Learning Multimodal Data using Deep Neural Networks in Classification

by

Eunsil Seok

(Under the Direction of Nicole A. Lazar)

Abstract

Learning multimodal data has become increasingly widespread as collecting complementary information from multiple sources becomes easier and principled statistical inferences on multimodal data provide deeper insights into an event of interest. Nevertheless, learning multimodal data poses practical challenges such as many modern data are high-dimensional, signals from different sources are often highly correlated, and the types and dimensions of data coming from multiple sources are different. This dissertation studies effective learning methods to handle such problems in classification settings. Using deep neural network models, we propose new combining methods that take the importance of each modality into account and weight modalities based on their uncertainty for prediction. We demonstrate the practical efficacy of the proposed methods on digit classification and Alzheimer's disease classification problems. The numerical experiment results confirm that the proposed method shows promising performance compared to other multimodal methods and proper feature selection further improves classification accuracy. Moreover, the proposed method gives the most reliable prediction for a new observation compared to other multimodal data learning methods.

INDEX WORDS: Multimodal data, Deep neural network models, Classification, Alzheimer's disease

Learning Multimodal Data using Deep Neural Networks in Classification

by

Eunsil Seok

B.S., Sungkyunkwan University, Republic of Korea, 2013

B.E., Sungkyunkwan University, Republic of Korea, 2013

M.S., Seoul National University, Republic of Korea, 2015

A Dissertation Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Doctor of Philosophy

Athens, Georgia

2021

Learning Multimodal Data using Deep Neural Networks in Classification

by

Eunsil Seok

Major Professor: Nicole A. Lazar

Committee: Jeongyoun Ahn
Cheolwoo Park
Justin Strait

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
August 2021

Acknowledgments

I am grateful to a number of people who have supported the creation of this dissertation and have made my Ph.D. life beautiful in various ways. First and foremost, I would like to express my deepest appreciation to my advisor Prof. Nicole A. Lazar for her continuous support, motivation, and encouragement during my Ph.D. life.

My appreciation extends to the rest of my committee, Prof. Jeongyoun Ahn, Prof. Cheolwoo Park, and Prof. Justin Strait for their valuable advice and encouragement. In particular, I would like to express special gratitude to Prof. Cheolwoo Park for providing me various opportunities and insightful suggestions. I am also grateful to Prof. Lynne Seymour for her guidance and care as my supervisor. In addition, I thank the members of the Neuroimaging Data Analysis Group for their helpful comments and practical suggestions, and all of my colleagues and friends for their assistant and stimulating discussion.

Last but not least, I would like to thank my husband Yongchan, my parents, and my sister for their countless support and great patience at all times. Without their unconditional love, everything would have been impossible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many medical and computer vision applications, learning multimodal data has become increasingly widespread. There are several reasons for this. Different modalities provide complementary information and principled statistical inferences on multimodal data can improve our understanding of an event. In addition, it has become easier to collect multiple signals from different sources. To be more concrete, we give three real-world examples where multimodal data on an event are observed.

Alzheimer's disease (AD) classification The first example of learning multimodal data is Alzheimer's disease (AD) classification using different sources of information. The main goal of this problem is to make an early and accurate diagnosis of AD with information from multiple sources.

AD is the most common form of dementia diagnosed in people over 65 years of age. In 2019, 5.6 million Americans over the age of 65 are estimated to suffer from AD, and that number is expected to increase to 13.8 million by 2050 (Association, 2019; Hebert et al., 2013). However, current treatments can only decelerate the progression of AD but cannot cure a patient who already suffers from AD. Also, when a patient is in the phase of a prodromal form of AD which is a step that describes people who have symptoms of mild

brain malfunction but are able to perform everyday tasks, called mild cognitive impairment (MCI), it is often difficult to be diagnosed due to its very mild or insignificant symptoms. Therefore, developing a strategy to detect AD in the early stages before clinical manifestation is fundamentally important for timely treatment and delayed progression.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal multicenter study designed for the early detection of AD. The goal of the study is to detect AD at the earliest possible stage and identify biomarkers to enable tracking the progress of the disease. ADNI began in 2004 and has collected data on subjects in three different stages: AD patients, MCI subjects, and elderly controls, in other words, cognitively normal (CN) subjects. Different sources of data have been collected including neuroimaging data such as Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET), genetic data such as Single Nucleotide Polymorphisms (SNP), cognitive tests, cerebrospinal fluid (CSF) levels, and blood biomarkers as predictors of the disease.

Different data sources contain complementary information about AD. For example, MRI and PET contain anatomical and functional information about the brain, respectively. Also, SNP provides information about a patient's genetic AD risk factors and CSF levels measure two proteins that form abnormal brain deposits strongly linked to AD. However, different modalities are in different types and different dimensions. Some modalities such as imaging data and genetic data are high-dimensional, where the dimension is larger than the sample size. Also, signals from different sources are often highly correlated within and between modalities. Hence, selecting a set of important features from each modality to remove multicollinearity and noise, and considering the correlations to effectively combine information from multimodal data is necessary.

By considering the practical challenges of high-dimensionality and correlations within and between modalities, one can integrate information from different modalities effectively, which

leads to a deeper understanding and a better diagnosis of AD. Also, a reliable prediction for a new subject is crucial to make an accurate diagnosis.

Pig Stroke Project Another motivating example of multimodal data is the Pig Stroke Project from the Animal Science Department at the University of Georgia. The goals of this project are to identify biomarkers of stroke and their recovery patterns in the pig model and to differentiate the recovery patterns of stroke between the treated and non-treated groups.

In the experiment, researchers induced artificial strokes in 16 pigs. Subsequently, 7 pigs received treatment, while 9 did not. Throughout the experiment, the researchers collected multiple modalities of data including behavioral measurements, physiological measurements, and Diffusion Tensor Imaging (DTI) data from 3 days before the surgery to 12 weeks after the surgery.

The first modality, behavioral measurements, includes information on pigs' behaviors such as gait on the mat, object recognition as well as field exercises. For example, this includes the speed of pigs' gait, strength of the steps, time spent by pigs in front of novel and familiar objects, frequency of pigs' visit to objects, and velocity and duration of exercises. The second modality, physiological measurements, contains information directly measured from pigs' brains such as Apparent Diffusion Coefficient (ADC) values and Fractional Anisotropy (FA) values. ADC is a measurement of the magnitude of water molecules diffusion within a tissue, and FA is a scalar value that describes the degree of anisotropy of a diffusion process. Also, volumes of the left and right hemispheres, as well as changes in the volumes are recorded. The last modality, the DTI dataset is an image dataset that visualizes and quantifies the orientation and directional uniformity of water diffusion in brain tissue. Examples of the DTI dataset are illustrated in Figure 1.1. These image data contain information about the water flow in the pig's brains. Table 1.1 summarizes the eight datasets including description, variables, dimensions of each dataset, and types of measurement (physiological or behavioral). Three different modalities, behavioral measurements, physiological measurements, and DTI

dataset are in different types and different dimensions but they all contain information on pigs' stroke and their recovery.

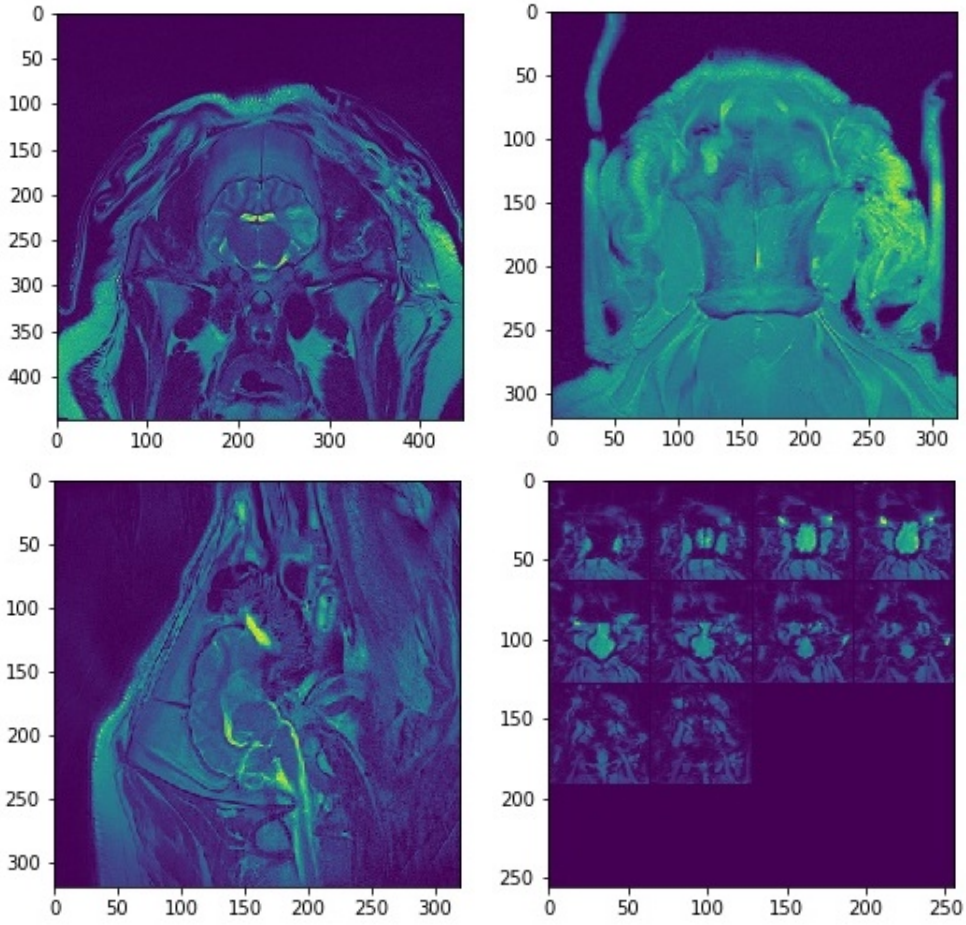

Figure 1.1: Examples of Diffusion Tensor Imaging (DTI) dataset. Different images show different directions of pigs' brains and visualize the water flow in the pigs' brains.

Table 1.1: Summary of eight datasets. Each dataset measures either behavioral or physiological features of pigs.

Dataset	Description	Variables	# variables	Feature
Gait	Gait on mats	Multiple measurement on gait for each day	65	behavioral
OR	Object recognition	Time spent and frequency at novel/familiar objects	6	behavioral
OF	Open field exercise	Duration and velocity	4	behavioral
ADC	Apparent Diffusion Coefficient	Ipsilateral/Contralateral measurement, slice 4-10	8	physiological
FA	Fractional Anisotropy from DTI	ROI 1-7	3	physiological
T2F	Tissue factor	Left/Right hemisphere and lesion volume	7	physiological
T2W	T2 weighted	Left/Right hemisphere, lesion volume and change	5	physiological
Weight	Weight of pigs	Weight of pigs	1	physiological

The researchers have collected all three modalities from 3 days before the artificial stroke up to 12 weeks after the surgery. Figure 1.2 illustrates the time points of the measured data. Here, treated pigs are depicted in red and non-treated pigs are depicted in yellow. Some of the pigs were alive until the end of the experiment while some died in the middle of the experiment. Points where no measurements are available are depicted in white which might be due to death of pigs or missing observations. These multiple measurements over time result in a number of variables that is much larger than the number of observations, which is only 16. Thus, the challenges of learning multimodal data in this project are missing data, high-dimensionality, and the correlations within and between modalities.

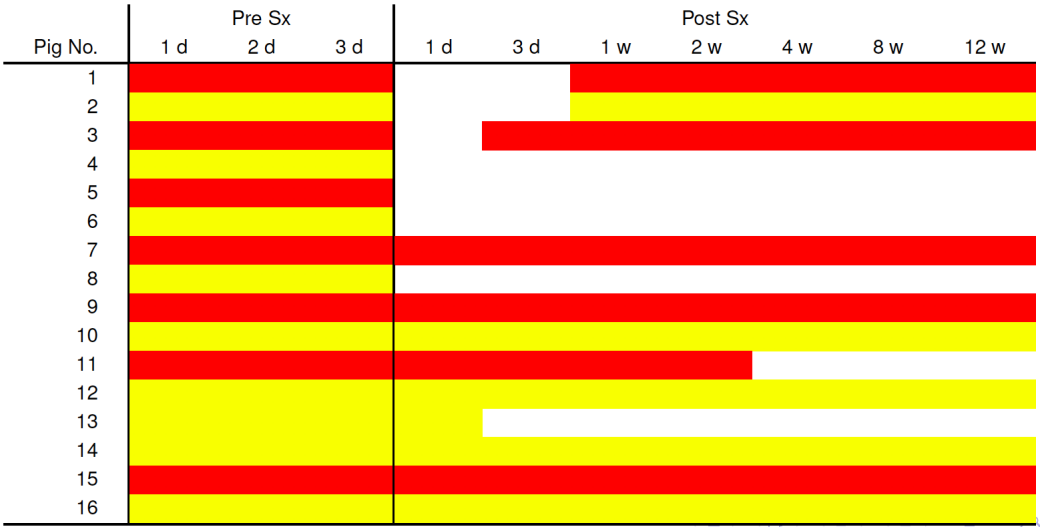

Figure 1.2: Pigs were observed from 3 days before the surgery up to 12 weeks after the surgery. Treated and non-treated are depicted in red and yellow, respectively, while white indicates missing observations.

Audio-visual speech classification Another example of learning multimodal data is audio-visual speech classification (Patterson et al., 2002; Matthews et al., 2002). Information on a speech is collected in both audio data and video data. In this way, audio data gives information on intonation, speed, and tone of the voice while video data can convey additional

information, such as gestures, facial expressions, lip movements, and more. Since both modalities include information on the same speech, correlation between audio and video data is inevitable. At the same time, the two modalities are in different types and dimensions, and one modality could have some information that the other does not have. So, capturing high-level features that contain more information than each modality leads to a better understanding of the speech (Ngiam et al., 2011).

The three real-world examples present common problems of multimodal data learning: data from multiple sources are in different types and dimensions, often high-dimensional, and variables are correlated to each other. By the nature of the multimodal data, these correlations are inevitable, but ignoring them may lead to inefficient methods and even can lead to inaccurate analysis. Accordingly, it is of critical importance to integrate information from different sources correctly and effectively in order to have a deeper insight into the event.

Learning multimodal data problems have been studied in the literature (Snoek et al., 2005; Gunes and Piccardi, 2005), and recently, many previous works are extended to exploit deep neural network models (Ordóñez and Roggen, 2016; Costa et al., 2017). In particular, learning multimodal data using deep neural networks models has been applied in various real-world applications such as audio-visual speech classification (Ngiam et al., 2011; Wöllmer et al., 2010), gesture recognition (Neverova et al., 2015), video description generation (Jin and Liang, 2016), and AD classification (Suk and Shen, 2013; Suk et al., 2014; Lu et al., 2018; Lee et al., 2019a,b).

We elaborate on the state-of-the-art learning methods. Zhou et al. (2017) suggested a three-step method to extract high-level features from multimodal data by considering pairwise combinations of modalities. But they did not consider feature selection in their method even though they dealt with high-dimensional datasets. Liu et al. (2018) investigated a new way to multiplicatively combine information that focuses more on reliable modalities

while reducing the emphasis on less reliable modalities. However, the computational cost is potentially expensive since they considered all possible combinations of modalities, giving $2^M - 1$ combinations with M modalities. Choi and Lee (2019) suggested a method that takes into account cross-modality correlations and missing data but employs randomness in information integration, which might give unstable performance. We provide more details on the three methods in Section 2.3.2.

In this dissertation, we study effective learning methods to handle high-dimensionality and correlation within and between modalities in classification settings. We propose new combining methods that take the importance of each modality into account and weight modalities based on their uncertainty for prediction. We demonstrate the practical efficacy of the proposed method on two problems: digit classification and AD classification. We confirm that the proposed method shows promising performance compared to other multimodal methods and we empirically show that the proposed method even has a type of synergy effect with feature selection so that it achieves the best accuracy among all combinations of settings. Furthermore, the proposed method gives the most reliable prediction with a new observation compared to other multimodal methods.

The remainder of the dissertation is organized as follows. In Chapter 2, we describe the multiclass classification and classification using deep neural networks including models, optimization, and Bayesian neural networks. Also, we review previous literature on methods to combine multimodal data using traditional statistical models and deep neural network models. In Chapter 3, we propose a framework for learning high-dimensional multimodal data including feature selection, information integration, and prediction interval. Also, we propose new combining methods for effective information integration. Chapter 4 illustrates the numerical experiments carried out for digit classification and AD classification, and provides results comparing models in Chapter 3. Finally, we conclude with remarks and future directions in Chapter 5.

Chapter 2

Learning Multimodal Data Using Deep Neural Networks in Classification

In this chapter, we give an overview of deep neural network models in classification settings and review previous literature on methods of multimodal learning. We first formally define the multiclass classification problem in Section 2.1. In Section 2.2, we describe deep neural network models, optimization process for the models, and Bayesian neural networks. Also, we introduce a dropout layer and describe optimization of neural networks with dropout layers in the context of approximate Bayesian variational inference. In Section 2.3, we review previous literature on methods of learning multimodal data using traditional statistical models and deep neural network models.

2.1 Multiclass Classification

In this section, we formally define the multiclass classification. Given a training data set $\mathcal{D}_{\text{tr}} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where input $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and output $y_i \in \mathcal{Y} := \{1, \ldots, K\}$

for $i \in [n] := \{1, \ldots, n\}$ and $d, K \in \mathbb{N}$. We suppose that every training sample (x_i, y_i) is independent and identically distributed (i.i.d.) from some underlying data distribution $P_{X,Y}$ defined on $\mathcal{X} \times \mathcal{Y}$. Let $f : \mathcal{X} \to \mathbb{R}^K$ be a score function, *i.e.*, $f(x) = (f_1(x), \ldots, f_K(x))$ and $f_i(x) \in \mathbb{R}$ denotes a score for the class $i \in \mathcal{Y}$. Here, we assume that the higher the score is, the more likely the corresponding class. We assume that we find the score function f in some function class $\mathcal{F}$. For example, $\mathcal{F}$ can be a class of logistic models that output probability estimates or a class of deep neural networks. Given the score function $f \in \mathcal{F}$, a classifier $s_f : \mathcal{X} \to \mathcal{Y}$, defined as $s_f(x) = \text{argmax}_{1 \le k \le K} f_k(x)$, predicts an outcome y for a new input x.

The performance of the classifier s_f, or equivalently the score function f, is evaluated through the misclassification error defined as $P_{X,Y}(Y \neq s_f(X))$. All together, the goal of multiclass classification is to find a model $\hat{f}$ using $\mathcal{D}_{\text{tr}}$ that minimizes the misclassification error, *i.e.*,

$$\hat{f} := \text{argmin}_{f \in \mathcal{F}} P_{X,Y}(Y \neq s_f(X)). \tag{2.1}$$

For the zero-one loss function $\ell_{01}(Y_1, Y_2) := I(Y_1 \neq Y_2)$ and an indicator function $I(\cdot)$, Equation (2.1) can be expressed as follows:

$$\hat{f} := \text{argmin}_{f \in \mathcal{F}} \mathbb{E}_{X,Y}[\ell_{01}(f(X), Y)],$$

where $\mathbb{E}_{X,Y}$ denotes the expectation taken over the distribution $P_{X,Y}$.

However, the objective function (2.1) is not feasible meaning that (sub)gradient methods cannot be applied because the zero-one loss ℓ_{01} is discontinuous and non-convex. To address this problem, a convex loss function is used as a proxy to the zero-one loss, and we call this function a surrogate loss. In case of binary classification problems[1], typical surrogate losses include the hinge loss $\ell_{\text{hinge}}(f(x), y) = \max(0, 1 - yf(x))$ and the logistic loss $\ell_{\text{logistic}}(f(x), y) =$

[1]Here, we slightly abuse notations. We consider $y \in \{-1, 1\}$ rather than $\{0, 1\}$, and we assume that $f : \mathcal{X} \to \mathbb{R}$ denotes a score function whose final prediction is defined as $I(f(x) > 0)$.

$1/(1+\exp(-yf(x)))$ (Hastie et al., 2009; Steinwart and Christmann, 2008). In case of multiclass classification, the cross-entropy loss is commonly used: for a true label y and a score vector $f(x;\theta) = (f_1(x;\theta), \ldots, f_K(x;\theta))^T$, the cross-entropy loss ℓ_{CE} is defined as

$$\ell_{\mathrm{CE}}(f(x;\theta), y) := -\sum_{k=1}^{K} I(y=k) \log f_k(x;\theta).$$

It is well known that a surrogate loss minimizer can minimize the zero-one loss under mild assumptions. To be specific, for a convex loss function ℓ, if ℓ is differentiable at 0 and $\ell'(0) < 0$, then ℓ is classification-calibrate (Bartlett et al., 2006; Ben-David et al., 2012) and the hinge, the logistic, and the cross-entropy loss functions satisfy this. All together, for a surrogate loss function ℓ, we optimize the surrogate objective function:

$$\hat{f} := \mathrm{argmin}_{f \in \mathcal{F}} \mathbb{E}_{X,Y}[\ell(f(X), Y)].$$

In practice, since we do not know the underlying data distribution $P_{X,Y}$, using the training dataset $\mathcal{D}_{\mathrm{tr}}$, we minimize an empirical risk given by

$$\hat{f} := \mathrm{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i). \tag{2.2}$$

Throughout this dissertation, we use the cross-entropy function ℓ_{CE} for the loss function since we consider multiclass classification problems.

2.2 Deep Neural Networks

2.2.1 Models

In this section, we elaborate on a set of deep neural network models. To be more specific, we denote the number of hidden layers by $L \in \mathbb{N}$, and for $l \in [L]$, we denote the number of

nodes in the l-th layer by $d_l \in \mathbb{N}$. For notational convenience, we regard the input and output layers as the 0-th and $(L+1)$-th hidden layer, respectively. That is, the dimensionalities of input and output layers are $d_0 := d$ and $d_{L+1} := K$, respectively. For $l \in [L+1]$, we define the l-th hidden layer $h_l \in \mathbb{R}^{d_l}$ recursively as

$$h_l = g_l(h_{l-1}) = \sigma_l(W_l h_{l-1} + b_l)$$

where $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ are weight matrices, $b_l \in \mathbb{R}^{d_l}$ are biases, and $\sigma_l(\cdot) : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ are nonlinear functions, also known as activation functions. For $\theta = \{(W_1, b_1), \ldots, (W_{L+1}, b_{L+1})\}$, a deep neural network score function $f(\cdot; \theta) : \mathcal{X} \to \mathbb{R}^K$ defined as a composition function of simple nonlinear functions is given by

$$\begin{aligned} f(x;\theta) &:= h_{L+1}(x) \\ &= g_{L+1} \circ \cdots \circ g_1(x) \\ &= \sigma_{L+1}(W_{L+1}(\cdots \sigma_1(W_1 x + b_1) \cdots) + b_{L+1}). \end{aligned}$$

As for the activation function, a univariate function is often applied in an element-wise manner, *i.e.*, for $l \in [L+1]$, $z_l \in \mathbb{R}^{d_l}$, and a univariate function $\sigma_{\text{uni},l} : \mathbb{R} \to \mathbb{R}$, the element-wise operations in hidden layers are performed as follows $\sigma_l(z_l) := (\sigma_{\text{uni},l}(z_{l,1}), \ldots, \sigma_{\text{uni},l}(z_{l,d_l}))$. For notational convenience, we suppress the subscript notation σ_{uni} into σ when contexts are clear. Here, choice of activation function affects the performance of the model. Some frequently used activation functions are shown in Table 2.1. For the Leaky ReLU function, α is a predefined hyperparameter, typically $\alpha = 0.01$ (Maas et al., 2013).

Using the softmax activation function in the last layer of the neural network model calculates the probabilities of each class. For this reason, throughout this dissertation, we

Table 2.1: Examples of frequently used activation functions.

Name	Activation function
Sigmoid	$\sigma(z) = \frac{1}{1+\exp(-z)}, \quad \forall z \in \mathbb{R}$
Hyperbolic tangent	$\sigma(z) = \tanh(z), \quad \forall z \in \mathbb{R}$
ReLU (Rectified Linear Unit)	$\sigma(z) = \max(0, z), \quad \forall z \in \mathbb{R}$
Leaky ReLU	$\sigma(z) = zI(z > 0) + \alpha z I(z \leq 0), \quad \forall z \in \mathbb{R}$
Softmax	$(\sigma(z))_k = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)} \quad \forall z \in \mathbb{R}^K, \quad \forall k \in [K]$

use the softmax activation function in the output layer. Hence,

$$f_k(x; \theta) = \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}, \quad \forall k \in [K]$$

where $z = W_{L+1}(g_L \circ \cdots \circ g_1(x)) + b_{L+1} \in \mathbb{R}^K$. Combining all, for $\theta = \{(W_1, b_1), \ldots, (W_{L+1}, b_{L+1})\}$, we define a set of unregularized deep neural network models as

$$\tilde{\mathcal{F}}_{\text{DNN}} := \{f(x; \theta) = \sigma_{L+1}(W_{L+1}(\cdots \sigma_1(W_1 x + b_1) \cdots) + b_{L+1})\}. \tag{2.3}$$

Since the function space in (2.3) is often too big, weights are regularized for learnability of the model (Schmidt-Hieber, 2020). Throughout this dissertation, we consider a set of models as follows:

$$\mathcal{F}_{\text{DNN}} := \{f(x; \theta) \in \tilde{\mathcal{F}}_{\text{DNN}} : \max_{l=1,\cdots,L+1} \|W_l\|_\infty \vee |b_l|_\infty \leq 1\}, \tag{2.4}$$

where $\|\cdot\|_\infty$ denotes maximum-entry norm, and $a \vee b = \max(a, b)$. This bounds network parameters by one.

In general, deep neural network models are flexible to different kinds of data and can be adapted to different problems. Also, massive parallel computations can be performed by using GPUs, and deep neural network models deliver better performance results compared to other traditional models when the amount of data is huge. In contrast, deep neural network

models are extremely expensive to train due to their complexity and require a lot of data to train. Also, overfitting is a major problem in neural network models particularly when the depth gets larger, since the number of parameters increases exponentially.

Example 1. *For a simple example, an illustration of a deep neural network with* $L = 2$, $d = 3, d_1 = 4, d_2 = 4, K = 2$ *is given in Figure 2.1. That is, dimensionalities of input and output layers are three and two, respectively, and two hidden layers have four nodes each. When the input value is given, it goes through a linear transformation and an activation function. Then, for each hidden layer, the linear transformation and activation function produce an output which is the input of the next hidden layer. For the output layer, in classification problem, a softmax function is used.*

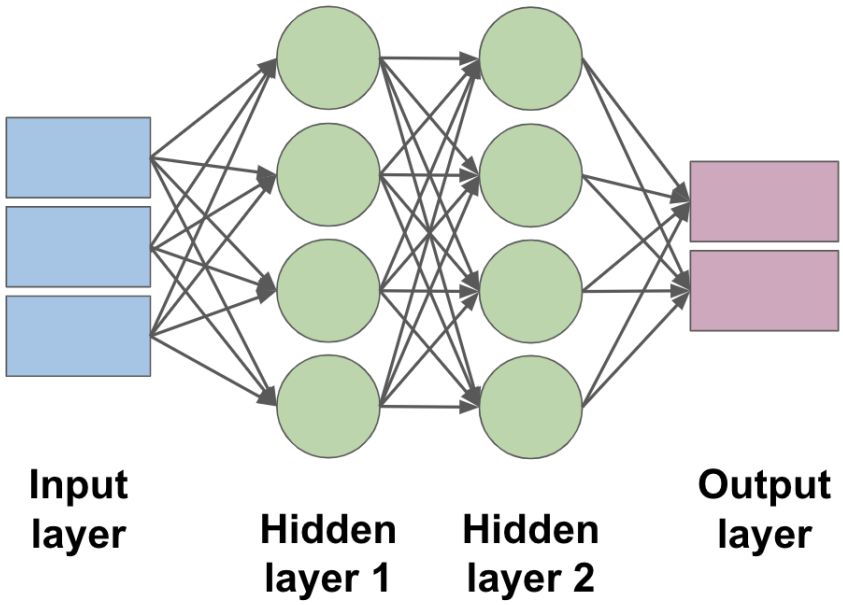

Figure 2.1: An illustration of a neural network model with two hidden layers with $d = 3, d_1 = 4, d_2 = 4$, and $K = 2$.

2.2.2 Optimization of Deep Neural Networks

As in Equation (2.2), the goal of the empirical risk minimization is to find an optimal parameter θ that minimizes the empirical loss function. When training deep neural network models, one of the common optimization algorithms is gradient descent. Gradient descent is an iterative method to minimize an objective function $\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i;\theta),y_i)$ by updating the current estimate in the opposite direction of the gradient of the object function $\nabla_\theta\Big(\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i;\theta),y_i)\Big)$. To be more specific, let θ_t be the t-th time estimate. For a learning rate $\eta > 0$, one step update of the gradient descent algorithm is expressed as follows:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta\Big(\frac{1}{n}\sum_{i=1}^{n}\ell(f(x_i;\theta),y_i)\Big). \tag{2.5}$$

Here, the learning rate is the size of the steps to take to reach a minimum.

The iteration (2.5) calculates a gradient of the empirical error and moves down along that gradient towards some minimum. In the gradient descent optimization, we compute the gradient based on the training set $\mathcal{D}_{\text{tr}}$. Hence, its computational cost is expensive for a large dataset. The larger the training set is, the slower the updates are. To solve this problem, the Stochastic Gradient Descent (SGD), where its update is computed using only one observation (x_i, y_i), is typically used. To be more specific, one step update of SGD is given by

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta\ell(f(x_i;\theta),y_i).$$

The SGD has strength in the computational cost but it is noisy because it only uses one observation. In practice, a combination of gradient descent and SGD is used: mini-batch stochastic gradient descent.

The mini-batch stochastic gradient descent splits the training dataset into small batches that are used to update model parameters. For mini-batch stochastic gradient descent, a fixed number of training examples, less than the actual dataset, is used. Here, the fixed number

is called batch size, and a set of the samples is a mini-batch. After randomly sampling the mini-batches S from $\{1, .., n\}$, we update the parameters as follows:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \Big(\frac{1}{|S|} \sum_{i \in S} \ell(f(x_i; \theta), y_i) \Big).$$

Mini-batch stochastic gradient descent seeks to find a balance between the stability of SGD and the efficiency of gradient descent. Mini-batch stochastic gradient descent is the most common implementation of gradient descent used in deep learning. The algorithm is described in Algorithm 1.

Algorithm 1 Mini-batch stochastic gradient descent

1: **Inputs:** a dataset $\mathcal{D}_{\text{tr}}$, a learning rate $\eta > 0$, a batch size B, a loss function ℓ.
2: Initialize parameters θ (e.g. He et al. (2015)).
3: **repeat**
4: Sample a subset $S \in [n]$ with $|S| = B$ uniformly at random.
5: Calculate gradient with respect to θ:

$$\widehat{\Delta\theta} = \frac{1}{|S|} \sum_{i \in S} \nabla_\theta \ell(f(x_i; \theta), y_i)$$

6: Update θ:

$$\theta \leftarrow \theta - \eta \widehat{\Delta\theta}$$

7: **until** θ has converged or for epoch size.

2.2.3 Bayesian Neural Networks

In this section, we describe Bayesian neural networks and two Bayesian inference methods. Also, we introduce a dropout layer and describe optimization of neural networks with dropout layers in the context of approximate Bayesian variational inference.

First, the neural network model $f(x; \theta)$ in Equation (2.4) gives the model likelihood $p(y = k \mid x, \theta) = f_k(x; \theta)$. Assuming the Bayesian neural network model, we define a prior distribution $p(\theta)$ on parameter $\theta \in \Theta$ including weights and bias in a neural network.

Applying Bayes Theorem, a posterior distribution can be written as:

$$p(\theta \mid \mathcal{D}_{\text{tr}}) \;=\; \frac{p(\mathcal{D}_{\text{tr}} \mid \theta)p(\theta)}{p(\mathcal{D}_{\text{tr}})} = \frac{\prod_{i=1}^{n} p(y_i \mid x_i, \theta)p(\theta)}{\int_{\Theta} \prod_{i=1}^{n} p(y_i \mid x_i, \theta)p(\theta)d\theta}.$$

Then, given the posterior distribution, the predictive distribution for a new input x^* and a new output y^* is:

$$p(y^* \mid x^*, \mathcal{D}_{\text{tr}}) = \int_{\Theta} p(y^* \mid x^*, \theta)p(\theta \mid \mathcal{D}_{\text{tr}})d\theta. \tag{2.6}$$

Learning this predictive model is often intractable since calculating the posterior $p(\theta \mid \mathcal{D}_{\text{tr}})$ requires integration with respect to the whole parameter space Θ, and that often does not have a closed form. Accordingly, two methods have been commonly used for training Bayesian neural network models: Markov Chain Monte Carlo (MCMC, Robert and Casella (2013)) and variational inference.

Firstly, MCMC is a strategy for generating samples while exploring the state space using a Markov chain mechanism. This mechanism is constructed so that the chain spends more time in the important regions. In particular, it is constructed so that the generated samples mimic samples drawn from the target distribution. For the MCMC algorithm to train Bayesian neural network models, Neal (1993) proposed the Hamiltonian Monte Carlo, an MCMC sampling approach that uses Hamiltonian dynamics. This yields a principled set of posterior samples without direct calculation of the posterior. However, this method is often computationally expensive since MCMC sampling should be done on a entire dataset and the sampled parameters need to be stored for inferences. More practical MCMC methods, by using stochastic optimization techniques, have been proposed recently (Welling and Teh, 2011; Chen et al., 2014) based on the first- or second-order Langevin dynamics. Nevertheless, these approaches might be inefficient in terms of memory usage because they require storing all sampled parameters.

Another common approach that is more efficient is variational inference (Graves, 2011). Variational inference uses approximation to the posterior distribution by a tractable variational distribution $q_\phi(\theta)$ with variational parameter $\phi \in \Phi$ for variational parameter space Φ. The optimal variational distribution is found by minimizing the Kullback-Leibler (KL) divergence between the variational distribution $q_\phi(\theta)$ and the posterior $p(\theta \mid \mathcal{D}_{\text{tr}})$, defined by

$$KL\{q_\phi(\theta) \parallel p(\theta \mid \mathcal{D}_{\text{tr}})\} := \int_\Theta q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta \mid \mathcal{D}_{\text{tr}})} d\theta. \tag{2.7}$$

Minimizing the KL divergence is equivalent to minimizing the negative evidence lower bound (ELBO) given by

$$-\int_\Theta q_\phi(\theta) \log p(y \mid x, \theta) d\theta + KL\{q_\phi(\theta) \parallel p(\theta)\}. \tag{2.8}$$

Note that the variational inference (2.8) transforms standard Bayesian learning from integration to an optimization problem. Due to this transformation, Bayesian inference can be performed by using a mini-batch stochastic gradient descent algorithm.

Dropout (Srivastava et al., 2014) is a layer to regularize a neural network by randomly dropping units during training. A dropout layer can be added to any type or structure of model or training procedure. This solves the overfitting problem, saves computational cost by regularizing models, and reduces network generalization error (Goodfellow et al., 2016).

We let y_i be the observed output corresponding to input x_i, and for $i \in [n]$ and $\theta = \{(W_1, b_1), \ldots, (W_{L+1}, b_{L+1})\}$, $f(x_i; \theta)$ be the output of a neural network model with L hidden layers and the cross-entropy loss function ℓ_{CE}. Denote by $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_l \in \mathbb{R}^{d_l}$ the neural network's weight matrices and the bias vectors for each layer $l \in [L+1]$, respectively. We often use L_2 regularization for the minimization objective, and the resulting objective function is

$$\mathcal{L}_{\text{dropout}}(\theta) := \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{CE}}(f(x_i; \theta), y_i) + \lambda \sum_{l=1}^{L+1} (\|W_l\|_2^2 + \|b_l\|_2^2) \tag{2.9}$$

where $\lambda > 0$ is the regularization parameter. With dropout layers, one samples binary random variables for every input and network unit in each layer. Every binary input takes value 1 with probability p_l for layer $l \in [L]$. If the binary variable takes value 0, the corresponding node is dropped, and it is ignored during the backpropagation (Rumelhart et al., 1986).

We now describe that neural networks with dropout layers whose optimization using mini-batch stochastic gradient descent is equivalent to Bayesian variational inference (Gal and Ghahramani, 2016). Deep neural networks have been widely used in different fields in applied machine learning but one weakness is that they do not capture model uncertainty. On the other hand, Bayesian models provide a framework for assessing model uncertainty. Gal and Ghahramani (2016) show that typical optimization of neural networks with dropout layers is equivalent to Bayesian variational inference with a specific variational distribution. Furthermore, by randomly sampling Bernoulli random variables in dropout layers, one could obtain a variational predictive distribution.

To be more specific, use a tractable variational distribution $q_\phi(\theta)$ whose columns are randomly set to zero, *i.e.*, we define $q_\phi(\theta)$ as:

$$W_l = M_l \cdot \text{diag}([z_{l,j}]_{j=1}^{d_{l-1}})$$

where $z_{l,j} \sim \text{Bernoulli}(p_l)$ for $l \in [L+1]$ and $j \in [d_{l-1}]$, given some probability p_l, and matrices M_l are variational parameters in $\mathbb{R}^{d_l \times d_{l-1}}$. In this way, when the binary variable has value 0, $z_{l,j} = 0$, then the unit j in the $(l-1)$-th layer is dropped out as an input to the l-th layer. Gal and Ghahramani (2016) prove that given $q_\phi(\theta)$, the KL divergence minimization in (2.7) is equivalent to the dropout minimization in (2.9).

2.3 Literature Review

In this section, we review previous literature on methods of learning multimodal data with traditional statistical models and deep neural network models.

2.3.1 Learning Multimodal Data Using Traditional Statistical Models

The most widely used techniques for multimodal data learning are early fusion and late fusion (Snoek et al., 2005; Gunes and Piccardi, 2005). They differ by when to combine the information available in the multimodal data: either early with raw data or late with extracted feature representations. In the early fusion method, information from different modalities is merged at the signal or feature level and combined information is considered as one input (Snoek et al., 2005; Potamianos et al., 2001). On the other hand, in late fusion, higher semantic level information is combined. The late fusion method constructs separate classifiers for each modality, trains the classifiers independently, and draws a final decision by combining the outputs of the classifiers (Verma and Tiwary, 2014). Although both of these frameworks are intuitive, they often ignore the correlation between modalities by concatenating either variables or representations from multiple sources.

To capture the correlation between modalities for different levels of fusion, various techniques have been used for information integration. For example, multi-kernel learning (MKL) based multimodal learning (Gehler and Nowozin, 2009; Gönen and Alpaydın, 2011; Bucak et al., 2013), linear weighted fusion (Yan et al., 2004), support vector machine (SVM, Iyengar and Nock (2003); Wu et al. (2004b,a); Zhu et al. (2006)), and majority voting (Radová and Psutka, 1997; Sanderson and Paliwal, 2004) are used for combining information from multiple sources. Most of the studies consider audio, visual, and text information and more recently, multimodal data learning has been used for disease diagnosis and prediction.

In particular, many studies have shown that integrating the complementary information from multimodal data helps enhance the AD diagnostic accuracy (Perrin et al., 2009; Kohannim et al., 2010; Walhovd et al., 2010; Hinrichs et al., 2011; Zhang et al., 2011; Zhang and Shen, 2012; Wee et al., 2012; Dai et al., 2012; Gray et al., 2013; Liu et al., 2014). Different techniques to combine information has been used such as SVM (Kohannim et al., 2010; Zhang et al., 2011; Zhang and Shen, 2012; Wee et al., 2012), MKL (Hinrichs et al., 2011; Liu et al., 2014), and random forest (Gray et al., 2013).

As datasets such as neuroimaging data and genetic data that contain information about AD are often high-dimensional, feature selection has been considered in multimodal data learning. Dai et al. (2012) suggested a method called multi-modal imaging and multi-level characteristics with multi-classifier (M3), which uses feature selection and linear discriminant analysis based method. Zhu et al. (2014) proposed a canonical feature selection method that integrates the ideas of canonical correlation analysis (CCA, Hotelling (1936)) and a sparse multi-task learning (MTL, Liu et al. (2012)) into a unified framework.

2.3.2 Learning Multimodal Data Using Deep Neural Network Models

Recently, deep neural networks are actively used in solving multimodal problems. Many previous works including early fusion and late fusion are extended to exploit deep neural network models (Ordóñez and Roggen, 2016; Costa et al., 2017). In particular, learning multimodal data using deep neural network models has been applied in various real-world applications such as audio-visual classification (Ngiam et al., 2011; Wöllmer et al., 2010), gesture recognition (Neverova et al., 2015), video description generation (Jin and Liang, 2016), and AD classification (Suk and Shen, 2013; Suk et al., 2014; Lu et al., 2018; Lee et al., 2019b,a).

We elaborate on three state-of-the-art learning methods. Firstly, Zhou et al. (2017) consider AD classification problem to predict different statuses of AD (CN, MCI, and AD), using MRI, PET, and SNP modalities. They suggest a three-step method to learn high-level features for different combinations of modalities to make use of the maximum number of available samples. The architecture of their method is in Figure 2.2. In the first stage, with each modality, high-level features are learned by their own neural networks independently. The role of the first step is to solve the data heterogeneity issue by learning latent features through multiple hidden layers to extract high-level features. Then, in the second step, joint features for all combinations of each modality are learned by combining two modalities at a time. With three modalities, MRI, PET and SNP, the three possible combinations are MRI-PET, MRI-SNP and PET-SNP. Learning joint features from each combination improves the performance of the model by fusing complementary information from the different modalities. At the last step, by fusing the learned joint features from the second stage, prediction on diagnostic labels can be made. Their method learns the latent features for different combinations of modalities, using the maximum number of available samples. However, even though they deal with high-dimensional datasets, they do not consider feature selection.

Liu et al. (2018) suggest a method that multiplicatively combines information from different modalities. Their main goal is to differentiate good modalities that contain more information, from bad modalities that contain less information than the others. They consider different mixtures of modalities by creating all possible combinations of modalities. That is, for M modalities, they consider the $2^M - 1$ possible combinations. On each mixture of modalities, they extract high-level feature representations denoted as p_c for $c \in [2^M - 1]$ as follows,

$$p_c = \sum_{s \in M_c; M_c \subset \{1, \cdots, M\}} f(x_s)$$

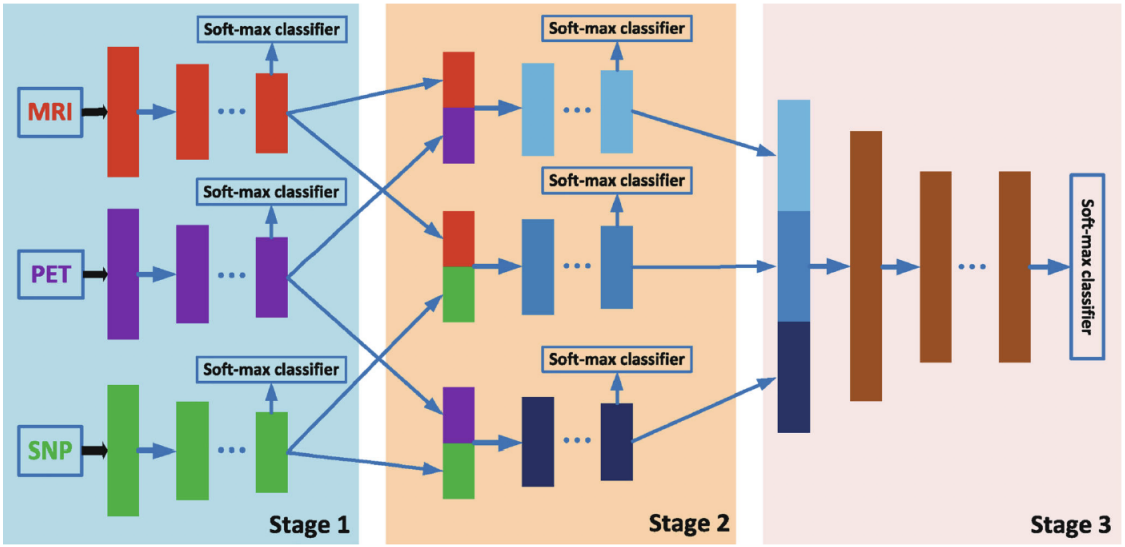

Figure 2.2: An illustration of multimodal data learning architecture suggested by Zhou et al. (2017). This figure is from Figure 1 of Zhou et al. (2017).

where $f(\cdot) : \mathbb{R}^{|M_c|} \to \mathbb{R}^K$ indicates a neural network model with input x_s, the set of variables in M_c. Thus, p_c is the representation of the mixture of modalities in set M_c, that gathers signals from all modalities in M_c.

Now they combine information from different combinations of modalities p_c, by giving weights q_c for $c \in [2^M - 1]$ that are calculated as follows:

$$q_c^{(k)} = \left(\prod_{j \neq c} (1 - p_j^{(k)}) \right)^{\beta/(2^{M-1})}$$

where the superscript (k) denotes the true class index for $k \in [K]$ and β is a hyper parameter to control the strength of weights. Then the final objective function is

$$\ell^{(k)} = -\sum_{c=1}^{|M_c|} q_c^{(k)} \log p_c^{(k)}$$

for $k \in [K]$ and this objective function is part of the loss function associated with a particular class. The model predicts the class with the smallest class loss, *i.e.*,

$$\hat{y} = \arg\min_k \ell^{(k)}.$$

In this method, they consider all possible signal correlation and complementariness across modalities by taking account of all possible combinations of modalities. However, some combinations might be noisy or contain redundant information. Also, the computation cost is potentially expensive since they consider all $2^M - 1$ combinations with M modalities.

Choi and Lee (2019) suggest a method to find a model called EmbraceNet that supports high compatibility with existing deep learning architectures and considers cross-modal correlations thoroughly. The suggested model has two parts, a docking layer and an embracement layer. The docking layer converts the feature representation from all modalities to the same dimension and the embracement layer combines information using randomness.

To be more specific, for $m \in [M]$, let $x^{(m)}$ be the input of the m-th modality; the m-th neural network $h_L^{(m)}$ has L hidden layers. Then, a docking layer for the m-th modality computes $d^{(m)}$ given by

$$d^{(m)} = \sigma_{L+1}(W_{L+1}^{(m)} h_L^{(m)}(x^{(m)}) + b_{L+1}^{(k)}) \in \mathbb{R}^c,$$

where $W_{L+1}^{(m)}$, $b_{L+1}^{(m)}$, and $\sigma_{L+1}^{(m)}$ are defined in a similar way as before, and c is the prespecified output size, called embracement size. From the docking layer, c-dimensional M vectors are obtained. The embracement layer combines these into a single vector.

In the embracement layer, firstly draw multinomial random variables. That is, let $R = [r^{(1)}, \ldots, r^{(M)}]$ be a $\mathbb{R}^{c \times M}$ matrix such that each row follows the multinomial distribution,

i.e., let $r_i \in \mathbb{R}^M$ be the i-th row of R, then

$$r_i \sim \text{Multinomial}(1, p),$$

where $p = (p_1, \cdots, p_M)^T$ is some probability vector. Note that only one value of r_i is 1 and the rest are 0. After that compute the embracement layer output e as

$$e = \sum_{m=1}^{M} r^{(m)} \bullet d^{(m)}, \tag{2.10}$$

where $\bullet$ is the Hadamard product. Lastly, obtained output vector e is passed through the terminal network, which outputs a final score in K-dimension for K-class classification. In the paper, Choi and Lee (2019) suggest $p = (1/M, \ldots, 1/M)^T$, which randomly selects a modality for each element of e. The architecture of the suggested model, EmbraceNet is illustrated in Figure 2.3.

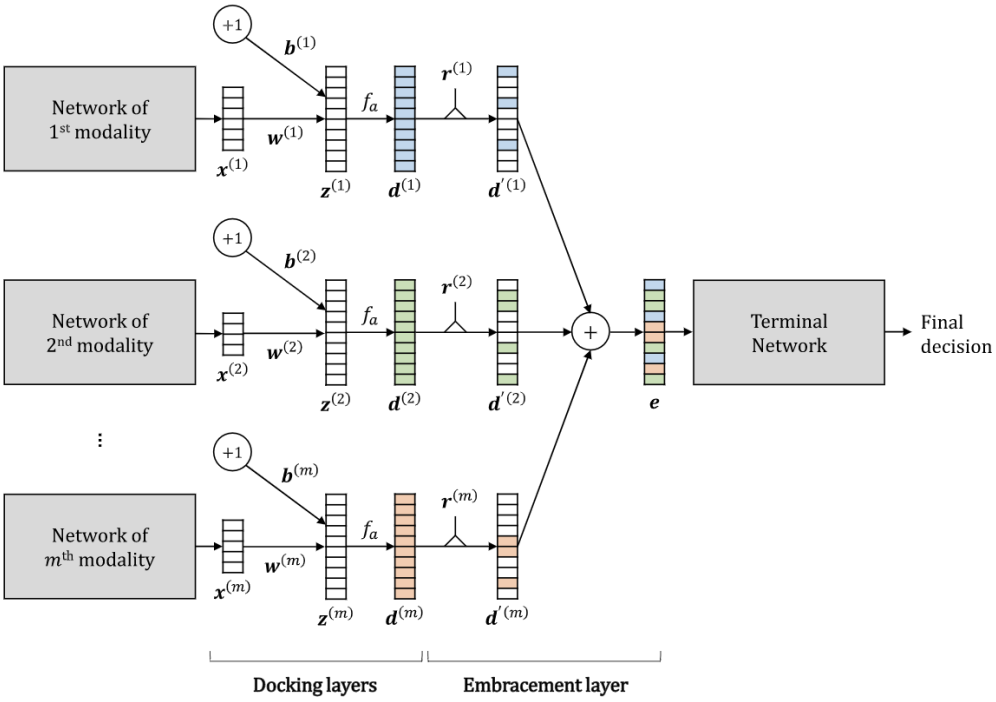

Figure 2.3: Architecture of the EmbraceNet suggested by Choi and Lee (2019). This figure is from Figure 1 of Choi and Lee (2019).

Chapter 3

A Proposed Framework for Learning Multimodal Data

In this chapter, we propose a framework for learning high-dimensional multimodal data. Our framework consists of three parts, (i) feature selection (Section 3.1), (ii) information integration (Section 3.2), and (iii) prediction intervals (Section 3.3). Also, we propose new combining methods for effective information integration in Section 3.2.

3.1 Feature Selection

As it has become easier to obtain big data, it is more and more common to face a large p small n setting where we have more variables than the number of observations. Dealing with high-dimensional data requires feature selection to remove multicollinearity, spurious correlation, noise, and more. We consider three different feature selection methods: Principal Component Analysis (PCA, Hotelling (1936)), Sure Independence Screening method (SIS, Fan and Lv (2008)), and Concrete Autoencoder (CA, Abid et al. (2019)).

Principal Component Analysis (PCA) The first method is Principal Component Analysis (PCA), a well-known dimension reduction technique that uses the dependencies between the variables to represent a dataset in a more tractable, lower-dimensional form, without losing too much information. The main idea is to find uncorrelated linear transformations of the variables which maximize variance. The first principal component (PC) is the direction in space along which the transformation has the largest variance. The second PC is the direction which maximizes variance among all directions orthogonal to the first. The r-th component is the variance-maximizing direction orthogonal to the previous $r-1$ components. That is, for a given $n \times d$ data matrix X, with rank $r < d$, we assume the column means of X are all zero. Then, for $i \in [r]$, find $a_i \in \mathbb{R}^d$ such that

$$\max_{i \in [r]} a_i^T X^T X a_i$$
$$\text{subject to } \|a_i\|_2^2 = 1 \text{ and } \|a_i^T a_j\| = 0 \text{ for } j \in [i-1].$$

There are r linear transformations of the original variables. Then, we can take the first p PCs that are enough to explain a large portion of the total variability (e.g. 85% or 90%), and this reduces the dimension of the data from d to p.

Sure Independence Screening method (SIS) The second method is the Sure Independence Screening method (SIS) proposed by Fan and Lv (2008). SIS selects a set of important variables by considering the marginal correlation between each variable with the response variable. Consider a linear regression model

$$y = X\beta + \epsilon \tag{3.1}$$

where $y = (y_1, \cdots, y_n)^T$ is an n-dimensional response vector, $X = (x_1, \cdots, x_d)$ is an $n \times d$ design matrix, $\beta = (\beta_1, \cdots, \beta_d)^T$ is a d-dimensional coefficient vector, and $\epsilon = (\epsilon_1, \cdots, \epsilon_n)^T$

is an n-dimensional error vector. SIS ranks the d features using marginal correlations and retains the p covariates with the largest absolute correlations in the set $\widehat{\mathcal{M}}$, *i.e.*,

$$\widehat{\mathcal{M}} = \{1 \leq j \leq d : |\widehat{\text{corr}}(x_j, y)| \text{ is among the top } p \text{ largest ones.}\} \tag{3.2}$$

where $\widehat{\text{corr}}(x_j, y) := \frac{\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)(y_k-\bar{y})}{\sqrt{\sum_{k=1}^{n}(x_{jk}-\bar{x}_j)^2}\sqrt{\sum_{k=1}^{n}(y_k-\bar{y})^2}}$ for $j \in [d]$ denotes the sample correlation, $\bar{y} := n^{-1}\sum_{k=1}^{n} y_k$, and $\bar{x}_j := n^{-1}\sum_{k=1}^{n} x_{jk}$.

When the response is a discrete variable, we use the generalized linear model (GLM) with a canonical link that assumes the conditional distribution of y given X belongs to the canonical exponential family. That means we assume the following density function:

$$\begin{aligned} f_n(y; X, \beta) &= \prod_{i=1}^{n} f_0(y_i; \theta_i) \\ &= \prod_{i=1}^{n} \left(c(y_i) \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi} \right) \right) \end{aligned} \tag{3.3}$$

where $\{f_0(y;\theta) : \theta \in \mathbb{R}\}$ is a family of distributions in the regular exponential family with dispersion parameter $\phi \in (0, \infty)$, $(\theta_1, \cdots, \theta_n)^T = X\beta$, $b(\cdot)$ and $c(\cdot)$ are known functions. The proposed SIS method for linear regression can be extended to the GLM setting by considering magnitude of the maximum marginal loglikelihood estimator $\hat{\beta}_j$, the maximizer of the quasi-loglikelihood function $l(\beta_j) = \log f_n(y; x, \beta_j)$ (Fan and Lv, 2018). Then, replace $\widehat{\text{corr}}(x_j, y)$ in the set of reduced variables $\widehat{\mathcal{M}}$, Equation (3.2), with $\hat{\beta}_j$, *i.e.*,

$$\widehat{\mathcal{M}} = \{1 \leq j \leq d : |\hat{\beta}_j| \text{ is among the top } p \text{ largest ones.}\}$$

It is noteworthy that various regularization methods can be incorporated in this step, Equation (3.3). For example, LASSO, SCAD, and Dantzig selector, to name a few (Tibshirani, 1996; Fan and Li, 2001; Candes and Tao, 2007; Bickel et al., 2009).

Concrete Autoencoder (CA) The concrete autoencoder is a variation of the standard autoencoder (Hinton and Salakhutdinov, 2006) for discrete feature selection. Let P_X be a probability distribution defined on $\mathcal{X} \subset \mathbb{R}^d$. The main goal of CA is to learn a set of important features S where $S \subseteq [d]$ with $|S| = p$ and a reconstruction function $f_\theta(\cdot) : \mathbb{R}^p \to \mathbb{R}^d$, that minimizes the expected loss between the reconstructed sample $f_\theta(x_S)$ and the original sample x, where $x_S \in \mathbb{R}^p$ consists of elements x_i such that $i \in S$, *i.e.*, find S such that:

$$\arg\min_{S,\theta} \mathbb{E}_{P_X}[\|f_\theta(x_S) - x\|_2], \tag{3.4}$$

where $\mathbb{E}_{P_X}$ denotes the expectation taken over the distribution P_X.

To find an optimal solution for Equation (3.4), Abid et al. (2019) propose a concrete selector layer that uses a specific random variable, called Concrete random variable. One should first sample a d-dimensional vector of i.i.d. samples, denoted by g, from a Gumbel distribution Gumbel (1954). Then, with parameters $\alpha \in \mathbb{R}^d_{>0}$, and $T \in (0, \infty)$, the j-th element of a Concrete random variable $m \in \mathbb{R}^d$, m_j is defined as:

$$m_j = \frac{\exp\left((\log \alpha_j + g_j)/T\right)}{\sum_{k=1}^{d}(\exp\left(\log \alpha_k + g_k\right)/T)}. \tag{3.5}$$

As the temperature $T \to 0$, the concrete random variable approaches the discrete distribution, resulting in one-hot vectors[1] with $m_j = 1$ with probability $\alpha_j / \sum_p \alpha_p$ for $j \in [d]$. That is, as the temperature T decreases to zero, each node in the concrete selector layer outputs exactly one of the input features. During a training phase, the model learns a set of important features as well as a reconstruction function $f_\theta(\cdot)$ that has the minimum expected loss with the original sample x.

We use three feature selection methods to remove high-dimensionality and correlation within and between modalities. By comparing different sizes of sets of selected features for

[1] A one-hot encoding gives a representation of categorical variables as binary vectors. Each integer value is represented as a binary vector that is all 0 values except the index of the integer, which is marked with a 1.

each method, we find the best set of important features in terms of classification accuracy for a multimodal model.

3.2 Combining Representation from Multimodal Data

In learning multimodal data from different sources, an effective information combining method leads to a better understanding about the event. In this section, we propose new combining methods for effective information integration, and we describe five different combining methods including two newly proposed methods.

Since we consider high-dimensional multimodal data, we assume that each modality is fed into a separate neural network — we call this the individual neural network — and the individual network outputs are combined. To combine outputs, we give a weight to each modality, where the weight reflects the importance of each modality.

To be more specific, let M be the number of modalities. For the output layer, we use a dense layer with output size the number of classes, K, and the softmax activation function. That is, we obtain a score vector at the end of each individual neural network, and each element is a probability estimate for each class. In other words, let $p^{(m)} \in \mathbb{R}^K$ for $m \in [M]$ be the output from the m-th individual neural networks. Since $p^{(m)}$ is a score vector, we denote each element by $p^{(m)} = (p_1^{(m)}, \cdots, p_K^{(m)})^T$ where $p_k^{(m)}$ indicates a probability estimate for the k-th class from the m-th modality. Then, we define a weight for each modality, w_m for $m \in [M]$ and integrate the M score vectors to obtain a combined score vector z_{comb} as follows:

$$z_{\text{comb}} = \sum_{m=1}^{M} w_m p^{(m)}. \tag{3.6}$$

That is, the output z_{comb} is a weighted average of the score vectors $p^{(m)}$. The combined score vector z_{comb} is used for the final classification decision. Figure 3.1 illustrates a general

structure for high-dimensional multimodal data learning. We do not assume any structure such as depth or kind of layers on the individual neural networks.

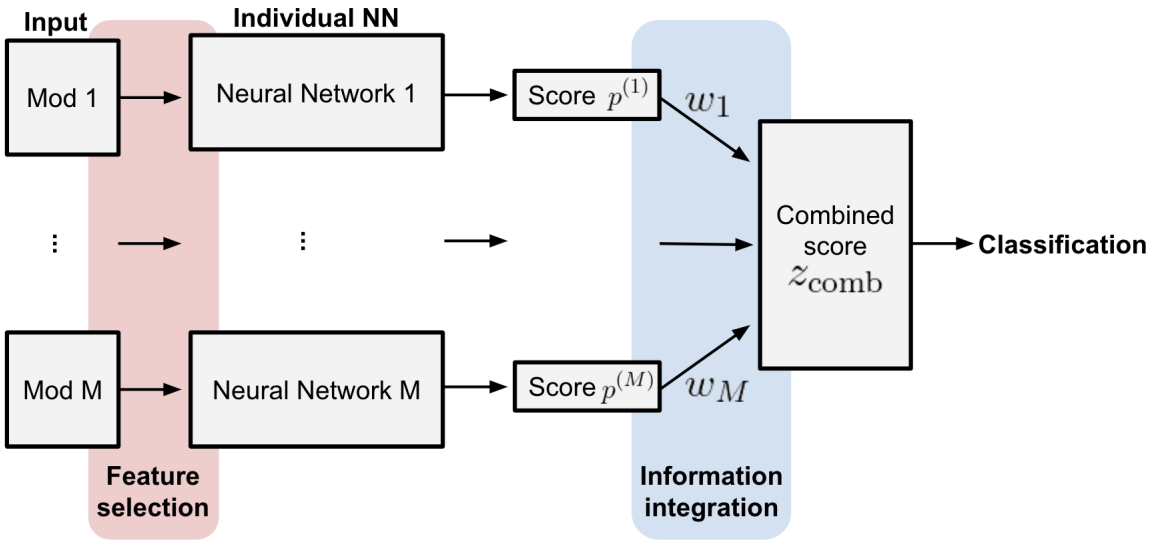

Figure 3.1: An illustration of the proposed framework for learning high-dimensional multimodal data.

A specification of w_m in Equation (3.6) is determined by a combining method. We consider five different combining methods: (1) Simple Average (SA), (2) Weighted Average (WA), (3) Concept Drift (CD), (4) Weighted Average Concept Drift with probability (WACDp), and (5) Weighted Average Concept Drift with entropy (WACDe).

Simple Average (SA) Simple Average (SA) method is to use uniform weights for all modalities, *i.e.*, for $m \in [M]$

$$w_m = \frac{1}{M}.$$

The combined score z_{comb} in Equation (3.6) is calculated as:

$$z_{\text{comb}} = \sum_{m=1}^{M} \frac{1}{M} p^{(m)}.$$

That is, the final score vector z_{comb} is a simple average of score vectors from each modality.

Weighted Average (WA) Weighted Average (WA) is a method to use predetermined weights derived from the accuracy of single modality models. This process consists of two steps: the weight computation step and the training step.

We obtain average accuracy of the single modality model by training a single modality model. For $m \in [M]$, let Acc_m be the classification accuracy of single modality models. We calculate a weight for each modality with the softmax function

$$w_m = \frac{\exp(\text{Acc}_m)}{\sum_{j=1}^{M} \exp(\text{Acc}_j)}$$

for $m \in [M]$. We fix the calculated weights in Equation (3.6) for training a multimodal model. Weights used in this method reflect the overall performance of single modality models.

Concept Drift (CD) CD is a method based on Wang et al. (2017). They suggest an ensemble model to handle concept drift, a phenomenon that the conditional distribution of an output given an input changes over time (Bartlett, 1992; Kelly et al., 1999; Yang et al., 2005). They consider data that are collected in a sequential manner, called batch, and propose a constrained penalized regression aggregation using multiple batches. Their approach assigns weights to the batches and constructs a combined prediction model based on the weights. Note that the concept of the batch in this literature is different from the one that we consider in deep neural network models.

For a sequence of data batches that consist of input and output pairs, $\{D_m\}_{m=1}^{M}$, suppose the observations in D_m are random samples from some unknown distribution P_m. Let $x \in \mathcal{X} \subset \mathbb{R}^d$ be a predictor vector and $y \in \mathcal{Y}$ be a response, and we assume that

$$y = f_m(x) + \epsilon,$$

where random error ϵ satisfies $\mathbb{E}(\epsilon) = 0$ and independent of x. Let $\hat{f}_m$ be a fitted model for $m \in [M]$. Using the inputs from the most recent data batch $D_M = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, the predicted values are:

$$\hat{f}(x_i) = (\hat{f}_1(x_i), \cdots, \hat{f}_{M-1}(x_i), \hat{f}_M^{(-i)}(x_i)),$$

where $\hat{f}_M^{(-i)}(x_i)$ is a leave-one-out estimate of $\hat{f}_M(x_i)$. Then, the design matrix and the response for the model aggregation are:

$$X = \begin{pmatrix} \hat{f}_1(x_1) & \cdots & \hat{f}_{M-1}(x_1) & \hat{f}_M^{(-1)}(x_1) \\ \hat{f}_1(x_2) & \cdots & \hat{f}_{M-1}(x_2) & \hat{f}_M^{(-2)}(x_2) \\ \vdots & \ddots & \vdots & \vdots \\ \hat{f}_1(x_n) & \cdots & \hat{f}_{M-1}(x_n) & \hat{f}_M^{(-n)}(x_n) \end{pmatrix}. \tag{3.7}$$

The final predictive model has the form of $\hat{f}_{\text{fin}}(x) = \sum_{m=1}^{M} w_m \hat{f}_m(x)$ where $w = (w_1, \cdots, w_M)^T$. The key idea of the proposed method is to combine information from multiple batches for the final prediction. We apply this idea to our multimodal data setting by regarding modalities as a unit of models instead of batches.

To apply the idea of Wang et al. (2017) to our multimodal data learning setting, we consider the design matrix in Equation (3.7) with score vectors from the individual neural networks. In that way, each column indicates a modality, and each row is a probability estimate for a class, *i.e.*,

$$X = \begin{pmatrix} p_1^{(1)} & \cdots & p_1^{(M-1)} & p_1^{(M)} \\ p_2^{(1)} & \cdots & p_2^{(M-1)} & p_2^{(M)} \\ \vdots & \ddots & \vdots & \vdots \\ p_K^{(1)} & \cdots & p_K^{(M-1)} & p_K^{(M)} \end{pmatrix} = (p^{(1)}, \cdots, p^{(M)}).$$

This gives the final predictive model $\hat{f}_{\text{fin}}(x)$ equivalent to z_{comb} as in (3.6). Hence, we minimize the empirical risk in Equation (2.2) where $f(x) = z_{\text{comb}}$ during training.

During the training phase, weights are calculated based on obtained score vectors from individual neural networks, $p^{(1)}, \cdots, p^{(M)}$. Each score vector passes through a dense layer with output size 1, and we obtain M scalar values, denoted by $u_1, \cdots, u_M$. Then, using the softmax function, we calculate weights w_m for each modality as follows:

$$w_m = \frac{\exp(u_m)}{\sum_{j=1}^{M} \exp(u_j)}$$

for $m \in [M]$. We call this method Concept Drift (CD) and unlike other methods, this method learns weights during model training. Note that CD is equivalent to Attention (Luong et al., 2015).

Now we propose two methods that use deterministic functions to calculate weights based on score vectors at each step of the training phase, and we call these methods WACDp and WACDe, respectively. The key idea of both approaches is to assign a weight to each modality based on its importance. The proposed methods are easily interpretable in that weights directly answer which modalities are important and how much so.

Weighted Average Concept Drift with probability (WACDp) Weighted Average Concept Drift with probability (WACDp) calculates weights based on probability estimates. We consider different approaches for multiclass classification and binary classification, but the main idea of the method is the same: focus more on a modality with smaller uncertainty and less on a modality with larger uncertainty.

For the multiclass classification case with K classes, we use the maximum value among probability estimates for classes. That is, we take the maximum probability estimates from

the score vectors and use the softmax function to calculate a weight for each modality, *i.e.*,

$$u_m = \max(p_1^{(m)}, \cdots, p_K^{(m)}), \quad w_m = \frac{\exp(u_m)}{\sum_{j=1}^{M} \exp(u_j)},$$

for $m \in [M]$. If the maximum probability estimate is large, that means the model is more certain about the prediction, so we give larger weights, and vice versa.

For the binary classification case, we consider the predicted probability for the class 1, say p^m. We consider the performance of the model is better when p^m is close to 0 or 1 rather than near 0.5. Hence, we use the absolute distance between the predicted probability p^m and 0.5 to calculate weights, *i.e.*,

$$u_m = |p^m - 0.5|, \quad w_m = \frac{\exp(u_m)}{\sum_{j=1}^{M} \exp(u_j)},$$

for $m \in [M]$. Note that the binary classification case is equivalent to shifting the multiclass classification case by 0.5.

Weighted Average Concept Drift with entropy (WACDe) The last method, Weighted Average Concept Drift with entropy (WACDe), is to use the entropy of the probability estimates. Weight for each modality is obtained as:

$$u_m = -\sum_{k=1}^{K} p_k^{(m)} \log p_k^{(m)}, \quad w_m = \frac{\exp(1/u_m)}{\sum_{j=1}^{M} \exp(1/u_j)},$$

for $m \in [M]$. We obtain entropy of the score vector for the m-th modality, u_m, then use the softmax function with reciprocal of u_m to calculate the weights. In this way, the model focuses more on the important modalities and less on minor modalities because it assigns larger weight when a predicted probability shows more certainty. Hence, the weight reflects the importance of the modality and thus provides an intuitive interpretation.

Even though the key idea WACDp and WACDe of considering the importance of modalities is the same so that both methods assign larger weights to the modalities with less uncertainty, WACDe has larger differences in weight values. By considering entropy instead of probability value itself, the weight difference is amplified so that WACDe assigns a larger weight to the important modality than WACDp does. This leads major modality to have more contribution to the combined score vector when using WACDe than WACEp.

3.3 Prediction Interval

In this section, we describe how to obtain a prediction interval for a new observation from a trained model. In Section 2.2.3, we show that the optimization of neural networks with dropout layers is equivalent to Bayesian variational inference. Using this characteristic, we can obtain Bayesian neural network models with dropout layers, and thus obtain model uncertainty and prediction intervals.

After model training, we consider the variational predictive distribution that approximates

$$q(y^*|x^*) = \int_{\Theta} p(y^*|x^*, \theta) q_\phi(\theta) d\theta$$

for $\theta \in \Theta$. Then, we estimate the first moment of the predictive distribution using the variational predictive distribution. We first sample T sets of Bernoulli realizations[2] $\{z_{t,1}, \cdots, z_{t,L}\}_{t=1}^T$ with $z_{t,l} = [z_{t,l,j}]_{j=1}^{d_l}$ for $l \in [L]$. This gives random sets $\{W_{t,1}, \ldots, W_{t,L}\}_{t=1}^T$ which generate a set of random outputs $\{\hat{y}_t^*(x^*)\}_{t=1}^T$ for a given test data point x^*. Then, for $k \in [K]$, a posterior mean is estimated as follows,

$$\mathbb{E}_{q(y^*|x^*)}(I(y^* = k)) \approx \frac{1}{T} \sum_{t=1}^{T} (\hat{y}_t^*(x^*))_k, \tag{3.8}$$

[2]Here, we slightly abuse the subscript notation to denote t-th set of Bernoulli realizations.

where $\mathbb{E}_{q(y^*|x^*)}(\cdot)$ denotes the expectation taken over the variational predictive distribution. In practice, this is equivalent to performing T stochastic forward passes through the network and taking the average of the results.

For a given significance level $\alpha \in (0, 1)$, a $100(1-\alpha)\%$ prediction interval for a given class $k \in [K]$ is created by calculating the $(\alpha/2)$-th and $(1-\alpha/2)$-th quantiles from the empirical predictive posterior distribution. To be specific, the α-th percentile value of an empirical predictive posterior distribution for a given class k is defined as

$$q^*_{\alpha,k} := \inf \left\{ \tilde{y}_k \in \{(\hat{y}^*_t(x^*))_k\}_{t=1}^T \mid \frac{1}{T} \sum_{t=1}^{T} I\Big((\hat{y}^*_t(x^*))_k \leq \tilde{y}_k \Big) \geq \alpha \right\}.$$

Then, the $100(1-\alpha)\%$ prediction interval is given as

$$(q^*_{\frac{\alpha}{2},k}, q^*_{1-\frac{\alpha}{2},k}).$$

Gal and Ghahramani (2016) assess model classification confidence and predictive performance of dropout on ten different datasets. In particular, they compare uncertainty quality obtained from dropout in neural networks to that of Probabilistic Backpropagation (Hernández-Lobato and Adams, 2015) and a variational inference technique in Bayesian neural networks (Graves, 2011) which are state-of-the-art methods developed to capture uncertainty. As a result, uncertainty from dropout has significant improvement in root mean square error as well as predictive log-likelihood, meaning dropout gives better uncertainty estimates compared to other methods and this will lead to better prediction intervals.

Chapter 4

Numerical Experiments

In this chapter, we perform several numerical experiments for two problems: digit classification and Alzheimer's disease (AD) classification. We compare six different combining methods and inspect which method is the most effective and provides a reliable solution based on classification accuracy and prediction interval, in Section 4.2.1 and 4.2.3, respectively. Also, we examine how feature selection can affect classification performance in Section 4.2.2. We empirically show that selecting important features can significantly improve the model performance while reducing high-dimensionality.

4.1 Experiment Settings

In this section, we first elaborate on datasets, network architectures, and implementation settings for the two classification problems.

4.1.1 Digit Classification

Dataset For the first numerical experiment, we use the MNIST dataset of handwritten digits (LeCun et al., 2010). The MNIST dataset consists of a training set of 60000 images and a test set of 10000 images of 28×28 pixels. We use the first 2000 images for training and

all 10000 images for testing in our experiment. With the sampled MNIST dataset, we create a corrupted dataset, called MNIST-C using corruption functions given in Mu and Gilmer (2019). We use seven different corruptions (defocus blur, glass blur, impulse noise, rotate, scale, shear, and zigzag) to create the MNIST-C dataset. Figure 4.1 shows five examples of the MNIST-C data for different corruptions. The first row displays the images without corruption and from the second to the eighth rows show examples of corrupted images using defocus blur, glass blur, impulse noise, rotate, scale, shear, and zigzag corruption, respectively. Some corruptions, for example, defocus blur, glass blur, and impulse noise make numbers not quite recognizable, while others such as rotate, scale, and zigzag corruptions retain clear images of the digits.

Each dataset of corrupted digits is considered as one modality. In addition to the corrupted modalities, we also use zero images as one modality. That means, we use one modality that is empty and has no information. We use this modality to examine the effect of a modality that has no information on the event.

We consider eight combinations of four modalities and denote them C1 to C8. Table 4.1 gives the eight combinations of four modalities. For the first four combinations, C1 to C4, we fix the three modalities, namely defocus blur, glass blur, and impulse noise, and vary the fourth modality as shear, rotate, scale and zigzag. The reason we fix the first three modalities and use different fourth modalities is to consider different combinations of accuracy. For the next four combinations, C5 to C8, we use zero modality instead of defocus blur modality in the first four combinations, C1 to C4. In other words, we fix three modalities, glass blur, impulse noise, and zero, and use four different fourth modalities: shear, rotate, scale and zigzag. By using the zero modality that does not have any information on the digits, we can observe the effect of a modality that is not useful at all.

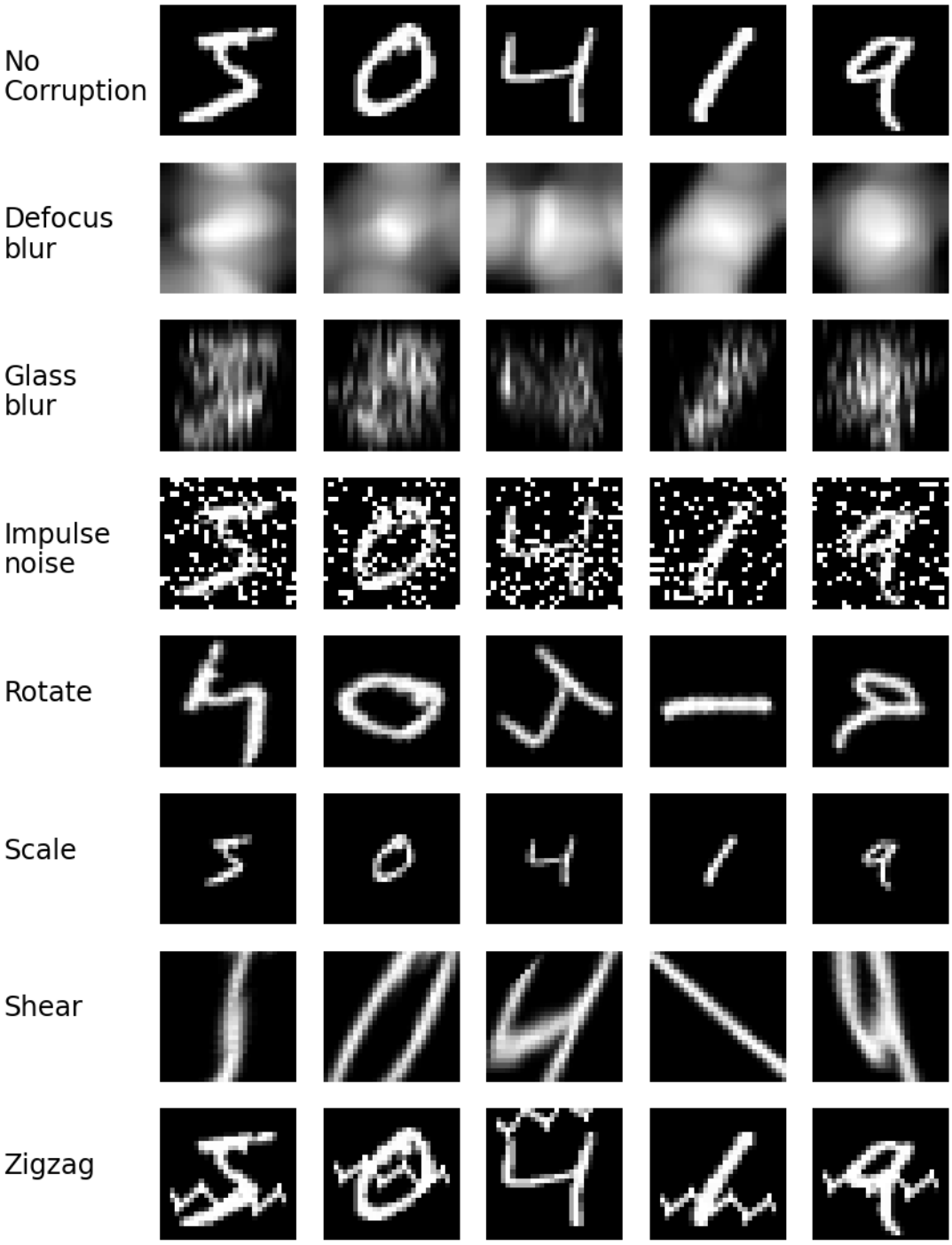

Figure 4.1: Five examples of MNIST and MNIST-C dataset. The first row shows the images with no corruption. The images from the second to the eighth row are examples of defocus blur, glass blur, impulse noise, rotate, scale, shear, and zigzag corrupted images, respectively.

Table 4.1: Eight combinations of four modalities are listed: C1 to C8. The first four combinations, C1 to C4, include three fixed modalities, glass blur, impulse noise and defocus blur and use four different modalities, shear, rotate, scale and zigzag, as the fourth modality. Similarly, combinations C5 to C8 have glass blur, impulse noise and zero modality as fixed and vary the fourth modality.

Combination	Modalities
C1	Glass blur, Impulse noise, Defocus blur, Shear
C2	Glass blur, Impulse noise, Defocus blur, Rotate
C3	Glass blur, Impulse noise, Defocus blur, Scale
C4	Glass blur, Impulse noise, Defocus blur, Zigzag
C5	Glass blur, Impulse noise, Zero, Shear
C6	Glass blur, Impulse noise, Zero, Rotate
C7	Glass blur, Impulse noise, Zero, Scale
C8	Glass blur, Impulse noise, Zero, Zigzag

Network architecture Figure 4.2 illustrates the architecture of the model. In this experiment, we do not conduct feature selection. Each input modality is fed into the individual neural network, and we obtain a score vector for each modality. Each individual neural network includes a convolutional layer, max pooling layer, another convolutional layer with a dropout layer, and a dense layer as the output layer. We use the output layer with the softmax activation function so that we obtain a score vector from each modality. That means, for each modality, features are summarized into vectors in the same dimension, the number of classes, 10, where each element is a probability estimate for each class. Performance of single modality models is obtained at this step with score vectors. After combining information from four modalities using different combining methods, we use the combined score vector to make the final classification decision for multimodal models.

Implementation details Among the 2000 training images, 1800 images are used for training the model, and the remaining 200 images are used for model validation.

For training the model, we use batch size 64, 100 epochs, and early stopping with patience 10. This means that if the model reaches a minimum validation loss during training and a

better model is not achieved for 10 more epochs, the training process stops and the minimum loss is taken. The number of repetitions is 100, and the performance of models is compared with the average accuracy.

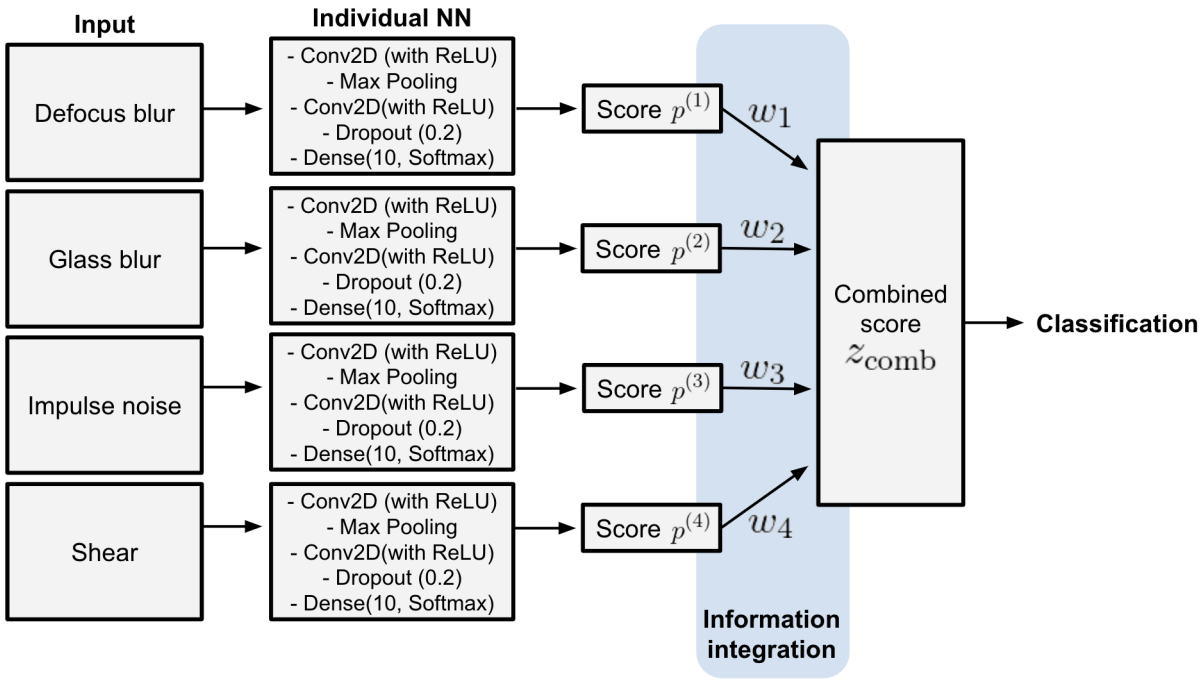

Figure 4.2: An illustration of neural network architecture when the input combination is C1 (defocus blur, glass blur, impulse noise, and shear modality). Each individual neural network includes a dropout layer and outputs a score vector. A combined score obtained is used for the final classification.

4.1.2 Alzheimer's Disease (AD) Classification

Datasets For experiments regarding AD classification, we use datasets from ADNI. We use three modalities: MRI, demographic and genetic modality.

MRI modality is not raw image data but values calculated from image data, such as volume, thickness, depth, convexity, to name a few. Among 2150 variables in the MRI dataset, 1075 are measurements from the left side of the brain and the other 1075 are measurements from the corresponding right side of the brain. We have 628 subjects in total including 133 AD,

305 MCI and 190 CN. For the demographic modality, we have 19 variables that includes information about subjects such as diagnosis, age, gender, years of education, ethnicity, race, ApoE4 genotype, mini-mental state examination (MMSE) score, and more. Demographic modality also includes 628 subjects in total who are the same ones in the MRI modality: 133 AD, 305 MCI and 190 CN subjects. Lastly, genetic modality is the gene expression data taken from blood samples of the patients and consists of 49395 variables and 745 subjects. Since diagnosis of 745 subjects in the genetic modality is not available, we only can use the subjects who have information in the other two modalities. This gives only 220 subjects including 102 CN and 118 MCI subjects who have information in all three modalities. Thus, we use just those 220 subjects in our analysis. During preprocessing the modalities, we remove 9 variables from the genetic modality that are demographic information, and 12 variables from the demographic modality that are duplicated information including subject ID, date, and diagnosis. As a result, we use MRI modality with 2150 variables, demographic modality with 7 variables, and genetic modality with 49386 variables for the analysis. Table 4.2 summarizes the three modalities.

Table 4.2: Summary of three modalities in AD classification including description, dimension and sample size.

Modality	Description	Dimension
MRI	Values calculated from image data, such as volume, thickness, depth, convexity, and more.	2150
Demographic data	Age, gender, ethnicity, race, years of education, ApoE4 genotype, and MMSE score.	7
Genetic data	Gene expression data taken from blood samples of the patients.	49386

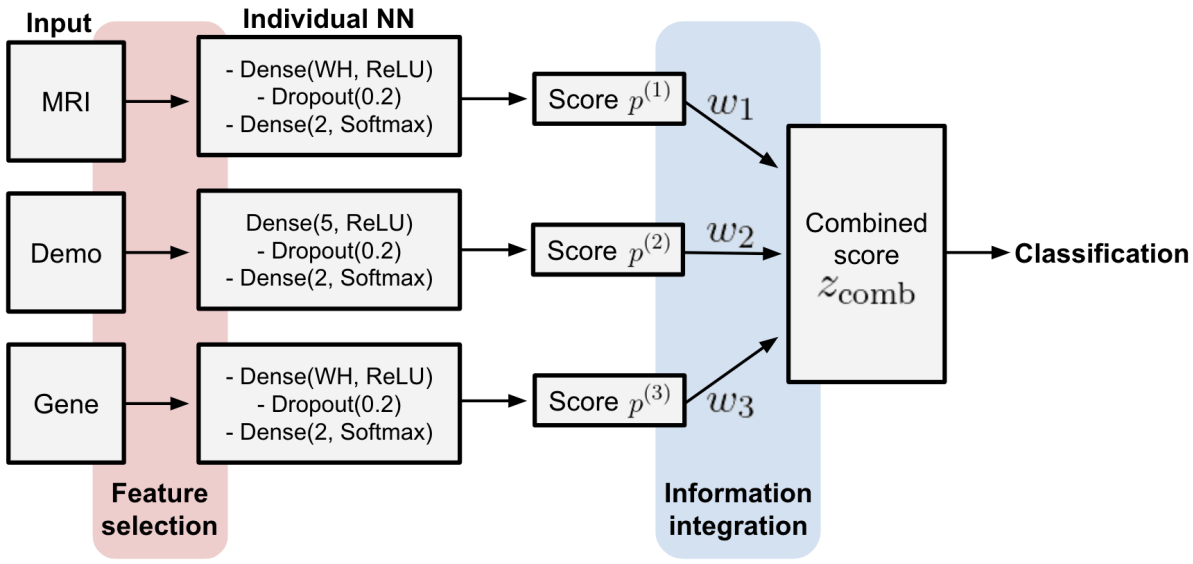

Figure 4.3: An illustration of neural network architecture for AD classification. WH indicates the width of the hidden layer and they vary as the input size varies. 5, 10 and 50 are used for the input size 5, 20 and 100, respectively.

Network architecture Figure 4.3 gives the architecture of the model including feature selection and information combining steps.

High-dimensional modalities, MRI and genetic modalities, have highly correlated variables so feature selection is necessary to remove multicollinearity and errors. We use the three feature selection methods: PCA, CA and SIS. All three methods are described in Section 3.1. We use different sizes of sets for feature selection, and we call these sets `Small`, `Medium`, `Large` and `Very large`. We use the `Very large` set only for genetic modality since `Medium` set is large enough to contain enough information of MRI modality and `Large` set shows no improvement in model performance, which will be further discussed in Section 4.2.2. The sizes of four different sets are around $5, 20, 100$, and 200. The exact number of features selected from different methods are in Table 4.3. For PCA, since PCs are independent of each other, the sizes are exactly $5, 20, 100$, and 200. In contrast, for CA and SIS, the number of selected features varies due to the correlation between variables. Once we select sets of variables for each size using CA and SIS, we remove variables that have correlations higher

than 0.7 with other variables. By selecting different sizes of sets of important variables, we find what size is optimal and contains the most information by comparing the accuracy. Note for the demographic modality, we use all variables since it is not high-dimensional data.

Table 4.3: The number of selected variables for different sizes of sets using three methods: PCA, CA, and SIS.

Modality	Method	`Small`	`Medium`	`Large`	`Very large`
MRI	PCA	5	20	100	-
	CA	5	17	93	-
	SIS	5	14	72	-
Genetic	PCA	5	20	100	200
	CA	5	30	93	230
	SIS	6	18	86	234

For individual neural networks, we use a dense layer with dropout layer with dropout rate 0.2, and a dense layer as the output layer. Input size varies from 5 to the number of variables in the original data, 2150 and 49386 for MRI and genetic data, respectively, the first dense layer has a different hidden layer width depending on the input size. The output dense layer has output size of 2, the number of classes. We use a softmax activation function for the last layer to obtain probability estimates. With each score vector from each modality, the accuracy of single modality models is obtained. Then, we combine score vectors with different weights from the six different combining methods for final classification and obtain the accuracy of the multimodal models.

Implementation details Sample size 220 is divided into 200 for training and 20 for testing. Within the training set, randomly chosen 160 subjects were used for training the model, and the other 40 subjects were used for validation.

For training, the number of the epoch is 100 and early stopping with patience 10 is used. This means, while training when the model reaches minimum validation loss and no better model is achieved for 10 epochs after that, the training process stops. Both train and test

batch size is 20 and the learning rate is 0.001. The entire process of training the model is repeated 100 times, and we calculate the average accuracy and its standard error to compare the performance of models.

Methods Throughout the experiments, we compare the six different combining methods: (i) SA, (ii) WA, (iii) EmbNet, (iv) CD, (v) WACDp, and (vi) WACDe. All methods are described in Section 3.2 except for EmbNet, which is described in Section 2.3.2.

4.2 Experiment Results

4.2.1 Model Performance Comparison

In this section, we compare the performance of multimodal models and the best single modality models, and show the proposed method, WACDe performs better than other multimodal models.

Digit classification We compare the performance of multimodal models and the best single modality models in Table 4.4. We compute the average accuracy and its standard error based on 100 repetitions. Each column indicates different combinations of modalities, C1 to C4 in Table 4.1. The first row gives the highest accuracy among the four single modality models. The second to the seventh rows give the six different combining methods. Further details on the performance of single modality models are in Appendix A.1.

We anticipate the accuracy increases since we use more information by combining multimodal data. However, surprisingly, not all the multimodal models have better performance than single modality models. In the case of C3, the single modality model is actually better than any multimodal model. We further investigate this finding with similarity between images in Section 4.2.4. In case of C1, C2, and C4, EmbNet and CD methods have lower accuracy than the best single modality. For C1 and C2, SA, WA, WACDp and WACDe

achieve better performance than the best single modality, but in case of C4 the single modality is better than the multimodal methods except for WACDe. Both WACDp and WACDe use weights that reflect the importance of modalities while WACDe amplifies the difference between the weights by considering entropy instead of the probability values itself. It is noteworthy that WACDe, based on the optimal weights, achieves higher accuracy than other methods in many cases. That is, using optimal weights that reflect the information quality of each modality improves the model performance. Also, WACDe gives intuitive interpretation in that weights directly answer which modalities are important and how much so.

Table 4.4: Average accuracy (and standard error) of 100 repetitions of four combinations of modalities, C1-C4. The first row is the highest average accuracy of a single modality for each combination. The second through seventh rows give the performance of the six different combining methods. Bold text indicates the highest performance.

	C1	C2	C3	C4
Best single modality	0.9201 (0.0005)	0.9303 (0.0005)	**0.9641 (0.0002)**	0.9486 (0.0004)
SA	0.9260 (0.0005)	0.9345 (0.0004)	0.9429 (0.0004)	0.9384 (0.0004)
WA	0.9275 (0.0004)	0.9353 (0.0004)	0.9460 (0.0003)	0.9406 (0.0003)
EmbNet	0.7155 (0.0182)	0.7843 (0.0143)	0.7380 (0.0215)	0.7447 (0.0199)
CD	0.9165 (0.0007)	0.9279 (0.0008)	0.9462 (0.0009)	0.9366 (0.0009)
WACDp	0.9320 (0.0004)	0.9391 (0.0003)	0.9486 (0.0003)	0.9428 (0.0003)
WACDe	**0.9348 (0.0003)**	**0.9428 (0.0003)**	0.9556 (0.0002)	**0.9488 (0.0002)**

Table 4.5 presents results when we use the zero modality as one of the modalities in the analysis. Similar to Table 4.4, each row gives results from six different combining methods, and the first row is from the best single modality model. Each column indicates different combinations of modalities, C5 to C8 in Table 4.1. Most results are similar even though we include a modality with no information. This is because the zero modality is pure noise to the other three modalities, but does not damage the integrated information. In Table 4.4, WACDe shows the highest accuracy among all multimodal models in C1 and C2, and this exceeds the best single modality models. Even when one of the modalities we use is noise, in C5 and C6, the promising performance of WACDe is maintained. Also, for C7 and C8, we observe that WACDe has the highest performance among the six combining methods. As a

result, the effective combining method is important because using optimal weights leads to a better performance of the model.

Table 4.5: Average accuracy (and standard error) of 100 repetitions of four combinations of modalities, C5-C8. The first row is the highest average accuracy of a single modality for each combination. The second through seventh rows give the performance of the six different combining methods. Bold text indicates the highest performance.

	C5	C6	C7	C8
Best single modality	0.9201 (0.0005)	0.9303 (0.0005)	**0.9641 (0.0002)**	**0.9486 (0.0004)**
SA	0.9269 (0.0004)	0.9355 (0.0003)	0.9474 (0.0003)	0.9416 (0.0003)
WA	0.9293 (0.0004)	0.9369 (0.0004)	0.9503 (0.0003)	0.9429 (0.0003)
EmbNet	0.6947 (0.0202)	0.6942 (0.0251)	0.6998 (0.0244)	0.7011 (0.0211)
CD	0.9109 (0.0009)	0.9215 (0.0011)	0.9413 (0.0009)	0.9303 (0.0011)
WACDp	0.9305 (0.0004)	0.9386 (0.0003)	0.9500 (0.0003)	0.9435 (0.0003)
WACDe	**0.9339 (0.0003)**	**0.9426 (0.0003)**	0.9551 (0.0002)	0.9475 (0.0002)

AD classification We first compare performance of the best single modality model and multimodal models. Table 4.6 gives the average accuracy and its standard error based on 100 repetitions of multimodal models as well as the best single modality model. In the table, we observe that the best single modality model has accuracy of 0.7075, and none of the multimodal models achieve this. Among the six combining methods, WACDe is the best with accuracy 0.6340, and this is much lower than the best single modality model. This might contradict intuition in that multimodal data contains more information. MRI and genetic modalities have two problems: high-dimension and high correlation. Both modalities contain a large number of variables, 2150 and 49386, respectively, while the sample size is 220. Also, the variables are highly correlated to each other in conveying information about AD. That could be why single modality model is better than the multimodal models. However, the two problems can be solved by feature selection, which we give further details in Section 4.2.2.

Table 4.6: Average accuracy (and standard error) of 100 repetitions for multimodal models. The first row is the accuracy of the best single modality model and from the second to the seventh row are those of multimodal models. Bold text indicates the highest performance.

	`All data`
Best single modality	**0.7075 (0.0058)**
SA	0.6185 (0.0082)
WA	0.6195 (0.0094)
EmbNet	0.5845 (0.0127)
CD	0.6180 (0.0099)
WACDp	0.6245 (0.0085)
WACDe	0.6340 (0.0090)

4.2.2 Is Feature Selection Helpful?

In this section, we see how feature selection affects the performance of a multimodal model. We show that feature selection is helpful for dimension reduction and high correlation removal, leading to performance improvements. Also, we present a type of synergy effect between feature selection and the multimodal model.

AD classification First, we show the difference between results of two feature selection methods, CA and SIS, via correlation maps. We do not create correlation maps for PCA since PCA finds linear combinations of the original variables, and they are independent to each other. Figure 4.4 illustrates correlation maps for `Medium` set of selected variables using CA and SIS in left and right panel, respectively. Each row of the figure indicates MRI, genetic modality and all modalities. Correlation maps of all modalities include selected variables in MRI modality, entire demographic modality, and selected variables in genetic modality. Hence, correlation maps of all three modalities include a correlation map of MRI modality in the left top corner and of genetic modality in the right bottom corner. For visualization, we omit the variable names, which can be found in Table A.4 and Table A.5 for MRI and genetic modality, respectively. Also, we do not apply any feature selection for the demographic modality since it is not high-dimensional data.

Surprisingly, there is no overlap between CA selected variables and SIS selected variables for `Medium` set. Also, correlation patterns from the two feature selection methods are very different. In general, for both MRI and genetic modality, SIS selected features show stronger correlations. SIS selected variables show moderate correlation to each other, while most of CA selected variables are not correlated to each other. This pattern also can be observed between modalities. The last row of Figure 4.4 illustrates correlations between three modalities, which are shown on the right top corner. For CA method, the right top corner is very light, meaning modalities are not much correlated to each other, while those are stronger for SIS method. These different patterns lead to different performance of the model, where SIS method has higher accuracy than CA method. Correlation maps of other sizes of selected variables are in Appendix A.2.

We show the effects of feature selection on single modality models. Table 4.7 presents the performance of single modality models with the three different feature selection methods: PCA, CA, and SIS. The first column, named `All data`, gives accuracy of single modality models without feature selection, and from the second to the fifth column give accuracy of different sizes of selected features. We use four different sizes of a set, `Small`, `Medium`, `Large`, and `Very large`, and the number of selected variables varies for different feature selection methods. The exact numbers of variables selected for each method are in Table 4.3.

For `All data`, both MRI and genetic data are high-dimensional and the variables are highly correlated as well, so the performance for both modalities is poor. However, by selecting a set of important features using SIS method, we observe significant increase in accuracy. In contrast, both PCA and CA methods are not effective in our experiments. They even show decreases in the performance of the models after selecting the variables.

Using SIS, `Medium` set of the MRI modality gives the highest accuracy of 0.7145, which is significantly higher than `All data`, 0.6010. With more selected variables, `Large` set of the MRI modality shows a decrease in accuracy, meaning that `Medium` set is large enough

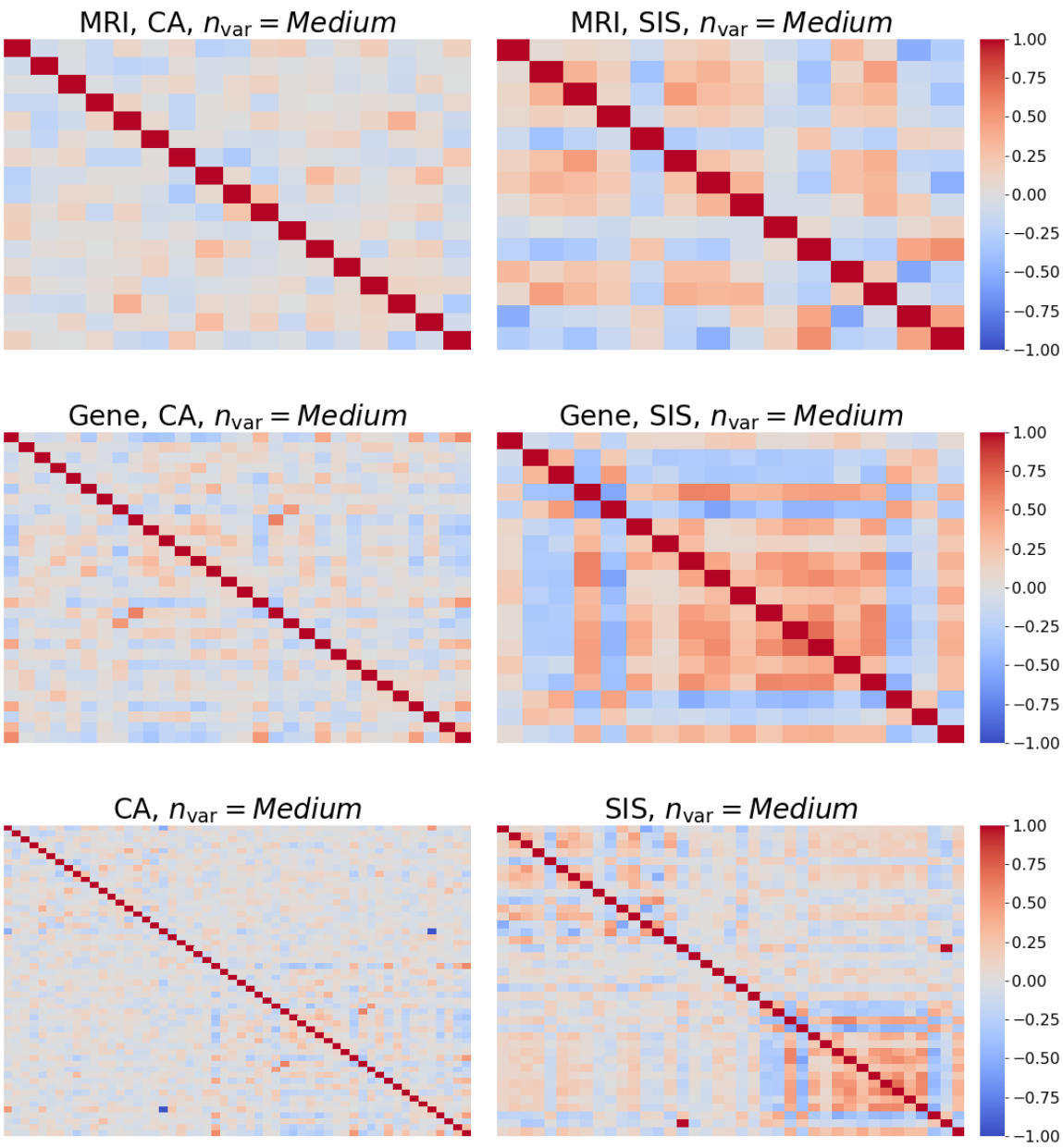

Figure 4.4: Correlation maps of `Medium` set of selected variables using (left) CA method and (right) SIS method. Each row indicates a different dataset where the first row is selected variables from MRI modality, the second row is that from genetic modality, and the last row is selected variables from MRI, genetic modalities and the entire demographic modality.

to contain useful information. Thus, we do not explore `Very large` set for MRI modality. For the genetic modality, the performance of the model increases as the size of the selected features set gets larger, and the model achieves the highest accuracy of 0.8960 with `Very large` set of selected variables. This is a significant increase compared to the accuracy without feature selection, 0.5740. Accordingly, for the multimodal models, we use variables selected from the SIS method. In this way, we consider the modalities with only useful information retained, which is efficient for information integration.

Table 4.7: Average accuracy (and its standard error) of 100 repetitions for single modality models with feature selection. Bold text indicates the highest performance for each modality.

		`All data`	`Small`	`Medium`	`Large`	`Very large`
Demo		0.7075 (0.0058)	-	-	-	-
MRI	PCA	0.6010 (0.0074)	0.4525 (0.0069)	0.5245 (0.0071)	0.5745 (0.0089)	-
	CA		0.4755 (0.0076)	0.5395 (0.0074)	0.4690 (0.0052)	-
	SIS		0.7070 (0.0051)	**0.7145 (0.0061)**	0.6850 (0.0075)	-
Gene	PCA	0.5740 (0.0104)	0.5495 (0.0093)	0.5805 (0.0095)	0.6395 (0.0084)	0.5020 (0.0080)
	CA		0.5095 (0.0115)	0.5885 (0.0075)	0.5500 (0.0092)	0.5250 (0.0081)
	SIS		0.6860 (0.0102)	0.7480 (0.0088)	0.8255 (0.0066)	**0.8960 (0.0045)**

We present results from the multimodal models in Table 4.8. Each row gives a different combining method as well as the best single modality model and each column gives different sets of selected variables. The first to the third columns give results of `Small`, `Medium`, and `Large` set of selected variables and the last column gives the result from `Best` combination of modalities. `Best` combination is composed of the sets that have the highest accuracy for each modality: `Medium` set for MRI modality, `All data` for demographic modality, and `Very large` set for genetic modality.

Feature selection solves the problem regarding high-dimensionality and high correlation in Section 4.2.1. Overall, using multimodal data achieves better performance than using a single modality for all sizes of selected variables. Also, WACDe has the best performance among all combining methods except for `Medium` set of variables. For `Medium` set of variables, WACDp

has the best performance, but it is similar to WACDe. Hence, by solving high-dimensional problem using an appropriate feature selection method, SIS, the multimodal models perform better than the single modality models, particularly using WACDe.

Table 4.8: Average accuracy (and its standard error) of 100 repetitions for multimodal models with feature selection. Bold text indicates the highest performance.

	`Small`	`Medium`	`Large`	`Best`
Best single modality	0.7075 (0.0058)	0.7480 (0.0088)	0.8255 (0.0066)	0.8960 (0.0045)
SA	0.7460 (0.0070)	0.7700 (0.0061)	0.8225 (0.0064)	0.8575 (0.0068)
WA	0.7495 (0.0072)	0.7775 (0.0073)	0.8310 (0.0070)	0.8765 (0.0057)
EmbNet	0.6795 (0.0098)	0.7065 (0.0088)	0.7750 (0.0100)	0.8200 (0.0125)
CD	0.7505 (0.0078)	0.7570 (0.0087)	0.8150 (0.0084)	0.8730 (0.0067)
WACDp	0.7515 (0.0077)	**0.7875 (0.0068)**	0.8260 (0.0060)	0.8860 (0.0054)
WACDe	**0.7560 (0.0073)**	0.7840 (0.0074)	**0.8395 (0.0053)**	**0.9035 (0.0050)**

We observe that not all the ensemble models achieve higher performance than single modality models. In other words, using more information does not necessarily increase accuracy. For example, with `Best` combination of modalities, the highest single modality model has accuracy of 0.8960. However, the four combining methods, SA, WA, EmbNet, and CD show markedly worse performance than the best single model. WACDe performs significantly better than any other combining models with accuracy of 0.9035, and this is the only method that exceeds the highest performance of the single modality model of genetic modality, 0.8960. WACDe uses weights that reflect the certainty of the different modalities, and achieves higher performance compared to the best single modality model. Therefore, we observe a type of synergy between selecting a set of important features and using optimal weights for the modalities. This leads to the best performance of the model.

4.2.3 How Does an Information Integration Method Affect Prediction Intervals?

In this section, we explore which information integration method creates a reliable prediction interval. To construct a prediction interval, we build Bayesian neural network models by using dropout layers. We focus on the AD classification problem.

AD classification With a trained model, we can obtain prediction intervals for new observations since we have dropout layers in our neural network models (Gal and Ghahramani, 2016). In this experiment, we use models trained with `All data` and `Best` set of modalities, `Medium` for MRI modality, all demographic modality, and `Very large` for genetic modality. We sample 30 sets of realized vectors drawn from Bernoulli to obtain 95% prediction intervals. Figure 4.5 and Figure 4.6 give prediction intervals of samples in the test dataset with `Best` set of modalities. In each figure, dots indicate the average predicted probabilities, and vertical dotted lines indicate 95% prediction intervals. Furthermore, blue (*resp.* red) intervals are the samples with correct (*resp.* incorrect) predictions.

The prediction intervals using SA and WA have consistent width as the predicted probability changes. On the other hand, prediction intervals using EmbNet and CD have very different width across test samples. Especially, EmbNet has wide intervals when the predicted probability is smaller than 0.5. As for CD, when predicted probabilities are close to either 0 or 1, intervals are very narrow. However, when predicted probabilities are close to 0.5, the intervals are very wide, close to covering the entire range of possible values. For WACDp, predictions can be separated into three groups: ones that are close to 0, close to 0.5 or close to 1. Also, incorrect predictions are made when probability estimates are very close to 0.5 and all those incorrectly predicted intervals contain 0.5. On the other hand, WACDe has narrower intervals compared to other methods. Also, width of intervals with predicted

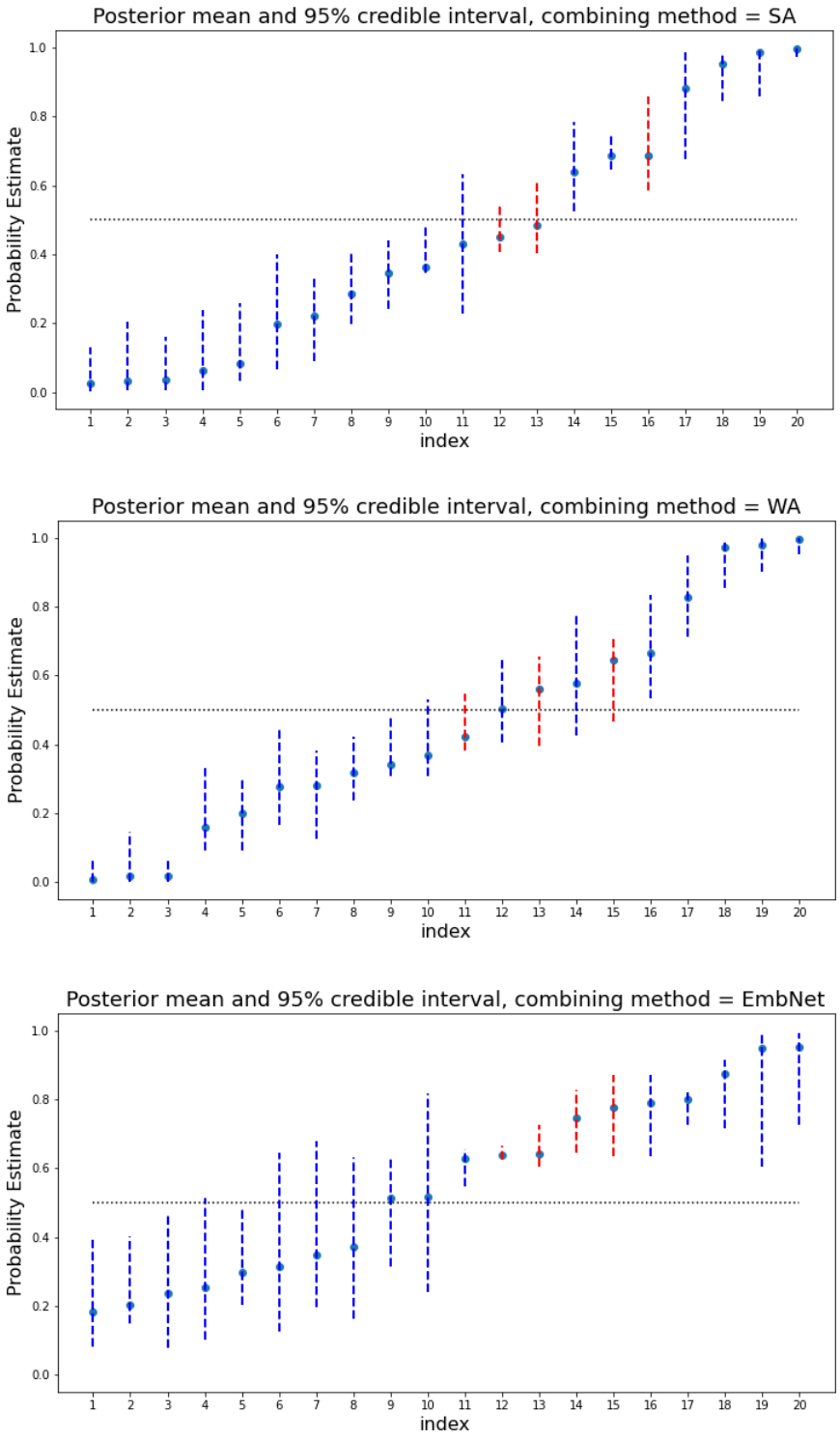

Figure 4.5: 95% prediction interval for `Best` set using combining methods SA (top), WA (middle), and EmbNet (bottom), respectively. Dots indicate the average probability estimates, and vertical dotted lines indicate 95% prediction intervals with blue (*resp.* red) representing samples with correct (*resp.* incorrect) predictions.

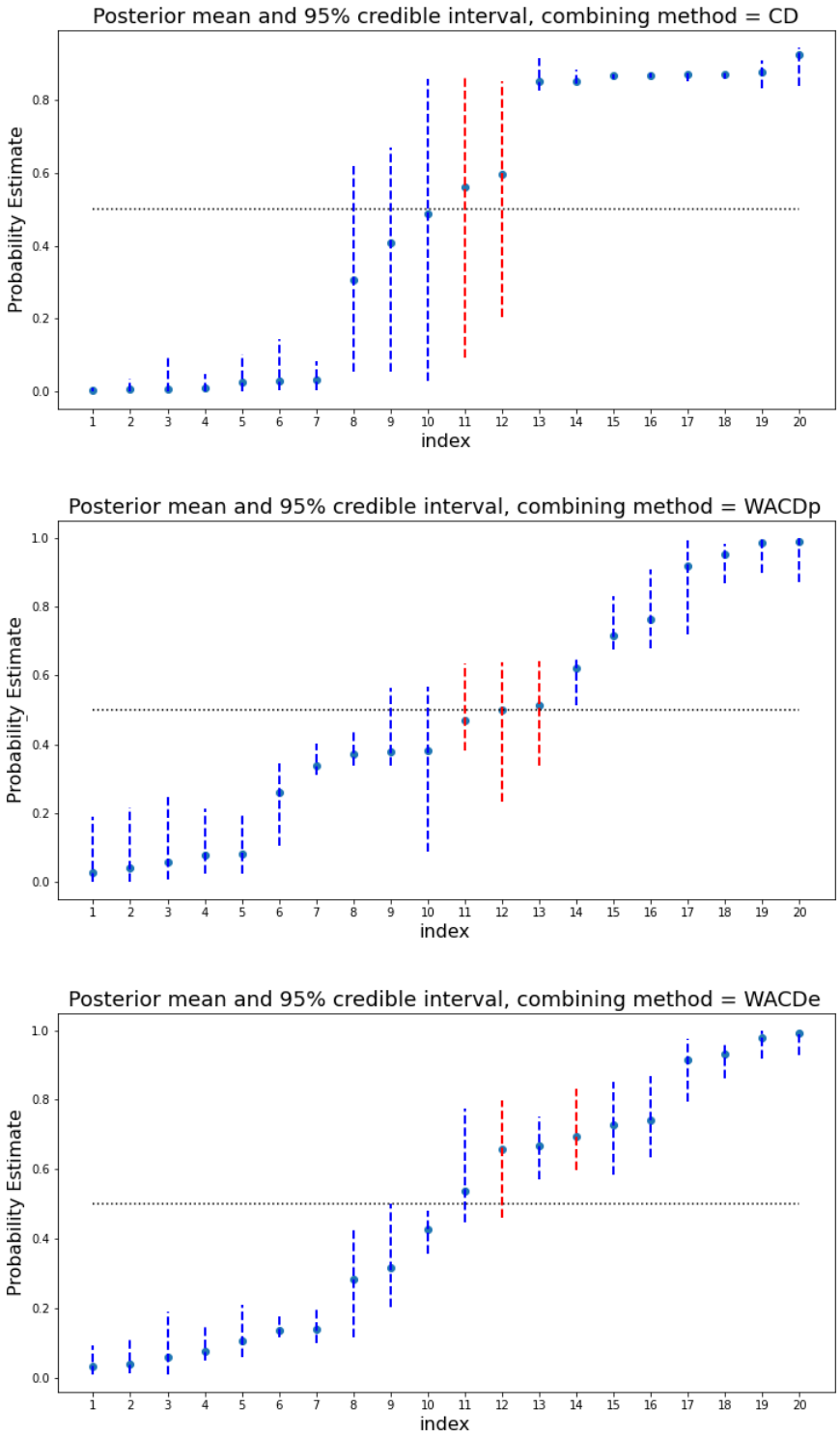

Figure 4.6: 95% prediction interval for `Best` set using combining methods CD (top), WACDp (middle), and WACDe (bottom), respectively. Dots indicate the average probability estimates, and vertical dotted lines indicate 95% prediction intervals with blue (*resp.* red) representing samples with correct (*resp.* incorrect) predictions.

probabilities close to either 0 or 1 are very small, hence, this method provides the most reliable predictions.

In summary, CD has very large variability when the model is unsure about its prediction, while SA and WA are quite consistent regardless of the predicted probabilities. In contrast, WACDe has narrower intervals compared to other methods, which means the predictions are the most reliable.

Similarly, Figure 4.7 and Figure 4.8 give the prediction intervals from different methods when we do not select features. We find the general trends are similar, but with more red intervals since `All data` have poor model performances. However, we still observe that WACDp and WACDe methods have narrower intervals than other methods. Hence, we conclude combining methods affect the certainty of the prediction. WACDp and WACDe give better predictions than other combining methods.

4.2.4 Similarity Between Images

In this section, we investigate the similarity between images to give further insight on the results from the digit classification experiment in Section 4.2.1. From the result of C3 in Table 4.4, we observe that none of the multimodal models achieve better performance than the single modality models. Scale corruption is just changing the size of the written digit and nothing else, so the scale modality has very clear digits as shown in Figure 4.1 and the accuracy is 0.9641. Accordingly, using three more modalities, defocus blur, glass blur, and impulse noise, to combine information actually results in adding noise and losing signals in the scale modality.

To explain this phenomenon, we use the Scale-Invariant Feature Transformation (SIFT, Lowe (2004)) to measure similarity between a corrupted image and an uncorrupted image. To calculate similarity, we detect distinctive features, called keypoints, in images and calculate

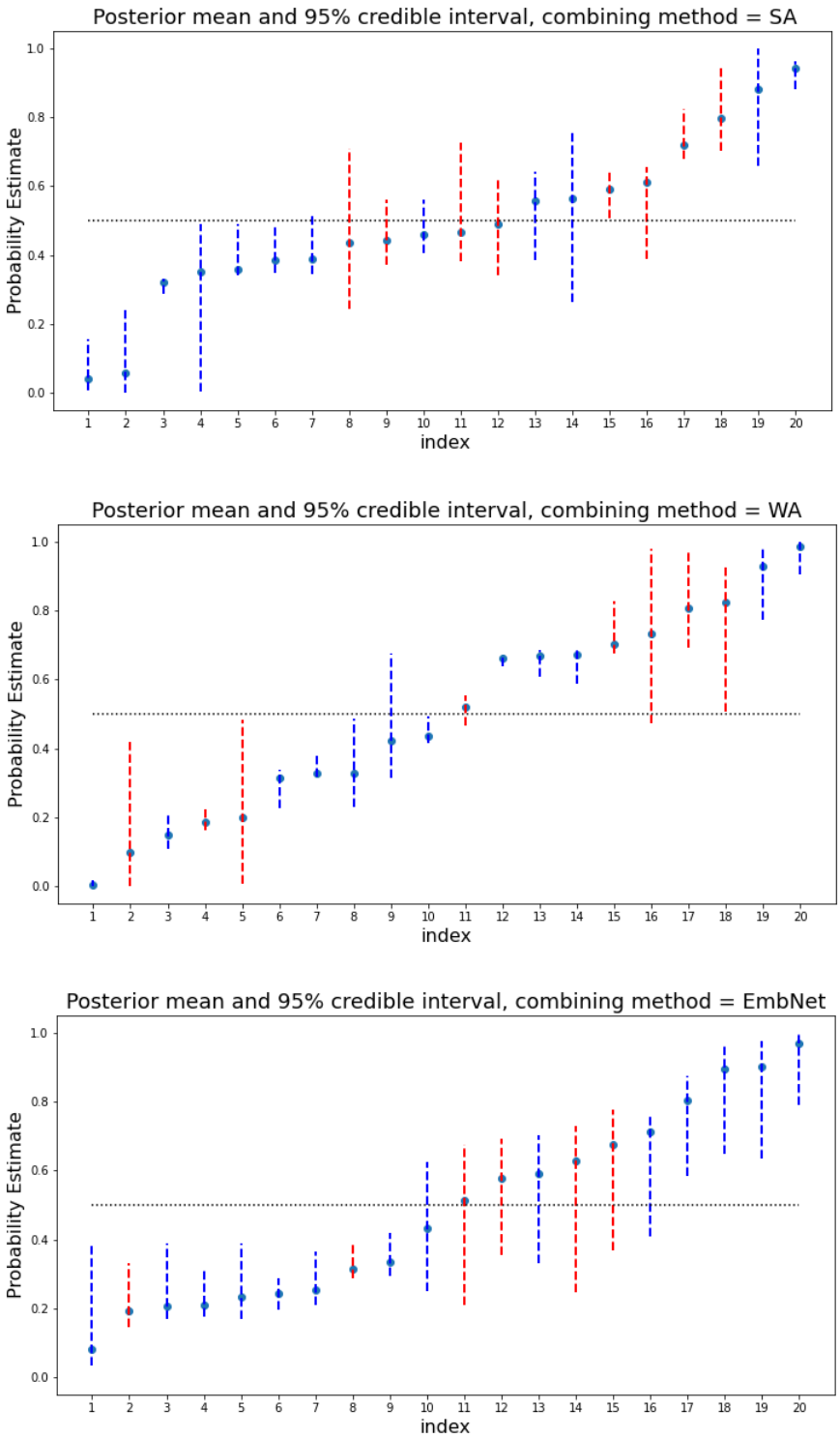

Figure 4.7: 95% prediction interval for `All data` using combining methods SA (top), WA (middle), and EmbNet (bottom), respectively. Dots indicate the average probability estimates, and vertical dotted lines indicate 95% prediction intervals with blue (*resp.* red) representing samples with correct (*resp.* incorrect) predictions.

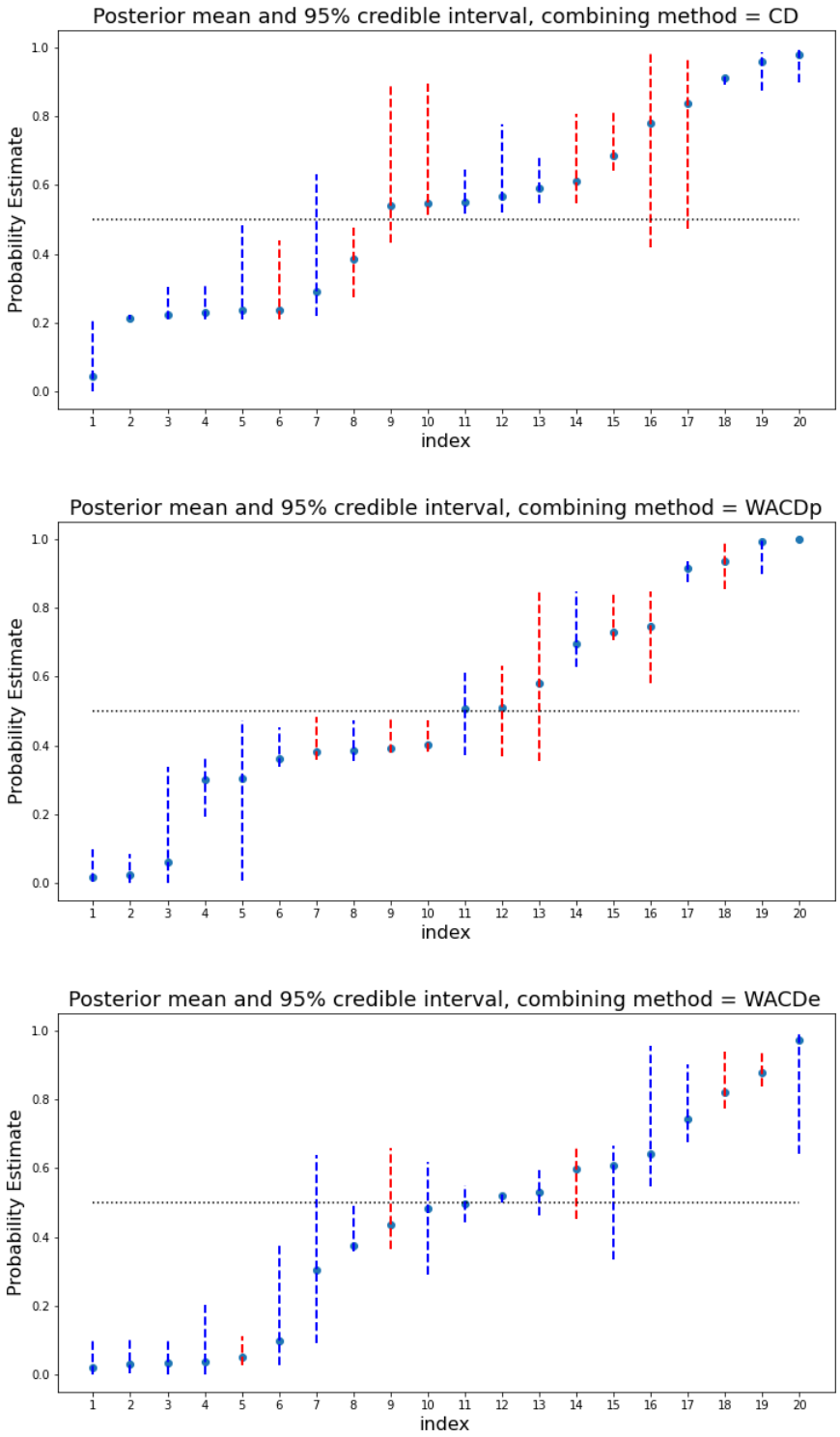

Figure 4.8: 95% prediction interval for `All data` using combining methods CD (top), WACDp (middle), and WACDe (bottom), respectively. Dots indicate the average probability estimates, and vertical dotted lines indicate 95% prediction intervals with blue (*resp.* red) representing samples with correct (*resp.* incorrect) predictions.

the average proportion of matched keypoints. By doing so, we compare clarity of digits in different corrupted images.

We first present seven examples of matched keypoints between corrupted images and non-corrupted image in Figure 4.9. In each Figure, the left panel is the corrupted image, and the right panel is the original image, and the white lines indicate matched keypoints. As for the defocus blur and glass blur corrupted images, we find no and only two matched keypoints, respectively, while zigzag corrupted images have a lot of lines matched between two images.

The similarity between images is the average proportion of matched keypoints calculated with training data, 2000 images. Table 4.9 gives the similarity between seven corrupted images and the original image, respectively. Zigzag corruption has the highest similarity, 0.5354, which is very reasonable since we observe that images are identical except the zigzag line in the last image in Figure 4.9. The second highest one is scale corruption, 0.2759, where corruption only scales the digit, but nothing else. On the other hand, defocus blur and glass blur have very low similarity, 0.0288 and 0.0586. It is sensible as we observe in Figure 4.9. The first row of the figure illustrates defocus blur and glass blur corrupted images, and digits are not even recognizable. Compared to each other, we find zigzag and scale corruption has higher similarity than other ones, especially defocus blur and glass blur. Therefore, it is reasonable to say combining information does not increase the quality of information but adds noise to the scale modality in C3, and this results in a decrease in the model performance even though we use multiple modalities.

4.3 Summary

Through the numerical experiments, we verify that using multimodal data generally increases the performance of the model, but not always. When we have high-dimensional modalities

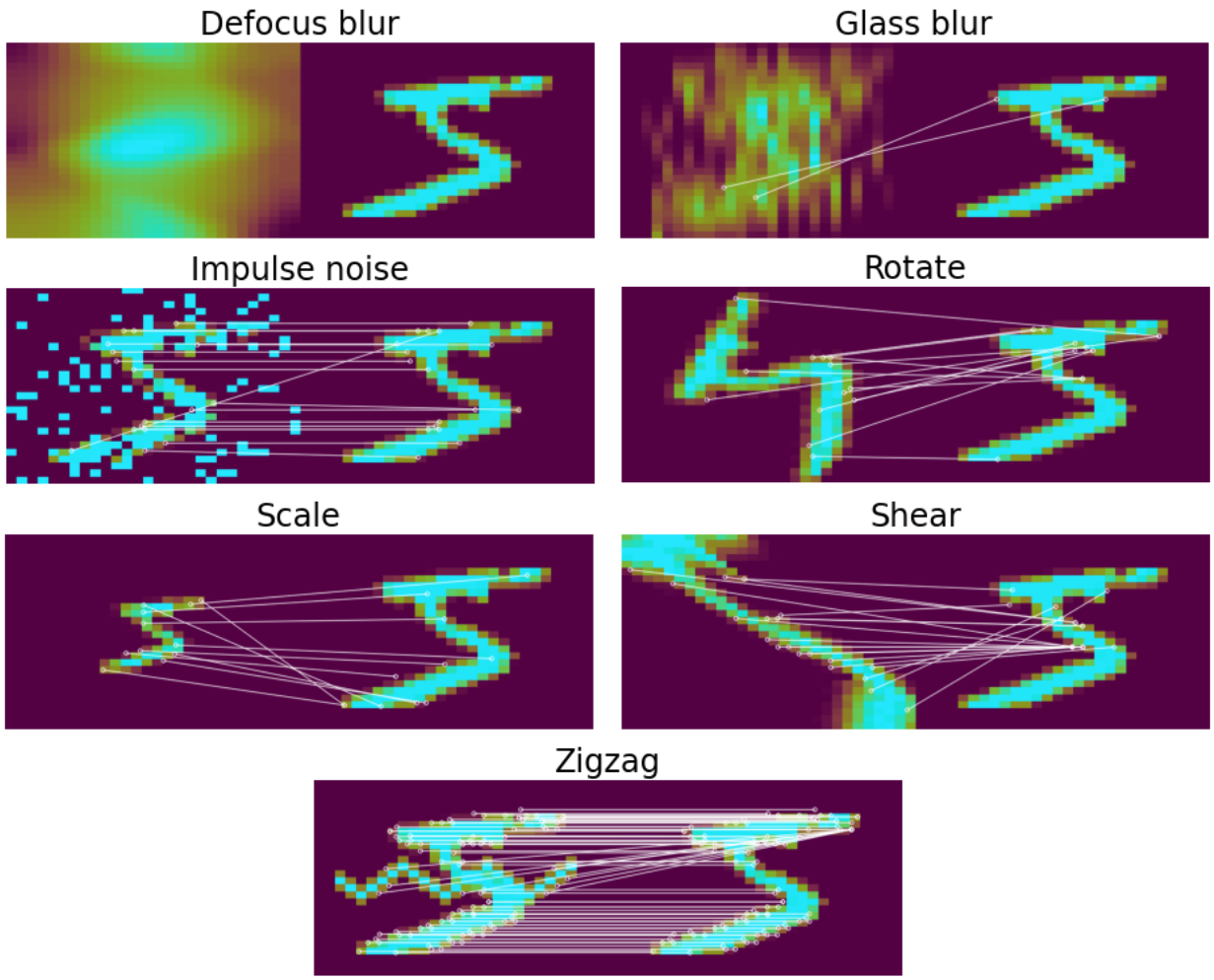

Figure 4.9: Seven figures of matched keypoints detected by SIFT. In each figure, left panel is corrupted image and right panel is the original image. White lines connected between images indicate matched keypoints.

Table 4.9: Similarity measurement based on SIFT between original image and corrupted images.

Corruption	Similarity
Defocus blur	0.0288
Glass blur	0.0586
Impulse noise	0.0954
Rotate	0.1747
Scale	0.2759
Shear	0.1699
Zigzag	0.5354

with highly correlated variables, a single modality model can perform better than the multimodal models. This problem can be solved by using an appropriate feature selection method. Particularly, we demonstrated that SIS improves the accuracy of both single modality models and multimodal models by selecting a set of important features.

In general, the proposed method, WACDe combining method shows promising results even when a modality does not contain any information about the event that can be considered as pure noise. In addition, WACDe has a better interpretation of the weights compared to the other methods since this method uses weights that reflect the importance of the modalities. We focus more on the important modalities and less on the minor modalities by assigning weights based on certainty of probability estimates. WACDe also presents a synergy effect with feature selection and leads to the best performance of the model among all possible combinations of modalities and all combining methods.

Furthermore, we demonstrate WACDe gives the most reliable prediction for a new observation by comparing width of prediction intervals. In practice, it is of fundamental importance to make an accurate prediction, especially in disease classification. All together, WACDe is a practical and promising method for real-world classification problems with multimodal data.

Chapter 5

Conclusion and Future Directions

Learning multimodal data has become increasingly widespread as collecting complementary information from multiple sources becomes easier and principled statistical inferences on multimodal data provide deeper insights into an event of interest. Even though multimodal data convey complementary information from various aspects of an event that gives better understanding, we often face practical challenges since modern data are often high-dimensional, signals from different sources are highly correlated to each other, and the types and dimensions of data coming from multiple sources are different. Accordingly, we considered effective learning methods to address such practical problems in classification. Using deep neural network models, we proposed new combining methods that take the importance of each modality into account.

We performed several numerical experiments on two problems, digit classification and Alzheimer's disease classification. The results showed that multimodal models generally performed better than single modality models, and the proposed combining method, WACDe had the highest overall accuracy. However, there were exceptions when collected data were high-dimensional and variables were highly correlated. We have shown this problem is solved

by using an appropriate feature selection, SIS in our experiments. With feature selection, WACDe showed further improvement in classification accuracy. Furthermore, WACDe gave the most reliable prediction for a new observation compared to other combining methods.

There are many open leads for future research. These include:

Feature selection considering correlation between modalities. In our method, we do not have any restrictions on feature selection methods. We use three different feature selection methods and SIS is the best method in terms of accuracy. For now, we only select sets of important features from each modality independently. However, as we have mentioned, signals from different sources are often highly correlated within and between modalities. Thus, if we select important features by considering the correlation between modalities as well as within modalities, we expect to have an improvement in the model performance.

Considering missing data increases the effective sample size. One of the problems discussed in the Pig Stroke Project and AD classification problem is missing data. It is natural to have missing blocks in multimodal data since not all subjects are available for all of the multiple sources. In the AD classification experiments, we have 628 available subjects for MRI and demographic modalities, and 745 subjects for genetic modality. However, using the subjects who have all information in the three modalities limits us to use only 220 subjects. By considering missing blocks, we can increase the effective sample size, which leads to an increase in information quality and model performance.

An end-to-end learning method. The current model considers feature selection and information integration separately. That gives a set of important features based on different metrics, but might not be the best one in terms of having highest classification accuracy. However, if we search for important features during the model training phase, we could find a set of features that could even increase the classification accuracy. Furthermore, in that way, we expect a type of synergy effect between feature selection and information integration that gives an improvement in the model performance.

Parkinson's disease (PD) prediction. Aging-related central nervous system (CNS) neurodegenerative diseases are a rapidly increasing social and financial burden with the rapid aging of advanced societies. Parkinson's disease (PD) is one of the most common diseases along with AD, and many recent studies have looked at the relationship between PD and trauma (Marras et al., 2014), genetics (Beilina and Cookson, 2016), and the environment (Goldman, 2014). Therefore, multiple sources such as demographic data, genetic data, environmental information, as well as neuroimaging data such as MRI and PET, contain information about PD of a person, and learning such multimodal data leads to a more accurate PD diagnosis and prediction.

Bibliography

Abid, A., Balin, M. F., and Zou, J. (2019). Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*.

Association, A. (2019). 2019 Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 15(3):321–387.

Bartlett, P. L. (1992). Learning with a slowly changing distribution. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 243–252. ACM.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Beilina, A. and Cookson, M. R. (2016). Genes associated with Parkinson's disease: Regulation of autophagy and beyond. *Journal of Neurochemistry*, 139:91–107.

Ben-David, S., Loker, D., Srebro, N., and Sridharan, K. (2012). Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Conference on Machine Learning*, pages 83–90. Omnipress.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., et al. (2009). Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37(4):1705–1732.

Bucak, S. S., Jin, R., and Jain, A. K. (2013). Multiple kernel learning for visual object recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1354–1369.

Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pages 1683–1691. ACM.

Choi, J.-H. and Lee, J.-S. (2019). EmbraceNet: A robust deep learning architecture for multimodal classification. *Information Fusion*, 51:259–270.

Costa, Y. M., Oliveira, L. S., and Silla Jr, C. N. (2017). An evaluation of convolutional neural networks for music classification using spectrograms. *Applied Soft Computing*, 52:28–38.

Dai, Z., Yan, C., Wang, Z., Wang, J., Xia, M., Li, K., and He, Y. (2012). Discriminative analysis of early Alzheimer's disease using multi-modal imaging and multi-level characterization with multi-classifier (M3). *NeuroImage*, 59(3):2187–2195.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B*, 70(5):849–911.

Fan, J. and Lv, J. (2018). Sure independence screening. In *Wiley StatsRef: Statistics Reference Online*, pages 1–8. American Cancer Society.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 221–228. IEEE.

Goldman, S. M. (2014). Environmental toxins and Parkinson's disease. *Annual Review of Pharmacology and Toxicology*, 54:141–164.

Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Graves, A. (2011). Practical variational inference for neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2348–2356.

Gray, K. R., Aljabar, P., Heckemann, R. A., Hammers, A., and Rueckert, D. (2013). Random forest-based similarity measures for multi-modal classification of Alzheimer's disease. *NeuroImage*, 65:167–175.

Gumbel, E. J. (1954). *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*, volume 33. US Government Printing Office.

Gunes, H. and Piccardi, M. (2005). Affect recognition from face and body: Early fusion vs. late fusion. In *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3437–3443. IEEE.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 18th IEEE International Conference on Computer Vision*, pages 1026–1034. IEEE.

Hebert, L. E., Weuve, J., Scherr, P. A., and Evans, D. A. (2013). Alzheimer disease in the United States (2010–2050) estimated using the 2010 census. *Neurology*, 80(19):1778–1783.

Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869. PMLR.

Hinrichs, C., Singh, V., Xu, G., and Johnson, S. C. (2011). Predictive markers for AD in a multi-modality framework: An analysis of MCI progression in the ADNI population. *NeuroImage*, 55(2):574–589.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28:321–377.

Iyengar, G. and Nock, H. J. (2003). Discriminative model fusion for semantic concept detection and annotation in video. In *Proceedings of the 11th Annual ACM International Conference on Multimedia*, pages 255–258. ACM.

Jin, Q. and Liang, J. (2016). Video description generation using audio and visual cues. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 239–242. ACM.

Kelly, M. G., Hand, D. J., and Adams, N. M. (1999). The impact of changing populations on classifier performance. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 367–371. ACM.

Kohannim, O., Hua, X., Hibar, D. P., Lee, S., Chou, Y.-Y., Toga, A. W., Jack Jr, C. R., Weiner, M. W., and Thompson, P. M. (2010). Boosting power for clinical trials using classifiers based on multiple biomarkers. *Neurobiology of Aging*, 31(8):1429–1442.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lee, G., Kang, B., Nho, K., Sohn, K.-A., and Kim, D. (2019a). MildInt: Deep learning-based multimodal longitudinal data integration framework. *Frontiers in Genetics*, 10:617.

Lee, G., Nho, K., Kang, B., Sohn, K.-A., and Kim, D. (2019b). Predicting Alzheimer's disease progression using multi-modal deep learning approach. *Scientific Reports*, 9(1):1–12.

Liu, F., Wee, C.-Y., Chen, H., and Shen, D. (2014). Inter-modality relationship constrained multi-modality multi-task feature selection for Alzheimer's disease and mild cognitive impairment identification. *NeuroImage*, 84:466–475.

Liu, J., Ji, S., and Ye, J. (2012). Multi-task feature learning via efficient l2, 1-norm minimization. *arXiv preprint arXiv:1205.2631*.

Liu, K., Li, Y., Xu, N., and Natarajan, P. (2018). Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lu, D., Popuri, K., Ding, G. W., Balachandar, R., and Beg, M. F. (2018). Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's disease using structural MR and FDG-PET images. *Scientific Reports*, 8(1):1–13.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. ACL.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, volume 30, page 3. Citeseer.

Marras, C., Hincapié, C. A., Kristman, V. L., Cancelliere, C., Soklaridis, S., Li, A., Borg, J., af Geijerstam, J.-L., and Cassidy, J. D. (2014). Systematic review of the risk of Parkinson's disease after mild traumatic brain injury: Results of the international collaboration on mild traumatic brain injury prognosis. *Archives of Physical Medicine and Rehabilitation*, 95(3):S238–S244.

Matthews, I., Cootes, T. F., Bangham, J. A., Cox, S., and Harvey, R. (2002). Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213.

Mu, N. and Gilmer, J. (2019). MNIST-C: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*.

Neal, R. M. (1993). Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, pages 475–482.

Neverova, N., Wolf, C., Taylor, G., and Nebout, F. (2015). Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 689–696. Omnipress.

Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.

Patterson, E. K., Gurbuz, S., Tufekci, Z., and Gowdy, J. N. (2002). CUAVE: A new audio-visual database for multimodal human-computer interface research. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2017–2020. IEEE.

Perrin, R. J., Fagan, A. M., and Holtzman, D. M. (2009). Multimodal techniques for diagnosis and prognosis of Alzheimer's disease. *Nature*, 461(7266):916–922.

Potamianos, G., Luettin, J., and Neti, C. (2001). Hierarchical discriminant features for audio-visual LVCSR. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 165–168. IEEE.

Radová, V. and Psutka, J. (1997). An approach to speaker identification using multiple classifiers. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1135–1138. IEEE.

Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Sanderson, C. and Paliwal, K. K. (2004). Identity verification using speech and face information. *Digital Signal Processing*, 14(5):449–480.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897.

Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 399–402. ACM.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Steinwart, I. and Christmann, A. (2008). *Support Vector Machines*. Springer Science & Business Media.

Suk, H.-I., Lee, S.-W., and Shen, D. (2014). Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, 101:569–582.

Suk, H.-I. and Shen, D. (2013). Deep learning-based feature representation for AD/MCI classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 583–590. Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288.

Verma, G. K. and Tiwary, U. S. (2014). Multimodal fusion framework: A multiresolution approach for emotion classification and recognition from physiological signals. *NeuroImage*, 102:162–172.

Walhovd, K., Fjell, A., Brewer, J., McEvoy, L., Fennema-Notestine, C., Hagler, D., Jennings, R., Karow, D., and Dale, A. (2010). Combining MR imaging, Positron-Emission Tomography, and CSF biomarkers in the diagnosis and prognosis of Alzheimer disease. *American Journal of Neuroradiology*, 31(2):347–354.

Wang, L.-Y., Park, C., Yeon, K., and Choi, H. (2017). Tracking concept drift using a constrained penalized regression combiner. *Computational Statistics & Data Analysis*, 108:52–69.

Wee, C.-Y., Yap, P.-T., Zhang, D., Denny, K., Browndyke, J. N., Potter, G. G., Welsh-Bohmer, K. A., Wang, L., and Shen, D. (2012). Identification of MCI individuals using structural and functional connectivity networks. *NeuroImage*, 59(3):2045–2056.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688. Citeseer.

Wöllmer, M., Metallinou, A., Eyben, F., Schuller, B., and Narayanan, S. (2010). Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, pages 2362–2365.

Wu, Y., Chang, E. Y., Chang, K. C.-C., and Smith, J. R. (2004a). Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 572–579. ACM.

Wu, Y., Lin, C.-Y., Chang, E. Y., and Smith, J. R. (2004b). Multimodal information fusion for video concept detection. In *Proceedings of the 2004 IEEE International Conference on Image Processing*, volume 4, pages 2391–2394. IEEE.

Yan, R., Yang, J., and Hauptmann, A. G. (2004). Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, pages 548–555. ACM.

Yang, Y., Wu, X., and Zhu, X. (2005). Combining proactive and reactive predictions for data streams. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 710–715. ACM.

Zhang, D. and Shen, D. (2012). Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. *NeuroImage*, 59(2):895–907.

Zhang, D., Wang, Y., Zhou, L., Yuan, H., and Shen, D. (2011). Multimodal classification of Alzheimer's disease and mild cognitive impairment. *NeuroImage*, 55(3):856–867.

Zhou, T., Thung, K.-H., Zhu, X., and Shen, D. (2017). Feature learning and fusion of multimodality neuroimaging and genetic data for multi-status dementia diagnosis. In *Proceedings of the 8th International Workshop on Machine Learning in Medical Imaging*, pages 132–140. Springer.

Zhu, Q., Yeh, M.-C., and Cheng, K.-T. (2006). Multimodal fusion using learned text concepts for image categorization. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 211–220. ACM.

Zhu, X., Suk, H.-I., and Shen, D. (2014). Multi-modality canonical feature selection for Alzheimer's disease diagnosis. In *Proceedings of the 17th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 162–169. Springer.

Appendix A

Appendix

A.1 Performance of Single Modality Models in Digit Classification

Table A.1 gives the result of single modality models. The table gives the average accuracy and its standard error of 100 repetitions for seven different single modalities and the zero modality. We observe the scale modality has the highest average accuracy, 0.9641, followed by the zigzag modality, 0.9486. It is clear that those modalities have high accuracy from Figure 4.1 in that digits are very clear and recognizable. On the other hand, the glass blur has the lowest average accuracy 0.7666, among the seven corruptions, and the zero modality has average accuracy 0.1049. This result is sensible in that the zero modality does not contain any information on labels, giving a random guess.

Table A.1: Average accuracy (and its standard error) of 100 repetitions of single modality models.

Corruption	Accuracy
Defocus blur	0.9112 (0.0009)
Glass blur	0.7666 (0.0008)
Impulse noise	0.9201 (0.0005)
Rotate	0.9303 (0.0005)
Scale	0.9641 (0.0002)
Shear	0.9025 (0.0005)
Zigzag	0.9486 (0.0004)
Zero	0.1049 (0.0005)

A.2 Feature Selection in AD Classification

Correlation maps of the different set of selected variables are in Figure A.1 to A.4. The left panel gives correlation maps of CA selected variables, and the right panel gives those of SIS selected variables. Each row of figures indicates MRI, genetic modality and all modalities. Correlation maps of all modalities include selected variables in MRI modality, entire demographic modality, and selected variables in genetic modality. Hence, correlation maps of all three modalities include correlation map of MRI modality in the left top corner and of genetic modality in the right bottom corner. For visualization, we omit the variable names, which can be found in Table A.2 to Table A.10. Surprisingly, variables selected from CA and SIS do not have much overlap. Only `Large` set has a few overlaps, 6 and 1 for MRI and genetic modality, respectively. Bold text in Table A.6 to Table A.8 indicates the overlaps.

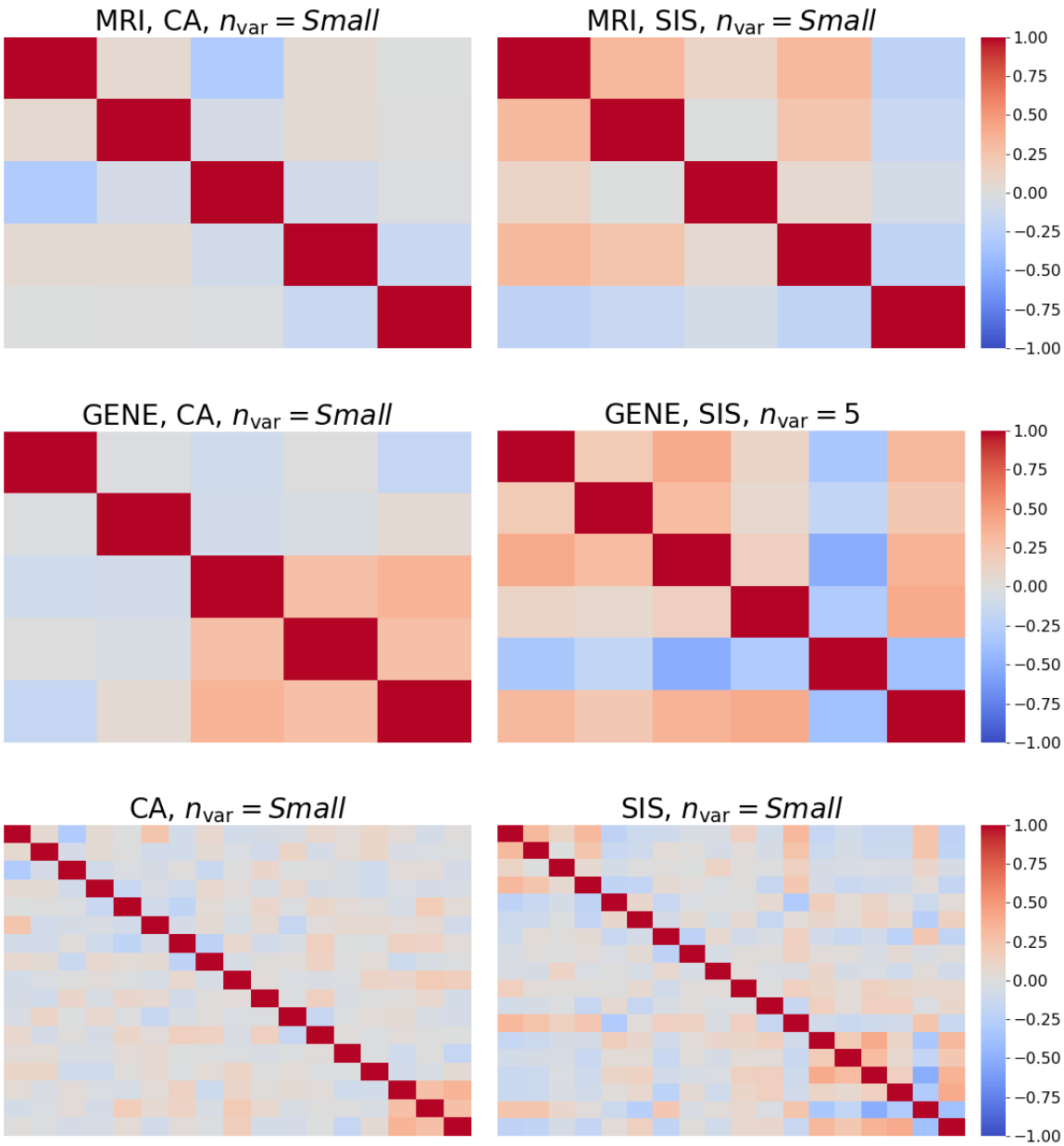

Figure A.1: Correlation maps of `Small` set of selected variables using (left) CA method and (right) SIS method. Each row indicates a different dataset where the first row is selected variables from MRI modality, the second row is that from genetic modality, and the last row is selected variables from MRI, genetic modalities and the entire demographic modality.

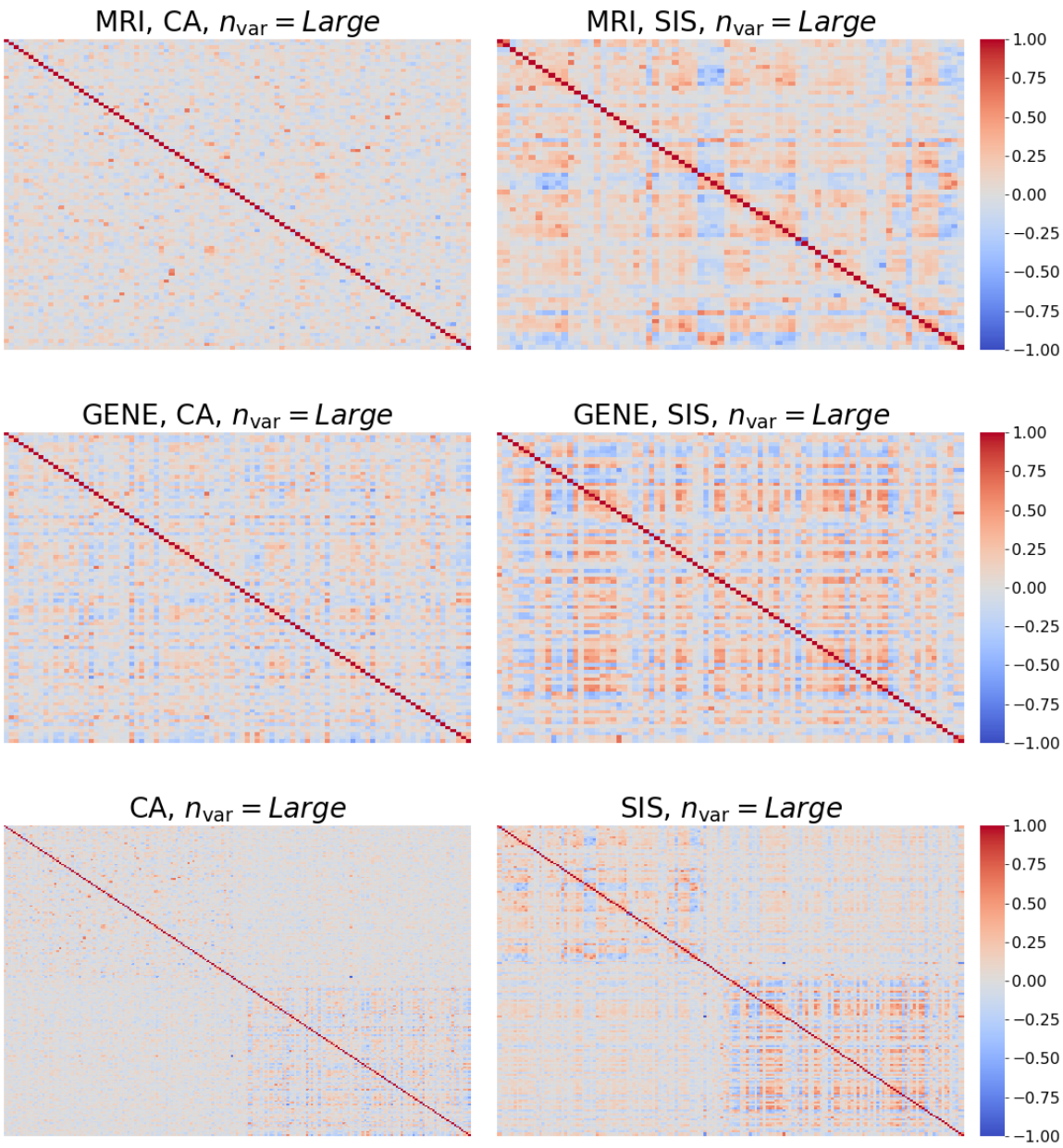

Figure A.2: Correlation maps of `Large` set of selected variables using (left) CA method and (right) SIS method. Each row indicates a different dataset where the first row is selected variables from MRI modality, the second row is that from genetic modality, and the last row is selected variables from MRI, genetic modalities and the entire demographic modality.

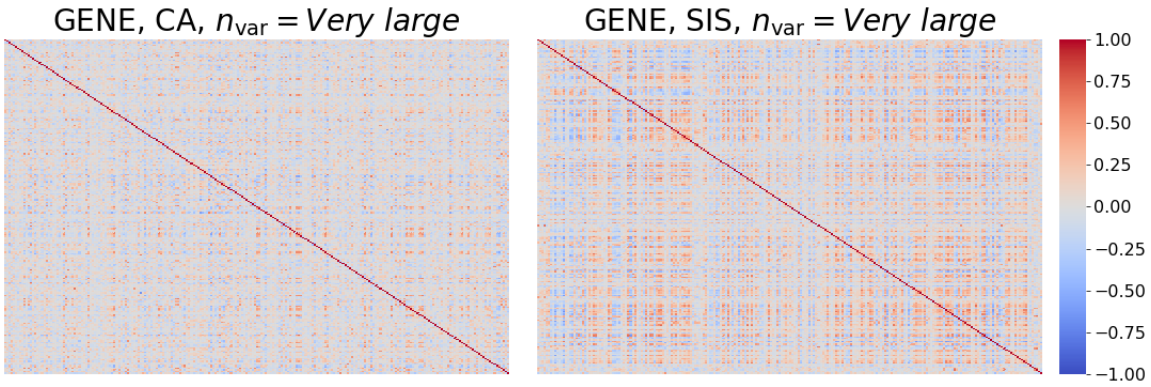

Figure A.3: Correlation maps of `Very large` set of selected variables using (left) CA method and (right) SIS method.

Figure A.4: Correlation maps of `Best` set of selected variables using SIS method. Note that Best combination occurs in CA method with Medium sets.

Table A.2: `Small` set of selected variables from MRI modality.

Method	Selected variables
CA	mean.curvature..mean.1005, travel.depth..skew.2017, FreeSurfer.thickness..mean.1005, FreeSurfer.convexity..MAD.2030, FreeSurfer.convexity..SD.2009
SIS	FreeSurfer.thickness..median.1006, FreeSurfer.thickness..25..1016, FreeSurfer.convexity..kurtosis.2018, FreeSurfer.thickness..median.2007, FreeSurfer.thickness..SD.2006

Table A.3: `Small` set of selected variables from genetic modality.

Method	Selected variables
CA	11726853_a_at, 11725788_a_at, 11746876_a_at, 11759899_at, 11754595_s_at
SIS	11725789_at, 11726854_s_at, 11731778_a_at, 11735339_at, 11754596_x_at, 11759900_at

Table A.4: `Medium` set of selected variables from MRI modality.

Method	Selected variables
CA	area.1022, FreeSurfer.thickness..kurtosis.1016, mean.curvature..MAD.1030, mean.curvature..kurtosis.1014, mean.curvature..mean.2005, travel.depth..skew.1005, FreeSurfer.thickness..median.1015, FreeSurfer.convexity..median.2035, mean.curvature..median.1035, geodesic.depth..MAD.2014, FreeSurfer.convexity..SD.2009, travel.depth..skew.2034, travel.depth..skew.2017, travel.depth..25..2003, mean.curvature..mean.1005, FreeSurfer.convexity..MAD.2030, FreeSurfer.thickness..mean.1005
SIS	mean.curvature..mean.1006, mean.curvature..75..1007, mean.curvature..75..1031, travel.depth..75..1014, FreeSurfer.thickness..median.1016, FreeSurfer.thickness..skew.1031, mean.curvature..75..2015, FreeSurfer.convexity..MAD.2002, FreeSurfer.convexity..kurtosis.2018, FreeSurfer.thickness..median.2007, FreeSurfer.thickness..SD.2006, FreeSurfer.thickness..skew.2028, FreeSurfer.thickness..25..2006, FreeSurfer.thickness..25..2015

Table A.5: `Medium` set of selected variables from genetic modality.

Method	Selected variables
CA	11725737_s_at, 11737279_at, 11737467_at, 11742486_s_at, 11748423_a_at, 11748879_a_at, 11716906_a_at, 11718275_x_at, 11731996_at, 11737784_x_at, 11742203_x_at, 11747153_x_at, 11749725_a_at, 11749840_a_at, 11763143_x_at, 11725788_a_at,11735944_x_at, 11736204_a_at, 11754220_a_at, 11733836_a_at, 11746876_a_at, 11715279_x_at, 11736880_x_at, 11728338_a_at, 11721744_a_at, 11724075_a_at, 11726855_at, 11726853_a_at, 11759899_at, 11754595_s_at
SIS	11715280_s_at, 11717863_a_at, 11719666_a_at, 11722681_at, 11722875_s_at, 11725789_at, 11726854_s_at, 11731778_a_at, 11735170_at, 11735339_at, 11739317_at, 11742204_a_at, 11742487_a_at, 11746877_a_at, 11748160_s_at, 11754596_x_at, 11757733_s_at, 11759900_at

Table A.6: `Large` set of selected variables using CA from MRI modality.

Method	Selected variables
CA	mean.curvature..MAD.1005, mean.curvature..skew.1012, travel.depth..SD.1018, travel.depth..kurtosis.1014, travel.depth..25..1011, geodesic.depth..median.1015, geodesic.depth..median.1025, geodesic.depth..mean.1034, geodesic.depth..kurtosis.1015, geodesic.depth..kurtosis.1028, geodesic.depth..kurtosis.1034, FreeSurfer.convexity..MAD.1005, FreeSurfer.convexity..kurtosis.1002, FreeSurfer.convexity..75..1013, **FreeSurfer.thickness..skew.1015**, FreeSurfer.thickness..kurtosis.1016, FreeSurfer.thickness..25..1015, mean.curvature..SD.2007, mean.curvature..skew.2022, **mean.curvature..75..2031**, travel.depth..mean.2018, travel.depth..kurtosis.2002, travel.depth..kurtosis.2018, geodesic.depth..mean.2008, geodesic.depth..skew.2031, geodesic.depth..kurtosis.2035, FreeSurfer.convexity..MAD.2024, **FreeSurfer.convexity..kurtosis.2022**, FreeSurfer.convexity..75..2031, FreeSurfer.thickness..kurtosis.2002, FreeSurfer.thickness..kurtosis.2017, **area.1024**, mean.curvature..SD.1014, FreeSurfer.convexity..SD.2031, FreeSurfer.convexity..MAD.1008, FreeSurfer.convexity..skew.1013, geodesic.depth..MAD.2021, geodesic.depth..kurtosis.2002, FreeSurfer.convexity..skew.2029, FreeSurfer.convexity..kurtosis.2002, FreeSurfer.thickness..kurtosis.2031, Volume.1005, Volume.1022, travel.depth..mean.2011, geodesic.depth..SD.2005, geodesic.depth..SD.2013, FreeSurfer.convexity..MAD.2016, FreeSurfer.thickness..SD.2005, mean.curvature..kurtosis.1013, mean.curvature..kurtosis.1022, travel.depth..MAD.1005, travel.depth..kurtosis.1009, FreeSurfer.convexity..MAD.1013, FreeSurfer.convexity..kurtosis.1028, FreeSurfer.thickness..skew.1030, travel.depth..75..2018, FreeSurfer.convexity..25..2022, travel.depth..kurtosis.1005, FreeSurfer.thickness..skew.1005, **FreeSurfer.convexity..MAD.1028**, mean.curvature..SD.2025, geodesic.depth..median.2014, Volume.1003, Volume.2011, geodesic.depth..25..1031, mean.curvature..skew.2021, geodesic.depth..25..2003, mean.curvature..kurtosis.1014, mean.curvature..skew.1030, geodesic.depth..mean.1008, FreeSurfer.convexity..75..1008, geodesic.depth..kurtosis.2015, mean.curvature..MAD.1014, travel.depth..kurtosis.2017, FreeSurfer.thickness..skew.1035, mean.curvature..kurtosis.2013, FreeSurfer.convexity..MAD.2008, FreeSurfer.thickness..mean.2017, **FreeSurfer.convexity..kurtosis.2018**, mean.curvature..mean.2005, FreeSurfer.thickness..kurtosis.1013, geodesic.depth..SD.1034, mean.curvature..kurtosis.1007, FreeSurfer.convexity..kurtosis.2017, mean.curvature..MAD.1030, travel.depth..kurtosis.2034, mean.curvature..median.1035, FreeSurfer.convexity..MAD.2009, FreeSurfer.convexity..median.2035, geodesic.depth..skew.2034, mean.curvature..mean.1005, FreeSurfer.thickness..mean.1005, FreeSurfer.convexity..MAD.2030

Table A.7: `Large` set of selected variables using SIS from MRI modality.

Method	Selected variables
SIS	area.1014, **area.1024**, mean.curvature..median.1006, Thickness..thickinthehead..1016 mean.curvature..median.1016, mean.curvature..MAD.1006, mean.curvature..MAD.1016, mean.curvature..mean.1014, mean.curvature..25..1015, mean.curvature..25..1030, mean.curvature..75..1007, mean.curvature..75..1031, travel.depth..median.1016, travel.depth..SD.1002, travel.depth..skew.1012, travel.depth..skew.1013, travel.depth..75..1014, geodesic.depth..SD.1009, geodesic.depth..25..1022, FreeSurfer.convexity..MAD.1006, **FreeSurfer.convexity..MAD.1028**, FreeSurfer.convexity..MAD.1034, FreeSurfer.convexity..SD.1022, FreeSurfer.convexity..75..1014, FreeSurfer.thickness..median.1009, FreeSurfer.thickness..MAD.1028, FreeSurfer.thickness..skew.1006, **FreeSurfer.thickness..skew.1015**, FreeSurfer.thickness..skew.1016, FreeSurfer.thickness..skew.1028, FreeSurfer.thickness..skew.1031, FreeSurfer.thickness..kurtosis.1017, FreeSurfer.thickness..25..1008, FreeSurfer.thickness..25..1034, FreeSurfer.thickness..75..1007, FreeSurfer.thickness..75..1015, area.2025, mean.curvature..median.2008, mean.curvature..median.2025, mean.curvature..MAD.2015, mean.curvature..mean.2006, mean.curvature..mean.2012, mean.curvature..mean.2014, mean.curvature..skew.2006, mean.curvature..75..2007, mean.curvature..75..2030, **mean.curvature..75..2031**, travel.depth..skew.2035, travel.depth..kurtosis.2035, travel.depth..25..2005, travel.depth..75..2022, geodesic.depth..SD.2014, geodesic.depth..25..2002, FreeSurfer.convexity..MAD.2022, FreeSurfer.convexity..MAD.2031, FreeSurfer.convexity..SD.2002, FreeSurfer.convexity..SD.2011, FreeSurfer.convexity..SD.2034, FreeSurfer.convexity..kurtosis.2012, **FreeSurfer.convexity..kurtosis.2018**, **FreeSurfer.convexity..kurtosis.2022**, FreeSurfer.convexity..75..2030, FreeSurfer.thickness..MAD.2003, FreeSurfer.thickness..MAD.2018, FreeSurfer.thickness..mean.2028, FreeSurfer.thickness..SD.2006, FreeSurfer.thickness..SD.2028,FreeSurfer.thickness..skew.2028, FreeSurfer.thickness..skew.2031, FreeSurfer.thickness..25..2003, FreeSurfer.thickness..25..2015, FreeSurfer.thickness..75..2007

Table A.8: `Large` set of selected variables from genetic modality.

Method	Selected variables
CA	11715386_at, 11723297_a_at, 11731441_at, 11731995_x_at, 11733836_a_at, 11734435_at, 11746012_a_at, 11746971_s_at, 11715387_at, 11717862_x_at, 11720806_a_at, 11730578_at, 11731446_s_at, 11737784_x_at, 11739781_a_at, 11747533_a_at, 11754952_x_at, 11758073_s_at, 11763550_x_at, 11718275_x_at, 11737279_at, 11737412_at, 11755787_s_at, 11763143_x_at, 11763228_x_at, 11724074_a_at, 11734685_at, 11741189_x_at, 11744049_at, 11723494_a_at, 11724127_a_at, 11732500_a_at, **11747154_a_at**, 11747764_a_at, 11748159_a_at, 11749166_a_at, 11753666_x_at, 11754829_s_at, 11757022_x_at, 11759814_at, 11759815_a_at, 11764045_at, 11734663_at, 11735625_a_at, 11748423_a_at, 11763238_x_at, 11764063_s_at, 11716906_a_at, 11717876_a_at, 11721672_at, 11749649_x_at, 11753831_x_at, 11735338_at, 11745011_x_at, 11726812_a_at, 11739346_s_at, 11748879_a_at, 11719665_a_at, 11725904_a_at, 11763533_a_at, 11735944_x_at, 11746493_a_at, 11749840_a_at, 11757856_s_at, 11725737_s_at, 11740672_x_at, 11742486_s_at, 11754193_a_at, 11723901_a_at, 11763857_x_at, 11716410_at, 11736204_a_at, 11743066_s_at, 11757732_x_at, 11715279_x_at, 11763836_x_at, 11754220_a_at, 11728338_a_at, 11758437_s_at, 11757232_at, 11737958_x_at, 11742203_x_at, 11721744_a_at, 11747153_x_at, 11731777_a_at, 11736880_x_at, 11749725_a_at, 11726855_at, 11726853_a_at, 11725788_a_at, 11746876_a_at, 11759899_at, 11754595_s_at
SIS	11715132_x_at, 11715280_s_at, 11715965_at, 11717521_x_at, 11717959_a_at, 11718245_at, 11719666_a_at, 11720421_at, 11720552_a_at, 11720802_s_at, 11721056_x_at, 11721632_a_at, 11721805_at, 11722875_s_at, 11722971_a_at, 11723151_s_at, 11725151_at, 11725405_a_at, 11725748_a_at, 11725789_at, 11725888_at, 11726347_x_at, 11726814_x_at, 11726854_s_at, 11726856_at, 11727540_a_at, 11729818_s_at, 11730580_s_at, 11731145_a_at, 11731373_at, 11731778_a_at, 11734222_at, 11734557_a_at, 11735170_at, 11735339_at, 11736897_x_at, 11738110_at, 11740124_at, 11741190_a_at, 11741700_s_at, 11742487_a_at, 11743045_a_at, 11744717_a_at, 11745793_a_at, 11746393_x_at, 11746877_a_at, **11747154_a_at**, 11747176_a_at, 11747444_s_at, 11747754_a_at, 11747765_a_at, 11748160_s_at, 11748344_x_at, 11748676_a_at, 11749650_x_at, 11749723_a_at, 11749740_a_at, 11750247_x_at, 11750253_a_at, 11751271_a_at, 11752332_x_at, 11753357_a_at, 11753676_x_at, 11753685_a_at, 11754596_x_at, 11754685_a_at, 11754953_a_at, 11755009_s_at, 11755091_a_at, 11756774_a_at, 11756785_a_at, 11757684_a_at, 11757780_x_at, 11757968_s_at, 11758074_s_at, 11758542_x_at, 11759131_at, 11759900_at, 11760554_at, 11761779_a_at, 11762163_at, 11762464_at, 11762537_at, 11763144_x_at, 11763792_a_at, 11763858_a_at

Table A.9: `Very large` set of selected variables using CA from genetic modality.

Method	Selected variables
CA	11740163_at, 11752137_a_at, 11719094_x_at, 11747156_x_at, 11743182_x_at, 11745677_a_at, 11740673_a_at, 11720521_at, 11744776_a_at, 11752515_a_at, 11735038_a_at, 11742750_a_at, 11726203_a_at, 11721651_at, 11715834_x_at, 11759653_at, 11730546_x_at, 11754034_a_at, 11760418_a_at, 11717256_at, 11723902_at, 11740086_a_at, 11718876_at, 11725418_a_at, 11760972_x_at, 11727765_at, 11752148_a_at, 11720712_a_at, 11752717_x_at, 11736881_a_at, 11752991_at, 11729571_a_at, 11763229_x_at, 11741655_a_at, 11757538_a_at, 11744066_a_at, 11724534_a_at, 11740420_s_at, 11754478_x_at, 11724912_at, 11716906_a_at, 11746708_at, 11762453_at, 11750554_a_at, 11756167_a_at, 11747739_a_at, 11723363_at, 11732657_a_at, 11744604_at, 11724043_a_at, 11749617_a_at, 11734469_a_at, 11746972_a_at, 11739811_a_at, 11734692_x_at, 11744717_a_at, 11759402_s_at, 11728349_a_at, 11741112_x_at, 11763221_x_at, 11736933_a_at, 11750247_x_at, 11725295_s_at, 11756279_a_at, 11729765_a_at, 11758208_s_at, 11730110_a_at, 11760319_at, 11748880_a_at, 11757793_s_at, 11759378_at, 11722882_at, 11750211_a_at, 11717513_a_at, 11739303_a_at, 11736897_x_at, 11741190_a_at, 11719761_at, 11746493_a_at, 11725348_x_at, 11725906_a_at, 11728863_at, 11716103_a_at, 11745360_a_at, 11715387_at, 11753935_a_at, 11734275_at, 11749502_a_at, 11744855_at, 11729114_s_at, 11732063_a_at, 11732110_a_at, 11728089_a_at, 11719554_at, 11722875_s_at, 11738069_a_at, 11756141_s_at, 11718662_s_at, 11716046_a_at, 11718788_a_at, 11740503_x_at, 11756146_x_at, 11750593_a_at, 11749740_a_at, 11742898_x_at, 11726621_a_at, 11751314_a_at, 11737413_at, 11748457_a_at, 11734518_at, 11744453_s_at, 11738664_at, 11725694_at, 11731497_a_at, 11717247_a_at, 11719479_at, 11758487_s_at, 11748344_x_at, 11744521_x_at, 11756637_a_at, 11752143_a_at, 11736960_a_at, 11724251_x_at, 11763920_at, 11758866_at, 11747765_a_at, 11739979_a_at, 11749070_x_at, 11730226_x_at, 11720923_a_at, 11718407_a_at, 11763534_x_at, 11740278_s_at, 11752332_x_at, 11735170_at, 11739317_at, 11724543_s_at, 11730620_a_at, 11754273_a_at, 11734947_a_at, 11735339_at, 11739809_at, 11745977_a_at, 11733884_a_at, 11724261_a_at, 11761194_at, 11718323_at, 11720713_a_at, 11763858_a_at, 11749467_a_at, 11721745_at, 11737137_a_at, 11741717_a_at, 11758438_s_at, 11738462_at, 11728984_a_at, 11726856_at, 11734081_a_at, 11754069_a_at, 11762122_a_at, 11728176_a_at, 11743827_a_at, 11731903_at, 11741528_a_at, 11726346_a_at, 11751684_a_at, 11723667_at, 11763472_x_at, 11721984_at, 11728339_at, 11737959_at, 11763239_x_at, 11742519_at, 11717655_a_at, 11757233_at, 11735892_a_at, 11742487_a_at, 11727369_a_at, 11719005_a_at, 11729344_at, 11725738_at, 11746512_a_at, 11747546_a_at, 11742204_a_at, 11725789_at, 11728857_at, 11721543_a_at, 11749726_x_at, 11740666_at, 11724806_s_at, 11718850_a_at, 11717863_a_at, 11733815_at, 11715280_s_at, 11748907_a_at, 11724275_s_at, 11750910_a_at, 11724820_a_at, 11758282_s_at, 11763228_x_at, 11731557_at, 11738012_at, 11753667_s_at, 11754942_x_at, 11715965_at, 11744050_s_at, 11756095_x_at, 11720410_s_at, 11749074_a_at, 11737175_at, 11743803_s_at, 11731778_a_at, 11752889_x_at, 11756425_a_at, 11759705_x_at, 11726294_s_at, 11727931_x_at, 11734124_a_at, 11738810_a_at, 11759900_at, 11726492_a_at, 11746877_a_at, 11719898_s_at, 11726854_s_at, 11719666_a_at, 11754596_x_at, 11719412_at, 11742208_a_at, 11754221_s_at, 11725905_a_at

Table A.10: `Very large` set of selected variables using SIS from genetic modality.

Method	Selected variables
SIS	11715132_x_at, 11715150_s_at, 11715280_s_at, 11715388_s_at, 11715965_at, 11716237_s_at, 11716374_s_at, 11716587_at, 11716941_at, 11717521_x_at, 11717625_a_at, 11717672_s_at, 11717721_s_at, 11717742_a_at, 11717780_a_at, 11717959_a_at, 11718245_at, 11718494_at, 11718676_x_at, 11718678_a_at, 11719116_a_at, 11719308_a_at, 11719666_a_at, 11719724_at, 11719917_at, 11720028_x_at, 11720103_a_at, 11720314_s_at, 11720421_at, 11720498_at, 11720552_a_at, 11720636_s_at, 11720782_a_at, 11720802_s_at, 11721540_a_at, 11721632_a_at, 11721633_s_at, 11721648_at, 11721770_x_at, 11721805_at, 11722460_at, 11722526_at, 11722681_at, 11722875_s_at, 11722882_at, 11722971_a_at, 11723151_s_at, 11723363_at, 11723495_a_at, 11723799_x_at, 11723842_s_at, 11723902_at, 11724006_a_at, 11724008_a_at, 11724675_a_at, 11724989_a_at, 11724994_at, 11725151_at, 11725154_a_at, 11725293_at, 11725544_a_at, 11725662_a_at, 11725748_a_at, 11725789_at, 11725888_at, 11725905_a_at, 11726035_at, 11726323_a_at, 11726347_x_at, 11726393_at, 11726583_s_at, 11726615_a_at, 11726774_a_at, 11726854_s_at, 11726856_at, 11727141_s_at, 11727227_a_at, 11727256_a_at, 11727290_a_at, 11727307_x_at, 11727540_a_at, 11727916_a_at, 11728315_at, 11728530_a_at, 11729158_at, 11729374_at, 11729533_s_at, 11729818_s_at, 11729874_at, 11730016_s_at, 11730080_x_at, 11730579_a_at, 11730602_at, 11731145_a_at, 11731373_at, 11731778_a_at, 11732447_x_at, 11732892_at, 11732965_a_at, 11733082_a_at, 11733273_a_at, 11733857_x_at, 11733873_a_at, 11733892_a_at, 11734110_at, 11734222_at, 11734557_a_at, 11735170_at, 11735226_a_at, 11735296_at, 11735315_a_at, 11735339_at, 11735385_s_at, 11735810_at, 11736897_x_at, 11737032_x_at, 11737645_a_at, 11737835_at, 11738001_a_at, 11738110_at, 11738810_a_at, 11739317_at, 11739763_a_at, 11740124_at, 11740319_s_at, 11740595_at, 11741190_a_at, 11741234_a_at, 11741400_a_at, 11741659_a_at, 11741700_s_at, 11741901_x_at, 11741932_a_at, 11742117_a_at, 11742204_a_at, 11742440_at, 11742698_at, 11742862_a_at, 11742911_at, 11743031_a_at, 11743038_at, 11743045_a_at, 11744717_a_at, 11744764_s_at, 11744832_x_at, 11744888_a_at, 11745095_x_at, 11745331_s_at, 11745583_a_at, 11745793_a_at, 11745801_s_at, 11746027_a_at, 11746358_a_at, 11746393_x_at, 11746655_a_at, 11746877_a_at, 11746972_a_at, 11747070_a_at, 11747154_a_at, 11747176_a_at, 11747230_a_at, 11747444_s_at, 11747558_a_at, 11747754_a_at, 11747765_a_at, 11748077_x_at, 11748160_s_at, 11748344_x_at, 11748796_a_at, 11748870_x_at, 11749089_a_at, 11749650_x_at, 11749723_a_at, 11749740_a_at, 11749873_s_at, 11749984_s_at, 11750246_a_at, 11750247_x_at, 11750253_a_at, 11750265_x_at, 11750488_a_at, 11751271_a_at, 11751454_x_at, 11751718_a_at, 11751762_s_at, 11752046_x_at, 11752332_x_at, 11753357_a_at, 11753667_s_at, 11753676_x_at, 11753685_a_at, 11753713_x_at, 11753823_a_at, 11753832_x_at, 11754596_x_at, 11754685_a_at, 11754953_a_at, 11754980_a_at, 11755009_s_at, 11755091_a_at, 11755349_a_at, 11755863_a_at, 11756167_a_at, 11756516_a_at, 11756774_a_at, 11757023_x_at, 11757474_x_at, 11757684_a_at, 11757780_x_at, 11757968_s_at, 11758074_s_at, 11758131_s_at, 11758542_x_at, 11758554_s_at, 11758707_s_at, 11759131_at, 11759705_x_at, 11759900_at, 11760012_at, 11760504_at, 11760554_at, 11761271_x_at, 11761742_at, 11761779_a_at, 11761888_x_at, 11762163_at, 11762227_x_at, 11762464_at, 11762537_at, 11762611_x_at, 11763239_x_at, 11763253_x_at, 11763534_x_at, 11763858_a_at