Qi Wang

Multimodal learning gains increasing attention in recent years as heterogeneous data modalities are being collected from diverse domains or extracted from various feature extractors and used for learning. Multimodal learning is to integrate predictive information from different modalities to enhance the performance of the learned models. For example, when modeling Alzheimer's disease, multiple brain imaging modalities are collected from the patients, and effectively fusion from which is shown to be beneficial to predictive performance.

Multimodal learning is associated with many challenges. One outstanding challenge is the severe overfitting problems due to the high feature dimension when concatenating the modalities. For example, the feature dimension of diffusion-weighted MRI modalities, which has been used in Alzheimer's disease diagnosis, is usually much larger than the sample size available for training. To solve this problem, in the first work, I propose a sparse learning method that selects the important features and modalities to alleviate the overfitting problem. Another challenge in multimodal learning is the heterogeneity among the modalities and their potential interactions. My second work explores non-linear interactions among the modalities. The proposed model learns a modality invariant component, which serves as a compact feature representation of the modalities and has high predictive power. In addition to utilize the modality invariant information of multiple modalities, modalities may provide supplementary information, and correlating them in the learning can be more informative. Thus, in the third work, I propose multimodal information bottleneck to fuse supplementary information from different modalities while eliminating the irrelevant information

from them. One challenge of utilizing the supplementary information of multiple modalities is that most work can only be applied to the data with complete modalities. Modalities missing problem widely exists in multimodal learning tasks. For these tasks, only a small portion of data can be used to train the model. Thus, to fully use all the precious data, in the fourth work, I propose a knowledge distillation based algorithm to utilize all the data, including those that have missing modalities while fusing the supplementary information.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Jiayu Zhou, for his insight, guidance, and supporting during my Ph.D study. I benefited from his advice, particularly when exploring new ideas and writing papers. This dissertation would not have been possible without the assistance of him. The experiences with him are my lifelong assets. Also, I am very grateful to my co-advisor, Dr. Pang-Ning Tan, for his scientific advice, knowledge and many insightful discussions and suggestions. I would like to thank the committee members, Dr. Jiliang Tang and Dr. Chenxi Li, for their valuable interactions and feedback.

I would like to extend my sincere thanks to Dr. Patricia Sonora, Dr.Kendra Spence Cheruvelil and the members from Data-Intensive Landscape Limnology Lab. I have had the pleasure to work with them for three years. I gratefully acknowledge the assistance of Dr. Liang Zhan, Dr. Paul Thompson and Dr. Hiroko Dodge. I must also thank the members of ILLIDAN Lab as they inspired me a lot through discussions and seminars.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The wide availability of data from multiple data modalities has brought increasing attention to multimodal learning. In general, modalities are defined as sets of heterogeneous features that are collected from diverse domains or extracted from various feature extractors [126]. The sets of features could provide both shared and supplementary information of the subjects. Since different modalities are extracted from different domains or feature extractors, the representations of the modalities may be very distinct from each other. Multimodal learning is to integrate predictive information from different modalities to enhance the performance of the learned models. For example, it is common that images are accompanied by text descriptions or categorical tags. Leveraging information from tags and text descriptions usually provides a more complementary description of images than images alone because of the inherent relatedness. Moreover, since the data collection or feature extraction process for the modalities are separately, the noise induced by the collection or extraction process is specific to each data modalities. Multimodal learning can reduce the effect of noise by learning the common structure across multiple modalities. Therefore, learning from multiple modalities can potentially help to improve performance. As another example, in the medical area, multiple data such as different kind of MRI data, gene data, blood biochemical index are available. When doctors diagnose some complex disease such as Alzheimer's disease, they usually ask the patients to do multiple tests such as brain imaging tests, laboratory tests, mental status tests and neuropsychological tests. Different test result provides different type of evaluation of the

patients. Combining the all the results provides comprehensive and accurate information of the patient and can help the doctors rule out other conditions that cause similar symptoms. Therefore, when using machine learning algorithms to modeling Alzheimer's disease, multimodal learning demonstrated better performance than single modal learning [117, 135, 91, 143].

Multimodal learning is associated with many challenges. (1) Since multiple modalities are used when building the model, the total feature dimension is much larger than the feature dimension for single modal learning models. If directly concatenat the features from different modalities and build single modal learning models, the models may suffer from severe overfitting problem, especially when the feature dimension of each modality is considerably large. Thus, the first challenge is how to build robust models for modalities that have high feature dimensions to prevent overfitting problem. (2) The second challenge for multimodal learning is the heterogeneity among the modalities and their potential interactions. Since modalities are collected from diverse domains or feature extractors, they are not linearly interacted. Using linear models for these modalities limits the power of multimodal learning. (3) One motivation to use multimodal learning is that different modalities provide supplementary information to the subjects. Modalities may have noise or irrelevant information to the following tasks. When learning the common structure of modalities, the irrelevant information and noise are automatically eliminated. However, when combining the supplementary information from modalities and learning a joint representation, the irrelevant information and noise are not removed. So, the third challenge is how to eliminate the noise of irrelevant information from the modalities and only leave useful information when learning the joint representation of all the modalities. (4) Most exiting works that address the supplementary information across the modalities can only be applied to the data with complete modalities, which wastes a lot of precious data. The last challenge I would like to address is the how to build multimodal models with the data having missing modalities.

In this dissertation, I propose four approaches to address the aforementioned challenges respectively. In my first work, I propose a sparse model to select the important features as well as modalities to alleviate the overfitting problem for multimodal learning when the modalities' feature dimension or the modalities number is too large. In my second work, I propose a framework to fuse multiple data modalities for predictive modeling using deep matrix factorization, which explores the non-linear interactions among the modalities and exploits such interactions to transfer knowledge and enable high-performance prediction. Specifically, the proposed collective deep matrix factorization decomposes all modalities simultaneously to capture non-linear structures of the modalities in a supervised manner, and learns a modality-specific component for each modality and a modality invariant component across all modalities. The modality invariant component serves as a compact feature representation of patients that has high predictive power. To solve third challenge, I propose a supervised multimodal learning framework based on the information bottleneck principle to filter out irrelevant and noisy information from multiple modalities and learn an accurate joint representation. Specifically, the proposed method maximizes the mutual information between the labels and the learned joint representation while minimizing the mutual information between the learned latent representation of each modality and the original data representation. For the fourth challenge, I propose a framework based on knowledge distillation, utilizing the supplementary information from all modalities, and avoiding discarding data with missing modalities. Specifically, I first train models on each modality independently using all the available data. Then the trained models are used as teachers to teach the student model, which is trained with the samples having complete modalities.

The four approaches are all validated on Alzheimer's disease (AD) modeling problems. AD is a severe neurodegenerative disease causing 60% to 70% dementia [124]. It starts with vanished memory and progresses to an advanced stage followed by cognitive function loss, which

ultimately leads to death. Currently, AD ranks the sixth leading cause of death in the U.S. and the number of patients affected is expected to reach 13.4 million by the year 2050, which induces substantial burden on the healthcare system [8]. The transitional stage between expected cognitive decline of normal aging and AD, mild cognitive impairment (MCI) has been considered as suitable for possible early therapeutic intervention for AD [85]. Effective diagnosis of MCI or dementia can greatly benefit public health and reduce healthcare burden. Alzheimer's disease can only be definitively diagnosed after death by exterminating the brain tissue in an autopsy [19]. Occasionally, doctors determine whether a person is a possible patient or normal aging using biomarkers of the living body. One commonly used biomarker is brain medical imaging as it shows the microscopic structure of the brain and has the key role as a "window on the brain" [51]. However, analyzing brain medical imaging results requires considerable time and effort. In the areas that lack of doctors experienced in Alzheimer's disease, it is difficult to diagnose the disease even with brain medical imaging. In the past years, various machine learning models have been developed to model diseases [110, 88] and some of them even have better diagnostic accuracy than experienced doctors [79]. Therefore, developing effective machine learning models could greatly reduce the cost needed to diagnosis Alzheimer's disease. In this dissertation, I show how to apply the proposed algorithm to Alzheimer's disease diagnosis. In the following four subsections, I give a brief introduction to each work.

1.1 Discriminative Fusion of Multiple Brain Networks

Recently, with the development of diffusion-weighted magnetic resonance imaging techniques that map patterns of connections in the brain. Many researchers have begun to model the brain as a network of interconnected brain regions, or connectome [102]. The properties of these networks

can then be studied mathematically with network theory. Mathematically, a brain network at the macro-scale is typically expressed by a connectivity matrix, in which each element represents some property of the connection between each pair of brain regions [101]. These network-derived features provide clues about how characteristic network disruptions occur and how they may progress in Alzheimer's disease. Diffusion MRI is a variant of standard anatomical MRI that is sensitive to microscopic properties of the brain's white matter that are not detectable with standard anatomical MRI. The general process of reconstructing a structural brain network includes two main steps [134]. The first step extracts the dominant diffusion direction(s) at each voxel based on a diffusion MRI signal model. Some popular models include the diffusion tensor, the orientation distribution function (ODF), or a probabilistic mixture of tensors [70], among others. The next step is whole brain tractography based on these voxel- level diffusion direction(s). Currently, there are two main classes of tractography methods: deterministic and probabilistic approach. Based on whole brain tractography result, brain networks can be computed by combining the pattern of fiber tracts with some specific anatomical partitioning scheme, and measuring some property of the connections between each pair of brain regions, such as their density or integrity.

Theoretically, different algorithmic methods to map structural connections should ultimately provide a consistent anatomical description of the brain. Even so, this may not be true in reality. Different tractography methods recover different sets of fibers (Fig. 1.1), and the fiber bundles that best differentiate patients from controls may be extracted by some algorithms but not others [133]. Different tract tracking methods vary in their ability to perform robustly on dataset of different quality. And there is no general principle to decide which tractography method or network model is most sensitive to disease effects in clinical research studies [134]. I therefore combine all these networks and build predictive model with them. The challenge to combine these networks is that the dimensions for the modalities are too high compared with the sample size. To address

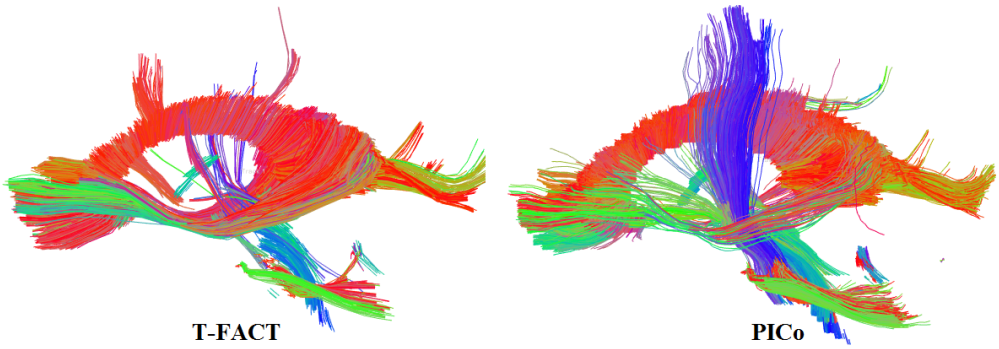


Figure 1.1: **Different tractography methods detect different sets of fibers.** Here I show the fibers generated by two tractography algorithms (T-FACT [78] and PICo [84]), passing through the same brain slice.

this challenge, I create a sparse learning framework to optimally fuse the networks. The benefit of sparse learning is it selects important components. Therefore, the feature dimension and the modality numbers are reduced and the overfitting problem is alleviated.

1.2 Multimodal Disease Modeling via Collective Deep Matrix Factorization

In addition to the brain networks mentioned in the first work, there are multiple biological measures such as T1 weighted MRI and genotype available. T1-weighted MRI (T1 MRI) can capture structural information of gray matter in the brain. Combining T1 MRI and brain networks from diffusion weighted MRI together provides a comprehensive illustration of the brain than utilizing them separately. Moreover, prior studies strongly favor a joint analysis on multiple modalities including imaging and genetics, since it has been shown that genetic variants have played a significant role in the onset of the disease [93, 15, 130, 129]. Combining the three modalities provide complementary information on brain structure and function, thus improve capability in differentiating between normal aging subjects and MCI patients [116, 89].

However, few prior studies combined two types of MRI imaging in detecting MCI, let alone

a joint model that incorporate imaging modalities and genetic information. One reason is the limited sample size. It is usually very costly to construct large cohort studies that involve imaging and genetic data. For example, more than $60 million has been devoted to the first stage of Alzheimer's Disease Neuroimaging Initiative (ADNI) to collect 819 subjects' brain imaging data, genetic data and other biological samples. Different biological data modalities have different feature dimensions. For example, imaging data contains hundreds to thousands features, while the feature dimension of genetic data is around 1 million. Due to the high dimensionality of brain images and genetic markers, directly combining multiple modalities will increase the feature dimension drastically, which not only makes it difficult to extract valid predictive signals, but also induces overfitting problems. Also, some subjects do not have genetic data or dMRI data because they did not participate some parts of the study. Directly combining multiple modalities means those subjects must be discarded, which significantly reduces the sample size. Moreover, different modalities describe different aspects of brain: T1 MRI captures areas composed of neurons while dMRI estimates connection between those areas; the genotype impacts the disease in a way that is not directly related to brain structure and function. As such, all these data modalities are interacting in a complicated manner, suggesting that directly combining feature spaces may not lead to effective integration.

Analysis of high dimensional data can greatly benefit from its intrinsic low-rank structures since exploiting the low-rank structure of the high dimensional data allows us to significantly reduce the feature dimensionality while maintain most information in data. When the sample size is limit, it could reduce the overfitting problem. Recent studies have identified such low-rank properties in imaging and genetic data [142, 73, 121]. Matrix factorization techniques [68, 61] are powerful tools to recover the low rank structure of a matrix and have been widely used in many data mining and machine learning applications. Because of its capability to denoise data, such

approach is especially attractive in processing noisy data such as genetics and imaging. Matrix factorization also provides an integrated approach to fuse multiple data modalities by mapping different modalities to a shared subspace. This method has been widely applied in network analysis [25] and clustering [6]. Matrix factorization techniques have a strong linear assumption that objects interact with each other linearly in a low dimensional subspace. However, brain as well as genotype-phenotype interactions have inherent complex structure [36, 62, 41]. For example, it has been identified that human brain functional networks have a hierarchical modular organization structure [76]. Thus, the linear assumption in traditional matrix factorization may fail to capture the complexity, nonlinearities and hierarchical interactions among different modalities in AD research.

In this work, I propose a deep matrix factorization framework to fuse information from multiple modalities and transfer predictive knowledge in order to differentiate MCI patients from cognitive normal subjects. Specifically, I build a nonlinear hierarchical deep matrix factorization framework which decomposes each modality into a modality invariant component and a modality specific component guided by supervision information. The proposed collective deep matrix factorization delivers higher predictive performance than its linear counterpart, since its deep nonlinear structure can discover the hidden complexity and nonlinearity of original data, and map original data which are not linear separable into a representation that can make subjects easier to be separated. Moreover, the modality specific term can be used to uncover complicated interactions among different modalities that cannot be discovered by traditional matrix factorization methods. I perform extensive empirical studies on the Alzheimer's Disease Neuroimaging Initiative dataset [1] to identify MCI patients by fusing three modalities including T1 MRI, dMRI, and genotype. I also compare the proposed method with state-of-the-art deep multimodal algorithms including deep neural net-

[1] http://adni.loni.usc.edu

work, DCCA [9] and DCCAE [119]. The results demonstrate the effectiveness of the proposed approach.

1.3 Multimodal Information Bottleneck

In addition to learn the common structure of the modalities, another motivation to use multimodal learning is that multiple modalities provide supplementary descriptions of the same subjects and correlating them in the learning can be more informative. When utilizing all the information from different modalities, the performance is expected to be improved compared to learning with the information from only one modality. During the past years, multiple methods have been proposed to combine the supplementary information. For example, kernel-based algorithms use the multiple kernel methods to combine the kernels of different modalities from linear combination methods such as linear convex combination [113] to nonlinear combination methods [114]. With the development of deep learning, multiple neural networks [92, 64] are used to extracted abstract feature representations for each modality. Then, the extracted representations from all modalities are fused in different ways such as concatenation to combine the supplementary information.

When learning common structure of the modalities, the noise or irrelevant information could be eliminated automatically. However, when learning the joint representation of the modalities and fuse all the supplementary information together, the noise or irrelevant information is very likely to be included into the joint representation, which increase the model complexity and cause overfitting problem. Therefore, how to effective fuse the useful supplementary information from all the modalities are very challenging.

More recently, a novel supervised learning method [127] based on the information bottleneck principle [108] has gained increasing attention due to its ability to find a concise representation

of the features, taking into account the trade-off between performance and complexity from an information theory perspective. However, the main drawback of this method is that it employs a linear projection to bridge the representation of each modality. As the relationship between different modalities are often complicated, a simple linear projection would constrain the type of information that can be fused from the different modalities.

Deep learning has been successfully used to learn abstract representation from the raw input data [67]. DCCA and DCCAE are two examples of successful methods using deep neural networks to extract features from each modality and learn their joint representation. These methods have demonstrated better performance compared to traditional linear CCA. However, adopting deep neural networks to information bottleneck based multimodal learning formulation remains a challenging problem. For the information bottleneck approach, the information between different representations are measured in terms of their mutual information. Computing mutual information requires estimation of the posterior distribution, which is computationally intractable when the model is complicated.

In this work, I propose a deep multimodal information bottleneck method to fuse supplementary knowledge from multiple modalities to improve predictive performance. The proposed framework consists of two parts. The first part is to extract concise and relevant latent representation from each modality while the second part fuses the latent representations to learn the joint representation of all modalities. The proposed deep multimodal learning framework adopts the information bottleneck principle to supervise the learning by finding the best representation that balances model complexity and accuracy. The framework also employs a variational inference approach [59, 7] to overcome the challenge of computing mutual information efficiently. The variational inference approach provides an approximate solution to the original optimization problem by maximizing the variational lower bound of the target objective function. Since the variational bound can be easily

optimized by standard gradient descent methods, the problem becomes computationally tractable. I apply this algorithm to the classification of MCI with NC for Alzheimer's disease and the results show signification performance improvement compared with the baselines.

1.4 Multimodal Learning with Incomplete Modalities

One common drawback of the methods fusing the supplementary information is that they usually can only be trained on the samples that have complete modalities, and in practice there are very few samples of such kind, especially when considering a large number of modalities. For example, when studying Alzheimer's disease, only partial subjects have the diffusion-weighted MRI while only another part of subjects has genetic data available. The existing methods may have to discard a large portion of data collected through huge efforts. One solution to deal with the data with incomplete modalities is to impute the missing modalities. After imputation, standard multimodal learning methods can be used to combine the supplementary information. The incompleteness of modalities leads to block missing of features. Therefore, classical matrix completion methods such as matrix factorization [131] and etc. can not be used to impute the missing modalities. Some advanced imputation methods such as cascaded residual autoencoder [111] and adversarial training [21, 118, 105, 83], which have similar structure as GAN, have been proposed to deal with the modality missing problem. These solutions, however, may introduce unwanted imputation noise when imputing the missing modalities [37]. Especially when the size of samples having complete modalities is small, the modalities imputed by such methods may have a negative effect on the performance of the following tasks [37].

In this work, I propose a new multimodal learning framework to integrate the supplementary information of multiple modalities. This method utilizes all the samples include the ones with

incomplete modalities. The proposed method is based on knowledge distillation [46]. I first train models for each modality separately with all the data available. Then, I treat the trained models as teachers to teach a student model. The student model is a multimodal learning model which fuses the supplementary information from multiple modalities. It is trained with the soft labels labeled by the teacher models and the true one-hot label. Since the teacher models are trained with each modality separately, the sample size is much larger than the samples used to train the student model. With enough data, the well-trained teachers act as experts on each modality. The student then learns from these experts and combine the knowledge from all the experts. Compared with existing methods, our method does not discard the samples with incomplete modalities nor impute them. Instead, I use these samples to train the teacher models to make sure the teacher models are experts. To verify the effectiveness of our method, I demonstrate experiments on synthetic data and real-world data such as Alzheimer's disease dataset and some benchmark datasets.

Chapter 2

Related Works

2.1 Co-training Approach

Co-training is a semi-supervised approach. It is first proposed to deal with a classification setting in which limited labeled samples and a large number of unlabeled samples are available for two distinct modalities [16]. There are two assumptions on co-training:

- Each data modality provides complementary information of the samples;
- The two data modalities are conditionally independent given the class labels.

Co-training method separates the samples into two sets, labeled set L and unlabeled set U. It first creates a smaller pool $U' \subseteq U$. Then two weak classifiers are trained for the two modalities, i.e., h_1, h_2, using the limited labeled data from L. Next, h_1 and h_2 are used to label p positive samples and n negative samples that they feel most confident from U'. Those newly labeled samples are then added to L. U' is replenished by drawing $2p+2n$ samples from U randomly. Now, L is enlarged by the $2n+2p$ samples. Those steps are repeated for a predefined number of steps, and finally, two descent classifiers will be obtained. The intuition of this method is to use the samples added by h_1 to train h_2 and vice versa [65]. After repeating for enough times, h_1 and h_2 will agree with each other. Hence, the unlabeled data here is used to prune the hypothesis space for h_1 and h_2 such that the final search spaces are compatible.

We note that one assumption of co-training is that each modality is conditionally independent given class labels. The intuition behind this assumption is that, when two modalities are conditionally independent, each time the added samples are as informative as random samples and the learning should thus progress [81]. However, in some cases, this assumption cannot be satisfied. Then, the added samples may not be informative and the learning process may fail. Co-EM algorithm is an algorithm based on the original co-training algorithm to loosen this assumption [81]. It can be shown that co-EM works even when the conditional independence assumption is violated. Denote the two modalities as s_1 and s_2. This algorithm first trains a classifier using the labeled data from L on s_1. Denote this classifier as h_1. Then h_1 is used to probabilistically label all the unlabeled data in U. Next, another classifier h_2 is trained using the labeled data and the probabilistic labeled data on s_2, and h_2 is used to re-label the data in U. Repeat those steps for some iterations and the final classifiers are obtained. Compared with co-training, co-EM uses one learner to assign labels to all the unlabeled samples and the second classifier is learned using all the probabilistic labeled samples. Hence, it does not require the added samples to be as informative as random samples. However, since co-EM needs to assign probabilistic labels, the classifiers that can be used are limited.

Co-regularization approach [95] is developed based on co-training. This method uses regularization to reach an agreement across different modalities. Denote $\mathscr{H}_1$ and $\mathscr{H}_2$ as two Reproducing Kernel Hilbert Spaces of functions defined on the input space. Denote the labeled set as L and unlabeled set as U. Co-regularization learns the following prediction function[96]:

$$f^* = \frac{1}{2}(f_1^*(x) + f_2^*(x)), \tag{2.1}$$

where $f_1^* \in \mathscr{H}_1$, $f_2^* \in \mathscr{H}_2$, f_1^* and f_2^* are learned by a convex optimization problem:

$$(f_1^*, f_2^*) = \underset{f_1 \in \mathscr{H}_1, f_2 \in \mathscr{H}_2}{\arg\min} \gamma_1 \|f_1\|_{H_1}^2 + \gamma_2 \|f_2\|_{H_2}^2 + \mu \sum_{i \in U} [f_1(x_i) - f_2(x_i)]^2 + \sum_{i \in L} V(y_i, f(x_i)), \quad (2.2)$$

where the first two terms are used to control the model complexity, and γ_1, γ_2 are regularization parameters, the third term is used to enforce that the learned hypotheses agree with each other on different modalities for the unlabeled data. The last term is the empirical loss on the labeled data using $f = \frac{1}{2}(f_1 + f_2)$, and V denotes the loss function. Compared with co-training, this method is non-greedy, convex and easy to implement [95, 96]. There are also multiple variants of co-regularization dealing with different problems, such as Co-regularized Least Squares which minimizes the agreement in a least-square sense [95, 18], Co-regularized Laplacian SVM [95], co-regularized clustering [65] which uses co-regularization to regularize the clustering hypothesis to obtain consistent clusters across different modalities.

2.2 Linear approaches

2.2.1 Canonical correlation analysis

Given two sets of variables, when the number of variables is large, it is not easy to use the covariance matrix of those two sets of variables to find the dependence between them. Sometimes even if the variable number is small, in the current coordinate system, it is still hard to see the relation between them directly. Canonical correlation analysis (CCA) is a widely used method to solve this challenging problem [49]. CCA identifies the relation between two sets of variables by maximizing the correlation between the weighted linear combination of one set of variables and that of the other set of variables. It can be viewed as projecting the original two sets of variables to a low-

dimensional subspace, such that the correlation between the two set of variables is maximized in the new subspace. Hence, it is much easier to analyze variable dependence in the learned subspace than in the original spaces. We will review the classical CCA and its applications to multimodal learning in this section.

Given two random vectors $x \in \mathbb{R}^d$ with the mean m_x and $y \in \mathbb{R}^p$ with the mean m_y. We assume that $d > p$. A random vector is defined to be a vector of random variables. The correlation between x and y measures the linear relation between the two random vectors. Consider the following linear combination:

$$a = w_x^T x, \tag{2.3}$$

$$b = w_y^T y, \tag{2.4}$$

where a and b are two random variables and $w_1 \in \mathbb{R}^d$, $w_2 \in \mathbb{R}^p$. The correlation between a and b is given by:

$$\text{Corr}(a,b) = \frac{w_x^T \Sigma(x,y) w_y^T}{(w_x^T \Sigma(x,x) w_x)^{1/2} (w_y^T \Sigma(y,y) w_y)^{1/2}}. \tag{2.5}$$

CCA seeks vectors w_1 and w_2 such that $\text{Corr}(a,b)$ is maximized, i.e.,

$$w_x^*, w_y^* = \underset{w_x, w_y}{\arg\max}\ \text{Corr}(a,b). \tag{2.6}$$

Finding top k canonical variate pairs is equivalent to solve the following maximization problem:

$$\begin{aligned} W_x^*, W_y^* &= \max_{W_x, W_y} tr(W_x' \Sigma(x,y) W_y), \\ \text{s.t.} \quad & W_x' \Sigma(x,x) W_x = W_y' \Sigma(y,y) W_y = I \\ & w_{xi} \Sigma(x,y) w_{yj} = 0 \quad \text{for } i \neq j \end{aligned} \tag{2.7}$$

Project original random variables to a new subspace: $W_x^* = (w_{x1}^*, w_{x2}^*, \ldots, w_{xk}^*)$, and $W_y^* = (w_{y1}^*, w_{y2}^*, \ldots, w_{yk}^*)$ serve as two mapping matrices which project x and y to a k dimensional subspace and form two k dimensional vectors, i.e, $(a_1^*, a_2^*, ..., a_k^*)$ and $(b_1^*, b_2^*, ..., b_k^*)$. CCA identifies the projection leading to a possible joint structure for the two set of random variables [52]. Since k is usually set to be much smaller than p, $(a_1^*, a_2^*, ..., a_k^*)$ and $(b_1^*, b_2^*, ..., b_k^*)$ are the new representation of the original random vectors which have much lower dimensionality than the original vectors but yet keep most of the joint information of x ad y. Hence, CCA can be used to reduce dimensionality for the data. Figure 2.1 illustrates how CCA projects variables into a low-dimensional subspace. Solid lines represent the first canonical correlation vectors w_{x1}^*, w_{y1}^*, and dashed lines represent the second canonical correlation vectors w_{x2}^*, w_{y2}^*. In this example, the original dimensionality for x and y are 4 and 3 respectively. The dimensionality of the new subspace is set to be 2. The projection matrix $W_x^* = \{w_{x1}^*, w_{x2}^*\}$ projects x to $\{a_1^*, a_2^*\}$ and the matrix $W_y^* = \{w_{y1}^*, w_{y2}^*\}$ projects y to $\{b_1^*, b_2^*\}$, where (a_1^*, b_1^*) is the first canonical variate pair and (a_2^*, b_2^*) is the second canonical variate pair.

Applying canonical correlation analysis to multimodal learning: In multimodal learning, data are collected from multiple modalities for a set of samples. Consider a two-modal problem, i.e., $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times p}$, where n is the sample size, d and p are feature dimensions corresponding to the two modalities. We suppose that $d \geq p$. In real world applications, we usually do not know

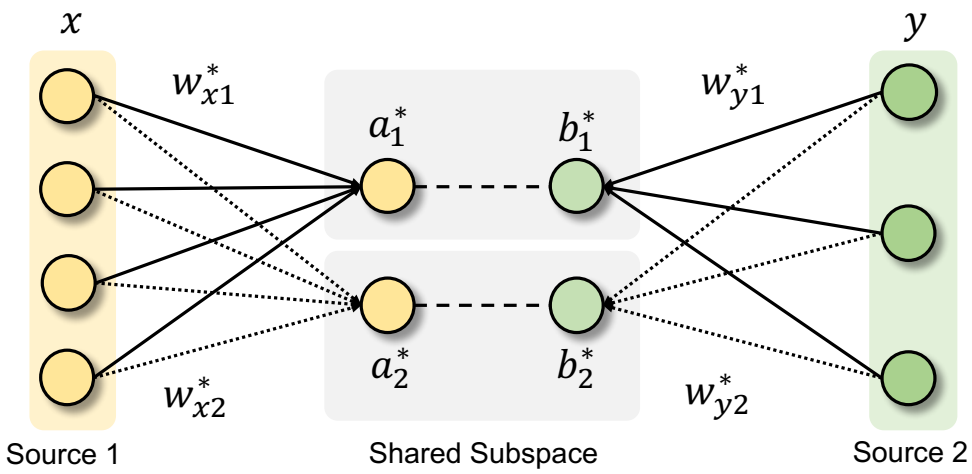


Figure 2.1: **Example of canonical correlation analysis (CCA) involving two data modalities.** The data points in the first data modality are $\mathbb{R}^4$ and those in the second data modality are $\mathbb{R}^3$. Solid lines represent the first canonical correlation vectors $w_{x,1}^*, w_{y,1}^*$, and dashed lines represent the second canonical correlation vectors $w_{x,2}^*, w_{y,2}^*$. Data points from the original data modalities are projected to a new common subspace. In this subspace, the dimensionality of the data is reduced from $\mathbb{R}^4$ or $\mathbb{R}^3$ to $\mathbb{R}^2$.

the distribution of data, i.e., the mean and covariance are unknown. In order to use CCA, we need to use sample mean and sample covariance to estimate the mean and covariance of the distribution.

CCA has been widely used in multimodal learning. For example, when large unlabeled data are available for two modalities and only limited labeled data are available, if the feature dimensionality is very large, learning a good model only by the labeled data is not easy. A possible way to solve this challenging problem is to utilize the unlabeled data to construct a projection by CCA to reduce the feature dimensionality. [39] provides a theoretical guarantee that such dimensionality reduction can reduce the number of labeled sample needed. In clustering area, high-dimensional data clustering is a difficulty problem. By using CCA, the dimensionality of the data can be reduced which makes the clustering problem easier. Moreover, CCA allows information to be transferred between the two modalities. Such transfer can lead to potential improvements on the cluster quality. For example, video and audio data clustering quality can be significantly improved if CCA is applied [27]. When dealing with action data, vector CCA can also be extended to tensor CCA

which can be used to pair-wisely analyze aligned and holistic action volumes [58].

2.2.2 Collective matrix factorization

Matrix factorization has been extensively studied in many domains such as compressive sensing, recommender systems and computer vision [24, 23, 20, 55, 69, 68]. When a matrix is used to describe the relationship between two entities, matrix factorization can be used to learn latent variables/profiles associated with the entities through their interactions (i.e., values in the matrix). For example, in the Netflix problem [61], user profiles and item profiles are learned through identifying the subspace by the user-item interaction matrix.

Classical matrix factorization seeks to approximate a matrix with a low-rank matrix, by explicitly learning the matrix factors. Given a data matrix $X \in \mathbb{R}^{m\times n}$, matrix factorization learns two reduced matrix factors $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$, such that $X \approx UV^T$, and $r < \min(m,n)$ is the upper bound of the rank of the approximated matrix UV^T (the rank of UV^T can be less than r if columns of U or V are linearly dependent). The factors U and V are typically learned via an objective function:

$$\min_{U,V} \; d(X,UV^T), \text{ s.t. } U \in \mathscr{S}_1, V \in \mathscr{S}_2, \tag{2.8}$$

where $d(X,Y)$ is a distance metric function measuring the difference between matrices X and Y, and S_1 and S_2 are two constrains imposed on the factor matrices X and Y.

Typically the distance metric $d(X,Y)$ is chosen to be the Frobenius norm of the difference between X and Y. However, when missing values present in X, $d(X,Y)$ can be defined as the squared ℓ_2 distance between all the observed elements in X and their corresponding elements in Y. As such, we are able to learn matrix factors even with missing values, and the learned matrix

factors can then be used to estimate the missing values under the low-rank assumption. This is the setup for matrix completion [22] and is commonly used in recommender systems [61]. The constraints $\mathscr{S}_1$ and $\mathscr{S}_2$ specify the feasible regions of the matrix factors to induce many desired properties, such as non-negativity $\mathscr{S} = \{U|U_{i,j} \geq 0, \forall i, j\}$ in non-negative matrix factorization [69] and sparsity $\mathscr{S} = \{U|\|U\|_1 \leq z\}$ for interpretable factors [140]. In addition, the complexity control can be implemented using Frobenius constraints $\mathscr{S} = \{U|\|U\|_F^2 \leq z\}$, which are equivalent to the Frobenius norm regularizations [60].

The approximation in (4.1) addresses important semantics in data analysis. When the data matrix X describes the relationship between two types of entities, the factors U and V can be thought of as latent features or latent representations of the entities. For example, in recommender systems we use $X_{i,j}$ to describe the relationship (e.g., rating) between a user i and an item j. The row vector $\mathbf{u}^i \in \mathbb{R}^r$ gives a r-dimensional latent feature representation for the user i and similarly, the row vector $\mathbf{v}^j \in \mathbb{R}^r$ is a latent representation of the item j. The two types of latent profile interact with each other linearly in the latent subspace $\mathbb{R}^r$, i.e., the observed relationship in $X_{i,j}$ can be explained as $\mathbf{u}^i(\mathbf{v}^j)^T$.

In collective matrix factorization, the latent representation/subspace perspective of matrix factorization allows us to link multiple data modalities, when the entities involved in the modalities are overlapped. In multimodal modeling, assume there are t data modalities $X_1 \in \mathbb{R}^{n \times d_1}, \ldots, X_t \in \mathbb{R}^{n \times d_t}$ describing different sets of features of the same set of n samples, where $d_1, d_2, ...d_t$ are the feature dimension for each modality. For example, X_1 is the matrix the images data, X_2 is the matrix of the text descriptions associated with those images and X_3 is the tags matrix for the images. Then, we can apply the matrix factorization procedure to factorize all the datasets and connect the

factorizations by enforcing a shared subject latent representation:

$$\min_{U,\{V_i\}_{i=1}^{t}} \sum_{i=1}^{t} d(X_i, UV_i^T) \text{ s.t. } U \in \mathscr{S}_0, V_i \in \mathscr{S}_i, i = 1,2,...t,$$

where the latent representation U is thus jointly learned from multiple modalities. The U matrix is called modality invariant, as the representation now captures intrinsic properties of the objects. When performing regression and classification on the objects, we can use the latent representation instead of using features from raw data matrices X_i, since the latent representation U contains the common structure and the shared information across all modalities.

Collective matrix factorization has been applied in various multimodal learning problems. For example, it can be used to transfer knowledge from text to image to build more robust text-to-image transfer learning models [128]. It is also used to fuse information between user-tag and user-item [56] to develop more reliable recommender system, when the users' information is limited. In network similarity learning, it is used to combine topological structure, content, and user supervision to build models better than those built on a single modality [25].

2.3 Nonlinear approaches

2.3.1 Kernel canonical correlation analysis

Kernel methods enable nonlinear learning by implicitly mapping the original feature space to a high-dimensional feature space. When applying linear learning methods in the high-dimensional feature space, we are implicitly performing non-linear learning [48]. It is widely used in machine learning and pattern analysis algorithms such as kernel support vector machine [31] and kernel principal components analysis [77]. Similarly, the concept of kernel can be used to enable non-

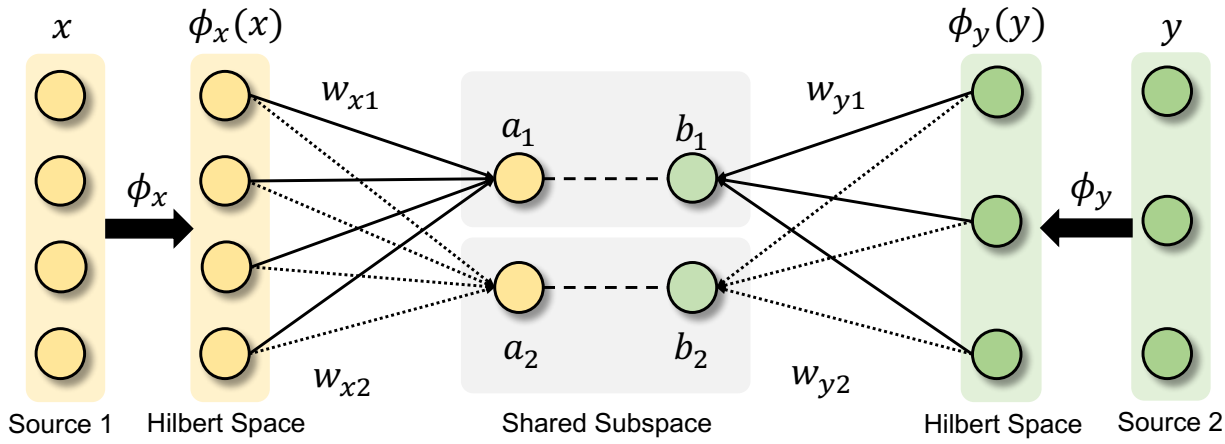


Figure 2.2: **Illustration of kernel canonical correlation analysis (kernel CCA).** The kernel CCA projects data from two modalities to a Hilbert space and identifies a subspace that maximizes canonical correlation of the projected data.

linearity in CCA, called kernel CCA [4]. Assume that we have two data modalities, kernel CCA first projects data from the two modalities to a Hilbert space, i.e., $X \rightarrow \phi_x(X) \in H_x$ and $Y \rightarrow \phi_y(Y) \in H_y$. It then maximizes the correlation between the projected data points $a := w_x^T \phi_x(X)$ and $b := w_y^T \phi_y(Y)$. The concept of kernel CCA is illustrated in Figure 2.2.

Due to the capability to deal with nonlinear correlated data, kernel CCA is widely used in multimodal learning. For example, it can be used for phonetic recognition when articulatory measurements and acoustic features are available [10]. When applying kernel CCA on those two modalities, the non-discriminative information is largely uncorrelated and therefore filtered out. Hence, the learned projections only incorporate the correlated information and can deliver better phonetic classification performance than the original features. Kernel CCA is also used in facial expression recognition problems [136]. Facial images can provide geometric information about the facial expression. Meanwhile, in the learning phase, there are some semantic ratings describing the basic expressions such as happiness, sadness, surprise, anger, disgust and fear. Kernel CCA is used to learn the correlation between the geometric information and the semantic information and project those two feature vectors to a subspace where they have linear dependence. In the new

subspace, it is easier to build linear regression or classification models between the two modalities, than in the original subspace. Hence, given a test image, associated semantic rating can be estimated by it. In social media area, people share the events they attended on social media websites. Identifying unique events from these websites and grouping information for the same events is a cumbersome task due to the high dimensionality of the data collected from social media and the nonlinear dependence between different modalities. Kernel CCA can effectively learn a semantic representation of potentially correlated feature sets. It can be used to learn a joint representation from images and texts/tags/user names. The new features can be concatenated as a new feature vector for clustering social events [3]. This method delivers better performance than those only use data from one modality.

2.3.2 Deep canonical correlation analysis

Even though kernel CCA can be used to learn nonlinear representations, this method is not easy to scale when the size of training data is large. Moreover, the representations learned by kernel CCA is dependent on the kernels used. If the kernels are not suitable for the data, this method may fail. Recently, deep neural network has shown its strong ability to learning nonlinear representations [104, 28, 14, 90]. Therefore, deep CCA is proposed [9, 120] to learn flexible and data-driven nonlinear representations from two modalities. Given two data modalities, deep CCA learns two deep nonlinear mappings which map the two modalities to new representations such that the canonical

correlation of the new representations is maximized [119] [1]:

$$\begin{aligned} \min_{\theta_f,\theta_g,W_x,W_y} & -\frac{1}{N} tr(W_x^T f(X) g(Y)^T W_y), \\ s.t.\ \ & W_x^T (\frac{1}{N} f(X) f(X)^T + r_x I) W_x = I, \\ & W_y^T (\frac{1}{N} g(Y) g(Y)^T + r_y I) W_y = I, \\ & w_{xi}^T f(X) g(Y)^T w_{yj} = 0 \quad \text{for } i \neq j, \end{aligned} \tag{2.9}$$

where X and Y are input data of two modalities. f and g denote two full-connected deep neural networks which produce nonlinear mappings. θ_f, θ_g are parameters of the two networks. N is the sample size. W_x and W_y are canonical correlation vectors defined in Section 2.2.1. We use regularized covariance instead of original covariance to prevent overfitting and r_x, r_y are regularization parameters (we assume the data are centered). w_{xi} is the i-th column of W_x and w_{yj} is the j-th column of W_y. Figure 2.3 is the overview of deep CCA. In CCA, the mappings are W_x^T and W_y^T for two modalities, which produce linear projections. It may be difficulty to accurately reconstruct one modality from the other due to the possible non-linear interaction between the two modalities [119]. In deep CCA, the final mapping functions for the two modalities are $W_x^T f(\cdot)$ and $W_y^T g(\cdot)$. They learn the possible nonlinear interaction and project the two modalities to a subspace in which they are easily to reconstruct the other one.

One alternative view of deep CCA is that it learns two kernels from data for kernel CCA. Sometimes we do not know what kind of kernels are best suitable for the data. Hence, the kernel we choose may not provide an appropriate nonlinear mapping for the data. In this case, deep neural network is a better choice than a prescribed kernel, as the 'non-linear transformation' is learned

[1] We use biased covariance to make it consistent with the original formulation proposed in [119]. Since N is a constant, it doesn't affect the optimal soulutions of model paramters if we use biased covariance or unbiased covariance.

from the data. This is empirically demonstrated by the experiments on articulatory speech data and MINST data.

We note that deep CCA can also be combined with other deep learning techniques. For example, it can be combined with autoencoder [119]. In addition to deep CCA, this model also contains two autoencoders to reconstruct the learned views. It optimizes an objective that maximizes the canonical correlation between the projected representations and minimizes the reconstruction error of the autoencoders simultaneously [2]:

$$
\begin{aligned}
\min_{\theta_x,\theta_y,W_x,W_y} & -\frac{1}{N} tr\left(W_x^T f(X) g(Y)^T W_y\right) \\
& + \frac{\lambda}{N}\sum_{i=1}^{N}\left(\|x_i - p(f(x_i)))\|^2 + \|y_i - q(g(y_i))^2\|\right), \\
s.t.\ & W_x^T\left(\frac{1}{N} f(X) f(X)^T + r_x I\right) W_x = I, \\
& W_y^T\left(\frac{1}{N} g(Y) g(Y)^T + r_y I\right) W_y = I, \\
& w_{xi}^T f(X) g(Y)^T w_{yj} = 0 \quad \text{for } i \neq j,
\end{aligned}
\tag{2.10}
$$

where x_i is the i-th sample from the first modality. y_i is the i-th sample from the second modality. $\lambda > 0$ is a trade-off parameter to control the reconstruction error. Other notations are the same with deep CCA in Eq. (2.9). Compared with deep CCA's formulation in Eq. (2.9), this formulation considers the reconstruction error of two autoencoders in the form of regularizations, in which each autoencoder maximizes the lower bound of the mutual information between the inputs and learned features [115]. Meanwhile, CCA can be viewed as maximizing the mutual information between the canonical variate pairs, i.e., the projected features of the two modalities [17]. Hence, this method offers a trade-off between the information captured in the input-feature mapping within

[2] note1

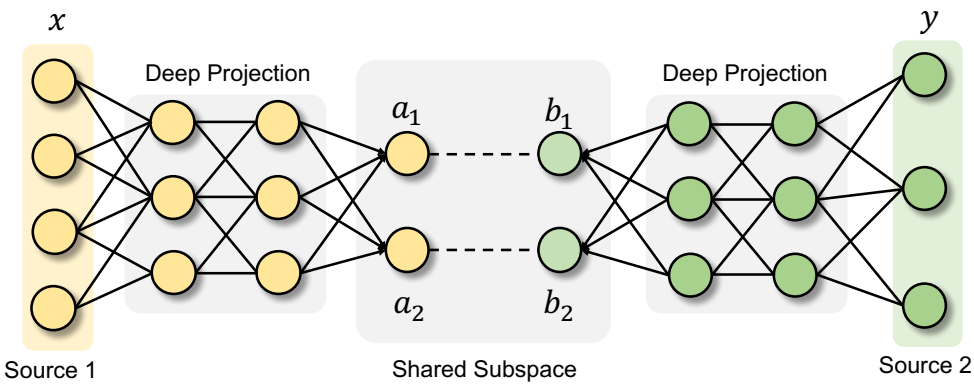


Figure 2.3: **Illustration of deep canonical correlation analysis structure [9].** It learns two deep non-linear mappings which map two modalities to new representations such that the canonical correlation of new feature vectors is maximized.

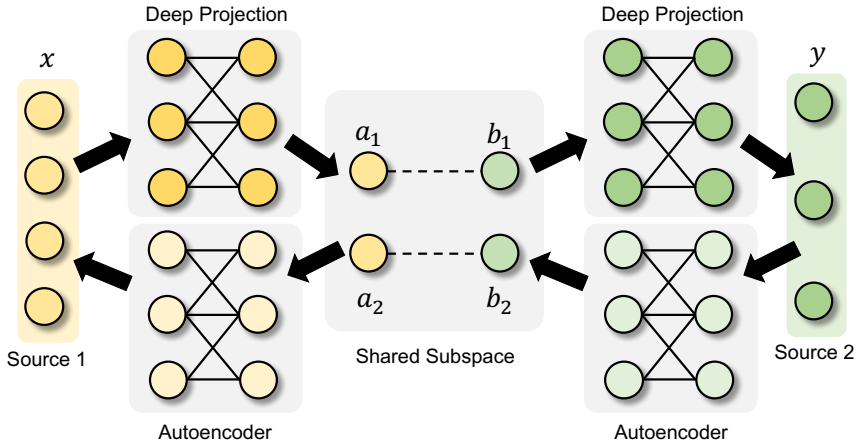


Figure 2.4: **Illustration of deep canonical correlated autoencoders [119].** It simultaneously maximizes the canonical correlation between the projected representations and minimizes reconstruction error of the autoencoders.

each modality on one hand, and the information in the feature-feature relationship across modalities on the other hand [119]. The framework is illustrated in Figure 2.4.

2.3.3 Multimodal deep Boltzmann machine

In addition to the discriminative models introduced above, generative approaches are also widely used in multimodal learning. Generative approaches model the joint probability of multiple modalities.

One example is multimodal deep Boltzmann machine (DBM) [103]. This method is based on restricted Boltzmann machine (RBM). We first briefly review some basic concepts of RBM. RBM is a network of symmetrically coupled binary random variables or units. RBM contains two layers of units. The first layer contains visible units (input) $x \in \{0,1\}^m$, and the second layer contains hidden units $h \in \{0,1\}^n$, where n is the number of the hidden units, and m is the number of the visible units. The hidden units and the visible units are connected. No visible-to-visible or hidden-to-hidden interaction is allowed. Figure 2.5 (a) is an illustration of RBM. W is the interaction between the hidden units and the visible units. For all Boltzmann machines, the joint probability distribution between units is calculated by energy function E as known from statical physics:

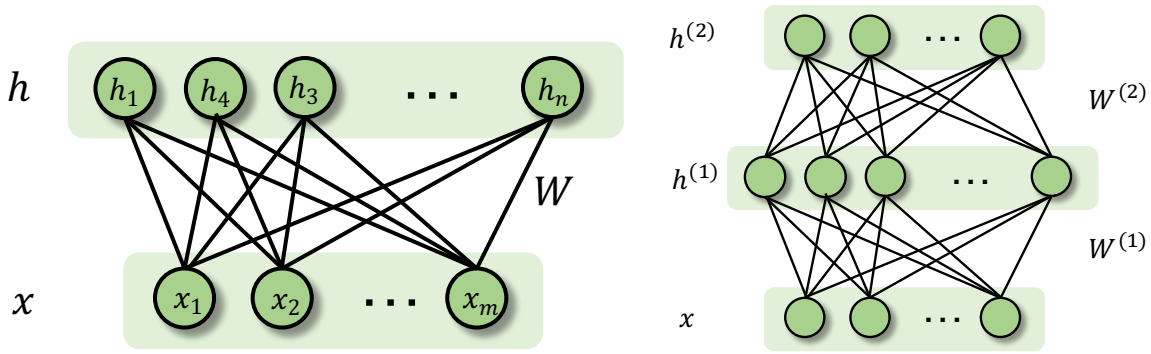
$$p = \frac{1}{Z}\exp(-E),$$

where Z is a normalization factor to make sure the integral over p is 1. For the RBM illustrated in Figure 2.5 (a), the energy function is $E(x,h|W) = -x^T W h$, if we ignore self-energy for the two layers and only consider the interaction energy between the two layers. Hence, the joint distribution of the visible units and the hidden units is:

$$p(x,h|W) = \frac{1}{Z(W)}\exp(x^T W h),$$

where $Z(W)$ is the normalization factor parameterized by the network parameter W.

When dealing with challenging applications, we may need abstract internal representation. In these cases, the two-layer structure of RBMs may not be able to produce a satisfactory performance. This limitation can be overcome by DBM. Similar to RBM, DBM is a network of symmetrically coupled stochastic binary units [103]. It contains visible units (input), and several layers of



(a) (b)

Figure 2.5: **Example of restricted Boltzmann machine and deep Boltzmann machine.** (a) An example of restricted Boltzmann machine (RBM), where h is the hidden layer. x is the visible layer. (b) An example of deep Boltzmann machine (DBM). It contains two hidden layers and one visible layer.

hidden units $h^{(i)} \in \{0,1\}^{F_i}$, where i represents the i-th hidden layer and F_i is the units number of the i-th hidden layer. Figure 2.5 (b) illustrates an example of DBM with two hidden layers, where $W^{(1)}$ and $W^{(2)}$ are the weight matrix to connect consecutive layers which measure the interactions between layers. The energy function is $E(x,h^{(1)},h^{(2)}|W^{(1)},W^{(2)}) = -x^T W^{(1)} h^{(1)} - (h^{(1)})^T W^{(2)} h^{(2)}$, and hence, the joint distribution of the input units and the two hidden units is given by:

$$p(x,h^{(1)},h^{(2)}|W^{(1)},W^{(2)}) = \frac{1}{Z(W^{(1)},W^{(2)})} \exp(x^T W^{(1)} h^{(1)} + (h^{(1)})^T W^{(2)} h^{(2)}).$$

In multimodal learning, multiple modalities may have distinct statistic properties. For example, text features are discrete and image features are continuous. Since DBM can extract abstract representations, in most cases, it is more suitable for multimodal learning than RBM. Figure 2.6 is an example of using three-hidden-layer DBM to learn the joint representations from two modalities [103]. The two modalities can either be texts, images, tags, videos, etc. As an example, we use texts and images as the two modalities. In Figure 2.6, $v^m \in \mathbb{R}^D$ and $v^t \in \mathbb{N}^K$ denote image input and text input, respectively. $h^{(im)}$, $h^{(it)}$ with $i = 1,2$, and $h^{(3)}$ are the hidden layers. Each modality

has two specific hidden layers. $h^{(3)}$ is the learned joint representation. Since image features are real-valued, the visible-hidden interaction energy should use the form of Gaussian RBM. The energy between the image visible layer and the two specific hidden layers of the image part is given by ([103]):

$$E(v^m, h^{(1m)}, h^{(2m)}|\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F_1}\frac{v_i^m}{\sigma_i}W_{ij}^{(1m)}h_j^{(1m)} + \sum_{i=1}^{D}\frac{(v_i^m - b_i)^2}{2\sigma_i^2} - \sum_{j=1}^{F_1}\sum_{l=1}^{F_2}h_j^{(1m)}W_{jl}^{(2m)}h_l^{(2m)},$$

where $\theta = \{W^{(1m)}, W^{(2m)}, b, \sigma\}$ are model parameters. $W^{(1m)}$ is the weight between the input layer and the first hidden layer. $W^{(2m)}$ is the weight between the first hidden layer and the second hidden layer. b is the bias of the input layer. σ is the standard deviation of the Gaussian distribution. It can be the same for all the visible units or independent for each visible unit if the data is not whitened [63]. In this energy function, the first two terms are the interaction between the input (visible) units and the first hidden layer. The third term is the energy between the first hidden layer and the second hidden layer. The joint distribution of those layers can be calculated by this energy function. The joint distribution of text is similar to that of the image component, except that the energy between the input units and the first hidden layer needs to be changed to a Replicated Softmax model [47] to deal with the text input. This model can be easily extended to other data modalities by modifying the energy of the input layer according to the data property of the input. The energy between the second hidden layer of each modalities with the third hidden layer (the joint representation layer) is:

$$E(h^{(3)}, h^{(2t)}, h^{(2m)}|\theta) = -(h^{(2m)})^T W^{(3m)}h^{(3)} - (h^{(2t)})^T W^{(3t)}h^{(3)}. \tag{2.11}$$

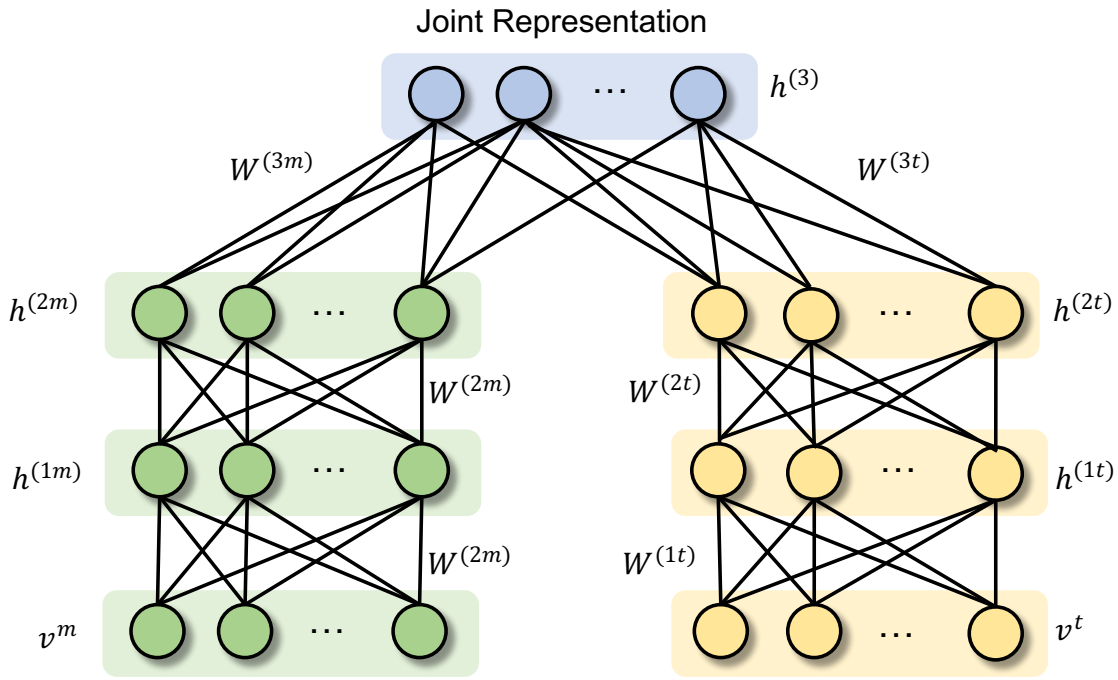


Figure 2.6: **The illustration of a multimodal Deep Boltzmann machine [103].** It models the joint distribution of data from two modalities, and thus provides a joint representation.

The joint distribution of those units is $p(h^{(3)}, h^{(2t)}, h^{(2m)}|\theta) = \frac{1}{Z(\theta)} \exp(-E(h^{(3)}, h^{(2t)}, h^{(2m)}|\theta))$. Given these distributions, we can compute the joint distribution of the inputs from multiple modalities: [103]:

$$p(v^m, v^t|\theta) = \sum_{\mathbf{h}} p(h^{(2m)}, h^{(2t)}, h^{(3)}|\theta)(\sum_{h^{(1t)}} p(v^t, h^{(1t)}, h^{(2t)}|\theta))$$
$$(\sum_{h^{(1m)}} p(v^m, h^{(1m)}, h^{(2m)}|\theta)),$$

where $\mathbf{h} = \{h^{(1m)}, h^{(2m)}, h^{(1t)}, h^{(2t)}, h^{(3)}\}$. For generative models, the model parameters can be learnt by maximizing the likelihood. In this model, exact maximum the likelihood is intractable, but they can still be learnt by variational approach approximately [103].

Figure 2.7 (a) shows a multimodal RBM, and Figure 2.7 (b) presents a different view of this model. The difference between those two models is that the deep model has many layers to trans-

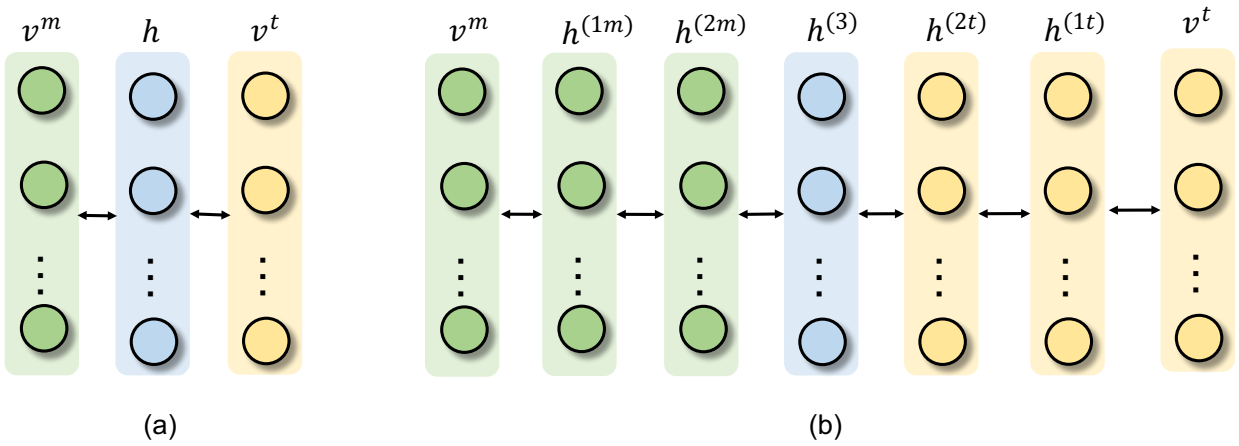


Figure 2.7: **Different multimodal Boltzmann machines[103].** (a) contains only one hidden layer. (b) contains multiple hidden layers. The task to remove modal-specific component is distributed in different layers in the deep model. It can be easier to extract joint representations for (b) than (a).

form features. In some cases, the statistic properties of different modalities are rather different. For example, in the previous example, text features are discrete and image features are continuous. Directly learning a joint representation from different modalities through a restricted Boltzmann machine may not be feasible then. It needs extra bridges between the joint representation and the inputs of each modality. In DBM, each layer successively transforms the representation into a slightly more abstract level and removes part of modal-specific correlations [103]. Hence, the middle layer can be viewed as a modal-free representation, while the inputs are modal-full representations. Compared with a simple multimodal RBM, the task to remove modal-specific components is distributed in different layers in the deep model. Therefore, it is much easier to extract joint representations for the deep model than the shallow model.

Chapter 3

Discriminative Fusion of Multiple Brain Networks

In neuroimaging research, brain networks derived from different tractography methods may lead to different results and perform differently when used in classification tasks. As there is no ground truth to determine which brain network models are most accurate or most sensitive to group differences, we developed a new sparse learning method that combines information from multiple network models. We used it to learn a convex combination of brain connectivity matrices from 9 different tractography methods, to optimally distinguish people with early mild cognitive impairment from healthy control subjects, based on the structural connectivity patterns. Our fused networks outperformed the best single network model, Probtrackx (0.89 versus 0.77 cross- validated AUC), suggesting its potential for numerous connectivity analysis.

3.1 Methodology

3.1.1 Preliminary

Since this work is based on sparse logistic regression, we give a brief introduction to sparse logistic regression here.

In linear models, the sparsity means a feature variable is determined to be irrelevant if the cor-

responding weight is zero. Therefore, some irrelevant feature variables are discarded in the model and have no contribution to the final classification model. Sparse learning algorithms such as sparse logistic regression for classification are powerful tools to build models from high dimensional data with low computational cost. The sparsity is achieved by adding sparsity-inducing regularization terms on the weight vector w such as $\lambda\|w\|_1$ to the objective function, and the final weight, or the model, is sparse with high probability. Let $x_i \in \mathbb{R}^d$ denotes one subject where d is the number of feature variables we used, which will be elaborated later. The binary class label of this subject is denoted by $y_i \in \{-1, 1\}$, where a MCI subject is denoted as -1 and a NC subject is denoted as +1. Given n samples $\{\{x_1, y_1\}, \{x_2, y_2\}, ..., \{x_n, y_n\}\}$, the loss function for the sparse logistic regression is:

$$l = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i(w^T x_i + c))) + \lambda\|w\|_1 \tag{3.1}$$

where c is the intercept, and is a tunable regularization parameter that is greater than or equal to 0. Here we use l_1 norm to regularize the weight vector - this will yield sparsity in the weight vector. When λ equals 0, there is no sparsity in weight vector. As λ increases, more entries in weight vector turn to 0. When λ is large enough, all the entries in weight vector become 0. By minimizing the loss function, we obtain the optimal weight vector $\hat{w}$ and intercept $\hat{c}$. For a new subject $\tilde{x}$, the probability that this subject belongs to class $\hat{y}$ is:

$$P(\tilde{y}|\tilde{x}) = \frac{1}{1 + \exp(-\tilde{y}(\hat{w}^T \tilde{x} + \hat{c}))} \tag{3.2}$$

If the probability of this subject belonging to the NC group is greater than 0.5, this subject will be labeled as NC. Otherwise this subject will be labeled as MCI.

3.1.2 Overview

Fig. 3.1 summarizes the overview of our fusion approach to build "consensus networks" based on fusing networks from multiple tract tracing methods. From diffusion MRI scans of multiple subjects, we extract different brain networks with whole brain tractography. Though our proposed fusion approach is not limited to structural networks computed from dMRI tractography, here we use the nine tractography methods studied in our previous work [134], which include methods that are classified as tensor-based deterministic, orientation distribution function (ODF)-based deterministic, and probabilistic approaches. Each network reconstruction method describes brain connectivity from a different perspective, and none is universally better than all others for diagnostic classifications tasks. Therefore when it comes to building models from diffusion MRI images, it is intuitive to fuse different brain networks and leverage the predictive information from all the networks. However, the key question is how to fuse the different networks and build effective predictive models from the fused models. As far as we know, there is no principled approach proposed to combine networks for use in predictive models. As shown in the experimental section, simple numerical averaging of nodal edge weights may not be able to boost the predictive performance. Instead, we propose to learn how to fuse the networks from data, such that the combination gives the optimal predictive performance. First, we study fused networks computed as a convex combination of different brain networks. We describe a new machine learning model to simultaneously learn the coefficients of the convex combination as well as the classifier parameters. As a result, the combination coefficients are learned to maximize the predictive performance of the classifier and meanwhile the classifier is learned specifically to use the combined network.

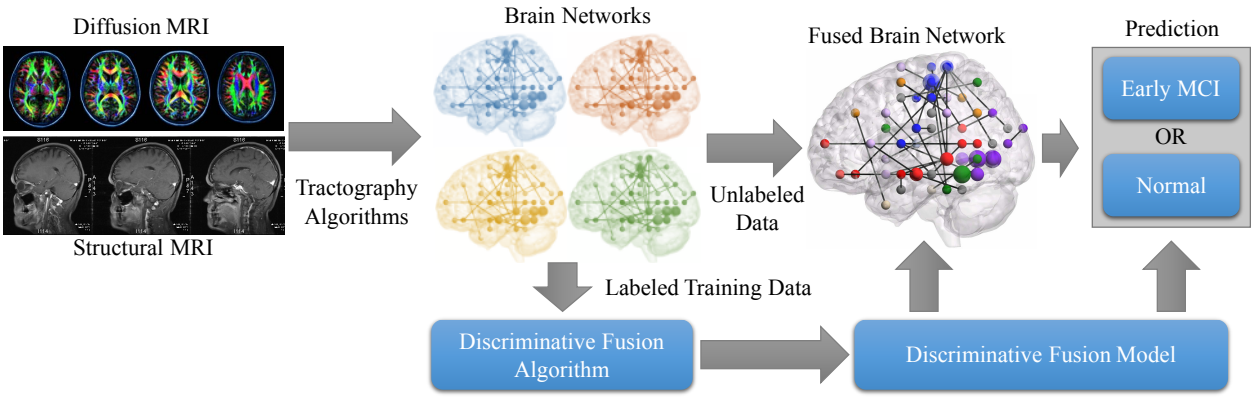


Figure 3.1: **Overview of our network fusion framework.** Multiple types of brain networks are computed by applying different tractography methods to the participants' diffusion MRI data [134]. Different brain networks are combined using a sparse learning method and the optimal convex combination is used for classification. The combination coefficients and the classifiers are simultaneously learned from the training data and cross-validated.

3.1.3 Discriminative Fusion:

Our proposed discriminative fusion (DFUSE) is a data-driven model that includes a training stage and a prediction stage. In the training stage the DFUSE algorithm learns the optimal combination coefficients and a logistic regression classifier from a set of patients with known medical classification. In the prediction stage, the brain networks from a patient are combined according to the coefficients. The combined network is then used by the classifier to give a prediction for the medical classification problem.

Formulation. Given a set of diffusion MRI scans from N patients, we apply different tractography methods to obtain M brain networks for each participant. Let $\mathbf{x}_i^{(m)}$ denotes a vector representation of the m-th brain network for patient i ($i \in [1,N], m \in [1,M]$), in which each element is a numerical representation of a connection property (e.g., density or integrity) between two brain regions. We would like to combine all networks for each participant into a single network using a convex combination, i.e., the combined network $x_i(\tau) = \sum_{m=1}^{M} \tau_m x_i^{(m)}$, where $\tau = [\tau_1 \dots \tau_M]$ is the vector of combination coefficients, and the convex combination gives $\sum_{m=1}^{M} \tau_m = 1; \tau_m \geq 0, \forall \tau_m$.

Convex combination is one type of linear combination that gives a clear interpretation on how much each original network contributes to the fused network. For the N subjects used for training, we also have diagnostic label information stored in $y = [y_1, \ldots, y_N]$, where $y_i = 1$ if the patient is case and -1 if control.

To learn the combination of the networks, we propose a machine learning formulation that jointly learns the classifier parameters and the combination coefficient, which solves the following optimization problem:

$$\begin{aligned} &\min_{w,c,\tau} \Sigma_{i=1}^{N} \ell(w, c, \tau; x_i, y_i) + \lambda \|w\|_1, \\ &\text{s.t.} \quad \Sigma_{m=1}^{M} \tau_m = 1; \tau_m \geq 0, \forall \tau_m \end{aligned} \tag{3.3}$$

where w and c are classifier parameters, the constraints on τ ensures a convex combination, the logistic loss is:

$$\ell(w, c, \tau; x_i, y_i) = \log\left(1 + exp\left(-y_i(x_i(\tau)^T w + c)\right)\right).$$

The ℓ_1-norm induces sparsity in the parameters $\mathbf{w}$ [72, 141, 139, 138], such that the classifier learns a subset of predictive connections and only uses these connections in the classifier. The sparsity parameter λ controls the sparsity of the model. A smaller λ allows more connections to be involved in the model. The optimization problem in (3.3) can be solved by proximal block coordinate descent [12, 112, 125]. Once the optimization process has converged, we obtain the optimal combination coefficients τ^* and classifier parameters w^* and c^*.

3.1.4 Optimization

The objective function in Eq. (3.3) is a convex function. So, it has global solution. Since there are non-differentiable terms, we use proximal gradient descent to optimize it. We first compute the gradient with respect to all parameters. We denote

$$L = \Sigma_{i=1}^{N} \ell(w, c, \tau; x_i, y_i) + \lambda \|w\|_1 \tag{3.4}$$

Denote X to be the tensor that is formed by stack all $x^(j)$ with $j = 1, ...m$. Then, the shape of X is $n \times d \times m$. The gradient of L with respect to w is

$$\frac{\partial L}{\partial w} = \frac{1}{N} x(\tau)^T (y \cdot (\sigma(-y \cdot (x(\tau)w + c))) \tag{3.5}$$

where $\cdot$ denote dot product. σ is the sigmoid function.

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{3.6}$$

The gradient with respect to τ is

$$\frac{\partial L}{\partial \tau} = \frac{y}{N} \cdot (\sigma(y \cdot (x(\tau)w + c) X w \tag{3.7}$$

The gradient with respect to c is

$$\frac{\partial L}{\partial c} = \frac{y}{N} \sigma(y \cdot (x(\tau)w + c) \cdot y \tag{3.8}$$

Proximal Gradient Descent: Proximal gradient descent [107] is widely used to optimize the objective function with both differentiable and non-differentiable terms. We first start from the general form of proximal gradient descent and then apply it to our problem.

Given objective function

$$f(x) = g(x) + h(x) \tag{3.9}$$

where $g(x)$ is a convex differentiable function and $h(x)$ is a convex non-differentiable function. If we only consider the differentiable part for $f(x)$, i.e. $f(x) = g(x)$, we can use gradient descent to optimize it, i.e.

$$x^{k+1} = x^k - t\nabla f(x) \tag{3.10}$$

It is equivalent to optimize solve the following optimization problem.

$$x^{k+1} = \arg\min_z f(x^k) + \nabla f(x^k)^T (z - x^k) + \frac{1}{2t}\|z - x^k\|^2 \tag{3.11}$$

However, $h(x)$ is not differentiable. The strategy is to leave $h(x)$ unchanged. So the update for x^{k+1} is

$$x^{k+1} = \arg\min_z f(x^k) + \nabla f(x^k)^T (z - x^k) + \frac{1}{2t}\|z - x^k\|^2 + h(z) \tag{3.12}$$

Eq. (3.12) means that when we update z/x, we minimize the non-differentiable term $h(z)$. So, in each step, we update the smooth term using the gradient descent to make sure it is moving to the direction that makes the function value smaller and meanwhile the $h(z)$ is also minimized and is

to-warding to our goal, i.e., minimize the objective function.

Eq. (3.12) can be written as

$$x^{k+1} = \arg\min_z \frac{1}{2t}\|z - (x^k - t\nabla g(x^k))\|^2 + h(z) \tag{3.13}$$

We define the proximal mapping as follows.

$$Prox(x^{k+1}) = \arg\min_z \frac{1}{2t}\|x - z\|^2 + h(z) \tag{3.14}$$

Then, Eq. (3.13) can be written as

$$x^{k+1} = Prox(x^k - t_k\nabla g(x^k)) \tag{3.15}$$

In our proposed form, we have two non-differentiable terms.

$$h_1(w) = \|w\|_1 \tag{3.16}$$

$$h_2(w) = simplex(\tau) \tag{3.17}$$

where we use $simplex(x)$ denote the constraint $\sum x_i = 1, x_i \geq 0$.

Projections: Next, we will show how to optimize Eq. (3.14) with these two non-differentiable terms. We first start with a general simplex projection.

$$\min_x \frac{1}{2}\|x - y\| \tag{3.18}$$

$$s.t. x^T\mathbf{1} = 1 \tag{3.19}$$

$$x \geq 0 \tag{3.20}$$

The Lagrangian of the problem in Eq. (3.18) is

$$L(x,\lambda,\beta) = \frac{1}{2}\|x-y\|^2 - \lambda(x^T\mathbf{1}-1) - \beta^T x \tag{3.21}$$

where λ and β are Lagrangian multipliers. At the optimal point, we have the KKT condition

$$x_i - y_i - \lambda - \beta_i = 0 \tag{3.22}$$

$$x_i \geq 0 \tag{3.23}$$

$$\beta_i \geq 0 \tag{3.24}$$

$$x_i\beta_i = 0 \tag{3.25}$$

$$\sum_i x_i = 1 \tag{3.26}$$

From Eq. (3.25) and Eq. (3.23) we have (1) if $x \geq 0$, $\beta_i = 0$ and $y_i + \lambda_i \geq 0$, AND (2) if $x = 0$, $\beta_i \geq 0$ and $y_i + \lambda = \beta_i$. Thus, we can sort the x and y in the descent order.

$$y_1 \geq y_2 \geq, ..., \geq y_\rho \geq y_{\rho+1} \geq, ..., \geq y_d \tag{3.27}$$

$$x_1 \geq x_2 \geq, ..., \geq x_\rho = x_{\rho+1} =, ..., = x_d \tag{3.28}$$

where we have when $i > \rho$, all $x_1 = 0$. From Eq. (3.26) we have

$$\lambda = \frac{1}{\rho}(1 - \sum_i^\rho y_i) \tag{3.29}$$

WIth Shealev-Shwartz and Singer Theorem, we have the solution for ρ is

$$\rho = \{j, \max\{1 \leq j \leq d : y_j + \frac{1}{j}(1 - \sum_{i}^{j} y_j) > 0\} \tag{3.30}$$

Next, we show how to project to l_1 ball. Suppose we have the projection

$$\min \frac{1}{2}\|x-y\|^2 + \lambda \|x\|_1 \tag{3.31}$$

Since $\|x\|^2 = \sum_i x_i^2$ and $\|x\|_1 = \sum_i |x_i|$, we optimize each dimension separately for Eq. (3.31), i.e.,

$$\frac{1}{2}(x_1 - y_1)^2 + \lambda |x_1| \quad \text{with} i = 1, ...d \tag{3.32}$$

To solve Eq. (3.32), we use the subgradient method. The subdifferential of $|x|$ is $sign(x)$ and $\frac{d(x-y^2}{dx} = 2(x-y)$. Thus, we have $0 \in \partial f(x^*)$ where x^* denote the optimal solution. Therefore, we have

$$x^* = sign(x) \max(|x| - \lambda, 0) \tag{3.33}$$

3.2 Experiments

3.2.1 Dataset

The imaging datasets analyzed for in this study were collected from 16 sites across the United States and Canada in the second stage of the Northern American Alzheimer's Disease Neuroimaging Initiative (ADNI2). In total, 124 subjects' diffusion MRI and structural MRI data were analyzed. Detailed subject inclusion, exclusion criteria and scanning protocols can be found in the

ADNI2 website. These 124 subjects include 51 normal elderly controls (NCs), 73 individuals diagnosed with early mild cognitive impairment (eMCI).

3.2.2 Brain Networks

For each subject, we computed 9 brain networks using nine methods, including 4 tensor-based deterministic algorithms: FACT (T-FACT) [78], the second-order Runge–Kutta (T-RK2) [11], the tensorline (T-TL) [66], and interpolated streamline (T-SL) methods [29], two deterministic tractography algorithms based on fourth order spherical harmonic derived ODFs – FACT (O-FACT) and RK2 (O-RK2), and three probabilistic approaches: "ball-and-stick model based probabilistic tracking" Probtrackx (Probt) [13], the Hough voting method [2] and the probabilistic index of connectivity (PICo) method [84]. Each brain network describes detected connections between 113 cortical and subcortical regions-of-interest (ROIs), which are defined by using the Harvard Oxford Cortical and Subcortical Probabilistic Atlas [33]. Therefore we can use a vector of dimension 6328 ($113 \times 112/2$) to represent all connections of distinct ROIs pairs in each network. Please see [134] for details of computing these nine brain networks.

3.2.3 Experiment Settings

In the first experiment we compared the predictive performance of individual networks, in terms of area under the ROC curve (AUC), sensitivity and specificity. These are standard metrics measuring algorithm performance in classification problems. We also provide two intuitive fusion methods for baseline comparisons. The first method concatenates vectors from all networks (B-CON), resulting in a feature vector of dimension 56952. The second method combines the networks by averaging of all of the individual networks; this can be considered as a special case of the general

linear combination ($\tau_i = 1/9, \forall i$). For all the patients, we used 10-fold cross validation, i.e., each time we use the brain networks from 90% patients to train a classifier, and the 10% to test the classifier and compute performance metrics. For all individual brain networks as well as the two baseline methods, we use sparse logistic regression to train classifiers. For the proposed DFUSE, the classifier is trained using algorithms in Section 3.1. As the sample size is too small to generate extra validation data for model selection (the selection of hyper parameter λ in the sparse logistic regression), we report the best performance for all methods.

3.2.4 Results

Averaged classification results over 10 iterations are given in Table 3.1. Our proposed DFUSE algorithm significantly outperformed all other competing methods (p-value < 0.001). DFUSE has an average AUC of 0.89, compared to 0.77 achieved by the best individual method, which used only the Probtrackx (Probt) networks. DFUSE also had the highest average sensitivity of 0.84 and specificity of 0.77, compared to the second highest sensitivity of 0.72 achieved by tensor-based FACT (T-FACT) and 0.69 by the Probtrackx networks. No individual brain network generation method had a predictive power that was even close to the one from the fused brain network. This significant improvement in predictive performance supports our hypothesis about the benefits of fusion for brain networks.

Two other baseline network combination methods also did not perform well: the predictive performance of the feature concatenation (B-CON) does not even perform as well as the best individual brain network. This may be because, for the B-CON method, there are too many features presented to the classifier (over 56k), relative to the number of subjects (samples) available to train it. Only $\sim$110 samples are available here to train the classifier at every iteration (90% of the total of 124 subjects). On the other hand, the AUC of the simple average brain network (B-AVG)

	AUC	Sensitivity	Specificity
DFUSE	$\mathbf{0.89 \pm 0.09}$	$\mathbf{0.84 \pm 0.16}$	$\mathbf{0.77 \pm 0.07}$
B-CON	0.58 ± 0.10	0.56 ± 0.21	0.50 ± 0.07
B-AVG	0.55 ± 0.15	0.58 ± 0.20	0.49 ± 0.08
B-ENS	0.79 ± 0.11	0.71 ± 0.25	0.72 ± 0.09
T-FACT	0.59 ± 0.11	0.72 ± 0.25	0.44 ± 0.14
T-RK2	0.58 ± 0.11	0.56 ± 0.25	0.49 ± 0.10
T-SL	0.62 ± 0.14	0.48 ± 0.27	0.64 ± 0.26
T-TL	0.58 ± 0.14	0.60 ± 0.21	0.48 ± 0.07
O-FACT	0.62 ± 0.09	0.60 ± 0.19	0.51 ± 0.09
O-RK2	0.60 ± 0.13	0.60 ± 0.21	0.53 ± 0.07
PICo	0.58 ± 0.10	0.56 ± 0.21	0.50 ± 0.07
Hough	0.66 ± 0.11	0.64 ± 0.23	0.54 ± 0.11
Probt	0.77 ± 0.08	0.70 ± 0.22	0.69 ± 0.08

Table 3.1: **Quantitative comparison of classifiers using different brain networks to predict the early MCI.** We compare the performance of each individual brain networks from tractography, simple network combination, and our network fusion method (DFUSE). The average and variance of area under the ROC curve (AUC), sensitivity and specificity over 10 splittings are reported. The proposed DFUSE significantly outperforms all other methods on this problem (p-value < 0.001).

network	τ	network	τ	network	τ
T-FACT	0.025	T-Rk2	0.014	T-SL	0.023
PICo	0.058	Hough	0.010	Probt	0.871
T-TL	0	O-FACT	0	O-RK2	0

Table 3.2: **Combination coefficients τ of 9 networks.**

is 0.55, which is even poorer than the worst performing brain network T-TL, at 0.58. Arbitrary combinations of brain networks may not help for the task of distinguishing early MCI from NCs. Task specific fusion as proposed in this paper may be more beneficial.

3.3 Discussion

One attractive property of the proposed DFUSE approach is that we can obtain an interpretable combination coefficient τ, indicating how much each of the individual brain networks contributes to the final combined network. The average combination coefficients for all networks are given

in Table 3.2. We see that in the combination, Probtrackx has the heaviest weight of 0.871 (all elements of τ range from 0 to 1), averaged over 10 iterations. This is consistent with the finding that Probtrackx is also the best predictive individual network as shown in Table 3.1. On the other hand, the weights of T-TL, O-FACT, O-RK2 are consistently zeros, i.e., they do not contribute to the combined network. As such, the combination offers a guide to which tractography methods to run (clearly not all methods need to be run for problems where they are given zero weight). Moreover, the networks with zero weights are not the same as the least white individual networks (T-RK2, PICo, T-FACT). The inconsistency shows that networks with weak predictive power may still have valuable connection information to complement other better performed networks. It is possible to leverage clustering analysis [137] and explore different sub-modalities within the networks, and we will leave this interesting analysis in our future work.

Because of the sparsity introduced on the model $\mathbf{w}$, we are also able to inspect what are the important connections contributing to the final classifiers. By averaging the non-zero weights for each connection from different experiments, we can generate a ranked list of connections, many of which are previously known to be relevant to the progression of Alzheimer's. Here are a few connections that appear in the top of the list: Right Temporal Pole $\Leftrightarrow$ Right Precentral Gyrus, Left Pallidum $\Leftrightarrow$ Left Caudate, Left Lingual Gyrus $\Leftrightarrow$ Left Thalamus, Left Cingulate Gyrus Anterior Division $\Leftrightarrow$ Left Frontal Medial Cortex, Right Planum Polare $\Leftrightarrow$ Right Hippocampus.

3.4 Summary

In this work, we developed a new method for discriminative fusion of multiple brain networks to detect early mild cognitive impairment (MCI). We simultaneously learned a convex combination of different brain networks to best detect early MCI, and a classifier that works with the combined

brain network. As the networks are fused in a way that maximizes the discriminative power between normal controls and early MCI subjects, the results from the fused network significantly improve on single brain networks as well as simple fusion methods.

Chapter 4

Multimodal Disease Modeling via Collective Deep Matrix Factorization

Alzheimer's disease (AD), one of the most common causes of dementia, is a severe irreversible neurodegenerative disease that results in loss of mental functions. The transitional stage between the expected cognitive decline of normal aging and AD, mild cognitive impairment (MCI), has been widely regarded as a suitable time for possible therapeutic intervention. The challenging task of MCI detection is therefore of great clinical importance, where the key is to effectively fuse predictive information from multiple heterogeneous data sources collected from the patients. In this work, we propose a framework to fuse multiple data modalities for predictive modeling using deep matrix factorization, which explores the non-linear interactions among the modalities and exploits such interactions to transfer knowledge and enable high performance prediction. Specifically, the proposed collective deep matrix factorization decomposes all modalities simultaneously to capture non-linear structures of the modalities in a supervised manner, and learns a modality specific component for each modality and a modality invariant component across all modalities. The modality invariant component serves as a compact feature representation of patients that has high predictive power. The modality specific components provide an effective means to explore imaging genetics, yielding insights into how imaging and genotype interact with each other non-linearly in the AD pathology. Extensive empirical studies using various data modalities provided by Alzheimer's

Disease Neuroimaging Initiative (ADNI) demonstrate the effectiveness of the proposed method for fusing heterogeneous modalities.

4.1 Methodology

4.1.1 Matrix factorization

Classical matrix factorization seeks to approximate a matrix with a low-rank matrix, by explicitly learning the matrix factors. Given a data matrix $X \in \mathbb{R}^{m\times n}$, matrix factorization learns two reduced matrix factors $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$, such that $X \approx UV^T$, and $r < \min(m,n)$ is the upper bound of the rank of the approximated matrix UV^T (the rank of UV^T can be less than r if columns of U or V are linearly dependent). The factors U and V are typically learned via an objective function:

$$\min_{U,V}\ d(X,UV^T),\ \text{s.t. } U \in \mathscr{S}_1, V \in \mathscr{S}_2, \tag{4.1}$$

where $d(X,Y)$ is a distance metric function measuring the difference between matrices X and Y, and S_1 and S_2 are two constrains imposed on the factor matrices X and Y.

Typically the distance metric $d(X,Y)$ is chosen to be the Frobenius norm of the difference between X and Y. However, when missing values present in X, $d(X,Y)$ can be defined as the squared ℓ_2 distance between all the observed elements in X and their corresponding elements in Y. As such, we are able to learn matrix factors even with missing values, and the learned matrix factors can then be used to estimate the missing values under the low-rank assumption. This is the setup for matrix completion [22] and is commonly used in recommender systems [61]. The constraints $\mathscr{S}_1$ and $\mathscr{S}_2$ specify the feasible regions of the matrix factors to induce many desired properties, such as non-negativity $\mathscr{S} = \{U | U_{i,j} \geq 0, \forall i,j\}$ in non-negative matrix factorization [69]

and sparsity $\mathscr{S} = \{U | \|U\|_1 \leq z\}$ for interpretable factors [140]. In addition, the complexity control can be implemented using Frobenius constraints $\mathscr{S} = \{U | \|U\|_F^2 \leq z\}$, which are equivalent to the Frobenius norm regularizations [60].

4.1.2 Collective matrix factorization for multimodal analysis

The approximation in (4.1) addresses important semantics in data analysis. When the data matrix X describes the relationship between two types of entities, the factors U and V can be thought of as latent features or latent representations of the entities. For example, in recommender systems we use $X_{i,j}$ to describe the relationship (e.g., rating) between a user i and an item j. The row vector $\mathbf{u}^i \in \mathbb{R}^r$ gives a r-dimensional latent feature representation for the user i and similarly the row vector $\mathbf{v}^j \in \mathbb{R}^r$ is a latent representation of the item j. The two types of latent profile interact with each other linearly in the latent subspace $\mathbb{R}^r$, i.e., the observed relationship in $X_{i,j}$ can be explained as $\mathbf{u}^i(\mathbf{v}^j)^T$.

The latent representation/subspace perspective of matrix factorization allows us to link multiple data modalities, when the entities involved in the modalities are overlapped. In multimodal modeling, assume we have two datasets $X_1 \in \mathbb{R}^{n \times d_1}$ and $X_2 \in \mathbb{R}^{n \times d_2}$ describing the same set of objects from two sets of features. For example, we study a set of n patients. X_1 includes d_1 features from T1 MRI modality and X_2 includes d_2 features from dMRI modality. Then we can apply the matrix factorization procedure to factorize both datasets and connect the two factorizations by enforcing a shared patient latent representation:

$$\min_{U,V_1,V_2} d(X_1, UV_1^T) + d(X_2, UV_2^T), \text{ s.t. } U \in \mathscr{S}_0, V_i \in \mathscr{S}_i, i = 1,2,$$

where the latent representation U is thus jointly learned from two modalities. We call this U

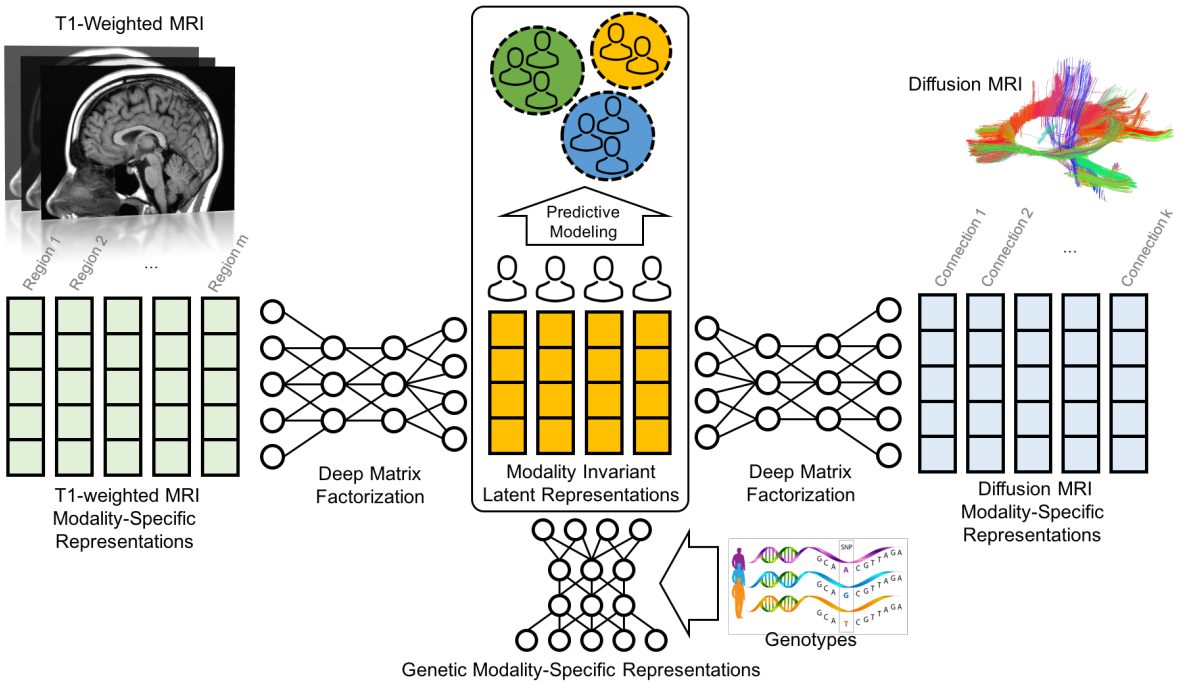


Figure 4.1: **Illustration of proposed collective deep matrix factorization (CDMF) framework.** In this example, CDMF fuses information from three modalities: T1 weighted MRI, diffusion MRI, and genotypes (SNPs) to learn a modality invariant latent representation, to perform predictive modeling.

matrix modality invariant, as the representation now captures intrinsic properties of the patients. When performing regression and classification on patients, instead of using features from raw data matrices X_1 and X_2, we can use the latent representation. We can easily generalize this approach to handle more data modalities.

4.1.3 Capturing complex interactions via collective deep matrix factorization

One essential assumption associated to the classical matrix factorization is the linear dependence in the matrix. Therefore, it implicitly specifies that the latent representations learned from collective matrix factorization have to interact with each other linearly in the learned latent subspace. However, this assumption is too restrictive in many applications, especially in the modeling of

Alzheimer's disease, where imaging modalities and genetic modality are likely to link through a highly non-linearly manner. To capture the complex interactions among modalities, we thus propose a novel framework to fuse multiple data modalities through deep matrix factorization. Assume we have t data modalities $X_1 \in \mathbb{R}^{n\times d_1}, \ldots, X_t \in \mathbb{R}^{n\times d_t}$ describing different views of the same set of n samples. We use a deep neural network $g_\theta(.)$ parameterized θ to factorize each modality, i.e., $X_i \approx U g_{\theta_i}(V_i)$, where in this work we use a structured deep neural network with k layers:

$$g_{\theta_i}(V_i) = f(W_{(k,i)} f(W_{(k-1,i)} f(\ldots, f(W_{(1,i)} V_i)),$$

where $W_{(j,i)}$ is the network weight at the j-th layer, $\theta_i = \{W_{(k,i)}, W_{(k-1,i)}, \ldots, W_{(1,i)}\}$ collectively denotes network weights, and f is a non-linear activation function. The deep network serves as a highly non-linear mapping between input matrix X_i and U, and projects the latent representations non-linearly to the same latent space. We call this $g_{\theta_i}(V_i)$ modality specific component for i-th modality. We can thus perform collective deep matrix factorization (CDMF) to associate multiple data modalities:

$$\min_{U, \{V_i, \theta_i\}_{i=1}^t} \sum_{i=1}^t d(X_i, U g_{\theta_i}(V_i)) \text{ s.t. } U \in \mathscr{S}_0, V_i \in \mathscr{S}_i, \forall i.$$

We would like to highlight one property of collective deep matrix factorization that modality invariant component/representation can have different dimensions from modality components, i.e., U and V can be different, and V in different modalities can also be different. This flexibility is desired especially when different modalities contain different amount of information, and thus the optimal latent representations may have different dimensions. We also note that one way to control the complexity of networks under multiple modalities is to enforce shared network structures, i.e.,

$\{g_{\theta_i}\}$ have the same architecture and share the same parameter values, except for the last layer.

In many applications, our ultimate goal is to build predictive models from multi-modal analysis. To achieve this, we can integrate predictive modeling and collective deep matrix factorization during learning, such that predictive modeling uses latent representations learned from collective deep matrix factorization as input features. Assume that we are given supervision information $\{y_1,\ldots,y_n\}$ for the n subjects, and a linear model for the prediction task $h(U;\mathbf{w}) = U\mathbf{w}$ (with a dummy variable to include bias). Given a latent representation U_j (i.e. the j-th row of U matrix) for the j-th subject and its corresponding label y_j, we use a proper loss function $\ell(h(U_j;\mathbf{w}),y_j)$ (e.g., logistic loss for classification and least squares for regression). The proposed supervised CDMF formulation is thus given by:

$$\begin{aligned} \min_{\mathbf{w},U,\{V_i,\theta_i\}_{i=1}^t} & \sum_{j=1}^n \ell(h(U_j;\mathbf{w}),y_j) + \sum_{i=1}^t \alpha_i d(X_i, U g_{\theta_i}(V_i)) \\ \text{s.t. } & U \in \mathscr{S}_0, V_i \in \mathscr{S}_i, \forall i, \end{aligned} \tag{4.2}$$

where α_i is a tunable parameter to control knowledge fusion proportion of the i-th modality, specifying how much that the modality influences the learning of the modality invariant component. When α_i is large, a less reconstruction error for this modality will be achieved when minimizing overall loss, and therefore the learned representation U contains more information of this modality, and vice versa. Figure 4.1 illustrates the proposed framework fusing three modalities: dMRI, T1 MRI and genotypes (SNPs).

Optimization and initialization. The formulation can be solved efficiently by TensorFlow [1]. However, since the objective in (4.2) is highly non-convex and gradient algorithms may easily trapped in local optima, a good initialization is important for training the network. In this work, we propose to iteratively apply linear matrix factorizations in the original data matrix, and use lin-

ear and hierarchical matrix factors to initialize the deep neural networks. As such, the initialization is similar to a valid linear matrix factorization, and the algorithm iteratively explore non-linear effects within linear latent spaces and capture non-linearity in the network during learning process. Technically we can choose arbitrary linear factorization methods in (4.1) for initialization, however, we find in our experiments that singular vectors given by iterative singular value decomposition (SVD) usually provide decent models that outperform other factorization methods. This may due to fact that orthogonal basis obtained by SVD characterize the optimal linear subspace of the data matrix.

Handling modalities with missing subjects. In many applications especially medical cases, some data modalities may not be available to all samples. For example, some subjects did not participate the genetic study and thus lack genotype information. Besides, in the first stage of ADNI study there are no diffusion MRI imaging available, leading to structured missing patterns in the dataset [132]. Since $\{X_i\}$ involve different sets of subjects, such missing modalities will cause dimension problems in U, and thus the modalities cannot be projected to the same U. One way to overcome this issue is to discard all the subjects with missing modalities and make the dimensions consistent across modalities. However, this approach will significantly reduce the number of samples and thus compromise the predictive performance. We therefore extend the proposed formulation to deal with it. We define an indicator matrix for each modality, where for the i-th modality it is denoted by $I_i \in \mathbb{R}^{n\times n}$, whose j-th row is given by:

$$(I_i)_j = \begin{cases} \mathbf{0} & \text{if the this modality is missing for } j\text{-th subject} \\ \mathbf{e}_j & \text{otherwise} \end{cases},$$

where $\mathbf{e}_j \in \mathbb{R}^n$ is n-dimensional standard basis with only j-th entry as 1. The revised formulation

is given by:

$$\min_{\mathbf{w},U,\{V_i,\theta_i\}_{i=1}^t} \sum_{j=1}^n \ell(h(U_j;\mathbf{w}),y_j) + \sum_{i=1}^t \alpha_i d(\hat{X}_i, I_i U g_{\theta_i}(V_i))$$
$$\text{s.t. } U \in \mathscr{S}_0, V_i \in \mathscr{S}_i, \forall i, \tag{4.3}$$

where $\hat{X}_i$ is an augmented data matrix, whose j-th row is given by:

$$\hat{X}_i^j = \begin{cases} \mathbf{0} & \text{if this subject lacks of } i\text{-th modality} \\ X_i^j & \text{(original features) otherwise} \end{cases}.$$

By multiplying indicators and replacing X_i by $\hat{X}_i$, the corresponding rows of subjects with missing modality will be 0 for this modality, which has no effect on loss. This approach would ensure that we use all the information available during the learning.

Application in Disease Modeling. Even though the proposed CDMF framework can be used in various data mining applications, here we emphasize on its advantages in our specific disease modeling problem. The goal of MCI diagnosis is to differentiate between MCI subjects and normal cognitive (NC) subjects, which is a classification problem. We thus use CDMF in Eq. (4.3) with a logistic loss, in which knowledge from different modalities is fused in a supervised manner such that only the part that is more relevant to group difference of MCI and NC will be fused to the latent representation U, which in turn can improve prediction. This property is important for our multimodal disease modeling since the modalities may contain knowledge that is not relevant to the desired learning task. Without proper guidance, the irrelevant knowledge may negatively impact the representation leading to suboptimal predictive performance. For example, brain imaging may contain information of other inherited brain diseases or aging properties, likewise for genetic data.

If the fusion process is carried out in an unsupervised manner, we may not obtain a U that is most informative regarding the progression of MCI.

Association study of multiple modalities. The interactions between latent representations are of great interests in the community (e.g., generate predictions in the recommender system), and can reveal important insights into how different modalities are connected to each other. Although it is straight forward in linear case that we can use inner products $\mathbf{u}^i(\mathbf{v}^j)^T$, we cannot directly compute this way in CDMF since the modalities are connected through non-linear networks. Instead, we can use the following transformed latent factors:

$$\tilde{V}_i = f(W_{(k,i)}f(W_{(k-1,i)}f(\ldots,f(W_{(1,i)}V_i)), \tag{4.4}$$

which is a mapping matrix that contains the modality specific information of the corresponding modality. All the columns of this matrix form the specific feature space of this modality. Hence, we can calculate the association of any features between any two modalities using the transformed latent factors $\tilde{V}_i$. Let $C_{i,j}(m,n)$ denote the cosine similarity between the m-th column from $\tilde{V}_i$ and the n-th column from $\tilde{V}_j$. When $C_{i,j}(m,n)$ is large, the m-th feature of i-th modality is highly related with the n-th feature of j-th modality and a small $C_{i,j}(m,n)$ indicates the association between those features is weak. This provides a novel tool to study the imaging genetics, identifying how genotypes influence brain structures under specific tasks (e.g., MCI prediction in our case).

4.2 Experiments

4.2.1 Dataset and features

Data from two stages of ADNI are used in this study: ADNI1 and ADNI2. Detail demographic characteristics and missing data information are listed in Table 4.1. Whole genome sequencing (WGS) SNPs are provided by ADNI and used as genetic modality in our study. For MRI, ADNI1 participants are scanned by 1.5T or 3T MRI scanner while all ADNI2 participants are scanned by 3T MRI scanner [1]. FreeSurfer V5.3 is adopted to extract 333 measures include the area, thickness, cortical volume, subcortical volume and white matter volume from T1 MRI to form T1 MRI modality. For dMRI, we first parcellate the brain into 113 cortical and subcortical region-of-interests (ROIs) according to the Harvard Oxford Cortical and subcortical Probabilistic Atlas [33]. Then we reconstruct the whole-brain tractography using an ODF-based probabilistic approach: PICo[32]. Finally, a brain network is generated in which the nodes indicate ROIs and the edges are determined by the proportion of fibers intersecting with each pair of ROIs. As such, each brain network is a 113×113 symmetric matrix with 6328 distinct edges. These 6328 edges are used as the feature variables for dMRI modality.

4.2.2 Data preprocessing

Imaging modalities preprocessing. ADNI1 and ADNI2 use different scanner protocol which may introduce biases for the datasets. Hence, we decide to harmonize the cohorts by removing this cohort effect. We create an indicator variable to differentiate ADNI1 and ADNI2 with 1 for all subjects from ADNI1 and -1 for all subjects from ADNI2. In addition, age and sex are common confounders biasing the analysis. In this study, generalized linear regression approach [80] is used

[1] http://adni.loni.usc.edu/data-samples/mri/

ADNI1 Cohort	NC	MCI	Total
Age	75.84±4.95	74.48±7.48	75.17±6.68
Sex	115M/108F	247M/138F	362M/246F
total subjects	223	385	608
Subjects with dMRI	0	0	0
Subjects with T1 MRI	223	385	608
Subjects with genotype	202	348	550
ADNI2 Cohort	NC	MCI	Total
Age	69.36 ± 15.40	71.68 ± 9.93	70.96 ± 11.89
Sex	22M/28F	71M/41F	93M/69F
total subjects	50	112	162
Subjects with dMRI	50	112	162
Subjects with T1 MRI	50	112	162
Subjects with genotype	27	82	109

Table 4.1: **Demographic information of subjects.**

to remove all confounders including age, sex and cohort index. It assumes each observed variable is linearly dependent on the confounder variables and fitting a generalized linear model can remove confounders' effect. Denote the observed variable of variable X as X^{obs} and the original variable as X^{ori}. The linear dependence of X^{obs} and X^{ori} is:

$$X^{obs} = w_1 \cdot age + w_2 \cdot sex + w_3 \cdot cohort + X^{ori},$$

where w_1, w_2, w_3 are coefficients of confounders. Let (w_1, w_2, w_3) be $\boldsymbol{w}$ and $(age_i, sex_i, cohort_i)$ be t_i, where i denotes the i-th subject. Coefficients can be obtained by solving a linear regression:

$$\boldsymbol{w}^* = \min_{\boldsymbol{w}} \sum_{i=1}^{n} (\boldsymbol{w}^T t_i - X_i^{obs})^2. \tag{4.5}$$

After solving Eq. (4.5), the original feature variable is given by:

$$X^{ori} = X^{obs} - (w_1 \cdot age + w_2 \cdot sex + w_3 \cdot cohort).$$

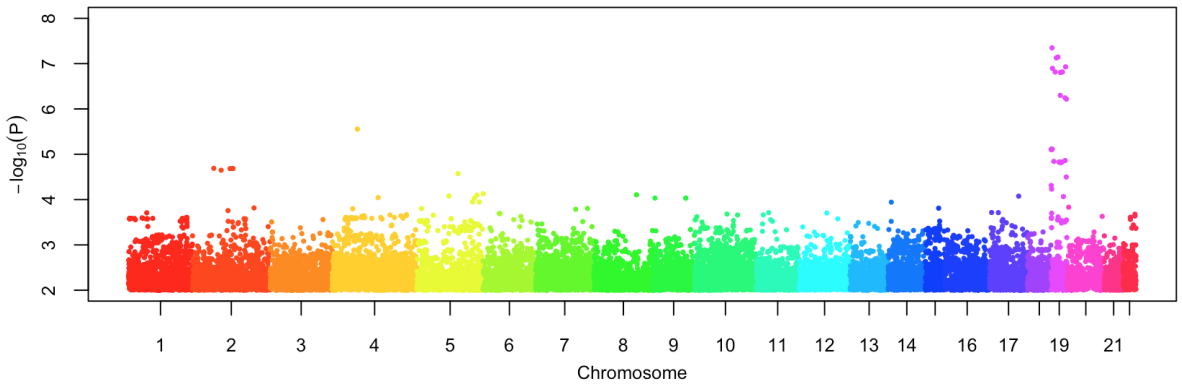


Figure 4.2: **Manhattan plot for SNPs with adjusted p value greater than 2.** Colors indicate different chromosomes.

We apply this on both T1 MRI data and dMRI data and will only use X^{ori} in the downstream experiments.

Genetic modality preprocessing. Genetic data is preprocessed by standard quality control using PLINK[2] and then impute using MaCH[3]. SNPs with minor allele frequency (MAF) less than 5% or missing values greater than 5% are discarded. Subjects with missing values greater than 10% at all SNPs are removed. Finally, 659 subjects with reading values on $6,566,154$ SNPs are attained.

In order to extract more relevant features, we apply genome-wide association study (GWAS) on our data. In detail, we regress patient state NL/MCI on each SNP using logistic regression, with p-value generated and adjusted to $-\log_{10}$ scale. Larger adjusted p-value indicates strong association between response and the marker. Figure 4.2 shows SNPs with adjust p value greater than 2 on each chromosome. SNPs on chromosome 19 have stronger association with MCI than others, suggesting crucial effects of this chromosome on the Alzheimer's deterioration. Finally, the top 200 significant SNPs for each iteration are retained as features for our downstream analysis. Since SNPs are categorical, i.e. $\{0,1,2\}$, we use the one-hot coding to be the feature representation. Hence, the final feature dimension for genetic modality is 600.

[2]http://pngu.mgh.harvard.edu/ purcell/plink/
[3]http://csg.sph.umich.edu/abecasis/MaCH/

4.2.3 Predict performance

Comp. #	Shallow collective matrix factorization		
	linear	sigmoid	square
30	0.529 ± 0.080	0.616 ± 0.102	0.564 ± 0.011
50	0.587 ± 0.069	0.593 ± 0.120	0.718 ± 0.076
70	0.610 ± 0.079	0.644 ± 0.075	0.659 ± 0.161
90	0.526 ± 0.065	0.597 ± 0.086	0.634 ± 0.097
110	0.656 ± 0.089	0.681 ± 0.116	0.658 ± 0.106
130	0.561 ± 0.024	0.613 ± 0.105	0.668 ± 0.127
Comp. #	Deep collective matrix factorization		
	linear	sigmoid	square
30	0.519 ± 0.099	0.653 ± 0.139	0.719 ± 0.142
50	0.594 ± 0.151	0.646 ± 0.078	0.693 ± 0.100
70	0.573 ± 0.135	0.593 ± 0.165	0.758 ± 0.115
90	0.519 ± 0.093	0.610 ± 0.146	$\mathbf{0.805 \pm 0.073}$
110	0.558 ± 0.083	0.542 ± 0.048	0.726 ± 0.027
130	0.553 ± 0.124	0.544 ± 0.110	0.679 ± 0.152
Comp. #	Other deep multimodal methods		
	DCCA	DCCAE	DNN
30	0.770 ± 0.065	0.723 ± 0.031	0.617 ± 0.143
50	0.722 ± 0.088	0.743 ± 0.094	0.604 ± 0.026
70	0.689 ± 0.134	0.780 ± 0.054	0.560 ± 0.111
90	0.684 ± 0.089	0.703 ± 0.042	0.579 ± 0.068
110	$*$	0.735 ± 0.135	0.627 ± 0.165
130	$*$	0.699 ± 0.089	0.689 ± 0.131

Table 4.2: **Prediction performance of different models using ADNI2's T1 MRI and dMRI in terms of AUC.** With an appropriate activation function and components' number, our method outperforms than all other methods. $*$] means not applicable due to the algorithm design.

In this section, we evaluate the performance of our method and compare with other methods using ADNI dataset. The distance metric $d(X,Y)$ we used in the following experiments is $\|X - Y\|_F^2$. We perform experiments on three different settings.

In the first setting, only ADNI2 dataset and its two modalities: T1 MRI and dMRI are covered. In this setting, no modality has missing subjects. We randomly select 90% subjects as the training set and 10% subjects as the testing set. Our main assumption is deep matrix factorization can ex-

Comp. #	Shallow collective matrix factorization		
	linear	sigmoid	square
30	0.702 ± 0.019	0.672 ± 0.137	0.708 ± 0.024
50	0.749 ± 0.052	0.793 ± 0.034	0.742 ± 0.063
70	0.743 ± 0.063	0.696 ± 0.037	0.747 ± 0.061
90	0.754 ± 0.046	0.756 ± 0.059	0.749 ± 0.049
110	0.791 ± 0.027	0.798 ± 0.058	0.786 ± 0.032
130	0.671 ± 0.049	0.652 ± 0.058	0.679 ± 0.048
Comp. #	Deep collective matrix factorization		
	linear	sigmoid	square
30	0.634 ± 0.065	0.665 ± 0.044	0.627 ± 0.768
50	0.701 ± 0.064	0.735 ± 0.061	0.681 ± 0.039
70	0.778 ± 0.059	0.749 ± 0.011	0.784 ± 0.055
90	0.775 ± 0.063	0.801 ± 0.023	$\mathbf{0.821 \pm 0.015}$
110	0.806 ± 0.049	0.792 ± 0.031	0.800 ± 0.032
130	0.717 ± 0.037	0.705 ± 0.049	0.759 ± 0.044
Comp. #	Other deep multimodal methods		
	DCCA	DCCAE	DNN
30	0.801 ± 0.101	0.737 ± 0.063	0.758 ± 0.098
50	0.732 ± 0.041	0.753 ± 0.014	0.767 ± 0.069
70	0.788 ± 0.084	0.813 ± 0.047	0.756 ± 0.087
90	0.746 ± 0.159	0.750 ± 0.124	0.757 ± 0.078
110	0.759 ± 0.151	0.780 ± 0.058	0.754 ± 0.070
130	0.739 ± 0.183	0.774 ± 0.074	0.754 ± 0.056

Table 4.3: **Prediction performance of different models using ADNI2 and ADNI1's T1 MRI and dMRI in terms of AUC.** Although dMRI modality lacks of a large number of subjects, performance is still improved a lot compared with that only uses ADNI2 data.

Components #	Shallow collective matrix factorization		
	linear	sigmoid	square
30	0.684 ± 0.051	0.658 ± 0.039	0.766 ± 0.115
50	0.767 ± 0.019	0.772 ± 0.032	0.818 ± 0.076
70	0.763 ± 0.059	0.759 ± 0.020	0.797 ± 0.049
90	0.772 ± 0.070	0.775 ± 0.030	0.767 ± 0.081
110	0.822 ± 0.018	0.795 ± 0.005	0.803 ± 0.014
130	0.702 ± 0.067	0.669 ± 0.055	0.689 ± 0.071
Components #	Deep collective matrix factorization		
	linear	sigmoid	square
30	0.632 ± 0.019	0.665 ± 0.042	0.670 ± 0.052
50	0.707 ± 0.054	0.737 ± 0.064	0.719 ± 0.073
70	0.781 ± 0.065	0.750 ± 0.010	0.799 ± 0.040
90	0.784 ± 0.071	0.797 ± 0.019	$\mathbf{0.852 \pm 0.018}$
110	0.811 ± 0.047	0.782 ± 0.030	0.779 ± 0.008
130	0.728 ± 0.048	0.705 ± 0.055	0.725 ± 0.105

Table 4.4: **Prediction performance of fusing genetic knowledge and imaging knowledge using ADNI1 and ADNI2 in terms of AUC.** Genetic modality can be successfully integrated with imaging modalities.

Components #	30	50	70
DNN	0.674 ± 0.114	0.666 ± 0.108	0.669 ± 0.119
Components	90	110	130
DNN	0.667 ± 0.090	0.656 ± 0.080	0.671 ± 0.098

Table 4.5: **Prediction performance of DNN using ADNI1 and ADNI2 in terms of AUC.** Genetic modality can be successfully integrated with imaging modalities.

tract high-level nonlinear features to improve diagnosis performance. In order to prove it, we compare deep models with shallow models, i.e. one layer matrix factorization, and compare nonlinear models with linear models. Two main nonlinear functions are used in our experiments: $sigmoid(x)$ and x^2. In deep models, we focus on those with two hidden layers. After some preliminary experiments, we fix the first layer's components to be 162, i.e. $V_j \in R^{162 \times d_j}$ for $j = 1, 2, ..., t$ and vary second layer's components from 30 to 130, i.e. $W_{1,j} \in R^{r \times 162}$ where $r \in \{30, 50, 70, 90, 110, 130\}$. Hence, $U \in R^{n_j \times r}$. How the new features' dimension affects performance can be traced by varying r. We report average area under ROC curve (AUC) over three iterations in Table 4.2. We imple-

mented the proposed model using TensorFlow [1]. All the experiments were run on GT1080 or Titan X. It takes approximately 3 minutes to train one model.

When using x^2 as activation function and setting components number to be 90, our model outperforms all other models. We observe when the activation function is inappropriate, i.e, $sigmoid(x)$ for our case, the AUC is very low. Hence, choosing a suitable activation function is very important. Only certain nonlinear functions can correctly fit this dataset and extract the desired features. Also, we find the number of components is crucial for all different models. An inappropriate number of components will reduce the performance drastically. When the number of components is too small, new feature representation is not rich enough to capture the complex hidden information. But when this number becomes too large, they contain too many redundant features. Since sample size is not large enough, it causes overfitting and reduces testing performance. We also compare our method with three state-of-the-art multimodal learning algorithms: DCCA, DCCAE and deep neural network. Since training sample size is 90, when the components number of new feature representation is larger than 90, DCCA's code[4] reports error. Hence, we set it to be $\{30, 50, 70, 90\}$ for DCCA. The deep neural network has two parts. The first part is used to remove modality specific information. It has two two-layer sub-networks corresponding to two modalities. The first layer is the input layer. To make the network consistent. The second layer contains 162 neurons for each sub-network. The outputs of two sub-networks are concatenate to a vector and used as the input of the second part of the whole network to fuse knowledge and implement classification tasks. The second part has three layers. The first layer is the input layer where the output of the first part is fed. The second layer contains $\{30, 50, 70, 90, 110, 130\}$ units. The third layer is a logistic regression layer. To compare with our model, the two parts are jointly trained. The results are reported in the last three columns in Table 4.2. Our method outperforms

[4]http://ttic.uchicago.edu/ wwang5/dccae.html

all baselines.

In the second setting, we include all ADNI1 subjects' imaging data into the training set. Compared with the first setting, dMRI modality has a lot of missing subjects in this setting. Also, this setting's training sample size is much larger than the previous one. In order to compare the performance of these two settings, the testing data set and all the other model settings are the same as in the first setting. Since DNN, DCCA and DCCAE cannot deal with modality with missing subjects, we fill all the missing values with the mean over all available samples for each modality. Average AUC is reported in Table 4.3 for all models and similar trends are observed in these results with those in the first setting. Moreover, we find under the same experiment settings, almost all models' performance is higher than that of the previous one. It shows our extended formulation can successfully deal with modality with missing subjects and leverage partial knowledge in this modality to greatly improve overall performance.

In the last setting, we include genetic modality as the third modality and fuse genetic knowledge and imaging knowledge to improve diagnosis performance. We preform GWAS on each iteration's training set to select SNPs involved in our experiment. To compare with the second setting, all the model settings are the same as in previous settings. Average AUC is reported in Table 4.4 and TabRefDNN. Since DCCA and DCCAE cannot deal with three modalities, we only use DNN as baseline. With the same training sample size, DNN's performance is much worse than that of previous setting, which implies concatenating all the output of each sub-network as fusion method does not work for this case. That is because features from the genetic modality are discrete and the matrix is very sparse, while features for two imaging modalities are continues and the matrices are extremely dense. They have different statistical properties. However, for our method, the performance for this setting is much better than that of the second setting, which implies genetic modality can be successfully integrated with imaging modalities by our method even though the

modalities are radically different.

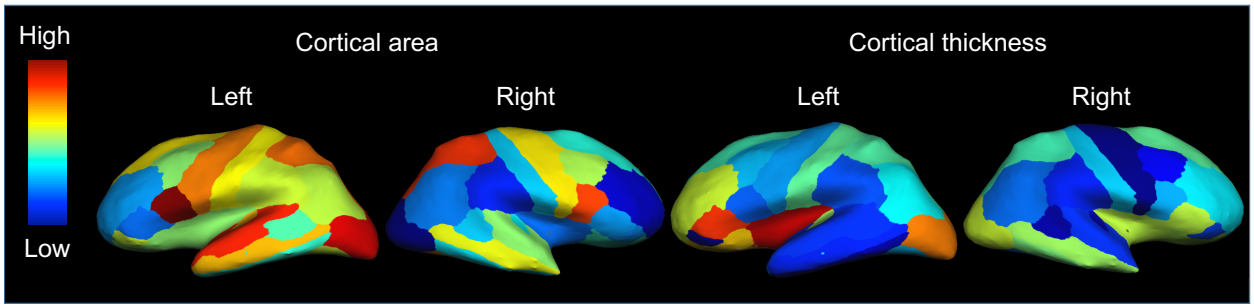


Figure 4.3: **Brain maps of the significance level at each ROI for the most associated SNP within that ROI.**

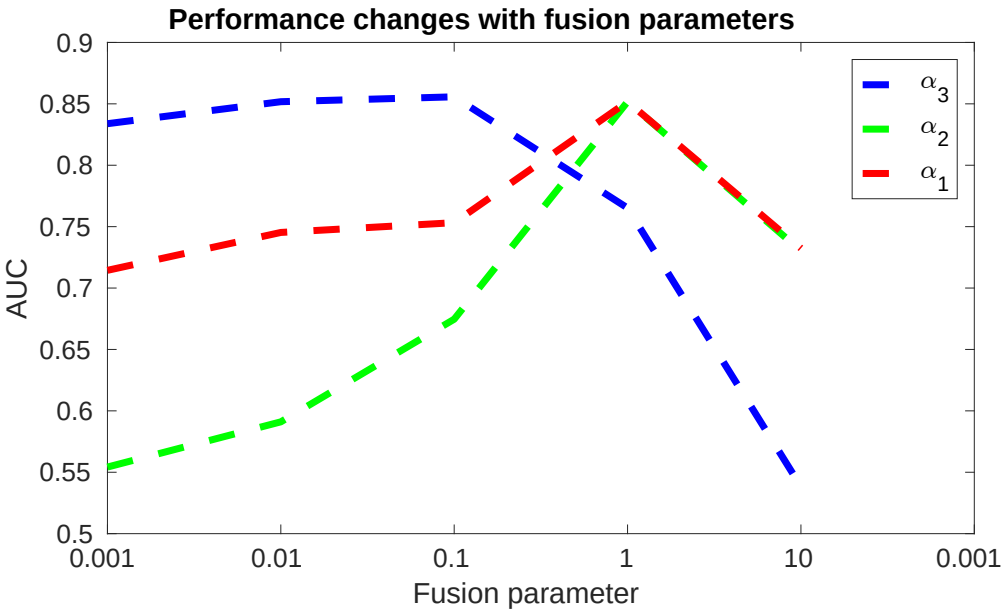


Figure 4.4: **Testing performance with varying α parameters.**

At last, we report sparse logistic regression results on each single modality as single modality baselines. The results are shown in Table 4.6. Experiments on ADNI2 dataset have the same training testing splitting method as the first setting and experiments on ADNI1 + ADNI2 dataset have the same splitting way as the second setting. We see single modality's average AUC is lower than the highest AUC in all three settings. Hence, only by fusing knowledge from different modalities can we achieve descent performance.

4.2.4 Effects of knowledge fusion parameters

Knowledge fusion parameters control how much knowledge a modality is fused into modality invariant term. In this section, we show how these parameters affect performance. Let α_1, α_2, α_3 be the parameters to control knowledge fusion of dMRI, T1 MRI and SNPs respectively. The training set and testing set are split in the same way as the third setting in the last section. We focus on deep model with 2 hidden layers, with x^2 as activation function. The components of the first layer and the second layer is 162 and 90 respectively.

We first fix α_1 and α_2 to be 1 and vary α_3 to see how α_3 affects performance. The results is shown in Figure 4.4 in blue line. We see when we increase α_3, the performance first increases slightly. But when α_3 is larger than 0.1, the performance decreases very fast if we continue increasing it. That is because genetic modality is noisier than imaging modalities. With a small α_3, i.e. 0.1, this model can tolerant a larger reconstruction error for genetic modality. Hence, the model is robust to the noise in genetic modality. When α_3 becomes larger, the reconstruction error of genetic modality must be small in order to achieve a low total loss. More noise distorts U, which reduces the performance. But when α_3 is too small, some useful knowledge of this modality cannot all be fused to U, which also reduces the performance. Hence, only with a suitable fusion parameter can the model correctly fuses all the useful knowledge of genetic modality. Next, we fix α_3, α_1 to be 0.1 and 1 receptively and vary α_2 to see how α_2 affects performance. We also fix α_3, α_2 to be 0.1 and 1 respectively and vary α_1 to see the effects of changing α_1. The results are shown in Figure 4.4 in green line and red line. These two are very similar to each other since they both control knowledge fusion of imaging modalities. We see when α_1 and α_2 reach 1, the performance reaches the highest. Hence, imaging modalities need to contribute more knowledge to U than genetic modality to make a better performance.

	ADNI2		
	T1 MRI	dMRI	SNPs
AUC	0.71 ± 0.04	0.63 ± 0.07	0.63 ± 0.14
	ADNI1+ADNI2		
	T1 MRI	dMRI	SNPs
AUC	0.72 ± 0.06	-	0.67 ± 0.27

Table 4.6: **Results of applying sparse logistic regression on each single modality in terms of AUC.** ADNI1 study did not collect dMRI.

4.2.5 Imaging-genetics association

In this section, we present imaging-genetics association uncovered by modality specific components. We compute the association between SNPs with cortical thickness and area on 68 ROIs. This association indicates how significant a brain imaging feature is associated with a SNP under the task of predicting MCI. In Figure 4.3, we show the map of the significance level at each ROI for the most associated SNP within that ROI. The first two figures are based on cortical area features for left and right brain respectively and the last two figures are for cortical thickness features. Warmer colors represent stronger association and cooler colors indicate the opposite. Our results show that there are some cluster patterns which indicate those ROIs are highly related to each other in respect of MCI. Top 6 significant T1 MRI features are: right cuneus thickness, right parahippocampal area, right posterior cingulate thickness, left pars opercularis area, left cuneus thickness and right frontal pole thickness. Among those features, cuneus thickness, posterior cingulate thickness, frontal pole thickness and parahippocampal region are identified significantly associated with MCI [82, 26, 45, 34]. The SNPs most related to these 6 features are: rs10414043, rs429358, rs429358, rs8141950, rs11178933, rs10414043 respectively. All the SNPs except rs8141950 are located at Chromosome19 which has been identified to be highly associated with MCI and AD [30, 71]. Especially, rs429358 locates in the fourth exon of the APOE gene [57] in Chromosome19, which has been extensively reported as the genetic risk factor for the late-onset

of AD. rs8141950, located on Chromosome22, has also been found to be closely related to AD [5]. This shows that our method can correctly uncover imaging-genetic association in respect of MCI. This association can be used to analyze how the genotype influences brain structures and provide a potential way to explore the mechanism behind MCI and AD.

4.3 Summary

In this work, we proposed collective deep matrix factorization to fuse knowledge from different modalities. Specifically, we build uniform nonlinear hierarchical deep matrix factorization framework across different modalities which decomposes each modality into a modality specific component and a modality invariant component that serves as a learned feature representation. We also add supervision on the modality invariant component to guide the learning process. The proposed method can exploit complicated non-linear interactions among different modalities and learn a feature representation which is compact and more relevant to our predictive problem. Also, the modality specific term can be used to uncover complicated imaging-genetic associations. We perform extensive experiments on ADNI dataset and show the proposed method significantly improves predictive performance.

Chapter 5

Multimodal Information Bottleneck

In many classification problems, the predictions can be enhanced by fusing information from different data modalities. In particular, when the information from different modalities complement each other, it is expected that multimodal learning will lead to improved predictive performance. In this paper, we proposed a supervised multimodal learning framework based on the information bottleneck principle to filter out irrelevant and noisy information from multiple modalities and learn an accurate joint representation. Specifically, our proposed method maximizes the mutual information between the labels and the learned joint representation while minimizing the mutual information between the learned latent representation of each modality and the original data representation. As the relationships between different modality are often complicated and nonlinear, we employed deep neural networks to learn the latent representation and to disentangle their complex dependencies. However, since the computation of mutual information can be intractable, we employed the variational inference method to efficiently solve the optimization problem. We performed extensive experiments on various synthetic and real-world datasets to demonstrate the effectiveness of the framework.

5.1 Methodology

5.1.1 Information Bottleneck Method

Information bottleneck [108] is an approach based on information theory. It formalizes the intuitive ideas about information to provide a quantitative measure of “meaningful" and “relevant" [108]. It provides a tradeoff between accuracy and complexity. This method has been widely used in clustering [98, 109, 42], ranking [50] and classification [97]. Exact solution does not exist if the latent representation is learned by deep neural networks. In [7], the authors applied information bottleneck to single-modal learning and proposed to use the variational method to optimize it. Instead of directly solving the optimization problem of information bottleneck, the authors first calculated a lower bound of the original target. Then the lower bound was maximized to push the results closer to the optimal solution to the original optimization problem. The distributions of the posteriors were learned by the neural networks. The method also utilized the reparameterization trick for efficient training. Information bottleneck is also used in multimodal learning. In [127], the authors proposed to use information bottleneck to learn a joint latent representation. The joint latent representation was a combination of the linear projection of all the modalities. The projection matrices were learned by the information bottleneck approach. Although the approach achieves decent results, it is limited to linear projection. Therefore, we propose a nonlinear deep version of multimodal information bottleneck to overcome this limitation.

Information bottleneck is an information-based approach to find the best tradeoff between the accuracy and complexity. Given data X with labels Y, information bottleneck aims to find a concise and accurate latent representation of X. Denote the latent representation as Z. Information

bottleneck solves the following optimization problem:

$$\max_{Z} I(Y,Z) \quad \text{s.t.} I(X,Z) \leq \gamma, \tag{5.1}$$

where $I(Y,Z)$ is the mutual information between Y and Z whereas $I(X,Z)$ is the mutual information between X and Z. The mutual information between any two random variables X and Y is defined as:

$$I(X,Y) = \int_Y \int_X p(x,y) \log(\frac{p(x,y)}{p(x)p(y)}) dxdy,$$

where $p(x,y)$ is the joint probability density function of X and Y while $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y.

Eq. (5.1) maximizes the mutual information between Y and Z to make sure the learned Z contains information about Y as much as possible. If there is no constraint on Z, the solution would be $Z = X$. But in most cases, X contains noise or other irrelevant information to Y. Therefore, a constraint must be applied to Z to ensure that the learned Z provides a concise representation that contains less noise and irrelevant information compared with X. This constraint reduces the model complexity and improves the model's generalization ability. Eq. (5.1) can also be relaxed to the following formulation:

$$\max_{Z} I(Y,Z) - \alpha I(X,Z),$$

where α is a regularization parameter to control the tradeoff between $I(Y,Z)$ and $I(X,Z)$.

5.1.2 Deep multimodal information bottleneck.

In multimodal learning, information bottleneck can be used to learn the joint discriminative representation as it can remove the irrelevant information and noise of each modality. Since for real-world data, the relation between multiple modality are likely to be nonlinear and complex, in this paper, we propose a deep multimodal information bottleneck method to map the original representation to a nonlinear representation that can make subjects easier to be separated.

Given two modalities X_1, X_2 and the class labels Y, the proposed method aims to learn a joint representation Z to fuse the information from all modalities. The model contains two parts. The first part is to learn the hidden representations from all the modalities. Each modality has one hidden representation. This part is to remove the noise and irrelevant information from X_1 and X_2 as much as possible to make sure the learned representations are very concise. We use Z_1 and Z_2 to denote the hidden representations for X_1 and X_2, respectively. The second part is to fuse the hidden representations using a neural network as

$$Z = f_\theta(Z_1, Z_2), \tag{5.2}$$

where f denote the network and θ the network parameter. This part is to transfer knowledge from all modalities and learn a joint representation Z. These two parts are learned jointly by the information bottleneck as

$$\begin{aligned} \max_{Z,Z_1,Z_2} & I(Y,Z) - \alpha I(X_1,Z_1) - \beta I(X_2,Z_2), \\ \text{s.t. } & Z = f_\theta(Z_1,Z_2), \end{aligned} \tag{5.3}$$

where α and β are regularization parameters. The first term is to maximize the mutual information

between the joint representation and the label Y to make sure the learned joint representation are discriminative according to the class labels. The last two terms are to minimize the mutual information between the latent representation of each modality and its original data representation. These two terms reduce the model complexity to make the model more generalizable, since they can filter out the irrelevant and noisy information.

5.1.3 Optimization

The major challenge of solving Eq. (5.3) is that the mutual information terms are computationally intractable. Recently, variational methods [59, 7, 38] are widely used to deal with such problems. Variation methods maximize the variational lower bounds of the objective functions instead of directly maximizing them. These methods use some known distributions to approximate the intractable distributions, and provide lower bounds of the original objective functions. By increasing the lower bounds, we can obtain approximate solutions to the original objective functions. To obtain the variational lower bound of Eq. (5.3), we first need to find the joint probability density function of all the variables including the latent variables. Using Bayes' rule, the joint probability density function of X_1, X_2, Z_1, Z_2, Y, Z can be expressed as

$$\begin{aligned} p(x_1, x_2, z_1, z_2, y, z) = & p(z|z_1, z_2, x_1, x_2, y) \\ & p(z_1|z_2, x_1, x_2, y) \\ & p(z_2|x_1, x_2, y)p(x_1, x_2, y). \end{aligned} \tag{5.4}$$

Since Z_1 is learnt from X_1, we thus assume given X_1, Z_1 is independent of Z_2, X_2, Y. Similarly, we assume given X_2, Z_2 is independent of X_1, Y, and given Z_1, Z_2, Z is independent of X_1, X_2, Y.

Therefore, we have the following equalities:

$$p(z_1|z_2,x_1,x_2,y) = p(z_1|x_1),$$
$$p(z_2|x_1,x_2,y) = p(z_2|x_2),$$
$$p(z|z_1,z_2,x_1,x_2,y) = p(z|z_1,z_2).$$

Using these assumptions, the joint probability density function can be simplified as

$$\begin{aligned} p(x_1,x_2,z_1,z_2,y,z) =& p(z|z_1,z_2)p(z_1|x_1) \\ & p(z_2|x_2)p(x_1,x_2,y). \end{aligned} \tag{5.5}$$

First, let us start with $I(Y,Z)$. Since $p(y|z)$ is intractable, we use a distribution $q(y|z)$, which will be learned from the network, to approximate $p(y|z)$. The KL-divergence between $p(y|z)$ and $q(y|z)$ is always non-negative. Therefore, we have

$$\begin{aligned} & KL[p(y|z),q(y|z)] \geq 0 \\ \Rightarrow & \int dydz\ p(y,z)\log(p(y|z)) \geq \int dydz\ p(y,z)\log(q(y|z)). \end{aligned} \tag{5.6}$$

The mutual information between Y and Z is

$$\begin{aligned} I(Z,Y) &= \int dydzp(y,z)\log\frac{p(y,z)}{p(y)p(z)} \\ &= \int dydzp(y,z)\log\frac{p(y|z)}{p(y)}. \end{aligned}$$

Using Eq. (5.6), we have

$$
\begin{aligned}
I(Y,Z) &\geq \int dydz\ p(y,z)\log\frac{q(y|z)}{p(y)} \\
&= \int dydz\ p(y,z)\log q(y|z) - \int dy\ p(y)\log p(y).
\end{aligned}
$$

Since $-\int dy\ p(y)\log p(y)$ is the entropy of the labels, and this term have no effect on the optimization, we can directly drop it. Therefore, the variation lower bound of $I(Y,Z)$ is

$$
\begin{aligned}
I(Y,Z) &\geq \int dydz\ p(y,z)\log q(y|z) \\
&= \int dydzdx_1dx_2dz_1dz_2\ p(x_1,x_2,z_1,z_2,y,z)\log q(y|z).
\end{aligned}
$$

By using the joint probability density function in Eq. (5.5), the variational lower bound of the mutual information between Z and Y can be written as

$$
\begin{aligned}
I(Y,Z) \geq & \int dx_1dx_2dy\ p(x_1,x_2,y) \\
& \int dzdz_1dz_2p(z|z_1,z_2)p(z_1|x_1)p(z_2|x_2)\log q(y|z). \qquad (5.7)
\end{aligned}
$$

Next, we need to find the upper bound of $I(X_1,Z_1)$. Since $p(z_1)$ is intractable, we use $r_1(z_1)$ to approximate $p(z_1)$. Similarly, we use the property of the KL-divergence between $p(z_1)$ and $r_1(z_1)$.

$$
\begin{aligned}
& KL[p(z_1),r_1(z_1)] \geq 0 \\
\Rightarrow & \int dzp(z_1)\log p(z_1) \geq \int dzp(z_1)\log r(z_1).
\end{aligned}
$$

Therefore, the mutual information between Z_1 and X_1 is

$$\begin{aligned} I(Z_1,X_1) &= \int dz_1 dx_1 p(x_1,z_1) \log \frac{p(z_1|x_1)}{p(z_1)} \\ &\leq \int dz_1 dx_1 \; p(x_1,z_1) \log \frac{p(z_1|x_1)}{r_1(z_1)} \\ &= \int dx_1 dx_2 dy dz_1 \; p(x_1,x_2,z_1,y) \log \frac{p(z_1|x_1)}{r_1(z_1)}. \end{aligned}$$

Using the assumption that given x_1, z_1 is independent of all other variables, we have

$$\begin{aligned} I(Z_1,X_1) \leq & \int dx_1 dx_2 dy p(x_1,x_2,y) \\ & \int dz_1 \; p(z_1|x_1) \log \frac{p(z_1|x_1)}{r_1(z_1)}. \end{aligned} \tag{5.8}$$

Similarly, for $I(Z_2,X_2)$, we have

$$\begin{aligned} I(Z_2,X_2) \leq & \int dx_1 dx_2 dy p(x_1,x_2,y) \\ & \int dz_2 \; p(z_2|x_2) \log \frac{p(z_2|x_2)}{r_2(z_2)}. \end{aligned} \tag{5.9}$$

With Eq. (5.7), Eq. (5.8) and Eq. (5.9), the final variational lower bound is:

$$\begin{aligned} & I(Y,Z) - \alpha I(X_1,Z_1) - \beta I(X_2,Z_2) \\ \geq & \int dx_1 dx_2 dy \; p(x_1,x_2,y) \\ & (\int dz dz_1 dz_2 p(z|z_1,z_2) p(z_1|x_1) p(z_2|x_2) \log q(y|z) \\ & - \alpha \int dz_1 \; p(z_1|x_1) \log \frac{p(z_1|x_1)}{r_1(z_1)} \\ & - \beta \int dz_2 \; p(z_2|x_2) \log \frac{p(z_2|x_2)}{r_2(z_2)}). \end{aligned}$$

The integral over x_1, x_2 and y can be approximated by Monte Carlo sampling [94]. Therefore,

$$
\begin{aligned}
& I(Y,Z) - \alpha I(X_1,Z_1) - \beta I(X_2,Z_2) \\
& \geq \frac{1}{N}\sum_{i}^{N}\{\int dz dz_1 dz_2 p(z|z_1,z_2)p(z_1|x_1)p(z_2|x_2) \\
& \log q(y|z) - \alpha \int dz_1 p(z_1|x_1)\log\frac{p(z_1|x_1)}{r_1(z_1)} \\
& - \beta \int dz_2 p(z_2|x_2)\log\frac{p(z_2|x_2)}{r_2(z_2)}\},
\end{aligned}
$$

where N is the sample size of the total sampled data. Next, we assume $p(z_1|x_1), p(z_2|x_2)$ and $p(z|z_1,z_2)$ are Gaussian. The means and variances of the Gaussian distributions are all learned from deep neural networks, i.e.,

$$
\begin{aligned}
p(z_1|x_1) &= \mathcal{N}(\mu_1(x_1;\phi_1), \Sigma_1(x_1;\phi_1)), \\
p(z_2|x_2) &= \mathcal{N}(\mu_2(x_2;\phi_2), \Sigma_2(x_2;\phi_2)), \\
p(z|z_1,z_2) &= \mathcal{N}(\mu(z_1,z_2;\theta), \Sigma(z_1,z_2,;\theta)),
\end{aligned}
$$

where μ_1, μ_2, μ and $\Sigma_1, \Sigma_2, \Sigma$ are the networks to learn the means and variances for $p(z_1|x_1)$, $p(z_2|x_2)$ and $p(z|z_1,z_2)$. ϕ_1, ϕ_2 and θ are network parameters for the networks to learn $p(z_1|x_1)$, $p(z_2|x_2)$ and $p(z|z_1,z_2)$, respectively. Since z_1, z_2 and z are all random variables, backpropagation through those random variables may cause problems. Therefore, we use the reparameterization

trick here, i.e.,

$$z_1 = \mu(x_1;\phi_1) + \Sigma(x_1;\phi_1)\varepsilon_1,$$

$$z_2 = \mu(x_1;\phi_1) + \Sigma(x_1;\phi_1)\varepsilon_2,$$

$$z = \mu(z_1,z_2;\theta) + \Sigma(z_1,z_2,;\theta)\varepsilon,$$

where $\varepsilon,\varepsilon_1,\varepsilon_2 \sim \mathcal{N}(0,I)$. By using this reparameterization trick, randomness is transfered to $\varepsilon,\varepsilon_1,\varepsilon_2$, which do not affect the backpropagation. Therefore, the final loss is

$$\max \frac{1}{N}\sum^{N}\{\mathbb{E}_{\varepsilon}\mathbb{E}_{\varepsilon_1}\mathbb{E}_{\varepsilon_2}\log q(y|z) - \alpha\mathbb{E}_{\varepsilon_1}\log\frac{p(z_1|x_1)}{r_1(z_1)} - \beta\mathbb{E}_{\varepsilon_2}\log\frac{p(z_2|x_2)}{r_2(z_2)}\}. \tag{5.10}$$

Three Monte Carlo sampling procedures are used are used here to approximate the integrals. $p(z_1|x_1), p(z_2|x_2)$ are all learned from neural networks. Note that the first term in Eq. (5.10) is the cross-entropy between y and z. Thus, we can use a deep neural network with a softmax layer as output to calculate the class probabilities and the cross-entropy loss.

5.1.4 Generalize to multiple modalities

The proposed deep multimodal information bottleneck framework can be easily generalized to settings with more than 2 modalities by adding corresponding information constraint terms. Given v modalities $\{X_1,X_2,...,X_v\}$, the formulation of the proposed method is

$$\max_{Z,Z_1,Z_2,...Z_v} I(Y,Z) - \sum_{i}^{v}\alpha_i I(X_i,Z_i), \tag{5.11}$$

where Z_i is the latent representation of X_1. α_i is the regularization parameter to regularize the mutual information between X_i and Z_i. Following the procedures in Section 5.1.3, the final loss for Eq. (5.11) is

$$\max \frac{1}{N} \sum^{N} \{\mathbb{E}_{\varepsilon} \mathbb{E}_{\varepsilon_1} \mathbb{E}_{\varepsilon_2} ... \mathbb{E}_{\varepsilon_v} \log q(y|z) - \alpha_i \mathbb{E}_{\varepsilon_i} \log \frac{p(z_i|x_i)}{r_i(z_i)}\}, \tag{5.12}$$

where $\varepsilon, \varepsilon_1, \varepsilon_2, ...\varepsilon_v \sim \mathcal{N}(0, I)$. $r_i(z_i)$ are assumed as $r_i(z_i) \sim \mathcal{N}(0, I)$. Each $p(z_i|x_i)$ are Gaussian with μ and Σ leaned from a deep neural network.

5.2 Experiments

In this section, we present the experimental results on synthetic and real-world datasets. The baseline algorithms used for comparison include linear CCA [27], DCCA [9], DCCAE [119], and the fully-connected deep neural network (DNN), which uses two fully-connected neural networks to directly extract latent representations Z_1, Z_2 and then uses a deep neural network to fuse Z_1 and Z_2 to make prediction. One intuitive baseline for multimodal learning is to concatenate the features from all the modalities and treat the concatenated features as one modality. In our experiments, we use single-modal information bottleneck [7] as the model for this baseline and denote this baseline as *singlemodal12*. We also provide the results of single-modal learning using information bottleneck [7] for each modality, and use *singlemodal1, singlemodal2* to denote the baselines using the first modality and the second modality. We denote the proposed method as *deep IB*.

5.2.1 Synthetic datasets

. The data are synthesized in the following way. First, we sample $2n$ points from two Gaussian distributions, i.e., $\mathcal{N}(0.5\mathbf{e}, I)$ and $\mathcal{N}(-0.5\mathbf{e}, I)$ to form Z. Samples from each distribution form one class. Each class has n data points. Then, we directly use Z to generate $X1$ and $X2$ by setting $X_i = f(D) + \text{noise}$, where $D = [Z, \text{extra-features}]$ with $i \in \{1,2\}$ and f is a nonlinear function. Extra-features here are used to distort the classification and are sampled from another two Gaussian distributions, i.e., $\mathcal{N}(\mathbf{e}, I)$ and $\mathcal{N}(-\mathbf{e}, I)$. We sample m data from the first Gaussian distribution and $2n-m$ samples from the second Gaussian distribution. We concatenate the extra-features to the useful features to distort the classification. Extra-features are illustrated in Figure 5.1. In Figure 5.1, the row represents the samples and the column represents the features. Extra-features have different class property compared with the useful features. In all the synthetic data experiments, we set $m = 2n/3$ for the first modality and $m = 2n/6$ for the second modality. Extra-features widely exist in multimodal learning scenario. For example, when we collect all the genetic data from people with a gene-related disease and healthy people, the genetic data contain not only information to classify the disease, but also gender information. The features that describe the gender information are extra-features. The effect of those features needs to be eliminated in the classification process. The noise is sampled from $\mathcal{N}(0, t * I)$, where t denotes the noise level and is changed in the experiments to test the algorithms' ability to eliminate the effect from noise. f is $tanh(tanh(D)) + 0.1$ for the first modality and $sigmoid(D) - 0.5$ for the second modality.

5.2.1.1 Setting 1

. In the first setting, we change the noise level t and compare our model with other baselines. t is the relative noise level which is calculated as $t = a \times \max(abs(X_i))$ for each modality, where

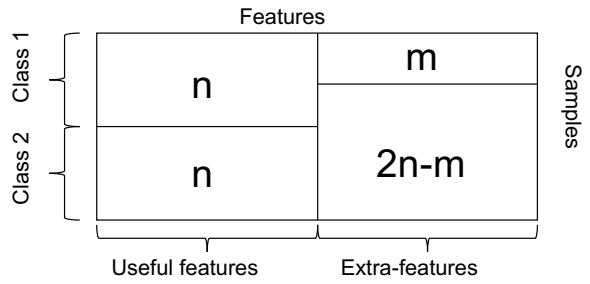


Figure 5.1: **Illustration of extra-features in the synthetic data experiments.** Extra-features have different class property with the useful features.

Noise level (a)	0.2	0.4	0.6
singlemodal1	0.070 ± 0.016	0.111 ± 0.020	0.135 ± 0.025
singlemodal2	0.132 ± 0.025	0.205 ± 0.040	0.253 ± 0.025
singlemodal12	0.064 ± 0.012	0.083 ± 0.012	0.125 ± 0.011
linear CCA	0.064 ± 0.027	0.109 ± 0.023	0.143 ± 0.035
DCCA	0.065 ± 0.024	0.096 ± 0.023	0.132 ± 0.029
DCCAE	0.075 ± 0.017	0.098 ± 0.008	0.139 ± 0.044
DNN	0.061 ± 0.004	0.094 ± 0.012	0.128 ± 0.025
deep IB	0.059 ± 0.016	0.073 ± 0.011	0.122 ± 0.019
Noise level (a)	0.8	1.0	1.2
singlemodal1	0.166 ± 0.025	0.192 ± 0.030	0.206 ± 0.040
singlemodal2	0.283 ± 0.031	0.313 ± 0.032	0.338 ± 0.035
singlemodal12	0.154 ± 0.031	0.164 ± 0.023	0.181 ± 0.037
linear CCA	0.165 ± 0.038	0.194 ± 0.041	0.209 ± 0.043
DCCA	0.154 ± 0.033	0.173 ± 0.034	0.198 ± 0.043
DCCAE	0.165 ± 0.026	0.182 ± 0.028	0.203 ± 0.041
DNN	0.156 ± 0.024	0.164 ± 0.037	0.191 ± 0.031
deep IB	0.139 ± 0.017	0.158 ± 0.017	0.171 ± 0.026

Table 5.1: **Average errors of all methods under different noise levels.**

abs means the absolute value. We set a to be $\{0.2 : 0.2 : 1.2\}$. The sample size per class is set to be 500. The useful feature dimension is 20, and the extra-feature dimension is 5. α and β are tuned in $[1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2]$. For the subnetworks that extract features from X_1 and X_2 for all the deep models including DCCA, DCCAE, we tune the number of layers in $[3, 4, 5]$ and the node number for each layer is tuned in $[256, 512, 1024]$. The activation function is ReLU. For the subnetworks that fuse the extracted features from all modalities, we tune

the number of layers in $[1,2,3]$ and the node number is tuned in $[128,256,512]$. The activation function is ReLU. For all the experiments, we use 80% data as training and the rest as testing and repeat the experiments for 5 times. We report the average errors for all methods in Table 5.1. From Table 5.1, we see when noise increases, the performance becomes worse for all the methods. Single-modal methods are all worse than supervised multimodal methods. Simple concatenation of two modalities is not as good as deep IB. Compared with CCA-based method, we see supervision information improves the performance a lot. DNN is a challenging baseline as shown in the results. DNN has a similar network structure with deep IB. The difference between DNN and deep IB is that DNN tries to extract latent features by directly maximizing the cross-entropy between the outputs of the network and labels, while deep IB not only maximizes the cross-entropy between the outputs of the network and labels, but also constrains the model complexity by reducing the information between Z_1 and X_1, and between Z_2 and X_2. Therefore, the generalization performance of deep IB is better than DNN.

5.2.1.2 Setting 2

. In the second setting, we vary the sample size per class to see how the performance changes. The noise level a is set to be 1.0. The useful features dimension is 20, and the extra-feature dimension is set to be 5. The models are tuned in the same way as the first setting. We report the errors for all methods in Table 5.2. From the table, we see increasing the sample size improves the performance for all methods. We observe some similar patterns with that of Setting 1. For example, deep IB results are better than all other single-modal methods results. CCA-based methods are not as good as supervised methods. One specific observation is that when the sample size is large enough, i.e., greater than 1100, DNN's performance is better than deep IB. That is because deep IB has the assumption to reduce the model complexity. When the sample size is large enough, deep IB

Sample per class	300	500	700
singlemodal1	0.178 ± 0.035	0.192 ± 0.030	0.175 ± 0.022
singlemodal2	0.333 ± 0.043	0.313 ± 0.032	0.319 ± 0.032
singlemodal12	0.230 ± 0.080	0.173 ± 0.023	0.164 ± 0.023
linear CCA	0.163 ± 0.030	0.194 ± 0.041	0.166 ± 0.038
DCCA	0.165 ± 0.013	0.173 ± 0.033	0.158 ± 0.026
DCCAE	0.169 ± 0.008	0.182 ± 0.028	0.154 ± 0.033
DNN	0.173 ± 0.024	0.164 ± 0.032	0.161 ± 0.032
deep IB	0.162 ± 0.015	0.158 ± 0.017	0.143 ± 0.021
Sample per class	900	1100	1300
singlemodal1	0.179 ± 0.015	0.192 ± 0.014	0.183 ± 0.005
singlemodal2	0.326 ± 0.021	0.317 ± 0.010	0.316 ± 0.024
singlemodal12	0.174 ± 0.021	0.180 ± 0.008	0.185 ± 0.011
linear CCA	0.159 ± 0.011	0.173 ± 0.011	0.170 ± 0.023
DCCA	0.151 ± 0.018	0.165 ± 0.013	0.155 ± 0.013
DCCAE	0.154 ± 0.003	0.178 ± 0.017	0.160 ± 0.008
DNN	0.146 ± 0.016	0.143 ± 0.004	0.139 ± 0.008
deep IB	0.139 ± 0.009	0.143 ± 0.007	0.143 ± 0.006

Table 5.2: **Average errors of all methods under different sample sizes.**

underfits the data, while DNN has no assumption. Therefore, when the sample size is large enough, direct using DNN delivers the highest accuracy.

5.2.1.3 Setting 3

. In the third setting, we change the extra-feature dimension to see how the extra-feature dimension affects the results. In this setting, the sample size per class is set to be 500. The noise level a is 1.0. The useful feature dimension is fixed as 20. The errors are shown in Table 5.3. From Table 5.3, we see deep IB outperforms all the other methods with any extra-feature dimension. When the extra-feature dimension increases, the data contain more irrelevant information, which makes the classification to be distorted. We see when the extra-feature dimension increases, the error of DNN, CCA-based methods increase a lot. However, for IB based methods including the single-modal baselines, the errors are stable when the extra-feature dimension is larger or equal to 25.

Extra-feature dim	5	15	25
singlemodal1	0.192 ± 0.030	0.194 ± 0.036	0.199 ± 0.020
singlemodal2	0.313 ± 0.032	0.333 ± 0.024	0.342 ± 0.019
singlemodal12	0.164 ± 0.023	0.181 ± 0.027	0.194 ± 0.024
linear CCA	0.194 ± 0.041	0.192 ± 0.038	0.225 ± 0.030
DCCA	0.173 ± 0.033	0.179 ± 0.040	0.187 ± 0.016
DCCAE	0.182 ± 0.028	0.185 ± 0.037	0.195 ± 0.020
DNN	0.164 ± 0.037	0.197 ± 0.020	0.201 ± 0.022
deep IB	0.158 ± 0.017	0.174 ± 0.053	0.175 ± 0.013
Extra-feature dim	35	45	55
singlemodal1	0.198 ± 0.042	0.194 ± 0.019	0.193 ± 0.016
singlemodal2	0.327 ± 0.020	0.334 ± 0.026	0.332 ± 0.021
singlemodal12	0.189 ± 0.041	0.191 ± 0.026	0.189 ± 0.017
linear CCA	0.205 ± 0.047	0.255 ± 0.027	0.286 ± 0.012
DCCA	0.181 ± 0.018	0.183 ± 0.026	0.201 ± 0.040
DCCAE	0.193 ± 0.023	0.215 ± 0.026	0.221 ± 0.032
DNN	0.216 ± 0.025	0.219 ± 0.015	0.225 ± 0.032
deep IB	0.174 ± 0.035	0.173 ± 0.019	0.176 ± 0.023

Table 5.3: **Average errors of all methods under different extra-feature dimensions.**

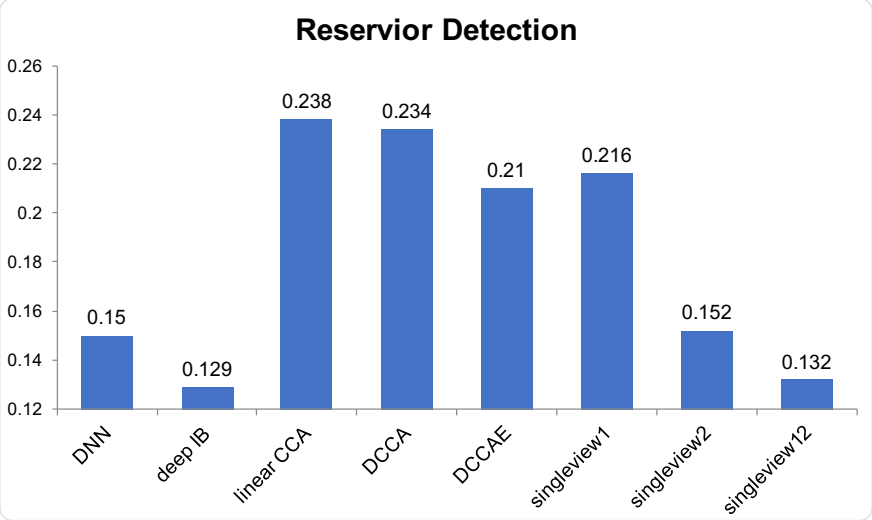


Figure 5.2: **Average error for reservoir detection task.**

From the results, we conclude that IB-based methods are more robust to extra-features.

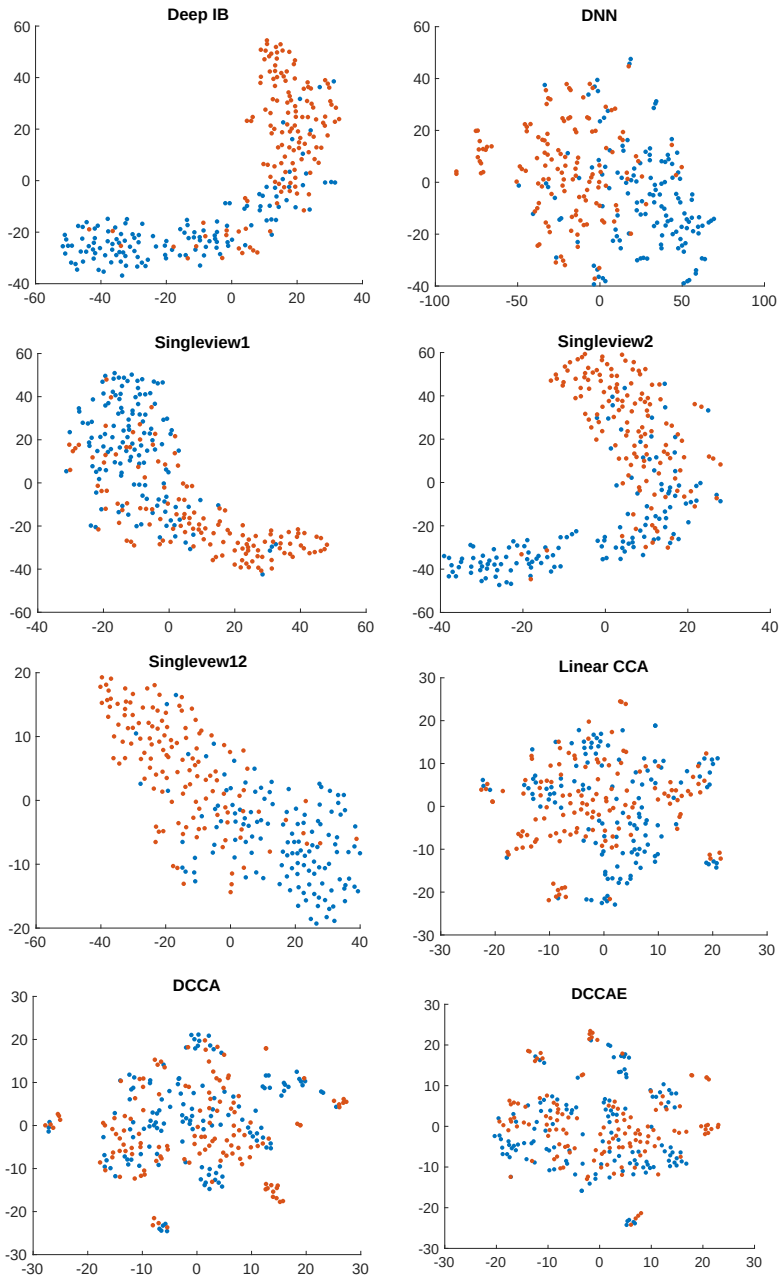


Figure 5.3: **T-Distributed Stochastic Neighbor Embedding for the final joint representations for reservoirs detection models.** Blue dots are natural lakes and red dots are reservoirs.

5.2.2 Case study: reservoir detection

. In this case study, we compare all the models on a reservoir detection dataset. Reservoirs for this dataset are sampled with ArcMap 10.3.1 by joining dam features from the US Army Corps' National Inventory of Dams with lake polygons over 4 hectares from the LAGOS database [100].

For comparison, we also select a proportional number of natural lakes from the major river watershed that each reservoir is located in. The sample size for this dataset is 1327 with 660 natural lakes and 667 reservoirs. There are two modalities available in this dataset. The first one is the boundary of the lakes. Boundary features of each lake and reservoir are exported using ArcMap. Each boundary file is a 224×224 image. To deal with the boundary data, we first use VGG16 to extract features. We use the last fully-connected layer's output as the features. The dimension is 4096. Since the sample size is not large, we use PCA to reduce the feature dimension by keeping the top 1% singular values. The reduced feature dimension is 75. The second modality is the features extracted from Google Earth. The features include the area of the lakes, shape length, classes of the general types of parent material of soil on the surface, classes of landforms, NED-derived mTPI ranging from negative (valleys) values to positive (ridges) values, NED-derived CHILI index ranging from 0 (very cool) to 225 (very warm). In total, there are 21 features. We split the data into training and testing as the synthetic data experiments and report the average error in Figure 5.2. From the figure, we see deep IB outperforms all other methods. In Figure 5.3, we also qualitatively show the final joint representations learned by all methods with t-Distributed Stochastic Neighbor Embedding [74]. The final joint representation is the output of the layer that is connected with the final linear classifier. For example, for deep IB, DNN and the single-modal methods, the final joint representations are the outputs of the layer before the last layer. For the CCA-based methods, the final learned representations are the projected representations from the first modality. In Figure 5.3, blue dots are natural lakes and red dots are reservoirs. We see that the separation qualities are consistent with the performance in Figure 5.2.2.

ADNI2	NC	MCI	Total	p-value
Number	50	112	163	-
Age	69.36 ± 15.40	71.68 ± 9.93	70.96 ± 11.89	0.0016
Sex	22M/28F	71M/41F	93M/69F	0.0040
NACC	HC	MCI	Total	p-value
Number	329	57	386	-
Age	60.96 ± 8.96	73.60 ± 7.93	63.82 ± 9.73	0.0100
Sex	107M/222F	38M/19F	145M/241F	0.0046

Table 5.4: **Demographic information for the two cohorts (ADNI2 and NACC).** The p-values for 695 the difference between ADNI2 and NACC are 0.023 for sex and 3.88e-23 for age. The last column 696 is the p-value for the difference between MCI and NC.

	ADNI2	NACC
b value	1000 s/mm^2	1300 s/mm^2
Number of b0 images	5	8
Number of diffusion weighted images	42	40
T1 MRI voxel size	$1.0156 \times 1.0156 \times 1.2$ mm^3	$1.0 \times 1.9 \times 1.2$mm^3
T1 MRI TR	6.98 ms	8.16 ms
T1 MRI TE	2.85 ms	3.18 ms
T1 MRI Image dimension	$256 \times 256 \times 196$	$256 \times 256 \times 156$
dMRI voxel size	$2.7 \times 2.7 \times 2.7$mm^3	$0.94 \times 0.94 \times 2.9$mm^3
dMRI TR	9050ms	8000ms
dMRI TE	Minimum	81.8 ms
dMRI Image dimension	$128 \times 128 \times 59$	$256 \times 256 \times 52$

Table 5.5: **Parameters for dMRI and T1 MRI data for ADNI2 and NACC.**

5.2.3 Case study: Alzheimer's disease classification

5.2.3.1 Data Preprocessing

The data we used is the union of ADNI2 and NACC dataset. There are two classes, i.e., normal control (NC), mild cognitive impairment (MCI). NACC has 329 NC and 57 MCI. ADNI2 has 50 NC and 112 MCI. Demographic characteristics of the two datasets is summarized in Table 5.4. dMRI and T1w MRI data for each subject was analyzed. Table 5.5 summarizes the key data collection parameters for the two cohorts.

Two types of feature variables were extracted in this study. The first type is from the gray matter

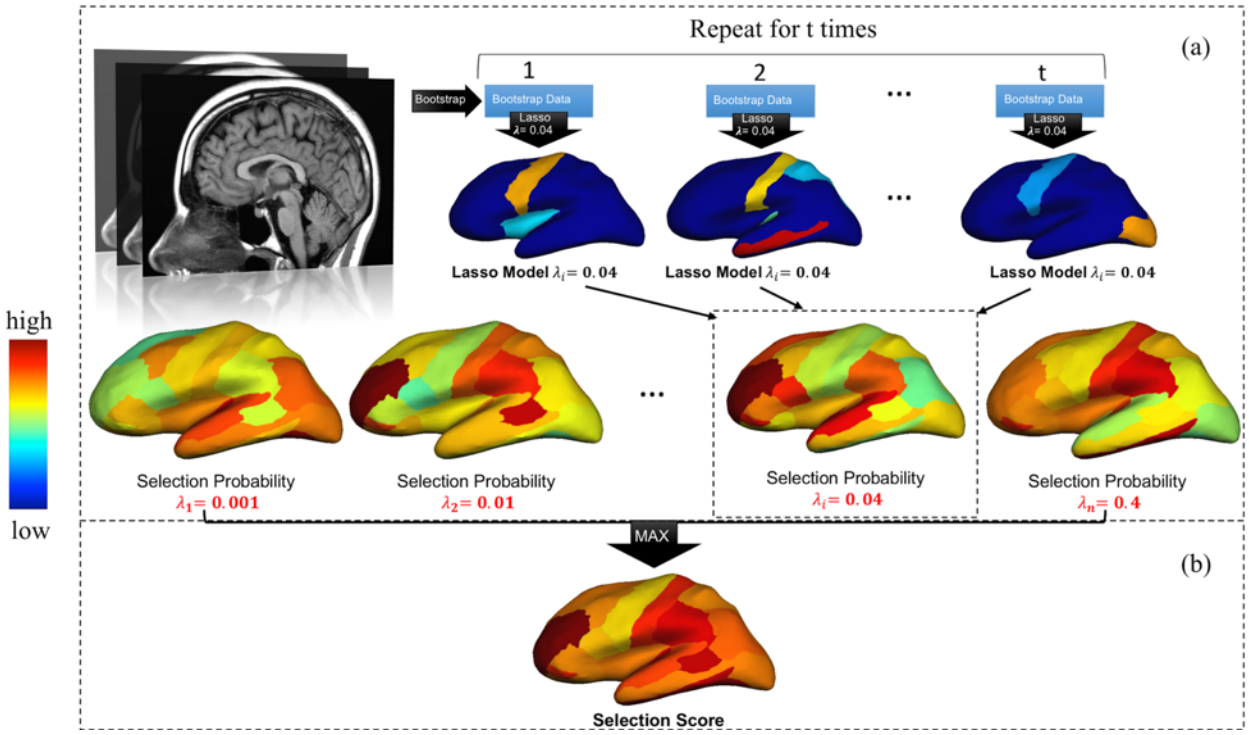


Figure 5.4: **The pipeline of computing the stability score.** The warmer color indicates a higher probability of selection. (a) Calculating the selection probability using different regularization parameters. (b) Illustration of using selection probability to calculate stability score.

using T1w MRI. FreeSurfer was used to extract 136 measurements including cortical volume and thickness for 68 brain ROIs based on Desikan-Killiany atlas [33]. The second type is from dMRI-derived structural connectome or network. The brain structural connectome was constructed using PICo [84], a whole-brain probabilistic tractography algorithm and 113 ROIs defined on the Harvard Oxford Cortical and subcortical Probabilistic Atlas [33, 40]. The details of computing the brain network can be referred to [134]. Each subject's network has a dimension of 113x113, with 6,328 distinct edges connecting 113 brain ROIs (the edges are not directional and thus the network is symmetric).

For both the ADNI2 and NACC cohorts, the number of subjects is limited, especially when we need subjects to have both valid T1 MRI and dMRI available. When performing classification modeling, the dimension of feature variables will be much larger than the sample size for both

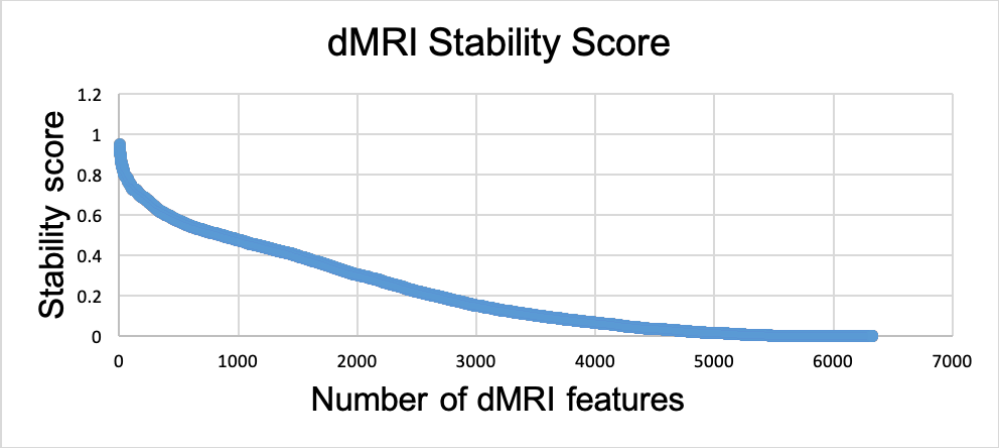


Figure 5.5: **The distribution of the stability scores for the dMRI features.**

	ROI1	ROI2
1	Right Parahippocampal Gyrus,	Right Heschl"S Gyrus Posterior Division
2	Right Amygdala	Left Cerebellum
3	Right Inferior Temporal Gyrus, Temporooccipital Part	Right Supramarginal Gyrus, Anterior Division
4	Brainstem	Left Insular Cortex
5	Left Insular Cortex	Left Frontal Opercular Cortex
6	Right Superior Temporal Gyrus, Posterior Division	Left Supramarginal Gyrus, Posterior Division
7	Left Caudate	Left Pallidum
8	Left Frontal Pole	Left Frontal Opercular Cortex
9	Left Inferior Temporal Gyrus, Temporooccipital Part	Right Supramarginal Gyrus, Posterior Division
10	Right Superior Parietal Lobule	Left Planum Temporale

Table 5.6: **Top 10 dMRI feature variables identified.**

dMRI. This would lead to the "curse of dimensionality" problem where our classification models overfit training data and deliver poor generalization power. Since not all feature variables are related to the AD progression, we perform a feature variable selection procedure that ranks all the variables according to their relevance to the classification problem, and include only those feature variables in our models. We use the powerful stability selection method and use stability score as our criterion for relevance. We select sparse logistic regression in Chapter 3 as the sparse model to select the features. Given a set of regularization parameters $\{\lambda_1, \lambda_2, ...\lambda_n\}$, for each regularization

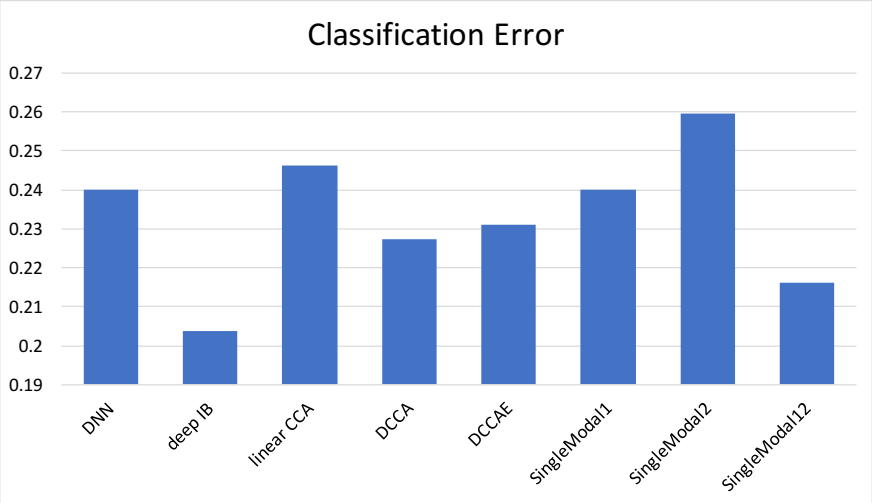


Figure 5.6: **Average error for classifying MCI with AD.**

parameter λ we obtain a set of feature variables S^{λ} that contribute to the final classification model in the corresponding sparse model. Stability selection is a variable selection method based on subsampling in combination with high-dimensional sparse learning algorithms. Instead of selecting one model, stability selection perturbs the data (e.g., by subsampling) many times, and we identify consistent feature variables that are included in the model, under different values of the parameter λ, across bootstrap datasets [75]. Intuitively, feature variables selected in this way are more consistently relevant to the target problem than feature variables selected only by sparse algorithms. Stability selection works as follows: we first randomly select 50% of training samples and apply sparse logistic regression to the selected training samples with regularization parameter λ_i to build a sparse model. Let F denote the whole feature variables set and $f \in F$ denote the index of a particular feature variable in the set. The set of feature variables selected by this model is denoted by:

$$U_{\lambda_i} = \{f : w_{\lambda_i,f} \neq 0\} \tag{5.13}$$

We repeat this procedure for $t = 1000$ times. Selection probability for each feature variable is

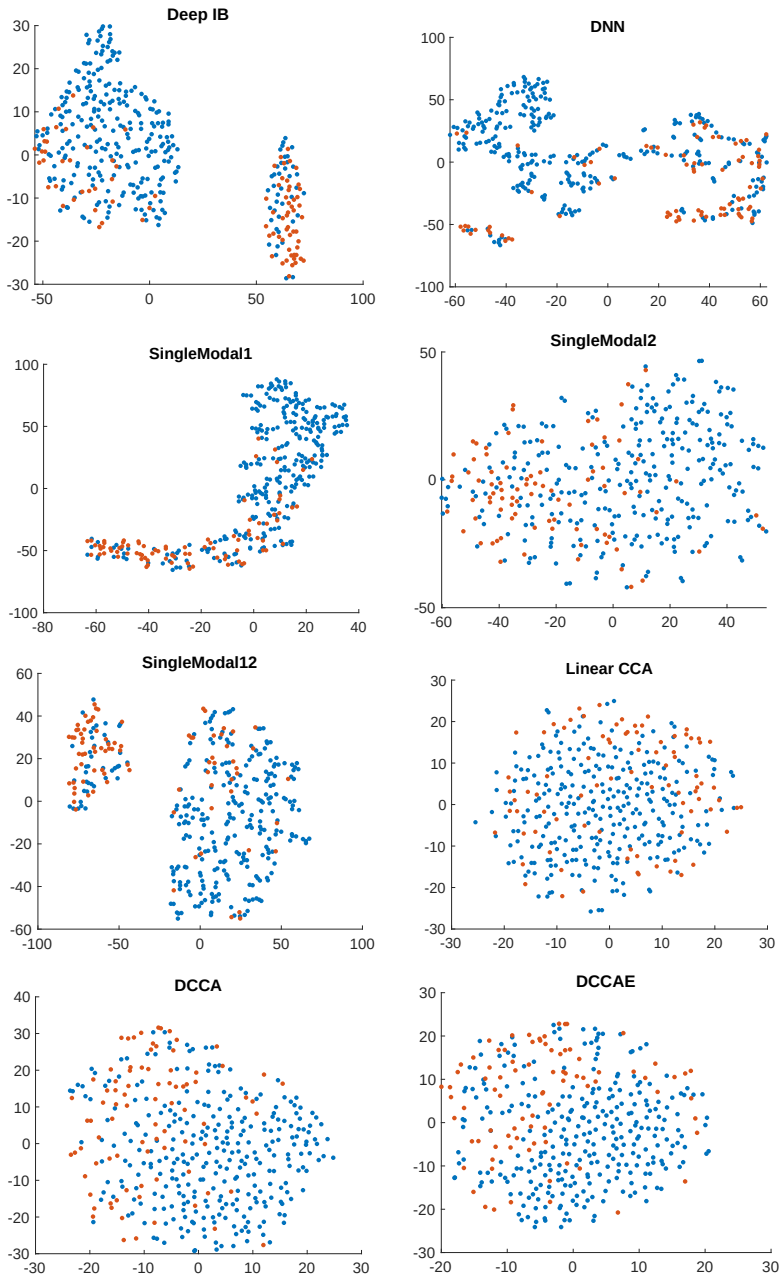


Figure 5.7: **T-Distributed Stochastic Neighbor Embedding for the final joint representations for classifying MCI with NC.** Blue dots are NCs and red dots are MCIs.

calculated as follows:

$$Pr_{f,\lambda_i} = \sum I(f \in U_{\lambda_i})/t \tag{5.14}$$

Dataset	XRMB	MNIST	Wiki
singlemodal1	0.185 ± 0.003	0.075 ± 0.006	0.449 ± 0.024
singlemodal2	0.271 ± 0.003	0.160 ± 0.012	0.337 ± 0.018
singlemodal12	0.179 ± 0.006	0.057 ± 0.009	0.336 ± 0.006
linear CCA	0.358 ± 0.004	0.235 ± 0.006	0.741 ± 0.017
DCCA	0.231 ± 0.006	0.187 ± 0.012	0.478 ± 0.049
DCCAE	0.226 ± 0.005	0.170 ± 0.020	0.499 ± 0.037
DNN	0.168 ± 0.006	0.060 ± 0.056	0.311 ± 0.017
deep IB	0.161 ± 0.005	0.056 ± 0.002	0.298 ± 0.005

Table 5.7: **Average errors for three benchmark datasets.**

where $I(\cdot)$ is the indication function: $I(c) = 1$ when c is true and $I(c) = 0$ when $c$is false. The procedure of calculating selection probability is illustrated in the upper portion of Figure 5.4.

Then we vary the regularization parameter many times and calculate selection probability under these regularization parameters. By these selection probabilities, stability score for feature variable f is calculated as follows:

$$Sc(f) = \max_{\lambda_i}(Pr_{f,\lambda_i}) \tag{5.15}$$

With stability score, we can rank the variables and choose only top k stable variables, or a stability score that is larger than a pre-set threshold. The computation of the stability score is shown in the lower portion of Figure 5.4. After selecting feature variables by stability score, the feature dimension is reduced drastically. We will use the new feature variables set to build our model. Figure 5.5 shows the distribution of the stability scores for T1 MRI features features. select the top 172 features which have the top 30% stability scores as the final features for this dMRI.

We split the data into training and testing datasets with the ratio 9:1 and repeat the experiments for 5 times. The classification error is shown in Figure 5.6 and the TSNE of the hidden representations are in Figure 5.7. We see our method outperforms all other methods.

5.2.4 Other benchmark datasets

In this section, we report the performance on three benchmark datasets. The datasets we used are

- Wisconsin X-Ray Mircro-Beam (XRMB) [119, 122]: the first modality is 273D acoustic inputs, the second modality is 112D articulatory inputs [1].
- MNIST [119]: two modalities are generated from MNIST datasets. The first modality is a random rotation of the original images. The second modality is generated by adding noise to the original images. Both modalities have 784 features [2].
- Wiki [35]: the dataset contains 2866 images-text pairs. Each image is represented by 128D inputs and text is represented by 10D inputs. There are 10 classes in total.

The average errors are shown in Table 5.7. From the table, we see for all the benchmark datasets, the proposed method performs the best among all the methods, which verifies the effectiveness of the proposed method.

5.3 Summary

In this work, we proposed a novel multimodal learning model based on information bottleneck. The model encouraged the latent representation keeping target information as much as possible while containing the information of original features as little as possible to reduce the model complexity. To learn the complicated relationship between modalities and within modalities, we used a deep neural network to learn the latent representation. Since the mutual information terms were intractable, we maximized the lower bound of the formulation instead of directly maximizing it. We

[1] We did not use the whole dataset since some baselines are quite slow. We randomly sampled 50000 data points for training and sampled 6000 points for testing from the first 10 classes.

[2] We did not use the whole dataset since some baselines are very time-consuming. We sampled 5000 data for training and 1000 for testing.

demonstrated experiments on various synthetic and real-world datasets to show the effectiveness of the proposed method.

Chapter 6

Multimodal Learning with Incomplete Modalities

6.1 Methodology

In this section, we first give a brief introduction to knowledge distillation [46]. Then, we introduce our method which leverages knowledge distillation to conduct multimodal learning with supplementary information.

6.1.1 Knowledge Distillation

Knowledge distillation is used to transfer "dark knowledge" from a teacher to a student. To transfer knowledge, the teacher is first trained on a dataset. Denote the trained teacher model as $Te(\phi)$ with ϕ denotes the parameters of the teacher model. Then, the student is trained to mimic the output of the teacher on the training dataset. Given a dataset $D = \{\{X_1, y_1\}, \{X_2, y_2\} \ldots \{X_N, y_N\}\}$ used to train the student, the teacher is first applied on the data and label the data with the logits. We assume there are in total C classes, and the labels are thus given by:

$$z_i = Te(X_i; \phi), \tag{6.1}$$

where $z_i \in \mathbb{R}^{C\times 1}$ is the logits labeled by the teacher model for sample X_i. The student model is then trained with both the true one-hot label $\{y_i, y_2, \ldots, y_N\}$ and the logits $\{z_1, z_2, \ldots, z_N\}$. Suppose the student model is a deep neural network $f(\theta)$ parameterized by θ. It takes X_i as input and outputs a $C \times 1$ vector which is the logit vector. Then, a SoftMax function is added to the logit vector to output the probability of X_i to be classified as C classes. The loss function of training the student network is:

$$\min_{\theta} l = \sum_{i}^{N} l_c(X_i, y_i; \theta) + l_d(X_i, z_i; \theta). \tag{6.2}$$

where l_c is a classification loss with the true one-hot label with the form:

$$l_c(X_i, y_i; \theta) = H(\sigma(f(X_i; \theta)), y_i), \tag{6.3}$$

where H is the negative cross-entropy loss, and $\sigma(x) : \mathbb{R}^C \to \mathbb{R}^C$ is the SoftMax function:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{C} e^{x_k}} \quad \text{for } i = 1, 2, \ldots, C. \tag{6.4}$$

$l_d(X_i, z_i; \theta)$ is the distillation loss. Examples of the distillation loss include negative cross-entropy loss or KL-divergence. Without loss of generality, we adopt KL-divergence as the distillation loss:

$$l_d(X_i, z_i; \theta) = D_{KL}(\sigma_T(f(X_i; \theta); T), \sigma_T(z_i; T)). \tag{6.5}$$

where $\sigma_T(x; T)$ denotes the SoftMax with temperature T:

$$\sigma_T(x; T)_j = \frac{e^{\frac{x_j}{T}}}{\sum_{k=1}^{C} e^{\frac{x_k}{T}}}. \tag{6.6}$$

With temperature T, the output probability is rescaled and smoothed. If temperature T is large, the probability will be more smooth compared with a small temperate T. The output of $\sigma_T(z_i;T)$ is called the "soft label", which is labeled by the teacher model on the sample X_i. It is believe that the "soft labels" contain more information than the one-hot label [46].

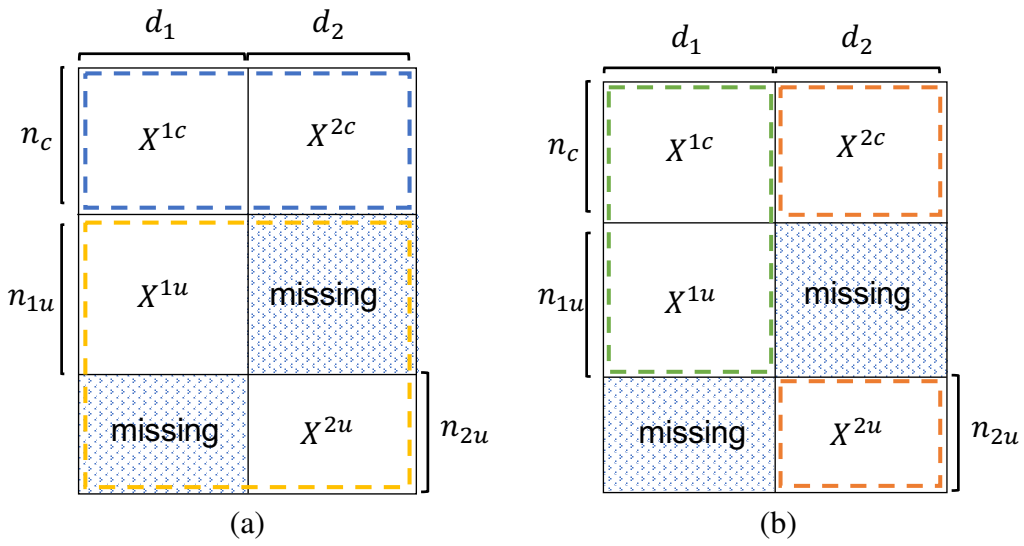


Figure 6.1: **Pattern of the data.** (a) shows the structure of a dataset with two modalities. Samples in the blue dashed-line box have complete modalities and samples in the yellow dashed-line box only have one modality available. (b) Illustration of samples used to train teacher models for two-modal learning. Samples in the green dashed-line box are used to train the first teacher and samples in the orange dashed-line box are used to train the second teacher.

6.1.2 Multimodal learning with missing modalities

For multimodal learning, it is rather common that some samples do not have complete modalities. Below we first start our discussions on two modalities and then generalize our method to multiple modalities.

Given two modalities $\{X^1 \in \mathbb{R}^{n_1 \times d_1}, X^2 \in \mathbb{R}^{n_2 \times d_2}\}$ with labels, we denote the samples have complete modalities as $\{X^{1c} \in \mathbb{R}^{n_c \times d_1}, X^{2c} \in \mathbb{R}^{n_c \times d_2}, y^c \in \mathbb{R}^{n_c}\}$. Samples only have the first modality are denoted as $\{X^{1u} \in \mathbb{R}^{n_{1u} \times d_1}, y^{1u} \in \mathbb{R}^{n_{1u}}\}$ and samples only have the second modality are de-

noted as $\{X^{2u} \in \mathbb{R}^{n_{2u} \times d_2}, y^{2u} \in \mathbb{R}^{n_{2u}}\}$ with $n_1 = n_c + n_{1u}$ and $n_2 = n_c + n_{2u}$. In Figure 6.1, (a) shows the structure of the data. Samples in the blue dashed-line box are these with complete modalities and samples in yellow dashed-line box only have one modality available. To utilize all the samples, we first train two single modal models with all the available data including the samples with missing modalities. These two models are then acting as teacher models in our framework. We assume that the two teachers are two neural networks $g_1(\phi_1)$ and $g_2(\phi_2)$ with parameters ϕ_1 and ϕ_2. $g_1(\phi_1)$ takes the samples from $[X^{1c}, X^{1u}]$ as input and outputs the logits and $g_1(\phi_1)$ takes the samples from $[X^{2c}, X^{2u}]$ as input and output the logits. The two teachers are trained by minimizing the following loss functions:

$$
\begin{aligned}
Te_1(\phi_1) &= \min_{\phi_1} \sum_i^{n_1} H(\sigma(g_1(X_i^1; \phi_1)), y_i), \\
Te_2(\phi_2) &= \min_{\phi_2} \sum_i^{n_2} H(\sigma(g_2(X_i^2; \phi_2)), y_i)
\end{aligned} \tag{6.7}
$$

Then, we use the two teachers to label the samples in $\{X^{1c}, X^{2c}\}$. The logits for the i-th sample are:

$$
z_i^1 = Te_1(X_i^{1c}; \phi_1), \quad z_i^2 = Te_2(X_i^{2c}; \phi_2), \tag{6.8}
$$

where z_i^j denotes the logit labeled by teacher j for the i-th sample.

In order to fuse the supplementary information from different modalities, we train a student model with multimodal DNN (M-DNN) [87]. The M-DNN for two modalities contains two branches. Each branch takes one modality as input and is followed with several nonlinear fully-connected layers. The outputs of all the branches are concatenated to form a joint representation. Then, the joint representation is connected to a linear layer to output the logits z. The reason we

use such a model as the student model is that the joint representation learned with this model contains the supplementary information of the two modalities. If we train the M-DNN as the methods in [123], i.e., only use the samples with complete modalities $\{X^{1c}, X^{2c}, y^c\}$ to train the model, the sample size is limited to be n_c. If n_c is very small compared with n_1 and n_2, a large amount of useful information is discarded and the samples for training the model is not enough. Thus, we propose to train the M-DNN with the information from the two teachers $Te_1(\phi_1)$ and $Te_2(\phi_2)$ to improve the performance as the two teachers are trained on much larger datasets. The final classification performance for each teacher might be not good enough since each teacher only has access to one modality. But the teachers can do the best to learn classifier with these modalities, provide the expertise for these modalities and teach the student with this knowledge. Denote the student network as $f(\theta)$ with θ representing the parameters. The loss function for the proposed method is:

$$\begin{aligned}\min_{\theta} l = \min_{\theta} \sum_{i}^{n_c} & l_c(X_i^1, X_i^2, y_i; \theta) + \alpha l_{d1}(X_i^1, X_i^2, y_i; \theta, Te_1(\phi_1)) \\ & + \beta l_{d2}(X_i^1, X_i^2, y_i; \theta, Te_2(\phi_2)),\end{aligned} \tag{6.9}$$

where $l_c(X_i^1, X_i^2, y_i; \theta)$ is the classification loss as

$$l_c(X_i^1, X_i^2, y_i; \theta) = H(\sigma(f(X_i^1, X_i^2; \theta)), y_i).$$

l_{d1}, l_{d2} are distillation loss, α and β are two tunable parameters to control how much knowledge the student model needs from the teacher models. If the parameter is large, it means the student model needs more knowledge from this teacher than a small regularization parameter. The formulations

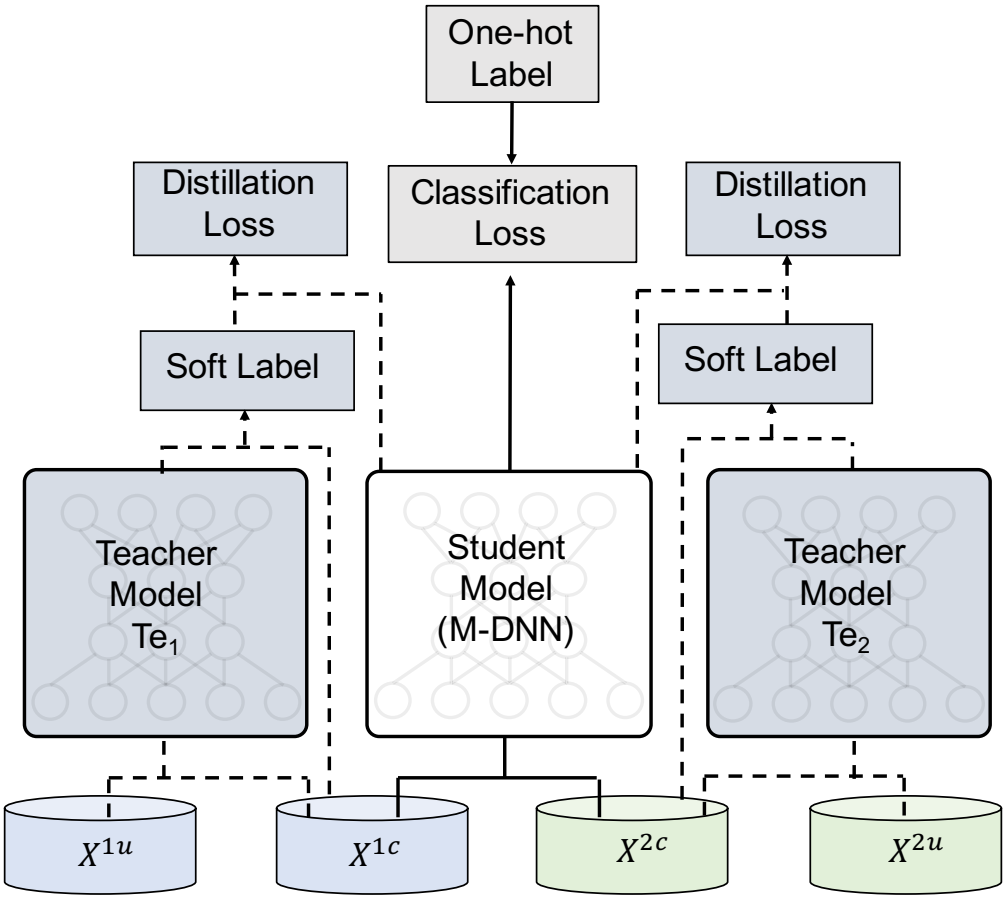


Figure 6.2: **Overview of the proposed teacher-student model.** We first train teacher models with all the available data including the samples have missing modalities. Then, we use the soft labels labeled by the teacher models along with the one-hot label to train the student model.

of l_{d1} and l_{d2} are:

$$l_{d1}(X_i^1, X_i^2, y_i; \theta, Te_1(\phi_1)) = D_{KL}(\sigma_T(f(X_i^1, X_i^2; \theta)), \sigma_T(z_i^1)), \tag{6.10}$$

$$l_{d2}(X_i^1, X_i^2, y_i; \theta, Te_2(\phi_2)) = D_{KL}(\sigma_T(f(X_i^1, X_i^2; \theta)), \sigma_T(z_i^2)). \tag{6.11}$$

Figure 6.2 overviews of the proposed framework.

We would like to highlight the difference between the proposed method with two similar and intuitive methods. The first one is *late fusion*, i.e., fusion at the decision level, which directly combines the labels/logits labeled by the teacher models as the final prediction. Since the teachers

only have partial knowledge of the data, the data labeled by the teachers may not be perfect. Researches have shown for most cases late fusion performs worse than *early fusion*, i.e., feature level fusion [99, 43]. In our proposed method, we not only utilize the labels from the teachers, but also perform early fusion with the M-DNN. So, the performance is expected better than late fusion. Another method is to use the teachers as feature extractors to extract abstract features and then use these abstract features as new sets of features to replace the original inputs to train a multimodal model. The performance of this method may perform well when different modalities only have common or shared information and modality-specific noises. However, when different modalities contain supplementary information, the abstract features extracted by each teacher models may have already lost some useful information as the teacher models are trained on only one modality and are biased. Therefore, its performance is likely to be worse than the proposed method. We will show the performance of these methods in the experiment session.

Mechanism of the proposed method: The underlying mechanism of the proposed approach can be illustrated using gradient analysis. The gradient of the classification loss with respect to the output probability of the k-th class is:

$$\frac{\partial l_c}{\partial p_k} = \sum_i^N (p_{ik} - y_{ik}),$$

where y_{ik} denote the one-hot label of sample i for class k, p_{ik} denote the output probability of sample i for class k. Let L_d denote all the distillation losses, the gradient of the distillation losses with respect to the output probability p_k is:

$$\begin{aligned}\frac{\partial L_d}{\partial p_k} = \frac{\partial}{\partial p_k}(\alpha \sum_i^N D_{KL}(\sigma_T(z_i), \sigma_T(z_i^1)) \\ + \beta \sum_i^N D_{KL}(\sigma_T(z_i), \sigma_T(z_i^2)))\end{aligned}$$

$$= \alpha \sum_i^N (\log p_{ik} - \log q_{ik}^1) + \beta \sum_i^N (\log p_{ik} - \log q_{ik}^2) \tag{6.12}$$

$$\approx \alpha \sum_i^N (p_{ik} - q_{ik}^1) + \beta \sum_i^N (p_{ik} - q_{ik}^2), \tag{6.13}$$

where q_{ik}^m is the soft label produced by teacher m for sample i at class k with $m = 1,2$. We use $log(1+x) \approx x$ to get (6.13) from (6.12). The gradient of the total loss with respect to p_k is:

$$\begin{aligned}
\frac{\partial l}{\partial p_k} &= \sum_i^N ((p_{ik} - y_{ik}) + \alpha(p_{ik} - q_{ik}^1) + \beta(p_{ik} - q_{ik}^2)) \\
&= \sum_i^N (1 + \alpha \frac{p_{ik} - q_{ik}^1}{p_{ik} - y_{ik}} + \beta \frac{p_{ik} - q_{ik}^1}{p_{ik} - y_{ik}})(p_{ik} - y_{ik}) \qquad (6.14) \\
&= \sum_i^N w_{ik}(p_{ik} - y_{ik}), \qquad (6.15)
\end{aligned}$$

where $w_{ik} = (1 + \alpha(p_{ik} - q_{ik}^1)/(p_{ik} - y_{ik}) + \beta(p_{ik} - q_{ik}^1)(p_{ik} - y_{ik}))$. Eq. (6.15) indicates the samples are reweighted by w_{ik}. w_{ik} is determined by the soft labels and the confidence of the soft labels. If both teachers labeled the sample correctly and the confidence p_{ik} for the correct label is high, the weight w_{ik} is around $(1 + \alpha + \beta)$ for this sample. If only one teacher labeled the sample correctly and the confidence is high, the weight is $(1 + \alpha)$ or $(1 + \beta)$, which is smaller than the samples that are correctly labeled by both teachers with high confidence. If the teachers both make mistakes or if they labeled correctly but with very low confidence, the weight is lower than the aforementioned two cases. So, the proposed method reweights the samples with the teachers' labels and the confidence of the teachers and assign higher weights for the samples that are correctly labeled by the teachers with high confidence.

Generalize to multiple modalities: Given m modalities $X^1 \in \mathbb{R}^{n_1 \times d_1}$, $X^2 \in \mathbb{R}^{n_2 \times d_2}$, ... $X^m \in \mathbb{R}^{n_m \times d_m}$, the dataset could be divided into n parts: (1) samples with complete modalities $X^{ic} \in \mathbb{R}^{n_c \times d_i}$ with $i = \{1,2,...m\}$; (2) samples with one modality available $X^{iu} \in \mathbb{R}^{n_{ui} \times d_i}$ with $i =$\{1,

2, ..., m}; (3) samples with two modalities available $X^{ku\{ij\}} \in \mathbb{R}^{n_{u\{ij\}} \times d_k}$ with $i, j = \{1, 2, ...n\}$ and $k = \{i, j\}$. $X^{ku\{ij\}}$ is the k-th modality for the subset that samples contains i-th and j-th modality; ... (n) samples with $n-1$ modalities available $X^{ku_{\{M\backslash i\}}} \in \mathbb{R}^{n_{\{M\backslash i\}} \times d_k}$ with $i = \{1, 2, ..., m\}$. We use $\{M\}$ denote the set of the index for all m modalities, i.e., $\{M\} = \{1, 2, ..., m\}$. $\{M\backslash i\}$ denotes the set without index i. k is an index taken from the set $\{M\backslash i\}$. $X^{ku_{\{M\backslash i\}}}$ is the k-th modality for the subset in which samples contain $\{M\backslash i\}$ modalities. We train the teacher models in a hierarchical manner. First, we train teacher models on each modality separately and obtain Te_i with $i = \{1, 2, \ldots, m\}$. Then, we use these models to teach the teacher models trained with two modalities and obtained teacher model Te_{ij} with $i, j = \{1, 2, \ldots, m\}$. Next, we use all the Te_{ij} to teach the teacher models trained with three modality and so forth. Finally, we obtain all the teachers hierarchically. Denote the teachers trained with h modalities as the h-level teachers. $\{C_h\}$ is the set that composed by all the combination of h indexes sampled from set M. The size of $\{C_h\}$ is $\binom{m}{h}$. For example, if $\{M\}$=$\{1,2,3,4\}$,$\{C_2\} = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}\}$ and $\{C_3\} = \{\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}\}$. H-level teacher models are trained on the modalities indexed by the elements in $\{C_h\}$. For the above example, there are four 3-level teachers, i.e., a teacher trained with the modalities 1,2,3, a teacher trained with the modality 1,2,4, a teacher trained with modalities 1,3,4 and a teacher trained with modalities 2,3,4. Denote the model of the t-th teacher from the h-level teachers by $Te_{C_{ht}}(\phi_{ht})$ with ϕ_{ht} denoting the network parameters and C_{ht} denoting the t-th element of set $\{C_h\}$. For the above example, $C_{23} = \{1,4\}$. $Te_{C_{ht}}(\phi_{ht})$ is trained by minimizing the following loss function:

$$\min_{\phi_{ht}} l_{C_{ht}} = \min_{\phi_{ht}} \sum_{i}^{N_{C_{ht}}} l_c(\{X_i^{ku_{C_{ht}}}\}_{k=C_{ht}}, y_i^{u_{C_{ht}}}; \phi_{ht}) + \sum_{i}^{N_{C_{ht}}} \sum_{j}^{|C_{h-1}|} \alpha_j l_d(\{X_i^{ku_{C_{ht}}}\}_{k=C_{ht}}; Te_{C_{(h-1)j}}), \tag{6.16}$$

where $|C_{h-1}|$ is the size of set C_{h-1} and $N_{C_{ht}}$ is the size of sample having modalities indexed by C_{ht}. After obtaining all teachers, we train the final student model with all the teachers.

One potential issue is that if we have a lot of modalities, the number of teacher models can be very lager. For m modalities, the complete number of teacher models is $2^m - 2$. As such, we cannot build all the teachers to train the student model due to the computational cost. As a solution, we propose to prune the teachers to improve the scalability of the proposed framework. A simple pruning strategy is to select a subset of teachers to train the student model. Basically, after first-level teachers are trained, i.e., single-modal teachers. We only select the teachers that have high performance to train the second level teachers. The modalities that have bad performance are also discarded when building the second level teachers. We build teachers at all other levels in the same way. Finally, all the remaining teachers are used to teach a student model build with m modalities. This pruning method drastically reduces the number of teachers and make the proposed method scalable. For example, for a dataset with five modalities, if in the first level we eliminate two teachers and in the second level we eliminate one teacher, the total teacher number is reduced to five. We demonstrate experiment on synthetic data to show the process of pruning and verify its effectiveness. Here, we use figures to illustrate the pruning procedure. Suppose we have three modalities. If we use all the teachers to train the student. The total number of teachers need to be trained are 6 (see Figure 6.3). Denote the teachers as Te_1, Te_2, Te_3, Te_{12}, Te_{13}, Te_{23}. If the performance of Te_3 is relative low compared with Te_1 and Te_2, we remove Te_3. In the meanwhile,

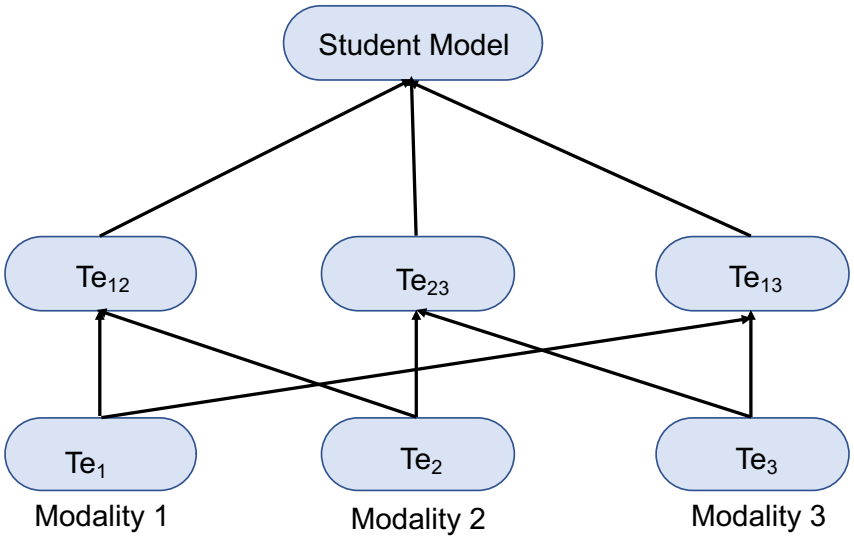


Figure 6.3: **Total teachers need to be trained with three modalities.**

we remove Te_{13} and Te_{23} since both Te_{13} and Te_{23} need the teaching of Te_3. The pruning procedure is shown in Figure 6.4. If all the first-level teachers performances are good but Te_{13}'s performance is relative low compared with Te_{12} and Te_{23}, we remove Te_{13}. We do not need to remove other teachers since there is no high-level teachers.

6.2 Experiment

In this section, we validate the proposed method and baselines on both synthetic and real datasets. The baselines included are (1) Te_i: the i-th teacher model (we use DNN as the teacher model in all the experiments[1]), (2) M-DNN: multimodal DNN trained only with the complete samples, (3) T-DNN: first using teacher models to extract abstract features and then training a DNN with the concatenation of these abstract features as input, (4) CAS-AE [111]: first using cascade residue autoencoder to impute the missing modalities and then training multimodal DNN with the origi-

[1]Other models could also be used as teacher models. The reason we use DNN as the teacher model in our work is that DNN model's performance is relatively high compared with other commonly used classifiers. Ensemble models also have high performance. But DNN model could generate soft labels more easily than ensemble models.

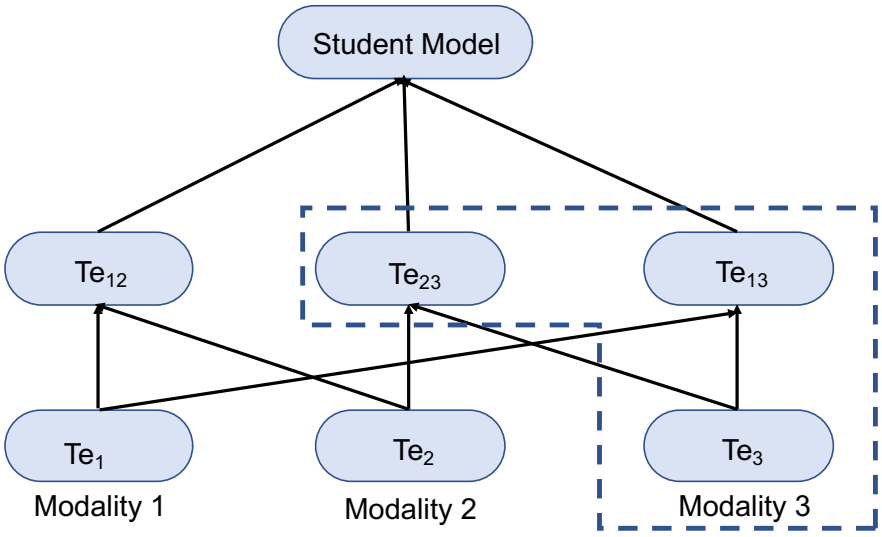


Figure 6.4: **Total teachers need to be trained with pruning (low-level teacher).** If a low level teacher's performance is not good. We could remove this teacher and the upper lever teachers which need the teaching from the low level teacher.

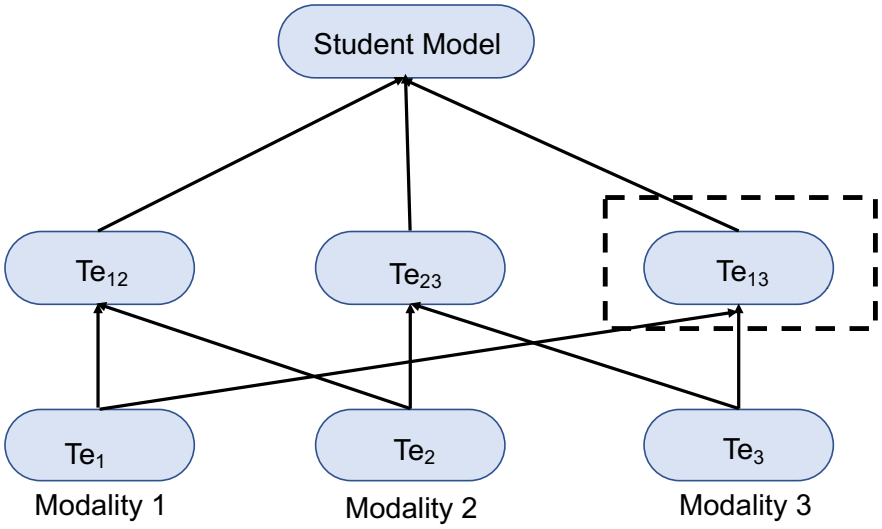


Figure 6.5: **Total teachers need to be trained with pruning (high-level teacher).** If a high level teacher's performance is not good. We could remove this teacher. We do not need to pruning other teachers since there is not upper level teacher.

nal data and imputed data (5) ADV [21]: first using adversarial learning to generate the missing modalities and then training multimodal DNN with the original data and imputed data, (6) Subspace: multimodal subspace learning [106], (7) CCA [54]: canonical correlation analysis, (8) DCCA [9]: deep canonical correlation analysis, (9) T-LATE: weighted adding the teachers' logits, (10) MCTN [2] [86]: multimodal cyclic translation network. Our method is denoted as TS.

6.2.1 Synthetic data experiments

Setting 1: We synthesize data with two modalities in the following steps: (1) We draw n samples from $\mathcal{N}(1,I)$ and $\mathcal{N}(-1,I)$ separately. Samples from each normal distribution form one modality. Denote these samples as X^1 and X^2. The feature dimension is fixed to 32. (2) We then generate random weight matrices $W_1^1 \in \mathbb{R}^{32\times64}, W_1^2 \in \mathbb{R}^{64\times64}, W_2^1 \in \mathbb{R}^{32\times64}, W_2^2 \in \mathbb{R}^{64\times64}$ and use these weight matrices with ReLU function to transform the X^1 and X^2 to abstract features, i.e., $ReLU(ReLU(X^1W_1^1)W_1^2)$ and $ReLU(ReLU(X^2W_2^1)W_2^2)$. (3) After obtaining the transformed features for the two modalities, we concatenate those features to form the joint features and use a linear layer to transform the joint features to logits z. The final class label is $\sigma(z)$. When synthesizing the data, we make sure the number of samples for each class to be the same by generating more than n samples and downsampling. (4) We random select $a\%$ samples to be X^{1c} and X^{2c}. The remaining samples are divided into two equal parts. We remove one modality for each part to form X^{1u} and X^{2u}. So, X^{1u} and X^{2u} all have $n(1-a\%)/2$ samples. For each class, we randomly choose 80% of data as the training set, 10% as the validation set, and 10% as the testing set. We repeat the experiments for 5 times.

In this setting, we fix the number of samples per class to be 400 and change the class number

[2] We use fully connected neural networks instead of RNNs for the encoder, decoder and the prediction subnetwork since our data are not time series data.

in $\{2,5,7,10,12\}$. The samples with complete modalities are fixed to be 40%. The missing rate for each modality is 30%. The teacher model is a DNN model with 3 hidden layers and the hidden nodes are tuned in $\{32,64,128,256\}$. For TS and M-DNN, we fix the network structure to be identical to the one used to generate the data but with unknown weight matrices. Since the two modalities have equal contribution to the output when we synthesize the data, we set α to be equal to β and is tuned in $\{0.1,0.2,\ldots,0.9\}$. Temperature T is tuned in $\{1,5,10,15,20\}$. For T-DNN, we use the layer before the output layer of the teacher models as the abstract features. These abstract features are concatenated to form new features. Then, we train a DNN model with the new features. The DNN model has 3 hidden layers with node number being tuned in $\{64,128,256\}$. For each block of the autoencoder in the CAS-AE model, the encoder has 3 layers and decode has 3 layers. The encoded feature dimension is fixed to be 64 since the original data has 32 features for each modality. The node number for the hidden layer of the encoder and decoder is tuned in $\{128,256,512\}$. We follow the steps in the [111] to tune the number of the autoencoder block, i.e., the joint optimization of the entire network is performed when adding one autoencoder block. During the training phase, we randomly choose half samples from the complete samples to remove one modality and the other half to remove the other modality. Then, we train the CAS-AE to reconstruct the removed modalities. After the training, the CAS-AE is used to impute the missing modalities for the incomplete samples. Finally, we train a multimodal DNN using all the imputed samples and the complete samples together. The structure of the multimodal DNN used here is the same as the student model of TS and M-DNN model. For ADV, the encoder part is a 3 layer DNN, the hidden node number is tuned in $\{128,256,512\}$. The structure of discriminator is a 3-layer DNN with hidden number be tuned in $\{128,256,512\}$. Since ADV can only impute one modality in one time, we first use the first modality to impute the second modality with the complete samples as the training data. Then, we use the imputed samples and the complete samples as training data

to train a second model to impute the first modality. After we impute all the missing part, we train a multimodal DNN to perform the classification. The structure of the multimodal DNN is the same with the student model of TS and M-DNN model. The formulation of Subspace baseline is identical to Eq. (2) in [106]. We initialize the latent factors by SVD of the concatenation of two modality to improve the performance of this model. The latent factor rank is tuned in $\{16, 32, 64\}$. For CCA and DCCA, the projected feature dimension is tuned in $\{16, 32\}$. The hidden node of DCCA is tuned in $\{64, 128, 256, 512\}$ and the hidden layer number is fixed to be 3. For T-LATE, we first use the training samples to learn the optimal weights for each teacher. Then, we use the learned weights and teacher models to label the testing samples. For MCTN, the hidden node of encoder and decoder is tuned in $\{64, 128, 256, 512\}$ and the hidden layer number is fixed to be 3. The prediction subnetwork has one hidden layer and the hidden node number is fixed to be 128.

The results are shown in Figure 6.6. We see that TS outperforms all other models. The performance of Te_1 and Te_2 are much worse than M-DNN since each teacher only has access to the information of one modality. Although they are well-trained with all the available data, the information loss still makes the performance to be worse than M-DNN. The performance of the ADV and CAS-AE is lower than M-DNN because the imputed samples have low quality with limited samples having complete modalities. Although these two methods enlarge the sample size, they still cannot outperform M-DNN. Especially for ADV, the performance is much lower than M-DNN and CAS-AE since adversarial training is much more difficult than training an autoencoder. The difference between T-LATE and the TS model increases as the class number increase which implies that late fusion does not work well when the class number is large. The key difference between our model and T-DNN is that our model uses teachers to teach the student via labeling the samples, but T-DNN directly uses the features extracted by the teachers as the input features. The samples and model structures to train the teachers and the student models are all the same for

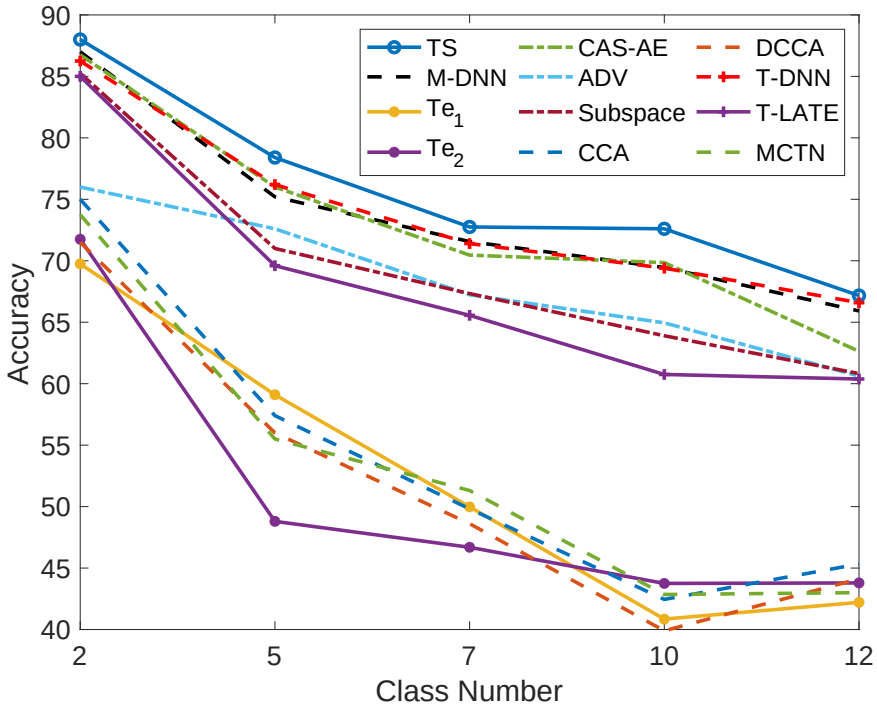


Figure 6.6: **Classification accuracy for Setting 1.** The proposed method (TS) outperforms all the other baselines.

the two methods. However, the performance of T-DNN is worse than the proposed method. One reason is that features extracted by teachers have lost some useful information.

Setting 2: In the second setting, the data are synthesized the same way as Setting 1. We change the rate of samples with complete modalities (complete rate) to be $\{60\%, 50\%, 40\%, 30\%, 20\%\}$. All the model structure and parameter settings are identical to Setting 1. The results are shown in Figure 6.7. We see similar patterns as Setting 1. When the complete rate is large, the performance of TS and M-DNN or CAS-AE is almost the same. But when the complete rate is small enough, TS is much better than M-DNN and CAS-AE since M-DNN and CAS-AE are trained well with a large complete rate. When the complete rate is small, there is no enough data to train them. T-DNN and T-LATE show the opposite pattern with M-DNN and CAS-AE, i.e., the difference between TS and these two models is smaller with a small complete rate than that with a large complete rate. T-DNN and T-LATE rely less on the complete samples. When the complete rate is small, the benefit of

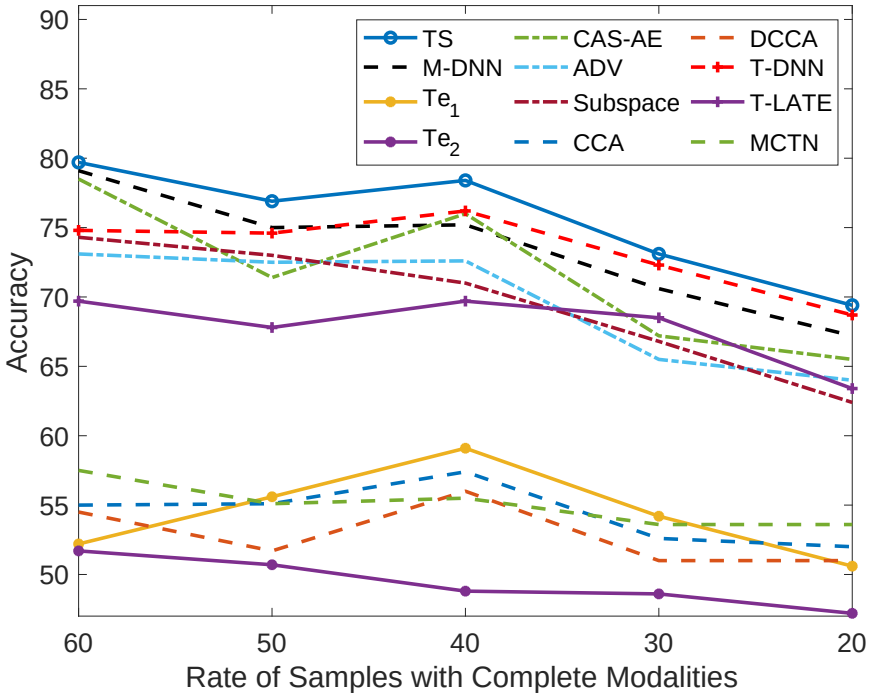


Figure 6.7: **Classification accuracy for Setting 2.** The proposed method (TS) outperforms all the other baselines.

using large data to train the teachers makes them perform much better than the models only using complete samples. For our proposed model, we utilize this benefit to make sure the performance to be good when complete samples are scarce.

Setting 3: In this setting, we show the results of 5-modality synthetic data experiments. The challenge of 5-modality learning is from scalability since there are too many teachers available. We test the proposed pruning strategy in this section. The dataset is synthesized in the following way. (1) We draw n samples from $\mathcal{N}(1,I)$ and $\mathcal{N}(-1,I)$ separately. Samples from each normal distribution form one modality. Denote these samples as X^1 and X^2. The feature dimension is fixed to 32. (2) We use a random matrix $T \in \mathbb{R}^{32\times 32}$ to linearly transform X^1 to form the third modality, i.e., $X^3 = X^1 T$. (4) We take first half features from X^2 and then multiply a random matrix $M \in \mathbb{R}^{16\times 32}$ to form modality 4. (5) We then draw n samples from $\mathcal{N}(0,I)$. The feature dimension is set to 32. These samples form the fifth modality. But when forming

the joint representation, we only use the first half features of the fifth modality, denoted by $X^5_{1/2}$. (6) We then generate a random weight matrices $W^1_1, W^2_1, W^1_2, W^2_2$ and W^1_5, W^2_5. The size is 32 for W^1_1, W^1_2, 64×64 for W^2_1, W^2_2, 16×32 for W^1_5 and 32×32 for W^2_5. (7) We use ReLU as the nonlinear activation function. The joint representation is the concatenation of $ReLU(ReLU(X^1 W^1_1)W^2_1)$, $ReLU(ReLU(X^2 W^1_2)W^2_2)$ and $ReLU(ReLU(X^5_{1/2} W^1_5)W^2_5)]$ We only use X^1, X^2 and X^5 to form joint representation because X^3 and X^4 are generated by X^1 and X^2. (8) A linear layer is added to the joint representation to generate the logits z. The final class label is $\sigma(z)$. (9) We randomly select 40% samples to be X^{1c}, X^{2c}, X^{3c}, X^{4c}, X^{5c}. We divide the remaining samples into three equal parts. We remove one modality for each part to form X^{1u}, X^{2u} and X^{5u}. X^{3u} has the same missing pattern with X^{1u} and X^{4u} has the same missing pattern with X^{2u}. For each class, we choose 80% of data as training, 10% as validation, and 10% as testing. Experiments are repeated 5 times.

We set the number of samples per class to be 1000 and the class number to be 5. We first train the teachers with every single modality. Then, we compare the performance of these teachers. The results are shown in Table 6.1. From Table 6.1, we see the performance of 4-th teacher and 5-teacher is relatively low compared with other teachers. Thus, we only use the first 3 teachers and modalities to form the two-modal teachers, which are Te_{12}, Te_{23} and Te_{13}. Then, we find the performance of Te_{13} is much worse than the performance of Te_{12} and Te_{23}. So, we do not need to train a 3-modality model with modality $1, 2, 3$ as the teacher since it contains both the modality 1 and the modality 3. The final teacher we used are Te_1, Te_2, Te_3, Te_{12}, and Te_{23}. If we do not select teachers, the teacher number will be $2^5 - 1 = 31$. But now, we only need 5 teachers. As a comparison, we train models with modality 5 and 4 and then use them as teachers along with all the 5 teachers to teach the student model. The performance drops to 70.76 ± 0.01. So, when the performance of one teacher is too bad, we do not use this teacher to teach the student. We note that although modality 5 alone has bad performance, it still contributes to the joint representation

as shown in the steps when we synthesize the data. We thus only use this method to select teachers but not the modalities used to train the student model.

6.2.2 Experiments on Alzheimer's diagnosis

In this subsection, we report the experiment performance on the union of two-stage of ADNI datasets [3], i.e., ADNI1 and ADNI2, and NACC dataset [4]. These datasets contain brain imaging data of subjects with different stages of Alzheimer's disease. Two modalities are used in this experiments. The first one is T1 MRI. 136 cortical volume and thickness features are extracted for 68 brain region of interests (ROIs) based on Desiken-Killiany atlas [33]. The second modality is dMRI-derived structural network. We use PICo [84] to construct brain networks for 113 ROIs based on the Harvard Oxford Cortical and subcortical Probabilistic Atlas [33, 40]. Since the network is undirected, we extract the upper triangle of the weighted adjacency matrix to form 6328 features. Finally, We use stability selection [116, 75] to select the top 172 features which have the top 30% stability scores as the final features for this modality. Our task is to classify if the subject is normal control (NC), mild cognitive impairment (MCI) or dementia (AD). ADNI1 data have 223 NC, 385 MCI and 186 AD. ADNI2 data have 50 NC, 112 MCI and 39 AD. NACC data have 329 NC, 57 MCI and 53 AD. ADNI2 and NACC have both dMRI and T1 MRI modalities while ADNI1 only has T1 MRI.

We train the teacher networks, the student network and M-DNN before the fusion layer with 4 hidden layers and the hidden node number is tuned in $\{256, 512, 1024\}$. After the fusion layer, a linear layer with SoftMax classifier is added to complete the classification. α and β are tuned in {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} separately. For CAS-AE, we use 4 layers for

[3] http://adni.loni.usc.edu

[4] https://www.alz.washington.edu

Model	Te_1	Te_2	Te_3	Te_4	Te_5
ACC	47.80 ± 0.09	47.04 ± 0.17	44.98 ± 0.26	34.48 ± 0.05	22.80 ± 0.04
Model	Te_{12}	Te_{23}	Te_{13}	M-DNN	TS
ACC	71.32 ± 0.03	68.84 ± 0.10	47.40 ± 0.3	71.44 ± 0.07	72.28 ± 0.03

Table 6.1: **Classification accuracy of Setting 3.** We use the selected teachers to train the student model. As compassion, the accuracy drops to 70.76 ± 0.01 when adding non-selected teachers Te_4 and Te_5.

Model	TS	M-DNN	Te_1
Acc	75.48 ± 0.07	73.26 ± 0.08	69.67 ± 0.06
Model	Te_2	Subspace	MCTN
Acc	62.98 ± 0.01	67.66 ± 0.04	69.05 ± 0.11
Model	CCA	DCCA	CAS-AE
Acc	61.03 ± 0.47	72.70 ± 0.46	71.11 ± 0.01
Model	ADV	T-DNN	T-LATE
Acc	72.70 ± 0.05	72.27 ± 0.10	74.21 ± 0.01

Table 6.2: **The classification accuracy for all the models trained on the union of ADNI and NACC datasets.**

encoder and 4 layers for decoder. The encoded features dimension is tuned in $\{128, 256\}$. For ADV, the hidden layer number for encoder and the discriminator is set to be 4. The node number is tuned in $\{256, 512, 1024\}$. For Subspace, we tune the rank in $\{32, 64, 128\}$. The projected feature dimension of CCA and DCCA is tuned in $\{32, 64, 128\}$. For MCTN, the hidden layer number is fixed to be 4 for encoder and decoder and the hidden node nunmber is tuned in $\{256, 512, 1024\}$. The prediction subnetwork hidden number is fixed to be 256. We random select 90% samples as training set and the rest as testing set. We repeat the experiment 5 times.

The average classification accuracy is reported in Table 6.2. We see our proposed method outperforms all other baselines. Te_1 is the teacher model trained on T1 MRI and Te_2 is the teacher model trained on the dMRI modality. For this dataset, all the samples have the first modality and only part of the samples have the second modality. So, the performance of Te_1 is much higher than the performance of Te_2. This is also reflected in the regularization parameters α and β. The best performance for our proposed model is reached when α is 0.7 and β is 0.0. Since dMRI modality

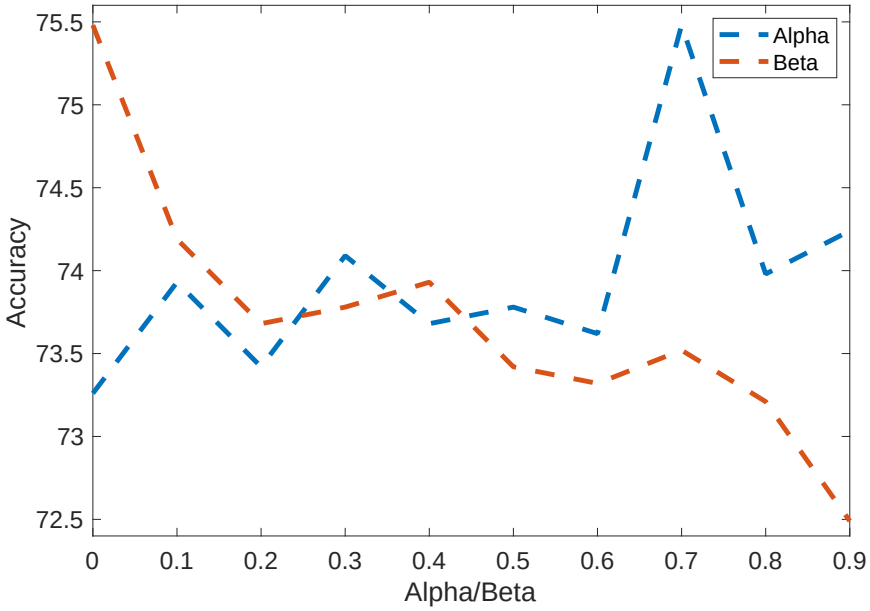


Figure 6.8: **Accuracy with different α and β.** α is fixed to be 0.7 while changing β and β is fixed to be 0.0 while changing α.

is missing for some samples and T1 MRI modality is complete for all samples, the single teacher training on dRMI will be useless. Thus, when β is 0.0, the performance is the highest. Figure 6.8 shows how the accuracy changes with the parameter α and β. In this figure, we change α when fixing β to be 0.0 and change β when fixing α to be 0.7. The performance decreases with the increasing of β. Meanwhile, the teacher trained with T1 MRI improves the performance a lot with a large α. We also show the top important T1 MRI features for Te_1, M-DNN and TS model in Figure 6.9, Figure 6.10, Figure 6.11 and the top important dMRI features for Te_1, M-DNN and TS model inFigure 6.12 (the top important dMRI features for M-DNN and TS model are the same sime the dMRI teacher do not have contribution to the training of the student model). The features are ranked by the absolute weights value between the input layer and the first hidden layer. We sum all the absolute values of the weights that are connected with the input node as the relative importance of the associated input feature. We see there are some overlapping between the top important features of the three models but still some top features are very different for Te_1 and TS/M-DNN.

For example, right isthmuscingulate thickness is ranked the third most important feature for the teacher models and the most important features for the student models. Left entorhinal volume is the second most important feature for M-DNN/TS but does not in the top 10 important features for the Te_1. Both two features have been proved to be related to Alzheimer's disease [53, 44]. The difference between the importance of the features causes T-DNN to be worst than TS model as T-DNN uses the features extracted by Te_1. Training with two modalities simultaneously leads to different feature ranks since the two modalities are coupled and influence each other. Some features in one modality alone do not show to be important. But these features could be very important with the presence of some features from the other modality.

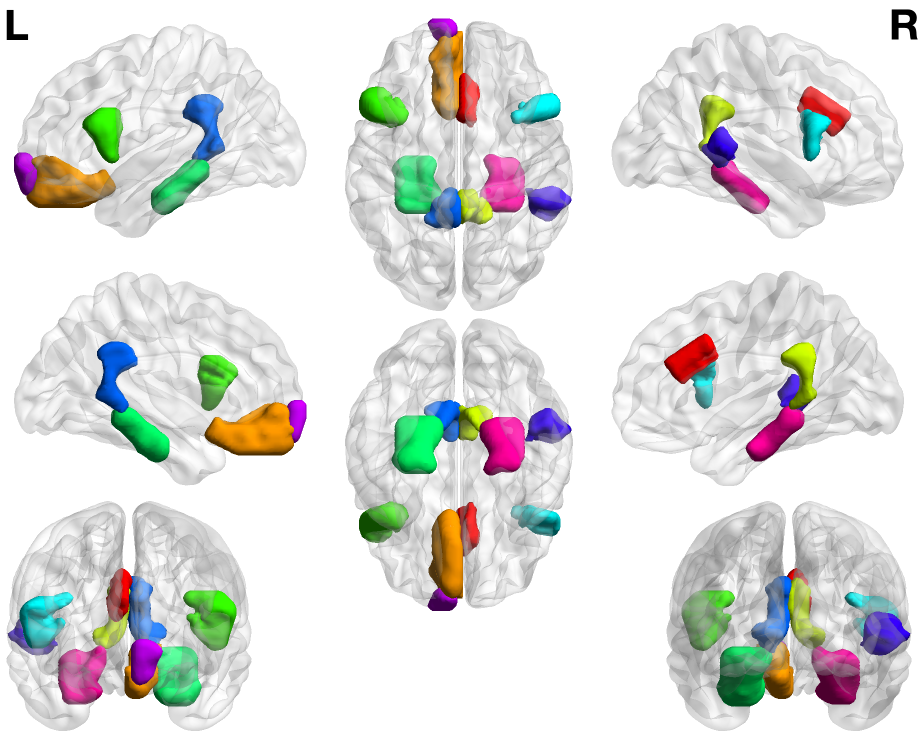


Figure 6.9: **The top 10 important T1 MRI features for Te_1 trained on the union of NACC and ADNI datasets.**

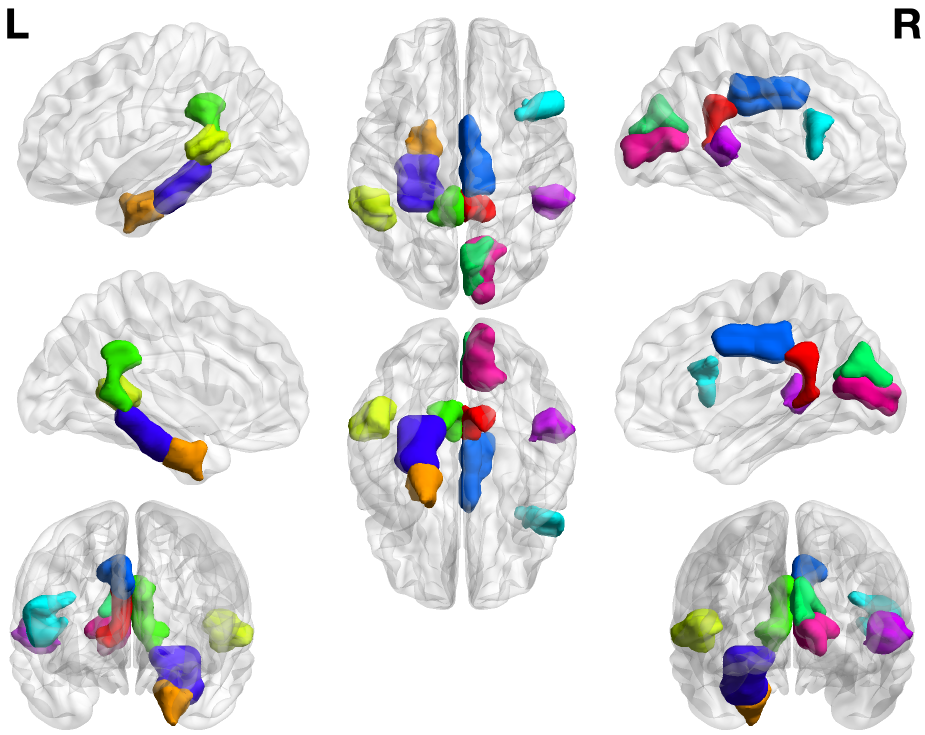


Figure 6.10: **The top 10 important T1 MRI features for M-DNN trained on the union of NACC and ADNI datasets.**

6.2.3 Experiments on other real-world datasets

In this section, we report the performance on three additional real-world datasets. The first one is Alzheimer's disease data from [132], which has 3 modalities and 3 classes available, i.e., MRI, PET, Proteomics. The feature dimensions for these 3 modalities are 305, 116 and 147, respectively. In this dataset, 648 subjects have MRI data. 372 subjects have PET data. 496 subjects have Proteomics data. Only 215 subjects have all three modalities. We randomly split the data into the training set and testing set with the ratio $0.9 : 0.1$. The parameters are tuned the same way as Section 6.2.2. We repeat the experiments for 5 iterations. The average accuracy is shown in Table 6.5. From the table, we see the performance of M-DNN is even worst than Te_{13} since when training the M-DNN with all the three modalities, the sample size is much smaller than that used to train Te_{13}. But with the teaching step, the performance improves a lot and outperforms the

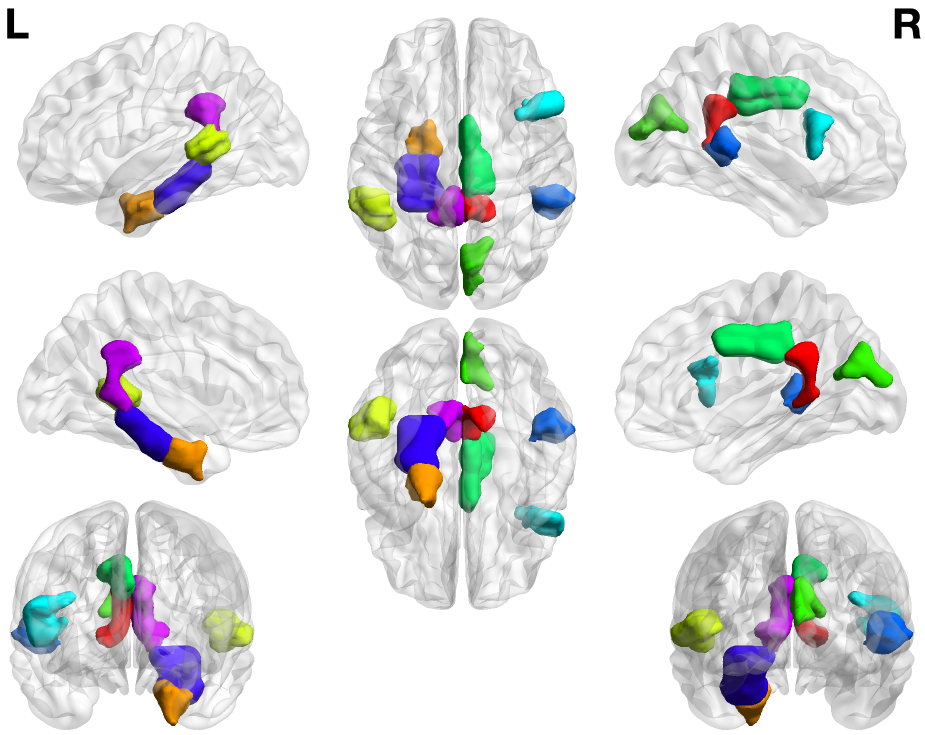


Figure 6.11: **The top 10 important T1 MRI features for TS trained on the union of NACC and ADNI datasets.**

performance of Te_{13}.

Another two real-world datasets we used are MNIST and XRMB [119]. For MNIST data, we subsample 10,000 as training data, 1,000 samples as validation data and 1,000 samples as testing data. The class number is 10. MNIST has two modalities with 784 features for each modality. For XRMB data, we subsample 19,500 samples for training, 1,950 for validation and 1,950 for testing. The class number for XRMB is 39. Two modalities are available for XRMB data with 273 and 112 features. Since these data do not have missing modalities, we randomly choose $a\%$ of samples to be the samples with complete modalities. And for the rest part of the data, we split them into two parts and remove one modality for each part. We change the rate of complete modalities in $\{40\%, 30\%, 20\%, 10\%\}$. The parameters are tuned the same way as Section 6.2.2 except for the node number. The hidden layer node number is tuned in {512, 1024, 2048}. The encoded feature

Rate	40%	30%	20%	10%
TS	66.13 ± 0.03	64.77 ± 0.01	63.19 ± 0.01	58.36 ± 0.01
M-DNN	62.66 ± 0.01	60.59 ± 0.01	57.18 ± 0.01	50.33 ± 0.03
Te_1	56.05 ± 0.01	53.13 ± 0.01	51.08 ± 0.01	44.57 ± 0.01
Te_2	45.73 ± 0.01	42.59 ± 0.01	41.63 ± 0.01	37.93 ± 0.01
CAS-AE	59.75 ± 0.02	57.96 ± 0.01	56.58 ± 0.01	53.84 ± 0.01
ADV	59.37 ± 0.01	57.83 ± 0.02	56.37 ± 0.01	53.60 ± 0.01
Subspace	45.25 ± 0.02	41.63 ± 0.01	38.08 ± 0.01	34.15 ± 0.01
DCCA	41.94 ± 0.41	41.64 ± 0.46	33.53 ± 0.13	32.86 ± 0.34
T-DNN	65.14 ± 0.02	63.11 ± 0.01	61.61 ± 0.01	56.59 ± 0.01
T-ENS	63.69 ± 0.01	61.91 ± 0.02	59.70 ± 0.02	56.13 ± 0.01
MCTN	53.58 ± 0.01	51.18 ± 0.02	47.78 ± 0.03	40.38 ± 0.02

Table 6.3: **The classification accuracy of all the models trained on XRMB dataset.**

Rate	40%	30%	20%	10%
TS	96.46 ± 0.01	96.00 ± 0.01	95.42 ± 0.01	92.34 ± 0.01
M-DNN	93.70 ± 0.01	92.04 ± 0.03	89.04 ± 0.02	86.46 ± 0.01
Te_1	93.04 ± 0.01	91.78 ± 0.01	90.72 ± 0.02	87.12 ± 0.01
Te_2	78.82 ± 0.09	74.52 ± 0.02	69.66 ± 0.06	57.08 ± 0.08
CAS-AE	94.54 ± 0.01	94.26 ± 0.01	93.72 ± 0.01	91.48 ± 0.01
ADV	94.98 ± 0.01	94.42 ± 0.01	94.32 ± 0.01	91.74 ± 0.01
Subspace	86.70 ± 0.01	84.34 ± 0.02	79.76 ± 0.04	72.28 ± 0.06
DCCA	87.38 ± 0.09	84.70 ± 0.15	81.60 ± 0.16	76.72 ± 0.31
T-DNN	95.18 ± 0.01	94.92 ± 0.01	92.25 ± 0.01	92.28 ± 0.04
T-ENS	95.90 ± 0.01	94.74 ± 0.01	94.44 ± 0.01	90.50 ± 0.01
MCTN	92.24 ± 0.01	90.22 ± 0.01	88.86 ± 0.01	85.02 ± 0.02

Table 6.4: **The classification accuracy of all the models trained on MNIST dataset.**

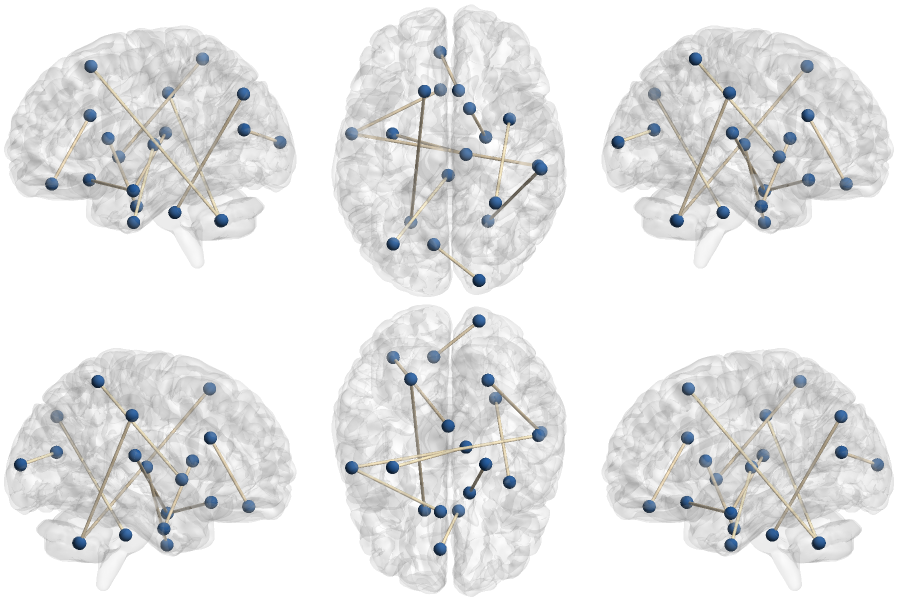
dimension for CAS-ADV is tuned in {128, 256, 512}. The projected feature numbers for CCA, DCCA and Subspace are tuned in {128, 256, 512} for MNIST and {32, 64, 100} for XRMB. The experiments are repeated for 5 times and the results are shown in Table 6.3 and Table 6.4. We see that our method outperforms all other baselines under different missing rates.

Model	TS	M-DNN	T-DNN
Accuracy	55.57 ± 0.02	47.43 ± 0.05	45.57 ± 0.02
Model	Te_{12}	Te_{13}	Te_{23}
Accuracy	48.57 ± 0.02	54.43 ± 0.06	52.29 ± 0.06
Model	Te_1	Te_2	Te_3
Accuracy	48.14 ± 0.01	45.14 ± 0.31	47.43 ± 0.24
Model	CAS-AE	ADV	T-ENS
Accuracy	53.27 ± 0.02	53.04 ± 0.06	53.86 ± 0.02

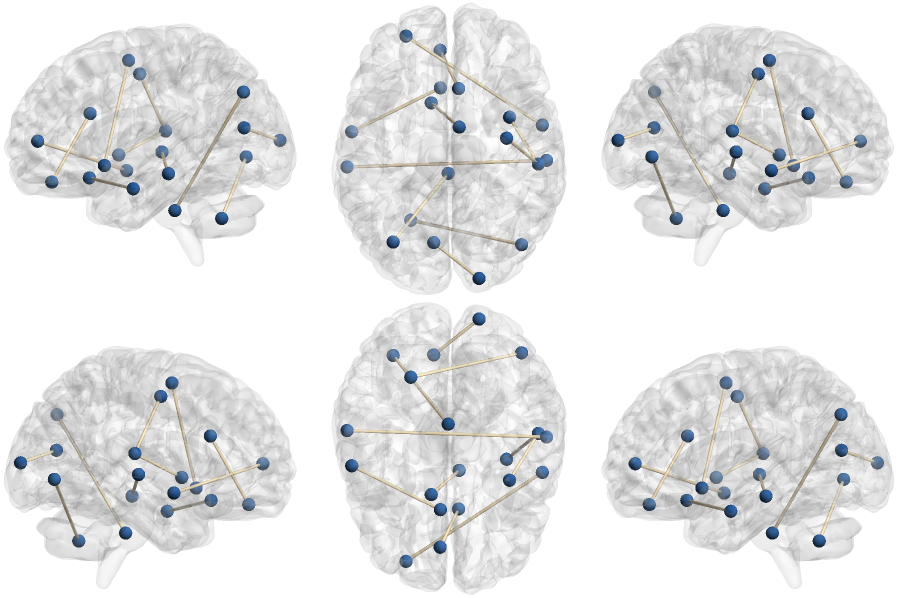
Table 6.5: **The classification accuracy for the models trained on Alzheimer's disease data from [132].**

6.3 Summary

In this work, we proposed a novel framework to fuse the supplementary information of multiple modalities for the datasets with missing modalities. We first trained models on each modality with all the available data to obtain teacher models. Then, we used these teacher models to teach a multimodal DNN network by knowledge distillation. Since the teacher models were trained on relatively larger datasets compared with the datasets used to train the student model, the teachers were experts on each modality and the expertise could help the student to improve the performance. The experiment results on both synthetic and real-world data verified the effectiveness of the proposed method.



(a) Te_2



(b) TS/M-DNN

Figure 6.12: **The top 10 important dMRI features for models trained on the union of NACC and ADNI datasets.**

Chapter 7

Conclusion

In this dissertation, I propose four algorithms for multimodal learning and demonstrate how the proposed algorithms help modeling the Alzheimer's disease. The four algorithms have different assumptions and fit different problems and data types.

The first algorithm adopts a convex combination of the modalities. It requires the feature dimensions of the modalities to be the same. One assumption of the algorithm is the modalities are linearly interacted. Therefore, when using this algorithm, the interaction of modalities is expected to be linear.

The second algorithm can be applied to modalities with different dimensions. It does not require the modalities to be linearly interacted. The assumption of the second algorithm is that each modality has enough information on the subject and may contain some useless information. For example, the brain imaging data contain not only Alzheimer's disease information but also the brain functions information. Moreover, when collecting the data, instruments may be inaccurate which makes the data noisy. Since the second algorithm learns the common part of the modalities, the noise and the irrelevant information of the modalities are not included in the modality-invariant component.

The third algorithm is to fuse the supplement information of the modalities. It assumes each modality only has partial information of the subjects. Combining the information from all the subjects provides a more comprehensive description of the subjects. Since modalities may have

irrelevant information and noise, this algorithm filters irrelevant information and noise when learning the joint representation. Therefore, this algorithm can be applied to the modalities that having incomplete information on the subject, and the performance is expected to be better than the existing algorithm when the modalities have irrelevant information and noise.

The fourth algorithm is proposed to deal with the data having missing modalities. The second algorithm can also be applied to the data having missing modalities. The difference between this algorithm and the second algorithm is that the second algorithm assumes each modality has complete information of the subjects. The fourth algorithm does not have this assumption. It is worth mentioning that the student model could be replaced by a variant of multimodal algorithms although the student model used in this dissertation is a multimodal DNN which fuses the supplementary information of the modalities. Therefore, this algorithm can also be applied to the modalities having complete information on the subject and learn the common structure of the modalities.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Iman Aganj, Christophe Lenglet, Neda Jahanshad, Essa Yacoub, Noam Harel, Paul M Thompson, and Guillermo Sapiro. A Hough transform global probabilistic approach to multiple-subject diffusion MRI tractography. *Medical Image Analysis*, 15(4):414–425, 2011.

[3] Unaiza Ahsan and Irfan Essa. Clustering social event images using kernel canonical correlation analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 800–805, 2014.

[4] Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

[5] H Akalın, Yahya Karaman, H Demirtaş, N İmamoğlu, Yusuf Özkul, et al. Evaluation of the nucleolar organizer regions in alzheimer's disease. *Gerontology*, 51(5):297–301, 2005.

[6] Zeynep Akata, Christian Thurau, and Christian Bauckhage. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In *16th Computer vision winter workshop*, 2011.

[7] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[8] Alzheimer's Association. 2013 alzheimer's disease facts and figures. 2013.

[9] Galen Andrew, Raman Arora, Jeff A Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013.

[10] Raman Arora and Karen Livescu. Multi-view cca-based acoustic features for phonetic recognition across speakers and domains. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7135–7139. IEEE, 2013.

[11] Peter J Basser, Sinisa Pajevic, Carlo Pierpaoli, Jeffrey Duda, and Akram Aldroubi. In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine*, 44(4):625–632, 2000.

[12] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[13] TEJ Behrens, H Johansen Berg, Saad Jbabdi, MFS Rushworth, and MW Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *NeuroImage*, 34(1):144–155, 2007.

[14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[15] Lars Bertram and Rudolph E Tanzi. Thirty years of alzheimer's disease genetics: the implications of systematic meta-analyses. *Nature Reviews Neuroscience*, 9(10):768–778, 2008.

[16] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[17] Magnus Borga. Canonical correlation: a tutorial. *On line tutorial http://people. imt. liu. se/magnus/cca*, 4:5, 2001.

[18] Ulf Brefeld, Thomas Gärtner, Tobias Scheffer, and Stefan Wrobel. Efficient co-regularised least squares regression. In *Proceedings of the 23rd international conference on Machine learning*, pages 137–144. ACM, 2006.

[19] Alistair Burns, Margaret Reith, Robin Jacoby, and Raymond Levy. 'how to do it'—obtaining consent for autopsy in alzheimer's disease. *International Journal of Geriatric Psychiatry*, 5(5):283–286, 1990.

[20] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[21] Lei Cai, Zhengyang Wang, Hongyang Gao, Dinggang Shen, and Shuiwang Ji. Deep adversarial learning for multi-modality missing data completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1158–1166. ACM, 2018.

[22] Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

[23] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[24] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[25] Shiyu Chang, Guo-Jun Qi, Charu C Aggarwal, Jiayu Zhou, Meng Wang, and Thomas S Huang. Factorized similarity learning in networks. In *ICDM*, pages 60–69. IEEE, 2014.

[26] Yu-Ling Chang, Mark W Jacobson, Christine Fennema-Notestine, Donald J Hagler, Robin G Jennings, Anders M Dale, Linda K McEvoy, Alzheimer's Disease Neuroimaging Initiative, et al. Level of executive function influences verbal memory in amnestic mild cognitive impairment and predicts prefrontal and posterior cingulate thickness. *Cerebral Cortex*, 20(6):1305–1313, 2010.

[27] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, pages 129–136. ACM, 2009.

[28] Minmin Chen, Kilian Q Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1476–1484, 2014.

[29] Thomas E Conturo, Nicolas F Lori, Thomas S Cull, Erbil Akbudak, Abraham Z Snyder, Joshua S Shimony, Robert C McKinstry, Harold Burton, and Marcus E Raichle. Tracking neuronal fiber pathways in the living human brain. *Proceedings of the National Academy of Sciences*, 96(18):10422–10427, 1999.

[30] EH Corder, AM Saunders, WJ Strittmatter, DE Schmechel, PC Gaskell, GWet al Small, AD Roses, JL Haines, and Margaret A Pericak-Vance. Gene dose of apolipoprotein e type 4 allele and the risk of alzheimer's disease in late onset families. *Science*, 261(5123):921–923, 1993.

[31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[32] Maxime Descoteaux, Rachid Deriche, Thomas R Knosche, and Alfred Anwander. Deterministic and probabilistic tractography based on complex fibre orientation distributions. *IEEE transactions on medical imaging*, 28(2):269–286, 2009.

[33] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *NeuroImage*, 31(3):968–980, 2006.

[34] DP Devanand, Ravi Bansal, Jun Liu, Xuejun Hao, Gnanavalli Pradhaban, and Bradley S Peterson. Mri hippocampal and entorhinal cortex mapping in predicting conversion to alzheimer's disease. *Neuroimage*, 60(3):1622–1629, 2012.

[35] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2075–2082, 2014.

[36] Robin D Dowell, Owen Ryan, An Jansen, Doris Cheung, Sudeep Agarwala, Timothy Danford, Douglas A Bernstein, P Alexander Rolfe, Lawrence E Heisler, Brian Chin, et al. Genotype to phenotype: a complex problem. *Science*, 328(5977):469–469, 2010.

[37] Craig K Enders. *Applied missing data analysis*. Guilford press, 2010.

[38] Otto Fabius and Joost R van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[39] Dean P Foster, Sham M Kakade, and Tong Zhang. Multi-view dimensionality reduction via canonical correlation analysis. *Technical Report TR-2008-4*, 2008.

[40] Jean A Frazier, Sufen Chiu, Janis L Breeze, Nikos Makris, Nicholas Lange, David N Kennedy, Martha R Herbert, Eileen K Bent, Vamsi K Koneru, Megan E Dieterich, et al. Structural brain magnetic resonance imaging of limbic and thalamic volumes in pediatric bipolar disorder. *American Journal of Psychiatry*, 162(7):1256–1265, 2005.

[41] David C Glahn, Paul M Thompson, and John Blangero. Neuroimaging endophenotypes: strategies for finding genes influencing brain structure and function. *Human brain mapping*, 28(6):488–501, 2007.

[42] Shiri Gordon, Hayit Greenspan, and Jacob Goldberger. Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations. In *null*, page 370. IEEE, 2003.

[43] Hatice Gunes and Massimo Piccardi. Affect recognition from face and body: early fusion vs. late fusion. In *2005 IEEE international conference on systems, man and cybernetics*, volume 4, pages 3437–3443. IEEE, 2005.

[44] Leticia Gutiérrez-Galve, Manja Lehmann, Nicola Z Hobbs, Matthew J Clarkson, Gerard R Ridgway, Sebastian Crutch, Sebastien Ourselin, Jonathan M Schott, Nick C Fox, and Josephine Barnes. Patterns of cortical thickness according to apoe genotype in alzheimer's disease. *Dementia and geriatric cognitive disorders*, 28(5):461–470, 2009.

[45] Päivi Hartikainen, Janne Räsänen, Valtteri Julkunen, Eini Niskanen, Merja Hallikainen, Miia Kivipelto, Ritva Vanninen, Anne M Remes, and Hilkka Soininen. Cortical thickness in frontotemporal dementia, mild cognitive impairment, and alzheimer's disease. *Journal of Alzheimer's Disease*, 30(4):857–874, 2012.

[46] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[47] Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[48] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[49] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[50] Winston H Hsu, Lyndon S Kennedy, and Shih-Fu Chang. Video search reranking via information bottleneck principle. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 35–44. ACM, 2006.

[51] Keith A Johnson, Nick C Fox, Reisa A Sperling, and William E Klunk. Brain imaging in alzheimer disease. *Cold Spring Harbor perspectives in medicine*, 2(4):a006213, 2012.

[52] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 4. Prentice-Hall New Jersey, 2014.

[53] K Juottonen, MP Laakso, R Insausti, M Lehtovirta, A Pitkänen, K Partanen, and H Soininen. Volumes of the entorhinal and perirhinal cortices in alzheimer's disease. *Neurobiology of aging*, 19(1):15–22, 1998.

[54] Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*, pages 82–96. Springer, 2007.

[55] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[56] Bu Sung Kim, Heera Kim, Jaedong Lee, and Jee-Hyong Lee. Improving a recommender system by collective matrix factorization with tag information. In *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, pages 980–984. IEEE, 2014.

[57] Jungsu Kim, Jacob M Basak, and David M Holtzman. The role of apolipoprotein e in alzheimer's disease. *Neuron*, 63(3):287–303, 2009.

[58] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[59] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[60] Marius Kloft, Ulf Brefeld, Pavel Laskov, Klaus-Robert Müller, Alexander Zien, and Sören Sonnenburg. Efficient and accurate lp-norm multiple kernel learning. In *NIPS*, pages 997–1005, 2009.

[61] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[62] Michael Krawczak, Susanna Nikolaus, Huberta von Eberstein, Peter JP Croucher, Nour Eddine El Mokhtari, and Stefan Schreiber. Popgen: population-based recruitment of patients and controls for the analysis of complex genotype-phenotype relationships. *Public Health Genomics*, 9(1):55–61, 2006.

[63] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[64] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[65] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 393–400, 2011.

[66] Mariana Lazar, David M Weinstein, Jay S Tsuruda, Khader M Hasan, Konstantinos Arfanakis, M Elizabeth Meyerand, Benham Badie, Howard A Rowley, Victor Haughton, Aaron Field, et al. White matter tractography using diffusion tensor deflection. *Human Brain Mapping*, 18(4):306–321, 2003.

[67] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[68] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[69] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2001.

[70] Alex D Leow, Siwei Zhu, Liang Zhan, Katie McMahon, Greig I de Zubicaray, Matthew Meredith, MJ Wright, AW Toga, and PM Thompson. The tensor distribution function. *Magnetic Resonance in Medicine*, 61(1):205–214, 2009.

[71] Chia-Chan Liu, Takahisa Kanekiyo, Huaxi Xu, and Guojun Bu. Apolipoprotein e and alzheimer disease: risk, mechanisms and therapy. *Nature Reviews Neurology*, 9(2):106–118, 2013.

[72] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 547–556. ACM, 2009.

[73] Xiaoxiao Liu, Marc Niethammer, Roland Kwitt, Nikhil Singh, Matt McCormick, and Stephen Aylward. Low-rank atlas image analyses in the presence of pathologies. *IEEE transactions on medical imaging*, 34(12):2583–2591, 2015.

[74] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[75] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[76] David Meunier, Renaud Lambiotte, Alex Fornito, Karen D Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Hierarchy and dynamics in neural networks*, 1(2), 2010.

[77] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[78] Susumu Mori, Barbara J Crain, VP Chacko, and Peter Van Zijl. Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of Neurology*, 45(2):265–269, 1999.

[79] Abdullah-Al Nahid and Yinan Kong. Involvement of machine learning for breast cancer image classification: a survey. *Computational and mathematical methods in medicine*, 2017, 2017.

[80] John A Nelder and R Jacob Baker. Generalized linear models. *Encyclopedia of statistical sciences*, 1972.

[81] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

[82] Eini Niskanen, Mervi Könönen, Sara Määttä, Merja Hallikainen, Miia Kivipelto, Silvia Casarotto, Marcello Massimini, Ritva Vanninen, Esa Mervaala, Jari Karhu, et al. New insights into alzheimer's disease progression: a combined tms and structural mri study. *PLoS One*, 6(10):e26113, 2011.

[83] Yongsheng Pan, Mingxia Liu, Chunfeng Lian, Tao Zhou, Yong Xia, and Dinggang Shen. Synthesizing missing pet from mri with cycle-consistent generative adversarial networks for alzheimer's disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 455–463. Springer, 2018.

[84] Geoffrey JM Parker, Hamied A Haroon, and Claudia AM Wheeler-Kingshott. A framework for a streamline-based probabilistic index of connectivity (PICo) using a structural

interpretation of mri diffusion measurements. *Journal of Magnetic Resonance Imaging*, 18(2):242–254, 2003.

[85] Ronald C Petersen, Rachelle Doody, Alexander Kurz, Richard C Mohs, John C Morris, Peter V Rabins, Karen Ritchie, Martin Rossor, Leon Thal, and Bengt Winblad. Current concepts in mild cognitive impairment. *Archives of neurology*, 58(12):1985–1992, 2001.

[86] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6892–6899, 2019.

[87] Snehashis Roy, John A Butman, Daniel S Reich, Peter A Calabresi, and Dzung L Pham. Multiple sclerosis lesion segmentation from brain mri via fully convolutional neural networks. *arXiv preprint arXiv:1803.09172*, 2018.

[88] Javad Salimi Sartakhti, Mohammad Hossein Zangooei, and Kourosh Mozafari. Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (svm-sa). *Computer methods and programs in biomedicine*, 108(2):570–579, 2012.

[89] Marzia A Scelsi, Raiyan R Khan, Marco Lorenzi, Leigh Christopher, Michael D Greicius, Jonathan M Schott, Sebastien Ourselin, and Andre Altmann. Genetic study of multimodal imaging alzheimer's disease progression score implicates novel loci. *Brain*, 141(7):2167–2180, 2018.

[90] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[91] Tijn M Schouten, Marisa Koini, Frank de Vos, Stephan Seiler, Jeroen van der Grond, Anita Lechner, Anne Hafkemeijer, Christiane Möller, Reinhold Schmidt, Mark de Rooij, et al. Combining anatomical, diffusion, and resting state functional magnetic resonance imaging for individual classification of mild and moderate alzheimer's disease. *NeuroImage: Clinical*, 11:46–51, 2016.

[92] Alexander G Schwing and Raquel Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

[93] Dennis J Selkoe. Amyloid β-protein and the genetics of alzheimer's disease. *Journal of Biological Chemistry*, 271(31):18295–18298, 1996.

[94] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[95] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-

supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, pages 74–79, 2005.

[96] Vikas Sindhwani and David S Rosenberg. An rkhs for multi-view learning and manifold co-regularization. In *Proceedings of the 25th international conference on Machine learning*, pages 976–983. ACM, 2008.

[97] Noam Slonim, Rachel Somerville, Naftali Tishby, and Ofer Lahav. Objective classification of galaxy spectra using the information bottleneck method. *Monthly Notices of the Royal Astronomical Society*, 323(2):270–284, 2001.

[98] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 208–215. ACM, 2000.

[99] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402, 2005.

[100] Patricia A Soranno, Linda C Bacon, Michael Beauchene, Karen E Bednar, Edward G Bissell, Claire K Boudreau, Marvin G Boyer, Mary T Bremigan, Stephen R Carpenter, Jamie W Carr, et al. Lagos-ne: a multi-scaled geospatial and temporal database of lake ecological context and water quality for thousands of us lakes. *GigaScience*, 6(12):1–22, 2017.

[101] Olaf Sporns. *Networks of the Brain*. MIT press, 2011.

[102] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: a structural description of the human brain. *PLoS Comput Biol*, 1(4):e42, 2005.

[103] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[104] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[105] Qiuling Suo, Weida Zhong, Fenglong Ma, Ye Yuan, Jing Gao, and Aidong Zhang. Metric learning on healthcare data with incomplete modalities. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3534–3540. AAAI Press, 2019.

[106] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. Incomplete multi-view weak-label learning. In *IJCAI*, pages 2703–2709, 2018.

[107] Ryan Tibshirani. Proximal gradient descent and acceleration. *Lecture Notes*, 2010.

[108] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[109] Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In *Advances in neural information processing systems*, pages 640–646, 2001.

[110] Lucas R Trambaiolli, Ana C Lorena, Francisco J Fraga, Paulo AM Kanda, Renato Anghinah, and Ricardo Nitrini. Improving alzheimer's disease diagnosis with machine learning techniques. *Clinical EEG and neuroscience*, 42(3):160–165, 2011.

[111] Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1405–1414, 2017.

[112] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[113] Grigorios Tzortzis and Aristidis Likas. Kernel-based weighted multi-view clustering. In *2012 IEEE 12th international conference on data mining*, pages 675–684. IEEE, 2012.

[114] Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1065–1072. ACM, 2009.

[115] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[116] Qi Wang, Lei Guo, Paul M Thompson, Clifford R Jack Jr, Hiroko Dodge, Liang Zhan, Jiayu Zhou, Alzheimer's Disease Neuroimaging Initiative, et al. The added value of diffusion-weighted mri-derived structural connectome in evaluating mild cognitive impairment: A multi-cohort validation. *Journal of Alzheimer's Disease*, 64(1):149–169, 2018.

[117] Qi Wang, Liang Zhan, Paul M Thompson, Hiroko H Dodge, and Jiayu Zhou. Discriminative fusion of multiple brain networks for early mild cognitive impairment detection. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 568–572. IEEE, 2016.

[118] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Partial multi-view clustering via consistent gan. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1290–1295. IEEE, 2018.

[119] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. On deep multi-view repre-

sentation learning. In *ICML*, pages 1083–1092, 2015.

[120] Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4590–4594. IEEE, 2015.

[121] Yishu Wang, Dejie Yang, and Minghua Deng. Low-rank and sparse matrix decomposition for genetic interaction data. *BioMed research international*, 2015, 2015.

[122] John Westbury, Paul Milenkovic, Gary Weismer, and Raymond Kent. X-ray microbeam speech production database. *The Journal of the Acoustical Society of America*, 88(S1):S56–S56, 1990.

[123] Jennifer Williams, Steven Kleinegesse, Ramona Comanescu, and Oana Radu. Recognizing emotions in video using multimodal dnn feature fusion. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, pages 11–19, 2018.

[124] World Health Organization. Dementia fact sheet n362. Retrieved at `https://web.archive.org/web/20150318030901/http://www.who.int/mediacentre/factsheets/fs362/en`, 2016. Retrieved 13 January 2016.

[125] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

[126] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[127] Chang Xu, Dacheng Tao, and Chao Xu. Large-margin multi-viewinformation bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1559–1572, 2014.

[128] Liu Yang, Liping Jing, and Michael K Ng. Robust and non-negative collective matrix factorization for text-to-image transfer learning. *IEEE transactions on Image Processing*, 24(12):4701–4714, 2015.

[129] Tao Yang, Jun Liu, Pinghua Gong, Ruiwen Zhang, Xiaotong Shen, and Jieping Ye. Absolute fused lasso & its application to genome-wide association studies. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[130] Tao Yang, Jie Wang, Qian Sun, Derrek P Hibar, Neda Jahanshad, Li Liu, Yalin Wang, Liang Zhan, Paul M Thompson, and Jieping Ye. Detecting genetic risk factors for Alzheimer's disease in whole genome sequence data via Lasso screening. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pages 985–989. IEEE, 2015.

[131] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 765–774. IEEE, 2012.

[132] Lei Yuan, Yalin Wang, Paul M Thompson, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61(3):622–632, 2012.

[133] Liang Zhan, Neda Jahanshad, Yan Jin, Arthur W Toga, Katie L McMahon, Greig de Zubicaray, Nicholas G Martin, Margaret J Wright, Paul M Thompson, et al. Brain network efficiency and topology depend on the fiber tracking method: 11 tractography algorithms compared in 536 subjects. In *10th International Symposium on Biomedical Imaging (ISBI)*, pages 1134–1137. IEEE, 2013.

[134] Liang Zhan, Jiayu Zhou, Yalin Wang, Yan Jin, Neda Jahanshad, Gautam Prasad, Talia M Nir, Cassandra D Leonardo, Jieping Ye, Paul M Thompson, et al. Comparison of nine tractography algorithms for detecting abnormal structural brain networks in Alzheimer's disease. *Frontiers in Aging Neuroscience*, 7, 2015.

[135] Daoqiang Zhang, Yaping Wang, Luping Zhou, Hong Yuan, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multimodal classification of alzheimer's disease and mild cognitive impairment. *Neuroimage*, 55(3):856–867, 2011.

[136] Wenming Zheng, Xiaoyan Zhou, Cairong Zou, and Li Zhao. Facial expression recognition using kernel canonical correlation analysis (kcca). *IEEE transactions on neural networks*, 17(1):233–238, 2006.

[137] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in neural information processing systems*, pages 702–710, 2011.

[138] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.

[139] Jiayu Zhou, Zhaosong Lu, Jimeng Sun, Lei Yuan, Fei Wang, and Jieping Ye. Feafiner: biomarker identification from medical data through feature generalization and selection. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1034–1042. ACM, 2013.

[140] Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. From micro to macro: data driven phenotyping by densification of longitudinal electronic medical records. In *SIGKDD*, pages 135–144. ACM, 2014.

[141] Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for pre-

dicting disease progression. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 814–822. ACM, 2011.

[142] Hongtu Zhu, Zakaria Khondker, Zhaohua Lu, and Joseph G. Ibrahim. Bayesian generalized low rank regression models for neuroimaging phenotypes and genetic markers. *Journal of the American Statistical Association*, 109(507):977–990, 2014.

[143] Chen Zu, Biao Jie, Mingxia Liu, Songcan Chen, Dinggang Shen, Daoqiang Zhang, Alzheimer's Disease Neuroimaging Initiative, et al. Label-aligned multi-task feature learning for multimodal classification of alzheimer's disease and mild cognitive impairment. *Brain imaging and behavior*, 10(4):1148–1159, 2016.