VARIATIONAL BAYES DEEP NEURAL NETWORK: THEORY, METHODS AND APPLICATIONS

By

Zihuan Liu

A DISSERTATION

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

Statistics – Doctor of Philosophy

2021

ABSTRACT

VARIATIONAL BAYES DEEP NEURAL NETWORK: THEORY, METHODS AND APPLICATIONS

By

Zihuan Liu

Bayesian neural networks (BNNs) have achieved state-of-the-art results in a wide range of tasks, especially in high dimensional data analysis, including image recognition, biomedical diagnosis and others. My thesis mainly focuses on high-dimensional data, including simulated data and brain images of Alzheimer's Disease. We develop variational Bayesian deep neural network (VBDNN) and Bayesian compressed neural network (BCNN) and discuss the related statistical theory and algorithmic implementations for predicting MCI-to-dementia conversion in multi-modal data from ADNI.

The transition from mild cognitive impairment (MCI) to dementia is of great interest to clinical research on Alzheimer's disease (AD) and related dementias. This phenomenon also serves as a valuable data source for quantitative methodological researchers developing new approaches for classification. The development of VBDNN is motivated by an important biomedical engineering application, namely, building predictive tools for the transition from MCI to dementia. The predictors are multi-modal and may involve complex interactive relations. In Chapter 2, we numerically compare performance accuracy of logistic regression (LR) with support vector machine (SVM) in classifying MCI-to-dementia conversion. The results show that although SVM and other ML techniques are capable of relatively accurate classification, similar or higher accuracy can often be achieved by LR, mitigating SVM's necessity or value for many clinical researchers. Further, when faced with many potential features that could be used for classifying the transition, clinical researchers are often unaware of the relative value of different approaches for variable selection. Other than algorithmic feature selection techniques, manually trimming the list of potential predictor variables can also protect against over-fitting and also offers possible insight into why selected features are important to the model. We demonstrate how similar performance

can be achieved using user-guided, clinically informed pre-selection versus algorithmic feature selection techniques.

Besides LR and SVM, Bayesian deep neural network (BDNN) has quickly become the most popular machine learning classifier for prediction and classification with ADNI data. However, their Markov Chain Monte Carlo (MCMC) based implementation suffers from high computational cost, limiting this powerful technique in large-scale studies. Variational Bayes (VB) has emerged as a competitive alternative to overcome some of these computational issues. Although the VB is popular in machine learning, neither the computational nor the statistical properties are well understood for complex modeling such as neural networks. First, we model the VBDNN estimation methodology and characterize the prior distributions and the variational family for consistent Bayesian estimation (in Chapter 3). The thesis compares and contrasts the true posterior's consistency and contraction rates for a deep neural network-based classification and the corresponding variational posterior. Based on the complexity of the deep neural network (DNN), this thesis assesses the loss in classification accuracy due to VB's use and guidelines on the characterization of the prior distributions and the variational family. The difficulty of optimization associated with variational Bayes solution has been quantified as a function of the complexity of the DNN.

Chapter 4 proposes using a BCNN that takes care of the large p small n problem by projecting the feature space onto a smaller dimensional space using a random projection matrix. In particular, for dimension reduction, we propose randomly compressed feature space instead of other popular dimension reduction techniques. We adopt a model averaging approach to pool information across multiple projections. As the main contribution, we propose the variation Bayes approach to simultaneously estimate both model weights and model-specific parameters. By avoiding using standard Monte Carlo Markov Chain and parallelizing across multiple compression, we reduce both computation and computer storage capacity dramatically with minimum loss in prediction accuracy. We provide theoretical and empirical justifications of our proposed methodology.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

Firstly, I would like to express my sincere gratitude to my advisors Prof. Maiti and Prof. Bhattacharya for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. Bender, Prof. Hong, and Prof. Zhu for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

My sincere thanks also goes to Prof. Wang from Western Connecticut State University and Dr. Zhao from Cleveland Clinic, who provided me an opportunity to join their team as researcher, and who gave access to the laboratory and research facilities. Without their precious support it would not be possible to conduct this research.

Thanks to the Michigan State University Graduate School for awarding me a Dissertation Completion Fellowship, providing me with the financial means to complete this thesis. In addition, I would like to thank my parents for their wise counsel and sympathetic ear. You are always there for me. Finally, I could not have completed this dissertation without the support of my friends, LiangLiang, Zhang and CheukYin, Lee, who provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

TABLE OF CONTENTS

LIST OF TABLES viii

LIST OF FIGURES ix

LIST OF ALGORITHMS xi

CHAPTER 1 INTRODUCTION 1
1.1 MCI-to-dementia conversion 1
1.2 Bayesian Deep Neural Networks 2
1.3 Variational inference 3
1.4 Posterior Consistency 5
1.5 Bayesian Compressed Neural Networks 7

CHAPTER 2 A ROLE FOR PRIOR KNOWLEDGE IN STATISTICAL CLASSIFICATION OF THE TRANSITION FROM MCI TO ALZHEIMER'S DISEASE 10
2.1 Introduction 10
2.2 Transition from MCI to dementia 10
2.3 Materials and Data 14
2.3.1 Data Used in Classification 15
2.3.2 Clinical Cognitive Assessment and Genetic data 16
2.3.3 MRI data 19
2.4 Method and Algorithm 19
2.4.1 Logistic Regression 21
2.4.2 Support Vector machine 22
2.4.3 Experimental Design 23
2.5 Results and Analysis 25
2.5.1 Comparison with different modalities 27
2.5.2 Comparison of Pre-selection and L_1 norm 28
2.5.3 Comparison of Groups One and Two 28
2.5.4 Comparison of LR and SVM 29
2.6 Discussion and Conclusion 32

CHAPTER 3 CONSISTENT VARIATIONAL BAYES CLASSIFICATION WITH DEEP NEURAL NETWORKS 38
3.1 Introdution 38
3.2 The Neural Networks Classifier and Likelihoods 38
3.3 Bayesian Inference with Variational Algorithm 41
3.3.1 Prior Choice 41
3.3.2 Variational Inference 42
3.3.3 Black Box Variational Algorithm using score function estimator 44
3.3.4 Control Variate: Stabilizing the stochastic gradient 45
3.3.5 RMSprop Learning Rate: Stabilizing the learning rate. 47

3.3.6 Classification using variational posterior 48
3.4 Posterior and Classification Consistency 48
3.4.1 Posterior consistency and its implication in practice 49
3.4.2 Discussion of the proof 52
3.4.3 Classification consistency 55
3.5 Simulation Studies. 56
3.5.1 Simulation Scenarios 56
3.5.2 Parameters choice for statistical and computational models. 56
3.5.3 Gradient stabilization paramaters. 57
3.5.4 Testing accuracy and convergence. 60
3.5.5 Large number of layers and challenges. 61
3.6 Numerical Properties and Alzheimer's Disease Study 62
3.6.1 Parameters choice for statistical and computational models. 63
3.6.2 Gradient stabilization paramaters. 64
3.6.3 Testing accuracy and convergence. 65
3.6.4 Numerical comparison with popular models 65
3.7 Conclusion and Discussion 66

CHAPTER 4 LEARNING INTRINSIC DIMENSIONALITY OF FEATURE SPACE WITH VARIATIONAL BAYES NEURAL NETWORKS 68
4.1 Introduction 68
4.2 Bayesian neural network for random projection based compressed feature space . . 72
4.2.1 Bayesian neural network model 72
4.2.2 Compression in the feature space with random projections 72
4.2.3 Prior choice 73
4.3 Variational Bayes model averaging for pooling multiple instances of random projection. 74
4.3.1 Bayesian model averaging 74
4.3.2 ELBO derivation 74
4.4 Intrinsic dimensionality and prediction 76
4.4.1 Optimal dimension neighborhood selection 76
4.4.2 Classification based on optimal neigborhood choice 76
4.5 Algorithm and its implementation. 77
4.6 Theoretical results. 78
4.7 Numerical Study 80
4.7.1 Problem setup 80
4.7.2 Datasets 81
4.7.3 Simulated data 82
4.7.4 ADNI Data 82
4.7.5 MNIST Data 83
4.8 Results 83
4.8.1 Optimal dimensional region 83
4.8.2 Comparative Baselines 85
4.8.3 Experimental Results 86
4.9 Conclusion 89

CHAPTER 5 CONCLUSIONS, DISCUSSION, AND DIRECTIONS FOR FUTURE RESEARCH 90
5.1 Conclusions and discussion 90
5.2 Directions for future research 91

APPENDICES 93
APPENDIX A SUPPLEMENT FOR CONSISTENT VARIATIONAL BAYES CLASSIFICATION WITH DEEP NEURAL NETWORKS 94
APPENDIX B SUPPLEMENT FOR LEARNING INTRINSIC DIMENSIONALITY OF FEATURE SPACE WITH VARIATIONAL BAYES NEURAL NETWORKS 118

BIBLIOGRAPHY 130
*

LIST OF TABLES

Table 2.1: Sample Sizes by Timing and Diagnosis: Group One and Two 15

Table 2.2: Clinical Features and Cognitive Assessment Score of Group One 18

Table 2.3: Pre-selected MRI Features of Group One 20

Table 2.4: Modalities 25

Table 2.5: Top 10 features of Group One obtained by L_1 regularization 27

Table 2.6: LR and SVM performance of Group One (Time = 3 years) for models on single- and multi-modal feature sets 30

Table 2.7: LR and SVM performance of Group Two (Time =2 years) for single-data and multi-modal data 31

Table 3.1: Performance of algorithms algorithms 1, 2, 4, 5 for scenario 1. 60

Table 3.2: Performance of algorithms 1, 2, 4, 5 for scenario 2 61

Table 3.3: Performance of algorithms 1, 2, 4, 5 for scenario 1 and scenario 2 for 3 layers. . 62

Table 3.4: Performance of algorithms 1, 2, 4, 5 for ADNI. 65

Table 3.5: Performance for different classifiers. LR: Logistic regression. SVM: Support vector machine. ANN: Frequentist artificial neural network. SG-MCMC: Stochastic gradient MCMC Bayesian neural network 66

Table 4.1: Summary of data,where n, p and c denote the numbers of samples, features and classes. 83

Table 4.2: RPVBNN setting for evaluation 86

Table 4.3: Table: Small simulated data performance 87

Table 4.4: Table: Large simulated data performance 88

Table 4.5: Table: ADNI data performance 88

Table 4.6: MNIST data performance in term of testing accuracy and time (based on 500 epochs) 89

LIST OF FIGURES

Figure 2.1: Comparison of distributions for baseline predictor variables between MCI-S and MCI-C groups. (a) The mean MMSE score in MCI-S is higher than in MCI-C. (b) Mean Learning scores of MCI-C and MCI-S groups are 2.5 and 5. . . 16

Figure 2.2: Comparisons between MCI-S and MCI-C groups on baseline predictor variables. The y-axis of panels (a) through (d) represents the number of participants developing AD. Blue and red bars represent non-converters and converters, respectively. Panel (a) shows a greater number of converters than non-converters for both men and women. Panel (b) shows more than half of MCI-C subjects are APOE4 carriers and approximately 70% MCI-S subjects are non-APOE4 carriers. Panel (c) shows MCI-S subjects have the relatively lower CDR score and MCI-C subjects have higher CDR score. The number of people in MCI-C group has a downward trend as CDR score increases. Panel (d) shows MCI-C subjects have the relatively higher ADASQ4 score. The average of ASADQ4 score of MCI-S and MCI-C subjects are approximately 5 and 8, respectively. 17

Figure 2.3: Flowchart of the LR and SVM method A) ROI-P: ROI level data with Pre-selection; B) ROI-NP: ROI level data with No Pre-selection; C) CCAR: Clinical, Cognitive assessments score, APOE4 and ROI level data. 26

Figure 2.4: Model performance on ROI feature set by number of features for LR and SVM. Panel (a) shows dramatic growth in AUC with LR as the number of features increases from 1 to 30, and then becoming more static at approximately 74% - i.e., as the number of features increases from 30 to 40, but drops significantly when the number of features reaches to 41. Panel (b) shows the AUC increased dramatically as the number of features grows from 1 to 28, but fluctuated after 29. The optimal number of ROI features for both methods are 29 and 28, and their corresponding optimized AUC were approximately 74.0% and 78.0%. . . 36

Figure 2.5: Model performance on CCA feature set by number of features for LR and SVM. Figure (a) shows there is a significant increase in the AUC with LR as the number of features increases from 1 to 5, then there is a slight decrease in the testing accuracy when the number of features is greater than 5. Figure (b) shows the AUC shot up dramatically as the number of features increases from 1 to 4. The optimal number of CCA features obtained by LR and SVM are 5 and 4, and their corresponding optimized AUC are approximately 84.0% and 83.0%. 37

Figure 3.1: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 1 layer. 58

Figure 3.2: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 2 layers. 58

Figure 3.3: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 2 for 1 layer. 59

Figure 3.4: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 2 layers. 59

Figure 3.5: ELBO convergence of algorithms 1, 2, 4, 5 for ADNI. 64

Figure 4.1: Small simulated data: $N = 32$ 83

Figure 4.2: Large simulated data: $N = 128$ 84

Figure 4.3: ADNI data: $N = 128$ 84

Figure 4.4: MNIST data: $N = 128$ 85

LIST OF ALGORITHMS

Algorithm 1: BBVI 45

Algorithm 2: BBVI-CV 46

Algorithm 3: RPVBNN 77

Algorithm 4: BBVI-RMS 94

Algorithm 5: BBVI-CV-RMS 95

CHAPTER 1

INTRODUCTION

Introduction

In this thesis, we develop variational Bayesian deep neural network (VBDNN) and Bayesian compressed neural network (BCNN) and discuss the related statistical theory and algorithmic implementations in the context of classification, such as classifying MCI-to-dementia conversion. Chapter 1 reviews the background, research questions and development of Bayesian neural network (BNN). Chapter 2 introduces the prediction of the transition from mild cognitive impairment (MCI) to dementia for brain images of Alzheimer's Disease using traditional machine learning models (logistic regression and support vector machine). Finally, chapter 3 introduces the VBDNN estimation methodology and the choice of the prior distributions and the variational family. In particular, we discuss the statistical framework for neural networks based classification problem and provide posterior consistency and classification consistency. Chapter 4 introduces a variational Bayes neural network predictive model for addressing the curse of dimensionality (small n large p) by compressing the feature space using random projection matrices. Finally, chapter 5 introduces Conclusions, Discussion, and Suggestions for Future Research.

1.1 MCI-to-dementia conversion

The transition from mild cognitive impairment (MCI) to dementia is of great interest to clinical research on Alzheimer's disease (AD) and related dementias. Alzheimer's disease (AD) is a progressive, age-related, neurodegenerative disease and the most common cause of dementia [147, 148, 67]. Behaviorally, Alzheimer's dementia is commonly preceded by mild cognitive impairment (MCI), a syndrome characterized by declines in memory and other cognitive domains that exceed cognitive decrements associated with normal aging [148, 103]. However, the prodromal symptoms of MCI are not prognostically deterministic: individuals with MCI tend to progress to diagnoses of probable AD at a rate of 8%-15% per year, and many conversions are detectable within 3 years of initial presentation [24, 44, 2]. Research efforts to provide new insights into the incidence of MCI-to-AD conversion have focused largely on clinically or biologically relevant features (i.e., neuroimaging markers, clinical exam data, neuropsychological test scores) and on different methods for statistical classification [145].

1.2 Bayesian Deep Neural Networks

Due to the universal approximation theory of stochastic functions and larger access to computational power, Bayesian Deep Neural Networks (BDNNs) are fashionable in machine learning and statistics for classification and prediction from big data. The BDNNs based prediction has several advantages over standard parametric statistical models. They implicitly consider the interactions or dependence among predictor variables and model the unknown functional relationship between the predictors and responses. For example, we consider classifying Alzheimer's disease status from brain imaging an important biomedical problem. The image features are segmented into voxels or regions of interest (ROI's). Due to their physical adjacency and biological proximities, a simple parametric model or semi-parametric models, such as logistic regression or generalized additive models may not be appropriate. Besides the dependence (spatial) among the predictors, some network structures might be in the feature space while modeling the brain images. The BDNNs can take into account these data features without any explicit assumptions about their dependence structure. Further, these studies often have additional features but in different modes such as genetic and demographic information, brings additional complexity while modeling dependence among the features. Thus, machine learning-based approaches, such as deep neural networks, become useful in this type of application. Bayesian neural networks (BDNNs) have been comprehensively studied by [7], [95], [71], and many others. More recent developments which establish the efficacy of BDNNs can be found in [120], [93], [61], [80], [64] and the references therein. The estimation of the posterior distribution is a key part of Bayesian inference and represents the information about the uncertainties for both data and parameters. However, the exact analytical solution for the posterior distribution is intractable as the number of parameters is very large and the functional form of a neural network does not lend itself to exact integration (see [11]). Several approaches have been proposed for solving posterior distribution of weights of BDNNs, based on both optimization-based techniques such as variational inference (VI), and sampling-based approach, such as Markov Chain Monte Carlo (MCMC).

1.3 Variational inference

Markov Chain Monte Carlo (MCMC) techniques are typically used to obtain sampling-based estimates of the posterior distribution. Indeed, BDNNs with MCMC have not seen widespread adoption due to computational cost in terms of both time and storage on a large dataset, [66, 94, 132, 139]. In contrast to MCMC, VI tends to converge faster, and it has been applied to many popular Bayesian models, such as factorial models and topic models [79, 9, 8]. We want to take a variational approximation approach for posterior estimation in the context of deep neural network classification models. The basic idea of VI is that it first defines a family of variational distributions and then minimizes the Kullback-Leibler (KL) divergence with respect to the variational family. Many recent works have discussed the application of variational inference to Bayesian deep neural networks e.g., [50], [11], [121]. Although there is a plethora of literature on variational inference for neural networks, the theoretical properties of the variational posterior in BDNNs remain relatively unexplored and this limits the use of this powerful computational tool beyond the machine learning community.

Some of the previous works that focused on theoretical properties of variational posterior include the frequentist consistency of variational inference in parametric models in presence of latent variables (see [135]). Optimal risk bounds for mean-field variational Bayes for Gaussian mixture (GM) and Latent Dirichlet allocation (LDA) models have been discussed in [99]. The work of [142] proposed α-variational inference Bayes risk for GM and LDA models. The [149] discusses the variational posterior contraction rates in Gaussian sequence models, infinite exponential families and piece-wise constant models. The works of [105] and [122] study the posterior contraction rates for Bayesian sparse deep neural network models under spike and slab priors. Three more closely related works which study variational posterior are: (1) [3] discusses the contraction rates of VB in sparse BDNN models with spike and Gaussian slab priors and mean-field spike and Gaussian slab variational family (2) [22] discusses the contraction rates of a tempered VB solution with spike and Gaussian slab priors and mean-field spike and Gaussian slab variational family and (3) [6] discusses consistency of VB for single-layer neural network models with Gaussian priors and mean field Gaussian variational family. All these three works focus on a regression setting unlike our classification set up, which in turn allows for the generalization of VBDNNs to generalized linear models. Further, none of these works discuss computational details and the theoretical guidelines of BDNN to achieve the desired level of accuracy.

The work of [6] does not establish contraction rates or deal with deep networks. The works of both [22] and [3] establish contraction rates, however, their notion of convergence do not agree with the classical definition of posterior contraction as established in Theorem 2.1 in [47] wherein one needs to find the rates at which variational posterior gives probability to shrinking Hellinger neighborhoods of the true density. The notion of contraction as used in [22] considers the contraction rate of the quantity $\theta/||f_0 - f_\theta||_\infty$ instead of Hellinger neighborhoods of the true density. The work of [3] on the other hand considers the posterior expectation of the square of the Hellinger distance instead of the posterior probability of shrinking Hellinger neighborhoods. Note, in terms of the notion of consistency, our work is similar to those of [105] (Theorem 5.1) and [122] (Theorem 2.1) but in the context of variational posterior instead of the true posterior.

The derivation of the posterior contraction rates in the classical sense provided us the additional advantage to quantify the loss incurred due to the use of VI approach over MCMC approach on the classification accuracy of the BDNN's, a result which to the best of our knowledge, does not exist in the literature. Additionally, our current work does not assume a sparsity constant s^* which can control the overall complexity of the model. We instead start with a dense network and break down the complexity of a deep neural network into three components (1) the number of layers (2) the number of nodes and (3) the strength of interactions between active nodes. Then, we study the impact of each of these components on the consistency, contraction rates and classification accuracy of the variational posterior based on BDNN. Finally, this thesis adopts the control variates and adaptive learning rate approach as proposed in [107] to BDNNs. This allowed us to analyze the stability of the numerical optimization used for obtaining a variational Bayes solution as a function of the complexity of the model. We like to emphasize that, unlike the high-dimensional regression model, the sparsity constant s^* is not well defined in DNN as the layers can be thought of a sequence and there should not be any gap between layers.

1.4 Posterior Consistency

To evaluate the validity of a posterior in non-parametric models, one must establish its consistency and contraction rates. Unlike any of the previous works, we establish the posterior consistency and contraction rates of the variational posterior in the classical sense, see theorems 3.4.1 and 3.4.2. For a simple consistency result, one needs to show that the posterior concentrates around the Hellinger neighborhood of the true density function with overwhelming probability. A deep neural network model for which the input feature space and number of layers is fixed enjoys consistency properties irrespective of the true function under study as long as the total number parameters of grow at a rate smaller than the sample size n. In this direction, we establish that posterior probability of an ε- Hellinger neighborhood grows at the rate $1 - 2e^{-n\varepsilon^2/2}$ in contrast to the slower $1 - \nu$, $\nu \to 0$ as $n \to \infty$ rate for the variational posterior. For establishing the rates of contraction, one needs to show that the posterior concentrates around shrinking Hellinger neighborhoods of the true density with overwhelming probability. To determine the rates of contraction, one needs assumptions on the neural network solution that approximates the true function and the number of total parameters being less than n. Treating the input feature space as the number of nodes in the 0^{th} layer, we found that the approximating neural network solution must satisfy three main properties (1) the number of layers grows at a rate smaller than $\log n$ (2) the number of nodes in each layer are well controlled (3) the number of connections between active nodes is well controlled. In this direction, we establish that the true posterior probability of a shrinking $\varepsilon\epsilon_n$- Hellinger neighborhood grows at the rate $1 - 2e^{-n\varepsilon^2\epsilon_n^2/2}$ in contrast to the slower $1 - \nu$ rate for the variational posterior.

For BDNN, we next establish the connection between posterior contraction rate and classification accuracy. In this direction, we first show that the classification accuracy of a consistent posterior asymptotically approaches the Bayes classifier's classification accuracy. With no assumptions on the true function, we show that a deep neural network model for which the number of input features and number of layers is fixed, we show that the convergence rates of the classification accuracy are the same for both variational approximation and true posterior. However, under suitable assumptions on the approximating neural network solution as described in the above paragraph, we establish that the classification accuracy of variational posterior approaches to the classification accuracy of the Bayes classification at the rate $\epsilon_n^{2/3}$ in contrast to the higher rate of ϵ_n for the true posterior. This interesting theoretical discovery quantifies the loss to the use of variational posterior instead of using the true posterior density.

We provide prior elicitation for Bayesian estimation. Our detailed mathematical treatment provides theoretical guidelines for selecting the prior distributions that might affect prediction accuracy. For example, even one works with fairly vague priors, there is a limit for choosing the hyper-parameter values to achieve a desired level of consistency. We also discuss how the choice of variational distribution along with the prior distribution affects the posterior consistency.

Besides the theoretical validation, the challenges of implementing a VI based approach is two folds: (1) the choice of the variational family (2) the optimization of the KL-divergence. For the first issue, we show that a simple mean-field Gaussian variational family suffices for posterior consistency along with good numerical performance. For the second issue, the current paper discusses the associated computational challenges of using a VI approach and provides statistically principled guidelines to overcome the same. We first adapted the black-box variational inference (BBVI) algorithm in [107] to the classification based on BDNN's and used Monte Carlo estimates of the gradient of the evidence lower bound (ELBO) for stochastic optimization of the variational parameters. We then adapted the control variates approach as in [107] to allow for faster convergence to the solution. We found that control variates offers a great deal of improvement in terms of time management when using one or two layers. With increase in the number of layers, it was observed that using adaptive learning rates like Adagrad as in [107] can offer huge advantage to allow for stable optimization. We, however, propose the use of the RMSprop due to its superior performance over Adagrad. Finally we discuss in detail the learning rate selection, number of Monte-Carlo samples and other tuning controls in the context of variational Bayes implementation.

1.5 Bayesian Compressed Neural Networks

Bayesian neural networks (BNNs) have achieved state-of-the-art results in a wide range of tasks, especially in high dimensional data analysis, including image recognition, biomedical diagnosis and others. One of the major disadvantages in using neural networks and deep networks is that they require a huge number of training data due to a large number of inherent parameters [140, 45]. For example, high-dimensional neural networks have been widely applied with regularization, dropout techniques or early stopping to prevent overfitting [118, 143]. Furthermore, most commonly used dimensional reduction techniques include Lasso [17], Ridge [58], Elastic net [152], Sparse group lasso [116], Bayesian Lasso [98], Horseshoe prior [16], principal component analysis [115]. Even though the l_1 and l_2 norm can force the weights to become zero or small, they do not have the regularizing effect of making the computed function simpler [70]. Additionally, all these methods rely on the use of whole data, which severely increases the cost of both computation and memory storage.

In this thesis, we propose the use of a BNN on a compressed feature space to take care of the large p small n problem by projecting the feature space onto a smaller dimensional space using a random projection matrix. Random-projection (RP) is a powerful dimension reduction technique which uses RP matrices to map data into low-dimensional spaces. The use of RP in high dimensional statistics is motivated from the Johnson–Lindenstrauss Lemma [27] which states for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, $\epsilon \in (0,1)$ and $d > 8 \log n/\epsilon^2$, there exists a linear map $f : \mathbb{R}^p \to \mathbb{R}^d$ such that $(1-\epsilon)||x_i - x_j||_2^2 \leq ||f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)||_2^2 \leq (1+\epsilon)||x_i - x_j||_2^2$ for $i, j = 1, \cdots, n$. The properties of the RPs and their applications to statistical problems were furthered explored in [33, 13], etc..

To reduce the sensitivity to the choice of random matrices, one must pool information obtained from multiple projections. In this thesis, we adopt a Bayesian model averaging approach for combining information across multiple instances RP based neural networks. There are two main challenges of implementing Bayesian modeling averaging (1) due to the convoluted structure of the neural network likelihood, closed form expressions do not exist for the posterior distribution under each model (2) posterior distribution of model weights is completely intractable and no closed form solutions exist. Thereby, the implementation of standard Markov Chain Monte Carlo (MCMC) is next to impossible. Further, the computation and storage cost associated with MCMC implementation is humongous since each posterior model weight depends on the remaining models' posterior model weight.

To address the challenges of MCMC implementation, we use variational inference (VI) [63, 9] approach to provide an approximate solution for Bayesian model averaging (BMA) to allow for combining of BNNs with multiple instances of compression on the feature space. There has been a plethora of literature implementing variational inference in the neural networks [10]. However, their implementation makes use of the entire feature space, thereby putting a great burden on computational stability and memory storage. We address two main challenges in this thesis (1) developing a variational Bayes (VB) solution for BNNs with compressed feature space (2) providing a VB solution for doing BMA across multiple instances of RP. Further, we establish the posterior contraction rates for the variational posterior for classification (the theory is extendable to regression set up with minor modifications). In this direction, we provide characterization of the prior, variational posterior and the RP matrix which guarantees the convergence of the variational Bayes neural network (VBNN) under the compressed feature space to the true density of the observations.

The main advantage of implementing a BMA approach is that it gives the posterior model weights under each compression of feature space. The so obtained posterior model weights in turn induce a probability distribution on the projected dimension of the feature space. The mode of this probability distribution concentrates around the intrinsic dimensionality of the feature space.

The BMA approach is then applied to a pool of RP matrices whose projected dimension lies in a neighborhood of the intrinsic dimensionality to improve the prediction performance. Finally, we study the numerical behavior of the proposed procedure in the light of simulation and real data sets. To the best of our knowledge, no literature provides theoretical guarantees and computation algorithms of VBNNs with compressed feature space.

CHAPTER 2

A ROLE FOR PRIOR KNOWLEDGE IN STATISTICAL CLASSIFICATION OF THE TRANSITION FROM MCI TO ALZHEIMER'S DISEASE

2.1 Introduction

The transition from mild cognitive impairment (MCI) to dementia is of great interest to clinical research on Alzheimer's disease (AD) and related dementias. This phenomenon also serves as a valuable data source for quantitative methodological researchers developing new approaches for classification. However, the growth of machine learning (ML) approaches for classification may falsely lead many clinical researchers to underestimate the value of logistic regression (LR), which often demonstrates classification accuracy equivalent or superior to other ML methods. Further, when faced with many potential features that could be used for classifying the transition, clinical researchers are often unaware of the relative value of different approaches for variable selection. Using data from the Alzheimer's Disease Neuroimaging Initiative (ADNI), the present study investigated automated and theoretically-guided feature selection techniques in the context of LR and support vector machine (SVM) classification methods for predicting conversion from MCI to dementia. The present findings demonstrate how similar performance can be achieved using user-guided, clinically informed pre-selection versus algorithmic feature selection techniques. These results show that although SVM and other ML techniques are capable of relatively accurate classification, similar or higher accuracy can often be achieved by LR, mitigating SVM's necessity or value for many clinical researchers.

2.2 Transition from MCI to dementia

Alzheimer's disease (AD) is a progressive, age-related, neurodegenerative disease and the most common cause of dementia [147, 148, 67]. Behaviorally, Alzheimer's dementia is commonly preceded by mild cognitive impairment (MCI), a syndrome characterized by declines in memory and other cognitive domains that exceed cognitive decrements associated with normal aging [148, 103]. However, the prodromal symptoms of MCI are not prognostically deterministic: individuals with MCI tend to progress to diagnoses of probable AD at a rate of 8%-15% per year, and many conversions are detectable within 3 years of initial presentation [24, 44, 2]. Research efforts to provide new insights into the incidence of MCI-to-AD conversion have focused largely on clinically or biologically relevant features (i.e., neuroimaging markers, clinical exam data, neuropsychological test scores) and on different methods for statistical classification [145].

For clinical researchers, however, there may be a tendency to conflate more sophisticated, novel analytic approaches and the value of multimodal information from neuroimaging and clinical assessment. Moreover, whereas statisticians may inherently understand the comparability of different quantitative approaches, the novelty of both big data and data-driven approaches for studying MCI-to-AD conversion may lead clinical researchers to assume that such data-driven methods are inherently superior to more theoretically-grounded approaches. Thus, the value of using extant findings and domain expertise to help guide and constrain the application of newer data-driven approaches capable of capitalizing on emergent big data may be a particularly important consideration for clinical researchers.

Statistical classification in clinical research has traditionally utilized binary logistic regression (LR). However, key attributes of modern clinical and neuroimaging data, including high dimensionality and the presence of ground truth estimates of pathology and diagnosis provide new opportunities for quantitative research. This has led to a substantial expansion in the use of data from the Alzheimer's Disease Neuroimaging Initiative (ADNI; http://adni.loni.usc.edu) for quantitative research and methodological development, particularly by researchers utilizing and developing prediction and classification methods in machine learning (ML). Besides LR, support vector machine (SVM) has quickly become the most common type of ML classifier for diagnostic prediction and classification with ADNI data. In general, LR works well when the data is linearly separable and the number of data is greater than the number of features. Moreover, SVM and LR have similar misclassification rates (MCRs) when used to diagnose malignant tumors from imaging data [19, 30].

Indeed, before the rapid expansion of ML research and applied work over the past decade, many clinical researchers and those outside of engineering and mathematically intensive disciplines had little exposure to classification approaches other than LR. Despite its growing popularity, the relative benefits of SVM or other forms of ML [101, 87] over LR for such classification are not always apparent. Although this may be of little surprise to statisticians and quantitative researchers, such perspectives are often lost on clinical researchers, whose implicit beliefs in the superiority of ML is driven by the volume of publications, rather than through training or empirical demonstration.

Most efforts to develop new classification methods for prediction of MCI-to-AD conversion are well suited to integrate measures from multiple sources such as demographics, clinical rating scores, neuropsychological testing, neuroimaging, genetic markers, etc. However, identifying which combination of features most accurately classifies conversion from MCI to AD is a key challenge for ADNI, and may vary by method. The L_1 norm regularization method (i.e., L_1) is a highly used feature selection technique for LR and SVM. L_1 is popular for addressing circumstances in which the number of features is quite large or even larger than the sample size. Despite some risk of abusing the statistical terminology, the problem is often generically referred to as the “small n, large p" or high dimensional problem. The L_1 technique has dual impacts, namely the algorithm can (i) optimize a higher number of parameters in comparison to sample size, and (ii) reduce the effective number of parameters (i.e., performing variable selection). This powerful technique has been implemented in ADNI data with LR [144]. Furthermore, L_1 and other algorithmic feature selection methods used in ML suffer from one key limitation: they are agnostic to theoretical considerations, and as such, they cannot interpret why selected features are meaningful and important to the model. When sampling from a large pool of features, the algorithmic approaches fail to consider prior knowledge of features and their associations with the relevant systems in variable selection. Therefore, domain expertise and prior knowledge may afford additive or differential value for choosing features and interpreting model results over algorithmic feature selection methods alone.

However, most real-world problems occur in the context of additional information about each potential feature and its conceptual relationship with the phenomenon being classified. Other than using L_1 feature selection, manually trimming the list of potential predictor variables can also protect against over-fitting, and also offers potential insight into why selected features are important to the model. When guided by prior knowledge, user-guided or 'manual' feature selection may be a valuable additional step to help minimize potentially spurious effects. This perspective is frequently lost on applied researchers, as most commonly used variable selection algorithms are context-free – that is, they only look at relationships within the data set, and cannot factor in the wider meanings of variables. Furthermore, this also means that automated algorithms may identify relationships among a large number of predictor variables that are spurious and are unlikely to generalize outside the data set. Although there are a vast number of potential neuroimaging features in ADNI data, the present study focused only on regional brain volumes segmented from structural magnetic resonance imaging (MRI) data, the most commonly used neuroimaging datatype for classifying MCI-to-dementia conversion. In contrast to prior studies that used a limited set of volumetric brain features, the present study utilized data generated by modern multi-atlas segmentation methods and analyses included up to 259 features - anatomically specific gray and white matter volumes. However, the large pool of extant findings from studies evaluating regional brain MRI volumetry in prediction and classification of MCI-to-dementia conversion using both limited and expansive feature sets also provides a valuable set of priors for relevant brain regions [18, 43, 123, 91, 42, 108]. Thus, applied researchers are often left with the conundrum of more confirmatory approaches that use few regions in classification or more exploratory methods in which prior findings have little value.

The present study addressed two questions regarding commonly-used classification approaches for predicting MCI-to-dementia conversion in multi-modal data from ADNI. First, we compared performance accuracy of binary LR with SVM in classifying MCI-to-dementia conversion. Second, we asked if applying prior knowledge in feature selection outperforms algorithmic variable selection alone. We hypothesized that 1) LR would perform comparably to SVM, and 2) user-guided variable selection would outperform algorithmic variable selection alone. This work is intended to demonstrate to clinical researchers the benefit of using ML in an informed fashion, rather than as a 'black box' that obscures clear interpretation. Moreover, we wish to emphasize that this study is not meant to highlight a novel innovation in quantitative methods, but rather to provide an important example to applied researchers regarding the comparable value of ML methods and importance of domain expertise in classification with ADNI data.

2.3 Materials and Data

The data used in the preparation of this study were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI). ADNI is an ongoing joint public-private effort to utilize neuroimaging, other biological markers, and clinical and neuropsychological assessment to measure the incidence and progression of MCI to early dementia. Determination of sensitive and specific markers of preclinical AD and MCI is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as reduce the time and cost of clinical trials. Data in the present study came from all sites across the U.S and Canada. All ADNI study participants included in the present analyses were between 55 and 90 years old, spoke English or Spanish as their native language, and had a study partner who provided an independent assessment of functioning.

This study used a subset of the 819 participants from ADNI-1 diagnosed with MCI at baseline and for whom the data from demographic, clinical cognitive assessments, APOE4 genotyping, and MRI measurements were also available. To evaluate differences in classification performance due to participant inclusion and drop out, we subdivided the sample into two overlapping groups. After applying other criteria for inclusion, Group One included all patients whose follow-up period was at least 36 months (n = 265); Group Two consisted of all patients with follow-up assessments at 24 months (n = 308). Although the ADNI study protocol includes additional follow-up visits at 6-month intervals, the present study only evaluated baseline data for features (i.e., clinical, neuropsychological, brain volumetric) in classification analyses. In addition, identification of stable vs. converting clinical outcomes only considered longer-term outcomes based on assessments at 2 and 3 years after baseline. The final samples included 265 and 308 study participants in Groups One and Two, respectively, who met criteria for inclusion. Both Groups included participants who were stable in their diagnosis (MCI-S) and those who converted to a diagnosis of dementia over the 2 or 3 years (MCI-C). Table 2.1 shows the participant characteristics. Diagnostic criteria for MCI included an MMSE score at baseline between 24 and 30, a CDR score of 0.5, and a subjective memory complaint, in addition to objective memory loss measured by education-adjusted scores on the Logical Memory II subscale of the Wechsler Memory Scale, generally preserved activities of daily living and no dementia. The diagnostic criteria for dementia were an MMSE score between 20 and 26, and a CDR score between 0.5 and 1.0. The clinical status of each participant diagnosed with MCI was re-assessed at each follow-up visit and updated to reflect one of several outcomes (e.g., MCI or dementia subtypes). The MCI-C and MCI-S group designations were based on this follow-up clinical diagnosis and marked as either 1 for MCI-C or 0 for MCI-S in classification study.

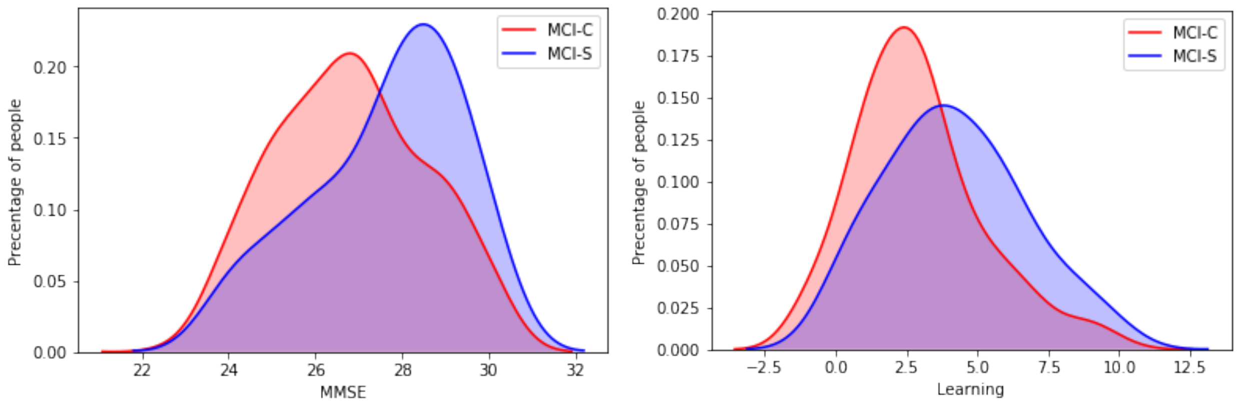
Table 2.1: Sample Sizes by Timing and Diagnosis: Group One and Two

Group	Time	# MCI-S (y=0)	# MCI-C (y=1)	# Total patients
One	36 months	101	164	265
Two	24 months	122	186	308

Table shows the number of MCI-C, MCI-S and total subjects in Group One and Two. The number of MCI-C patients is higher than MCI-S patients in both groups.

2.3.1 Data Used in Classification

Evaluation of extant reports of common predictors of conversion from MCI to dementia focused on dimensions of neuropsychological test performance, clinical assessment, genetic data, and regional brain volumes. In the present study, we first divided these variables into two sets of features, with all non-brain volumetric variables in one set and all variables representing regional brain volumes in a second set. In addition, we created a third set of features from the volumetry feature set that only included 26 of the 259 brain volumes. Henceforth, we refer to models that only include one of these three feature sets as 'single-modality,' whereas models that combine brain and non-brain feature sets are referred to as 'multi-modal.'



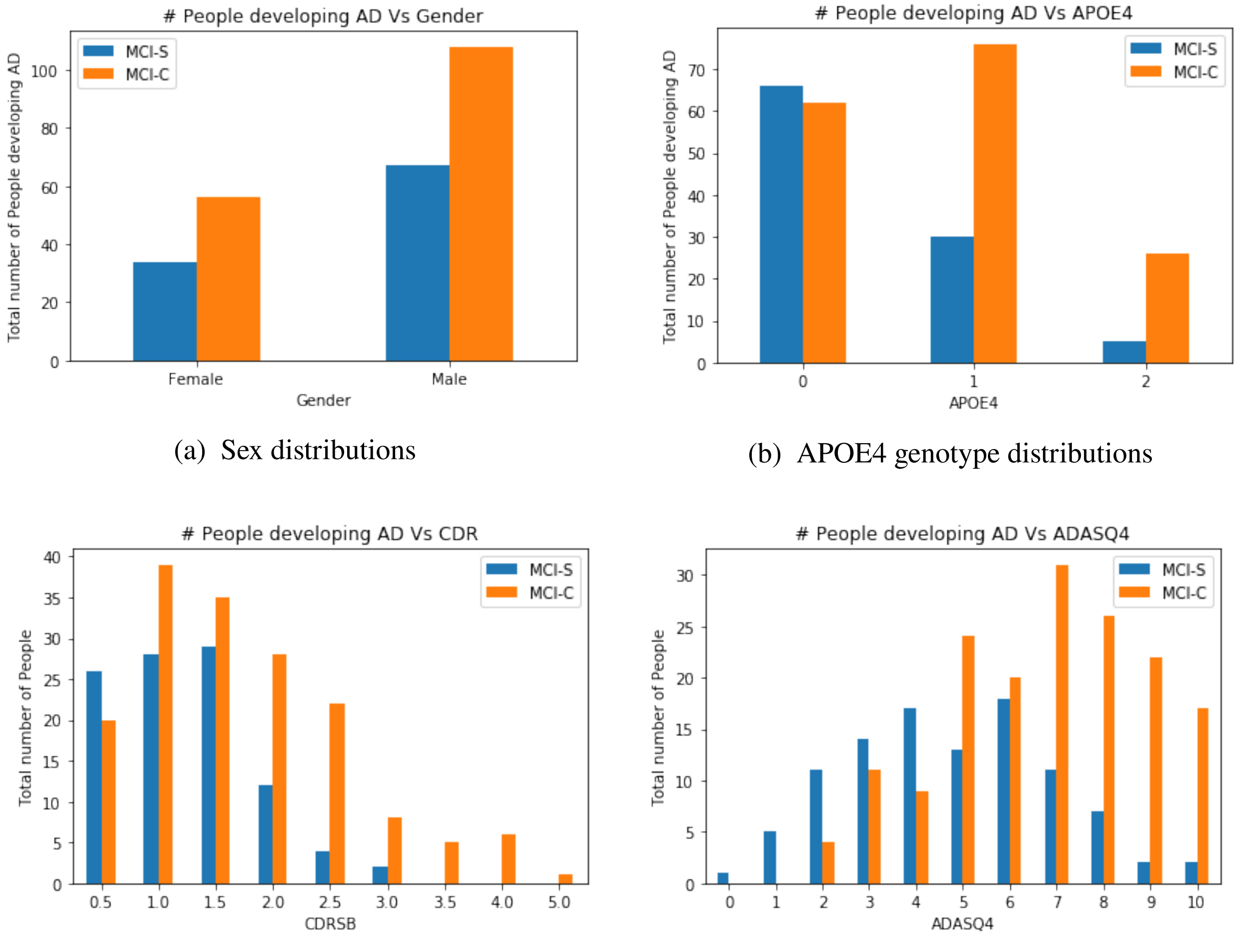
(a) MMSE scores in MCI-C and MCI-S groups

(b) Learning in MCI-C and MCI-S groups

Figure 2.1: Comparison of distributions for baseline predictor variables between MCI-S and MCI-C groups. (a) The mean MMSE score in MCI-S is higher than in MCI-C. (b) Mean Learning scores of MCI-C and MCI-S groups are 2.5 and 5.

2.3.2 Clinical Cognitive Assessment and Genetic data

We considered a total of 19 clinical features as potential predictors of MCI-to-AD progression in our classification analyses. These included the following assessment scores: the Mini Mental State Examination (MMSE), Clinical Dementia Rating Sum of Boxes (CDR-SB), Alzheimer's Disease Assessment Scale-cognitive sub-scale (ADAS-cog), Functional Activities Questionnaire (FAQ) measures of activities of daily living, Trail Making Test-B (TRABSCOR), the immediate and delayed recall components of the Rey Auditory Verbal Learning Test (RAVLT), the Digit-Symbol Coding test (DIGT) and the Digit Symbol Substitution Test from the Preclinical Alzheimer Cognitive Composite (mPACCdigit). We also considered genotype for carriers of the epsilon-4 allele of the apolipoprotein E (APOE) gene [145] as a genetic predictor in this study. Table 2.2 summarizes all 19 clinical, demographic and genetic features used in this study. Preliminary comparison of six clinical and genetic predictors by MCI-C and MCI-S subgroups showed five of them (APOE4, ADAS4, CDR, MMSE and RAVLT.learning) significantly differ between the



(a) Sex distributions

(b) APOE4 genotype distributions

(c) CDR distributions

(d) ADAS distributions

Figure 2.2: Comparisons between MCI-S and MCI-C groups on baseline predictor variables. The y-axis of panels (a) through (d) represents the number of participants developing AD. Blue and red bars represent non-converters and converters, respectively. Panel (a) shows a greater number of converters than non-converters for both men and women. Panel (b) shows more than half of MCI-C subjects are APOE4 carriers and approximately 70% MCI-S subjects are non-APOE4 carriers. Panel (c) shows MCI-S subjects have the relatively lower CDR score and MCI-C subjects have higher CDR score. The number of people in MCI-C group has a downward trend as CDR score increases. Panel (d) shows MCI-C subjects have the relatively higher ADASQ4 score. The average of ASADQ4 score of MCI-S and MCI-C subjects are approximately 5 and 8, respectively.

groups, whereas one (SEX) does not. Fig 2.1 and 2.2 illustrate the distribution of these predictors for both groups. Overall, in comparison to MCI-S participants, those in the MCI-C group were more cognitively and functionally impaired at baseline, exhibited greater verbal memory impairments, and included a greater proportion of APOE4 carriers.

Table 2.2: Clinical Features and Cognitive Assessment Score of Group One

Characteristics	MCI-S	MCI-C	Test statistic	P-value
Age(years)	74.34 ± 7.78	74.84 ± 6.83	-0.528	> 0.5[a]
Education(years)	15.57 ± 2.94	15.73 ± 2.91	-0.527	> 0.5[b]
Sex, % female	33.67%	34.14%	0	1[b]
APOE4 carriers %	34.65%	62.19%	17.900	< 0.001[a]
CDRSB	1.23 ± 0.61	1.72 ± 0.92	-5.237	< 0.001[a]
MMSE score	27.61 ± 1.74	26.82 ± 1.71	3.645	< 0.001[a]
ADAS11	8.89 ± 3.79	12.29 ± 4.16	-6.823	< 0.001[a]
ADAS13	14.48 ± 5.50	20.01 ± 5.79	-7.795	< 0.001[a]
ASASQ4	4.76 ± 2.19	6.77 ± 2.21	-7.339	< 0.001[a]
RAVLT.immediate	36.21 ± 10.10	29.10 ± 7.98	6.021	< 0.001[a]
RAVLT.learning	4.19 ± 2.47	2.91 ± 2.26	4.231	< 0.001[a]
RAVLT.forgetting	4.31 ± 2.59	4.47 ± 2.15	-1.501	0.135[a]
RAVLT.perc.forgeting	51.55 ± 31.04	72.85 ± 30.45	-5.464	< 0.001[a]
LEDLTOTAL	4.96 ± 2.36	3.41 ± 2.66	4.931	< 0.001[a]
DIGTSCOR	40.75 ± 11.09	36.72 ± 10.96	2.883	< 0.005[a]
TRABSCOR	109.43 ± 62.94	132.09 ± 71.36	-2.704	0.007[a]
FAQ	1.50 ± 2.99	4.96 ± 4.79	-7.243	< 0.001[a]
mPACCdigit	−5.376 ± 2.96	−8.06 ± 2.96	7.174	< 0.001[a]
mPACCtrailsB	−5.47 ± 3.06	−8.22 ± 2.98	7.174	< 0.001[a]

Table only for Group One where has 265 patients and 36 months follow-up time. Values are shown as mean ± standard deviation or percentage. Test statistics and P-values for differences between MCI-S and MCI-C are based on (a) t-test or (b) chi- square test. MCI-S = non-progressive MCI; MCI-P = progressive MCI; APOE = apolipoprotein E; MMSE = Mini-Mental State Examination. RAVLT = The Rey Auditory Verbal Learning Test (immediate: sum of 5 trails; learning: trial 5-trial 1; Forgetting: trial 5-delayed; perc.forgetting: Precent forgetting); DIGT = The Digit- Symbol Coding test; TRAB = Trail Making tests; CDRSB = Clinical Dementia Rating Scaled Response; FAQ = Activities of Daily living Score; ADAS = Alzheimer's Disease Assessment Scale–Cognitive sub- scale; mPACCdigit = the Digit Symbol Substitution Test from the Preclinical Alzheimer Cognitive Composite;

2.3.3 MRI data

Structural MRI data were collected according to the ADNI acquisition protocol using T1-weighted scans (GradWarp, B1 Correction, N3, Scaled) [36]. These data included baseline structural MRI scans of 840 ADNI participants, including 230 diagnosed as cognitively normal, 200 with diagnoses of dementia, and 410 diagnosed with MCI. Processing for ROI-based volumetric data used in the present study included brain extraction [34] and a multi-atlas, consensus-based label fusion scheme for anatomical parcellation [35] to generate template-based ROIs deformed to individual subject space. MRI scans were automatically segmented into 145 anatomic regions of interest (ROIs) spanning the entire brain. An additional 114 derived ROIs were calculated by combining single ROIs within a tree hierarchy, to obtain volumetric measurements from larger structures [36]. In total, 259 ROIs were measured and used as potential predictors of MCI-to-dementia progression in this study.

One of the goals of this study is to investigate if manually selecting predictors improves a model's performance. Based on the extant literature [68], we manually selected 26 out of 259 features as theoretically significant predictors of MCI to dementia progression (Table 2.3) [18, 43, 123, 91, 42, 108]. While many brain regions have been reported as showing some relationship to MCI-to-dementia progression, prior reports and reviews clearly implicate hippocampal and entorhinal cortical volumes as markers of such conversion. In addition, we manually selected additional regions based on their common occurrence across reports, including cingulate gyrus, precuneus, amygdala, inferior frontal gyrus, superior parietal lobule, and lobar white matter volumes.

2.4 Method and Algorithm

In the following section, we utilize binary LR and SVM classification techniques to investigate which approach yields superior discrimination accuracy in the context of ADNI data. Prior comparisons of logistic regression and SVM have reported that SVM requires fewer variables than logistic regression to achieve an equivalent level of misclassification rate (MCR) [131, 30]. These also report SVM performs better than LR with microarray expression data [30]. Furthermore,

Table 2.3: Pre-selected MRI Features of Group One

Characteristics	MCI-S	MCI-C	Test statistic	P-value
HippoR	3684 ± 438	3366 ± 437	5.735	< 0.001
HippoL	3414 ± 418	3105 ± 388	5.994	< 0.001
flWMR	96720 ± 6218	96976 ± 5585	-0.338	0.73
flWML	93671 ± 5836	94238 ± 5160	-0.802	0.42
plWMR	47197 ± 3415	47141 ± 3098	0.135	0.89
plWML	50149 ± 3714	50038 ± 3467	0.242	0.81
tlWMR	56076 ± 3252	55934 ± 2931	0.359	0.72
tlWML	55412 ± 3396	55468 ± 3023	-0.136	0.89
ACgCR	3167 ± 756	3128 ± 641	0.438	0.66
ACgCL	4104 ± 787	4075 ± 689	0.312	0.76
EntR	2189 ± 365	1983 ± 373	4.412	< 0.001
EntL	2050 ± 399	1844 ± 356	4.240	< 0.001
MCgCR	4176 ± 547	4200 ± 541	-0.341	0.73
MCgCL	3988 ± 493	4002 ± 559	-0.213	0.83
MFCR	1581 ± 342	1505 ± 524	1.805	0.07
MFCL	1566 ± 285	1548 ± 291	0.487	0.62
OpIFGR	2575 ± 608	2425 ± 546	2.021	0.04
OpIFGL	2465 ± 550	2361 ± 579	1.466	0.14
OrIFGR	1252 ± 315	1196 ± 362	1.322	0.18
OrIFGL	1514 ± 335	1398 ± 356	2.658	< 0.001
PCgCR	3679 ± 466	3528 ± 415	2.657	< 0.001
PCgCL	3991 ± 442	3789 ± 424	3.676	< 0.001
PCuR	10129 ± 1193	9862 ± 1313	1.701	0.09
PCuL	10005 ± 1263	9759 ± 1299	1.522	0.13
SPLR	8867 ± 1140	8693 ± 1219	1.180	0.02
SPLL	8880 ± 1192	8662 ± 1313	1.390	0.17

Values are shown as mean ± standard deviation or percentage. Test statistics and P-values for differences between MCI-C and MCI-S are based on t-test. MCI-S = non-progressive MCI; MCI-C = progressive MCI. HippoR = Right Hippocampus; HippoL = Left Hippocampus; flWMR = frontal lobe WM right; flWML = frontal lobe WM left; plWMR = parietal lobe WM right; plWML = parietal lobe WM left; tlWMR = temporal lobe WM right; tlWML = temporal lobe WM left; ACgCR=Right ACgG anterior cingulate gyrus; ACgCL=Left ACgG anterior cingulate gyrus; EntR = Right Ent entorhinal area; EntL = Left Ent entorhinal area; MCgCR = Right MCgG middle cingulate gyrus; MCgCL = Left MCgG middle cingulate gyrus; MFCR = Right MFC medial frontal cortex; MFCL = Left MFC medial frontal cortex; OpIFGR = Right OpIFG opercular part of the inferior frontal gyrus; OpIFGL = Left OpIFG opercular part of the inferior frontal gyrus; OrIFGR = Right OrIFG orbital part of the inferior frontal gyrus; OrIFGL = Left OrIFG orbital part of the inferior frontal gyrus; PCgCR = Right PCgG posterior cingulate gyrus; PCgCL = Left PCgG posterior cingulate gyrus; PCuR = Right PCu precuneus; PCuL = Left PCu precuneus; SPLR = Right SPL superior parietal lobule; SPLL = Left SPL superior parietal lobule.

SVMs have a nice dual form, giving sparse solutions when using the kernel trick. In addition, both methods involve minimizing some cost associated with the misclassification based on likelihood ratio for a probabilistic model. Therefore, LR and SVM share common roots in statistical pattern recognition, which we utilize in the comparison of their performance on multi-modal ADNI data.

2.4.1 Logistic Regression

Logistic regression (LR) is the most commonly used machine learning approach for binary classification. In the past decade this has been applied to task of MCI-to-dementia conversion [29, 144, 82]. In the present study, we consider a supervised learning task where we are given M training examples $\{D = (x_i, y_i), i = 1, ...M\}$. Here each $x_i \in \Re^N$ is N dimensional feature vectors, and $y_i \in \{0, 1\}$ is a class label. The goal of LR is to model the probability p of a random variable $\boldsymbol{y}$ being 1 or 0 given the experimental data $\boldsymbol{x}$. The logistic regression model is defined as follows:

$$logit \;\; p = log\frac{p}{1-p} \tag{2.1}$$

Logit, the natural logarithm of the odds, is the key concept that underlies logistic regression. The equation for LR is:

$$log\frac{P(y_i = 1|x_i; \boldsymbol{\beta})}{1 - P(y_i = 1|x_i; \boldsymbol{\beta})} = \sum_{j=1}^{N} \boldsymbol{\beta_j x_{ij}} \tag{2.2}$$

where $\boldsymbol{\beta} = (\beta_1, ...\beta_N)^T$ are the parameters or weights of the logistic regression model, $\boldsymbol{x_{ij}} = (x_{i1}, ...x_{iN}), i = 1, ...M$. Also, $P(y_i = 1|x_i, \boldsymbol{\beta})$ is the probability that ith MCI patient will develop dementia and $P(y_i = 0|x_i, \boldsymbol{\beta})$ is the probability that ith MCI patient will not develop dementia. Denote $P(y_i = 1|x_i; \boldsymbol{\beta}) = h(x_i)$, then

$$h(x_i) = \frac{1}{1 + exp(\sum_{j=1}^{N} -\boldsymbol{\beta_j} x_{ij})} \tag{2.3}$$

LR is usually trained by minimizing an error function; an appropriate choice of such a function for binary classification problems is the cross-entropy error:

$$e_i(\boldsymbol{\beta}) = -y_i log(h(x_i)) - (1 - y_i) log(1 - h(x_i))) \tag{2.4}$$

The total cost over the data $\{D = (x^i, y^i), i = 1, ...M\}$ is:

$$J(\boldsymbol{\beta}) = -\frac{1}{M}[\sum_{i=1}^{M} y_i log(h(x_i)) - (1 - y_i) log(1 - h(x_i))] \tag{2.5}$$

Consider the problem of finding the maximum likelihood estimate (MLE) of the parameters $\boldsymbol{\beta}$ for the unregularized logistic regression model. To find the optimized weights $\boldsymbol{\beta}$, the total cost needs to be minimized. The optimization function can be written:

$$\boldsymbol{\beta}^{optimal} = min_{\boldsymbol{\beta}} - \frac{1}{M}[\sum_{i=1}^{M} y_i log(h(x_i)) - (1 - y_i) log(1 - h(x_i))] \tag{2.6}$$

Solving Eq. (2.6) yields the optimal weights of $\boldsymbol{\beta}$. However, the model-building challenge is to abstract the underlying distribution from the particular instance D of samples because of the relatively small sample size, as compared to the number of features. The problem of replicating the data set instead of identifying the underlying distribution is known as overfitting [37]. To avoid the overfitting problem, it is often necessary to apply a dimension reduction technique. L_1 and L_2 norm are widely used to avoid overfitting, especially when there is a only small number of training examples, or when there is a larger number of features to be learned. L_1 norm or $lasso$ is also often used for feature selection, and has been shown to generalize well in the presence of many irrelevant features [76, 109]. L_1 regularization is implemented by adding L_1 norm to the cost function; the cost function and the optimization function were based on the following:

$$J(\boldsymbol{\beta}) = -\frac{1}{M}[\sum_{i=1}^{M} y_i log(h(x_i)) - (1 - y_i) log(1 - h(x_i))] + \lambda|\boldsymbol{\beta}| \tag{2.7}$$

and

$$\boldsymbol{\beta}^{optimal} = min_{\boldsymbol{\beta}}\{-\frac{1}{M}[\sum_{i=1}^{M} y_i log(h(x_i)) - (1 - y_i) log(1 - h(x_i))] + \lambda|\boldsymbol{\beta}|\} \tag{2.8}$$

where λ is positive tuning parameter. This Eq. (2.8) is refereed to as L_1 regularized logistic regression.

2.4.2 Support Vector machine

Support Vector Machine (SVM) is another classification and regression method that can handle high-dimensional feature vectors. Algorithmically, SVMs build optimal boundaries between data sets by solving a constrained quadratic optimization problem [23, 112, 129, 128, 127]. The number of studies applying SVM to evaluate classification of conversion from MCI to dementia has grown over the past decade [147, 148, 24, 145, 56, 68, 31, 59, 28, 136].

We briefly review basic support vector machines with linear kernal (SVM-linear) for classification problems: Let $\boldsymbol{\beta}^T h(x) + \beta_0 = 0$ denote an equidistant hyperplane (decision surface) to the closest point of each class on the new space. The goal of SVMs is to find $\boldsymbol{\beta}$ and β_0 such that $|\boldsymbol{\beta}^T h(x) + \beta_0| = 1$ for all points closer to the hyperplane. In the following classifier construction, one assumes that:

$$\boldsymbol{\beta}^T h(x_i) + \beta_0 = \begin{cases} \geq 1 \; if \; y_i = 1 \\ \leq -1 \; if \; y_i = 0 \end{cases} \tag{2.9}$$

such that the distance from the closest point of each class to the hyperplane is $1/||\boldsymbol{\beta}||$ and the distance between the two groups is $2/||\boldsymbol{\beta}||$. To maximize the margin, the SVM requires the solution of the following optimization primal problem [151]:

$$min_{\boldsymbol{\beta},\boldsymbol{\beta}_0} \sum_{i=1}^{M}\{1 - y_i[\beta_0 + \sum_{j=1}^{N} \beta_j^T h_j(x_{ij})]\} \tag{2.10}$$

where h_j is the kernel function which is a linear function for SVM-linear. Specifically we choose, $h_j(x_j) = x_j$ for j-th covariate.

To make the algorithm work for highly correlated features and improve the fitted model's prediction accuracy, we reformulate our optimization by adding L_1-norm of β, i.e. the $lasso$ penalty as follows:

$$min_{\boldsymbol{\beta},\boldsymbol{\beta}_0} \sum_{i=1}^{M}\{1 - y_i[\beta_0 + \sum_{j=1}^{N} \beta_j^T h_j(x_{ij})]\} + \lambda||\beta||_1 \tag{2.11}$$

where λ is the tuning parameter that controls the trade-off between loss and penalty. The lasso penalty shrinks the fitted coefficients β towards zero, and hence benefits from the reduction in fitted coefficients' variance.

2.4.3 Experimental Design

We built four different classifiers, each designed to classify individual ADNI participants as belonging to either the MCI-C group or the MCI-S group: Classifier 1 is logistic regression (C-LR);

Classifier 2 is logistic regression with L_1 norm (C-LR-1); Classifier 3 is support vector machine (C-SVM), and Classifier 4 is SVM with L_1 norm (C-SVM-1). To test the classifiers' performance, we constructed five different data sources (Table 2.4). The first three single-modality data sets included clinical cognitive assessment scores and APOE4 status (CCA), all MRI volumes (ROI-NP), and MRI volumes with pre-selection (ROI-P), respectively. Two additional multi-modal data sets were constructed by combining the CCA data separately with ROI-NP and ROI-P data sets (i.e., brain volumes with and without pre-selection). Furthermore, it is interesting to note that the number of MCI-S subjects is 101 (38%) in the Group One and 122 (39%) in Group Two, which makes the data rather imbalanced. Consequently, to precisely report the results obtained from the models, the present study also assessed additional model performance parameters, including AUC score, sensitivity and specificity (accuracy coefficient is unreliable for imbalanced data). The prediction procedure consisted of three processing stages for Group One (Time=36 months) and Group Two (Time=24 months): 1) Split data as training, validation, testing set; 2) Train classifiers using training set, tune hyper-parameter using the validation set, and assess classifiers using testing set, then train classifiers again using L_1 norm on the same training set; 3) Report the testing accuracy, AUC score, sensitivity and specificity of each classifier on single-modality data. Specifically, the first stage used 80% of the sample as a training set while the remaining 20% of the data constituted the testing set. In the second stage, the optimal subsets of features of each data source are determined and chosen following application of L_1 norm. We then list the top 10 features of each data set for each of the models. In the last stage, we report AUC score, sensitivity (percent of MCI-C subjects correctly classified), and specificity (percent of MCI-S subjects correctly classified) as measures of classification accuracy. To protect against over-fitting and to avoid optimistically-biased estimates of model performance, we report 20 measures of predictive performance for each classifier (1-4); for these different partitions of the data, we report the mean and standard deviation of testing accuracy, AUC score, sensitivity, and specificity (Tables 6 & 7). We also investigate the relationship between the number of features and model performance. Finally, we compare the performance of LR with SVM based on their ability to handle the problem with a large number of covariates. Figure 2.3

illustrates the diagram of the prediction framework.

Table 2.4: Modalities

Data sources	# features
Single-modality	
Clinical Cognitive Assessments score and APOE4 data (CCA)	19
ROI with no pre-selection data (ROI-NP)	259
ROI with pre-selection data (ROI-P)	26
Multi-modal	
CCA and ROI with no pre-selection data (CCAR-NP)	278
CCA and ROI with pre-selection data (CCAR-P)	45

2.5 Results and Analysis

Cross-validation and choice of λ

We adopted 10-fold cross-validation to tune the hyper-parameters for each model, which included dividing the data into separate sets for training and validation. The ratio of case in training and validation was 8:2. Here, the training set was used to train the model and the validation set was used to select the hyper-parameters. The results of a 10-fold cross-validation run are summarized with the mean and standard deviation of the model skill scores based on testing data. Cross-validation was also applied to tune the hyper-parameters; λ is used to denote the hyper-parameters for both LR-L_1 and SVM-L_1. To select the optimized λ, we tried different values of the λ; results reported here include values of $\lambda = 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ and applied them to the Eq (2.8) and (2.11). Next, we selected the λ value based on the best cross-validation score and used the selected λ with Classifiers 2 and 4 to select optimal features. For brevity, the model performance estimates are reported in Tables 2.6 and 2.7 for each different modalities, and the top 10 selected features are reported in Table 2.5. For example, the best λ for ROI-NP-L_1 was 0.01 and the top 3 optimal features selected by LR were left amygdala, right accumbens area, and right middle

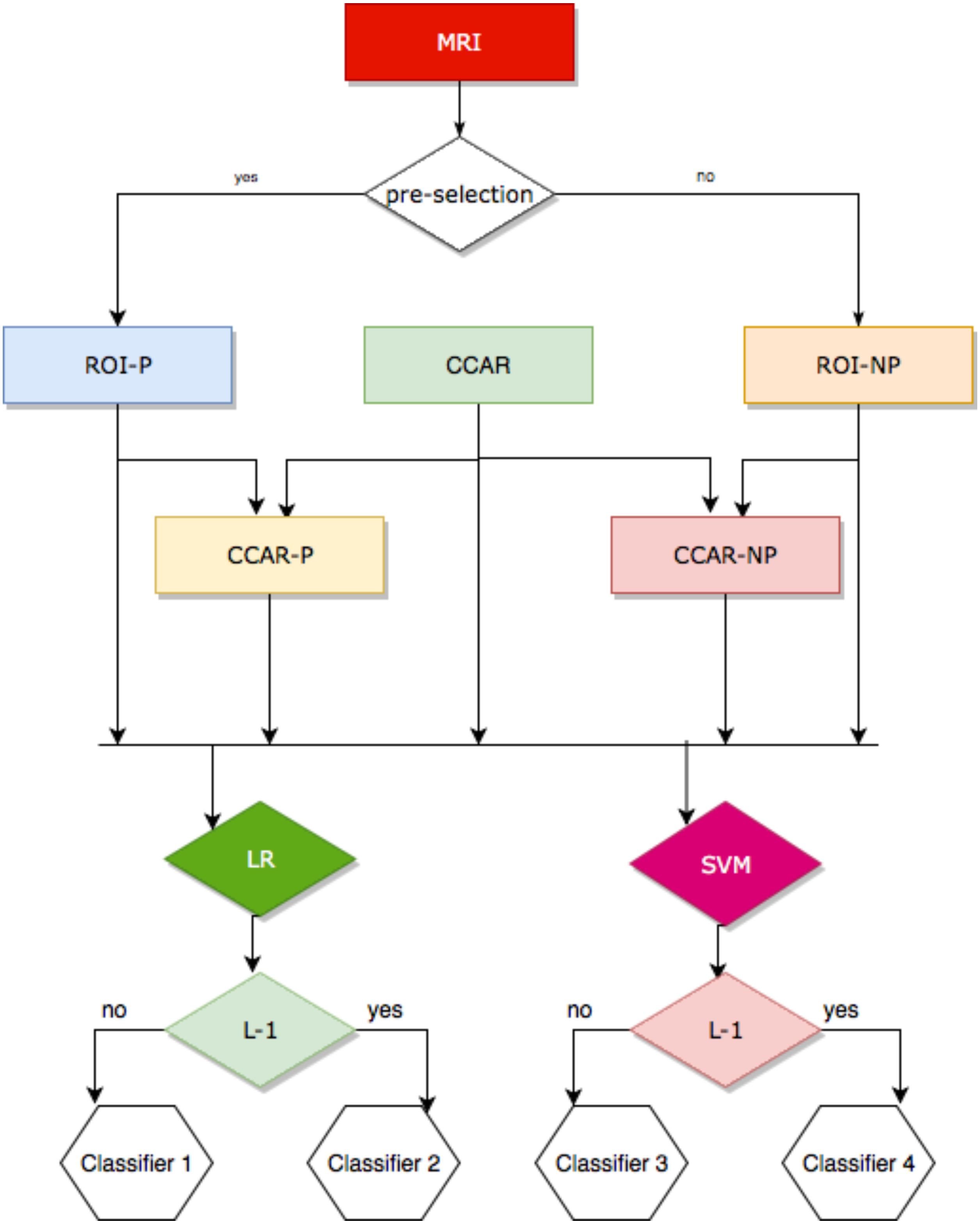


Figure 2.3: Flowchart of the LR and SVM method A) ROI-P: ROI level data with Pre-selection; B) ROI-NP: ROI level data with No Pre-selection; C) CCAR: Clinical, Cognitive assessments score, APOE4 and ROI level data.

temporal gyrus. After hyper-parameters were selected, we adopted a 10-fold cross-validation again to avoid optimistically-biased estimates of model performance. In each iteration, 212 of the 265 participants are selected by simple random sampling as training cases and the remaining 53 were used as test cases. The approximate 4:1 ratio of training to test cases is, of course, arbitrary.

Table 2.5: Top 10 features of Group One obtained by L_1 regularization

Source	LR-L1 (Classifier 2)			SVM-L1 (Classifier 4)		
Data	CCA	ROI-NP	CCAR-NP	CCA	ROI-NP	CCAR-NP
1	FAQ	AmyL	FAQ	FAQ	AmyL	FAQ
2	mPACCtrailsB	AccmR	AmyL	Yrs. Educ.	AccmR	AmyL
3	APOE4	MTGR	ADASQ4	APOE4	AOrGL	AccmR
4	ADASQ4	HippoL	HippoL	mPACCdigit	PCgGL	AOrGL
5	Learning	AOrGL	MTGR	ADASQ4	HippoL	PTR
6	Yrs. Educ.	PrGR	APOE4	Learning	PrGR	AnGR
7	Forgetting	PCgGL	AOrGL	ADAS11	POrGR	APOE4
8	mPACCdigit	InfR	Learning	mPACCtrailsB	PTR	PCgGL
9	ADAS13	POR	mPACCtrailsB	DELTOTAL	LOrGL	Learning
10	ADAS11	MOGL	mPACCdigit	Forgetting	MOrGL	POrGR

AccmR = Right Accumbens Area; AmyL = Left Amygdala; HippoL = Left Hippocampus; InfR = Right Inf Lat Vent; AOrGL = Left anterior orbital gyrus; AnGR = Left angular gyrus; LOrGL = Left lateral orbital gyrus; MOGL = Left middle occipital gyrus; MOrGL = Left medial orbital gyrus; MTGR = Right middle temporal gyrus; PCgGL = Left posterior cingulate gyrus; POR = Right parietal operculum; POrGR = Right posterior orbital gyrus; PrGR = Right precentral gyrus; PTR = Right planum temporale

2.5.1 Comparison with different modalities

We compared the performance of each classifier (1-4) on the five different feature sets (Table 2.4) based on estimates of AUC, sensitivity and specificity. As shown in Table 2.6, the results of using LR with L_1 regularization (Classifier 2) can achieve the high AUC of 81.2% and sensitivity of 81.4% on single-modality data (CCA), which is considerably better than performance of LR on the other four modalities. Similarly, the best AUC and sensitivity achieved by SVM are 81.4% and 81.6% based on the combination of CCA and SVM-L1. Furthermore, we also found the highest accuracy achieved by both classifiers without applying regularization is based on the single-modality data (CCA); this indicated both classifiers perform best on single-modality data.

2.5.2 Comparison of Pre-selection and L_1 norm

We found that using prior knowledge to inform feature selection improves model performance and protects against over-fitting. As shown in Table 2.6, model performance (i.e., AUC) on ROI-P (64.3%) and CCAR-P (76.3%) outperformed ROI-NP (60.6%) and CCAR-NP (60.1%). However, the performance of Classifier 2 on the ROI-NP-L_1 and CCAR-NP-L_1 data sets had AUC score of 64.1% and 64.0%, while the ROI-P-L_1 and CCAR-P-L_1 had respective AUC scores of 64.3% and 77.9%; this suggests that user-guided pre-selection significantly improved model performance over L_1 norm. In addition, the SVM (Classifiers 3 & 4) had similar and comparable results with LR classifiers. First, as with the LR models, the observed AUC estimates for CCAR-P and ROI-P (69.2% and 64.1%, respectively), were superior to AUCs from the CCAR-NP (59.1%) and ROI-NP analyses (61.4%). Classifier 4 exhibited similar performance on the CCAR-P-L_1 as Classifier 2, with an AUC value of 79.6% – higher than the model for CCAR-NP-L_1 (74.0%). Therefore, manually selecting features improves model's performance whether L_1 norm is applied, or not. Second, these results show it is necessary and important to use pre-selection because both LR and SVM models on CCAR-P-L_1, with respective AUC estimates of 77.9% and 78.5%, exhibited superior performance over the models without such pre-selection (i.e., LR and SVM on CCAR-NP-L_1 had AUC estimates of 64.0% and 74.0%, respectively).

2.5.3 Comparison of Groups One and Two

In addition to the results from models of Group One (i.e., MCI-to-AD conversion over 36 months), we also evaluated the performance of Group Two (i.e., MCI-to-AD conversion over 24 months) in an effort to gain further insight regarding possible benefits of shorter or longer assessment periods on classification of the progression of MCI to dementia. Table 2.7 summarizes the predictive performance of LR and SVM for Group Two. Similarly, we also evaluated classifier performance for single- and multi-modality feature sets. The best result is obtained by using SVM-L_1 model (Classifier 4) on CCAR-P, and its corresponding AUC, Sn and Sp are 76.2%, 60.1% and 79.2%, which verifies the assumption that manually selecting techniques improves the model's performance again. However, it warrants mention that all classifiers' performance on the Group One data outperformed the same classifiers' performance on the same data sets in Group Two. For example, Classifier 2 of Group One on CCA achieved AUC and Sn values of 81.2% and 83.1%, which is considerably better than the same classifier of Group Two on CCA (i.e., 76.3% and 79.8%). Similarly, Classifier 3 for ROI-NP had an AUC of 61.4% for Group One and 56.6% for Group Two. The experimental results indicated superior model performance on data obtained using longer than using shorter follow-up periods. Given the uncertainty in conversion, a longer time window for assessment of cognitive and functional change clearly yields more accurate classification.

2.5.4 Comparison of LR and SVM

In addition to comparing classification between different time windows of assessment, we also compared performance differences between LR and SVM. The results, including models' ability to address the over-fitting problem of LR and SVM methods with different modalities are displayed in Table 2.6, 2.7 and Fig.2.4, and 2.5. First, it is worth noting that both LR and SVM do not work well if no L_1 penalization used, since Classifiers 2 and 4 outperform Classifiers 1 and 3 on the same data set. Second, it is worth noting that SVM has a better performance on MRI data when the L1 feature selection method is employed. Third, it was possible to obtain good performance accuracy using LR, which had equivalent model performance as SVM for "large p" data (ROI-P), as evidenced by respective AUC estimates for Classifiers 1 and 3 of 64.3% and 64.1%. Finally, as shown in Fig. 2.5 and 2.4, the SVM method is more stable and robust than LR to the large number of features when n is small. To summarize, the best performance of Group One was achieved by Classifier 4 (SVM with L_1 norm) when using multi-modal – i.e., CCAR-L1, had an AUC of 81.4%.

Table 2.6: LR and SVM performance of Group One (Time = 3 years) for models on single- and multi-modal feature sets

Source	LR (Classifier 1 and 2)				SVM (Classifier 3 and 4)				Features
Modality	Test Acc %	AUC %	Sp %	Sn %	Test Acc %	AUC %	Sp %	Sn %	# Features
CAA	74.3 ± 6.0	80.8 ± 7.0	62.3 ± 12.1	81.5 ± 6.2	72.4 ± 6.9	80.0 ± 7.3	53.6 ± 13.2	79.4 ± 7.7	$19^{(2)}$ $19^{(1)}$
ROI-NP	58.1 ± 7.0	60.6 ± 8.1	45.5 ± 13.4	65.3 ± 7.9	59.5 ± 7.3	61.4 ± 8.5	46.5 ± 11.9	67.3 ± 8.5	$259^{(2)}$ $259^{(1)}$
ROI-P	64.4 ± 6.5	64.3 ± 6.6	46.1 ± 10.4	75.0 ± 9.6	62.1 ± 5.9	64.1 ± 6.2	43.6 ± 9.5	78.4 ± 10.4	$26^{(2)}$ $26^{(1)}$
CCAR-NP	57.6 ± 7.2	60.1 ± 8.1	44.8 ± 12.9	65.1 ± 9.0	57.8 ± 6.8	59.1 ± 7.0	45.9 ± 10.4	65.1 ± 7.5	$278^{(2)}$ $278^{(1)}$
CCAR-P	72.7 ± 6.4	76.3 ± 6.5	60.5 ± 10.4	80.4 ± 8.2	66.9 ± 6.0	69.2 ± 6.4	53.6 ± 13.2	74.4 ± 10.5	$45^{(2)}$ $45^{(1)}$
CCA-L_1	74.9 ± 6.4	81.2 ± 6.7	61.3 ± 12.0	83.1 ± 6.6	74.2 ± 6.0	81.4 ± 6.9	61.6 ± 11.5	81.6 ± 5.9	$3^{(2)}$ $4^{(1)}$
ROI-NP-L_1	62.2 ± 6.6	64.1 ± 7.9	53.1 ± 13.1	68.1 ± 7.2	62.7 ± 5.8	67.0 ± 6.7	53.7 ± 11.6	67.7 ± 7.4	$27^{(2)}$ $29^{(1)}$
ROI-P-L_1	64.4 ± 6.5	64.3 ± 6.2	46.2 ± 11.0	74.9 ± 9.6	64.4 ± 5.7	64.7 ± 5.8	46.7 ± 11.1	75.4 ± 8.3	$17^{(2)}$ $5^{(1)}$
CCAR-NP-L_1	62.6 ± 7.2	64.0 ± 8.2	51.8 ± 12.7	69.5 ± 7.3	67.4 ± 6.4	74.0 ± 7.4	55.7 ± 12.1	74.1 ± 7.1	$27^{(2)}$ $18^{(1)}$
CCAR-P-L_1	73.1 ± 6.5	77.9 ± 5.9	61.6 ± 10.9	79.6 ± 7.7	73.5 ± 6.2	78.5 ± 6.4	61.6 ± 9.3	80.8 ± 7.5	$25^{(2)}$ $14^{(1)}$

Predictive performance of LR and SVM (mean ± standard deviation) for all models. Performance estimates include testing accuracy (Test Acc %), area under the cureve (AUC), sensitivity (Sn), and specificity (Sp). The number (#) of features was determined via (1): Classifier 2; (2): Classifier 4.

Table 2.7: LR and SVM performance of Group Two (Time =2 years) for single-data and multimodal data

Source	LR (Classifier 1 and 2)				SVM (Classifier 3 and 4)				Features
Modality	Test Acc %	AUC %	Sp %	Sn %	Test Acc %	AUC %	Sp %	Sn %	# Features
CAA	69.9 ± 5.3	76.2 ± 5.5	56.7 ± 9.0	79.3 ± 7.3	69.4 ± 5.4	75.4 ± 5.5	56.7 ± 8.8	78.5 ± 7.1	$19^{(2)}$ $19^{(1)}$
ROI-NP	58.1 ± 4.2	58.8 ± 5.6	49.7 ± 7.1	64.4 ± 5.9	57.8 ± 5.0	56.6 ± 6.4	50.3 ± 7.1	62.9 ± 7.5	$259^{(2)}$ $259^{(1)}$
ROI-P	63.4 ± 4.7	65.8 ± 4.3	43.7 ± 10.2	77.8 ± 8.6	64.5 ± 4.7	66.2 ± 5.0	44.5 ± 8.5	79.1 ± 9.1	$25^{(2)}$ $25^{(1)}$
CCAR-NP	57.3 ± 4.0	58.8 ± 5.4	47.5 ± 8.3	64.3 ± 5.8	56.6 ± 5.5	56.4 ± 5.2	48.9 ± 7.9	62.3 ± 10.4	$278^{(2)}$ $278^{(1)}$
CCAR-P	70.2 ± 5.4	74.0 ± 5.0	56.7 ± 9.5	80.6 ± 7.0	69.5 ± 4.9	72.0 ± 5.3	58.1 ± 8.1	78.0 ± 8.2	$45^{(2)}$ $45^{(1)}$
CCA-L_1	70.1 ± 4.8	76.3 ± 5.3	56.8 ± 9.9	79.8 ± 7.6	70.4 ± 4.9	76.4 ± 7.7	56.8 ± 9.8	79.4 ± 7.7	$4^{(2)}$ $4^{(1)}$
ROI-NP-L_1	62.2 ± 6.0	64.7 ± 6.0	48.9 ± 9.2	72.0 ± 6.8	60.8 ± 4.5	65.9 ± 6.1	53.6 ± 7.5	64.3 ± 7.9	$31^{(2)}$ $29^{(1)}$
ROI-P-L_1	64.1 ± 4.6	66.8 ± 3.8	42.8 ± 11.3	79.8 ± 8.4	65.4 ± 4.0	67.8 ± 3.9	46.3 ± 9.4	81.1 ± 7.2	$14^{(2)}$ $6^{(1)}$
CCAR-NP-L_1	62.6 ± 6.3	64.8 ± 6.0	49.1 ± 9.1	72.1 ± 6.1	64.5 ± 5.1	71.7 ± 4.8	55.4 ± 7.8	71.4 ± 8.9	$32^{(2)}$ $26^{(1)}$
CCAR-P-L_1	70.0 ± 5.5	74.3 ± 5.5	57.8 ± 8.0	78.3 ± 8.8	71.3 ± 4.9	76.2 ± 4.7	60.1 ± 7.1	79.2 ± 8.5	$27^{(2)}$ $14^{(1)}$

For each modality, the predictive performance of LR and SVM are shown (mean ± standard deviation), including testing accuracy, AUC, sensitivity (Sn), specificity (Sp), # features is the number of features; # features is the number of features; this parameter was determined via (1): Classifier 2; (2): Classifier 4.

2.6 Discussion and Conclusion

In this thesis, we applied two machine learning methods under multiple conditions, to test accuracy in classifying patients with MCI who progress to clinically-defined dementia (MCI-C) from those who remain stable (MCI-S). Using multi-modal data from ADNI, we compared LR and SVM classification accuracy and pre-selection dimensional reduction techniques - i.e., feature selection as informed by prior findings in clinical neuroscience and by L_1 norm. Notably, the present results demonstrate important boundaries for applying feature selection techniques in statistical classification of MCI-to-dementia conversion. Specifically, we found that while using L_1 for pre-selection can improve accuracy, it also benefits from a more limited, theoretically based set of feature inputs. In addition, we found that model performance benefited from a longer window of assessment. These results have implications for studies utilizing multi-modal data for such classification, including features from clinical neuropsychological assessment, demographic and genetic markers, MRI-based volumetric brain measures, and other modalities.

Comparison of user-defined and L_1 pre-selection for LR and SVM classifiers yielded multiple noteworthy findings, consistent with previously published reports [147, 148, 24, 29, 145, 56, 144, 68]. First, the classification results showed that the model using multi-modal data with cognitive, clinical, and volumetric data (CCAR) achieved better classification accuracy than the methods based on single-modality (CCA, ROI). Moreover, the AUC of CCAR based on LR or SVM was either statistically significantly or at least numerically greater than those based on the single-modality model. Based in AUC, we reported the highest accuracy was observed for CCAR data at 78.5% by L_1 SVM and 77.9% by L_1 LR. Second, SVM demonstrated several advantages over LR in discriminating MCI-C from MCI-S (Fig. 2.4). For one, SVM performance tended to be more stable than LR when the number of features was relatively large. In other words, the model performance of SVM on ROI data remained more stable than LR when using larger numbers of features without user-defined pre-selection. In particular, SVM performance on ROI data improved as the number of features increased from 20 and 30. In contrast, the AUC values for ROI data sets remained fairly static despite increasing the number of features. However, LR model performance decreased gradually after the number of ROI features reached 40. Third, the classification results clearly demonstrate that manually selecting features on MRI data not only improved the model performance and protected the classifier from overfitting, but also affords easier interpretation of each selected feature's contribution to the model. In addition, we show that pre-selection improves performance: Tables 2.6 and 2.7 suggest it is the best strategy to obtain the maximum model performance, compared to features selection based on L_1 norm.

The present findings can also be interpreted in the context of other reports over the past decade that also investigated the prognostic capacity of brain volumetry data to predict the conversion of MCI to dementia, using either SVM or LR, and that also combined volumetry data with other imaging and biomarker modalities such as MRI, functional MRI (fMRI), positron emission tomography (PET) to cerebral spinal fluid (CSF) protein markers [147, 148, 24, 29, 145, 56, 144, 68, 78, 130, 69]. In addition, one can vary the degrees of non-linearity and flexibility in the model by employing different kernel functions. For example, Young et al (2013) report [145], results from both SVM and Gaussian process (GP) classification on MCI progression in ADNI data using MRI, PET, APOE4, and CSF biomarkers. In contrast the present study and with other published work that used MCI-C and MCI-S groups as training and test data sets, they trained a classifier to distinguish cognitively normal older adults from those diagnosed as probable AD. They reported that the accuracy using GP – an AUC value of 79.5% – was substantially higher than using any individual modality or using multi-kernel SVM. Other studies of MCI-to-dementia classification reporting high accuracy have also implemented other approaches such as multiple kernel learning (pMKL) classification techniques using clinical, MRI and plasma biomarkers data. One method using this approach to identify the important features first grouped the data set into five different data sources and then applied a filter-wrapper approach of feature selection techniques in combination with Joint Mutual Information (JMI) criterion to achieve an AUC of 82% [68].

We also found consistently superior classification performance in patients classified under a longer window of assessment. MCI-to-dementia conversion is a process that can take several years to reliably track an individual from onset of amnestic MCI to early-stage dementia [145, 92, 75].

For the modeled features to be of use for classification necessitates well-defined, if not orthogonal classes. However, MCI is not inherently prodromal to dementia: a large proportion of individuals with MCI never progress, either reverting to cognitively normal status or remaining rather stable. Furthermore, others may show early evidence of brain atrophy that precedes cognitive impairment by years. In order to account for this variable timing, others have employed methods such as supervised learning using time windows [102]; however, even those methods strongly benefit from longer follow-up periods. Thus, MCI is an inherently heterogeneous and poorly-defined class, particularly in terms of the relationships between brain characteristics and the likelihood and timing of further cognitive decline. Most recent computational neuroimaging studies in the past few years have utilized multi-modal features [24, 31, 82, 59, 28, 114, 89, 90, 133, 136]. For example, when Ding et al applied SVM with PET and MRI data to classify the transition from MCI to AD, they reported the sensitivity and specificity were 66.67% and 64.52% [31]. In addition to PET and structural MRI data, CSF protein markers can be used to predict progression from MCI to AD, in addition to proteomic, demographic and cognitive data [28, 113, 21]. By applying LR with L_1 norm to CSF markers for classifying individual patients as belonging to either the MCI-C and MCI-S group, one study reported a sensitivity and specificity of 80% and 75% [82]. Furthermore, Varatharajah and colleagues (2020) showed SVM-linear outperforms other advanced classification methods, including linear classifiers—multiple kernel learning (MKL) with linear kernels, SVM with a linear kernel, and generalized linear model (GLM), in predicting transition from MCI to AD [130]. In general, LR works well when the data is linearly separable and the number of data is greater than the number of features, whereas SVM with Gaussian Kernel is mostly used when the data is not linearly separable. In addition to LR and SVM, deep neural network approaches also offer benefits [78, 119], but have not had the extent of application in ADNI data as SVM and LR. Using a novel LR, artificial neural network (ANN) model and decision tree (DT) model for classifying the progression of MCI to AD, Kuang (2021) reported that the ANN exhibited the highest sensitivity at 82.1% [69].

In conclusion, models applying prior knowledge for classification and prediction of MCI-to-dementia conversion outperform those without pre-selection. This theoretically guided pre-selection of features from MRI-based regional brain volumes appears to protect the model against over-fitting. In addition, the present findings demonstrate that SVM classifier performance is more stable than LR for dealing with the “large p" problem. Clinical researchers should note the value of evaluating different classification and pre-selection approaches in application to clinical or research questions, and be mindful that not all machine learning techniques are equally beneficial for modeling specific clinical outcomes.

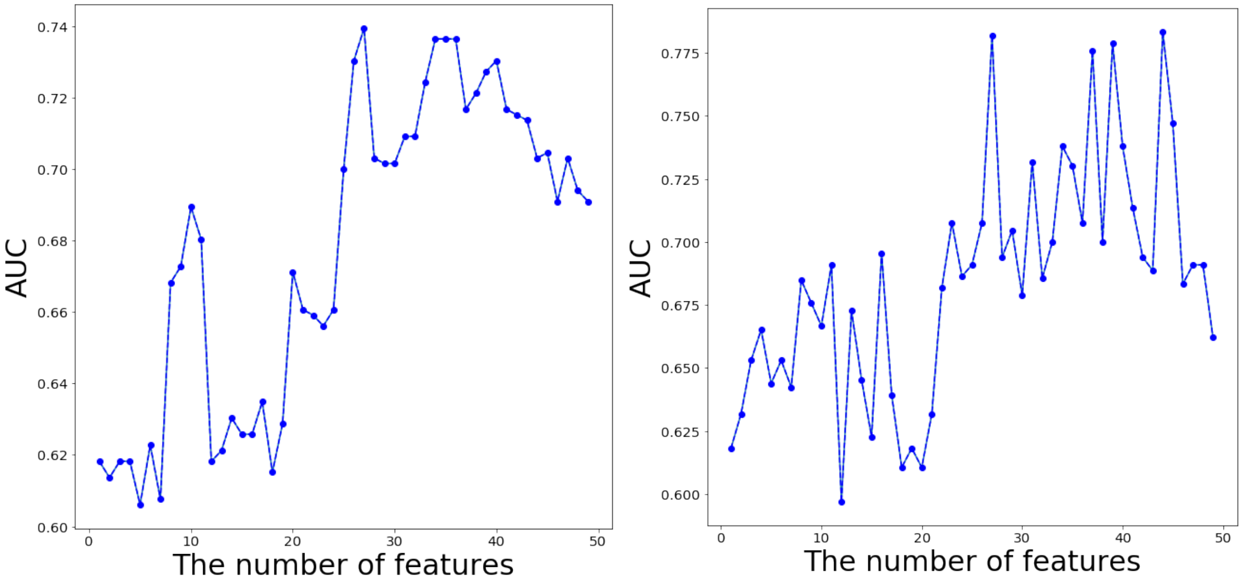


Figure 2.4: Model performance on ROI feature set by number of features for LR and SVM. Panel (a) shows dramatic growth in AUC with LR as the number of features increases from 1 to 30, and then becoming more static at approximately 74% - i.e., as the number of features increases from 30 to 40, but drops significantly when the number of features reaches to 41. Panel (b) shows the AUC increased dramatically as the number of features grows from 1 to 28, but fluctuated after 29. The optimal number of ROI features for both methods are 29 and 28, and their corresponding optimized AUC were approximately 74.0% and 78.0%.

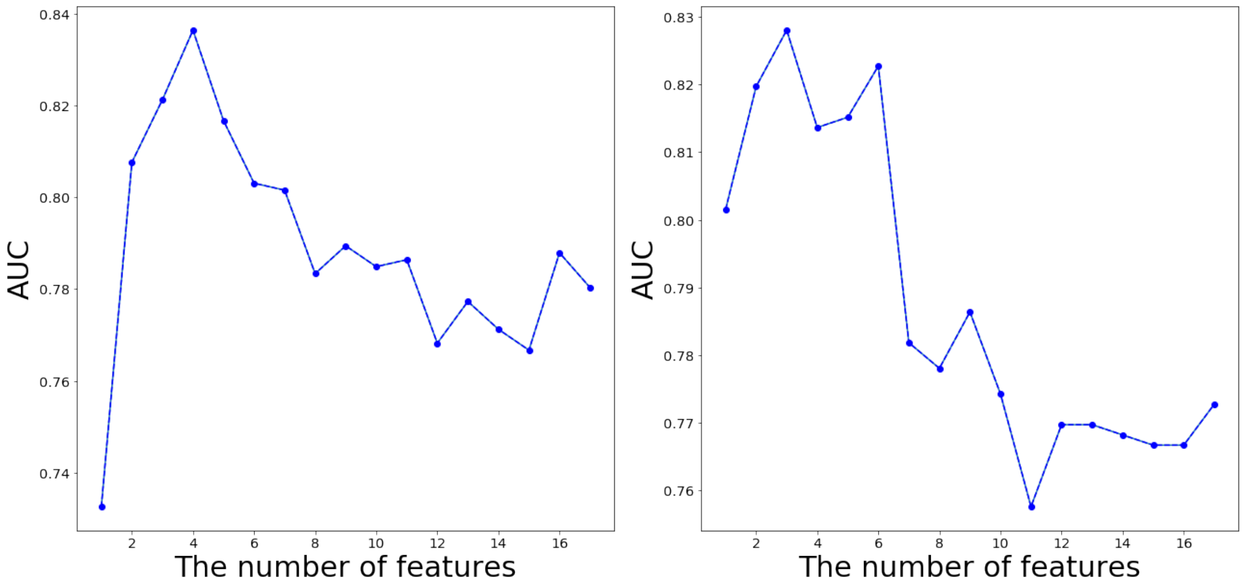


Figure 2.5: Model performance on CCA feature set by number of features for LR and SVM. Figure (a) shows there is a significant increase in the AUC with LR as the number of features increases from 1 to 5, then there is a slight decrease in the testing accuracy when the number of features is greater than 5. Figure (b) shows the AUC shot up dramatically as the number of features increases from 1 to 4. The optimal number of CCA features obtained by LR and SVM are 5 and 4, and their corresponding optimized AUC are approximately 84.0% and 83.0%.

CHAPTER 3

CONSISTENT VARIATIONAL BAYES CLASSIFICATION WITH DEEP NEURAL NETWORKS

3.1 Introdution

Bayesian deep neural network (BDNN) models are ubiquitous in classification problems; however, their Markov Chain Monte Carlo (MCMC) based implementation suffers from high computational cost, limiting the use of this powerful technique in large-scale studies. Variational Bayes (VB) has emerged as a competitive alternative to overcome some of these computational issues. This thesis focuses on the variational Bayesian deep neural network (VBDNN) estimation methodology and discusses the related statistical theory and algorithmic implementations in the context of classification. For a deep neural network-based classification, the thesis compares and contrasts the true posterior's consistency and contraction rates and the corresponding variational posterior. Based on the complexity of the deep neural network (DNN), this thesis provides an assessment of the loss in classification accuracy due to VB's use and guidelines on the characterization of the prior distributions and the variational family. The difficulty of the numerical optimization for obtaining the variational Bayes solution has also been quantified as a function of the complexity of the DNN. The development is motivated by an important biomedical engineering application, namely building predictive tools for the transition from mild cognitive impairment to Alzheimer's disease. The predictors are multi-modal and may involve complex interactive relations.

3.2 The Neural Networks Classifier and Likelihoods

Let Y be a binary random variable taking values 0 or 1, representing the class levels and $X \in \mathbb{R}^p$ is a feature vector drawn from a feature space with some marginal distribution P_X. We consider the following binary classification problem

$$P(Y = 1|X = x) = \sigma(\eta_0(\boldsymbol{x})),\ P(Y = 0|X = x) = 1 - \sigma(\eta_0(\boldsymbol{x})) \tag{3.1}$$

where $\eta_0(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}$ is some continuous function and $\sigma(.) = e^{(.)}/(1 + e^{(.)})$ is the sigmoid function. Thus, $P_{X,Y}$, the joint distribution of (X, Y) is a product of the conditional distribution in (4.1) and the marginal distribution P_X. Borrowing some notations from [14] and [141], a classifier C is a Borel measurable function $C : \mathbb{R}^p \rightarrow \{0, 1\}$, with the interpretation that we assign a point $\boldsymbol{x} \in \mathbb{R}^p$ to class $C(\boldsymbol{x})$. The test error of a classifier C is given by

$$R(C) = \int_{\mathbb{R}^p \times \{0,1\}} I_{\{C(X) \neq Y\}} dP_{X,Y} \tag{3.2}$$

Based on (4.1), we define the Bayes classifier as

$$C^{\text{Bayes}}(\boldsymbol{x}) = \begin{cases} 1, & \sigma(\eta_0(\boldsymbol{x})) \geq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

The Bayes classifier is optimal ([46]) since it minimizes the mis-classification error risk in (4.2). However, the Bayes classifier is not useful in practice, since the function $\eta_0(\boldsymbol{x})$ is unknown. Thus, a classifier is obtained based on a set of training observations $\{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, which are drawn from $P_{X,Y}$. A good classifier based on the sample should have the risk tending to the Bayes risk as the number of observations tends to infinity, without any requirement for its probability distribution. This is so called universal consistency. Multiple methods have been adopted to estimate $\eta_0(\boldsymbol{x})$, including logistic regression (a linear approximation), generalized additive model (GAM, a nonparametric nonlinear approximation), deep neural networks (a complicated structure which is dense in continuous functions) etc. The first two methods usually work in practice with good theoretical foundation, however, they may fail to catch the complicated dependency of the feature vector $\boldsymbol{x}$ in a wide range of applications including the problem that we consider in this article. On the other hand, the neural network structure which can exploit the dependency implicitly without any specific parametric structure, has relatively few theoretical works establishing its statistical efficacy in Bayesian models. In this thesis, we thereby focus our attention on classification using deep neural networks.

Consider a single layer neural network model with p predictor variables. The layer has k_n nodes, where k_n may be a diverging sequence depending on n. The validity of neural network approximations is based on the universal approximation results [25], which states that the single layer neural network is able to approximate any continuous function with a quite small approximation error when k_n is large. Assume a Fourier representation of $\eta_0(\boldsymbol{x})$ of the form

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^p} e^{i\boldsymbol{\omega}^T\boldsymbol{x}} \tilde{F}(d\boldsymbol{\omega})$$

and denote $\Gamma_{B,C} = \{f(\cdot) : \int_B \|\boldsymbol{\omega}\|_2 |\tilde{f}|(d\boldsymbol{\omega}) < C\}$ for some bounded subset B of $\mathbb{R}^p$ containing zero for some constant $C > 0$. Then, for all functions $\eta_0 \in \Gamma_{B,C}$, there exist a single layer neural network output $\eta(\boldsymbol{x})$ such that $\|\eta - \eta_0\|_2 = O(1/\sqrt{k_n})$ [5]. This result ensures good approximation property of single layer neural network, and the convergence rate depends only on the number of nodes under mild conditions on $\eta_0(\boldsymbol{x})$. [77] proved that as long as the activation function is not algebraic polynomials, the single layer neural network is dense in the continuous function space, thus can be used to approximate any given continuous function.

We use $\boldsymbol{\theta}_n$ to index the set of all the parameters. For $p_n \times 1$ input vector $\boldsymbol{x}$, consider a deep neural network with L_n hidden layers and $k_{1n}, \cdots, k_{L_n n}$ being the number of nodes in the hidden layers. Let $k_{0n} = p_n + 1$ and $k_{(L_n+1)n} = 1$. It can be checked that the total number of parameters is $K_n = \prod_{v=0}^{L_n} k_{(v+1)n}(k_{vn} + 1)$ due to the formulation below.

$$\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) \quad = \quad \boldsymbol{b}_L + \boldsymbol{A}_L\psi(\boldsymbol{b}_{L-1} + \boldsymbol{A}_{L-1}\psi(\boldsymbol{b}_{L-2} + \boldsymbol{A}_{L-2}\psi(\cdots\psi(\boldsymbol{b}_1 + \boldsymbol{A}_1\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x})))$$

$\boldsymbol{b}_v, v = 0, \cdots, L$ are vectors of dimension $k_{v+1} \times 1$ and $\boldsymbol{A}_v, v = 1, \cdots, L-1$ are matrices each of dimension $k_{v+1} \times k_v$. We have suppressed the dependence on n for notation simplicity.

For the purposes of this thesis, we use the activation function to be the sigmoid function, $\psi(x) = e^x/(1 + e^x)$, although the theoretical results are valid to a wider class of activation functions such as tan-hyperbolic, Gaussian etc.. Thus, using the neural network in (3.4) as an approximation to the true function $\eta_0(\boldsymbol{x})$ in (4.1), the conditional probabilities of Y given $X = \boldsymbol{x}$ is given by

$$P(Y = 1|X = \boldsymbol{x}) = \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})), \quad P(Y = 0|X = \boldsymbol{x}) = 1 - \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})) \tag{3.4}$$

Assuming Bernoulli distribution, the conditional density of $Y|X = \boldsymbol{x}$ under the model is:

$$\ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}) = \exp\left(y\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \log\left(1 + e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})}\right)\right) \tag{3.5}$$

Thus, the likelihood function for the data $(\boldsymbol{y}_n, \boldsymbol{X}_n) = (y_i, \boldsymbol{x}_i)_{i=1}^n$ under the model is

$$L(\boldsymbol{\theta}_n) = \prod_{i=1}^{n} \ell_{\boldsymbol{\theta}_n}(y_i, \boldsymbol{x}_i) = \exp\left(\sum_{i=1}^{n} \left[y_i \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}_i) - \log\left(1 + e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}_i)}\right)\right]\right) \tag{3.6}$$

In view of (4.1), the conditional density of $Y|X = \boldsymbol{x}$ under the truth

$$\ell_0(y, \boldsymbol{x}) = \exp\left(y\eta_0(\boldsymbol{x}) - \log\left(1 + e^{\eta_0(\boldsymbol{x})}\right)\right) \tag{3.7}$$

Therefore, the likelihood function for the data under the truth is given by

$$L_0 = \prod_{i=1}^{n} \ell_0(y_i, \boldsymbol{x}_i) = \exp\left(\sum_{i=1}^{n} \left[y_i \eta_0(\boldsymbol{x}_i) - \log\left(1 + e^{\eta_0(\boldsymbol{x}_i)}\right)\right]\right) \tag{3.8}$$

3.3 Bayesian Inference with Variational Algorithm

3.3.1 Prior Choice

For Bayesian analysis, prior distributions have to be assigned for all parameters defining the model. Although one may have a prior knowledge concerning the function represented by a neural network, it is generally difficult to translate this into a meaningful prior on neural network weights. We assume an independent normal prior as follows:

$$p(\boldsymbol{\theta}_n) = \prod_{j=1}^{K_n} \frac{1}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{1}{2\sigma_{jn}^2}(\theta_{jn} - \mu_{jn})^2} \tag{3.9}$$

(A0) For $\boldsymbol{\sigma}_n = [\sigma_{1n}, \cdots, \sigma_{K_n n}]$, $\boldsymbol{\sigma}_n^* = [1/\sigma_{1n}, \cdots, 1/\sigma_{K_n n}]$, assume

$$\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n) \quad ||\boldsymbol{\sigma}_n^*||_\infty = O(1),$$

where $||.||_\infty$ is the supremum norm of a vector as in definition A.0.1 in appendix B. Note, the above assumption ensures that the variance associated with each θ_{jn} do not grow at an arbitrarily large rate in which case the consistency of both the Bayesian and variational Bayes approach would break down. Restrictions on the mean parameter $\boldsymbol{\mu}_n = [\mu_{1n}, \cdots, \mu_{K_n n}]$ directly impact the consistency rate and are more case specific (see section 3.4 for a thorough discussion).

The reason for choosing the above form of prior is two folds: (1) first it guarantees that the true posterior distribution is consistent (2) second it guarantees, under a suitable choice of the variational family, the approximated variational posterior is also consistent. The choice of prior in (3.9) is not unique. Indeed, one can work with a much more generic class of priors such that (1) and (2) hold. Note, each prior comes with its own associated computation complexity, implementation and theoretical justification. We choose one which does a fairly good job under all these three criterion. In view of (3.6) and (3.9), posterior distribution of $\boldsymbol{\theta}_n$ given $\boldsymbol{y}_n = [y_1, \cdots, y_n]^\top$ and $\boldsymbol{X}_n = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^\top$ is

$$\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n) = \frac{\pi(\boldsymbol{\theta}_n, \boldsymbol{y}_n, \boldsymbol{X}_n)}{\pi(\boldsymbol{y}_n, \boldsymbol{X}_n)} = \frac{L(\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)}{\int L(\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n} \tag{3.10}$$

where $\pi(\boldsymbol{y}_n, \boldsymbol{X}_n)$ is free from the parameter and depends only on $\boldsymbol{y}_n$ and $\boldsymbol{X}_n$.

3.3.2 Variational Inference

As a first step to variational inference (VI) procedure, one has to start with a variational family. Given several options, we work with one which is simple, computationally and structurally tractable, and more importantly they provide statistically consistent posterior estimation. We posit a mean field Gaussian variational family of the form

$$\mathcal{Q}_n = \left\{ q(\boldsymbol{\theta}_n) : q(\boldsymbol{\theta}_n) = \prod_{k=1}^{K_n} \frac{1}{\sqrt{2\pi s_{jn}^2}} e^{-\frac{1}{2s_{jn}^2}(\theta_{jn}-m_{jn})^2} \right\} \tag{3.11}$$

Note that the variational family assumes that each θ_{jn} is independent with mean and standard deviation equal to m_{jn} and s_{jn} respectively.

The variational posterior aims to reduce the KL-distance between the variational family and the true posterior [9, 41, 11]. For the true posterior, $\pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)$ in (4.9), the variational posterior is

$$\pi^* = \underset{q \in \mathcal{Q}_n}{\text{argmin}} \; d_{\text{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)). \tag{3.12}$$

where d_{KL}, the Kullback-Leibler (KL) divergence between a variational family member $q(\boldsymbol{\theta_n})$ and the true posterior $\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)$ is given by

$$d_{\mathrm{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) = \int \log(q(\boldsymbol{\theta}_n)/\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n))q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \tag{3.13}$$

Bases on (4.9), simplifying further, we get

$$\begin{aligned} d_{\mathrm{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) &= \int [\log q(\boldsymbol{\theta}_n) - \log \pi(\boldsymbol{\theta}_n, \boldsymbol{y}_n, \boldsymbol{X}_n)]q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \log \pi(\boldsymbol{y}_n, \boldsymbol{X}_n) \\ &= -\mathrm{ELBO}(q, \pi(., \boldsymbol{y}_n, \boldsymbol{X}_n)) + \log \pi(\boldsymbol{y}_n, \boldsymbol{X}_n) \end{aligned} \tag{3.14}$$

Since the last term in (3.14) does not depend q, optimizing (3.14) w.r.t. to q boils down to optimizing the first term. Indeed the first term is nothing but the negative of the evidence lower bound (ELBO). Thus in order to minimize the KL-distance, we shall instead maximize the ELBO between q and $\pi(., \boldsymbol{y}_n, \boldsymbol{X}_n)$. Alternatively, we define π^* as

$$\pi^* = \underset{q \in \mathcal{Q}_n}{\operatorname{argmax}} \ \mathrm{ELBO}(q, \pi(., \boldsymbol{y}_n, \boldsymbol{X}_n)) \tag{3.15}$$

To maximize the ELBO in (3.14), let $\mathcal{V}_q = (m_{1n}, \cdots, m_{K_n n}, s_{1n}^2, \cdots, s_{K_n n}^2)$ where m_{jn} and s_{jn} is the mean and standard deviation of θ_{jn} under the density q. Thus, each $q \in \mathcal{Q}_n$ is indexed by its parameters. Consequently,

$$\begin{aligned} \mathrm{ELBO}(q(.|\mathcal{V}_q), \pi(., \boldsymbol{y}_n, \boldsymbol{X}_n)) &= \int [\log \pi(\boldsymbol{\theta}_n, \boldsymbol{y}_n, \boldsymbol{X}_n) - \log q(\boldsymbol{\theta}_n|\mathcal{V}_q)]q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n \\ &= \int \log L(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n + \int [\log p(\boldsymbol{\theta}_n) - \log q(\boldsymbol{\theta}_n|\mathcal{V}_q)]q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n \\ &= \int \log L(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n - d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.)) = \mathcal{L}_{\mathcal{V}_q} - d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.)) \end{aligned} \tag{3.16}$$

The derivative of $d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.))$ w.r.t. $\mathcal{V}_q$ has a closed form expression (see appendix A). The key challenge is the derivative $\mathcal{L}_{\mathcal{V}_q}$ w.r.t. to $\mathcal{V}_q$ which we discuss next

$$\begin{aligned} \nabla_{\mathcal{V}_q}\mathcal{L}_{\mathcal{V}_q} &= \nabla_{\mathcal{V}_q} \int \log L(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n = \int \log L(\boldsymbol{\theta}_n)\nabla_{\mathcal{V}_q}q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n \\ &= \int \nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n|\mathcal{V}_q) \log L(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|\mathcal{V}_q)d\boldsymbol{\theta}_n = E_{q(.|\mathcal{V}_q)}(\log q(\boldsymbol{\theta}_n|\mathcal{V}_q) \log L(\boldsymbol{\theta}_n)) \end{aligned} \tag{3.17}$$

where the last equality holds since $\nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n|\mathcal{V}_q) q(\boldsymbol{\theta}_n|\mathcal{V}_q) = \nabla_{\mathcal{V}_q} q(\boldsymbol{\theta}_n|\mathcal{V}_q)$.

The black-box variational inference (BBVI) algorithm, [107], optimizes the ELBO using gradient descent method by making use of a similar approach. The key challenge in evaluating the gradient in (3.17) is the computation of the expectation. Exact computation of the expectation leads to high computational complexity whereas using noisy estimates leads to high variability. In section 3.3.3, we elucidate how to ensure fast and efficient estimation of the gradient.

3.3.3 Black Box Variational Algorithm using score function estimator

The gradient in (3.17) is difficult to evaluate for problems with complex likelihood structures arising out of deep network models. Alternatively, the above expectation is evaluated by sampling from the variational distribution and forming the corresponding Monte Carlo estimates of the gradient. We next explain the computation of Monte Carlo estimate of the gradient in (3.17) by using ideas similar to [107, 124]. Let $\mathcal{V}_q$ denote the current value of the variational parameters. We generate W samples from the variational distribution $q(.|\mathcal{V}_q)$ and define the noisy but unbiased estimate of $\nabla_{\mathcal{V}_q} \mathcal{L}_{\mathcal{V}_q}$ as

$$\widehat{\nabla_{\mathcal{V}_q} \mathcal{L}_{\mathcal{V}_q}} = \frac{1}{W} \sum_{w=1}^{W} \nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n[w]|\mathcal{V}_q) \log L(\boldsymbol{\theta}_n[w]) \tag{3.18}$$

where $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ are samples generated from $q(.|\mathcal{V}_q)$. Similarly, a noisy but unbiased estimate of the $\mathcal{L}_{\mathcal{V}_q}$ is given by

$$\widehat{\mathcal{L}}_{\mathcal{V}_q} = \frac{1}{W} \sum_{w=1}^{W} \log L(\boldsymbol{\theta}_n[w]) \tag{3.19}$$

Algorithm 1 provides the pseudocode summarizing the overall algorithm for BBVI.

Algorithm 1 BBVI

```
1. Fix an initial value for variational family parameters $\mathcal{V}_q^1$.
2. Fix a step size sequence $\rho_t, t = 1, \cdots$.
3. Set $t = 1$.
4. Simulate $W$ samples $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ from $q(.|\mathcal{V}_q^t)$.
5. Compute $\widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}}$ as in (3.18)
6. Update
   $$\mathcal{V}_q^{t+1} = \mathcal{V}_q^t + \rho_t \left( \widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}} - \nabla_{\mathcal{V}_q^t} d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.)) \right) \qquad (3.20)$$
7. Set $t = t + 1$.
8. Repeat steps 4-7 until the convergence of ELBO using $\widehat{\mathcal{L}}_{\mathcal{V}_q^t}$ as in (3.19) and
   $$\text{ELBO} = \widehat{\mathcal{L}}_{\mathcal{V}_q^t} - d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.))$$
```

In the implementation of the above algorithm, one needs to compute $d_{KL}(q(.|\mathcal{V}_q), p(.))$, $\nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n|\mathcal{V}_q)$ and $\nabla_{\mathcal{V}_q} d_{KL}(q(.|\mathcal{V}_q), p(.))$ for the variational parameters $\mathcal{V}_q$. For the choice of p and q as in (3.9) and (3.11), the explicit expressions have been presented in appendix A.

For the variational parameters $(s_{1n}, \cdots, s_{K_n n})$, the updating rule in (3.20) may lead to negative estimates. However, one must guard against this since variance terms cannot be negative. Thus, to perform the optimization, we reparametrize the variance terms as $s_{jn} = \log(1+e^{r_{jn}})$, $j = 1, \cdots, K_n$ and update the quantities r_{rn} in each step instead of s_{jn}. By chain rule, for any function $g(\mathcal{V}_q)$,

$$\nabla_{r_{jn}} g(\mathcal{V}_q) = \frac{e^{r_{jn}}}{(1 + e^{r_{jn}})} \nabla_{\mathcal{V}_q} g(\mathcal{V}_q)|_{s_{jn}=\log(1+e^{r_{jn}})}$$

where second term is the derivative of $g(\mathcal{V}_q)$ w.r.t. s_{jn} evaluated at $s_{jn} = \log(1+e^{r_{jn}})$. The explicit expressions of derivatives w.r.t. r_{jn} have also been provided in appendix A.

3.3.4 Control Variate: Stabilizing the stochastic gradient

We can use algorithm 1 to maximize the ELBO, however a major drawback is that the noisy estimator of the gradient has high variance. There are two major techniques to reduce the variance of gradients. One of them is “Rao-Blackwellization", where the idea is to replace the noisy estimate of gradient with its conditional expectation w.r.t. a subset of the variables, [107]. This method is useful when the posterior distribution is separable across subsets of variables or while dealing with latent variables. A convoluted likelihood as in (3.6) is not separable across the components of $\boldsymbol{\theta}_n$ and there are no latent variables in our model. We thereby refrain from using the Rao-Blackwellization approach.

Algorithm 2 BBVI-CV

```
1. Fix an initial value for variational parameter $\mathcal{V}_q^1$.
2. Fix a step size sequence $\rho_t, t = 1, \cdots$.
3. Set $t = 1$.
4. Simulate $W$ samples $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ from $q(.|\mathcal{V}_q^t)$.
5. Compute $c^{\star t} = \text{cov}(\boldsymbol{u}_1^t, \boldsymbol{u}_2^t)/\text{var}(\boldsymbol{u}_2^t)$ where $\boldsymbol{u}_1^t$ and $\boldsymbol{u}_2^t$ are same as in (3.22).
6. Compute $\widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}}$ as in (3.21).
7. Update
       $$\mathcal{V}_q^{t+1} = \mathcal{V}_q^t + \rho_t \left( \widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}} - \nabla_{\mathcal{V}_q^t} d_{\text{KL}}(q(.|\mathcal{V}_q), p(.)) \right)$$
8. Set $t = t + 1$.
9. Repeat steps 4-7 until the convergence of ELBO using $\widehat{\mathcal{L}}_{\mathcal{V}_q^t}$ as in (3.19) and
       $$\text{ELBO} = \widehat{\mathcal{L}}_{\mathcal{V}_q^t} - d_{\text{KL}}(q(.|\mathcal{V}_q), p(.))$$
```

Another method which also gives an efficient technique for stabilizing the gradient is called control variate (CV) (see [110, 97, 107]). We use CV to reduce the variance of the MC approximations of the gradients. The key idea behind the variance reduction as proposed in [110] is to replace the target function, whose expectation is being approximated by Monte Carlo, with an auxiliary function that has the same expectation but a smaller variance. To reduce the variance of the function $\xi(\phi)$, one instead considers the function $\hat{\xi}(\phi) = \xi(\phi) - b\left(\varphi(\phi) - E_q(\varphi(\phi))\right)$ where $\varphi(\phi)$ is function with finite expectation and c is a scalar. Such a choice ensures $E_q(\hat{\xi}(\phi)) = E_q(\xi(\phi))$ and $\text{Var}_q(\hat{\xi}(\phi)) = \text{Var}_q(\xi(\phi)) + c^2\text{Var}_q(\varphi(\phi)) - 2c\text{Cov}_q(\xi(\phi), \varphi(\phi))$ which is minimized at

$c^\star = \mathrm{Cov}_q(\xi(\phi), \varphi(\phi))/\mathrm{Var}_q(\varphi(\phi))$. Thus, greater the correlation between ξ and φ, greater the variance reduction. Similar to [107], we use $\nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}|\mathcal{V}_q)$ as a choice for $\varphi(\phi)$. The stochastic approximation of the gradient in (3.17) is then modified as

$$\widehat{\nabla_{\mathcal{V}_q}\mathcal{L}_{\mathcal{V}_q}} = \frac{1}{W}\sum_{w=1}^{W} \nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n[w]|\mathcal{V}_q)[\log L(\boldsymbol{\theta}_n[w]) - c^\star] \tag{3.21}$$

It is impossible to obtain an exact expression for $c^\star$, one thus uses $\widehat{c^\star} = \mathrm{cov}(\boldsymbol{u}_1, \boldsymbol{u}_2)/\mathrm{var}(\boldsymbol{u}_2)$,

$$u_1[w] = \nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n[w]|\mathcal{V}_q) \log L(\boldsymbol{\theta}_n) \qquad u_2[w] = \nabla_{\mathcal{V}_q} \log q(\boldsymbol{\theta}_n[w]|\mathcal{V}_q) \tag{3.22}$$

The extension of algorithm 1 with variance reduction of MC approximations due to CV is annotated as BBVI-CV and summarized in algorithm 2.

Similar to the implementation of algorithm 1, for the implementation of algorithm 2, we use the reparametrization of $s_{jn} = \log(1 + e^{r_{jn}})$ as explained in section 3.3.3.

3.3.5 RMSprop Learning Rate: Stabilizing the learning rate.

Note that both BBVI and BBVI-CV algorithms as described in section 3.3.3 and 3.3.4 work fairly well with a fixed learning rate for a single layer network. However, their performance deteriorates significantly when the neural networks have two or more layers. This can be attributed to the fact that the gradients for the different parameters changes at significantly different rates. In order to overcome these issues, a wide class of adaptive learning rates have been explored in [117], [150], etc. for the frequentist optimization of parameters in deep neural networks. One such popular technique which performs well in practice, called the RMSprop, was introduced in [57] where the gradient is divided by a running average of its recent magnitude.

As described in both [57] and [51], let G_t denote the value of the current gradient, then define

$$R_t = 0.9R_{t-1} + 0.1G_t^2, t = 1, 2, \cdots$$

and the replace the learning ρ_t by the effective learning rate $\rho_t/(\sqrt{R_t} + \epsilon)$ for some small $\epsilon > 0$. Numerical studies show for one layer network, RMSprop leads to faster convergence and for multiple layer networks, convergence is not possible without an adaptive learning rate similar to RMSprop. One could also experiment with other adaptive learning rates like AdaGrad, AdaDelta, ADAM, etc. to serve the same purpose as that of RMSprop (see [88] and [38] for more details on other adaptive learning rates.)

The updated version of BBVI and BBVI-CV using RMSprop, renamed as BBVI-RMS and BBVI-CV-RMS, are summarized as algorithms 4 and 5 and provided in appendix A.

3.3.6 Classification using variational posterior

Define, $\hat{\eta}(\boldsymbol{x})$, the variational estimator of $\eta_0(\boldsymbol{x})$ as

$$\hat{\eta}(\boldsymbol{x}) = \sigma^{-1}\left(\int \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right) \tag{3.23}$$

where π^* is the variational posterior. Analgous to (4.3), the classifier based on $\hat{\eta}(\boldsymbol{x})$ is

$$\hat{C}(\boldsymbol{x}) = \begin{cases} 1, & \sigma(\hat{\eta}(\boldsymbol{x})) \geq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

Note, the formulation in (3.23) guarantees that we directly approximate the main quantity of interest, $\sigma(\eta_0(\boldsymbol{x}))$ as in (4.1) by its posterior mean, $\int \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n$, which is empirically estimated as

$$\hat{\eta}^W(\boldsymbol{x}) = \frac{1}{W}\sum_{w=1}^{W} \sigma(\eta_{\boldsymbol{\theta}_n[i]}(\boldsymbol{x})) \tag{3.25}$$

where $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ are multiple samples from the variational posterior π^*. Since generation of multiple samples from the variational posterior is much cheaper, the order of error between (3.23) and (3.25) is negligible.

3.4 Posterior and Classification Consistency

In this section, we establish that the Bayesian inference procedure proposed in section 3.3 enjoys theoretical guarantees in terms of consistency of the posterior estimation and classification. For a simple Gaussian mean field family as in (3.11), we establish that the variational posterior (3.12) is consistent under suitable assumptions on the prior parameters. We also discuss how the true function η_0 impacts the rate of consistency of the variational posterior. Finally, we present how the consistency rates of the variational posterior differ from those of the true posterior.

Let f_0 and $f_{\boldsymbol{\theta}_n}$ be the joint density of the observations $(y_i, \boldsymbol{x}_i)_{i=1}^n$ under the truth and the model respectively. Without loss of generality, we assume $X_i \sim U[0,1]^{p_n}$, which implies $f_0(\boldsymbol{x}) = 1$ and $f_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = 1$. This implies that the joint distribution of $(y_i, \boldsymbol{x}_i)_{i=1}^n$ depends only the conditional distribution of $Y|X = \boldsymbol{x}$. From (4.1) and (3.4) with $\ell_{\boldsymbol{\theta}_n}$ and ℓ_0 as in (B.5) and (3.7),

$$\begin{aligned} f_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}) &= f_{\boldsymbol{\theta}_n}(y|\boldsymbol{x}) f_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = \exp\left(y\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \log\left(1 + e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})}\right)\right) = \ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}) \\ f_0(y, \boldsymbol{x}) &= f_0(y|\boldsymbol{x}) f_0(\boldsymbol{x}) = \exp\left(y\eta_0(\boldsymbol{x}) - \log\left(1 + e^{\eta_0(\boldsymbol{x})}\right)\right) = \ell_0(y, \boldsymbol{x}) \quad (3.26) \end{aligned}$$

We next define the Hellinger neighborhood of the true density function $f_0 = \ell_0$ as

$$\mathcal{U}_\varepsilon = \{\boldsymbol{\theta}_n : d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) < \varepsilon\} \tag{3.27}$$

where the Hellinger distance, $d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{\theta}_n})$ is given by

$$d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) = \left(\frac{1}{2}\int_{\boldsymbol{x}\in[0,1]^{p_n}} \sum_{y\in\{0,1\}} \left(\sqrt{\ell_0(y, \boldsymbol{x})} - \sqrt{\ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x})}\right)^2 d\boldsymbol{x}\right)^{1/2}.$$

Also, the Kullback-Leibler (KL) neighborhood of the true density function $f_0 = \ell_0$ is

$$\mathcal{N}_\varepsilon = \{\boldsymbol{\theta}_n : d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) < \varepsilon\} \tag{3.28}$$

where the KL distance, $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n})$ is given by

$$d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) = \int_{\boldsymbol{x}\in[0,1]^{p_n}} \sum_{y\in\{0,1\}} \left(\log \frac{\ell_0(y, \boldsymbol{x})}{\ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x})} \ell_0(y, \boldsymbol{x})\right) d\boldsymbol{x}$$

Let P_0^n denote the true distribution of $(\boldsymbol{y}_n, \boldsymbol{X}_n) = (y_i, \boldsymbol{x}_i)_{i=1}^n$ under the true density ℓ_0.

3.4.1 Posterior consistency and its implication in practice

In the following two theorems, we establish the posterior consistency of π^* defined in (3.12). In this direction, we show that the variational posterior concentrates in $\varepsilon-$small Hellinger neighborhoods of the true density ℓ_0. In theorem 3.4.1, we establish this result for a fixed choice of the neighborhood distance ε. In theorem 3.4.2, we establish the same result for shrinking neighborhood sizes of the true function ℓ_0. For both these theorems, the total number of parameters K_n allows to grow at a rate of n^a for some $0 < a < 1$. Note, theorem 1 is a simple consistency result and holds due to the universal approximation properties of neural networks (see [60]) when the number of layers and input variables are fixed. This is an important result since it shows, irrespective of the function under study, BDNN's enjoy consistency properties if the number of input variables and the number of layers are fixed. Additionally, we also provide a characterization on the prior distribution such as that the rate of growth of L_2 norm of the prior mean parameter necessary to guarantee the consistency result in theorem 3.4.1 (see A2) and contraction results in theorem 3.4.2 (see A4).

The theorem 3.4.2 studies the contraction rate of the variational posterior, it is more restrictive in nature and requires additional assumptions on the approximating neural network solution to the true function η_0 (see assumption (A3) below). We next describe how our current theoretical development contrast to the recent works of [105] and [3]. Firstly, theorem 2 establishes the variational posterior contraction rates following the classical definition of contraction as in the theorems of 2.1 of [47] and theorem 2.1 of [122]. It differs from the consistency results in [3] which deal with posterior expectation of square of Hellinger distance and [105] which consider the lower bound on $\theta/||\eta_{\theta_n} - \eta_0||_\infty$. Secondly, unlike the two aforementioned works, we assume a restriction only on the total number of parameters K_n in the system instead of developing the results for the same number of nodes in each layer, an assumption which can severely restrict the space of neural networks solution one works with. Third, both [105] and [3] assume that there exists a true sparse solution all of whose coefficients are bounded above by a constant B (see condition 4.3 in [3]). We impose no such restriction to begin with on our true neural network solution but derive the most relaxed condition on the joint growth of the number of nodes and strength of connections between active nodes to allow for rates of contraction to hold (see condition 3. in (A3)). Indeed, if we make the assumption that all coefficients of the neural net are bounded above by B, condition 3. in (A3) simplifies to a restriction only on the number of nodes as in [105] and [3]. Lastly, both these works establish their contraction results in context of regression problems which allows them to use results from [111]. However, our systematic development here requires the derivation of the tools for a classification set up and ideas may be extended to other generalized linear models.

Theorem 3.4.1 *Let $K_n \sim n^a$, $0 < a < 1$ and $p_n = p$, $L_n = L$ be constants independent of n.*
(A2) *The prior parameters in* (3.9) *satisfy assumption (A1) and $||\boldsymbol{\mu}_n||_2^2 = o(n)$. Then,*

$$\pi^*(\mathcal{U}_\varepsilon^c) \xrightarrow{P_0^n} 0$$

Here, $||.||_2$ is the L_2 norm of a vector as in definition A.0.1 in the appendix B. By the above theorem, for any $\nu > 0$, $\pi^*(\mathcal{U}_\varepsilon^c) < \nu$ with probability tending to 1 as $n \to \infty$. Under the conditions of theorem 3.4.1, it can be established that the true posterior satisfies $\pi(\mathcal{U}_\varepsilon^c|\boldsymbol{y}_n, \boldsymbol{X}_n) < 2e^{-n\varepsilon^2/2}$ with probability tending to 1 as $n \to \infty$ (see theorem A.0.18 part 1. in the appendix D). This implies that the probability of the $\varepsilon-$Hellinger neighborhoods of the true function ℓ_0 for the true posterior increases at the rate of $1 - 2e^{-n\varepsilon^2/2}$ in contrast to the slow rate of $1 - \nu$ for the variational posterior.

Theorem 3.4.2 *Suppose $K_n \sim n^a$, $0 < a < 1$, $L_n \sim \log n$, $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1 - a$. Suppose,*
(A3) *There exists a sequence of neural network functions $\eta_{\boldsymbol{\theta}_n^*}$ satisfying*

1. $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 = o(\epsilon_n^2)$
2. $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$
3. $\log\left(\sum_{v=0}^{L_n} k_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*\right) = O(\log n)$

where $a_{v'n}^* = \sup_{s=0,\cdots,k_{(v'+1)n}} ||\boldsymbol{A}_{v'}^*[s]]||_1$.
(A4) *The prior parameters satisfy assumption (A1) and $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$.*

By the above theorem, for any $\nu > 0$, $\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) < \nu$ with probability tending to 1 as $n \to \infty$. Under the conditions of theorem 3.4.2, it can be established that the true posterior satisfies $\pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\boldsymbol{y}_n, \boldsymbol{X}_n) < 2e^{-n\varepsilon^2\epsilon_n^2/2}$ with probability tending to 1 as $n \to \infty$ (see theorem A.0.19 part 1. in the appendix D). This implies that the probability of the shrinking $\varepsilon\epsilon_n-$Hellinger neighborhoods of the true function ℓ_0 for the true posterior increases at the rate of $1 - 2e^{-n\varepsilon^2\epsilon_n^2/2}$ in contrast to the slow rate of $1 - \nu$ for the variational posterior.

Remark: For a single layer, assumption (A3) condition 3. holds if the number of input features increases at a rate polynomial in n. As the number of layers increases, one needs the row sums in the true solution $A_v^*, v = 0, \cdots, L_n$ to be bounded. This shows that even with a control on the number of nodes, the strength of the signal into every active node node must be well controlled (this corresponds to edge selection following node selection).

3.4.2 Discussion of the proof

We next briefly outline the main steps in the the proof of theorems 3.4.1 and 3.4.2. The details are deferred to appendix C. The first step of the proof is to establish that $d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n))$ is bounded below by a quantity which is determined by the rate of consistency of the true posterior. The second step is to show $d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n))$ is bounded above at a rate which is greater than its lower bound if and only if the variation posterior is consistent. Note,

$$\begin{aligned}
& d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) \\
= & \int_{\mathcal{U}_\varepsilon} \pi^*(\boldsymbol{\theta}_n) \log \frac{\pi^*(\boldsymbol{\theta}_n)}{\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)} d\boldsymbol{\theta}_n + \int_{\mathcal{U}_\varepsilon^c} \pi^*(\boldsymbol{\theta}_n) \log \frac{\pi^*(\boldsymbol{\theta}_n)}{\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)} d\boldsymbol{\theta}_n \\
= & -\pi^*(\mathcal{U}_\varepsilon) \int_{\mathcal{U}_\varepsilon} \frac{\pi^*(\boldsymbol{\theta}_n)}{\pi^*(\mathcal{U}_\varepsilon)} \log \frac{\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)}{\pi^*(\boldsymbol{\theta}_n)} d\boldsymbol{\theta}_n - \pi^*(\mathcal{U}_\varepsilon^c) \int_{\mathcal{U}_\varepsilon^c} \frac{\pi^*(\boldsymbol{\theta}_n)}{\pi^*(\mathcal{U}_\varepsilon^c)} \log \frac{\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)}{\pi^*(\boldsymbol{\theta}_n)} d\boldsymbol{\theta}_n \\
\geq & \ \pi^*(\mathcal{U}_\varepsilon) \log \frac{\pi^*(\mathcal{U}_\varepsilon)}{\pi(\mathcal{U}_\varepsilon|\boldsymbol{y}_n, \boldsymbol{X}_n)} + \pi^*(\mathcal{U}_\varepsilon^c) \log \frac{\pi^*(\mathcal{U}_\varepsilon^c)}{\pi(\mathcal{U}_\varepsilon^c|\boldsymbol{y}_n, \boldsymbol{X}_n)}, \text{ by Jensen's inequality}
\end{aligned}$$

where $\mathcal{U}_\varepsilon$ as in (4.16) note that for any $\varepsilon > 0$, Since $\pi(\mathcal{U}_\varepsilon|\boldsymbol{y}_n, \boldsymbol{X}_n) \leq 1$, thus

$$\begin{aligned}
\geq & \ \pi^*(\mathcal{U}_\varepsilon) \log \pi^*(\mathcal{U}_\varepsilon) + \pi^*(\mathcal{U}_\varepsilon^c) \log \pi^*(\mathcal{U}_\varepsilon^c) - \pi^*(\mathcal{U}_\varepsilon^c) \log \pi(\mathcal{U}_\varepsilon^c|\boldsymbol{y}_n, \boldsymbol{X}_n) \\
\geq & -\pi^*(\mathcal{U}_\varepsilon^c) \log \pi(\mathcal{U}_\varepsilon^c|\boldsymbol{y}_n, \boldsymbol{X}_n) - \log 2, \quad \text{since } x \log x + (1-x) \log(1-x) \geq -\log 2 \\
= & -\pi^*(\mathcal{U}_\varepsilon^c) \left(\log \int_{\mathcal{U}_\varepsilon^c} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n - \log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \right) - \log 2
\end{aligned}$$

Thus, with

$$A_n = \log \int_{\mathcal{U}_\varepsilon^c} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \qquad B_n = -\log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \tag{3.29}$$

we get the following main step towards the proof of theorems 3.4.1 and 3.4.2.

$$\boxed{-\pi^*(\mathcal{U}_\varepsilon^c) A_n \leq d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) + |B_n| + \log 2} \tag{3.30}$$

In the above proof we have assumed $\pi^*(\mathcal{U}_\varepsilon) > 0$, $\pi^*(\mathcal{U}_\varepsilon^c) > 0$. If $\pi^*(\mathcal{U}_\varepsilon^c) = 0$, there is nothing to prove. If $\pi^*(\mathcal{U}_\varepsilon) = 0$, then following the steps of the proof in appendix C, we get $\varepsilon^2 = o_{P_0^n}(1)$ which is a contradiction. The first term A_n is decomposed as

$$e^{A_n} = \int_{\mathcal{U}_\varepsilon^c \cap \mathcal{F}_n} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n + \int_{\mathcal{U}_\varepsilon^c \cap \mathcal{F}_n^c} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n$$

where $\{\mathcal{F}_n\}_{n=1}^\infty$ is a suitably chosen sequence of sieves. Indeed our choice of $\mathcal{F}_n$ is given by

$$\mathcal{F}_n = \Big\{\boldsymbol{\theta}_n : |\theta_{jn}| \leq C_n, j = 1, \cdots, K(n)\Big\} \tag{3.31}$$

where $C_n = e^{n^b/K_n}$ in theorem 3.4.1 and $C_n = e^{n^b \epsilon_n^2/K(n)}$ in theorem 3.4.2 respectively where b is chosen to ensure Hellinger bracketing entropy (see definition A.0.2 in the appendix B) of $\mathcal{F}_n$ is well controlled (proposition A.0.16 in the appendix C). Secondly, the prior needs to give negligible probability outside $\mathcal{F}_n^c$ so that term e^{A_n} is well controlled. The prior in (3.9) satisfies this for theorem 3.4.1 and theorem 3.4.2 with assumptions (A1), (A2) and (A1), (A4) respectively.

The second quantity B_n is controlled by the rate at which the prior gives mass to shrinking KL neighborhoods of the true density ℓ_0. In theorem 3.4.1, this rate is controlled as long as the prior parameters in (3.9) satisfy (A1) and (A2). In theorem 3.4.2, the same rate is controlled as long as the prior parameters satisfies (A1) and (A4) and the true function η_0 has a neural network solution which satisfies assumption (A3).

Finally, we bound $d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n))$ by $d_{\mathrm{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n))$ for a suitable $q \in \mathcal{Q}_n$ (see propositions B.0.5 and A.0.17 in the appendix). From Relation (A.30) in the appendix,

$$d_{KL}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) \leq d_{\mathrm{KL}}(q, p) + \left|\int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| + \left|\log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| \tag{3.32}$$

The last term above is nothing but $|B_n|$. The second term is the most crucial quantity.

$$\left|\int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| \approx n \int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n.$$

For both the theorems 3.4.1 and 3.4.2, the right hand side can always be controlled by choosing $q = MVN(m_n^*, s_n^*)$ for a suitable choice of the sequence m_n^* and s_n^*. We discuss the choice of s_n^* in the appendix C. For theorem 3.4.1, $m_n^* = \theta_n^*$ where $\eta_{\boldsymbol{\theta}_n^*}$ is the finite neural network approximation of η_0 and for theorem 3.4.2, the $m_n^* = \theta_n^*$ corresponds to $\eta_{\boldsymbol{\theta}_n^*}$, the rate controlled neural network approximation of assumption (A3). Finally, the first term in (3.32) is determined by both prior and q. In theorem 3.4.1, it is controlled as long as the prior parameters in (3.9) satisfy (A1), (A2). In theorem 3.4.2, the same rate is controlled as long as the prior parameters satisfies (A1), (A4) and the sequence $\boldsymbol{\theta}_n^*$ satisfies assumption (A3).

In light of the above discussion, there are three main properties which a prior must satisfy to allow for the convergence of variational posterior. For any $\nu > 0$

1. For a sequence of sieves $\{\mathcal{F}_n\}_{n=1}^{\infty}$ with well controlled Hellinger bracketing entropy,

$$\int_{\mathcal{F}_n^c} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq e^{-n\epsilon_n^2 \nu}, n \to \infty$$

2. With $\mathcal{N}_\varepsilon$ as in (3.28),

$$\int_{\mathcal{N}_{\varepsilon\epsilon_n^2}} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \geq e^{-n\epsilon_n^2 \nu}, n \to \infty$$

3. For a q satisfying $\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n < \varepsilon, n \to \infty$,

$$d_{KL}(q, p) \leq n\epsilon_n^2 \nu, n \to \infty$$

Whereas condition 1 and 2 are standard assumptions for consistency of true posterior (see assumptions 1 and 2 in [4] and theorem 2 in [74]), condition 3 is an additional requirement which makes the variational posterior consistent. The proof presented in this section can be generalized to a much wider class of priors satisfying (1)-(3).

3.4.3 Classification consistency

In this section, we discuss the classification accuracy of the predictions made by the variational posterior by comparing to the optimal mis-classification error. In view of (4.2), let $R(\hat{C})$ and $R(C^{\text{Bayes}})$ denote the classification accuracy under the variational classifier in (3.24) and the Bayes classifier in (4.3) respectively, then

$$\begin{aligned}
&|R(\hat{C}) - R(C^{\text{Bayes}})| = |E_X E_{Y|X}[I_{\hat{C}(X)\neq Y} - I_{C^{\text{Bayes}}(X)\neq Y}]| \\
&= |E_X E_{Y|X}[(I_{\hat{C}(X)=0} - I_{C^{\text{Bayes}}(X)=0})\sigma(\eta_0(X)) + (I_{\hat{C}(X)=1} - I_{C^{\text{Bayes}}(X)=1})(1 - \sigma(\eta_0(X)))]| \\
&\leq 2E_X[I_{\hat{C}(X)\neq C^{\text{Bayes}}(X)}|\sigma(\eta_0(X)) - 1/2|] \\
&= 2E_X[I_{\sigma(\hat{\eta}(X))\geq 1/2, \sigma(\eta_0(X))<1/2}|\sigma(\eta_0(X)) - 1/2| + I_{\sigma(\hat{\eta}(X))<1/2, \sigma(\eta_0(X))\geq 1/2}|\sigma(\eta_0(X)) - 1/2|] \\
&\leq 2E_X|\sigma(\eta_0(X)) - \sigma(\hat{\eta}(X))| \qquad (3.33)
\end{aligned}$$

The above result establishes how the difference in classification accuracy depends on the logit links $\hat{\eta}(X)$ and $\eta_0(X)$ as defined in (3.23) and (4.1) respectively. Using the above result, in corollary 3.4.3, we establish the classification accuracy of the variational estimate $\hat{\eta}(\boldsymbol{x})$ under no assumptions on the true function $\eta_0(\boldsymbol{x})$. In corollary 3.4.4, we establish the same result under assumption (A3) on the true function $\eta_0(\boldsymbol{x})$. Note, although theorem 3.4.1 requires minimal assumptions, it gives a much weaker convergence result on the classification accuracy.

Corollary 3.4.3 *Under the conditions of theorem 3.4.1,*

$$|R(\hat{C}) - R(C^{\text{Bayes}})| \xrightarrow{P_0^n} 0$$

By the above corollary, for any $\nu > 0$, $|R(\hat{C}) - R(C^{\text{Bayes}})| < \nu$ with probability tending to 1 as $n \to \infty$. Under the conditions of theorem 3.4.1, it can be established that the true posterior also gives classification consistency at the same rate and there is no loss in using a variational posterior approximation (see theorem A.0.18 part 2. in the appendix D).

Corollary 3.4.4 *Under conditions of theorem 3.4.2, for every* $0 \leq \kappa \leq 2/3$*,*

$$\epsilon_n^{-\kappa}|R(\hat{C}) - R(C^{\text{Bayes}})| \xrightarrow{P_0^n} 0$$

By the above corollary, for any $\nu > 0$, $0 \leq \kappa \leq 2/3$, $|R(\hat{C}) - R(C^{\text{Bayes}})| < \nu\epsilon_n^k$ with probability tending to 1 as $n \to \infty$. Under the conditions of theorem 3.4.2, it can be established that the true posterior satisfies $|R(\hat{C}) - R(C^{\text{Bayes}})| < \nu\epsilon_n^k$ for every $\nu > 0$, $0 \leq \kappa \leq 1$ with probability tending to 1 as $n \to \infty$ (see theorem A.0.19 part 2. in the appendix D). Thus, the classification consistency occurs at the rate $\epsilon_n^{2/3}$ for the variational posterior in contrast to ϵ_n for the true posterior.

3.5 Simulation Studies.

In this section, we study the performance of the four algorithms viz BBVI, BBVI-CV, BBVI-RMS, BBVI-CV-RMS in the context of two simulation scenarios. We used approximate 2:1 ratio for training and test cases. All the covariates are normalized. We adopted a 10-fold cross-validation to avoid optimistically-biased estimates of model performance.

3.5.1 Simulation Scenarios

Scenario 1: We simulate $n = 3000$ observations from a 2-2-2-1 network, i.e. a neural network with 2 input features, 2 hidden layer with 2 nodes each and 1 output layer as

$$y_i = \begin{cases} 0, & \boldsymbol{b}_2 + \boldsymbol{A}_2\psi(\boldsymbol{b}_1 + \boldsymbol{A}_1(\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x}_i))) > 0 \\ 1, & \text{otherwise} \end{cases}$$

where $\boldsymbol{x}_i \in \mathbb{R}^2$, are i.i.d. from $N(0,1)$ and entries in $\boldsymbol{b}_j, \boldsymbol{A}_j$, $j = 0, 1, 2$ are i.i.d. from $U(0,1)$.

Scenario 2: We simulate $n = 3000$ observations from the following non linear function as

$$y_i = \begin{cases} 0, & 2e^{x_i[1]} + 3\sin(x_i[2]x_i[3]) + 4x_i[4]^3 - 3 > 0 \\ 1, & \text{otherwise} \end{cases}$$

where $\boldsymbol{x}_i \in \mathbb{R}^4$ are i.i.d. from $N(0,1)$.

3.5.2 Parameters choice for statistical and computational models.

In order to implement the BBVI, BBVI-CV, BBVI-RMS, and BBVI-CV-RMS, we need to make a valid choice of the prior parameters μ_{jn}, σ_{jn} for $j = 1, \cdots, K_n$ as in (3.9). We use the choice of

$\mu_{jn} = 0$ and $\sigma_{jn} = 1$ for our prior parameters. Indeed, this choice satisfies conditions (A1), (A2) and (A4) as assumed in the consistency proofs of theorems 3.4.1 and 3.4.2. Next, we need to make a choice on the number nodes in each hidden layer. We experiment with 1 and 2 hidden layers with 2 nodes in each layer. The choice of number of nodes satisfy the assumption of theorem 3.4.1 and 3.4.2.

3.5.3 Gradient stabilization paramaters.

The choice of the initial learning rate is $\rho_t = 1e^{-4}$, $t \geq 1$ for BBVI and BBVI-CV and $\rho_t = 1e^{-1}$, $t \geq 1$ for BBVI-RMS and BBVI-CV-RMS. These values were chosen to ensure the optimal performance of the algorithms, however little sensitivity to the initial choice was observed. As explained in section 3.3, to allow for stable optimization, we study the sensitivity to the different samples sizes S, use of control variates and the RMSprop based gradient descent method. The choice of sample size S is sensitive to the performance to model in terms of algorithmic stability and convergence time. Whereas each update with small sample size takes less time, the variability of the estimate is high. On the other hand a large sample size leads to less variable estimates but each update takes a much longer time. We experimented with $S = 200$, $S = 500$ and $S = 1000$. For scenario 1, Figures 3.1 and 3.2 illustrates how the ELBO changes with S for one and two layers respectively. For scenario 2, Figures 3.3 and 3.4 provide the same illustration for one and two layers respectively. It is evident that increase in S from 200 to 1000 stabilizes the ELBO and helps with a faster convergence.

As explained in section 3.3.4, the maximization of the ELBO requires stabilization of the variance of the stochastic gradient in (3.18) which is done by the use of control variate. For scenario 1, Figures 3.1 and 3.2 illustrates how the ELBO changes with use of control variates for one and two layers respectively. For scenario 2, Figures 3.3 and 3.4 provide the same illustration for one and two layers respectively. It is evident that the use of control variates stabilizes the ELBO by a huge margin and allows for its faster convergence. Finally, as explained in section A, the use of RMSprop stabilizes the optimization of ELBO by normalizing the gradients by their running

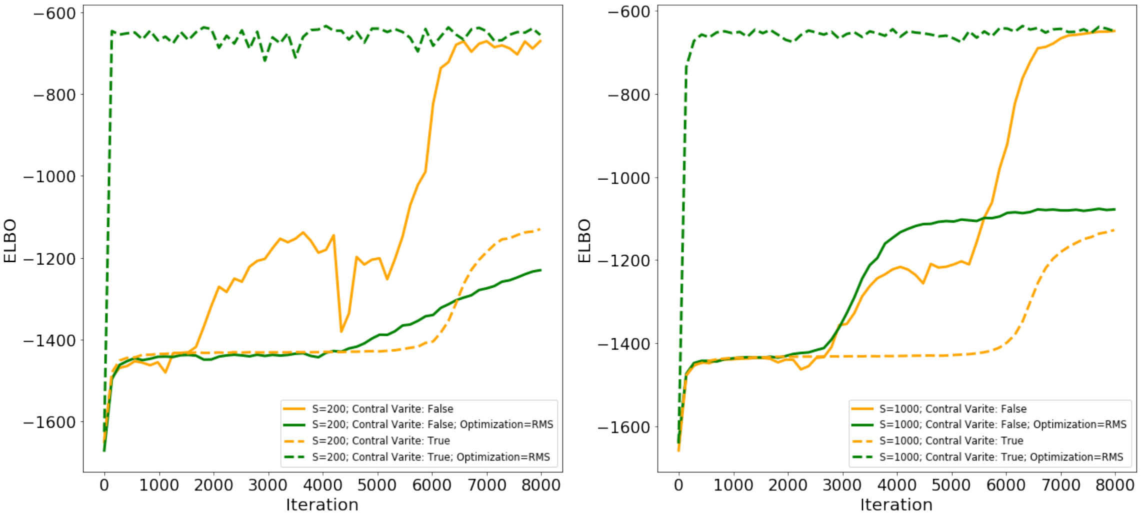


Figure 3.1: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 1 layer.

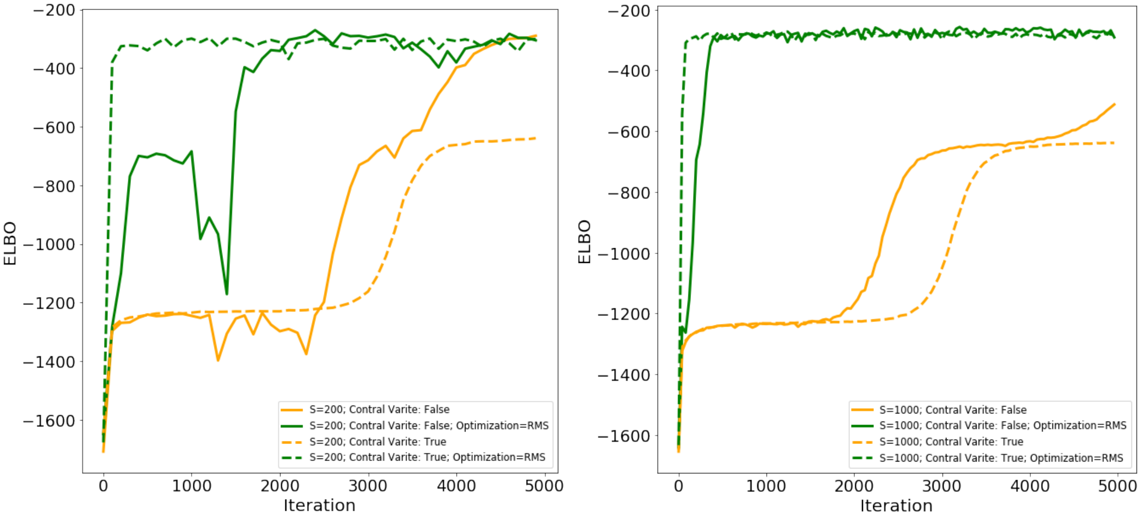


Figure 3.2: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 2 layers.

magnitude. For scenario 1, Figures 3.1 and 3.2 illustrates how the ELBO changes with use of RMSprop versus a fixed learning rate for one and two layers respectively. For scenario 2, Figures 3.3 and 3.4 provide the same illustration for one and two layers respectively. It is evident that the use of RMSprop leads to stable ELBO and faster convergence rates.

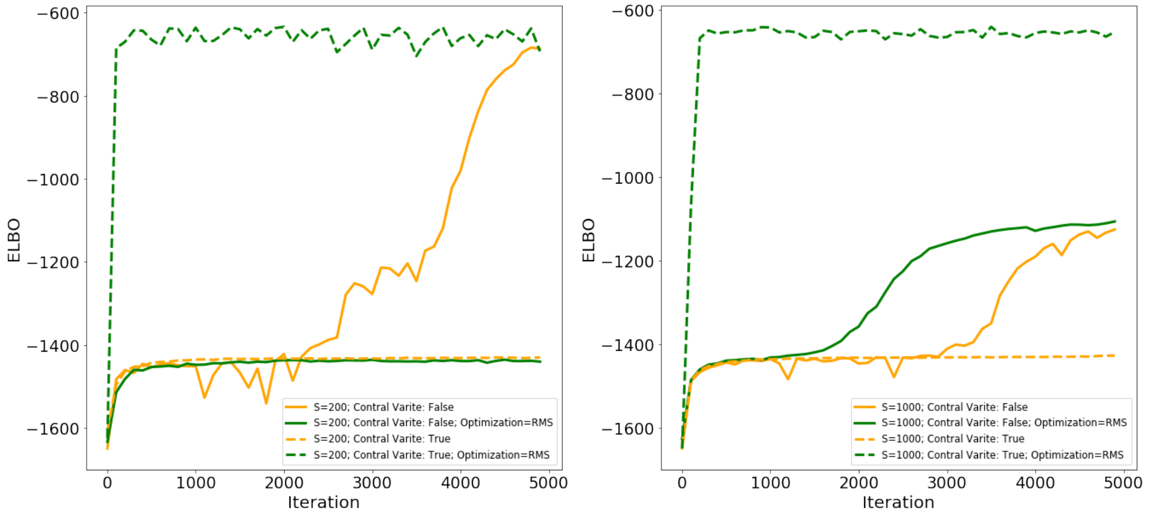


Figure 3.3: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 2 for 1 layer.

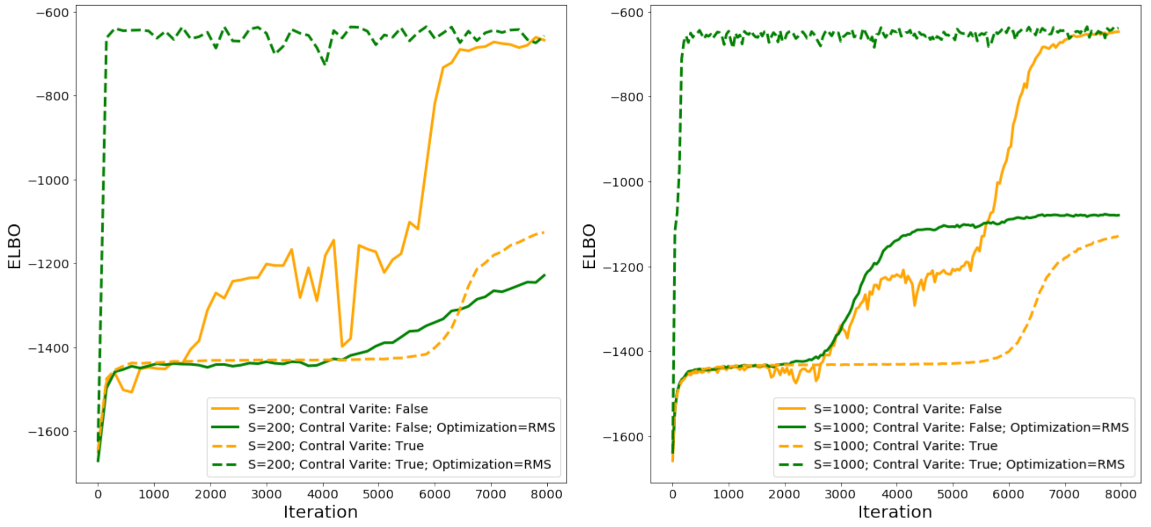


Figure 3.4: ELBO convergence of algorithms 1, 2, 4, 5 for scenario 1 for 2 layers.

			Testing accuracy(%)		Convergence time(s)	
Layers	Method	Sample size (S)	Fixed	RMSprop	Fixed	RMSprop
1	BBVI	200	97.41 ± 0.50	96.89 ± 0.93	23	114
		500	97.72 ± 0.38	97.52 ± 0.74	55	106
		1000	98.01 ± 0.33	97.38 ± 0.39	108	80
	BBVI-CV	200	97.82 ± 0.40	97.61 ± 0.60	21	6
		500	97.84 ± 0.40	97.67 ± 0.34	52	7
		1000	97.84 ± 0.42	97.94 ± 0.40	104	10
2	BBVI	200	97.79 ± 0.71	97.02 ± 1.10	200	98
		500	94.34 ± 3.82	97.75 ± 0.95	452	39
		1000	91.50 ± 5.17	98.11 ± 0.42	904	65
	BBVI-CV	200	96.34 ± 0.75	97.61 ± 0.44	118	17
		500	96.33 ± 0.73	97.30 ± 0.60	272	23
		1000	96.36 ± 0.74	97.74 ± 0.54	552	40

Table 3.1: Performance of algorithms algorithms 1, 2, 4, 5 for scenario 1.

3.5.4 Testing accuracy and convergence.

We evaluate the model's performance for all four algorithms BBVI, BBVI-CV, BBVI-RMS and BBVI-CV-RMS under two criteria (1) testing accuracy (2) convergence time. The test accuracy, $T(C)$ of a classifier is given by $1-R(C)$ where $R(C)$ is the mis-classification error rate as described in (4.2). The convergence criterion is defined as the point where Monte Carlo estimate of the ELBO as in (3.19) converges.

For scenario 1, table 3.1 gives the performance of four algorithms for 1 and 2 layers. The best average accuracy of 98.11% is obtained for BBVI-RMS with S = 1000 with 2 layers. The optimal time is achieved with BBVI-CV-RMS for S = 200 for one layer with average accuracy of 97.61%. We can make two conclusions here, although the true data is generated from 2 layer network structure, a one layer approximation is fairly competitive. BBVI-CV-RMS with S = 200 provides the best convergence time of nearly 6 sec for one layer and 17 sec for two layers with competitive accuracy. For scenario 2, table 3.2 gives the performance of four algorithms for 1 and 2 layers. The best average accuracy of 91.12% is obtained for BBVI-CV-RMS with S = 1000 with 2 layers. The optimal time is achieved with BBVI-CV-RMS for S = 200 for one layer with average accuracy of 91.11%. The improvement obtained by moving from 1 to 2 layers is only marginal. BBVI-CV-RMS with S = 200 provides the best convergence time of nearly 19 sec for one layer and 11 sec for two

			Testing accuracy(%)		Convergence time(s)	
Layers	Method	Sample size (S)	Fixed	RMSprop	Fixed	RMSprop
1	BBVI	200	83.66 ± 14.51	88.71 ± 7.12	190	15
		500	90.22 ± 0.54	90.32 ± 0.98	364	390
		1000	90.28 ± 0.75	90.41 ± 0.71	732	710
	BBVI-CV	200	90.51 ± 0.87	90.42 ± 0.64	17	19
		500	90.51 ± 0.87	90.65 ± 0.61	36	33
		1000	90.53 ± 0.91	90.78 ± 0.49	69	37
2	BBVI	200	88.40 ± 0.50	89.89 ± 0.88	256	421
		500	90.52 ± 0.38	90.48 ± 0.74	518	544
		1000	90.61 ± 0.33	90.32 ± 0.65	906	608
	BBVI-CV	200	90.62 ± 0.40	91.11 ± 0.58	444	11
		500	90.74 ± 0.40	90.98 ± 0.54	862	12
		1000	90.72 ± 0.42	91.12 ± 0.53	1646	13

Table 3.2: Performance of algorithms 1, 2, 4, 5 for scenario 2

layers with competitive accuracy.

3.5.5 Large number of layers and challenges.

We finally discuss the performance for all four algorithms BBVI-RMS and BBVI-CV-RMS when the number of layers are 3. For 3 layers, using a fixed learning rate does not allow for the maximization of the ELBO. This may be attributed to the different scales of the gradients for the different parameters. Similar behavior is also observed in parametric optimization of artificial deep neural networks (see [57] for more details). From table 3.3, it is evident that the improvement from using 3 layers over 2 layers provides only a marginal improvement for scenario 1. For scenario 2, the performance at 3 layers is worse than that in the case of 2 layers.

As explained in the previous sections, the performance of both BBVI-RMS and BBVI-CV-RMS improves with increase in sample size S. However, a great deal of sensitivity to choice of the initial learning rate was observed. The observed sensitivity was even more profound in the case of control variates especially under scenario 2. For scenario 1, the optimal learning rate ρ_t was found to be 0.1 and 0.3 (S=200) and 0.35(S=500 and 1000) for $t \geq 1$ for BBVI-RMS and BBVI-CV-RMS respectively. For scenario 2 under BBVI-RMS, the optimal learning rates were found to be $\rho_t = 0.055$, $\rho_t = 0.04$ and $\rho_t = 0.04$, $t \geq 1$ for $S = 200$, $S = 500$ and $S = 1000$ respectively. For scenario 2 under for BBVI-CV-RMS, the optimal learning rates $\rho_t = 0.4$, $\rho_t = 0.55$ and $\rho_t = 0.63$, $t \geq 1$ for $S = 200$, $S = 500$ and $S = 1000$ respectively. With the optimal choice of ρ_t at hand, the BBVI-CV-RMS provided faster convergence results with a comparable test accuracy to that of BBVI-RMS. This sensitivity to the choice of the initial learning rate especially in the case of control variates for large number of layers needs to be explored as a part of future work.

		Scenario 1		Scenario 2	
Method	S	Testing accuracy(%)	Time(s)	Testing accuracy(%)	Time(s)
BBVI-RMS	200	97.76 ± 0.87	218	84.68 ± 4.85	423
	500	97.65 ± 0.83	169	88.00 ± 5.56	631
	1000	98.21 ± 0.73	132	90.69 ± 0.67	714
BBVI-CV-RMS	200	96.23 ± 1.05	212	84.53 ± 8.90	33
	500	97.83 ± 0.81	166	88.28 ± 2.03	37
	1000	98.42 ± 0.72	124	89.33 ± 1.67	45

Table 3.3: Performance of algorithms 1, 2, 4, 5 for scenario 1 and scenario 2 for 3 layers.

3.6 Numerical Properties and Alzheimer's Disease Study

The transition from mild cognitive impairment (MCI) to Alzheimer's disease (AD) is of great interest for clinical researchers. Several studies over the past decade have shown and compared the performance of different machine learning methods on this classification task. For this classification problem, we illustrate the performance of variational Bayesian neural networks as developed under section 3.3 in terms of classification accuracy, numerical complexity and time of convergence. We implemented both algorithms, algorithm 1 and 2 and shall hence forth refer to them as BBVI and BBVI-CV respectively. For a comparative baseline, we also report the performance for several machine learning techniques as applicable to this task. We like to emphasize that, our primary goal here is to illustrate the computational methodology rather incremental improvement for a specific application.

Alzheimer's disease (AD) is a progressive, age-related, neurodegenerative disease and the most common cause of dementia [147, 148, 68]. Behaviorally, AD is commonly preceded by mild cognitive impairment (MCI), a syndrome characterized by decline in memory and other cognitive domains that exceed cognitive decrements associated with normal aging [148, 103]. However, the prodromal symptoms of MCI are not prognostically deterministic: individuals with MCI tend to progress to probable AD at a rate of 8%-15% per year, and most conversions occur within 3 years of presentation, [24, 44, 2]. We used T1-weighted MRI images from the collection of standardized datasets. The description of the standardized MRI imaging from ADNI can be found in http://adni.loni.usc.edu/methods/mri-analysis/adni-standardized-data.

This study used a subset of the MCI subjects from ADNI-1, who had data from demographic, clinical cognitive assessments, APOE4 genotyping, and MRI measurements. In total, there are 819 individuals with a baseline diagnosis of MCI, but we only consider patients whose follow-up period was at least 36 months and no missing values. The final samples included 265 subjects which included participants who were stable in their diagnosis (MCI-S) and those who converted to a diagnosis of AD over 3 years (MCI-C). We considered a total of 18 clinical predictors as potential of MCI-to-AD progression in our classification analyses. Structural MRI data were collected according to the ADNI acquisition protocol using T1-weighted scans (GradWarp, B1 Correction, N3, Scaled). Based on the extant literature, [68, 81], we used 24 ROI features as clinically significant of MCI to dementia progression.

The dependence and interactions among different modes of features (clinical, MRI) and within the modes may be different and hard to model explicitly. Thus, a neural network-based modeling is intuitive from predictive modeling and machine learning perspective. Of the 265 patients, 186 are selected by simple random sample as training cases and the remaining 79 as test cases. The approximate 2:1 ratio for training and test cases is, of course, arbitrary. All the covariates (except categorical variables) were z-normalized. The outcome y_i for the i^{th} patient is either 1 for MCI-C or 0 for MCI-S in classification study. 10-fold cross-validation is used to avoid optimistically-biased estimates of model performance.

3.6.1 Parameters choice for statistical and computational models.

In order to implement the BBVI, BBVI-CV, BBVI-RMS, and BBVI-CV-RMS, we use the choice of $\mu_{jn} = 0$ and $\sigma_{jn} = 1$ similar to section 3.6. For the number of layers, we found that one layer provides a good enough performance and inclusion of addition layers do not offer additional improvement in the accuracy. We tried with $k_{1n} = 2, 10, 20$ and obtained the best results at $k_{1n} = 10$, the results of which are reported in this thesis.

3.6.2 Gradient stabilization paramaters.

The choice of the initial learning rate is $\rho_t = 1e^{-4}$, $t \geq 1$ for BBVI and BBVI-CV and $\rho_t = 1e^{-1}$, $t \geq 1$ for BBVI-RMS and BBVI-CV-RMS. As explained in section 3.3, to allow for stable optimization, we study the sensitivity to the different samples sizes S, use of control variates and the RMSprop based gradient descent method. For ADNI, figure 3.5 illustrates how the ELBO changes with S. It is evident that increase in S from 200 to 1000 stabilizes the ELBO and helps with a faster convergence. For ADNI, figure 3.5 illustrates how the ELBO changes with use of control variates. It is evident that the use of control variates stabilizes the ELBO by a huge margin and allows for its faster convergence. Similary, figure 3.5 also illustrates how the ELBO changes with use of RMSprop versus a fixed learning rate. It is evident that the use of RMSprop leads to stable ELBO and faster convergence rates.

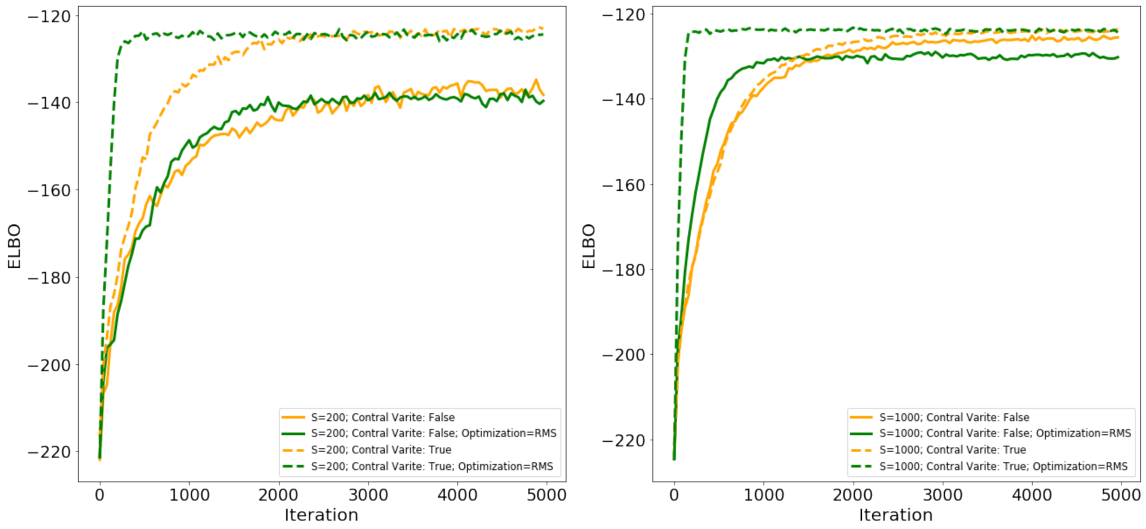


Figure 3.5: ELBO convergence of algorithms 1, 2, 4, 5 for ADNI.

3.6.3 Testing accuracy and convergence.

For ADNI, table 3.4 gives the performance of BBVI, BBVI-CV, BBVI-RMS, BBVI-CV-RMS for one layer. The best average accuracy of 76.88% was obtained for BBVI with $S = 200$ for BBVI. The optimal convergence time is achieved with BBVI-CV-RMS for $S = 200$ for one layer with average accuracy of 76.25% and convergence time is 36 seconds. Thus, the conclusions for real data corroborate the use of BBVI-CV-RMS for a single layer NN.

		Testing accuracy(%)		Convergence time(s)	
Method	Sample size (S)	Fixed	RMSprop	Fixed	RMSprop
BBVI	200	76.88 ± 3.32	75.75 ± 3.27	68	49
	500	76.75 ± 3.63	76.50 ± 3.90	105	62
	1000	76.75 ± 3.12	76.63 ± 3.21	231	65
BBVI-CV	200	76.75 ± 3.41	76.25 ± 3.83	146	36
	500	76.75 ± 3.58	76.63 ± 3.95	210	38
	1000	76.75 ± 3.71	76.75 ± 4.07	264	39

Table 3.4: Performance of algorithms 1, 2, 4, 5 for ADNI.

3.6.4 Numerical comparison with popular models

In this section, we numerically compare the testing accuracy of BBVI, BBVI-CV, BBVI-RMS and BBVI-CV-RMS and BBVI-CV to a few benchmark models which include logistic regression (LR) and support vector machine (SVM) as developed by [101, 87] and frequentist artificial neural network (ANN) [20, 54]. We also compared with a Bayesian neural network models which uses Stochastic Gradient MCMC [137] . For all neural network models, viz, artificial neural network (ANN) and Stochastic Gradient MCMC Bayesian neural network (SG-MCMC), the number of nodes are fixed at $k_n = 10$ with a single hidden layer.

Table 3.5 provides the training and testing accuracy and empirical standard errors for all methods under consideration. For the 4 models viz BBVI, BBVI-CV, BBVI-RMS and BBVI-CV-RMS, the results reported correspond to the optimal parameter combination which provides the best average test accuracy. Little to no difference was observed across different choices of the algorithm parameters (see table 3.4). LR, SVM, ANN and SG-MCMC have considerably larger standard errors for testing accuracy. One might observe an improvement in performance of SG-MCMC Bayesian neural network by optimally choosing their tuning parameters. However studying that is beyond the scope of this thesis as they are different methodology and the underlying statistical theories are not well established.

Classifier	Training accuracy (%)	Testing accuracy (%)
LR	82.1 ± 2.5	70.9 ± 5.5
SVM	80.3 ± 2.2	70.6 ± 5.5
ANN	82.0 ± 5.6	74.1 ± 6.8
SG-MCMC	80.8 ± 4.6	73.5 ± 5.9
BBVI	80.7 ± 2.1	76.9 ± 3.3
BBVI-CV	80.3 ± 2.3	76.8 ± 3.4
BBVI-RMS	81.2 ± 2.4	76.8 ± 3.3
BBVI-CV-RMS	82.8 ± 1.6	76.8 ± 4.1

Table 3.5: Performance for different classifiers. LR: Logistic regression. SVM: Support vector machine. ANN: Frequentist artificial neural network. SG-MCMC: Stochastic gradient MCMC Bayesian neural network

3.7 Conclusion and Discussion

The theoretical rigour and computational detail for variational Bayes neural network classifier presented in this article is novel and unique contribution to statistical literature. Although the variational Bayes is popular in machine learning, neither the computational method nor the statistical properties are well understood for complex modeling such as neural networks. We characterize the prior distributions and the variational family for consistent Bayesian estimation. The theory also quantifies the loss due to VB numerical approximation compared to the true posterior distribution. For practical implementation, we reveal that the algorithm may not be as simple and straightforward as it sounds in computer science literature, rather it requires careful crafting on several parameters associated in various steps. Nevertheless, the computation could be quite faster compared to popular Monte Carlo Markov Chain procedure of approximating the posterior distributions.

Although we build the framework on a multi-layer neural networks model with simplistic prior structure, the detail statistical theory and computational methodology are quite involved. This investigation opens up possibility of exploring much wider class of models and priors. For example, shrinkage priors, such as double exponential and horseshoe priors can be explored for building sparse neural networks or one can experiment with various other variational families. However, their computational details and associated statistical properties are not immediate. We hope this research will accelerate further development of statistical and computational foundation for variational inference in general machine learning research.

CHAPTER 4

LEARNING INTRINSIC DIMENSIONALITY OF FEATURE SPACE WITH VARIATIONAL BAYES NEURAL NETWORKS

4.1 Introduction

Bayesian neural networks (BNNs) have achieved state-of-the-art results in a wide range of tasks, especially in high dimensional data analysis including image recognition, biomedical diagnosis and others. One of the major disadvantage in using neural networks and deep networks is that they require a huge number of training data due to the large number of inherent parameters [140, 45]. For example, high-dimensional neural networks have been widely applied with regularization, dropout techniques or early stopping to prevent overfitting [118, 143]. Furthermore, most commonly used dimensional reduction techniques include Lasso [17], Ridge [58], Elastic net [152], Sparse group lasso [116], Bayesian Lasso [98], Horseshoe prior [16], principal component analysis [115]. Even though the l_1 and l_2 norm can force the weights to become zero or small, they do not have the regularizing effect of making the computed function simpler [70]. Additionally, all these methods rely on the use of whole data which severely increases the cost of both computation and memory storage.

In this chapter, we propose the use of a BNN on a compressed feature space to take care of the large p small n problem by projecting the feature space onto a smaller dimensional space using a random projection matrix. Random-projection (RP) is a powerful dimension reduction technique which uses RP matrices to map data into low-dimensional spaces. The use of RP in high dimensional statistics is motivated from the Johnson–Lindenstrauss Lemma [27] which states for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^p$, $\epsilon \in (0, 1)$ and $d > 8 \log n/\epsilon^2$, there exists a linear map $f : \mathbb{R}^p \to \mathbb{R}^d$ such that $(1 - \epsilon)||x_i - x_j||_2^2 \leq ||f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)||_2^2 \leq (1 + \epsilon)||x_i - x_j||_2^2$ for $i, j = 1, \cdots, n$. The properties of the RPs and their applications to statistical problems were furthered explored in [33, 13], etc..

In order to reduce the sensitivity to the choice of random matrices, one must pool information obtained from multiple projections. In this chapter, we adopt a Bayesian model averaging approach for combining information across multiple instances RP based neural networks. There are two main challenges of implementing Bayesian modeling averaging (1) due to the convoluted structure of the neural network likelihood, closed form expressions do not exist for the posterior distribution under each model (2) posterior distribution of model weights is completely intractable and no closed form solutions exist. Thereby, the implementation of standard Markov Chain Monte Carlo (MCMC) is next to impossible. Further, the computation and storage cost associated with MCMC implementation is humongous since each posterior model weight is dependent on the posterior model weight of the remaining models.

To address the challenges of MCMC implementation, we use variational inference (VI) [63, 9] approach to provide an approximate solution for Bayesian model averaging (BMA) to allow for combining of BNNs with multiple instances of compression on the feature space. There has been a plethora of literature implementing variational inference in the neural networks [10]. However, their implementation makes use of the entire feature space, thereby putting a great burden on computational stability and memory storage. We address two main challenges in this thesis (1) developing a variational Bayes (VB) solution for BNNs with compressed feature space (2) providing a VB solution for doing BMA across multiple instances of RP. Further, for a given instance of random compression, we establish the posterior contraction rates for the variational posterior for classification (the theory is extendable to regression set up with minor modifications). In this direction, we provide characterization of the prior, variational posterior and the RP matrix which guarantees the convergence of the variational Bayes neural network (VBNN) under the compressed feature space to the true density of the observations.

The main advantage of implementing a BMA approach is that it gives the posterior model weights under each compression of feature space. The so obtained posterior model weights in turn induce a probability distribution on the projected dimension of the feature space. The mode of this probability distribution concentrates around the intrinsic dimensionality of the feature space. To improve the prediction performance, the BMA approach is then applied to a pool of RP matrices whose projected dimension lie in a neighborhood of the intrinsic dimensionality. Finally, we study the numerical behavior of the proposed procedure in the light of simulation and real data sets. To the best of our knowledge there exist no literature which provides theoretical guarantees and computation algorithm of VBNNs with compressed feature space.

For a long time, people have studied feature reduction using projection matrix in both supervised and unsupervised learning. [146] proposed semisupvervised classification with graph construction and the idea of projection matrix which is used to preserve the local and global structure of data. In addition to semisupervised learning, projection method have been used in convolutional neural network, [125] introduced an efficient convolutional neural network which can control how much context information can be incorporated into each specific position using word-embedding projection matrix. In terms of unsupervised learning, [134] proposed an unsupervised adaptive embedding method which combined the calculation of projection matrix and construction of affinity graph together.

Early works on Bayesian neural networks (BNNs) have been comprehensively discussed by [85, 96, 71]. With the computational and information science advancement, recent developments with higher efficient BNNs can be found in [120, 93, 61, 64] and the references therein. However, with increase in the dimension of the feature space, the prediction accuracy of BNN's is severely compromised. To circumvent this issue, penalization and sparse network based approach has been studied by [80, 45, 140, 48, 3], etc.. The major drawback of these sparsity based methods is one needs to work on the entire data which increases the time of implementation by a manifold. With the work of [27], the idea of using RPs to overcome the curse of dimensionality became very popular. Further, RPs have been used in a wide range of statistical problems [1, 86, 84, 39, 55, 40, 49], etc.. To ensemble information across projections, [14] uses a bagging approach for classification problems as in [12]. On the other hand, the works of [52] and [53] propose the use of BMA in the context of linear regression and Gaussian processes.

There exists a plethora of literature implementing variational inference [9] to overcome the drawback of a full MCMC implementation. The majority of Black-box variational methods for

Bayesian learning of neural networks are based on Pathwise gradient estimator [41, 83, 15, 11, 121], which is computed using reparameterization trick [106]. Another line of Black-box variational extensions is based on the score-function estimator using Monte Carlo estimator to find the full gradient, including control variate [107] and stochastic search [97]. Theoretical properties of the variational posterior in context of individual models have been studied in the works of [6, 135, 100, 149, 3]. The works of [65] and [72] explore variational inference for BMA in the context of generalized linear models and graph on functions respectively. To the best of our knowledge, BMA in context of Bayesian neural networks with compressed feature space remains unexplored.

Firstly, we introduce the RP idea in a neural network predictive model where the feature space grow exponentially with training sample size which in turn significantly reduces the computational complexity and storage capacity associated with BNNs. Second, we apply the BMA idea in conjunction with VB to allow for parallelization across RPs without compromising the uncertainty quantification of a Bayesian approach. Third, we develop the associated statistical foundation, namely the posterior contraction of the variational posterior for BNNs under a compressed feature space. The theory not only provide trustworthiness to our model, the results also provide theoretical guidelines for prior selection and the choice of variational family of distributions. Fourth, we innovatively apply the learned posterior model weights to obtain the intrinsic dimensionality of the feature space. Fifth, to improve predictive accuracy, we employ VB with BMA on a subspace of RPs with projected dimension centred around the intrinsic dimensionality. Fifthly, we provide numerical results to enunciate that our proposed approach learns well the intrinsic dimension of feature space and beats the predictive performance of all competing methods for the large p small n problems. Lastly, the performance of the proposed methodology has been enunciated in the context real data sets like ADNI and MNIST.

4.2 Bayesian neural network for random projection based compressed feature space

4.2.1 Bayesian neural network model

For a binary random variable Y, representing the class levels 0 or 1, and a feature vector $X \in \mathbb{R}^p$ with some marginal distribution P_X, consider the classification problem

$$P(Y = 1|X = \boldsymbol{x}) = \sigma(\eta_0(\boldsymbol{x})) = 1 - P(Y = 0|X = \boldsymbol{x}) \tag{4.1}$$

where $\eta_0(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}$ is some continuous function and $\sigma(.) = e^{(.)}/(1 + e^{(.)})$ is the sigmoid function. Following [14] and [141], the test error of a classifier C is given by

$$R(C) = \int_{\mathbb{R}^p \times \{0,1\}} I_{\{C(X) \neq Y\}} dP_{X,Y} \tag{4.2}$$

where the joint density $P_{X,Y}$ is a product of (4.1) and P_X. The Bayes classifier is then

$$C^{\text{Bayes}}(\boldsymbol{x}) = \begin{cases} 1, & \sigma(\eta_0(\boldsymbol{x})) \geq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Since $\eta_0(\boldsymbol{x})$ is unknown, we thereby use single-layer neural network model approximation with k nodes:

$$\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{k_n} \beta_j \psi(\gamma_{j0} + \boldsymbol{\gamma}_j^T \boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \psi(\gamma_0 + \boldsymbol{\Gamma}\boldsymbol{x}) \tag{4.4}$$

where $\boldsymbol{\beta} = [\beta_0, \cdots, \beta_k]$, $\gamma_0 = [\gamma_{10}, \cdots, \gamma_{k0}]$ and $\boldsymbol{\Gamma} = [\gamma_1, \cdots, \gamma_k]$ and $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}, \gamma_0, \text{vec}(\boldsymbol{\Gamma}))$ is the set of all the parameters. Note, $\boldsymbol{\theta}$ is a $K \times 1$ vector where $K = 1 + k + k(p + 1)$. Both k and $(p >> n)$ grow as a function of n. We then use the following model for the problem in (4.1).

$$P(Y = 1|X = \boldsymbol{x}) = \sigma(\eta_{\boldsymbol{\theta}}(\boldsymbol{x})) = 1 - P(Y = 0|X = \boldsymbol{x}) \tag{4.5}$$

4.2.2 Compression in the feature space with random projections

There exists several choices for compressing the feature space $\boldsymbol{X}$ using RP matrices such as those proposed in [33, 13, 14, 27, 26], etc.. For a given choice of the compression matrix A, we consider single-layer neural network with k nodes for the input vector $A\boldsymbol{x}$ as

$$\eta_{\boldsymbol{\theta}}(A\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^\top \psi(\gamma_0 + \boldsymbol{\Gamma}(A\boldsymbol{x})) \tag{4.6}$$

where A is a $d_A \times p$ projection matrix, $\boldsymbol{\beta}$ and γ_0 are $k \times 1$ vector and $\boldsymbol{\Gamma}^\top = [\gamma_1, \cdots, \gamma_k]$ is now a $d_A \times k$ matrix. Thus, in the projected space the number of parameters reduce from $K = kp + 2k + 1$ to $K_A = kd_A + 2k + 1$. This leads to the following model under the projected space

$$P(Y = 1|X = \boldsymbol{x}) = \sigma(\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})) = 1 - P(Y = 0|X = \boldsymbol{x}) \tag{4.7}$$

Experiments with different projection matrices suggested the use of the one in [52]. In this method, we draw the elements A_{ij} independently, setting $A_{ij} = -1/\sqrt{a_*}$, with probability a_*^2, 0 with probability $2a_*(1 - a_*)$ and $1/\sqrt{a_*}$ with probability $(1 - a_*)^2$, with the rows of A then normalized using Gram-Schmidt orthogonalization. The parameter $a_* \in (0.1, 1)$ provides a handle on the sparsity of the projection matrix. We do not rely on the data to generate A. Also, the algorithmic implementation discussed can be generalized to any arbitrary class of projection matrices.

4.2.3 Prior choice

For the neural network $\eta_{\boldsymbol{\theta}_A}(A\boldsymbol{x})$ based on the projected input $A\boldsymbol{x}$, we assume an independent Gaussian prior on each of the entries of $\boldsymbol{\theta}_A$, i.e. $p(\boldsymbol{\theta}_A|M_A) = \text{MVN}(\boldsymbol{\mu}_A, \text{diag}(\boldsymbol{\sigma}_A))$, where $\text{diag}(\boldsymbol{\sigma}_A)$ is a diagonal matrix. With this choice of the prior and likelihood as in (4.7), the posterior distribution based on the compressed data set $(y_i, A\boldsymbol{x}_i)_{i=1}^n$ is given by

$$\pi(\boldsymbol{\theta}_A|M_A) = \frac{L(\boldsymbol{\theta}_A|M_A)p(\boldsymbol{\theta}_A|M_A)}{\int L(\boldsymbol{\theta}_A|M_A)p(\boldsymbol{\theta}_A|M_A)d\boldsymbol{\theta}_A} \tag{4.8}$$

where M_A is the model induced by random matrix A with corresponding likelihood $L(\boldsymbol{\theta}_A|M_A) = \prod_{i=1}^n \exp(y_i\eta_{\boldsymbol{\theta}}(A\boldsymbol{x}_i) - \log(1 + \exp(\eta_{\boldsymbol{\theta}}(A\boldsymbol{x}_i))))$. The denominator in (4.8) is free of $\boldsymbol{\theta}_A$.

4.3 Variational Bayes model averaging for pooling multiple instances of random projection.

4.3.1 Bayesian model averaging

Ensemble learning methods are most widely used in machine learning literature to pool across varying classifiers to solve given problem a [32]. In this section, we address the same problem from a Bayesian perspective. Let $\mathcal{A}$ denote a subspace of the space of all random matrices. We assume that each RP matrix induces a separate model M_A, $A \in \mathcal{A}$ on the data $\mathcal{D} = (y_i, \boldsymbol{x}_i)_{i=1}^n$. Thus, the predictive distribution of a new observation y_{n+1} given $\boldsymbol{x}_{n+1}$ is

$$p(y_{n+1}|\boldsymbol{x}_{n+1}, \mathcal{D}) = \int p(y_{n+1}|\boldsymbol{x}_{n+1}, M_A, \boldsymbol{\theta}_A, \mathcal{D})\pi(M_A, \boldsymbol{\theta}_A|\mathcal{D})d\mu(M_A, \boldsymbol{\theta}_A) \tag{4.9}$$

where μ is the product measure of counting and Lebesgue measure. Note, that in the implementation of (4.9), the most difficult quantity to compute is $\pi(M_A, \boldsymbol{\theta}_A|\mathcal{D})$. In [52], explicit forms could be obtained for linear regression model, something which is next to impossible for convoluted neural network structure. In the next section, we circumvent this issue using variational inference.

4.3.2 ELBO derivation

Let $\pi(M_A, \boldsymbol{\theta}_A|\mathcal{D})$ denote the joint density of the parameter and the model conditional on the data. We posit a variational distribution $q(M_A, \boldsymbol{\theta}_A)$ of the form $q(\boldsymbol{\theta}_A|M_A) \sim \text{MVN}(\boldsymbol{m}_A, \text{diag}(\boldsymbol{s}_A))$ where $\boldsymbol{s}_A$ is a diagonal matrix and $q(M_A)$ are weights for the individual model. Thus, our variational family may be expressed as

$$Q_n = \left\{ q(M_A, \boldsymbol{\theta}_A) = q(M_A)q(\boldsymbol{\theta}_A|M_A) = \frac{q(M_A)(2\pi)^{-K_A/2}}{|\text{diag}(\boldsymbol{s}_A)|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{m}_A)^\top(\text{diag}(\boldsymbol{s}_A))^{-1}(\boldsymbol{\theta}-\boldsymbol{m}_A)} \right\}$$

The optimal variational distribution minimizes the Kullback-Leibler distance between $\pi(.|\mathcal{D})$ and the variational family Q_n. Thus, $q^* = \underset{q \in Q_n}{\text{argmin}}\, d_{\text{KL}}(q, \pi(.|\mathcal{D}))$, where

$$\begin{aligned} d_{\text{KL}}(q, \pi(.|\mathcal{D})) &= E_Q(\log \pi(\boldsymbol{\theta}_A, M_A|\mathcal{D}) - \log q(\boldsymbol{\theta}_A, M_A)) \\ &= -\log \pi(\mathcal{D}) + \text{ELBO} \end{aligned}$$

where ELBO $= E_Q(\log \pi(\boldsymbol{\theta}_A, M_A, \mathcal{D}) - \log q(\theta_A, M_A))$. Since $-\log \pi(\mathcal{D})$ is independent of $\boldsymbol{\theta}_A$ and M_A, therefore $q^*(M_A, \boldsymbol{\theta}_A) = \underset{q \in Q_n}{\text{argmin}}\, \text{ELBO}(q, \pi(.|\mathcal{D}))$. We next simplify the ELBO as

$$\begin{aligned}
&E_Q(\log \pi(\boldsymbol{\theta}_A, M_A, \mathcal{D}) - \log q(\boldsymbol{\theta}_A, M_A)) \\
&= \sum_{A \in \mathcal{A}} q(M_A) E_{Q(\boldsymbol{\theta}_A|M_A)}(\log \pi(\mathcal{D}|M_A, \boldsymbol{\theta}_A) + \log \pi(\boldsymbol{\theta}_A|M_A) \\
&\quad + \log \pi(M_A) - \log q(\boldsymbol{\theta}_A|M_A) - \log q(M_A)) \\
&= \sum_{A} q(M_A) E_{Q(\boldsymbol{\theta}_A|M_A)}(\log L(\boldsymbol{\theta}_A|M_A) + \log p(\boldsymbol{\theta}_A|M_A) \\
&\quad + \log \pi(M_A) - \log q(\boldsymbol{\theta}_A|M_A) - \log q(M_A)) \\
&= \sum_{A} q(M_A)(\mathcal{L}(.|M_A) + \log \pi(M_A) - \log q(M_A))
\end{aligned}$$

where $\mathcal{L}(.|M_A) = E_{Q(\boldsymbol{\theta}_A|M_A)}(\log L(\boldsymbol{\theta}_A|M_A) + \log p(\boldsymbol{\theta}_A|M_A) - \log q(\boldsymbol{\theta}_A|M_A))$ is nothing but the ELBO under the model M_A. Note, that the derivative of the ELBO with respect to variational parameters $\boldsymbol{m}_A, \boldsymbol{s}_A$ is given by

$$\nabla_{\boldsymbol{m}_A, \boldsymbol{s}_A} \text{ELBO} = q(M_A) \nabla_{\boldsymbol{m}_A, \boldsymbol{s}_A} \mathcal{L}(.|M_A)$$

Since $q(M_A)$ is just constant, thus the gradient update for model specific variational parameters is nothing but the gradient update from each individual model. Also, equating the derivative of ELBO with respect to $q(M_A)$ to zero, we get

$$\begin{aligned}
&\nabla_{q(M_A)} \text{ELBO} = 0 \\
&\implies \log \pi(M_A) - \log q(M_A) + \mathcal{L}(.|M_A) - 1 = 0 \\
&\implies q(M_A) \propto \exp(\log \pi(M_A) + \mathcal{L}(.|M_A))
\end{aligned}$$

Thus, the optimal model weights are $q^*(M_A) = \exp(\log \pi(M_A) + \mathcal{L}^*(.|M_A)) / \sum_A \exp(\log \pi(M_A) + \mathcal{L}^*(.|M_A))$ where $\mathcal{L}^*(.|M_A)$ are is the optimal ELBO under models A.

Note, that the main advantage of the above derivation is that the models can be individually trained in a parallel fashion and the final model weights depend only on the final ELBO values from each model.

4.4 Intrinsic dimensionality and prediction

4.4.1 Optimal dimension neighborhood selection

Let $d_A \times p$ denote the dimension of a RP matrix $A \in \mathcal{A}$. Using section 4.3, one can obtain the posterior model weights $q^*(M_A)$. The values of d_A with largest values of the posterior model weights $q(M_A)$ tend to concentrate around the optimal dimension of the feature space.

Let $d_1 \leq d_2 \leq \cdots$ be an enumeration of the unique values of d_A, $A \in \mathcal{M}_A$. Define the average posterior probability of each dimension value d_i as

$$q_i^* = \frac{1}{|\mathcal{A}^i|} \sum_{A \in \mathcal{A}^i} q(M_A)$$

where $\mathcal{A}^i = \{A \in \mathcal{A} : d_A = d_i\}$. The plot of (i, q_i^*) attains its peak around optimal dimension of feature space for prediction of the response. Let $d^* = \underset{i}{\text{argmax}}\, q_i^*$, then for some $v_1, v_2 > 0$,

$$\mathcal{I}_{d^*} = [\lfloor d^*(1 - v_1) \rfloor, \lceil d^*(1 + v_2) \rceil] \tag{4.10}$$

is the optimal dimension neighborhood which is used for the final classification task. Finally let $\mathcal{A}_{d^*}$ be a subspace of RP matrices with dimension $d_A \times p$ where $d_A \in \mathcal{I}_{d^*}$.

4.4.2 Classification based on optimal neigborhood choice

Using section 4.3, obtain the variational distribution $q^*(M_A, \boldsymbol{\theta}_A)$ for every $A \in \mathcal{A}_{d^*}$. Let $\widehat{\eta}_A = \int \eta_{\boldsymbol{\theta}_A}(A\boldsymbol{x}_{n+1}) q^*(\boldsymbol{\theta}_A | M_A) d\theta_A$ be the variational Bayes estimator of under model M_A. Define

$$\widehat{\eta}(\boldsymbol{x}_{n+1}) = \sum_{A \in \mathcal{A}_{d^*}} q^*(M_A) \widehat{\eta}_A(\boldsymbol{x}_{n+1}) \tag{4.11}$$

Based on $\widehat{\eta}(\boldsymbol{x}_{n+1})$ define the classification rule as

$$\widehat{y}_{n+1} = \boldsymbol{I}[\widehat{\eta}(\boldsymbol{x}_{n+1}) \geq 0] \tag{4.12}$$

Remark: *Note, the proposed estimator $\hat{\eta}(\boldsymbol{x}_{n+1})$ is not the exact variational Bayes estimator of $\eta(\boldsymbol{x}_{n+1}) = \log(P(y_{n+1} = 1 | \boldsymbol{x}_{n+1}) / P(y_{n+1} = 0 | \boldsymbol{x}_{n+1}))$. However, it is a good enough approximator for sufficiently large training size and computationally way faster, especially when the number of models and test samples are large.*

4.5 Algorithm and its implementation.

Algorithm 3 RPVBNN

```
1. Initialization: $(\boldsymbol{m}_A^0, \boldsymbol{s}_A^0, \rho_A^{\{t\}})_{A\in\mathcal{A}}$ where $\rho_A^{\{t\}}$, $t \geq 0$ is step size sequence for model $A$.
2. Parallelization :
    a) Set $t = 1$,
    b) For $A \in \mathcal{A}$, calculate the gradient of $\widehat{\mathcal{L}}(.|M_A)$ in (4.14) with respect to $\boldsymbol{m}_A$ and $\boldsymbol{s}_A$.
    c) Update the parameters $\boldsymbol{m}_A^t$ and $\boldsymbol{s}_A^t$ as
        $$\begin{aligned} \boldsymbol{m}_A^{t+1} &= \boldsymbol{m}_A^t + \rho_A^t \nabla_{\boldsymbol{m}_A} \widehat{\mathcal{L}}(.|M_A)|_{\boldsymbol{m}_A=\boldsymbol{m}_A^t} \\ \boldsymbol{s}_A^{t+1} &= \boldsymbol{s}_A^t + \rho_A^t \nabla_{\boldsymbol{s}_A} \widehat{\mathcal{L}}(.|M_A)|_{\boldsymbol{s}_A=\boldsymbol{s}_A^t} \end{aligned}$$
    d) Set $t = t + 1$.
    e) Repeat steps (b)-(d) till convergence.
3. Model averaging:
    a) For the optimized values $(\boldsymbol{m}_A^*, \boldsymbol{s}_A^*)_{A\in\mathcal{A}}$, compute $(\widehat{\mathcal{L}}^*(.|M_A))_{A\in\mathcal{A}}$ using (4.14).
    b) Compute the model weights
        $$q^*(M_A) = \frac{\exp(\log \pi(M_A) + \widehat{\mathcal{L}}^*(.|M_A))}{\sum_{A\in\mathcal{A}} \exp(\log \pi(M_A) + \widehat{\mathcal{L}}^*(.|M_A))}$$
4. Optimal neighborhood selection: Using the values $(q^*(M_A))_{A\in\mathcal{A}}$ compute
    a) The optimal neighborhood $\mathcal{I}_{d^*}$ as in (4.10).
    b) The subspace $\mathcal{A}_{d^*}$ using based on $\mathcal{I}_{d^*}$ (see section 4.4.1).
5. Classification:
    a) Repeat steps (1)-(3) for $A \in \mathcal{A}_{d^*}$.
    b) Compute $\widehat{\eta}(\boldsymbol{x}_{n+1})$ and $\widehat{y}_{n+1}$ using relations (4.11) and (4.12) respectively.
```

Gradient update equations. For $q(\boldsymbol{\theta}_A|M_A) = \prod_{j=1}^{K_A}(1/(2\pi s_{Aj}^2)^{1/2})e^{-(\theta_{Aj}-m_{Aj})^2/(2s_{Aj}^2)}$ and $p(\boldsymbol{\theta}_A|M_A) = \prod_{j=1}^{k}(1/(2\pi\sigma_{Aj}^2)^{1/2})e^{-(\theta_{Aj}-\mu_{Aj})^2/(2\sigma_{Aj}^2)}$, the ELBO is

$$\begin{aligned}\mathcal{L}(.|M_A) &= E_{Q(\boldsymbol{\theta}_A|M_A)}(\log L(\boldsymbol{\theta}_A|M_A)) \\ &\quad - d_{\mathrm{KL}}(q(.|M_A), p(.|M_A)))\end{aligned}$$

where $d_{\mathrm{KL}}(q(.|M_A), p(.|M_A)))$ is given by

$$\sum_{j=1}^{K_A}\left(\log\frac{\sigma_{Aj}}{s_{Aj}} + \frac{s_{Aj}^2}{2\sigma_{Aj}^2} + \frac{(m_{Aj}-\mu_{Aj})^2}{2\sigma_{Aj}^2} - \frac{1}{2}\right) \tag{4.13}$$

Since $E_Q(\log L(\boldsymbol{\theta}_A|M_A))$ cannot be computed explicitly, generate W samples $\boldsymbol{\theta}_A[1], \cdots, \boldsymbol{\theta}_A[W]$ from $Q(\boldsymbol{\theta}_A|M_A)$ and compute $\widehat{L}(.|M_A) = (1/W)\sum_{w=1}^{W}\log L(\boldsymbol{\theta}_A[w]|M_A)$. Thus, the final gradient function which is optimized w.r.t. to the parameters $\boldsymbol{m}_A$ and $\boldsymbol{s}_A$ is given by

$$\widehat{\mathcal{L}}(.|M_A) = \widehat{L}(.|M_A) - d_{\mathrm{KL}}(q(.|M_A), p(.|M_A))) \tag{4.14}$$

Remark: *Since, we need the variance parameter s_{Aj} to be always positive, thus, we consider the reparametrization, $s_{Aj} = \log(1+e^{\tilde{s}_{Aj}})$ and update the parameters $\tilde{s}_{Aj}$ instead where $\nabla_{\tilde{s}_{Aj}}\widehat{\mathcal{L}}(.|M_A) = e^{\tilde{s}_{Aj}}/(1+e^{\tilde{s}_{Aj}})\nabla_{s_{Aj}}\mathcal{L}(.|M_A)|_{\log(1+e^{\tilde{s}_{Aj}})}$ where $\nabla_{s_{Aj}}\widehat{\mathcal{L}}(.|M_A)|_{\log(1+e^{\tilde{s}_{Aj}})}$ is the derivative of $\widehat{\mathcal{L}}(.|M_A)$ with respect to s_{Aj} evaluated at $s_{Aj} = \log(1+e^{\tilde{s}_{Aj}})$.*

4.6 Theoretical results.

In this section, we study the convergence properties of the variational posterior for a given projection matrix (without model averaging). The results presented here are similar in spirit to the notion of posterior consistency in [52].

Let $f_0(y, \boldsymbol{x})$ and $f_{\boldsymbol{\theta}}(y, \boldsymbol{x})$ be the joint density of the data $\mathcal{D} = (y_i, \boldsymbol{x}_i)_{i=1}^n$ under the truth and the model respectively. Without loss of generality, we assume $\boldsymbol{x}_i \sim U[0,1]^p$, which implies $f_0(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = 1$. This implies that the joint distribution of $(y_i, \boldsymbol{x}_i)_{i=1}^n$ depends only the conditional distribution of $Y|X = \boldsymbol{x}$. Thus, under the model indexed by the projection matrix A,

$$\begin{aligned}f_{\boldsymbol{\theta}}(y, \boldsymbol{x}) &= f_{\boldsymbol{\theta}}(y|\boldsymbol{x})f(\boldsymbol{x})f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \ell_{\boldsymbol{\theta}}(y, A\boldsymbol{x}) \\ f_0(y, \boldsymbol{x}) &= f_0(y|\boldsymbol{x})f_0(A\boldsymbol{x}) = \ell_0(y, \boldsymbol{x})\end{aligned} \tag{4.15}$$

where $\ell_\theta(yb, A\boldsymbol{x}) = \exp(y\eta_{\boldsymbol{\theta}}(A\boldsymbol{x}) - \log(1+\exp(\eta_{\boldsymbol{\theta}}(A\boldsymbol{x}))))$ and $\ell_0(y,\boldsymbol{x}) = \exp(y\eta_0(\boldsymbol{x}) - \log(1+\exp(\eta_0(\boldsymbol{x}))))$ are defined respectively. We next define the Hellinger neighborhood of the true function density function $f_0 = \ell_0$ as

$$\begin{aligned} \mathcal{U}_\varepsilon &= \{\boldsymbol{\theta} : d_{\mathrm{H}}(\ell_0, \ell_{\boldsymbol{\theta}}) > \varepsilon\} \\ 2d_{\mathrm{H}}^2(\ell_0, \ell_{\boldsymbol{\theta}}) &= \int_{\boldsymbol{x}} \sum_y \left(\sqrt{\ell_0(y,\boldsymbol{x})} - \sqrt{\ell_{\boldsymbol{\theta}}(y, A\boldsymbol{x})}\right)^2 d\boldsymbol{x}. \end{aligned} \tag{4.16}$$

We next give the set of conditions which ensure that the variational posterior for a given projection matrix A, is consistent to the true density function f_0. Recall p is the total number of covariates, k is the number of nodes. If the dimension of A is $d_A \times p$, then the total number of parameters in the model indexed by A is $K_A = 1 + k + k(d_A + 1)$, i.e. $\boldsymbol{\theta}$ is $K_A \times 1$ vector.

Let $\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x}) = \beta_0^* + \sum_{j=1}^k \beta_j^* \psi(\boldsymbol{\gamma}_j^{*\top}\boldsymbol{x})$ be the neural network which can approximate the true function $\eta_0(\boldsymbol{x})$ in L_∞ norm. The existence of such a neural network is guaraneteed by [60]. Suppose A is an orthonormal projection matrix, prior $p(\boldsymbol{\theta}_A) = \mathrm{MVN}(\boldsymbol{\mu}_A, \mathrm{diag}(\boldsymbol{\sigma}_A))$, $n\epsilon_n^2 \to \infty$ and the following conditions hold:

1. (C1): $k_A \log n = o(n\epsilon_n^2)$, $p = o(e^{n\epsilon_n^2})$.

2. (C2): $||\boldsymbol{\mu}_\beta||_1^2 = o(n\epsilon_n^2)$, $\log ||\boldsymbol{\sigma}_\beta||_\infty = O(\log n)$, $||\boldsymbol{\sigma}_\beta^{-1}||_\infty = O(1)$, $\sum_{j=1}^k ||A^\top \boldsymbol{\mu}_{j\gamma}||_1 = O(1)$, $\sup_{j=1,\cdots,k} \log ||\boldsymbol{\sigma}_{j\gamma}||_\infty = O(\log n)$, $\sup_{j=1,\cdots,k} ||\boldsymbol{\sigma}_{j\gamma}^{-1}||_\infty = O(1)$.

3. (C3): $||\eta_0 - \eta_{\boldsymbol{\theta}^*}||_\infty = o(\epsilon_n^2)$, $||\boldsymbol{\beta}^*||_1^2 = o(n\epsilon_n^2)$, $\sup_{j=1,\cdots,k} ||(I - A^\top A)\boldsymbol{\gamma}_j^*||_1 = o(n^{-1})$, $\sum_{j=1}^k ||\boldsymbol{\gamma}_j^*||_1^2 = O(1)$.

4. (C4): $\log ||A\boldsymbol{x}|| = O(\log n)$, $1/||A\boldsymbol{x}|| = o(n\epsilon_n^2)$

Condition (C1) gives restrictions on the number of effective parameters ($\sim kd_A$) and the true number of covariates ($\sim p$). Condition (C2) puts restrictions on the growth of the prior parameters. Note, although the condition $\sum_{j=1}^k ||A^\top \boldsymbol{\mu}_{j\gamma}||_1 = O(1)$ seems to depend on the matrix A, it can be easily ensured by setting $\boldsymbol{\mu}_{j\gamma} = \mathbf{0}$. Condition (C3) quantifies how fast the neural network solution converges to the true function while keeping their coefficients magnitude under control. Although, the condition $\sup_{j=1,\cdots,k} ||(I - A^\top A)\gamma_j^*||_1 = o(n^{-1})$ is restrictive, it holds for any γ_j^* in the column space of the projection matrix A. Condition (C4) for projection matrices relates to condition (iii) in Theorem 3.1 of [52].

For the posterior in (4.8), let the variational posterior be

$$q_A^* = \underset{q \in \mathcal{Q}_n^A}{\text{argmin}}\ \text{ELBO}\ (q, \pi(.|\mathcal{D}, M_A))$$

where $\mathcal{Q}_n^A = \{q(\boldsymbol{\theta}_A) = \text{MVN}(\boldsymbol{m}_A, \text{diag}(\boldsymbol{s}_A))\}$. For a fixed A, one can obtain q_A^* by following the step 2. in algorithm 3.

Theorem: *Suppose $n\epsilon_n^2 \to \infty$ and conditions (C1)-(C4) hold, then for any $\varepsilon > 0$,*

$$q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \overset{P_0^n}{\to} 0$$

where $P_0^{(n)}$ is the joint distribution of $(y_i, \boldsymbol{x}_i)_{i=1}^n$ under (4.1).

The proof has been presented in the supplement section.

The above proof shows that the variational posterior q_A^* concentrates around shrinking Hellinger neighborhoods of the true function f_0 with overwhelming probability.

4.7 Numerical Study

4.7.1 Problem setup

We mimic the RP generation mechanism as in section 4.2.2. We fix the value of a_* at 0.3 for this whole section. We also experimented with the RP mechanism in [14] where the A is taken to the matrix of left singular vectors in the eigenvalue decomposition of $\tilde{A}$ where $\tilde{A}$ has all entries drawn from $N(0, 1)$ distribution. However we omit the results since no significant improvement was observed. Further, the Algorithm 3 is also sensitive to the choice of the learning rate ρ, the number of projections and the batch size. In this thesis, we do not explore the sensitivity with respect to these parameters due to their small impact. For the number of projection matrices, we followed the 2-power rule as: let the range of the project dimension d is $[p_1, p_2]$, the number of projections is chosen as $N = u(\min\{2^i : 2^i - (p_2 - p_1) > 0\})$. This ensures that approximately

u number of d values are chosen from any unit sub-interval of $[p_1, p_2]$. We employ parallel programming technique across different projection to reduce computational time. We first learn the optimal dimension of the data and then use it to improve the predictive accuracy. We analyze the performance of the algorithm in light of four data sets, two simulated datasets generated using a non linear function in the input feature space and two real data set obtained from neuroimaging and computer vision studies.

4.7.2 Datasets

We consider four cases of the data sets. The details of each data are summarized in Table 4.1. In the first two cases, we use the non-linear system [80] to generate observations with varying number of features. For these two data sets, the intrinsic dimensionality of the feature space is defined by number of active variables used in the data generation. We employ our algorithm to validate if it can recover the intrinsic dimensionality of the feature space and provide desirable classification accuracy. For the remaining two data sets, the intrinsic dimensionality is unknown. In the third case, we use the ADNI dataset from neuroimaging studies. In the last case, we use the MNIST dataset from computer vision studies. The proposed algorithm allows us to learn the intrinsic dimensionality of the feature space in both neuroimaging and computer vision applications.

For implementation, all input variables are $z-$ normalized. We first employ RPVBNN to learn the intrinsic dimensionality of the dataset. With a knowledge on the optimal dimensionality, we compare RPVBNN with several other traditional algorithms (logistic regression, random forest and gradient boosting) and VBNN which is the standard variational Bayes neural network based on the whole feature space. For the simulated datasets and ADNI, to prevent over-fitting and optimistically-biased estimates of model performance, we consider 10 different splits of the data into train and test datasets. We report the mean and standard deviation of train and test accuracy and AUC score over the 10 splits. For the MNIST data, since author-defined splits exist [73], we only report the train and test accuracy and algorithm run time.

4.7.3 Simulated data

We generate two simulated data from

$$y = \begin{cases} 1 \text{ , if } e^{x_1^2} + x_2^2 + 5\sin(x_3 x_4) - 3 > 0 \\ 0 \text{ , otherwise} \end{cases} \tag{4.17}$$

Since the number of active variables is 4, the intrinsic dimensionality of feature space is 4. To test if RPVBNN can capture the intrinsic dimensionality, we consider two simulated data examples 1) small simulated data 2) large simulated data. For small simulated data, we work with a smaller number of covariates ($p = 20$) and for the large simulated data, larger number of covariates are used ($p = 200$). Note, for both datasets $n = 3000$ observations are generated from (4.17). However, $\boldsymbol{x} \sim \text{MVN}(0, \Sigma)$ with $\sigma_{ii} = 0.5$ and $\sigma_{ij} = 0.25$ and has dimension $p = 20$ and $p = 200$ under the small and large datasets respectively. The large simulated data exemplifies the small n large p problem. For the 10 splits of cross validation, the ratio of observations in the training and test is 7:3, thus the train and test data sets have 2100 and 900 subjects respectively.

4.7.4 ADNI Data

We utilized the data provided by Alzheimer's disease Neuroimaging Initiative (ADNI) database `http://www.loni.ucla.edu/ADNI`. ADNI is an ongoing joint public-private effort to utilize neuroimaging, other biological markers, and clinical and neuropsychological assessment to measure the incidence and progression of MCI to early AD. The data used consisted of 819 subjects with baseline characteristics, genetics and diagnosis of MCI. For consistency, we only consider patients whose follow-up period was at least 36 months and no missing values. The final samples included $n = 265$ subjects which included participants who were stable in their diagnosis (MCI-S) and those who converted to a diagnosis of AD over 3 years (MCI-C). We used $p = 277$ variables which included diagnosis, neuropsychological tests score, epsilon-4 allele of the apolipoprotein E (APOE) gene and ROIs levels features derived from T1 magnetic resonance imaging (MRI). Analogous to the simulated examples, the ratio of subjects in the training and testing was 7:3 for the 10 splits.

Table 4.1: Summary of data,where n, p and c denote the numbers of samples, features and classes.

Data	Source	n	p	c
Small Simulated data	[80]	3000	20	2
Large Simulated data	[80]	3000	200	2
ADNI	[62]	264	278	2
MNIST	[73]	70000	784	10

4.7.5 MNIST Data

In addition to the ADNI, we evaluate the model performance on computer vision data - MNIST. The MNIST dataset is a large collection of handwritten digits and from the National Institute of Standards and Technology (NIST). MNIST dataset contains $n = 70000$ images with 60000 and 10000 in train and test sets respectively and a feature space of dimension $p = 784$ [73].

4.8 Results

4.8.1 Optimal dimensional region

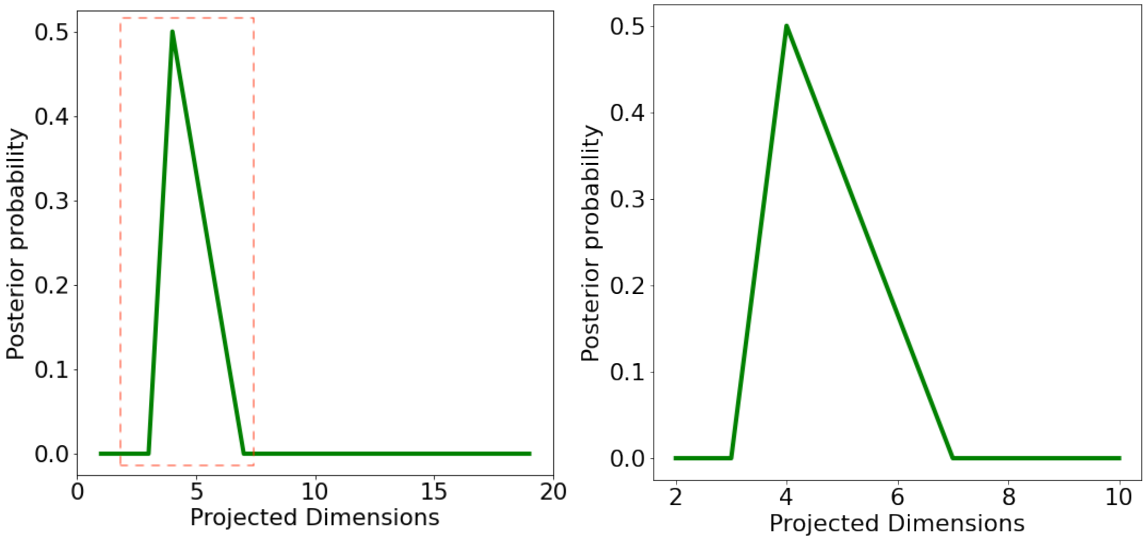


Figure 4.1: Small simulated data: $N = 32$

Using section 4.4.1, we obtain the average posterior probabilities of the projected dimensions. For all the four data sets, for learning the intrinsic dimension we employ RPVBNN with $k = 32$ nodes. Since obtaining the optimal dimension is a preprocessing step, we avoided experimentation with number of nodes in this step. Figures 4.1, 4.2, 4.3 and 4.4 give the average probability density

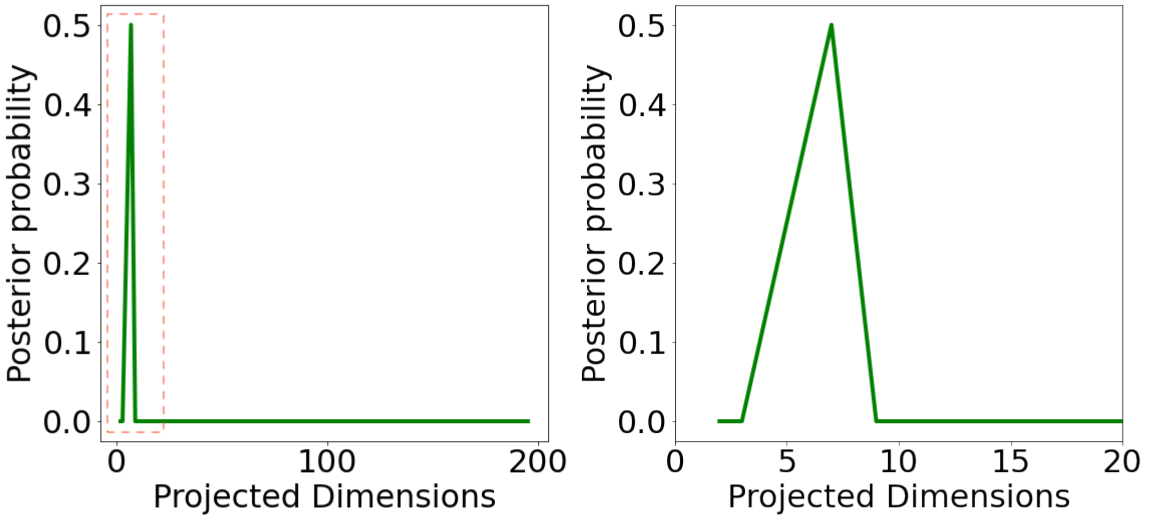


Figure 4.2: Large simulated data: $N = 128$

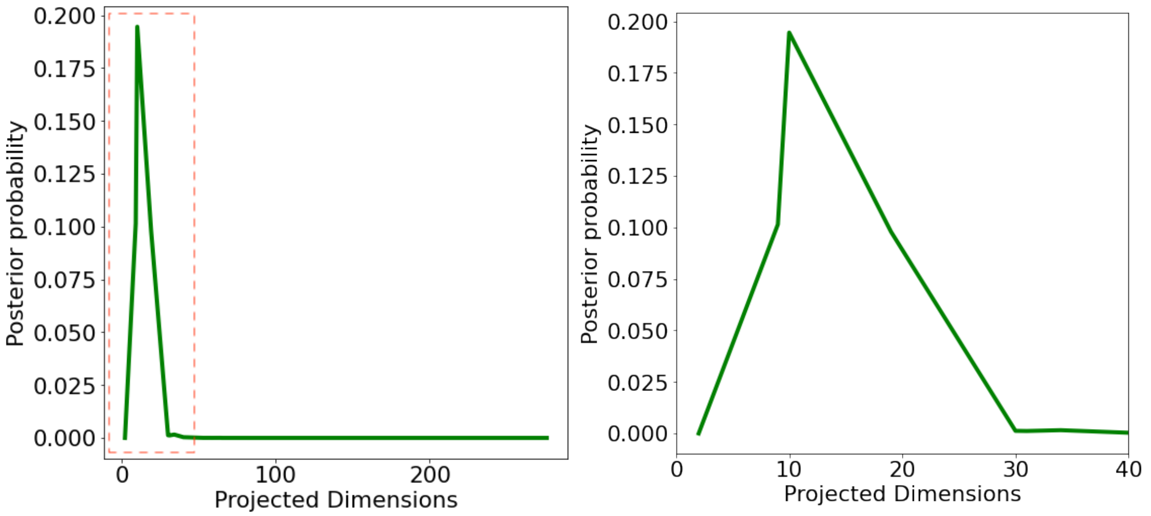


Figure 4.3: ADNI data: $N = 128$

curve as a function of the projected dimensions for small and large simulated data sets and ADNI and MNIST respectively. The intrinsic dimensionality estimate corresponds to the mode of this density curve while the optimal dimension neighborhood is a small interval around this posterior mode. For small simulated data, Figure 4.1 shows a dramatic growth in the average posterior probability as the number of projected dimensions increase from 3 to 5 with a significant drop when the number of projected features reach 7, followed by stabilization after 8. Thus, with a peak around 5, the optimal dimension neighborhood for small data is taken (3,7). For large data,

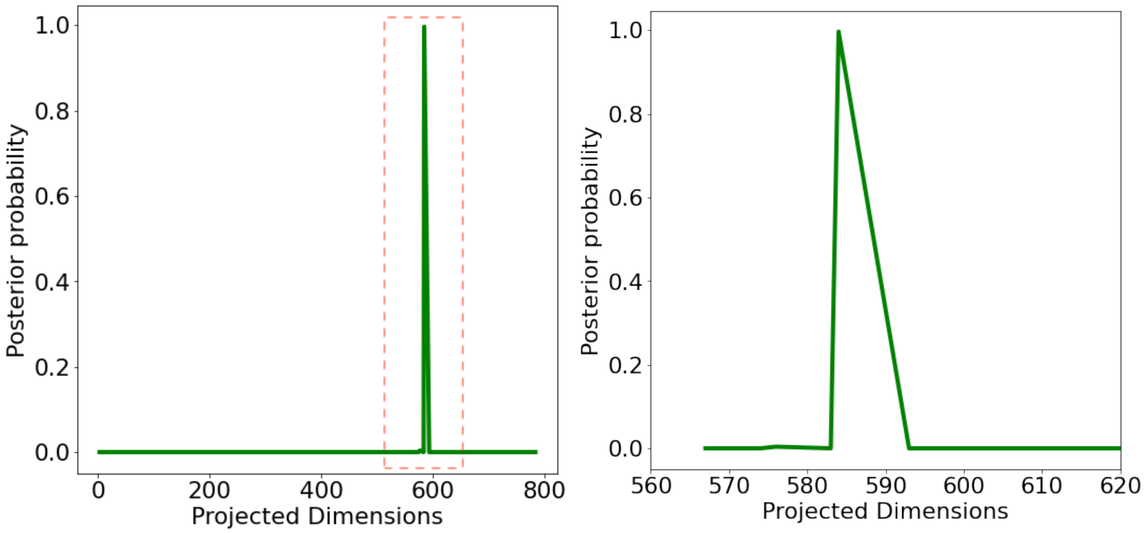


Figure 4.4: MNIST data: $N = 128$

Figure 4.2 shows that the average posterior probability peaks between 3 and 8 and stabilizes after 10. Thus, the optimal neighborhood for large data was taken to be (3,10). Note, for both small and large simulated datasets, the true intrinsic dimensionality was 4. The fact that the average posterior probability concentrates around 4 further corroborates that our algorithm learns well the intrinsic dimensionality of the feature space in regards to the prediction of response. Next, for ADNI data, Figure 4.3 shows that average posterior probability peaks between projected dimensions of 10 to 20 followed by stabilization after 30. Thus, the optimal dimensional neighborhood for ADNI data was chosen as (1,30). Finally, for MNIST data, Figure 4.4 shows that the optimal dimension neighborhood can be chosen as (580,600).

4.8.2 Comparative Baselines

For the two simulated examples and ADNI data, we consider 10 splits of the data as in section 4.7. With the optimal dimension neighborhood we use steps (4)-(5) of algorithm 3 to obtain the mean and standard deviation of train and test accuracy and AUC score of RPVBNN (see tables 4.3, 4.5 and 4.4 respectively). In addition to the performance of RPVBNN, we also provide results from logistic regression with L_1 penalty (LR-L_1), random forest (RF), gradient boosting (GB) as comparative baseline. In particular, we report the LR performance for varying values of

Table 4.2: RPVBNN setting for evaluation

Data	N	Learning rate	Batch size	Optimal Region
Small data	16	0.01	256	(3,7)
Large data	64	0.01	256	(3,10)
ADNI	64	0.01	185	(1,30)
MNIST	128	0.01	512	(580,600)

$\lambda = 0.01, 0.1, 0.5, 1, 5, 10, 100$ together with the performance at λ_0, the optimum λ obtained from 10-fold cross validation. To build both RF and GB models, we start with 50 trees and increase the number of trees by 100 trees each time until we see either no improvement of test accuracy or increase in the standard deviation of test accuracy [101]. Finally, for the same 10 splits, we report the results obtained using VBNN algorithm which works on the whole feature space without any compression (it is indeed a version of RPVBNN with $N = 1$, $d_A = p$ and $A = I$). The number of nodes for both RPVBNN and VBNN are varied as $k = 32, 64, 128$.

For the MNIST dataset, since user defined train and test splits already exist, we only report the train and test accuracy and algorithm run time for RPVBNN (see table 4.6). As a comparative baseline, we also provide the results of VBNN. For all the datasets, the details of RPVBNN settings including optimal dimension neighborhoods, the number of projections, learning rate and batch size are summarized in Table 4.2.

4.8.3 Experimental Results

For small simulated data, as shown in Table 4.3, the results of using RPVBNN with 128 hidden nodes can achieve a test accuracy and AUC of 94.88% and 95.88% respectively which is considerably better than performance of other learning algorithms. Also, the impact of the number of nodes is minimal which further justifies our attainment of optimal dimensionality neighborhood using only $k = 32$ nodes. Whereas for the small simulated dataset the second best performer was VBNN, its performance significantly deteriorates for the large simulated dataset. This is because the with a large feature space of $p = 200$, the training size of 2100 is way smaller. Since RPVBNN works with compressed feature space of $d_A \in [3, 10]$, it still has the best testing accuracy and AUC of 94.96%

and 96.63% respectively (see table 4.4). This clearly indicates that RPVBNN is an effective solution to the small n large p problem. Also, since at each instance one works with the compressed feature space, one gains a huge advantage in both memory storage and computational efficiency as long as the intrinsic dimensionality of feature space lies in a smaller dimensional subspace (although multiple compressions are needed, one can leverage parallelization across compressions).

The ADNI with $p = 277$ and training sample 180 is another example of a small n and large p problem. RPVBNN still continues (see table 4.5) to outperform all its competitors where VBNN suffers from the curse of dimensionality. Interestingly, overall gradient boosting seems to the second best performer after RPVBNN. For the MNIST data (see Table 4.2), note that VBNN with the best testing accuracy of 97.8% slightly outperforms the RPVBNN with the best test accuracy of 97.32. For MNIST, the training size $n = 60000$ is way larger than $p = 784$, is best performer. However, the average run time for one run based on 500 epochs using 128 nodes of VBNN is 2640 seconds while the same value with d_A in the optimal dimension neighborhood is 2350 seconds. To conclude, when $p >> n$, RPVBNN offers the biggest advantage in terms of memory storage, computational

Table 4.3: Table: Small simulated data performance

Model	Setting	Train Acc(%)	Test Acc(%)	AUC(%)
LR-l_1	$\lambda = 10$	67.63 ± 0.64	67.46 ± 1.04	68.04 ± 1.54
	$\lambda = 1$	67.59 ± 0.67	67.47 ± 0.97	68.07 ± 1.58
	$\lambda = 0.1$	67.32 ± 0.77	67.44 ± 0.97	68.47 ± 1.51
	$\lambda = 0.01$	65.49 ± 0.91	66.04 ± 1.36	68.77 ± 1.56
	$\lambda_0 = 0.1$	67.32 ± 0.77	67.44 ± 0.97	68.47 ± 1.51
RF	10 trees	66.83 ± 1.76	66.02 ± 1.84	73.76 ± 3.63
	25 trees	68.75 ± 1.91	67.92 ± 1.84	78.30 ± 3.67
	50 trees	68.90 ± 1.09	67.84 ± 1.65	80.07 ± 2.05
GB	10 trees	72.78 ± 0.78	72.05 ± 1.45	74.18 ± 1.58
	50 trees	79.78 ± 1.78	77.34 ± 1.73	88.74 ± 1.72
	100 trees	87.30 ± 1.52	83.22 ± 2.08	92.51 ± 0.88
	150 trees	90.81 ± 0.90	85.56 ± 2.01	93.78 ± 0.84
	250 trees	94.17 ± 0.81	86.91 ± 1.63	94.41 ± 0.66
	350 trees	94.07 ± 1.07	87.44 ± 1.98	94.62 ± 0.75
	450 trees	97.62 ± 0.62	87.80 ± 1.95	94.84 ± 0.86
VBNN	32 nodes	94.88 ± 0.52	89.84 ± 0.64	90.36 ± 0.71
	64 nodes	95.36 ± 0.38	90.27 ± 0.59	90.89 ± 0.53
	128 nodes	95.28 ± 0.45	90.28 ± 0.65	90.88 ± 0.56
RPVBNN	32 nodes	95.70 ± 0.40	94.77 ± 0.68	95.68 ± 0.56
	64 nodes	95.80 ± 0.42	94.80 ± 0.60	95.45 ± 0.64
	128 nodes	95.83 ± 0.31	94.88 ± 0.76	95.88 ± 0.43

Table 4.4: Table: Large simulated data performance

Model	Setting	Train Acc(%)	Test Acc(%)	AUC(%)
LR-l_1	$\lambda = 10$	71.34 ± 0.61	64.12 ± 1.58	65.88 ± 0.90
	$\lambda = 1$	71.39 ± 0.63	64.21 ± 1.34	66.13 ± 0.91
	$\lambda = 0.1$	70.90 ± 0.59	66.31 ± 1.35	68.13 ± 0.95
	$\lambda = 0.01$	65.75 ± 0.86	66.07 ± 1.09	70.62 ± 1.41
	$\lambda_0 = 0.01$	65.75 ± 0.86	66.07 ± 1.09	70.62 ± 1.41
RF	10 trees	62.78 ± 3.31	61.12 ± 3.70	66.21 ± 6.43
	25 trees	62.93 ± 3.93	61.69 ± 3.07	70.24 ± 3.43
	50 trees	60.04 ± 1.48	59.59 ± 1.58	71.82 ± 2.67
	100 trees	60.14 ± 1.61	59.54 ± 1.39	74.81 ± 2.36
GB	10 trees	73.51 ± 0.82	72.14 ± 1.78	74.43 ± 1.62
	50 trees	80.53 ± 1.30	77.45 ± 2.69	87.68 ± 3.15
	100 trees	87.79 ± 1.37	80.75 ± 2.36	90.69 ± 1.59
	150 trees	91.98 ± 1.06	82.25 ± 2.68	91.56 ± 1.57
	250 trees	96.31 ± 0.08	84.04 ± 2.99	92.41 ± 1.82
	350 trees	98.46 ± 0.05	84.64 ± 2.71	92.70 ± 1.71
	450 trees	99.40 ± 0.04	85.06 ± 2.55	92.98 ± 1.72
	550 trees	99.78 ± 0.01	84.97 ± 2.20	92.95 ± 1.70
VBNN	32 nodes	62.76 ± 1.21	60.88 ± 1.59	65.23 ± 1.78
	64 nodes	62.52 ± 1.71	60.90 ± 1.50	65.62 ± 1.39
	128 nodes	63.61 ± 1.13	61.42 ± 1.11	66.21 ± 1.06
RPVBNN	32 nodes	96.54 ± 0.22	94.70 ± 0.81	96.21 ± 0.45
	64 nodes	96.57 ± 0.41	94.89 ± 0.68	96.45 ± 0.63
	128 nodes	96.66 ± 0.28	94.96 ± 0.90	96.63 ± 0.41

Table 4.5: Table: ADNI data performance

Model	Setting	Train Acc(%)	Test Acc(%)	AUC(%)
LR-l_1	$\lambda = 10$	100.00 ± 0.00	65.75 ± 4.07	68.71 ± 3.75
	$\lambda = 1$	100.00 ± 0.00	63.25 ± 3.12	65.21 ± 4.44
	$\lambda = 0.1$	100.00 ± 0.00	61.00 ± 4.70	61.62 ± 4.00
	$\lambda = 0.01$	100.00 ± 0.00	60.12 ± 4.55	61.08 ± 4.15
	$\lambda_0 = 10$	100.00 ± 0.00	65.75 ± 4.07	68.71 ± 3.75
RF	10 trees	81.78 ± 2.23	68.87 ± 5.37	75.08 ± 5.18
	25 trees	82.38 ± 2.35	70.88 ± 4.43	79.29 ± 5.00
	50 trees	83.67 ± 1.79	72.00 ± 3.88	78.89 ± 4.90
	100 trees	83.08 ± 2.62	71.25 ± 4.50	79.95 ± 4.85
GB	10 trees	87.24 ± 1.51	73.25 ± 4.40	80.44 ± 4.23
	25 trees	95.41 ± 1.16	74.37 ± 5.34	81.32 ± 4.01
	50 trees	99.78 ± 0.35	74.87 ± 4.95	81.16 ± 4.17
	100 trees	100.00 ± 0.00	73.75 ± 3.95	80.79 ± 3.89
VBNN	32 nodes	62.51 ± 2.02	62.75 ± 4.67	62.54 ± 3.43
	64 nodes	62.51 ± 2.02	62.75 ± 4.67	62.54 ± 3.43
	128 nodes	62.51 ± 2.02	62.75 ± 4.67	62.54 ± 3.43
RPVBNN	32 nodes	78.57 ± 1.76	75.66 ± 3.80	81.88 ± 1.76
	64 nodes	78.62 ± 1.92	75.70 ± 4.85	82.12 ± 1.83
	128 nodes	78.84 ± 1.62	75.94 ± 3.84	82.33 ± 1.91

Table 4.6: MNIST data performance in term of testing accuracy and time (based on 500 epochs)

Model	Setting	Train Acc(%)	Test Acc(%)	Time(s)
VBNN	32 nodes	97.63	96.88	354
	128 nodes	98.08	97.33	758
	256 nodes	98.13	97.40	1385
	512 nodes	99.11	97.80	2640
RPVBNN	32 nodes	97.82	97.18	280
	128 nodes	97.84	97.29	720
	256 nodes	98.00	97.30	1143
	512 nodes	98.06	97.32	2350

efficiency and prediction accuracy in addition to the inference on the intrinstic dimensionality of the feature space. For $n >> p$, RPVBNN is equally competitive while still providing computational and memory gain as long as the input resides in a smaller dimensional subspace with respect to prediction.

4.9 Conclusion

In this chapter, we consider a variational Bayes neural network predictive model for addressing the curse of dimensionality (small n large p) by compressing the feature space using RP matrices. To remove the sensitivity to the choice of the RP matrix, we propose a model averaging approach to base our projection on the most relevant models. To improve computational complexity, we provide a variational inference technique which can estimate model specific parameters and model weights both at the same time. As a by-product, we use the posterior model weights of the projected dimensions to learn the intrinsic dimensionality of the feature space in context of prediction. The advantage of variational inference approach proposed in the context of Bayesian model averaging has two advantages (1) it has the computation gain of frequentist ensemble approaches since one can parallelize across different models (2) it provides the uncertainty quantification of associated with each random projection via posterior probabilities. The approach presented in this thesis can generalized to a wide class of problems arising out of Bayesian neural networks which require learning of the model importance or averaging across models.

CHAPTER 5

CONCLUSIONS, DISCUSSION, AND DIRECTIONS FOR FUTURE RESEARCH

5.1 Conclusions and discussion

In this thesis, we first applied two machine learning methods (LR and SVM) under multiple conditions, to test accuracy in classifying patients with MCI who progress to clinically-defined dementia (MCI-C) from those who remain stable (MCI-S). Using multi-modal data from ADNI, we compared LR and SVM classification accuracy and pre-selection dimensional reduction techniques - i.e., feature selection as informed by prior findings in clinical neuroscience and by L_1 norm. Notably, the present results demonstrate important boundaries for applying feature selection techniques in statistical classification of MCI-to-dementia conversion. Specifically, we found that while using L_1 for pre-selection can improve accuracy, it also benefits from a more limited, theoretically based set of feature inputs. In addition, we found that model performance benefited from a longer window of assessment. These results have implications for studies utilizing multi-modal data for such classification, including features from clinical neuropsychological assessment, demographic and genetic markers, MRI-based volumetric brain measures, and other modalities. This thesis also demonstrates that SVM classifier performance is more stable than LR for dealing with the “large p" problem. Clinical researchers should note the value of evaluating different classification and pre-selection approaches in application to clinical or research questions, and be mindful that not all machine learning techniques are equally beneficial for modeling specific clinical outcomes.

To further tackle the high dimensional data and variability and complexity of big data, we introduce the variational Bayes neural network and provide the theoretical rigour and computational detail for BDNNs. Although the variational Bayes is popular in machine learning, neither the computational method nor the statistical properties are well understood for complex modeling such as neural networks. We characterize the prior distributions and the variational family for consistent Bayesian estimation. The theory also quantifies the loss due to VB numerical approximation compared to the true posterior distribution. For practical implementation, we reveal that the algorithm may not be as simple and straightforward as it sounds in computer science literature, rather it requires careful crafting on several parameters associated in various steps. Nevertheless, the computation could be quite faster compared to popular Monte Carlo Markov Chain procedure of approximating the posterior distributions.

Even though BDNN has achieved higher model performnce in classifying the transtion from MCI to dementia, it fails to address the curse of dimensionality and learn the true dimensionlity of data. We then consider a variational Bayes neural network predictive model for addressing the curse of dimensionality (small n large p) by compressing the feature space using RP matrices. To remove the sensitivity to the choice of the RP matrix, we propose a model averaging approach to base our projection on the most relevant models. To improve computational complexity, we provide a variational inference technique which can estimate model specific parameters and model weights both at the same time. The derivation shows that use of variational inference provides a huge advantage by allowing parallelization across different models at hand. Unlike Markov Chain Monte Carlo, the variational technique proposed in this thesis allows to obtain optimal model weights after individual models have been trained, by just making model Evidence Lower Bound (ELBO) and prior model weights. The approach is generalizable to a wide class of problems where Bayesian model averaging is next to impossible due to the large dimension of the data or intractable likelihood.

5.2 Directions for future research

The future research is mainly focused on two aspects: choice of prior structure, Bayesian compressed deep neural network. Although this thesis builds the framework on a multi-layer neural networks model with simplistic prior structure, the detail statistical theory and computational methodology are quite involved. This investigation opens up possibility of exploring much wider class of models and priors. For example, shrinkage priors, such as double exponential and horseshoe priors can be explored for building sparse neural networks or one can experiment with various other variational families. However, their computational details and associated statistical properties are not immediate. We hope this research will accelerate further development of statistical and computational foundation for variational inference in general machine learning research.

Moreover, we explored the sensitiveness to the number of projections and dimension of the projection empirically. However, further investigation is needed in order to obtain a statistically optimal solution. Another interesting direction to pursue will be studying the impact of different projections and qualifying prediction accuracy as a function of the projection. This current work presents a proof of concept for shallow networks. However the methodology developed in this thesis can be extended to deep neural networks. Another interesting line of work will be extension to more complex feature spaces to learn the intrinsic dimensionality of these spaces.

APPENDICES

APPENDIX A

SUPPLEMENT FOR CONSISTENT VARIATIONAL BAYES CLASSIFICATION WITH DEEP NEURAL NETWORKS

Algorithms of variational implementation

Algorithm 4 BBVI-RMS

```
1. Fix an initial value for variational family parameters $\mathcal{V}_q^1$.
2. Fix a step size sequence $\rho_t, t = 1, \cdots$.
3. Set $t = 1$ and $\epsilon > 0$.
4. Simulate $W$ samples $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ from $q(.|\mathcal{V}_q^t)$.
5. Compute $\widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}}$ as in (3.18)
6. Compute
       $G_t = \widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}} - \nabla_{\mathcal{V}_q^t} d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.))$
       $R_t = 0.9R_{t-1} + 0.1G_t^2$
7. Update
       $\mathcal{V}_q^{t+1} = \mathcal{V}_q^t + \rho_t \frac{G_t}{\sqrt{R_t} + \epsilon}$        (A.1)
8. Set $t = t + 1$.
9. Repeat steps 4-7 until the convergence of ELBO using $\widehat{\mathcal{L}}_{\mathcal{V}_q^t}$ as in (3.19) and
       $\mathrm{ELBO} = \widehat{\mathcal{L}}_{\mathcal{V}_q^t} - d_{\mathrm{KL}}(q(.|\mathcal{V}_q), p(.))$
```

Algorithm 5 BBVI-CV-RMS

```
1. Fix an initial value for variational parameter $\mathcal{V}_q^1$.
2. Fix a step size sequence $\rho_t, t = 1, \cdots$.
3. Set $t = 1$.
4. Simulate $W$ samples $\boldsymbol{\theta}_n[1], \cdots, \boldsymbol{\theta}_n[W]$ from $q(.|\mathcal{V}_q^t)$.
5. Compute $c^{\star t} = \text{cov}(\boldsymbol{u}_1^t, \boldsymbol{u}_2^t)/\text{var}(\boldsymbol{u}_2^t)$ where $\boldsymbol{u}_1^t$ and $\boldsymbol{u}_2^t$ are same as in (3.22).
6. Compute $\widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}}$ as in (3.21).
7. Compute
       $G_t = \widehat{\nabla_{\mathcal{V}_q^t}\mathcal{L}_{\mathcal{V}_q^t}} - \nabla_{\mathcal{V}_q^t} d_{\text{KL}}(q(.|\mathcal{V}_q), p(.))$
       $R_t = 0.9R_{t-1} + 0.1G_t^2$
8. Update
       $\mathcal{V}_q^{t+1} = \mathcal{V}_q^t + \rho_t \frac{G_t}{\sqrt{R_t} + \epsilon}$
9. Set $t = t + 1$.
10. Repeat steps 4-7 until the convergence of ELBO using $\widehat{\mathcal{L}}_{\mathcal{V}_q^t}$ as in (3.19) and
       $\text{ELBO} = \widehat{\mathcal{L}}_{\mathcal{V}_q^t} - d_{\text{KL}}(q(.|\mathcal{V}_q), p(.))$
```

With q and p as in (3.11) and (3.9) respectively,

$$d_{\text{KL}}(q, p) = \sum_{j=1}^{K_n} \left(\log \frac{\sigma_{jn}}{s_{jn}} + \frac{s_{jn}^2}{2\sigma_{jn}^2} + \frac{(m_{jn} - \mu_{jn})^2}{2\sigma_{jn}^2} - \frac{1}{2} \right)$$

$$\nabla_{m_{jn}} d_{\text{KL}}(q, p) = \frac{(m_{jn} - \mu_{jn})}{2\sigma_{jn}^2} \qquad \nabla_{s_{jn}} d_{\text{KL}}(q, p) = -\frac{1}{s_{jn}} + \frac{s_{jn}}{\sigma_{jn}^2}$$

$$\nabla_{m_{jn}} \mathcal{L}_{\mathcal{V}_q} = E_{q(.|\mathcal{V}_q)} \left(\left(\frac{\theta_{jn} - m_{jn}}{s_{jn}^2} \right) \log L(\boldsymbol{\theta}_n) \right)$$

$$\nabla_{s_{jn}} \mathcal{L}_{\mathcal{V}_q} = E_{q(.|\mathcal{V}_q)} \left(\left(\frac{(\theta_{jn} - m_{jn})^2}{s_{jn}^3} - \frac{1}{s_{jn}} \right) \log L(\boldsymbol{\theta}_n) \right)$$

Preliminaries

A.0.1 Definitions

Definition A.0.1 *For a vector $\boldsymbol{\alpha}$ and a function g,*

1. $||\boldsymbol{\alpha}||_1 = \sum_i |\alpha_i|$, $||\boldsymbol{\alpha}||_2 = \sqrt{\sum_i \alpha_i^2}$, $||\boldsymbol{\alpha}||_\infty = \max_i |\alpha_i|$.

2. $||g||_1 = \int_{\boldsymbol{x}\in\chi} |g(\boldsymbol{x})|d\boldsymbol{x}$, $||g||_2 = \sqrt{\int_{\boldsymbol{x}\in\chi} g(\boldsymbol{x})^2 d\boldsymbol{x}}$, $||g||_\infty = \sup_{\boldsymbol{x}\in\chi} |g(\boldsymbol{x})|$

Definition A.0.2 (Bracketing number and entropy) *For any two functions l and u, define the bracket $[l, u]$ as the set of all functions f such that $l \leq f \leq u$. Let $||.||$ be a metric. Define an $\varepsilon-$bracket as a bracket with $||u - l|| \leq \varepsilon$. Define the bracketing number of a set of functions $\mathcal{F}^*$ as the minimum number of $\varepsilon-$brackets needed to cover $\mathcal{F}^*$, and denote it by $N_{[]}(\varepsilon, \mathcal{F}^*, ||.||)$. Finally, the Hellinger bracketing entropy, denoted by $H_{[]}(\varepsilon, \mathcal{F}^*, ||.||)$, is the natural logarithm of the bracketing number ([104]).*

Definition A.0.3 (Covering number and entropy) *Let $(V, ||.||)$ be a normed space, and $\mathcal{F} \subset V$. $\{V_1, \cdots, u_n\}$ is an $\varepsilon-$covering of $\mathcal{F}$ if $\mathcal{F} \subset \cup_{i=1}^N B(V_i, \varepsilon)$, or equivalently, $\forall\ \theta \in \mathcal{F}$, $\exists\ i$ such that $||\theta - V_i|| < \varepsilon$. The covering number of $\mathcal{F}$ denoted by $N(\varepsilon, \mathcal{F}, ||.||) = \min\{n : \exists\ \varepsilon -$ covering over $\mathcal{F}$ of size $n\}$. Finally, the Hellinger covering entropy, denoted by $H(\varepsilon, \mathcal{F}, ||.||)$, is the natural logarithm of the covering number ([104]).*

A.0.2 Lemmas

Lemma A.0.4 *With $H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)$ as in Definition A.0.2, for $H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2) \leq K_n \log(M_n/u)$,*

$$\int_0^\varepsilon H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)du \lesssim \varepsilon\sqrt{K_n(\log M_n - \log \varepsilon)}$$

Proof. See proof of lemma 7.14 in [6].

Lemma A.0.5 *Suppose q satisfies $\int d_{KL}(\ell_0, \ell_{\boldsymbol{\theta}_n})q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \leq \varepsilon$, then for any $\nu > 0$,*

$$P_0^n\left(\left|\int q(\boldsymbol{\theta}_n) \log \frac{L(\boldsymbol{\theta}_n)}{L_0} d\boldsymbol{\theta}_n\right| \geq n\nu\right) \leq \frac{\varepsilon}{\nu}$$

Proof. See proof of lemma 7.13 in [6].

Lemma A.0.6 *Suppose* $\mathcal{N}_\varepsilon = \{\boldsymbol{\theta}_n : d_{KL}(\ell_0, \ell_{\boldsymbol{\theta}_n}) < \varepsilon\}$ *and* $\int_{\mathcal{N}_\varepsilon} p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \geq e^{-n\varepsilon}, n \longrightarrow \infty$ *then for any* $\nu > 0$*,*

$$P_0^n\left(\left|\log\int \frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right| \geq n\nu\right) \leq \frac{2\varepsilon}{\nu}$$

Proof. See proof of lemma 7.12 in [6].

Lemma A.0.7 *Suppose,* $\int_{\mathcal{F}_n^c} p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \leq e^{-n\varepsilon}, n \longrightarrow \infty$ *for any* $\varepsilon > 0$*. Then, for every* $\tilde{\varepsilon} < \varepsilon$*.*

$$P_0^n\left(\int_{\boldsymbol{\theta}_n\in\mathcal{F}_n^c} \frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \geq e^{-n\tilde{\varepsilon}}\right) \leq e^{-n(\varepsilon-\tilde{\varepsilon})}$$

Proof. See proof of lemma 7.16 in [6].

Lemma A.0.8 *Let* $\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x}) = \boldsymbol{b}_L^* + \boldsymbol{A}_L^*\psi(\boldsymbol{b}_{L-1}^* + \boldsymbol{A}_{L-1}^*\psi(\cdots\psi(\boldsymbol{b}_1^* + \boldsymbol{A}_1^*\psi(\boldsymbol{b}_0^* + \boldsymbol{A}_0^*\boldsymbol{x})))$ *be a fixed neural network. Let* $\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = \boldsymbol{b}_L + \boldsymbol{A}_L\psi(\boldsymbol{b}_{L-1} + \boldsymbol{A}_{L-1}\psi(\cdots\psi(\boldsymbol{b}_1 + \boldsymbol{A}_1\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x})))$ *be a neural network such that*

$$|\theta_{jn} - \theta_{jn}^*| \leq \frac{\varepsilon}{\sum_{v=0}^{L}\tilde{k}_{vn}\prod_{v'=v+1}^{L_n}a_{v'n}^*}$$

where $\tilde{k}_{vn} = k_{vn} + 1$*. Then,*

$$\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})|dx \leq \varepsilon$$

Proof. In the proof, we suppress the dependence on n. Define the projection P_v as $P_V\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{b}_{V-1} + \boldsymbol{A}_{V-1}\psi(\cdots\psi(\boldsymbol{b}_1 + \boldsymbol{A}_1\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x})))$. We claim that

$$|P_V\eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s] - P_V\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]| \leq \frac{\varepsilon\sum_{v=0}^{V}\tilde{k}_v\prod_{v'=v+1}^{L}a_{v'}^*}{\sum_{v=0}^{L}\tilde{k}_v\prod_{v'=v+1}^{L}a_{v'}^*} \tag{A.2}$$

We prove this by induction. Let $v = 1$ as follows. Let $\tilde{\varepsilon} = \varepsilon/\sum_{v=0}^{L}\tilde{k}_v\prod_{v'=v+1}^{L}a_{v'}^*$, then

$$\begin{aligned}
&|P_1\eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s] - P_1\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]| \\
&\leq |\boldsymbol{b}_1 - \boldsymbol{b}_1^*[s]| + |\boldsymbol{A}_1[s]^\top\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x}) - \boldsymbol{A}_1^*[s]^\top\psi(\boldsymbol{b}_0^* + \boldsymbol{A}_0^*\boldsymbol{x})| \\
&\leq \tilde{\varepsilon} + ||\boldsymbol{A}_1[s] - \boldsymbol{A}_1^*[s]||_1 + \sum_{s'=0}^{k_1}|\boldsymbol{A}_1^*[s][s'](\psi(\boldsymbol{b}_0[s] + \boldsymbol{A}_0[s]^\top\boldsymbol{x}) - \psi(\boldsymbol{b}_0^*[s] + \boldsymbol{A}_0^*[s]^\top\boldsymbol{x}))| \\
&= \tilde{\varepsilon} + k_1\tilde{\varepsilon} + \tilde{\varepsilon}\sum_{s'=0}^{k_1}|\boldsymbol{A}_1^*[s][s']|(k_0+1) = \tilde{\varepsilon}(1 + k_1 + a_1^*(p_n+1)) \leq \tilde{\varepsilon}(\tilde{k}_1 + a_1^*\tilde{k}_0)
\end{aligned}$$

where the second line holds since $\psi(u) \leq 1$ and the third step is shown next. Let $u = -\boldsymbol{b}_0[s] - \boldsymbol{A}_0[s]^\top \boldsymbol{x}$ and $u_\delta = \boldsymbol{b}_0[s] + \boldsymbol{A}_0[s]^\top \boldsymbol{x} - \boldsymbol{b}_0^*[s] + \boldsymbol{A}_0^*[s]^\top \boldsymbol{x}$, then for $|u_\delta| < 1$

$$\begin{aligned} |\psi(u) - \psi(u + u_\delta)| &= \left| \frac{e^{u+u_\delta} - e^u}{(1 + e^{u+u_\delta})(1 + e^u)} \right| \leq \left| \frac{e^u(e^{u_\delta} - 1)}{(1 + e^u)(1 + e^{u+u_\delta})} \right| && \text{(A.3)} \\ &\leq \frac{e^u |e^{u_\delta} - 1|}{(1 + e^u)(1 + e^{u-1})} \leq |u_\delta| && \text{(A.4)} \end{aligned}$$

since $e^u/((1 + e^u)(1 + e^{u-1})) \leq 1/2$ and $|e^{u_\delta} - 1| \leq 2|u_\delta|$ for $|u_\delta| < 1$. Now, $|u_\delta| = |\boldsymbol{b}_0[s] - \boldsymbol{b}_0^*[s]| + \sum_{s'=0}^{p_n} |\boldsymbol{A}_0[s][s'] - \boldsymbol{A}_0^*[s][s']| \leq (p_n + 1)\tilde{\varepsilon} < 1$.

Suppose the result hold for $V - 1$, we show the result for V as follows:

$$\begin{aligned} &|P_V \eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s] - P_V \eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]| \\ &\leq |\boldsymbol{b}_V[s] - \boldsymbol{b}_V^*[s]| + |\boldsymbol{A}_V[s]^\top \psi(P_{V-1}\eta_{\boldsymbol{\theta}}(\boldsymbol{x})) - \boldsymbol{A}_V^*[s]^\top \psi(P_{V-1}\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x}))| \\ &\leq \tilde{\varepsilon} + ||\boldsymbol{A}_V[s] - \boldsymbol{A}_V^*[s]^\top||_1 + \sum_{s'=0}^{k_V} |\boldsymbol{A}_V^*[s][s'](\psi(P_{V-1}\eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s]) - \psi(P_{V-1}\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]))| \\ &\leq \tilde{\varepsilon} + ||\boldsymbol{A}_V[s] - \boldsymbol{A}_V^*[s]^\top||_1 + \sum_{s'=0}^{k_V} |\boldsymbol{A}_V^*[s][s'](P_{V-1}\eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s]) - \psi(P_{V-1}\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s])| \end{aligned}$$

where the second step follows since $\psi(u) \leq 1$ and the third step follows by relation (A.3) provided $|P_{V-1}\eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s] - P_{V-1}\eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]| \leq 1$. But this holds using relation (A.2) with $v = V - 1$.

Thus proceeding further we get

$$\begin{aligned} |P_V \eta_{\boldsymbol{\theta}}(\boldsymbol{x})[s] - P_V \eta_{\boldsymbol{\theta}^*}(\boldsymbol{x})[s]| &\leq \tilde{\varepsilon}(1 + k_V) + 2\tilde{\varepsilon} \sum_{s'=0}^{k_V} |W_V^*[s][s']| \sum_{v=0}^{V-1} \tilde{k}_v \prod_{v'=v+1}^{V-1} a_{v'}^* \\ &\leq \tilde{\varepsilon}\tilde{k}_v + \tilde{\varepsilon} \sum_{v=0}^{V-1} \tilde{k}_v \prod_{v'=v+1}^{V} \widetilde{\theta}'_v = \tilde{\varepsilon} \sum_{v=0}^{V} \tilde{k}_v \prod_{v'=v+1}^{V} a_{v'}^* \end{aligned}$$

This completes the proof.

Lemma A.0.9 *If* $|\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})| \leq \varepsilon$*, then* $|h_{\boldsymbol{\theta}_n}(\boldsymbol{x})| \leq 2\varepsilon$ *where*

$$h_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = \sigma(\eta_0(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})) + \log(1 - \sigma(\eta_0(\boldsymbol{x}))) - \log(1 - \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})))$$

Proof. Note that,

$$\begin{aligned}|h_{\boldsymbol{\theta}_n}(\boldsymbol{x})| &\leq |\sigma(\eta_0(\boldsymbol{x}))||\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})| + |\log(1 - \sigma(\eta_0(\boldsymbol{x})) - \log(1 - \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))| \\ &\leq |\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})| + \left|\log\left(1 + \sigma(\eta_0(\boldsymbol{x}))(e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_0(\boldsymbol{x})} - 1)\right)\right| \\ &\leq 2|\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})|\end{aligned}$$

where the second step follows by using $\sigma(x) = e^x/(1 + e^x) \leq 1$ and the proof of the third step is shown below.

Let $p = \sigma(\eta_0(\boldsymbol{x}))$, then $0 \leq p \leq 1$ and $r = \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_0(\boldsymbol{x})$, then

$$\left|\log\left(1 + \sigma(\eta_0(\boldsymbol{x}))(e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_0(\boldsymbol{x})} - 1)\right)\right| = |\log(1 + p(e^r - 1))|$$

$$\begin{aligned}r > 0 : &\quad |\log(1 + p(e^r - 1))| = \log(1 + p(e^r - 1)) \leq \log(1 + (e^r - 1)) = r = |r| \\ r < 0 : &\quad |\log(1 + p(e^r - 1))| = -\log(1 + p(e^r - 1)) \leq -\log(1 + (e^r - 1)) = -r = |r|\end{aligned}$$

Lemma A.0.10 *For* $\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) = \boldsymbol{b}_L + \boldsymbol{A}_L\psi(\boldsymbol{b}_{L-1} + \boldsymbol{A}_{L-1}\psi(\cdots\psi(\boldsymbol{b}_1 + \boldsymbol{A}_1\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x})))$,

$$\sup_{j=1,\cdots,K_n} \nabla_{\theta_j}\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) \leq \prod_{v'=1}^{L_n} a_{v'n}$$

where $a_{v'n} = \sup_{v=0,\cdots,k_{(v'+1)n}} ||\boldsymbol{A}_{v'}[v]]||_1$.

Proof. We suppress the dependence on n. Let $P_V = \boldsymbol{b}_V + \boldsymbol{A}_V\psi(\boldsymbol{b}_{V-1} + \boldsymbol{A}_{V-1}\psi(\cdots\boldsymbol{b}_1 + \boldsymbol{A}_1\psi(\boldsymbol{b}_0 + \boldsymbol{A}_0\boldsymbol{x})))$. Define $G_{V,V} = \mathbf{1}_{k_V+1}$ and for $V = 0, \cdots, L$, $V' = 0, \cdots, V - 1$, let

$$G_{V',V} = \boldsymbol{A}_V(\psi'(P_{V-1}) \odot \boldsymbol{A}_{V-1}(\psi'(P_{V-2}) \odot \cdots \boldsymbol{A}_{V+1}(\psi'(P_{V'}))))$$

where $\odot$ denotes component wise multiplication. With $\psi(P_{-1}) = \boldsymbol{x}$, we define

$$\begin{cases}\nabla_{\boldsymbol{b}_v}\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) & = G_{v,L}\mathbf{1}_{k_{v+1}} \\ \nabla_{\boldsymbol{A}_v}\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) & = G_{v,L}\mathbf{1}_{k_{v+1}}\psi(P_{v-1})^\top\end{cases}$$

By the above form and the fact that $\psi(u), \psi'(u), |x_i| \leq 1$, it can be easily checked by induction $|G_{v,L}| \leq \prod_{v'=v+1}^{L} a_{v'}$ which completes the proof.

Lemma A.0.11 *Let, $\widetilde{\mathcal{F}}_n = \{\sqrt{\ell} : \ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}), \boldsymbol{\theta}_n \in \mathcal{F}_n\}$ where $\ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x})$ is given by*

$$\ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}) = \exp\left(y\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \log\left(1 + e^{\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})}\right)\right) \tag{A.5}$$

and $\mathcal{F}_n$ is given by

$$\mathcal{F}_n = \left\{\boldsymbol{\theta}_n : |\theta_{jn}| \leq C_n, j = 1, \cdots, K_n\right\} \tag{A.6}$$

Then with $H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)$ is as in definition A.0.2,

$$\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} \sqrt{H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)}du \lesssim \varepsilon\sqrt{K_n((L_n + 1)\log K_n + (L_n + 2)\log C_n - \log\varepsilon)}$$

Proof. In this proof, we suppress the dependence on n. Note, by lemma 4.1 in [104],

$$N(\varepsilon, \mathcal{F}_n, ||.||_\infty) \leq \left(\frac{3C}{\varepsilon}\right)^K.$$

For $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{F}$, let $\widetilde{\ell}(u) = \sqrt{\ell_{u\boldsymbol{\theta}_1+(1-u)\boldsymbol{\theta}_2}(\boldsymbol{x}, y)}$.

Following equation (52) in [6], we get

$$\sqrt{\ell_{\boldsymbol{\theta}_1}(\boldsymbol{x}, y)} - \sqrt{\ell_{\boldsymbol{\theta}_2}(\boldsymbol{x}, y)} \leq K \sup_j \left|\frac{\partial\widetilde{\ell}}{\partial\theta_j}\right| ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_\infty \leq F(\boldsymbol{x}, y)||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_\infty \tag{A.7}$$

where the upper bound $F(\boldsymbol{x}, y) = (CK)^L$. This is because $|\partial\widetilde{\ell}/\partial\theta_j|$, the derivative of $\sqrt{\ell}$ w.r.t. is bounded above by $|\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})/\partial\theta_j|$ as shown below.

$$\begin{aligned}\left|\frac{\partial\widetilde{\ell}}{\partial\theta_j}\right| &= \left|\frac{1}{2}\frac{\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial\theta_j}\left(y - \frac{e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}}{1 + e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}}\right)\sqrt{e^{(y\eta_{\boldsymbol{\theta}}(\boldsymbol{x}) - \log(1+e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}))}}\right| \\ &\leq \left|\frac{1}{2}\frac{\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial\theta_j}\right|\left(\frac{e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}}{1 + e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}}\right)^{1/2}\left(\frac{1}{1 + e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}}\right)^{1/2} \leq \frac{1}{4}\left|\frac{\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial\theta_j}\right|\end{aligned}$$

Thus, using $e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}/(1 + e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})})$ and Lemma A.0.10, we get

$$\sup_{j=0,\cdots,K_n}\left|\frac{\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial\theta_j}\right| \leq \prod_{v=1}^{L} a_v^* = \prod_{v=1}^{L} k_v C \leq (KC)^L$$

In view of (B.6) and theorem 2.7.11 in [126], we have

$$N_{[]}(\varepsilon, \widetilde{\mathcal{F}}_n, ||.||_2) \leq \left(\frac{3K^{L+1}C^{L+2}}{2\varepsilon}\right)^K \implies H_{[]}(\varepsilon, \widetilde{\mathcal{F}}_n, ||.||_2) \lesssim K\log\frac{K^{L+1}C^{L+2}}{\varepsilon}$$

where $N_{[]}$ and $H_{[]}$ denote the bracketing number and bracketing entropy as in definition A.0.2.

Using, lemma A.0.4 with $M = K^{L+1}C^{L+2}$, we get

$$\int_0^{\varepsilon} \sqrt{H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)} du \lesssim \varepsilon\sqrt{K((L+1)\log K + 2(L+2)\log C - \log \varepsilon)}$$

Therefore,

$$\begin{aligned}\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2) du &\leq \int_0^{\sqrt{2}\varepsilon} H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2) du \\ &\lesssim \sqrt{2}\varepsilon\sqrt{K((L+1)\log K + (L+2)\log C - \log \sqrt{2}\varepsilon)}\end{aligned}$$

The proof follows by noting $\log \sqrt{2}\varepsilon \geq \log \varepsilon$.

A.0.2.1 Propositions

Proposition A.0.12 *Let $q(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\theta}_n^*, I_{K_n}/n^{2+2d})$ and $p(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\mu}_n, diag(\boldsymbol{\sigma}_n^2))$ where* $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$ *and* $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$. *Let $n\epsilon_n^2 \to \infty$, $K_n \log n = o(n\epsilon_n^2)$, $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$, $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$, then for any $\nu > 0$,*

$$d_{\mathrm{KL}}(q, p) \leq n\epsilon_n^2\nu$$

Proof.

$$\begin{aligned}d_{\mathrm{KL}}(q, p) &= \sum_{j=1}^{K_n}\left(\log\sqrt{n^{1+d}}\sigma_{jn} + \frac{1}{n^{1+d}\sigma_{jn}^2} + \frac{(\theta_{jn}^* - \mu_{jn})^2}{\sigma_{jn}^2} - \frac{1}{2}\right) \\ &\leq \frac{K_n}{2}((d+1)\log n - 1) + \sum_{j=1}^{K_n}\log\sigma_{jn} + \frac{1}{n^{1+d}}\sum_{j=1}^{K_n}\frac{1}{\sigma_{jn}^2} + 2\sum_{j=1}^{K_n}\frac{\theta_{jn}^{*\,2}}{\sigma_{jn}^2} + 2\sum_{j=1}^{K_n}\frac{\mu_{jn}^2}{\sigma_{jn}^2} \\ &\leq \frac{K_n}{2}((d+1)\log n - 2) + K_n \log ||\boldsymbol{\sigma}_n||_\infty + 2\left(\frac{K_n}{n} + ||\boldsymbol{\theta}_n^*||_2^2 + ||\boldsymbol{\mu}_n||_2^2\right)||\boldsymbol{\sigma}_n^*||_\infty = o(n\epsilon_n^2)\end{aligned}$$

where the second last inequality uses $\boldsymbol{\sigma}_n^* = 1/\boldsymbol{\sigma}_n$. The last equality follows since $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$, $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$, $K_n \log n = o(n\epsilon_n^2)$, $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$ and $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$.

Proposition A.0.13 *Let $p(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\mu}_n, diag(\boldsymbol{\sigma}_n^2)$ with* $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$, $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$. *Let $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 \leq \varepsilon\epsilon_n^2/4$, $n\epsilon_n^2 \to \infty$. Define,*

$$\begin{aligned}d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) &= \int_{\boldsymbol{x}\in[0,1]^{p_n}}\left(\sigma(\eta_0(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})) + \log\frac{1-\sigma(\eta_0(\boldsymbol{x}))}{1-\sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))}\right)d\boldsymbol{x} \\ \mathcal{N}_\varepsilon &= \left\{\boldsymbol{\theta}_n : d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) < \varepsilon\right\} \qquad \text{(A.8)}\end{aligned}$$

If $K_n \log n = o(n\epsilon_n^2)$*,* $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$*,* $\log(\sum_{v=0}^{L_n} k_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*) = O(\log n)$*,* $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$*,*

$$\int_{\boldsymbol{\theta}_n \in N_{\varepsilon\epsilon_n^2}} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \geq e^{-n\epsilon_n^2 \nu} \quad \forall \ \nu > 0$$

Proof. Let $\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x}) = \boldsymbol{b}_L^* + \boldsymbol{A}_L^* \psi(\boldsymbol{b}_{L-1}^* + \boldsymbol{A}_{L-1}^* \psi(\cdots \psi(\boldsymbol{b}_1^* + \boldsymbol{A}_1^* \psi(\boldsymbol{b}_0^* + \boldsymbol{A}_0^* \boldsymbol{x}))))$ be the neural network such that

$$||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq \frac{\varepsilon\epsilon_n^2}{4} \tag{A.9}$$

Such a neural network exists since $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 \leq \varepsilon\epsilon_n^2/4$.

Next define neighborhood $\mathcal{M}_{\varepsilon\epsilon_n^2}$ as follows

$$\mathcal{M}_{\varepsilon\epsilon_n^2} = \left\{ \boldsymbol{\theta}_n : |\theta_{jn} - \theta_{jn}^*| < \frac{\varepsilon\epsilon_n^2}{2\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*}, j = 1, \cdots, K_n \right\}$$

where $\tilde{k}_{vn} = k_{vn} + 1$. For every $\boldsymbol{\theta}_n \in \mathcal{M}_{\varepsilon\epsilon_n^2}$, by lemma B.0.1, we have

$$||\eta_{\boldsymbol{\theta}_n} - \eta_{\boldsymbol{\theta}_n^*}||_1 \leq \frac{\varepsilon\epsilon_n^2}{2} \tag{A.10}$$

Combining (A.9) and (B.1), we get for $\boldsymbol{\theta}_n \in \mathcal{M}_{\varepsilon\epsilon_n^2}$, $||\eta_{\boldsymbol{\theta}_n} - \eta_0||_1 \leq \varepsilon\epsilon_n^2/2$.

This, in view of lemma B.0.2, $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) \leq \varepsilon\epsilon_n^2$.

Let $\boldsymbol{\theta}_n \in \mathcal{N}_{\varepsilon\epsilon_n^2}$ for every $\boldsymbol{\theta}_n \in \mathcal{M}_{\varepsilon\epsilon_n^2}$. Therefore,

$$\int_{\boldsymbol{\theta}_n \in \mathcal{N}\varepsilon\epsilon_n^2} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \geq \int_{\boldsymbol{\theta}_n \in \mathcal{M}_{\varepsilon\epsilon_n^2}} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n$$

Let $\delta_n = \varepsilon\epsilon_n^2/(2\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*)$, then

$$\begin{aligned}
\int_{\boldsymbol{\theta}_n \in \mathcal{M}_{\varepsilon\epsilon_n^2}} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n &= \prod_{j=1}^{K_n} \int_{\theta_{jn}^* - \delta_n}^{\theta_{jn}^* + \delta_n} \frac{1}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{(\theta_{jn} - \mu_{jn})^2}{2\sigma_{jn}^2}} d\theta_{jn} \\
&= \prod_{j=1}^{K_n} \frac{2\delta_n}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{(\widehat{\theta}_{jn} - \mu_{jn})^2}{2\sigma_{jn}^2}}, \quad \widehat{\theta}_{jn} \in [\theta_{jn}^* - \delta_n, \theta_{jn}^* + \delta_n] \\
&= \prod_{j=1}^{K_n} e^{-\left(-\frac{1}{2}\log\frac{2}{\pi} - \log\delta_n + \log\sigma_{jn} + \frac{(\widehat{\theta}_{jn} - \mu_{jn})^2}{2\sigma_{jn}^2}\right)} \qquad \text{(A.11)}
\end{aligned}$$

where the second last equality holds by mean value theorem.

Note that $\widehat{\theta}_{jn} \in [\theta^*_{jn} - 1, \theta^*_{jn} + 1]$ since $\delta_n \to 0$, therefore

$$\frac{(\widehat{\theta}_{jn} - \mu_{jn})^2}{2\sigma^2_{jn}} \leq \frac{\max((\theta^*_{jn} - \mu_{jn} - 1)^2, (\theta^*_{jn} - \mu_{jn} + 1)^2)}{2\sigma^2_{jn}} \leq \frac{(\theta^*_{jn} - \mu_{jn})^2}{\sigma^2_{jn}} + \frac{1}{\sigma^2_{jn}}$$

where the last inequality follows since $(a+b)^2 \leq 2(a^2+b^2)$. Again using $(a+b)^2 \leq 2(a^2+b^2)$,

$$\begin{aligned}\sum_{j=1}^{K_n} \frac{(\widehat{\theta}_{jn} - \mu_{jn})^2}{2\sigma^2_{jn}} &\leq 2\sum_{j=1}^{K_n} \frac{\theta^{*\,2}_{jn}}{\sigma^2_{jn}} + 2\sum_{j=1}^{K_n} \frac{\mu^2_{jn}}{\sigma^2_{jn}} + \sum_{j=1}^{K_n} \frac{1}{\sigma^2_{jn}} \\ &\leq 2(||\boldsymbol{\theta}^*_n||_2^2 + ||\boldsymbol{\mu}_n||_2^2 + 1)||\boldsymbol{\sigma}^*_n||_\infty \leq n\nu\epsilon_n^2 \qquad \text{(A.12)}\end{aligned}$$

since $||\boldsymbol{\theta}^*_n||_2^2 = o(n\epsilon_n^2)$, $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$ and $||\boldsymbol{\sigma}^*_n||_\infty = O(1)$ and $n\epsilon_n^2 \to \infty$. Also,

$$\begin{aligned}-\log\delta_n + \log\sigma_{jn} &= \log 2 + \log(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n}) - \log\varepsilon\epsilon_n^2 \\ &\leq \log 2 + \log(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n}) + \log\sigma_{jn} - \log\varepsilon - 2\log\epsilon_n \\ &\leq \log 2 + O(\log n) + O(\log n) - \log\varepsilon + O(\log n)\end{aligned}$$

where the last follows since $\log||\boldsymbol{\sigma}_n||_\infty = O(\log n)$, $\log(\sum_{v=0}^{L_n} k_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n}) = O(\log n)$ and $1/n\epsilon_n^2 = o(1)$ which implies $-2\log\epsilon_n = o(\log n)$.

$$\sum_{j=1}^{K_n} -\frac{1}{2}\log\frac{2}{\pi} - \log\delta_n + \log\sigma_{jn} = O(K_n \log n) = o(n\epsilon_n^2) \qquad \text{(A.13)}$$

where the last inequality follows since $K_n \log n = o(n\epsilon_n^2)$,

Combining (B.3) and (B.4) and replacing (B.2), the proof follows.

Proposition A.0.14 *Let $q(\boldsymbol{\theta}_n) \sim MVN(\boldsymbol{\theta}^*_n, I_{K_n}/n^{2+2d})$, $d > d^*$ where $\sum_{v=0}^{L_n} k_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n} = O(n^{d^*})$, $d^* > 0$. Define*

$$h(\boldsymbol{\theta}_n) = \int_{\boldsymbol{x}\in[0,1]^{p_n}} \left(\sigma(\eta_0(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})) + \log\frac{1-\sigma(\eta_0(\boldsymbol{x}))}{1-\sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))}\right) d\boldsymbol{x}$$

Let $||\eta_0 - \eta_{\boldsymbol{\theta}^*_n}||_1 \leq \varepsilon\epsilon_n^2/4$ where $n\epsilon_n^2 \to \infty$. If $K_n \log n = o(n\epsilon_n^2)$, $||\boldsymbol{\theta}^*_n||_2^2 = o(n\epsilon_n^2)$, then

$$\int h(\boldsymbol{\theta}_n) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq \varepsilon\epsilon_n^2.$$

Proof. Since $h(\boldsymbol{\theta}_n)$ is a KL-distance, $h(\boldsymbol{\theta}_n) > 0$. We shall thus establish an upper bound.

$$\begin{aligned}\int h(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n &\le \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_0(\boldsymbol{x})|d\boldsymbol{x} \\ &\le \int\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}) - \eta_{\theta_n^*}(\boldsymbol{x})|d\boldsymbol{x}q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + ||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \\ &\le \int ||\eta_{\boldsymbol{\theta}_n} - \eta_{\boldsymbol{\theta}_n^*}||_1 q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \varepsilon\epsilon_n^2 \end{aligned} \tag{A.14}$$

where the first inequality is a consequence of lemma B.0.2 and the last inequality follows since $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 = o(\epsilon_n^2)$.

Let $S = \{\boldsymbol{\theta}_n : \cap_{j=1}^{K_n} |\theta_{jn} - \theta^*_{jn}| \le \varepsilon\epsilon_n^2/(\sum_{v=0}^{L} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n})\}$, then

$$\begin{aligned}&\int ||\eta_{\boldsymbol{\theta}_n} - \eta_{\boldsymbol{\theta}_n^*}||_1 q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \\ &= \int_S ||\eta_{\boldsymbol{\theta}_n} - \eta_{\boldsymbol{\theta}_n^*}||_1 q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \int_{S^c} ||\eta_{\boldsymbol{\theta}_n} - \eta_{\boldsymbol{\theta}_n^*}||_1 q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \\ &\le \varepsilon\epsilon_n^2 + \int_{S^c} |\boldsymbol{b}_L[s] - \boldsymbol{b}_L^*[s]|q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \int_{S^c} \sum_{s'=1}^{k_{L_n}} |\boldsymbol{A}_L[s][s'] - \boldsymbol{A}_L^*[s][s']|q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \\ &+ \sum_{s=1}^{k_{L_n}} |\boldsymbol{A}_L^*[1][s]| \int_{S^c} q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \end{aligned} \tag{A.15}$$

Let $S^c = \cup_{j=1}^{K_n} S_j^c$ where $S_j = \{|\theta_{jn} - \theta^*_{jn}| \le u_n\}$ where $u_n = \varepsilon\epsilon_n^2/(\sum_{v=0}^{L} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n})$. We first compute $Q(S^c)$ as follows:

$$\begin{aligned} Q(S^c) = Q(\cup_{j=1}^{K_n} S_j^c) \le \sum_{j=1}^{K_n} Q(S_j^c) &= \sum_{j=1}^{K_n} \int_{|\theta_{jn} - \theta^*_{jn}| > u_n} q(\theta_{jn})d\theta_{jn} \\ &= 2K_n\left(1 - \Phi\left(n^{1+d}u_n\right)\right) \end{aligned} \tag{A.16}$$

Using (A.16) in the last term of (A.15), we get

$$\begin{aligned}\sum_{s=1}^{k_{L_n}} |\boldsymbol{A}_L^*[1][s]| \int_{S^c} q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n = Q(S^c)\sum_{s=1}^{k_{L_n}} |\boldsymbol{A}_L^*[1][s]| &= a^*_{L_n n}K_n(1 - \Phi(n^{1+d}u_n)) \\ &= o(n\epsilon_n^2)O\left(n^d \frac{1}{n^{1+d}u_n} e^{-n^{2(1+d)}u_n^2}\right) = o(\epsilon_n^2) \end{aligned} \tag{A.17}$$

where the second step follows by Mill's ratio, $K_n = o(n\epsilon_n^2)$ and $\sum_{v=0}^{L_n} k_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^* = O(n^d)$ which implies $n^{1+d}u_n \to \infty$. The third step holds because

$$\frac{n^{1+d}}{n^{1+d}u_n} e^{-n^{2(1+d)}u_n^2} \leq e^{-n^{2(1+d)}u_n^2} = e^{-\left(\frac{n^{2(1+d)}\varepsilon^2\epsilon_n^4}{\log n(\sum_{v=0}^{L}\tilde{k}_{vn}\prod_{v'=v+1}^{L_n} a_{v'n}^*)^2} - (d+1)\right)} = o(1) \tag{A.18}$$

since $(\sum_{v=0}^{L} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*)^2 \log n = O(n^{2d^*} \log n) = o(n^{2d})$ for $d > d^*$ and $n^2\epsilon_n^4 \to \infty$.

For the second term in (A.15), let $S' = \{|\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| > u_n\}$

$$\begin{aligned}
&\int_{S^c} \left(|\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]|\right) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \\
&= \int_{S^c \cap S'} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n + \int_{S^c \cap S'^c} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \\
&\leq \int_{S'} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| q(\boldsymbol{b}_L[s]) d\boldsymbol{b}_L[s] + E_{q(\boldsymbol{b}_L[s])} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| Q(\tilde{S}^c),
\end{aligned} \tag{A.19}$$

$\tilde{S}^c$ is the union of all S^c_j, $j = 1, \cdots, K_n$ except the one corresponding to $\boldsymbol{b}_L[s]$.

$$\begin{aligned}
&\int_{S'} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| q(\boldsymbol{b}_L[s]) d\boldsymbol{b}_L[s] \\
&= \int_{|\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| > n^{1+d} u_n} \sqrt{\frac{n^{2+2d}}{2\pi}} (\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]) e^{-\frac{n^{2+2d}}{2} (\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s])^2} d\boldsymbol{b}_L[s] \\
&= \frac{2}{\sqrt{n^{2+2d}}} \int_{n^{1+d} u_n}^{\infty} \frac{u}{\sqrt{2\pi}} e^{-\frac{1}{2} u^2} du \leq e^{-n^{1+d} u_n}
\end{aligned} \tag{A.20}$$

Also, $E_{q(\boldsymbol{b}_L[s])} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| = \sqrt{2/\pi}(1/n^{1+d})$. Thus

$$\begin{aligned}
&E_{q(\boldsymbol{b}_L[s])} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| Q(\tilde{S}^c) \\
&= O\left(\frac{K_n}{n^{1+d}} \left(1 - \Phi\left(n^{1+d} u_n\right)\right)\right) \sim \frac{K_n}{n^{2(1+d)} u_n} e^{-n^{2(1+d)} u_n} \leq e^{-n^{2(1+v)} u_n^2}
\end{aligned} \tag{A.21}$$

where the first equality in the above step follows by observing that $Q(\tilde{S}^c)$ behaves analogous to $Q(S^c)$ which was computed in (A.16) and the second equality in the above step follows due to Mill's ratio and $\sum_{v=0}^{L} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n} = O(n^v)$ which implies $n^{1+d} u_n \to \infty$. The third inequality in the above step is a consequence of the fact that $K_n \leq n^{1+d}$.

Combining (A.17), (A.20) and (A.21), we get

$$\int_{S^c} \left(|\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]|\right) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq e^{-n^{1+d} u_n} \tag{A.22}$$

Note the third term in (A.15) can be handled similar to third term and it can be shown

$$\begin{aligned}
&\int_{S^c} |\boldsymbol{b}_L[s] - \boldsymbol{b}^*_L[s]| q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n + \int_{S^c} \sum_{s'=1}^{k_{L_n}} |\boldsymbol{A}_L[s][s'] - \boldsymbol{A}^*_L[s][s']| q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \\
&\leq k_{L_n+1} K_n e^{-n^{1+d} u_n} \leq= o((n\epsilon_n^2)^2) e^{-n^{1+d} u_n} \leq o(\epsilon_n^2) e^{-(n^{1+d} u_n - 2\log n)} = o(\epsilon_n^2)
\end{aligned} \tag{A.23}$$

where the last equality in the second step follows by $K_n = o(n\epsilon_n^2)$ and the argument in (A.18) by which $e^{-(n^{1+d}u_n - 2\log n)} = o(1)$.

Combining (A.17) and (A.23) with (A.15) the proof follows.

Proposition A.0.15 *Let $n\epsilon_n^2 \to \infty$. Let $p(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\mu}_n, diag(\boldsymbol{\sigma}_n^2))$ where $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$ and $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$. Suppose for some $0 < b < 1$, $K_n \log n = o(n^b\epsilon_n^2)$, then for $C_n = e^{n^b\epsilon_n^2/K_n}$ and $\mathcal{F}_n$ as in* (3.31)*, we have for any $\varepsilon > 0$,*

$$\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq e^{-n\varepsilon\epsilon_n^2}, n \to \infty$$

Proof. Let $\mathcal{F}_{jn} = \{\theta_{jn} : |\theta_{jn}| \leq C_n\}$

$$\mathcal{F}_n = \cap_{j=1}^{K_n} \mathcal{F}_{jn} \implies \mathcal{F}_n^c = \cap_{j=1}^{K_n} \mathcal{F}_{jn}^c$$

Note that

$$\begin{aligned}
\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n &\leq \sum_{j=1}^{K_n} \int_{\mathcal{F}_{jn}^c} \frac{1}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{(\theta_{jn}-\mu_{jn})^2}{2\sigma_{jn}^2}} d\theta_{jn} \\
&= \sum_{j=1}^{K_n} \int_{-\infty}^{-C_n} \frac{1}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{(\theta_{jn}-\mu_{jn})^2}{2\sigma_{jn}^2}} d\theta_{jn} + \sum_{j=1}^{K_n} \int_{C_n}^{\infty} \frac{1}{\sqrt{2\pi\sigma_{jn}^2}} e^{-\frac{(\theta_{jn}-\mu_{jn})^2}{2\sigma_{jn}^2}} d\theta_{jn} \\
&= \sum_{j=1}^{K_n} \left(1 - \Phi\left(\frac{C_n - \mu_{jn}}{\sigma_{jn}}\right)\right) + \sum_{j=1}^{K_n} \left(1 - \Phi\left(\frac{C_n + \mu_{jn}}{\sigma_{jn}}\right)\right)
\end{aligned}$$

Since $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2) \implies ||\boldsymbol{\mu}_n||_\infty = o(\sqrt{n}\epsilon_n)$. Since $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$ which implies for some $M > 0$, $d \geq 1$,

$$\min\left(\frac{|C_n - \mu_{jn}|}{\sigma_{jn}}, \frac{|C_n + \mu_{jn}|}{\sigma_{jn}}\right) \geq \frac{(C_n - \sqrt{n})}{n^d M} \geq e^{\log C_n - (d+1)\log n} - \frac{1}{n^{d-1/2}M} \sim e^{R_n \log n} \to \infty \tag{A.24}$$

where the last convergence holds since $K_n \log n = o(n^b\epsilon_n^2)$ implies $R_n = (n^b\epsilon_n^2)/(K_n \log n) - (d + 1) \to \infty$.

Thus, using Mill's ratio, we get:

$$\begin{aligned}
\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n &= O\left(\sum_{j=1}^{K_n} \frac{\sigma_{jn}}{C_n - \mu_{jn}} e^{-\frac{(C_n-\mu_{jn})^2}{2\sigma_{jn}^2}} + \sum_{j=1}^{K_n} \frac{\sigma_{jn}}{C_n + \mu_{jn}} e^{-\frac{(C_n+\mu_{jn})^2}{2\sigma_{jn}^2}}\right) \\
&\leq 2K_n e^{-\frac{(C_n-\sqrt{n})^2}{2n^2M^2}} \leq e^{-\varepsilon n\epsilon_n^2}
\end{aligned}$$

where the last asymptotic inequality holds because

$$\frac{(C_n - \sqrt{n})^2}{2n^d M^2} - \log 2K_n \sim \frac{1}{2}e^{2R_n \log n} - 2\log K_n \geq n\left(\frac{e^{2R_n}}{2} - \frac{2\log n}{n}\right) \geq \varepsilon n \epsilon_n^2$$

In the above step, the first asymptotic equivalence holds due to (A.24), the second inequality holds since $K_n \leq n$. The last inequality holds since $R_n \to \infty$ and $\log/n \to 0$.

Proposition A.0.16 *Let $n\epsilon_n^2 \to \infty$. Suppose $K_n \log n = o(n^b \epsilon_n^2)$, for some $0 < b < 1$, $L_n \sim \log n$ and $p(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\mu}_n, diag(\boldsymbol{\sigma}_n^2))$ where $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$ and $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$. Then for every $\varepsilon > 0$,*

$$\log \int_{\mathcal{U}^c_{\varepsilon\epsilon_n}} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq \log 2 - \varepsilon^2 n\epsilon_n^2 + o_{P_0^n}(1)$$

Proof. It suffices to show

$$P_0^n\left(\int_{\mathcal{U}^c_{\varepsilon\epsilon_n}} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n > 2e^{-\varepsilon n\epsilon_n^2}\right) \to 0, \;\; n \to \infty \tag{A.25}$$

$$\begin{aligned} &P_0^n\left(\int_{\mathcal{U}^c_{\varepsilon\epsilon_n}} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n > 2e^{-\varepsilon^2 n\epsilon_n^2}\right) \\ &\leq P_0^n\left(\int_{\mathcal{U}^c_{\varepsilon\epsilon_n}\cap\mathcal{F}_n} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n > e^{-\varepsilon^2 n\epsilon_n^2}\right) + P_0^n\left(\int_{\mathcal{F}_n^c} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n > e^{-\varepsilon^2 n\epsilon_n^2}\right) \end{aligned}$$

Using lemma B.0.4 with $\varepsilon = \varepsilon\epsilon_n$ and $C_n = e^{n^b\epsilon_n^2/K_n}$,

$$\begin{aligned} &\int_{\varepsilon^2\epsilon_n^2/8}^{\sqrt{2}\varepsilon\epsilon_n} H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2) du \\ &\lesssim \epsilon_n\varepsilon\sqrt{K_n((L_n+1)\log K_n + (L_n+2)\log C_n - \log \varepsilon\epsilon_n)} \\ &\leq \varepsilon\epsilon_n O(\max(\sqrt{K_n(L_n+1)\log K_n}, \sqrt{K_n(L_n+2)\log C_n}, \sqrt{-\log \epsilon_n})) \\ &\leq \varepsilon\epsilon_n \max(o(\epsilon_n\sqrt{n^b \log n}), O(\epsilon_n\sqrt{n^b \log n}), O(\sqrt{\log n})) \leq \varepsilon^2\epsilon_n^2\sqrt{n} \end{aligned}$$

where $H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)$ is as in definition A.0.2. The first inequality in the third step follows because $L_n \sim \log n$, $K_n \log n = o(n^b\epsilon_n^2)$ and $K_n \log C_n = n^b\epsilon_n^2$, $-\log \epsilon_n^2 \leq \log n$. The second inequality in the third step follows since $(n^b \log n)/n = o(1)$

By theorem 1 in [138], for some constant $C > 0$, we have

$$P_0^n\left(\int_{\boldsymbol{\theta}_n\in\mathcal{U}^c_{\varepsilon\epsilon_n}\cap\mathcal{F}_n}\frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n > e^{-\varepsilon^2n\epsilon_n^2}\right) \leq P_0^n\left(\sup_{\boldsymbol{\theta}_n\in\mathcal{U}^c_{\varepsilon\epsilon_n}\cap\mathcal{F}_n}\frac{L(\boldsymbol{\theta}_n)}{L_0} > e^{-\varepsilon^2n\epsilon_n^2}\right)$$
$$\leq 4\exp(-C\varepsilon^2n\epsilon_n^2) = o(n\epsilon_n^2) \tag{A.26}$$

Using proposition B.0.7 with $\varepsilon = 2\varepsilon$, we have

$$\int_{\boldsymbol{\theta}_n\in\mathcal{F}_n^c}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \leq e^{-2n\varepsilon^2{\epsilon_n}^2}$$

Therefore, using lemma A.0.7 with $\varepsilon = 2\varepsilon^2\epsilon_n^2$ and $\tilde{\varepsilon} = \varepsilon^2\epsilon_n^2$, we have

$$P_0^n\left(\int_{\mathcal{F}_n^c}\frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n > e^{-\varepsilon^2n\epsilon_n^2}\right) \leq e^{-\varepsilon^2n\epsilon_n^2} \to 0. \tag{A.27}$$

Combining (A.26) and (A.27), (A.25) follows.

Proposition A.0.17 *Let $p(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\mu}_n, diag(\boldsymbol{\sigma}_n^2))$, $||\boldsymbol{\sigma}_n||_\infty = O(n)$ and $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$.*

1. *Let $L_n = L$, $p_n = p$ independent of n. If $K_n\log n = o(n)$ and $||\boldsymbol{\mu}_n||_2^2 = o(n)$, then*

$$d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) = o_{P_0^n}(n) \tag{A.28}$$

2. *Let $K_n\log n = o(n\epsilon_n^2)$, $L_n \sim \log n$ and $||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2)$. There exists a neural network such $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 = o(n\epsilon_n^2)$, $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$ and $\log(\sum_{v=0}^{L_n}\tilde{k}_{vn}\prod_{v'=v+1}^{L_n}a^*_{v'n}) = O(\log n)$, then*

$$d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) = o_{P_0^n}(n\epsilon_n^2) \tag{A.29}$$

Proof. For any $q \in \mathcal{Q}_n$.

$$\begin{aligned}
d_{\mathrm{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) &= \int q(\boldsymbol{\theta}_n)\log q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n - \int q(\boldsymbol{\theta}_n)\log\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n\\
&= \int q(\boldsymbol{\theta}_n)\log q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n - \int q(\boldsymbol{\theta}_n)\log\frac{L(\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)}{\int L(\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n}d\boldsymbol{\theta}_n\\
&= d_{\mathrm{KL}}(q, p) - \int\log\frac{L(\boldsymbol{\theta}_n)}{L_0}q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \log\int\frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\\
&\leq d_{\mathrm{KL}}(q, p) + \left|\int\log\frac{L(\boldsymbol{\theta}_n)}{L_0}q(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right| + \left|\log\int\frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right| \quad (A.30)
\end{aligned}$$

Since π^* satisfies minimizes the KL-distance to $\pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)$ in the family $\mathcal{Q}_n$, therefore for any $\kappa > 0$

$$P_0^n\left(d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) > \kappa\right) \leq P_0^n\left(d_{\mathrm{KL}}(q, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) > \kappa\right) \tag{A.31}$$

Proof of part 1. Note, $K_n \log n = o(n)$, $||\mu_n||_2^2 = o(n)$, $||\boldsymbol{\sigma}_n||_\infty = O(n)$ and $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$. We take $q(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\theta}_n^*, \boldsymbol{I}_{K_n}/\sqrt{n})$ where $\boldsymbol{\theta}_n^*$ is defined next.

For $N \geq 1$, let $\eta_{\boldsymbol{\theta}_N^*}$ be a finite neural network approximation satisfying $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq \varepsilon/4$. The existence of such a neural network is always guaranteed by [60]. Define $\boldsymbol{\theta}_n^*$ same as $\boldsymbol{\theta}_N^*$ for all the non zero coefficients and zeros for all non existent coefficients.

Step 1 (a): Using proposition A.0.12, with $\epsilon_n = 1$, we get for any $\nu > 0$,

$$P_0^n(d_{\mathrm{KL}}(q, p) > n\nu) = 0 \tag{A.32}$$

where the above step follows $||\boldsymbol{\theta}_n^*||_2^2 = ||\boldsymbol{\theta}_N^*||_2^2 = ||\boldsymbol{\theta}_n^*||_2^2 = o(n)$.

Step 1 (b): Next, note that

$$\begin{aligned} d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) &= \int_{\boldsymbol{x}\in[0,1]^{p_n}} \left(\sigma(\eta_0(\boldsymbol{x})) \log \frac{\sigma(\eta_0(\boldsymbol{x}))}{\sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))} + (1 - \sigma(\eta_0(\boldsymbol{x}))) \log \frac{1 - \sigma(\eta_0(\boldsymbol{x}))}{1 - \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))}\right) d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in[0,1]^{p_n}} \left(\sigma(\eta_0(\boldsymbol{x}))(\sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x})) - \sigma(\eta_0(\boldsymbol{x}))) + \log \frac{1 - \sigma(\eta_0(\boldsymbol{x}))}{1 - \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))}\right) d\boldsymbol{x} \end{aligned} \tag{A.33}$$

Since $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 \leq \varepsilon/4$, using proposition B.0.5 with $\epsilon_n = 1$ and $\varepsilon = \varepsilon$

$$\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq \varepsilon$$

which follows by $||\boldsymbol{\theta}_n^*||_2^2 = ||\boldsymbol{\theta}_N^*||_2^2 = o(n)$ and $\log(\sum_{v=0}^{L} \tilde{k}_{vN} \prod_{v'=v+1}^{L} a_{v'N}^*) = O(\log n)$.

Therefore, by lemma A.0.5,

$$P_0^n\left(\left|\int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| > n\nu\right) \leq \frac{\varepsilon}{\nu}. \tag{A.34}$$

Step 1 (c): Since $||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 \leq \varepsilon/4$, using proposition B.0.3 with $\epsilon_n = 1$ and $\nu = \varepsilon$,

$$\int_{\boldsymbol{\theta}_n \in \mathcal{N}_\varepsilon} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \geq \exp(-n\varepsilon)$$

which follows by $||\boldsymbol{\theta}_n^*||_2^2 = ||\boldsymbol{\theta}_N^*||_2^2 = o(n)$ and $\log(\sum_{v=0}^{L} \tilde{k}_{vn} \prod_{v'=v+1}^{L} a_{v'n}^*) = O(\log n)$. Therefore, using lemma A.0.6, we get

$$P_0^n\left(\left|\log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| > n\nu\right) \leq \frac{2\varepsilon}{\nu} \tag{A.35}$$

Step 1 (d): From (A.31) and (A.30) we get

$$\begin{aligned} &P_0^n(d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) > 3n\nu) \leq P_0^n\left(d_{\mathrm{KL}}(q, p) > n\nu\right) \\ &+ P_0^n\left(\left|\int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| > n\nu\right) + P_0^n\left(\left|\log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| > n\nu\right) \leq \frac{3\varepsilon}{\nu} \end{aligned} \tag{A.36}$$

where the last inequality is a consequence of (A.32), (A.34) and (A.35).

Since ε is arbitrary, taking $\varepsilon \to 0$ completes the proof.

Proof of part 2. Note, $K_n \log n = o(n\epsilon_n^2)$, $||\mu_n||_2^2 = o(n\epsilon_n^2)$, $\log ||\boldsymbol{\sigma}_n||_\infty = O(\log n)$ and $||\boldsymbol{\sigma}_n^*||_\infty = O(1)$. Let $q(\boldsymbol{\theta}_n) = MVN(\boldsymbol{\theta}_n^*, \boldsymbol{I}_{K_n}/n^{2+2d}), d > d^*$ where $\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^* = O(n^{d^*})$, $d^* > 0$. We next define $\boldsymbol{\theta}_n^*$ as follows:

Let $\eta_{\boldsymbol{\theta}_n^*}$ be the neural satisfying

$$||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq \varepsilon\epsilon_n^2/4 \quad ||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$$

The existence of such a neural network is guaranteed since $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 = o(\epsilon_n^2)$.

Step 2 (a): Since $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$, by proposition A.0.12,

$$P_0^n(d_{\mathrm{KL}}(q, p) > \nu n\epsilon_n^2) = 0 \tag{A.37}$$

Step 2 (b): Since $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq \varepsilon\epsilon_n^2/4$, $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$ and $(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*) \log n = o(n\epsilon_n^2)$, by proposition B.0.5,

$$\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \leq \varepsilon\epsilon_n^2$$

Therefore, by lemma A.0.5,

$$P_0^n\left(\left|\int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n\right| > \nu n\epsilon_n^2\right) \leq \frac{\varepsilon}{\nu}. \tag{A.38}$$

Step 2 (c): Since $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq \varepsilon\epsilon_n^2/4$, $||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2)$ and $\log(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a^*_{v'n}) = O(\log n)$, by proposition B.0.3,

$$\int_{\boldsymbol{\theta}_n \in \mathcal{N}\varepsilon\epsilon_n^2} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \geq \exp(-\varepsilon n \epsilon_n^2)$$

Therefore, using lemma A.0.6, we get

$$P_0^n \left(\left| \log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \right| > \nu n \epsilon_n^2 \right) \leq \frac{2\varepsilon}{\nu} \tag{A.39}$$

Step 2 (d): From (A.31) and (A.30) we get

$$\begin{aligned} &P_0^n(d_{\text{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) > 3\nu n \epsilon_n^2) \leq P_0^n \left(d_{\text{KL}}(q, p) > \nu n \epsilon_n^2 \right) \\ &+ P_0^n \left(\left| \int \log \frac{L(\boldsymbol{\theta}_n)}{L_0} q(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \right| > \nu n \epsilon_n^2 \right) + P_0^n \left(\left| \log \int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \right| > \nu n \epsilon_n^2 \right) \leq \frac{3\varepsilon}{\nu} \end{aligned} \tag{A.40}$$

where the last inequality is a consequence of (A.37), (A.38) and (A.39).

Since ε is arbitrary, taking $\varepsilon \to 0$ completes the proof.

Consistency of the variational posterior.

Proof of Theorem 1.

We assume Relation (B.13) holds with A_n and B_n are same as in (3.29).

By assumptions (A1) and (A2), the prior parameters satisfy

$$||\boldsymbol{\mu}_n||_2^2 = o(n), \ \ \log ||\boldsymbol{\sigma}_n||_\infty = O(\log n), \ \ ||\boldsymbol{\sigma}_n^*||_\infty = O(1), \ \ \boldsymbol{\sigma}_n^* = 1/\boldsymbol{\sigma}_n.$$

Note $K_n \sim n^a$, $0 < a < 1$ which implies $K_n \log n = o(n)$. By proposition A.0.17 part 1.,

$$d_{\text{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) = o_{P_0^n}(n). \tag{A.41}$$

By step 1 (c) in the proof of proposition A.0.17

$$B_n = o_{P_0^n}(n) \tag{A.42}$$

Since, $K_n \sim n^a$, $K_n \log n = o(n^b)$, $a < b < 1$. Using proposition A.0.16 with $\epsilon_n = 1$,

$$-\pi^*(\mathcal{U}_\varepsilon^c) A_n \geq n\varepsilon^2 \pi^*(\mathcal{U}_\varepsilon^c) - \log 2 + o_{P_0^n}(1) = n\varepsilon^2 \pi^*(\mathcal{U}_\varepsilon^c) + O_{P_0^n}(1) \tag{A.43}$$

Thus, using (A.41), (A.42) and (A.43) in (B.13), we get

$$n\varepsilon^2\pi^*(\mathcal{U}_\varepsilon^c) + O_{P_0^n}(1) \le o_{P_0^n}(n) + o_{P_0^n}(n) \implies \pi^*(\mathcal{U}_\varepsilon^c) = o_{P_0^n}(1)$$

Proof of Theorem 2.

We assume Relation (B.13) holds with A_n and B_n are same as in (3.29).

Let $K_n \sim n^a$ and $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1 - a$. This implies $K_n \log n = o(n\epsilon_n^2)$.

By assumptions (A1) and (A4), the prior parameters satisfy

$$||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2), \ \log ||\boldsymbol{\sigma}_n||_\infty = O(\log n), \ ||\boldsymbol{\sigma}_n^*||_\infty = O(1), \ \boldsymbol{\sigma}_n^* = 1/\boldsymbol{\sigma}_n.$$

Also by assumption (A3),

$$||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 = o(\epsilon_n^2), \ ||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2), \ \log(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*) = O(\log n)$$

By proposition A.0.17 part 2.,

$$d_{\mathrm{KL}}(\pi^*, \pi(.|\boldsymbol{y}_n, \boldsymbol{X}_n)) = o_{P_0^n}(n\epsilon_n^2). \tag{A.44}$$

By step 2 (c) in the proof of proposition A.0.17

$$B_n = o_{P_0^n}(n\epsilon_n^2) \tag{A.45}$$

Since $K_n \sim n^a$, $K_n \log n = o(n^b\epsilon_n^2)$, $a + \delta < b < 1$. Using proposition A.0.16, it follows that

$$-\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c)A_n \ge \varepsilon^2 n\epsilon_n^2\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) - \log 2 + o_{P_0^n}(1) = \varepsilon^2 n\epsilon_n^2\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) + O_{P_0^n}(1) \tag{A.46}$$

Thus, using (A.44), (A.45) and (A.46) in (B.13), we get

$$n\varepsilon^2\epsilon_n^2\pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) + O_{P_0^n}(1) \le o_{P_0^n}(n\epsilon_n^2) + o_{P_0^n}(n\epsilon_n^2) \implies \pi^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) = o_{P_0^n}(1)$$

Proof of Corollary 1.

Let $\hat{\ell}_n(y, \boldsymbol{x}) = \int \ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x})\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n$.

$$\begin{aligned} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) &= d_{\mathrm{H}}\left(\int \ell_{\boldsymbol{\theta}_n}\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n, \ell_0\right) \\ &\le \int d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \quad \text{Jensen's inequality} \\ &= \int_{\mathcal{U}_\varepsilon} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n + \int_{\mathcal{U}_\varepsilon^c} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \\ &\le \varepsilon + o_{P_0^n}(1) \end{aligned}$$

Taking $\varepsilon \longrightarrow 0$, we get $d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) = o_{P_0^n}(1)$. Let

$$\hat{\eta}(\boldsymbol{x}) = \sigma^{-1}\left(\int \sigma(\eta_{\boldsymbol{\theta}_n}(\boldsymbol{x}))\pi^*(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right) \tag{A.47}$$

then, note that $\hat{\eta}(\boldsymbol{x}) = \log \frac{\hat{\ell}_n(1,\boldsymbol{x})}{\hat{\ell}_n(0,\boldsymbol{x})}$.

$$\begin{aligned}
2d_{\mathrm{H}}^2(\hat{\ell}_n, \ell_0) &= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} \sqrt{\hat{\ell}_n(y,\boldsymbol{x})\ell_0(y,\boldsymbol{x})}d\boldsymbol{x} \\
&= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sum_{y\in\{0,1\}} e^{\left\{\frac{1}{2}\left(y\hat{\eta}(\boldsymbol{x}) - \log(1+e^{\hat{\eta}(\boldsymbol{x})}) + y\eta_0(\boldsymbol{x}) - \log(1+e^{\eta_0(\boldsymbol{x})})\right)\right\}}d\boldsymbol{x} \\
&= 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \left(\sqrt{\sigma(\eta_0(\boldsymbol{x}))\sigma(\hat{\eta}(\boldsymbol{x}))} + \sqrt{(1-\sigma(\eta_0(\boldsymbol{x})))(1-\sigma(\hat{\eta}(\boldsymbol{x})))}\right)d\boldsymbol{x} \\
&\geq 2 - 2\int_{\boldsymbol{x}\in[0,1]^p} \sqrt{1 - (\sqrt{\sigma(\eta_0(\boldsymbol{x}))} - \sqrt{\sigma(\hat{\eta}(\boldsymbol{x}))})^2}d\boldsymbol{x} \\
&\geq \int_{\boldsymbol{x}\in[0,1]^p} (\sqrt{\sigma(\eta_0(\boldsymbol{x}))} - \sqrt{\sigma(\hat{\eta}(\boldsymbol{x}))})^2 d\boldsymbol{x} \geq \frac{1}{4}\int_{\boldsymbol{x}\in[0,1]^p} (\sigma(\eta_0(\boldsymbol{x})) - \sigma(\hat{\eta}(\boldsymbol{x})))^2 d\boldsymbol{x}
\end{aligned} \tag{A.48}$$

In the above equation, the sixth and the seventh step hold because $\sqrt{1-x} \leq 1 - x/2$ and $|p_1 - p_2| \leq |\sqrt{p_1} + \sqrt{p_2}||\sqrt{p_1} - \sqrt{p_2}| \leq 2|\sqrt{p_1} - \sqrt{p_2}|$ respectively. The fifth step holds because

$$\begin{aligned}
\left(\sqrt{p_1p_2} + \sqrt{(1-p_1)(1-p_2)}\right)^2 &= p_1p_2 + 1 - p_1 - p_2 + \sqrt{p_1p_2(1-p_1)(1-p_2)} \\
&\leq \sqrt{p_1p_2} + 1 - p_1 - p_2 + \sqrt{p_1p_2} = 1 - (\sqrt{p1} - \sqrt{p_2})^2
\end{aligned}$$

By (A.48) and Cauchy Schwartz inequality,

$$\begin{aligned}
\int_{\boldsymbol{x}\in U[0,1]^p} |\sigma(\eta_0(\boldsymbol{x})) - \sigma(\hat{\eta}(\boldsymbol{x}))|d\boldsymbol{x} &\leq \left(\int_{\boldsymbol{x}\in[0,1]^p} (\sigma(\eta_0(\boldsymbol{x})) - \sigma(\hat{\eta}(\boldsymbol{x})))^2 d\boldsymbol{x}\right)^{1/2} \\
&\leq 2\sqrt{2}d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) = o_{P_0^n}(1)
\end{aligned} \tag{A.49}$$

The proof follows in lieu (3.33).

Proof of Corollary 2.

We assume Relation (B.13) holds with A_n and B_n are same as in (3.29).

Let $K_n \sim n^a$ and $\epsilon_n^2 \sim n^{-\delta}$, $0 < \delta < 1 - a$. This implies $K_n \log n = o(n\epsilon_n^2)$.

Also, $K_n \log n = o(n^b \epsilon_n^2)$, $a + \delta < b < 1$. This implies $K_n \log n = o(n^b (\epsilon_n^2)^\kappa)$, $0 \leq \kappa \leq 1$. Thus, using proposition A.0.16 with $\epsilon_n = \epsilon_n^k$, we get

$$-\pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n^\kappa})A_n \geq \varepsilon^2 n \epsilon_n^{2\kappa} \pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n^\kappa}) - \log 2 + o_{P_0^n}(1) = \varepsilon^2 n \epsilon_n^{2\kappa} \pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n^k}) + O_{P_0^n}(1) \tag{A.50}$$

This together with (A.44), (A.45) and (B.13) implies

$$\pi^*(\mathcal{U}^c_{\varepsilon\epsilon_n^\kappa}) = o_{P_0^n}(\epsilon_n^{2-2\kappa})$$

Let $\hat{\ell}_n(y, \boldsymbol{x}) = \int \ell_{\boldsymbol{\theta}_n}(y, \boldsymbol{x}) \pi^*(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n$.

$$\begin{aligned} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) &\leq \int_{\mathcal{U}_{\varepsilon\epsilon_n^\kappa}} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0) \pi^*(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n + \int_{\mathcal{U}^c_{\varepsilon\epsilon_n^\kappa}} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0) \pi^*(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n \\ &\leq \varepsilon \epsilon_n^\kappa + o_{P_0^n}(\epsilon_n^{2-2\kappa}) \end{aligned}$$

Dividing by ϵ_n^κ on both sides we get

$$\frac{1}{\epsilon_n^\kappa} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) = o_{P_0^n}(\epsilon_n^{2-3\kappa}) + o_{P_0^n}(1) = o_{P_0^n}(1), \quad 0 \leq \kappa \leq 2/3.$$

By (A.49), for every $0 \leq \kappa \leq 2/3$,

$$\frac{1}{\epsilon_n^\kappa} \int_{\boldsymbol{x} \in [0,1]^{p_n}} |\sigma(\eta_0(\boldsymbol{x})) - \sigma(\hat{\eta}(\boldsymbol{x}))| d\boldsymbol{x} \leq \frac{1}{\epsilon_n^\kappa} 2\sqrt{2} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) = o_{P_0^n}(1).$$

The proof follows in lieu of (3.33).

Consistency of the true posterior.

From (4.9), note that

$$\pi(\mathcal{U}_\varepsilon^c | \boldsymbol{y}_n, \boldsymbol{X}_n) = \frac{\int_{\mathcal{U}_\varepsilon^c} L(\boldsymbol{\theta}_n) p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n}{\int L(\boldsymbol{\theta}_n) p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n} = \frac{\int_{\mathcal{U}_\varepsilon^c} (L(\boldsymbol{\theta}_n)/L_0) p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n}{\int (L(\boldsymbol{\theta}_n)/L_0) p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n} \tag{A.51}$$

Theorem A.0.18 *Suppose conditions of theorem 3.4.1 hold. Then,*

1.

$$P_0^n \left(\pi(\mathcal{U}_\varepsilon^c | \boldsymbol{y}_n, \boldsymbol{X}_n) \leq 2e^{-n\varepsilon^2/2} \right) \to 1, n \to \infty$$

2.

$$P_0^n(|R(\hat{C}) - R(C^{\mathrm{Bayes}})| \leq 8\sqrt{2}\varepsilon) \to 1, n \to \infty$$

Proof. By assumptions (A1) and (A2), the prior parameters satisfy

$$||\boldsymbol{\mu}_n||_2^2 = o(n),\ \ \log||\boldsymbol{\sigma}_n||_\infty = O(\log n),\ \ ||\boldsymbol{\sigma}_n^*||_\infty = O(1),\ \ \boldsymbol{\sigma}_n^* = 1/\boldsymbol{\sigma}_n.$$

Note $K_n \sim n^a$, $0 < a < 1$ which implies $K_n \log n = o(n)$. Thus, the conditions of proposition B.0.3 hold with $\epsilon_n = 1$.

$$P_0^n\left(\int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \leq e^{-n\nu}\right) \leq P_0^n\left(\left|\log\int \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right| > n\nu\right) \to 0, n \to \infty \quad \text{(A.52)}$$

where the above convergence follows from (A.35) in step 1 (c) in the proof of proposition A.0.17. Since $K_n \log n = o(n^b)$, $a < b < 1$, the conditions of proposition A.0.16 hold with $\epsilon_n = 1$.

$$P_0^n\left(\int_{\mathcal{U}_\varepsilon^c} \frac{L(\boldsymbol{\theta}_n)}{L_0} p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \geq 2e^{-n\varepsilon^2}\right) \to 0, n \to \infty \quad \text{(A.53)}$$

where the last equality follows from (A.25) with $\epsilon_n = 1$ in the proof of proposition A.0.16. Using (A.52) and (A.53) with (A.51), we get

$$P_0^n\left(\pi(\mathcal{U}_\varepsilon^c|\boldsymbol{y}_n, \boldsymbol{X}_n) \geq 2e^{-n(\varepsilon^2-\nu)}\right) \to 0, n \to \infty$$

Take $\nu = \varepsilon^2/2$ to complete the proof. Mimicking the steps in the proof of corollary 1,

$$\begin{aligned} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) &\leq \int d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n \quad \text{Jensen's inequality} \\ &= \int_{\mathcal{U}_\varepsilon} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n + \int_{\mathcal{U}_\varepsilon^c} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n \\ &\leq \varepsilon + 2e^{-n\varepsilon^2/2} \leq 2\varepsilon, \ \text{with probability tending to 1 as } n \to \infty \end{aligned}$$

where the second last inequality is a consequence of part 1. in theorem A.0.18. The remaining part of the proof follows by (A.49) and (3.33).

Theorem A.0.19 *Suppose conditions of theorem 3.4.2 hold. Then,*

1.

$$P_0^n\left(\pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\boldsymbol{y}_n, \boldsymbol{X}_n) \leq 2e^{-n\epsilon_n^2\varepsilon^2/2}\right) \to 1, n \to \infty$$

2.

$$P_0^n(|R(\hat{C}) - R(C^{\mathrm{Bayes}})| \leq 8\sqrt{2}\varepsilon\epsilon_n) \to 1, n \to \infty$$

Proof. By assumptions (A1) and (A4), the prior parameters satisfy

$$||\boldsymbol{\mu}_n||_2^2 = o(n\epsilon_n^2), \;\; \log||\boldsymbol{\sigma}_n||_\infty = O(\log n), \;\; ||\boldsymbol{\sigma}_n^*||_\infty = O(1), \;\; \boldsymbol{\sigma}_n^* = 1/\boldsymbol{\sigma}_n.$$

Also by assumption (A3),

$$||\eta_0 - \eta_{\boldsymbol{\theta}_n^*}||_1 = o(\epsilon_n^2), \;\; ||\boldsymbol{\theta}_n^*||_2^2 = o(n\epsilon_n^2), \;\; \log(\sum_{v=0}^{L_n} \tilde{k}_{vn} \prod_{v'=v+1}^{L_n} a_{v'n}^*) = O(\log n)$$

Note $K_n \sim n^a, 0 < a < 1$ and $\epsilon_n \sim n^{-\delta}, 0 < \delta < 1-a$, thus $K_n \log n = o(n\epsilon_n^2)$. Thus, the conditions of proposition B.0.3 hold.

$$P_0^n\left(\int \frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \le e^{-n\epsilon_n^2 \nu}\right) \le P_0^n\left(\left|\log\int \frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n\right| > n\epsilon_n^2\nu\right) \to 0, n \to \infty \tag{A.54}$$

where the above convergence follows from (A.39) in step 2 (c) in the proof of proposition A.0.17. Also, since $K_n \log n = o(n^b\epsilon_n^2)$, $a + \delta < b < 1$. Thus conditions of proposition A.0.16 hold.

$$P_0^n\left(\int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} \frac{L(\boldsymbol{\theta}_n)}{L_0}p(\boldsymbol{\theta}_n)d\boldsymbol{\theta}_n \ge 2e^{-n\epsilon_n^2\varepsilon^2}\right) \to 0, n \to \infty \tag{A.55}$$

where the last equality follows from (A.25) in the proof of proposition A.0.16.
Using (A.54) and (A.55) with (A.51), we get $P_0^n\left(\pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\boldsymbol{y}_n, \boldsymbol{X}_n) \ge 2e^{-n\epsilon_n^2(\varepsilon^2-\nu)}\right) \to 0, n \to \infty$. Take $\nu = \varepsilon^2/2$ to complete the proof. Mimicking the steps in the proof of corollary 2,

$$\begin{aligned}
d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) &\le \int d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n \quad \text{Jensen's inequality} \\
&= \int_{\mathcal{U}_{\varepsilon\epsilon_n}} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n + \int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} d_{\mathrm{H}}(\ell_{\boldsymbol{\theta}_n}, \ell_0)\pi(\boldsymbol{\theta}_n|\boldsymbol{y}_n, \boldsymbol{X}_n)d\boldsymbol{\theta}_n \\
&\le \varepsilon\epsilon_n + 2e^{-2n\epsilon_n^2\varepsilon^2} \le 2\varepsilon\epsilon_n, \;\; \text{with probability tending to 1 as } n \to \infty
\end{aligned}$$

where the second last inequality is a consequence of part 1. in theorem A.0.19 and the last inequality last equality follows since $\epsilon_n \sim n^{-\delta}$. Dividing by ϵ_n on both sides we get

$$\epsilon_n^{-1} d_{\mathrm{H}}(\hat{\ell}_n, \ell_0) \le 2\varepsilon, \;\; \text{with probability tending to 1 as } n \to \infty$$

The remaining part of the proof follows by (A.49) and (3.33).

APPENDIX B

SUPPLEMENT FOR LEARNING INTRINSIC DIMENSIONALITY OF FEATURE SPACE WITH VARIATIONAL BAYES NEURAL NETWORKS

Proof of Lemmas

Lemma B.0.1 *Consider,* $\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{k_n}\beta_j\psi(\gamma_j{}^\top A\boldsymbol{x})$ *and* $\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x}) = \beta_0^* + \sum_{j=1}^{k_n}\beta_j^*\psi(\gamma_j^{*\top}\boldsymbol{x})$*. If*

$$|\beta_j - \beta_j^*| \le \epsilon, \;\; j = 1, \cdots, k_n \qquad |\gamma_j{}^\top A\boldsymbol{x} - \gamma_j^{*\top}\boldsymbol{x}| \le \epsilon, \;\; j = 1, \cdots, k_n,$$

$$\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})| d\boldsymbol{x} \le 2\epsilon(k_n + ||\boldsymbol{\beta}^*||_1)$$

Proof. This proof uses somes ideas in the proof of theorem 1 in [74].

Note that $|\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})| \le |\beta_0 - \beta_0^*| + \sum_{j=1}^{k_n} |\beta_j\psi(\gamma_j^\top A\boldsymbol{x}) - \beta_j^*\psi(\gamma_j^{*\top}\boldsymbol{x})|$. Let $u_j = \gamma_j^\top A\boldsymbol{x}$, $r_j = \gamma_j^{*\top}\boldsymbol{x} - \gamma_j^\top A\boldsymbol{x}$, then $|\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})|$ is bounded above by

$$\begin{aligned} &\le |\beta_0 - \beta_0^*| + \sum_{j=1}^{k_n}\left|\frac{\beta_j}{1+e^{u_j}} - \frac{\beta_j^*}{1+e^{u_j+r_j}}\right| = |\beta_0 - \beta_0^*| + \sum_{j=1}^{k_n}\left|\frac{\beta_j(1+e^{u_j+r_j}) - \beta_j^*(1+e^{u_j})}{(1+e^{u_j+r_j})(1+e^{u_j})}\right| \\ &= |\beta_0 - \beta_0^*| + \sum_{j=1}^{k_n}\frac{|\beta_j - \beta_j^*| + |\beta_j e^{u_j+r_j} - \beta_j^* e^{u_j}|}{(1+e^{u_j+r_j})(1+e^{u_j})} = 2\sum_{j=0}^{k_n}|\beta_j - \beta_j^*| + \sum_{j=1}^{k_n}|\beta_j^*||e^{r_j} - 1| \end{aligned}$$

Since, $|r_j| < \epsilon < 1$, thus $|1 - e^{r_j}| < 2|r_j| \le 2\epsilon$, the proof follows.

Lemma B.0.2 *For any two functions,* η_0 *and* η_1*,*

$$h(\boldsymbol{x}) = \sigma(\eta_1(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_1(\boldsymbol{x})) + \log(1 - \sigma(\eta_0(\boldsymbol{x}))) - \log(1 - \sigma(\eta_1(\boldsymbol{x}))) \le 2|\eta_0(\boldsymbol{x}) - \eta_1(\boldsymbol{x})|$$

Proof. Using $\sigma(x) = e^x/(1+e^x) \le 1$

$$\begin{aligned} |h(\boldsymbol{x})| &\le |\sigma(\eta_0(\boldsymbol{x}))||\eta_0(\boldsymbol{x}) - \eta_1(\boldsymbol{x})| + |\log(1 - \sigma(\eta_0(\boldsymbol{x}))) - \log(1 - \sigma(\eta_1(\boldsymbol{x})))| \\ &\le |\eta_0(\boldsymbol{x}) - \eta_1(\boldsymbol{x})| + \left|\log\left(1 + \sigma(\eta_0(\boldsymbol{x}))(e^{\eta_1(\boldsymbol{x}) - \eta_0(\boldsymbol{x})} - 1)\right)\right| \le 2|\eta_0(\boldsymbol{x}) - \eta_1(\boldsymbol{x})| \end{aligned}$$

where the proof of the above line is as follows. Let $p = \sigma(\eta_0(\boldsymbol{x}))$, then $0 \leq p \leq 1$ and $r = \eta_1(\boldsymbol{x}) - \eta_0(\boldsymbol{x})$,

$$\left|\log\left(1 + \sigma(\eta_0(\boldsymbol{x}))(e^{\eta_1(\boldsymbol{x})-\eta_0(\boldsymbol{x})} - 1)\right)\right| = |\log\left(1 + p(e^r - 1)\right)|$$

$$r > 0: \quad |\log(1 + p(e^r - 1))| = \log(1 + p(e^r - 1)) \leq \log(1 + (e^r - 1)) = r = |r|$$

$$r < 0: \quad |\log(1 + p(e^r - 1))| = -\log(1 + p(e^r - 1)) \leq -\log(1 + (e^r - 1)) = -r = |r|$$

Lemma B.0.3 *Let $p(\boldsymbol{\theta}_n|A) = MVN(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, where $\boldsymbol{\Sigma}_n$ is diagonal. Let $\mathcal{N}_\nu = \left\{\boldsymbol{\theta}_n : d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) < \nu\right\}$,*

$$d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) = \int_{\boldsymbol{x}\in[0,1]^{p_n}} \sigma(\eta_0(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x})) + \log\frac{1 - \sigma(\eta_0(\boldsymbol{x}))}{1 - \sigma(\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}))} d\boldsymbol{x}.$$

Assume conditions (C1),(C2),(C3) and (C4) hold, then

$$\int_{\boldsymbol{\theta}_n \in \mathcal{N}_{\nu\epsilon_n^2}} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n \geq e^{-\nu n\epsilon_n^2}$$

Proof. This proof uses somes ideas in the proof of theorem 1 in [74].

By condition (C3), let $\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})$ be the neural network such that $||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_1 \leq ||\eta_{\boldsymbol{\theta}_n^*} - \eta_0||_\infty \leq \nu\epsilon_n^2/4$. Define

$$\mathcal{N}^0_{\nu\epsilon_n^2} = \left\{\boldsymbol{\theta}_n : |\beta_j - \beta_j^*|, |\gamma_j^\top A\boldsymbol{x} - \gamma_j^{*\top}\boldsymbol{x}| < \frac{\nu\epsilon_n^2}{8(k_n + ||\beta^*||_1)}, j = 1, \cdots, k_n\right\}$$

For every $\boldsymbol{\theta}_n \in \mathcal{N}^0_{\nu\epsilon_n^2}$, by lemma B.0.1, we have

$$\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})| d\boldsymbol{x} \leq \frac{\nu\epsilon_n^2}{4} \tag{B.1}$$

For $\boldsymbol{\theta}_n \in \mathcal{N}^0_{\nu\epsilon_n^2}$, $\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_0(\boldsymbol{x})| d\boldsymbol{x} \leq \nu\epsilon_n^2/2$ which with lemma B.0.2 implies $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) \leq \nu\epsilon_n^2$. Thus, for every $\boldsymbol{\theta}_n \in \mathcal{N}^0_{\nu\epsilon_n^2}$, $\boldsymbol{\theta}_n \in \mathcal{N}_{\nu\epsilon_n^2}$ which implies

$$\int_{\boldsymbol{\theta}_n \in \mathcal{N}_{\nu\epsilon_n^2}} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n \geq \int_{\boldsymbol{\theta}_n \in \mathcal{N}^0_{\nu\epsilon_n^2}} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n$$

Let $\alpha_j = \boldsymbol{x}^\top A^\top \gamma_j$ and $\alpha_j^* = \boldsymbol{x}^\top \gamma_j^*$, then $\mu_{j\alpha} = \boldsymbol{x}^\top A^\top \boldsymbol{\mu}_{j\gamma}$ and $\sigma_{j\alpha}^2 = \boldsymbol{x}^\top A^\top \boldsymbol{\Sigma}_{j\gamma} A\boldsymbol{x}$. Also, using $|x_s| \leq 1$,

$$\alpha_j^{*2} \leq \boldsymbol{x}^\top \gamma_j^* \gamma_j^{*\top} \boldsymbol{x} \leq ||\gamma_j^*||_1^2 \quad \mu_{j\alpha}^2 = \boldsymbol{x}^\top A^\top \boldsymbol{\mu}_{j\gamma} \boldsymbol{\mu}_{j\gamma}^\top A\boldsymbol{x} \leq ||A^\top \boldsymbol{\mu}_{j\gamma}||_1^2$$

$$\sigma_{j\alpha}^2 = \boldsymbol{x}^\top A^\top \boldsymbol{\Sigma}_{j\gamma} A\boldsymbol{x} \geq \frac{1}{||\boldsymbol{\sigma}_{j\gamma}^*||_\infty^2} ||A\boldsymbol{x}||_2^2 \quad \sigma_{j\alpha}^2 \leq ||\boldsymbol{\sigma}_{j\gamma}||_\infty^2 ||A\boldsymbol{x}||_2^2$$

Let $\delta = \nu\epsilon_n^2/(8(k_n + ||\boldsymbol{\beta}^*||_1))$, then

$$\begin{aligned}
\int_{\boldsymbol{\theta}_n \in \mathcal{N}^0_{\nu\epsilon_n^2}} p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n &= \prod_{j=1}^{k_n} \int_{\beta_j^*-\delta}^{\beta_j^*+\delta} \frac{1}{\sqrt{2\pi\sigma_{j\beta}^2}} e^{-\frac{(\beta_j-\mu_{j\beta})^2}{2\sigma_{j\beta}^2}} d\beta_j \int_{\alpha_j^*-\delta}^{\alpha_j^*+\delta} \frac{1}{\sqrt{2\pi\sigma_{j\alpha}^2}} e^{-\frac{(\alpha_j-\mu_{j\alpha})^2}{2\sigma_{j\alpha}^2}} d\alpha_j \\
&= \prod_{j=1}^{k_n} \frac{2\delta}{\sqrt{2\pi\sigma_{j\beta}^2}} e^{-\frac{(\tilde{\beta}-\mu_{j\beta})^2}{2\sigma_{j\beta}^2}} \frac{2\delta}{\sqrt{2\pi\sigma_{j\alpha}^2}} e^{-\frac{(\tilde{\alpha}-\mu_{j\alpha})^2}{2\sigma_{j\alpha}^2}}, \quad \tilde{\beta}_j \in \beta_j^* \pm \delta, \tilde{\alpha}_j \in \alpha_j^* \pm \delta \\
&= e^{\left(-\sum_{j=1}^{k_n}\left(-\log\frac{2}{\pi} - 2\log\delta + \log\sigma_{j\beta} + \log\sigma_{j\alpha} + \frac{(\tilde{\beta}_j-\mu_{j\beta})^2}{2\sigma_{j\beta}^2} + \frac{(\tilde{\alpha}_j-\mu_{j\alpha})^2}{2\sigma_{j\alpha}^2}\right)\right)} \qquad \text{(B.2)}
\end{aligned}$$

where the second last equality holds by mean value theorem. Note that $\tilde{\beta}_j \in \beta_j^* \pm 1$ and $\tilde{\alpha}_j \in \alpha_j^* \pm 1$ since $\delta \longrightarrow 0$, therefore

$$\begin{aligned}
\frac{(\tilde{\beta}_j - \mu_{j\beta})^2}{2\sigma_{j\beta}^2} &\leq \frac{\max((\beta_j^* - \mu_{j\beta} - 1)^2, (\beta_j^* - \mu_{j\beta} + 1)^2)}{2\sigma_{j\beta}^2} \leq \frac{(\beta_j^* - \mu_{j\beta})^2}{\sigma_{j\beta}^2} + \frac{1}{\sigma_{j\beta}^2}, \\
\frac{(\tilde{\alpha}_j - \mu_{j\alpha})^2}{2\sigma_{j\alpha}^2} &\leq \frac{(\alpha_j^* - \mu_{j\alpha})^2}{\sigma_{j\alpha}^2} + \frac{1}{\sigma_{j\alpha}^2}
\end{aligned}$$

which further implies $\sum_{j=1}^{k_n}\left((\tilde{\beta} - \mu_{j\beta})^2/(2\sigma_{j\beta}^2) + (\tilde{\alpha} - \mu_{j\alpha})^2/(2\sigma_{j\alpha}^2)\right)$ is bounded above by

$$\begin{aligned}
&\leq 2\sum_{j=1}^{k_n} \frac{\beta_j^{*2}}{\sigma_{j\beta}^2} + 2\sum_{j=1}^{k_n} \frac{\mu_{j\beta}^2}{\sigma_{j\beta}^2} + \sum_{j=1}^{k_n} \frac{1}{\sigma_{j\alpha}^2} + 2\sum_{j=1}^{k_n} \frac{\alpha_j^{*2}}{\sigma_{j\alpha}^2} + 2\sum_{j=1}^{k_n} \frac{\mu_{j\alpha}^2}{\sigma_{j\alpha}^2} + \sum_{j=1}^{k_n} \frac{1}{\sigma_{j\alpha}^2} \\
&\lesssim (||\boldsymbol{\beta}^*||_1^2 + ||\boldsymbol{\mu}_\beta||_1^2)||\boldsymbol{\sigma}_\beta^{-1}||_\infty^2 + \frac{1}{||A\boldsymbol{x}||_2^2} \sum_{j=1}^{k_n} (||\boldsymbol{\gamma}_j^*||_1^2 + ||A^\top \boldsymbol{\mu}_{\gamma_j}||_1^2)||\sigma_{j\gamma}^{-1}||_\infty^2 = o(n\epsilon_n^2) \qquad \text{(B.3)}
\end{aligned}$$

where the last equality follows since $||\boldsymbol{\beta}^*||_1^2, ||\boldsymbol{\mu}_\beta||_1^2, 1/||A\boldsymbol{x}||_2^2 = o(n\epsilon_n^2)$,

$\sum_{j=1}^{k_n} ||\gamma_j^*||_1^2, \sum_{j=1}^{k_n} ||A^\top \boldsymbol{\mu}_{\gamma_j}||_1^2 = O(1)$ and $||\boldsymbol{\sigma}_\beta^{-1}||_\infty, \sup_{j=1,\cdots,k_n} ||\boldsymbol{\sigma}_{j\gamma}^{-1}||_\infty = O(1)$.

$$\begin{aligned}
\sum_{j=1}^{k_n}(-\log\frac{2}{\pi} - 2\log\delta + \log\sigma_{j\beta} + \log\sigma_{j\alpha}) \leq \sum_{j=1}^{k_n}(2\log 8 + \log(k_n + ||\boldsymbol{\beta}^*||_1) + \\
\log\sigma_{j\beta} + \log\sigma_{j\alpha} - 2\log\epsilon_n \\
\lesssim k_n(\log k_n + \log||\boldsymbol{\beta}^*||_1 + \log||\boldsymbol{\sigma}_\beta||_\infty - 2\log\epsilon_n) \\
+ \sum_{j=1}^{k_n} \log||\boldsymbol{\sigma}_{j\alpha}||_\infty = o(n\epsilon_n^2) \qquad \text{(B.4)}
\end{aligned}$$

where the last equality follows since $k_n \log n = o(n\epsilon_n^2)$, $\log ||\beta^*||_1 = O(\log n)$, $\log ||\boldsymbol{\sigma}_\beta||_\infty = O(\log n)$, $-\log \epsilon_n = o(\log n)$ and $\sum_{j=1}^{k_n} \log ||\boldsymbol{\sigma}_{j\alpha}||_\infty \leq k_n \log ||A\boldsymbol{x}|| + k_n \sum_{j=1}^{k_n} \log ||\boldsymbol{\sigma}_{j\gamma}||_\infty = O(k_n \log n) = o(n\epsilon_n^2)$.

Using (B.3) and (B.4) in (B.2), the proof follows.

Lemma B.0.4 *Let, $\widetilde{\mathcal{F}}_n = \{\sqrt{\ell} : \ell_{\boldsymbol{\theta}_n}(y, A\boldsymbol{x}), \boldsymbol{\theta}_n \in \mathcal{F}_n\}$ where $\ell_{\boldsymbol{\theta}_n}(y, A\boldsymbol{x})$ is given by*

$$\ell_{\boldsymbol{\theta}_n}(y, A\boldsymbol{x}) = \exp\left(y\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \log\left(1 + e^{\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x})}\right)\right) \tag{B.5}$$

and $\mathcal{F}_n = \left\{\boldsymbol{\theta}_n : |\theta_j| \leq C_n, j = 1, \cdots, \tilde{k}_n\right\}$ where $\tilde{k}_n = k_n d_n + 2k_n + 1 = O(k_n d_n)$. Then,

$$\int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} \sqrt{H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)}du \lesssim \varepsilon\sqrt{k_n d_n(\log k_n + (1/2)\log p_n + 2\log C_n - \log \varepsilon)}$$

where $H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)$ is the hellinger bracketing entropy of $\tilde{\mathcal{F}}_n$ (see definition in 3 in [74]).

Proof. This proof uses somes ideas in the proof of lemma 2 of [74].

In this proof, let $\boldsymbol{\theta} = \boldsymbol{\theta}_n$. Note, by lemma 4.1 in [104], $N(\varepsilon, \mathcal{F}_n, ||.||_\infty) \leq (3C_n/\varepsilon)^{\tilde{k}_n}$.

For $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{F}_n$, let $\widetilde{\ell}(u) = \sqrt{\ell_{u\boldsymbol{\theta}_1 + (1-u)\boldsymbol{\theta}_2}(A\boldsymbol{x}, y)}$.

$$\sqrt{\ell_{\boldsymbol{\theta}_1}(A\boldsymbol{x}, y)} - \sqrt{\ell_{\boldsymbol{\theta}_2}(A\boldsymbol{x}, y)} \leq \tilde{k}_n \sup_{j=1,\cdots,\tilde{k}_n} \left|\partial\widetilde{\ell}/\partial\theta_j\right| ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_\infty \leq F(A\boldsymbol{x}, y)||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||_\infty \tag{B.6}$$

where the upper bound $F(A\boldsymbol{x}, y) = \tilde{k}_n\sqrt{p}_n C_n$. This is because $|\partial\widetilde{\ell}/\partial\theta_j|$, the derivative of $\sqrt{\ell}$ w.r.t. is bounded above by $|\partial\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})/\partial\theta_j|$ as shown below.

$$\left|\frac{\partial\widetilde{\ell}}{\partial\theta_j}\right| \leq \frac{1}{2}\frac{\partial\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})}{\partial\theta_j}\left(\frac{e^{\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})}}{1 + e^{\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})}}\right)^{1/2}\left(\frac{1}{1 + e^{\eta_{\boldsymbol{\theta}}(A\boldsymbol{x})}}\right)^{1/2}$$

Thus, using $e^u/(1 + e^u)$, $1/(1 + e^{\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}) \leq 1$, we get

$$2\left|\frac{\partial\widetilde{\ell}}{\partial\theta_j}\right| \leq \left|\frac{\partial\eta_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial\theta_j}\right| \leq \begin{cases} 1, & \theta_j = \beta_r \text{ for some } r = 0, \cdots, k_n \\ |\beta_r\psi'(\gamma_r^\top Ax)[Ax]_{r'}|, & \theta_j = \gamma_{rr'} \text{ for some } r = 0, \cdots, k_n, r' = 0, \cdots, d_n \end{cases}$$

Note, $|\beta_r| \leq C_n$, $|\psi'(u)| \leq 1$ and $|[Ax]_{r'}| = |\sum_{s=1}^{p_n} a_{r's}x_j| \leq (\sum_{s=1}^{p_n} a_{r's}^2)^{1/2}(\sum_{s=1}^{p_n} x_s^2)^{1/2} \leq \sqrt{p_n}$ since A is orthonormal and $|x_s| \leq 1$. Hence the bound on $F(A\boldsymbol{x}, y)$ follows.

In view of (B.6) and theorem 2.7.11 in [126] (also see theorem 3 in [74] for more details), we have

$$N_{[]}(\varepsilon, \widetilde{\mathcal{F}}_n, ||.||_2) \leq \left(\frac{3\tilde{k}_n\sqrt{p}_n C_n^2}{2\varepsilon}\right)^{\tilde{k}_n} \implies H_{[]}(\varepsilon, \widetilde{\mathcal{F}}_n, ||.||_2) \lesssim \tilde{k}_n \log \frac{\tilde{k}_n\sqrt{p}_n C_n^2}{\varepsilon}$$

where $N_{[]}$ and $H_{[]}$ denote the bracketing number and bracketing entropy as in definition 3 of [74]. Using, the proof of lemma 1 in [74] (equation (34)) with $d_n = \tilde{k}_n$ and $C_n = \sqrt{p_n}C_n^2$, we get

$$\int_0^{\varepsilon} \sqrt{H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)}du \lesssim \varepsilon\sqrt{k_n d_n(\log k_n + (1/2)\log p_n + 2\log C_n - \log\varepsilon)}$$

$$\begin{aligned}\implies \int_{\varepsilon^2/8}^{\sqrt{2}\varepsilon} H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)du &\leq \int_0^{\sqrt{2}\varepsilon} H_{[]}(u, \widetilde{\mathcal{F}}_n, ||.||_2)du \\ &\lesssim \varepsilon\sqrt{\tilde{k}_n(\log \tilde{k}_n + (\log p_n)/2 + 2\log C_n - \log\varepsilon)}\end{aligned}$$

Lemma B.0.5 *Let* $q(\boldsymbol{\theta}_n|A) \sim MVN(\boldsymbol{m}_n, \boldsymbol{S}_n)$ *with* $\mu_{j\beta} = \beta_j^*$, $\sigma_{j\beta} = 1/\sqrt{n}$, $\boldsymbol{m}_{j\gamma} = A\gamma_j^*$ *and* $\boldsymbol{S}_{j\gamma} = I_{d_n}/(n||A\boldsymbol{x}||_2)^2$. *Define*

$$d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) = \int_{\boldsymbol{x}\in[0,1]^{p_n}} \sigma(\eta_0(\boldsymbol{x}))(\eta_0(\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x})) + \log\frac{1-\sigma(\eta_0(\boldsymbol{x}))}{1-\sigma(\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}))}d\boldsymbol{x}.$$

Suppose conditions (C1) and (C3) hold, then

$$\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n})q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \leq \nu\epsilon_n^2, \ \forall \nu > 0$$

Proof. Since $\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n})$ is a KL-distance, $d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) > 0$. We shall thus establish an upper bound. By lemma B.0.2, $\int d_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n})q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n$ is upper bounded by

$$\begin{aligned}&\leq 2\int |\eta_0(\boldsymbol{x}) - \eta_{\theta_n}(A\boldsymbol{x})|d\boldsymbol{x} \\ &\leq \int\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_0(\boldsymbol{x}) - \eta_{\theta_n^*}(\boldsymbol{x})|d\boldsymbol{x}q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n + \\ &\int\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x}) - \eta_{\theta_n}(A\boldsymbol{x})|d\boldsymbol{x}q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \\ &\leq \frac{\nu\epsilon_n^2}{2} + \int \underbrace{\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x}) - \eta_{\theta_n}(A\boldsymbol{x})|d\boldsymbol{x}}_{h(\boldsymbol{\theta}_n)}\, q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\end{aligned}$$

$$\begin{aligned}\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\eta_{\boldsymbol{\theta}_n}(A\boldsymbol{x}) - \eta_{\boldsymbol{\theta}_n^*}(\boldsymbol{x})|d\boldsymbol{x} &\leq \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\beta_0 - \beta_0^*|d\boldsymbol{x} \\ &+ \sum_{j=1}^{k_n} \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\beta_j\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \beta_j^*\psi(\boldsymbol{\gamma}_j^{*\top} x)|d\boldsymbol{x} \\ &\leq |\beta - \beta_0^*| + \sum_{j=1}^{k_n} \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\beta_j\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \beta_j^*\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x})|d\boldsymbol{x} \\ &+ \sum_{j=1}^{k_n} \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\beta_j^*\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \beta_j^*\psi(\boldsymbol{\gamma}_j^{*\top} x)|d\boldsymbol{x} \\ &\leq \sum_{j=0}^{k_n} |\beta_j - \beta_j^*| \\ &+ ||\boldsymbol{\beta}^*||_1 \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \psi(\boldsymbol{\gamma}_j^* \top \boldsymbol{x})|d\boldsymbol{x}\end{aligned}$$

Therefore,

$$\begin{aligned}\int h(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \leq \sum_{j=0}^{k_n} &|\beta_j - \beta_j^*|q(\beta_j)d\beta_j + \\ ||\boldsymbol{\beta}^*||_1 &\int \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \psi(\boldsymbol{\gamma}_j^* \top \boldsymbol{x})|d\boldsymbol{x}q(\boldsymbol{\gamma}_j)d\boldsymbol{\gamma}_j \\ &= \frac{k_n}{n}\sqrt{\frac{2}{\pi}} + ||\boldsymbol{\beta}^*||_1 \int \int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\boldsymbol{\gamma}_j^\top A\boldsymbol{x}) - \psi(\boldsymbol{\gamma}_j^{*\top} \boldsymbol{x})|d\boldsymbol{x}q(\boldsymbol{\gamma}_j)d\boldsymbol{\gamma}_j\end{aligned} \tag{B.7}$$

Now, let $M_j = \{\gamma : |\gamma_j^\top A\boldsymbol{x} - \gamma_j^{*\top}\boldsymbol{x}| \leq \nu\epsilon_n^2/(16||\boldsymbol{\beta}^*||_1)\}$, then

$$
\begin{aligned}
&\int\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\gamma_j^\top A\boldsymbol{x}) - \psi(\gamma_j^{*\top}\boldsymbol{x})|d\boldsymbol{x}q(\gamma_j)d\boldsymbol{\gamma}_j \\
&= \int_{M_j}\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\gamma_j^\top A\boldsymbol{x}) - \psi(\gamma_j^{*\top}\boldsymbol{x})|d\boldsymbol{x}q(\gamma_j)d\boldsymbol{\gamma}_j + \\
&\int_{M_j^c}\int_{\boldsymbol{x}\in[0,1]^{p_n}} |\psi(\gamma_j^\top A\boldsymbol{x}) - \psi(\gamma_j^{*\top}\boldsymbol{x})|d\boldsymbol{x}q(\gamma_j)d\boldsymbol{\gamma}_j \\
&\leq \frac{\nu\epsilon_n^2}{8||\boldsymbol{\beta}^*||_1} + 2Q_{\gamma_j}(M_j^c)
\end{aligned}
\tag{B.8}
$$

Thus, combining (B.7) and (B.8) and using $k_n = o(n\epsilon_n^2)$, we get

$$
\int h(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \leq \frac{\nu\epsilon_n^2}{4} + \frac{\nu\epsilon_n^2}{8} + 2||\boldsymbol{\beta}^*||_1 Q_{\gamma_j}(M_j^c)
\tag{B.9}
$$

In the next steps, we deal with $Q_{\gamma_j}(M_j^c)$. Let $\delta = \nu\epsilon_n^2/(16||\boldsymbol{\beta}||_1^*)$

$$
P(|\gamma_j^\top A\boldsymbol{x} - \gamma_j^{*\top}\boldsymbol{x}| > \delta) = P(|\alpha_j - \alpha_j^*| \geq \delta)
\tag{B.10}
$$

where $\alpha_j = \boldsymbol{x}^\top A^\top \gamma_j$ and $\alpha_j^* = \boldsymbol{x}^\top \gamma_j^*$. Note that $\alpha_j - \alpha_j^* \sim N(\mu_{j\alpha}, \sigma_{j\alpha}^2)$ with $\mu_{j\alpha} = \boldsymbol{x}^\top A^\top A \gamma_j^* - x^\top \gamma_j^* = x^\top(A^\top A - I)\gamma_j^*$ and $\sigma_{j\alpha}^2 = (1/(n^2||A\boldsymbol{x}||_2^2))||A\boldsymbol{x}||_2^2 = 1/n^2$. Further note that since $|x_s| \leq 1$,

$$
|\mu_{j\alpha}| = \sum_{s=1}^{p_n} |x_s||[(I - A^\top A)\gamma_j^*]_s| \leq \sum_{s=1}^{p_n} |[(I - A^\top A)\gamma_j^*]_s| = ||(I - A^\top A)\gamma^*||_1 = o(n^{-1}) = o(\delta)
$$

where the last equality holds since $\delta \sim \epsilon_n^2/||\boldsymbol{\beta}^*||_1 \geq n^{-1}$ because $||\boldsymbol{\beta}^*||_1^2 = o(n\epsilon_n^2)$. This also implies, $(\delta \pm \mu_{j\alpha})/\sigma_{j\alpha} \sim (n\epsilon_n^2)/||\boldsymbol{\beta}^*||_1 \geq \sqrt{n}\epsilon_n \to \infty$ which implies

$$
\begin{aligned}
P(|\gamma_j^\top A\boldsymbol{x} - \gamma_j^{*\top}\boldsymbol{x}| > \delta) &= 1 - \Phi((\delta - \mu_{j\alpha})/\sigma_{j\alpha})) + 1 - \Phi((\delta + \mu_{j\alpha})/\sigma_{j\alpha}) \\
&\sim (\sigma_{j\alpha}/(\delta - \mu_{j\alpha}))\phi((\delta - \mu_{j\alpha})/\sigma_{j\alpha}) + \\
&(\sigma_{j\alpha}/(\delta + \mu_{j\alpha}))\phi((\delta + \mu_{j\alpha})/\sigma_{j\alpha}) \lesssim e^{-n\epsilon_n^2}
\end{aligned}
\tag{B.11}
$$

where the asymptotic equivalence in the second step is a consequence of Mill'ratio. Thus, using the above relation in (B.9),

$$
\int h(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \leq \nu\left(\frac{\epsilon_n^2}{4} + \frac{\epsilon_n^2}{8} + 2||\boldsymbol{\beta}^*||_1 e^{-n\epsilon_n^2}\right) \leq \frac{\nu\epsilon_n^2}{2}
$$

where the last equality holds $||\boldsymbol{\beta}^*||_1 = o(\sqrt{n\epsilon_n^2})$.

Lemma B.0.6 *Let $q(\boldsymbol{\theta}_n|A) \sim MVN(\boldsymbol{m}_n, \boldsymbol{S}_n)$ with $m_{j\beta} = \beta_j^*$, $s_{j\beta} = 1/\sqrt{n}$, $\boldsymbol{m}_{j\gamma} = A\gamma_j^*$ and $\boldsymbol{S}_{j\gamma} = I_{d_n}/(n||A\boldsymbol{x}||_2)^2$. Let, $p(\boldsymbol{\theta}_n|A) = MVN(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, where $\boldsymbol{\Sigma}_n$ is diagonal. Suppose conditions (C1), (C2), (C3) and (C4) hold, then*

$$d_{\text{KL}}(q, p) = o(n\epsilon_n^2)$$

Proof: With $\tilde{k}_n \sim k_n + 1 + k_n(d_n + 1)$, here $d_{\text{KL}}(q, p)$ can be simplified as

$$\begin{aligned}
&= \sum_{j=1}^{k_n}(\log \sqrt{n}\sigma_{j\beta} + \frac{1}{n\sigma_{j\beta}^2} + \frac{(\beta_j^* - \mu_{j\beta})^2}{\sigma_{j\beta}^2} + \sum_{j'=1}^{d_n}(2\log n||A\boldsymbol{x}||_2\sigma_{jj'\gamma} + \frac{1}{n^2||A\boldsymbol{x}||_2^2}+ \\
&\frac{(\gamma_j^* - \mu_{jj'\gamma})^2}{\sigma_{j\gamma}^2})) - \frac{\tilde{k}_n}{2} \\
&\lesssim \tilde{k}_n(\log n + \log ||A\boldsymbol{x}||_2 + \log ||\sigma_\beta||_\infty + 1/(n||A\boldsymbol{x}||_2)^2)+ \\
&k_n \sum_{j=1}^{k_n} \log ||\sigma_{j\gamma}||_\infty + ||\sigma_\beta^*||_\infty(||\boldsymbol{\beta}^*||_2^2 + ||\boldsymbol{\mu}_\beta||_2^2) \\
&+ \sum_{j=1}^{k_n} ||\sigma_{j\gamma}^*||_\infty(||\gamma_j^*||_2^2 + ||\boldsymbol{\mu}_{j\gamma}||_2^2) - \frac{\tilde{k}_n}{2} = o(n\epsilon_n^2)
\end{aligned} \tag{B.12}$$

where the last equality holds since $\tilde{k}_n \log n = o(n\epsilon_n^2)$, $\log ||A\boldsymbol{x}||_2 = \log ||\sigma_\beta||_\infty = \log ||\sigma_{j\gamma}||_\infty = O(\log n)$ and $1/||A\boldsymbol{x}||_2^2 = o(n\epsilon_n^2)$. Since $||.||_2 \leq ||.||_1$, $||\boldsymbol{\beta}^*||_2^2 = ||\boldsymbol{\mu}_\beta||_2^2 = o(n\epsilon_n^2)$, $\sum_{j=1}^{k_n} ||\gamma_j^*||_2^2 = O(1)$, $\sum_{j=1}^{k_n} ||\boldsymbol{\mu}_{j\gamma}||_2^2 = \sum_{j=1}^{k_n} ||A^\top \boldsymbol{\mu}_{j\gamma}||_2^2 = O(1)$, as consequence of which the proof follows.

Lemma B.0.7 *Let $p(\boldsymbol{\theta}_n|A) = MVN(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, where $\boldsymbol{\Sigma}_n$ is diagonal. Suppose conditions (C1) and (C2) hold, then*

$$\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \leq e^{-n\nu\epsilon_n^2}, \ \forall \nu > 0$$

where $\mathcal{F}_n = \{\boldsymbol{\theta}_n : |\theta_j| \leq C_n, j = 1, \cdots, \tilde{k}_n\}$ with $C_n = e^{\varepsilon n\epsilon_n^2/\tilde{k}_n}$.

Proof: This proof uses somes ideas in the proof of theorem 1 in [74]. Let $\mathcal{F}_{jn} = \{\theta_j : |\theta_j| \leq C_n\}$ which implies $\mathcal{F}_n = \cap_{j=1}^{\tilde{k}_n}\mathcal{F}_{jn} \implies \mathcal{F}_n^c = \cap_{j=1}^{\tilde{k}_n}\mathcal{F}_{jn}^c$. Note that $\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n$ is bounded above by $\sum_{j=1}^{\tilde{k}_n} P(\mathcal{F}_{jn}^c)$ which is

$$
\begin{aligned}
&= \sum_{j=0}^{k_n} \int_{(-\infty, C_n)\cup(C_n,\infty)} \frac{1}{\sqrt{2\pi\sigma_{j\beta}^2}} e^{-\frac{1}{2\sigma_{j\beta}^2}(\beta_j - \mu_{j\beta})^2} d\beta_j + \\
&\sum_{j=1}^{k_n} \sum_{j'=1}^{d_n} \int_{(-\infty, C_n)\cup(C_n,\infty)} \frac{1}{\sqrt{2\pi\sigma_{jj'\gamma}^2}} e^{-\frac{1}{2\sigma_{jj'\gamma}^2}(\gamma_{jj'} - \mu_{jj'\gamma})^2} d\gamma_{jj'} \\
&= \sum_{j=0}^{k_n} (2 - \Phi((C_n - \mu_{j\beta})/\sigma_{j\beta}) - \Phi((C_n + \mu_{j\beta})/\sigma_{j\beta})) \\
&+ \sum_{j=1}^{k_n} \sum_{j'=1}^{d_n} (2 - \Phi((C_n - \mu_{jj'\gamma})/\sigma_{jj'\gamma}) - \Phi((C_n + \mu_{jj'\gamma})/\sigma_{jj'\gamma})) \\
&= \sum_{j=0}^{k_n} (\sigma_{j\beta}/(C_n - \mu_{j\beta}))\phi((C_n - \mu_{j\beta})/\sigma_{j\beta}) + \sum_{j=0}^{k_n} (\sigma_{j\beta}/(C_n + \mu_{j\beta}))\phi((C_n + \mu_{j\beta})/\sigma_{j\beta}) \\
&+ \sum_{j=1}^{k_n} \sum_{j'=1}^{d_n} (\sigma_{jj'\gamma}/(C_n + \mu_{jj'\gamma}))\phi((C_n + \mu_{jj'\gamma})/\sigma_{jj'\gamma}) \\
&+ \sum_{j=1}^{k_n} \sum_{j'=1}^{d_n} (\sigma_{jj'\gamma}/(C_n - \mu_{jj'\gamma}))\phi((C_n - \mu_{jj'\gamma})/\sigma_{jj'\gamma})
\end{aligned}
$$

where the above equality holds due Mill's ratio and the fact $\mu_{j\beta}, \mu_{jj'\gamma} = o(\sqrt{n\epsilon_n^2})$, $\sigma_{j\beta}, \sigma_{jj'\gamma} = O(n^r), r > 0$ which implies $(C_n \pm \mu)/\sigma \geq (C_n - \sqrt{n})/(Mn^r) \sim C_n/n^r = \exp((n\epsilon_n^2/\tilde{k}_n)(\varepsilon - (r\tilde{k}_n \log n)/(n\epsilon_n^2))) \longrightarrow \infty$ since $\tilde{k}_n \log n = o(n\epsilon_n^2)$. Further,

$$
\begin{aligned}
\int_{\boldsymbol{\theta}_n \in \mathcal{F}_n^c} p(\boldsymbol{\theta}_n | A) d\boldsymbol{\theta}_n &\lesssim \sum_{j=1}^{k_n} (e^{-(C_n - \mu_{j\beta})^2/(2\sigma_{j\beta}^2)} + e^{-(C_n + \mu_{j\beta})^2/(2\sigma_{j\beta}^2)}) \\
&+ \sum_{j=1}^{k_n} \sum_{j'=1}^{d_n} (e^{-(C_n - \mu_{jj'\gamma})^2/(2\sigma_{jj'\gamma}^2)} + e^{-(C_n + \mu_{jj'\gamma})^2/(2\sigma_{jj'\gamma}^2)}) \\
&\sim \tilde{k}_n \exp(-\exp((n\epsilon_n^2/\tilde{k}_n)(\varepsilon - (r\tilde{k}_n \log n)/(n\epsilon_n^2)))) \leq e^{-\nu n\epsilon_n^2}
\end{aligned}
$$

since $(n\epsilon_n^2/\tilde{k}_n)(\varepsilon - (r\tilde{k}_n \log n)/(n\epsilon_n^2)) \geq \log(\nu n\epsilon_n^2)$ when $\tilde{k}_n \log n = o(n\epsilon_n^2)$ and $n\epsilon_n^2 \longrightarrow 0$.

Proof of Theorem 1

This proof uses some ideas in the proof of lemmas 3 and lemma 5 in [74].

Let $\mathcal{D}^A = (\boldsymbol{y}_n, A\boldsymbol{x}_1, \cdots, A\boldsymbol{x}_n)$, then

$$\begin{aligned}
& d_{\mathrm{KL}}(q_A^*, \pi(.|\mathcal{D}^A)) \\
= \;& \int_{\mathcal{U}_{\varepsilon\epsilon_n}} q_A^*(\boldsymbol{\theta}_n) \log \frac{q_A^*(\boldsymbol{\theta}_n)}{\pi(\boldsymbol{\theta}_n|\mathcal{D}^A)} d\boldsymbol{\theta}_n + \int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} q_A^*(\boldsymbol{\theta}_n) \log \frac{q_A^*(\boldsymbol{\theta}_n)}{\pi(\boldsymbol{\theta}_n|\mathcal{D}^A)} d\boldsymbol{\theta}_n \\
= \;& -q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}) \int_{\mathcal{U}_{\varepsilon\epsilon_n}} \frac{q_A^*(\boldsymbol{\theta}_n)}{q_A^*(\mathcal{U}_{\varepsilon\epsilon_n})} \log \frac{\pi(\boldsymbol{\theta}_n|\mathcal{D}^A)}{q_A^*(\boldsymbol{\theta}_n)} d\boldsymbol{\theta}_n \\
- \;& q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} \frac{q_A^*(\boldsymbol{\theta}_n)}{q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c)} \log \frac{\pi(\boldsymbol{\theta}_n|\mathcal{D}^A)}{q_A^*(\boldsymbol{\theta}_n)} d\boldsymbol{\theta}_n \\
\geq \;& q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}) \log \frac{q_A^*(\mathcal{U}_{\varepsilon\epsilon_n})}{\pi(\mathcal{U}_{\varepsilon\epsilon_n}|\mathcal{D}^A)} + q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \log \frac{q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c)}{\pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\mathcal{D}^A)}, \quad \text{by Jensen's inequality} \\
\geq \;& q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}) \log q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}) + q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \log q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) - q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \log \pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\mathcal{D}^A) \\
\geq \;& -q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \log \pi(\mathcal{U}_{\varepsilon\epsilon_n}^c|\mathcal{D}^A) - \log 2, \quad \text{since } x \log x + (1-x)\log(1-x) \geq -\log 2 \\
= \;& -q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) \left(\log \int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} \frac{L(\boldsymbol{\theta}_n|A)}{L_0} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n - \log \int \frac{L(\boldsymbol{\theta}_n|A)}{L_0} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n \right) - \log 2
\end{aligned}$$

Let $E_{1n} = \log \int_{\mathcal{U}_{\varepsilon\epsilon_n}^c} (L(\boldsymbol{\theta}_n|A)/L_0) p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n$, $E_{2n} = \log \int (L(\boldsymbol{\theta}_n|A)/L_0) p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n$ and $E_{3n} = \int \log(L_0/L(\boldsymbol{\theta}_n|A)) q(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n$. Then for any $q \in \mathcal{Q}_n$,

$$\begin{aligned}
-q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) E_{1n} &\leq d_{\mathrm{KL}}(q, \pi(.|\mathcal{D}^A)) - q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) E_{2n} + \log 2 \\
&= d_{\mathrm{KL}}(q, p(.|A)) - \int \log \frac{L(\boldsymbol{\theta}_n|A)}{L_0} q(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n + \\
&\log \int \frac{L(\boldsymbol{\theta}_n|A)}{L_0} p(\boldsymbol{\theta}_n|A) d\boldsymbol{\theta}_n + |E_{2n}| + \log 2 \qquad \text{(B.13)} \\
&\leq d_{\mathrm{KL}}(q, p(.|A)) + E_{3n} + (1 - q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c)) E_{2n} + \log 2 \\
&\leq o(n\epsilon_n^2) + E_{3n} + E_{2n} + \log 2 \qquad \text{(B.14)}
\end{aligned}$$

where the above inequality holds due to lemma B.0.6.

We show three main things, $E_{3n} = o_{P_0^n}(n\epsilon_n^2)$, $E_{2n} = o_{P_0^n}(n\epsilon_n^2)$ and $E_{1n} \geq \log 2 - n\varepsilon^2\epsilon_n^2 + o_{P_0^n}(1)$. This completes the proof because

$$q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) n\varepsilon^2\epsilon_n^2 \leq o(n\epsilon_n^2) + o_{P_0^n}(n\epsilon_n^2) + o_{P_0^n}(n\epsilon_n^2) + O_{P_0^n}(1) \implies q_A^*(\mathcal{U}_{\varepsilon\epsilon_n}^c) = o_{P_0^n}(1)$$

Handling E_{3n}: Note, $P_0^n(|E_{3n}| > \varepsilon n\epsilon_n^2)$ can be bounded above using Markov's inequality as

$$\begin{aligned}\frac{1}{\varepsilon n\epsilon_n^2}E_0^n\left(\left|\int q(\boldsymbol{\theta}_n|A)\log(L_0/L(\boldsymbol{\theta}_n|A))d\boldsymbol{\theta}_n\right|\right) &\leq \frac{1}{\varepsilon n\epsilon_n^2}E_0^n\left(\int q(\boldsymbol{\theta}_n|A)\left|\log(L_0/L(\boldsymbol{\theta}_n|A))\right|d\boldsymbol{\theta}_n\right)\\ &\leq \frac{1}{\varepsilon n\epsilon_n^2}\int q(\boldsymbol{\theta}_n|A)\int\left|\log(L_0/L(\boldsymbol{\theta}_n|A))\right|L_0 d\mu d\boldsymbol{\theta}_n\\ &\leq \int q(\boldsymbol{\theta}_n|A)\left(d_{KL}(L_0, L(\boldsymbol{\theta}_n|A)) + 2/e\right)d\boldsymbol{\theta}_n\\ &\leq \frac{1}{\varepsilon n\epsilon_n^2}\int(nd_{\mathrm{KL}}(\ell_0, \ell_{\boldsymbol{\theta}_n}) + 2/e)d\boldsymbol{\theta}_n \leq \nu/\varepsilon\end{aligned}$$

where the third step follows from lemma 4 in in [74] and the fourth step follows from lemma B.0.5. Since ν is arbitrary, $E_{3n} = o_{P_0^n}(n\epsilon_n^2)$. We next shown $E_{2n} = o_{P_0^n}(n\epsilon_n^2)$ as follows.

Handling E_{2n}: Note, $P_0^n(|E_{2n}| > \varepsilon n\epsilon_n^2)$ can be bounded above using Markov's inequality as

$$\begin{aligned}\frac{1}{\varepsilon n\epsilon_n^2}E_0^n\left(\left|\log\int(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\right|\right) &= \frac{1}{\varepsilon n\epsilon_n^2}\int\left|\log\int(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\right|L_0 d\mu\\ &\leq \frac{1}{\varepsilon n\epsilon_n^2}\left(d_{\mathrm{KL}}(L_0, L^*) + (2/e)\right)\end{aligned}$$

With $L^* = \int L(\boldsymbol{\theta}_n|A)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n$, the last equality follows from lemma 4 in [74]. Further,

$$\begin{aligned}d_{\mathrm{KL}}(L_0, L^*) = E_0^n\left(\log(L_0/L^*)\right) &= E_0^n\left(\log\left(L_0/\int L(\boldsymbol{\theta}_n|A)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\right)\right)\\ &\leq E_0^n\left(\log(L_0/\int_{N_{\nu\epsilon_n^2}} L(\boldsymbol{\theta}_n|A)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n)\right)\\ &\leq \int_{N_{\nu\epsilon_n^2}} p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n + \int_{N_{\nu\epsilon_n^2}} d_{KL}(L_0, L(\boldsymbol{\theta}_n|A))p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\\ &\leq -\log e^{-\nu n\epsilon_n^2} + \nu n\epsilon_n^2 = 2\nu n\epsilon_n^2\end{aligned}$$

where the second step follows from Jensens' inequality and the last step follows from lemma B.0.3. Lastly, we show $E_{1n} \geq -\log 2 + n\varepsilon^2\epsilon_n^2 + o_{P_0^n}(1)$ as follows.

Handling E_{1n}: For this, $\mathcal{F}_n = \{\boldsymbol{\theta}_n : |\theta_j| \leq E_{3n} = e^{n\varepsilon\epsilon_n^2/\tilde{k}_n}\}$. Thus, $P_0^n(E_{1n} \leq \log 2 - n\varepsilon^2\epsilon_n^2)$ is bounded above by

$$\underbrace{P_0^n\left(\int_{\mathcal{U}^c_{\varepsilon\epsilon_n}\cap\mathcal{F}_n}(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \geq e^{-n\varepsilon^2\epsilon_n^2}\right)+}_{E_{11n}}$$

$$\underbrace{P_0^n\left(\int_{\mathcal{F}_n^c}(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \geq e^{-n\varepsilon^2\epsilon_n^2}\right)}_{E_{12n}}$$

Using lemma B.0.4 with $\varepsilon = \varepsilon\epsilon_n$ and $C_n = e^{n\varepsilon\epsilon_n^2/\tilde{k}_n}$

$$\int_{\varepsilon^2\epsilon_n^2/8}^{\sqrt{2}\varepsilon\epsilon_n} H_{[]}(u,\widetilde{\mathcal{F}}_n,||.||_2)du \lesssim \varepsilon\epsilon_n\sqrt{\tilde{k}_n(\log k_n + (1/2)\log p_n + 2\log C_n - \log\epsilon_n)} \leq \varepsilon^2\epsilon_n^2\sqrt{n}$$

where the above equality holds since $\tilde{k}_n \log n = o(n\epsilon_n^2)$, $p_n = o(e^{n\epsilon_n^2/\tilde{k}_n})$ and $n\epsilon_n^2 \to \infty$. Therefore, by theorem 1 in [138], we have $E_{11n} \to 0, n \to \infty$.

Additionally, $P_0^n\left(\int_{\mathcal{F}_n^c}(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n \geq e^{-n\varepsilon^2}\right)$ is bounded above by Markov's inequality by

$$e^{n\varepsilon^2\epsilon_n^2}E_0^n\left(\int_{\mathcal{F}_n^c}(L(\boldsymbol{\theta}_n|A)/L_0)p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n\right) = e^{n\varepsilon^2\epsilon_n^2}\int_{\mathcal{F}_n^c}p(\boldsymbol{\theta}_n|A)d\boldsymbol{\theta}_n = e^{n\epsilon_n^2(\varepsilon^2-2\varepsilon^2)} \to 0$$

where the above equality holds due to lemma B.0.7 for $\nu = 2\varepsilon^2$. Thus, $E_{12n} \to 0, n \to \infty$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ailon, N., and Chazelle, B. Approximate nearest neighbours and the fast johnson-lindenstrauss transform. *Proceedings of the Symposium on Theory of Computing* (2006), 557–563.

[2] Allison, J., Rivers, R., Christodoulou, J., Vendruscolo, M., and Dobson, C. A relationship between the transient structure in the monomeric state and the aggregation propensities of α-synuclein and β-synuclein. *Biochemistry 53* (11 2014).

[3] Bai, J., Song, Q., and Cheng, G. Efficient variational inference for sparse deep learning with theoretical guarantee. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 466–476.

[4] Barron, A., Schervish, M. J., and Wasserman, L. The consistency of posterior distributions in nonparametric problems. *Ann. Statist. 27*, 2 (1999), 536–561.

[5] Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory 39*, 3 (1993), 930–945.

[6] Bhattacharya, S., and Maiti, T. Statistical foundation of variational bayes neural networks. *Neural Networks 137* (2021), 151–173.

[7] Bishop, C. M. Bayesian Neural Networks. *Journal of the Brazilian Computer Society 4*, 1 (1997), 61–68.

[8] Blei, D., Ng, A., and Jordan, M. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[9] Blei, D. M., and Lafferty, J. D. A correlated topic model of science. *The Annals of Applied Statistics 1*, 1 (2007), 17–35.

[10] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network.

[11] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *Proceedings of Machine Learning Research*, vol. 37. PMLR, 2015, pp. 1613–1622.

[12] Breiman, L. Bagging predictors. *Machine Learning 24*, 2 (Aug. 1996), 123–140.

[13] Candés, E. J., Romberg, J. K., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics 59*, 8 (2006), 1207–1223.

[14] Cannings, T. I., and Samworth, R. J. Random-projection ensemble classification. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*, 4 (2017), 959–1035.

[15] Carbonetto, P., and Stephens, M. Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis 7* (2012).

[16] Carvalho, C. M., Polson, N. G., and Scott, J. G. Handling sparsity via the horseshoe. D. van Dyk and M. Welling, Eds., vol. 5 of *Proceedings of Machine Learning Research*, PMLR, pp. 73–80.

[17] Casella, G., and Robert, C. P. Rao-blackwellisation of sampling schemes. *Biometrika 83*, 1 (1996), 81–94.

[18] Chapman, R., Mapstone, M., McCrary, J., Gardner, M., Porsteinsson, A., Sandoval, T., Guillily, M., Degrush, E., and Reilly, L. Predicting conversion from mild cognitive impairment to alzheimer's disease using neuropsychological tests and multivariate methods. *Journal of clinical and experimental neuropsychology 33* (02 2011), 187–99.

[19] Chen, S.-T., Hsiao, Y.-H., Huang, Y.-L., Kuo, S.-J., Tseng, H.-S., Wu, H.-K., and Chen, D.-R. Comparative analysis of logistic regression, support vector machine and artificial neural network for the differential diagnosis of benign and malignant solid breast tumors by the use of three-dimensional power doppler imaging. *Korean journal of radiology : official journal of the Korean Radiological Society 10* (08 2009), 464–71.

[20] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., and Zhang, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems.

[21] Cheng, B., Zhang, D., and Shen, D. Domain transfer learning for mci conversion prediction. vol. 15, pp. 82–90.

[22] Chérief-Abdellatif, B.-E. Convergence rates of variational inference in sparse deep learning. In *Proceedings of the 37th International Conference on Machine Learning* (13–18 Jul 2020), H. D. III and A. Singh, Eds., vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 1831–1842.

[23] Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[24] Cui, Y., Liu, B., Luo, S., Zhen, X., Fan, M., Liu, T., Zhu, W., Park, M., Jiang, T., Jin, J. S., and the Alzheimer's Disease Neuroimaging Initiative. Identification of conversion from mild cognitive impairment to alzheimer's disease using multivariate predictors. *PLOS ONE 6*, 7 (07 2011), 1–10.

[25] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems 2*, 4 (1989), 303–314.

[26] Dasgupta, S. Experiments with random projection. *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (2013).

[27] Dasgupta, S., and Gupta, A. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms 22*, 1 (2003), 60–65.

[28] Davatzikos, C., Bhatt, P., Shaw, L., Batmanghelich, K., and Trojanowski, J. Prediction of mci to ad conversion, via mri, csf biomarkers, and pattern classification. *Neurobiol Aging* (2011).

[29] Devanand, D., Liu, X., Tabert, M., Pradhaban, G., Cuasay, K., Bell, K., Mony, J., Doty, R., Stern, Y., and Pelton, G. Combining early markers strongly predicts conversion from mild cognitive impairment to alzheimer's disease. *Biological psychiatry 64* (09 2008), 871–9.

[30] Diego Alejandro Salazar, Jorge Ivan Velez, J. C. S. Comparison between svm and logistic regression: Which one is better to discriminate? *Revista Colombiana de Estadistica Numero especial en Bioestadistica 35* (06 2012), 223–237.

[31] Ding, J., and Huang, Q. Prediction of mci to ad conversion using laplace eigenmaps learned from fdg and mri images of ad patients and healthy controls. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)* (2017), pp. 660–664.

[32] Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. A survey on ensemble learning. *Frontiers of Computer Science 14* (2019), 241 – 258.

[33] Donoho, D. L. Compressed sensing. *IEEE Transactions on Information Theory 52*, 4 (2006), 1289–1306.

[34] Doshi, J., Erus, G., Ou, Y., Gaonkar, B., and Davatzikos, C. Multi-atlas skull-stripping. *Acad Radiol* (2013), 1566–1576.

[35] Doshi, J., Erus, G., Ou, Y., Resnick, S., Gur, R., Gur, R., Satterthwaite, T., and Davatzikos, C. Muse: Multi-atlas region segmentation utilizing ensembles of registration algorithms and parameters, and locally optimal atlas selection. *NeuroImage 127* (12 2015).

[36] Doshi, J., Erus, G., Rozycki, M., and Davatzikos, C. Hierarchical parcellation of mri using multi-atlas labeling methods. *Alzheimer's Disease Neuroimaging Initiative*.

[37] Dreiseitl, S., and Ohno-Machado, L. Logistic regression and artificial neural network classification models: A methodology review. *Journal of biomedical informatics 35* (10 2002), 352–9.

[38] Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*, 61 (2011), 2121–2159.

[39] Durrant, R., and Kaban, A. Sharp generalization error bounds for randomly-projected classifiers. In *Proceedings of the 30th International Conference on Machine Learning* (2013), S. Dasgupta and D. McAllester, Eds., vol. 28, PMLR, pp. 693–701.

[40] Durrant, R. J., and Kabán, A. Random projections as regularizers: Learning a linear discriminant from fewer observations than dimensions. *Machine Learning 99*, 2 (2015), 257–286.

[41] E, H. G., and Drew, V. C. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory, COLT'93*. ACM press, 1993, p. 5–13.

[42] Eckerstrom, C., Olsson, E., Borga, M., Ekholm, S., Ribbelin, S., Rolstad, S., Starck, G., Edman, A., Wallin, A., and Malmgren, H. Small baseline volume of left hippocampus is associated with subsequent conversion of mci into dementia: Th goteborg mci study. *J Neurol Sci 271(2)* (2008), 48–59.

[43] Ewers, M., Walsh, C., Trojanowski, J., Shaw, L., Petersen, R., Jack, C., Feldman, H., Bokde, ALand Alexander, G., Scheltens, P., Vellas, B., Dubois, B., Weiner, M., and Hampel, H. Prediction of conver- sion from mild cognitive impairment to alzheimer's disease dementia based upon biomarkers and neuropsychological test performance. *Neurobiol Aging 33(7)* (2012), 1203–1214.

[44] Farlow, M. Treatment of mild cognitive impairment (mci). *Curr. Alzheimer Res 6(4)* (2009), 273–297.

[45] Feng, J., and Simon, N. Sparse-input neural networks for high-dimensional nonparametric regression and classification, 2019.

[46] Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*. Springer series in statistics. Springer, New York, 2009.

[47] Ghosal, S., Ghosh, J. K., and van der Vaart, A. W. Convergence rates of posterior distributions. *The Annals of Statistics 28*, 2 (2000), 500 – 531.

[48] Ghosh, S., Yao, J., and Doshi-Velez, F. Model selection in bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research 20*, 182 (2019), 1–46.

[49] Goel, N., Bebis, G., and Nefian, A. Face recognition experiments with random projection. In *Proceedings of SPIE - The International Society for Optical Engineering* (2005), vol. 5776.

[50] Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011, pp. 2348–2356.

[51] Graves, A. Generating sequences with recurrent neural networks, 2014. arXiv.1308.0850.

[52] Guhaniyogi, R., and Dunson, D. B. Bayesian compressed regression. *Journal of the American Statistical Association 110*, 512 (2015), 1500–1514.

[53] Guhaniyogi, R., and Dunson, D. B. Compressed gaussian process for manifold regression. *Journal of Machine Learning Research 17*, 69 (2016), 1–26.

[54] Gurney, K. *An Introduction to Neural Networks*. Taylor & Francis, Inc., USA, 1997.

[55] Heinze, C., McWilliams, B., Meinshausen, N., and Krummenacher, G. Loco: Distributing ridge regression with random projections, 2015. arXiv:1406.3469.

[56] Hinrichs, C., Singh, V., Xu, G., and Johnson, S. Predictive markers for ad in a multi-modality framework: An analysis of mci progression in the adni population. *NeuroImage 55* (03 2011), 574–89.

[57] Hinton, G., Srivastava, N., and Swersky, K. Lecture 6a overview of mini-batch gradient descent. http://www.cs.toronto.edu/ hinton/coursera/lecture6/lec6.pdf.

[58] Hoerl, A. E., and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*, 1 (1970), 55–67.

[59] Hojjati, S. H., Ebrahimzadeh, A., Khazaee, A., and Babajani-Feremi, A. Predicting conversion from mci to ad using resting-state fmri, graph theoretical approach and svm. *Journal of Neuroscience Methods 282* (03 2017).

[60] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 5 (1989), 359–366.

[61] Hubin, A., Storvik, G., and Frommlet, F. Deep bayesian regression models. arXiv:1806.02160.

[62] Initiative, A. D. N. Accessed on: Nov. 3, 2020. [online]. available. *http://adni.loni.usc.edu*.

[63] Jaakkola, T., and Jordan, M. I. A variational approach to bayesian logistic regression problems and their extensions.

[64] Javid, K., Handley, W., Hobson, M. P., and Lasenby, A. Compromise-free bayesian neural networks. arXiv:2004.12211.

[65] Kejzlar, V., Bhattacharya, S., Son, M., and Maiti, T. Black box variational bayes model averaging, 2021.

[66] Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems* (2015), C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., pp. 2575–2583.

[67] Korolev, I. Alzheimer's disease: A clinical and basic science review (msrj: Medical student research journal). *Medical Student Research Journal 4* (09 2014), 24–33.

[68] Korolev, I. O., Symonds, L. L., Bozoki, A. C., and Initiative, A. D. N. Predicting progression from mild cognitive impairment to alzheimer's dementia using clinical, mri, and plasma biomarkers via probabilistic pattern classification. *PLOS ONE 11*, 2 (02 2016), 1–25.

[69] Kuang, J., Zhang, P., Cai, T., Zou, Z., Li, L., Wang, N., and Wu, L. Prediction of transition from mild cognitive impairment to alzheimer's disease based on a logistic regression-artificial neural network- decision tree model. *Geriatr Gerontol Int 21(1)* (2021), 43–47.

[70] Laarhoven, T. V. L2 regularization versus batch and weight normalization. arXiv:1706.05350.

[71] Lampinen, J., and Vehtari, A. Bayesian approach for neural networks–review and case studies. *Neural networks : the official journal of the International Neural Network Society 14*, 3 (2001), 257–274.

[72] Latouche, P., and Robin, S. Variational bayes model averaging for graphon functions and motif frequencies inference in w-graph models. *Statistics and Computing 26*, 6 (2015), 1173–1185.

[73] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[74] Lee, H. Consistency of posterior distributions for neural networks. *Neural Networks 13*, 6 (2000), 629 – 642.

[75] Lee, S., Bachman, A., Yu, D., Lim, J., and Ardekani, B. Predicting progression from mild cognitive impairment to alzheimer's disease using longitudinal callosal atrophy. *Alzheimer's and Dementia: Diagnosis, Assessment and Disease Monitoring 2* (03 2016).

[76] Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. Efficient l1 regularized logistic regression. In *AAAI* (2006).

[77] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks 6*, 6 (1993), 861–867.

[78] Li, F., Tran, L., Thung, K.-H., Ji, S., Shen, D., and Li, J. Robust deep learning for improved classification of ad/mci patients. In *Machine Learning in Medical Imaging* (Cham, 2014), G. Wu, D. Zhang, and L. Zhou, Eds., Springer International Publishing, pp. 240–247.

[79] Li, X., Li, C., Chi, J., and Ouyang, J. Variance reduction in black-box variational inference by adaptive importance sampling. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (2018), International Joint Conferences on Artificial Intelligence Organization, pp. 2404–2410.

[80] Liang, F., Li, Q., and Zhou, L. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association 113*, 523 (2018), 955–972.

[81] Liu, Z., Maiti, T., and Bender, A. A role for prior knowledge in statistical classification of the transition from mci to alzheimer's disease. unpublished report., 2020.

[82] Llano, D. A., Bundela, S., Mudar, R. A., Devanarayan, V., and for the Alzheimer's Disease Neuroimaging Initiative (ADNI). A multivariate predictive modeling approach reveals a novel csf peptide signature for both alzheimer's disease state classification and for predicting future disease progression. *PLOS ONE 12*, 8 (08 2017), 1–18.

[83] Logsdon, B., Hoffman, G., and Mezey, J. A variational bayes algorithm for fast and accurate multiple locus genome-wide association analysis. *BMC bioinformatics 11*, 58 (2010).

[84] Lopes, M., Jacob, L., and Wainwright, M. J. In *Advances in Neural Information Processing Systems* (2011), J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24, Curran Associates, Inc., pp. 1206–1214.

[85] MacKay, D. J. C. A practical bayesian framework for backpropagation networks.

[86] Marzetta, T. L., Tucci, G. H., and Simon, S. H. A random matrix-theoretic approach to handling singular covariance estimates. *IEEE Transactions on Information Theory 57*, 9 (2011), 6256–6271.

[87] Mckinney, W. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference* (01 2010).

[88] McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research 18*, 90 (2017), 1–50.

[89] Menikdiwela, M., Nguyen, C., and Shaw, M. Deep learning on brain cortical thickness data for disease classification. In *2018 Digital Image Computing: Techniques and Applications (DICTA)* (2018), pp. 1–5.

[90] Minhas, S., Khanum, A., Riaz, F., Alvi, A., and Khan, S. A. A nonparametric approach for mild cognitive impairment to ad conversion prediction: Results on longitudinal data. *IEEE Journal of Biomedical and Health Informatics 21*, 5 (2017), 1403–1410.

[91] Misra C, Fan Y, D. C. Baseline and longitudinal patterns of brain atrophy in mci pa- tients, and their use in prediction of short-term conversion to ad: Results from adni. *NeuroImage 44(4)* (2009), 1415–1422.

[92] Mitchell, A. J., and Shiri-Feshki, M. Temporal trends in the long term risk of progression of mild cognitive impairment: a pooled analysis. *Journal of Neurology, Neurosurgery & Psychiatry 79*, 12 (2008), 1386–1391.

[93] Mullachery, V., Khera, A., and Husain, A. Bayesian neural networks. arXiv:1801.07710.

[94] Nagapetyan, T., Duncan, A. B., Hasenclever, L., Vollmer, S. J., Szpruch, L., and Zygalakis, K. The true cost of stochastic gradient langevin dynamics. arXiv:1706.02692.

[95] Neal, R. M. Bayesian training of backpropagation networks by the hybrid monte-carlo method. https://www.cs.toronto.edu/∼radford/ftp/bbp.pdf.

[96] Neal, R. M. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996.

[97] Paisley, J., Blei, D., and Jordan, M. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML'12* (2012), ACM press, p. 1363–1370.

[98] Park, T., and Casella, G. The bayesian lasso. *Journal of the American Statistical Association 103*, 482 (2008), 681–686.

[99] Pati, D., Bhattacharya, A., and Yang, Y. On statistical optimality of variational bayes. In *Proceedings of Machine Learning Research*, A. Storkey and F. Perez-Cruz, Eds., vol. 84. PMLR, 2018, pp. 1579–1588.

[100] Pati, D., Bhattacharya, A., and Yang, Y. On statistical optimality of variational bayes. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (2018), A. Storkey and F. Perez-Cruz, Eds., vol. 84, PMLR, pp. 1579–1588.

[101] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[102] Pereira, T., Lemos, L., Cardoso, S., Silva, D., Pina Rodrigues, A., Santana, I., MendonÃ§a, A., Guerreiro, M., and C . Madeira, S. Predicting progression of mild cognitive impairment to dementia using neuropsychological data: A supervised learning approach using time windows. *BMC Medical Informatics and Decision Making 17* (07 2017).

[103] Petersen, R. C., Roberts, R. O., Knopman, D. S., Boeve, B. F., Geda, Y. E., Ivnik, R. J., Smith, G. E., and Jack, C. R. Mild cognitive impairment: ten years later. *Archives of neurology 66*, 12 (December 2009), 1447–1455.

[104] Pollard, D. Empirical processes: Theory and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics 2* (1990), i–86.

[105] Polson, N. G., and Ročková, V. Posterior concentration for sparse deep learning. In *Advances in Neural Information Processing Systems* (2018), S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc.

[106] Price, R. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory 4*, 2 (1958), 69–72.

[107] Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. arXiv:1401.0118.

[108] Risacher, S., Saykin, A., West, J., Shen, L., Firpi, H., and Mcdonald, B. Baseline mri predictors of conversion from mci to probable ad in the adni cohort. *Current Alzheimer research 6* (08 2009), 347–61.

[109] ROBERT, T. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* (1996), 267–288.

[110] ROSS, S. M. *Simulation*, fifth ed. Academic Press, 2013.

[111] SCHMIDT-HIEBER, J. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics 48*, 4 (2020), 1875 – 1897.

[112] SCHO€LKOPF B, S. A. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* Cambridge, 31. MA: MIT Press, 2002.

[113] SHAFFER, J., PETRELLA, J., SHELDON, F., CHOUDHURY, K., CALHOUN, V., COLEMAN, R., AND DORAISWAMY, P. Predicting cognitive decline in subjects at risk for alzheimer disease by using combined cerebrospinal fluid, mr imaging, and pet biomarkers. *Radiology 266* (12 2012).

[114] SHEN, T., JIANG, J., LI, Y., WU, P., ZUO, C., AND YAN, Z. Decision supporting model for one-year conversion probability from mci to ad using cnn and svm. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2018), pp. 738–741.

[115] SHLENS, J. A tutorial on principal component analysis. arXiv:1404.1100.

[116] SIMON, N., FRIEDMAN, J., HASTIE, T., AND TIBSHIRANI, R. A sparse-group lasso. *Journal of Computational and Graphical Statistics 22*, 2 (2013), 231–245.

[117] SINGH, B., DE, S., ZHANG, Y., GOLDSTEIN, T., AND TAYLOR, G. Layer-specific adaptive learning rates for deep networks, 2015. arXiv.1510.04609.

[118] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 56 (2014), 1929–1958.

[119] SUK, H.-I., AND SHEN, D. Deep learning-based feature representation for ad/mci classification. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013* (Berlin, Heidelberg, 2013), K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, Eds., Springer Berlin Heidelberg, pp. 583–590.

[120] SUN, S., CHEN, C., AND CARIN, L. Learning Structured Weight Uncertainty in Bayesian Neural Networks. A. Singh and J. Zhu, Eds., vol. 54 of *Proceedings of Machine Learning Research*, PMLR, pp. 1283–1292.

[121] SUN, S., ZHANG, G., SHI, J., AND GROSSE, R. B. Functional variational bayesian neural networks. In *7th International Conference on Learning Representations, ICLR 2019.* (2019), OpenReview.net.

[122] SUN, Y., SONG, Q., AND LIANG, F. Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association 0*, ja (2021), 1–42.

[123] Tabatabaei Jafari, H., Shaw, M., and Cherbuin, N. Cerebral atrophy in mild cognitive impairment: A systematic review with meta-analysis. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring 1* (12 2015).

[124] Taghia, J. Lecture notes. part III: Black-box variational inference. http://www.it.uu.se/research/systems_and_control/education/2018/pml/lectures/VILectute NotesPart3.pdf.

[125] Tan, Z., Chen, J., Kang, Q., Zhou, M., Abusorrah, A., and Sedraoui, K. Dynamic embedding projection-gated convolutional neural networks for text classification. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–10.

[126] van der Vaart, A., and Wellner, J. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. Springer, New York., 1996.

[127] Vapnik, V. *The Support Vector Method of Function Estimation*. Springer US, Boston, MA, 1998, pp. 55–85.

[128] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1999.

[129] Vapnik, V. N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[130] Varatharajah, Y., Ramanan, V., Iyer, R., and Vemuri, P. Predicting short-term mci-to-ad pro- gression using imaging, csf, genetic factors, cognitive resilience, and demographics. *Sci Rep 2235* (2019), 9.

[131] Verplancke, T., Looy, S., Benoit, D., Vansteelandt, S., Depuydt, P., De Turck, F., and Decruyenaere, J. Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies. *BMC medical informatics and decision making 8* (01 2009), 56.

[132] Wan, R., Zhong, M., Xiong, H., and Zhu, Z. Neural control variates for variance reduction. arXiv:1806.00159.

[133] Wang, B., Hong, R., Xu, Y., Zhou, F., and P, W. Identifying mild cognitive impairment conversion to alzheimer's disease from medical image information. In *2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)* (2016), pp. 1–2.

[134] Wang, J., Xie, F., Nie, F., and Li, X. Unsupervised adaptive embedding for dimensionality reduction. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–12.

[135] Wang, Y., and Blei, D. M. Frequentist consistency of variational bayes. *Journal of the American Statistical Association 114*, 527 (2019), 1147–1161.

[136] Wei, R., Li, C., Fogelson, N., , and Li, L. Prediction of conversion from mild cognitive impairment to alzheimer's disease using mri and structural network features. *Frontiers in aging neuroscience* (2016).

[137] Welling, M., and Teh, Y. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11* (2011), ACM press, pp. 681–688.

[138] Wong, W. H., and Shen, X. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *Annals of Statistics 23*, 2 (1995), 339–362.

[139] Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., and Gaunt, A. L. Deterministic variational inference for robust bayesian neural networks.

[140] Yang, K., and Maiti, T. On the classification consistency of high-dimensional sparse neural network. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2019), pp. 173–182.

[141] Yang, K., and Maiti, T. Statistical aspects of high-dimensional sparse artificial neural network models. *Machine Learning and Knowledge Extraction 2*, 1 (2020), 1–19.

[142] Yang, Y., Pati, D., and Bhattacharya, A. α-variational inference with statistical guarantees. *Annals of Statistics 48*, 2 (2020), 886–905.

[143] Yao, Y., Rosasco, L., and Caponnetto, A. On early stopping in gradient descent learning. *Constructive Approximation 26* (2007), 289–315.

[144] Ye, J., Farnum, M., Verbeeck, R., Lobanov, V., Raghavan, N., Novak, G., Dibernardo, A., and Narayan, V. Sparse learning and stability selection for predicting mci to ad conversion using baseline adni data. *BMC neurology 12* (06 2012), 46.

[145] Young, J., Modat, M., Manuel, J., Mendelson, A., Cash, D., and Ourselin, S. Accurate multimodal probabilistic prediction of conversion to alzheimer's disease in patients with mild cognitive impairment. *Neuroimage Clin* (05 2013), 735–745.

[146] Yu, Z., Ye, F., Yang, K., Cao, W., Chen, C. L. P., Cheng, L., You, J., and Wong, H.-S. Semisupervised classification with novel graph construction for high-dimensional data. *IEEE Transactions on Neural Networks and Learning Systems* (2020), 1–14.

[147] Zhang, D., and Shen, D. Multi-modal multi-task learning for joint prediction of clinical scores in alzheimer's disease. pp. 60–67.

[148] Zhang, D., Shen, D., and Initiative, A. D. N. Predicting future clinical changes of mci patients using longitudinal and multimodal biomarkers. *PLOS ONE 7*, 3 (03 2012), 1–15.

[149] Zhang, F., and Gao, C. Convergence rates of variational posterior distributions. *Annals of Statistics 48*, 4 (2020), 2180–2207.

[150] Zhu, C., Cheng, Y., Gan, Z., Huang, F., Liu, J., and Goldstein, T. Adaptive learning rates with maximum variation averaging, 2020. arXiv.2006.11918.

[151] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. 1-norm support vector machines. *MIT Press,* (2003), 49–56.

[152] Zou, H., and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*, 2 (2005), 301–320.