ABSTRACT

LIANG, SHUHAN. Flexible Statistical Machine Learning Methods for Optimal Treatment Decision. (Under the direction of Wenbin Lu and Rui Song.)

Personalized medicine aims to tailor treatment based on patients’ information. The goal is to find the optimal treatment regime: if everyone follows the optimal treatment, the best overall potential outcome can be achieved. In this dissertation, we focus on two approaches to estimate the optimal decision rule: The first approach is a learning algorithm based on concordance function with automatic variable selection (chapter 2). The second approach is using deep neural networks to better approximate intricate decision rules (chapter 3).

Machine learning has several successful applications in different research areas such as computer vision, finance, robotics, and operation research. This motivates us to incorporate machine learning techniques into estimating optimal treatment regime: We borrow the idea of support vector machine (SVM), a classical classification technique and use hinge loss as the surrogate loss function. We also incorporate convolutional neural network (CNN) and convexified convolutional neural network (CCNN) in multistage decision making.

In chapter 2, we proposed sparse concordance-assisted learning (SCAL) for optimal treatment regime estimation with automatic variable selection. The advantage of concordance-assisted learning (CAL) is that it makes use of pairwise information. The proposed estimator is more robust and has good statistical properties. We proposed several adaptations including smoothing the original loss function and incorporating variable selection to improve the decision accuracy. Identifying important variables can help reduce the cost in data collection. We improve the computational efficiency of the optimization by implementing
the Douglas-Rachford method. Both simulation studies and real data analysis indicate that SCAL has better performance than existing popular methods.

In chapter 3, we proposed a dynamic advantage learning algorithm under the deep neural network framework. Deep neural networks allow for more flexibility compared to traditional parametric methods. CNN has the advantage of parameter sharing, while CCNN relaxes the constraints of CNN for optimization purpose. We trained CNN and CCNN respectively for contrast function estimation and utilized backward induction for multiple decision making. The target of both neural networks are the inverse probability weighted estimator (IPWE) of contrast function. The results of empirical studies show that CCNN/CNN based A-learning outperforms penalized least square estimator.
Flexible Statistical Machine Learning Methods for Optimal Treatment Decision

by
Shuhan Liang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics
Raleigh, North Carolina
2018

APPROVED BY:

__________________________  _________________________
Eric Chi                  Yichao Wu

__________________________  _________________________
Wenbin Lu                 Rui Song
Co-chair of Advisory Committee
Co-chair of Advisory Committee
DEDICATION

To my family.
BIOGRAPHY

The author was born in Shijiazhuang, Hebei, China. She graduated from No. 1 high school of Shijiazhuang in 2009. In 2013, she earned a Bachelor of Science degree after studying at Zhejiang University. In the same year, she started her graduate study at North Carolina State University. In May 2015 she got her master's degree in statistics. She is working towards her Doctor of Philosophy degree under the valuable guidance from her advisors Dr. Wenbin Lu and Dr. Rui Song. Her expected graduation time is May 2018.
ACKNOWLEDGEMENTS

First of all, I would like to sincerely thank my advisors Dr. Wenbin Lu and Dr. Rui Song for their excellent mentorship. Dr. Lu and Dr. Song provide guidance with care and patience to my study as well as research in the past four years, which, I am grateful for forever. They provide tremendous help during our weekly meetings: They have an amazing sense in discovering promising directions and they always give me helpful suggestions when I was stuck for a long time. I benefit a lot from our group meetings where we read and discuss papers together. It also gives me a wonderful opportunity to practice my presentation skills.

I’d also like to thank Dr. Eric Chi and Dr. Yichao Wu for serving on my committee. Both of them provide constructive suggestions during my oral prelim exam. Dr. Chi provides a lot of help in the statistical computation part. Dr. Yichao Wu and Dr. Lan Wang provide helpful suggestions in improving theoretical results. I’d also like to thank Dr. Ruian Ke and Dr. Nora Traum to serve as the graduate school representative.

I’d like to express my gratitude to department of statistics at North Carolina State University. The department offers courses in different concentrations and it broadens my horizon. I’ve also accumulated rich experience in teaching and I am grateful for all the professors with whom I worked. Our department offers many seminars each semester. This is a great opportunity for me to follow the latest research in many areas. I was given many useful advices on graduate school life in the meet-the-speaker section. I’d like to thank Terry and Chris for the technical support. I’d like to thank Alison and Lanakila for always answering my questions regarding to our department policy patiently.
I am thankful that GlaxoSmithKline offered me the Graduate Industrial Trainee position during my fourth year. I’d like to thank my supervisors Alan and Jeffery for giving me the opportunity to know about the industry, especially the real world problems they are facing. What I’ve learned from them is helpful in my research. I would also like to acknowledge support from National Institute of Mental Health for providing the STAR*D data.

Last but not least, I want to say thank you to my family members. My parents and grandparents always encourage me to try my best to overcome difficulties. I’d like to thank all my friends for being supportive in this journey. It is the love from my friends and families that motivates me to work hard everyday and be the best me.
# TABLE OF CONTENTS

**LIST OF TABLES** ......................................................... viii

**LIST OF FIGURES** ......................................................... ix

**Chapter 1**  
Introduction ........................................................................ 1

**Chapter 2**  
Sparse Concordance-assisted Learning for Optimal Treatment Decision  
2.1 Introduction .................................................................. 4  
2.2 Method .......................................................................... 7  
2.2.1 Notation .................................................................. 7  
2.2.2 Concordance-assisted Learning Overview .................... 8  
2.2.3 Sparse Concordance-assisted Learning ......................... 10  
2.3 Error Bound for Order-2 U Statistics .............................. 14  
2.4 Simulation Studies ........................................................ 18  
2.4.1 Low Dimension ......................................................... 19  
2.4.2 High Dimension ......................................................... 20  
2.5 Application to STAR*D Study ........................................ 23  
2.6 Conclusion .................................................................. 25

**Chapter 3**  
Deep Advantage Learning for Optimal Dynamic Treatment Regime  
3.1 Introduction ................................................................ 33  
3.2 Literature Review ........................................................ 36  
3.2.1 Notation and assumption .......................................... 36  
3.2.2 Q-learning .............................................................. 37  
3.2.3 Advantage learning .................................................. 38  
3.2.4 DTR estimation with variable selection ....................... 38  
3.3 Method ........................................................................ 39  
3.3.1 Inverse probability weighted estimator ....................... 39  
3.3.2 Deep convolutional neural network for advantage learning .................................................. 40  
3.3.3 Deep convexified convolutional neural network for advantage learning 43  
3.3.4 Multi-stage CCNN and CNN ................................. 47  
3.4 Simulation ................................................................... 48  
3.5 Application to STAR*D Study ....................................... 50  
3.5.1 Dataset ................................................................. 50  
3.5.2 Processing ............................................................ 50  
3.5.3 Architecture .......................................................... 52  
3.5.4 Results ................................................................. 53  
3.6 Conclusions and future work ....................................... 53
<table>
<thead>
<tr>
<th>BIBLIOGRAPHY</th>
<th>58</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDICES</td>
<td>63</td>
</tr>
<tr>
<td>Supplementary Materials for Chapter 2</td>
<td>64</td>
</tr>
<tr>
<td>Proof of Lemma 0.1</td>
<td>65</td>
</tr>
<tr>
<td>Proof of Lemma 0.2</td>
<td>66</td>
</tr>
<tr>
<td>Proof of Lemma 1</td>
<td>67</td>
</tr>
<tr>
<td>Proof of Lemma 2</td>
<td>69</td>
</tr>
<tr>
<td>Proof of Lemma 3</td>
<td>72</td>
</tr>
<tr>
<td>Proof of Lemma 4</td>
<td>74</td>
</tr>
<tr>
<td>Proof of Equivalence</td>
<td>78</td>
</tr>
<tr>
<td>IPWE of Contrast Function</td>
<td>79</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2.1 Simulation results of sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL): low-dimensional case .............................................................. 26
Table 2.2 Simulation results of sparse concordance-assisted learning (SCAL): high dimensional case .............................................................. 27
Table 2.3 Simulation results of penalized outcome weighted learning (POWL): high dimensional case .............................................................. 28
Table 2.4 Simulation results for observational studies: sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL): .............................................................. 29
Table 2.5 Estimated values, difference in estimated values and its 95% CI .... 30
Table 2.6 Summary of treatment recommended by SCAL: ET stands for estimated treatment; RT stands for randomized treatment. .............. 31
Table 2.7 Summary of treatment recommended by POWL: ET stands for estimated treatment; RT stands for randomized treatment. .............. 32
Table 3.1 Lists of $STAR^D$ treatment options at level 2 and level 3 .......... 56
Table 3.2 Evaluation results for estimated values on STAR*D dataset .... 57
LIST OF FIGURES

Figure 3.1 Value function given patients received estimated optimal treatments for both stages ........................................ 55
With the development of science and technology, it is common that several treatment options are available for a disease. Patients’ responses vary across different medications. Even for the same medication, different dosage may lead to different clinical response. As a result, the treatment that results in the best outcome varies for each individual. One important question is how to find the optimal treatment for each individual based on patients' information such as family and medical history, diagnosis and clinical results. This so-called "optimal treatment regime" has received a lot of attention from researchers in recent years([Moo07],[Hen10],[Qia12],[Dav16]). Watkins & Dayan [WD92] modeled the conditional expectation of outcome (Q-function) and obtained the optimal treatment regime.
by selecting the action corresponding to the maximum Q-function. Murphy [Mur03] proposed the advantage learning (A-learning) algorithm. A-learning directly models contrast function, which is the difference in potential outcome given different treatments.

For many clinical trial datasets, the number of patients is relatively small compared to the number of covariates. To improve the accuracy for recommending the optimal treatment and reduce the amount of covariates required for making the decision, variable selection need to be performed. Identifying a set of variables that contains sufficient information for treatment selection is crucial since it controls overfitting and saves cost in information collection and decision making process.

Fan et al. [Fan16] proposed concordance-assisted learning. It is based on maximum rank correlation estimator. The treatment regime is determined by two parts: prescriptive index and threshold. Concordance-assisted learning makes better use of the available information through pairwise comparison. However the objective function is discontinuous and computationally hard to optimize. In chapter 2, we consider a continuous surrogate loss function to solve this problem. In addition, our algorithm ensures sparsity and easy interpretation of decision rule. We derive the $L_2$ error bound of the proposed estimator under ultra-high dimension. Simulation results of various settings and application to STAR*D both illustrate that the proposed method can estimate optimal treatment regime well even when the number of covariates is large.

Recently deep learning has achieved state-of-the-art performance on many difficult tasks ([Goo16]). The advantage of a deep neural network is that it allows for more flexibility in decision function. Deep neural networks can also identify important covariates automati-
Parameter sharing of convolutional neural network (CNN) greatly reduces the amount of parameters in the neural network, which allows for high scalability. In chapter 3, we present a deep advantage learning (A-learning) approach to estimate optimal dynamic treatment regime. A-learning models the advantage function, which is of direct relevance to the goal. It does not posit any assumptions on the baseline function. As a result, A-learning is more robust compared to Q-learning. We use the inverse probability weighting (IPW) method to estimate the difference between potential outcomes. We implemented different architectures of CNN and convexified convolutional neural networks (CCNN) ([Zha16b]). The proposed methods are applied to the STAR*D dataset. Together with the simulation results, they demonstrate better performance compared with the penalized least square estimator.
2.1 Introduction

A treatment regime is a decision rule that tailors treatment for each individual. Instead of randomly assigning treatments, we can select a specific treatment among a few options for
each patient based on his or her clinical, genetic and other health information. A decision rule is a procedure to decide which treatment should be picked and it is a function of available information for each patient. Optimal treatment regime aims to find the decision rule that would yield the most favorable outcome. Besides treatment type, treatments of interest also include different treatment combinations and dosage level variation. In reality, it often occurs that large number of patient level covariates are available. However, many of them have no qualitative interaction with treatment effect. Covariates may also be correlated with each other. Under such circumstances, variable selection for optimal treatment regime is necessary to avoid overfitting and increase model interpretation.

Many learning algorithms have been proposed to estimate optimal treatment regime ([Qia12]). Watkins & Dayan [WD92] modeled the conditional expectation of outcome (Q-function) and obtained the optimal treatment regime through maximizing the Q-function. Qian & Murphy [QM11] extended Q-learning using $l_1$-penalized least square (PLS). The estimator derived from the two-step procedure may not be consistent if the conditional mean is misspecified. Instead of modeling the outcome, Murphy [Mur03] proposed the advantage learning (A-learning) algorithm, which is based on modeling contrast function. A contrast function is the difference in potential outcome given different treatments. Lu et al. [Lu11] considered model selection for estimating optimal treatment regime via penalized least square. A-learning is more robust than Q-learning since it does not require a correct specification of the baseline function but a correct model of interaction term is still needed. In literature, it is common to adopt parametric models for Q-function or contract function. As a result, the corresponding decision rule derived from Q-learning or A-learning may be biased.
To further reduce the impact of model misspecification, Zhang et al. [Zha12a] proposed a value function estimator using inverse probability weighting. The optimal decision rule is derived by maximizing the value function estimator. Zhao et al. [Zha12c] proposed the outcome weighted learning (OWL) algorithm. The OWL approximates optimal treatment decision estimation by transforming an objective function in [Zha12a] to a classification loss. Larger reward observed indicates higher chance that the optimal decision rule would recommend the same treatment as the patient actually received. Song et al. [Son15b] extended this method to penalized outcome weighted learning (POWL). Penalty functions include lasso ([Tib96]) and SCAD ([FL01]).

Value search methods suffer from slow convergence and computation difficulties. Fan et al. [Fan16] proposed a novel concordance-assisted learning (CAL) algorithm to estimate optimal decision rule. Concordance function is motivated by maximizing value function using pairwise comparison between patients. Since concordance function can be estimated by a much smoother function, better asymptotic results can be obtained.

In this chapter, we show that concordance-assisted learning algorithm can be transformed to a classification problem. We replace 0-1 loss by a continuous surrogate function. In order to improve the accuracy of optimal treatment regime and interpretation of decision rule, we conduct variable selection by adding lasso penalty to the objective function. We derive error bound of the proposed estimator under ultra-high dimension. We illustrate that the proposed estimator has better performance than existing popular methods under different scenarios together with a clinical trial study.

In section 2, we reviewed and developed the concordance-assisted learning algorithm.
We continued to derive the $L_2$ error bound of coefficient estimation in section 3. Section 4 demonstrates the performance of sparse concordance-assisted learning at different settings. We present results of the proposed method for the STAR*D clinical trial in section 5. The proofs of all lemmas and theorems are provided in the Appendix.

2.2 Method

In this section, we first introduce notations and explain its usage. It is followed by a concordance-assisted learning overview. We then propose the sparse concordance-assisted learning algorithm and provide an algorithm to calculate the proposed estimator using Douglas-Rachford splitting method.

2.2.1 Notation

Let $X_i = (X_{i1}, X_{i2}, ..., X_{ip})^T$ denote the vector of covariates measured for the i-th patient, $A_i$ the assigned treatment and $Y_i$ the outcome after treatment. Let $X = (X_1, X_2, ..., X_n)^T$ denote the feature matrix. Assume that $(X_i, A_i, Y_i)$ are independent, identically distributed. $Y$ is a continuous variable and larger value of $Y$ indicates better treatment effect. Denote $g(X)$ as the individualized treatment regime (ITR), $P^g$ as the joint distribution of $(X, A = g(X), Y)$, $E^g(Y)$ as the expected outcome if all treatments follow $g(X)$. From now on we consider the case of a binary treatment, i.e., $A$ takes values in $\{0, 1\}$. Denote $\mu(a, X) = E(Y|A = a, X)$, [Zha12a] shows that $g^{opt}(X) = I\{\mu(1, X) > \mu(0, X)\}$. Here $g^{opt}(X)$ represents optimal treatment regime.

We also assume stable unit treatment value assumption (SUTVA) and no-unmeasured-confounders assumption holds. SUTVA([Rub80]), i.e., $Y = I(A = 0)Y^*(0) + I(A = 1)Y^*(1)$,
assumes that no interference exists between treatments of different units and no same
treatment variation exists for different units. $Y^*_i(a)$ is the potential outcome after receiving
treatment $a$ for subject $i$. The no-unmeasured-confounders condition, i.e., $\{Y^*_i(0), Y^*_i(1)\} \perp A_i|X_i$, implies all variables that affect treatment assignment or treatment-specific out-
comes are observed. The second assumption holds in a randomized trial.

2.2.2 Concordance-assisted Learning Overview

Concordance-assisted learning estimates optimal treatment regime by comparing the outcome gain of different treatments between individuals. Maximum Rank Correlation (MRC) estimator ([Ken38], [Han87], [CS98]) is chosen to estimate the concordance function. CAL further relaxes parametric assumptions and allows for more flexibility. Fan et al. [Fan16] showed that under certain conditions, optimal treatment regime estimated by concordance-assisted learning is the same as optimal treatment regime estimated by maximizing value functions.

The true optimal decision rule may not be linear, however, throughout the section, we only search the optimal decision rule within the class of linear decision rules, i.e. $g(X) = I(\beta^T X \geq \beta_0)$. This is partly because that linear decision rules are much easier to compute and interpret compared with nonlinear decision rules, and they generally can achieve high accuracy. CAL is a two-step procedure that first estimates the prescriptive index, i.e., a set of decision rules with fixed covariate weights by maximizing the concordance function:

$$C(\beta) = E \left\{ \left[ (Y_i^*(1) - Y_i^*(0)) - (Y_j^*(1) - Y_j^*(0)) \right] I(\beta^T X_i > \beta^T X_j) \right\}$$
and then threshold estimator is optimized based on the prescriptive index estimator. Let \( D(X_i) \) be the expected outcome gain of treatment 1 for the \( i^{th} \) subject, i.e. \( D(X_i) = E(Y_i|A_i = 1, X_i) - E(Y_i|A_i = 0, X_i) \). Concordance function is motivated by the following idea: for pairwise subjects \( i \) and \( j \), larger \( D(X_i) - D(X_j) \), which means subject \( i \) would benefit more by taking treatment 1 compared to subject \( j \), requires larger \( \beta^T X_i \) compared to \( \beta^T X_j \).

Define \( w_i = \frac{Y_i - \nu(X_i)[A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} \), here \( \nu(X) \) is any arbitrary function and \( \pi(X) = P(A_i = 1|X_i) \) is the propensity score. In practice we choose \( \nu(X) \) to be the mean response of the patients who receive treatment 0. Given \( X_i \), \( w_i \) is an unbiased estimator of \( D(X_i) \). The proof is given in Appendix C. Concordance-assisted learning can be summarized as follows:

1. Estimate the prescriptive index:

   \[
   \hat{\beta} = \arg \max_{\|\beta\|=1} \frac{1}{n(n-1)} \sum_{i \neq j} (w_i - w_j) I(\beta^T X_i > \beta^T X_j).
   \]

2. Estimate the threshold using the inverse probability weighted estimator (IPW) proposed by [Zha12a]:

   \[
   \hat{\beta}_0 = \arg \max_{\beta_0} \frac{1}{n} \sum_{i=1}^n \frac{\{Y_i - \nu(X_i)\} I\{A_i = \hat{g}(X_i)\}}{A_i \pi(X_i) + (1-\hat{g})[1-\pi(X_i)]},
   \]

   \[
   \hat{g}(X_i) = I(\hat{\beta}^T X_i > \beta_0).
   \]

Although in general the concordance-based estimator does not always lead to the actual optimal decision rule, under certain conditions, concordance-based estimator is the maximizer of value function ([Fan16]). Concordance-based estimator has attractive properties, including faster convergence rates, known asymptotic distribution (normal) and easy opti-
mization ([Fan16]). It is a very promising approach for optimal treatment regime estimation. In the next section, we will introduce sparse concordance-assisted learning (SCAL). Compared to CAL, it is easier to optimize and can achieve satisfactory accuracy under high dimension.

2.2.3 Sparse Concordance-assisted Learning

Notice that solving for \( \hat{\beta} \) is equivalent to minimize:

\[
\frac{1}{n(n-1)} \sum_{i \neq j} (w_i - w_j) I(\beta^T X_i < \beta^T X_j),
\]

subject to \( ||\beta|| = 1. \) (2.1)

(2.1) is equivalent to minimizing (see Appendix B):

\[
\sum_{w_i > w_j} (w_i - w_j) I(\beta^T X_i < \beta^T X_j),
\]

subject to \( ||\beta|| = 1. \)

This alternative expression reduces computation cost and ensures the convexity of the objective function. We replace the indicator loss function with the hinge loss. Hinge loss is a convex upper bound of the 0-1 loss function. It is often used for support vector machine ([CV95]), a popular classification method with good performance ([Gor04]). The optimization of hinge loss function can be solved in polynomial time. Due to the high dimension of \( p \), we use lasso penalty to estimate the optimal treatment regime and perform variable selection simultaneously. Lasso penalty also helps reduce the variance of the fitted coefficients ([Zhu04]). The prescriptive index estimated by the sparse concordance-assisted
learning algorithm (SCAL) is:

\[ \hat{\beta} = \arg \min_{\beta} \frac{2}{n(n-1)} \sum_{w_i > w_j} (w_i - w_j) \left[ 1 - \beta^T (X_i - X_j) \right]_+ + \lambda \sum_{j=1}^{p} |\beta_j|. \]

We then estimate the threshold parameter \( \beta_0 \) by:

\[ \hat{\beta}_0 = \arg \max_{\beta_0} \frac{1}{n} \sum_{i=1}^{n} w_i I(\hat{\beta}^T X_i > \beta_0). \]

The threshold parameter \( \beta_0 \) is estimated through grid search. In practice, the search range is \( [\min(\hat{\beta}^T X_i), \max(\hat{\beta}^T X_j)], 1 \leq i, j \leq n \). We sort the subjects by descending order of \( w_i \).

Therefore,

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j). \quad (2.2) \]

The objective function of SCAL can be written as:

\[ \hat{\beta} = \arg \min_{\beta} \frac{2}{n(n-1)} \sum_{i=1}^{n} \delta w_i (1 - \beta^T D_i)_+ + \lambda \sum_{j=1}^{p} |\beta_j|. \]

\[
D = \begin{pmatrix}
X_1^T - X_2^T \\
X_1^T - X_3^T \\
\vdots \\
X_1^T - X_n^T \\
X_2^T - X_3^T \\
\vdots \\
X_2^T - X_4^T \\
\vdots
\end{pmatrix}, \quad \delta w = \begin{pmatrix}
w_1 - w_2 \\
w_1 - w_3 \\
\vdots \\
w_1 - w_n \\
w_2 - w_3 \\
\vdots \\
w_2 - w_4 \\
\vdots
\end{pmatrix}.
\]
The optimization problem in step 1 is a weighted $L_1$-SVM problem. The objective function is convex and piecewise linear and many algorithms have been proposed to solve this problem. It can be solved by various linear programming and convex packages. Zhu et al. [Zhu04] proposed an algorithm to compute the whole solution path. Iterative algorithm like Spingarn’s Method is another good way to solve this problem. We use three methods: CVX, a package for specifying and solving convex programs ([MS14], [MS08]), GLPK (GNU Linear Programming Kit, [Glp]) and the method proposed by Spingarn [Spi85] to find the minimizer. Spingarn’s method of partial inverses implements Douglas-Rachford splitting for equality constrained convex problem ([DR56]). We add ancillary variables $\theta$ and reformulate (1) as:

$$
\min f_1(\beta) + f_2(\theta)
$$

subject to $\theta = D\beta$

where $f_1(\beta) = \lambda \|\beta\|_1, f_2(\theta) = \sum_{i=1}^{(l)} \delta w_i (1 - D_i \beta)_+.$

The iterative algorithm is as follows:
Repeat

1. $V_1^+ = (\text{prox}_{t_f^1}(\beta), \text{prox}_{t_f^2}(\theta))$, where
   \[ [\text{prox}_{t_f^1}(\beta)]_i = S(\beta_j, t\lambda), \]
   $S$ is soft thresholding operator: $S(x, \lambda) = \text{sgn}(x)(|x| - \lambda)_+$.
   \[ [\text{prox}_{t_f^2}(\beta)]_i = \begin{cases} 
   1 & \theta \in [1 - t\delta w_i, 1], \\
   \theta_i & \theta > 1, \\
   \theta_i + t\delta w_i & \theta < 1 - t\delta w_i.
   \end{cases} \]

2. $V_2^+ = \begin{pmatrix} I \\ D \end{pmatrix} R^T R^{-1} \left[ P_1(2V_1^+ - V_3) + D^T P_2(2V_1^+ - V_3) \right].$
   $P_1(\beta, \theta) = \beta, P_2(\beta, \theta) = \theta, RR^T = I + D^T D$ is Cholesky decomposition.

3. $V_3^+ = V_3 + V_2^+ - V_1^+.$

until convergence.

Output: $\beta$
We keep step-size parameter $t$ fixed at 1. The convergence is guaranteed ([Spi85]). The iterative algorithm greatly reduces memory and time cost. Under the case of ultra-high dimension, preprocessing requires $O(p^3)$ work to form and compute Cholesky decomposition and $O(p^2)$ work per iteration. In summary, CVX is the least computational efficient way to estimate prescriptive index and Spingarn’s Method is the only approach that can handle STAR*D trial in terms of its scale.

2.3 Error Bound for Order-2 U Statistics

Define $\beta^* = \arg\min_{\beta} L(\beta)$ where

$$L(\beta) = E \{ (w_i - w_j) I(w_i - w_j > 0)[1 - (X_i - X_j)^T \beta]_+ \}.$$

Then the gradient vector and Hessian matrix of the loss function $L(\beta)$ are:

$$S(\beta) = -E \{ (w_i - w_j) I(w_i - w_j > 0)I[1 - (X_i^T - X_j^T)\beta \geq 0](X_i - X_j) \},$$

$$H(\beta) = E \{ (w_i - w_j) I(w_i - w_j > 0)Dirac \delta[1 - (X_i^T - X_j^T)\beta] (X_i - X_j)(X_i^T - X_j^T) \},$$

where $Dirac \delta$ is the Dirac delta function. Denote the index set of active features as $T = \{1 \leq j \leq p : \beta_j^* \neq 0\}$ and $|T| = q$. $\hat{\beta}(\lambda) = \arg\min_{\beta} l_n(\beta, \lambda)$ is an estimator of $\beta^*$, where

$$l_n(\beta, \lambda) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j) I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)\beta]_+ + \lambda \|\beta\|_1.$$

We assume the following regularity conditions:
(A1) The densities of \( X_i, i = 1, 2, \ldots \) are continuous and have common support in \( \mathbb{R}^p \), and there exists a constant \( M_0 > 0 \) such that \( |X_{ij}| \leq M_0 \), \( i \in \mathbb{R}^+, j \in 1, \ldots, p \).

(A2) Denote \( Z_{ij} = X_i^T - X_j^T \) with probability density function \( f^*(z) \). There exists \( B(0, \delta_0) \), a ball centered at 0 with radius \( \delta_0 > 0 \) such that \( E[(w_i - w_j) I(w_i - w_j > 0) | Z_{ij} = z_{ij}] f^*(z_{ij}) > C_3 \) for every \( z_{ij} \in B(0, \delta_0) \).

(A3) \[
\int E[(w_i - w_j) I(w_i - w_j > 0) | z] z_k f^*(z) d z \neq 0 \text{ for some } k.
\]

(A4) There exists a constant \( M_1 \) s.t. \( \max_{d \in \mathbb{R}^p: \|d\|_0 \leq 2q} \frac{d^T D d}{\|d\|_2} \leq M_1 \) almost surely.

(A5) Denote \( \tilde{c} = \frac{c-1}{c+1} \) where \( c \) is a constant satisfying \( \lambda \geq c \|S(\hat{\beta})\|_{\infty} \), \( T \) is the set of significant coefficients (non-zero coefficients). There exists a constant \( M_2 > 0 \) such that

\[
\min_{d \in \mathbb{R}^p: \|d\|_0 \leq q, \|d_T\|_1 \geq \tilde{c} \|d_T\|_1} \frac{d^T H(\beta^*) d}{\|d\|_2^2} \geq M_2.
\]

(A6) \( q = O(n^{c_1}) \) for some \( 0 \leq c_1 < \frac{1}{2} \).

(A7) There exists a constant \( M_3 \) such that for any \( w_M \), \( P(|w_i| > w_M) < e^{(\lambda - \frac{w_M}{M_3})} \).

Condition (A1) ensures \( H(\beta) \) is well-defined and continuous in \( \beta \). Condition (A2) is similar to condition (A2) in [Koo08]. It guarantees \( L(\beta) \to \infty \) as \( \|\beta\| \to \infty \) and further guarantees the existence of \( \beta^* \). Condition (A3) implies that \( \beta^* \neq 0 \). Condition (A4) gives the upper bound of restricted eigenvalue (RE). It can guarantee the Gram matrix is positive definite over a subset of vectors ([Bic09]). Condition (A5) gives the lower bound for restricted eigenvalue of \( H(\beta^*) \). Condition (A6) restricts the divergence rate for the number of none-zero variables. Condition (A7) is a popular distribution assumption in literature.

**Lemma 1:** Assume condition (A1) and (A7) satisfied. Suppose \( \lambda = c \sqrt{32A(\alpha)(\log p)^3/n} \), \( c \) is...
some given constant, \( \alpha \) is a small probability and \( A(\alpha) \) is a constant such that

\[
(2 + n)p^{-A(\alpha)^{1/3}M_0^{-2/3}M_3^{-2/3} + 1} \leq \alpha
\]

we have

\[
P(\lambda \geq c ||\hat{S}(\beta^*)||_{\infty}) \geq 1 - \alpha.
\]

Lemma 2: Assume conditions (A1), (A4), (A6) and (A7) are satisfied, \( p > n \). Let

\[
B(h) = \frac{2}{n(n-1)} \left[ \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ \right. \\
- \left. \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)\beta^*]_+ \right. \\
- \sum_{i=1}^{n} \sum_{j=1}^{n} E[(w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ \right. \\
- \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)\beta^*]_+ \right].
\]

Then for sufficiently large \( n \),

\[
P\left( \sup_{||h||\leq q, ||h||_2 \neq 0} \frac{B(h)}{||h||_2} \geq (1 + C_2 \sqrt{M_1})q \sqrt{\frac{32\log p}{n}} \left[ M_3 q(C_2^2 - 2) \log p + M_3 \log 2n \right] \right) \leq 3p^{-q(C_2^2 - 2)}
\]

Lemma 2 guarantees that with high probability,

\[
\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (w_i - w_j)I(w_i - w_j > 0) \left[ [1 - (X_i^T - X_j^T)(\beta^* - h)]_+ - [1 - (X_i^T - X_j^T)\beta^*]_+ \right].
\]
is within a small range of its expectation. From now on, we choose \( h \) to be \( h = \beta^* - \hat{\beta}(\lambda) \).

**Lemma 3:** For \( \lambda \geq c \|S(\hat{\beta}^*)\|_\infty \),

\[
\|h_T\|_1 \geq \bar{c}\|h_{T^c}\|_1,
\]

where \( \bar{c} = \frac{c-1}{c+1} \), \( T \) is the set of significant coefficients (non-zero coefficients) and \( |T| \leq q \).

**Theorem 4:** Suppose (A1) - (A7) hold, then \( \hat{\beta} \) satisfies

\[
\|\hat{\beta} - \beta^*\|_2 \leq \sqrt{1 + \frac{1}{c} \left[ \frac{2\lambda \sqrt{q}}{M_2} + \frac{2C_4}{M_2} q^2 \sqrt{\frac{\log p}{n}} \left( \frac{5}{4} + \frac{1}{c} \right) \right]}
\]

with probability at least \( 1 - 3 p^{-q(C_2^2 - 2)^k} \), where \( C_4 \) is a constant.

When \( \lambda = c \sqrt{32A(a)(\log p)^3/n} \), the first term has order \( \sqrt{(\log p)^3 q / n} \) and the second term has order \( q^2 \sqrt{(\log p)^3 / n} \). Therefore, with high probability,

\[
\|\hat{\beta}(\lambda) - \beta^*\|_2 = O_p(q^2 \sqrt{(\log p)^3 / n}).
\]

The proofs of the lemmas and theorems are given in the Appendix. We first show the existence of \( \beta^* \) and it is non-trivial (formally stated and proved in Appendix A). Lemma 1 indicates a certain \( \lambda \) to bound the infinity norm of gradient vector with large probability. Lemma 2, and Lemma 3 are stepping stones on the path to proving Theorem 4. To be specific, Lemma 2 establishes the relationship between difference in loss functions and its expectation. Lemma 3. Similar results can be found in [Zha16a] and [Pen15]. Compared
to their proofs, our proof uses Hoeffding’s inequality for u-statistics of order 2. Another difference is that an unbounded weight exists in our objective function. To handle this challenge, we make assumptions on the tail distribution of weights and adjust the bound correspondingly.

### 2.4 Simulation Studies

In this section, we demonstrate the numerical performance of the proposed method. We simulate data from a randomized experiment and evaluate the estimated optimal treatment regimes using sparse concordance-assisted learning and penalized outcome weighted learning. Objective function of POWL is:

\[
\frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{Y_i}{A_i \pi(X_i) + (1 - A_i)(1 - \pi(X_i))} \left[ 1 - (2A_i - 1)g(X_i) \right] \right\} + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{2.3}
\]

Here \( g \) is a linear function, i.e. \( g(\beta, X_i) = \beta^T X_i + \beta_0 \). Notice that the solution to (2) will remain the same if every \( Y_i \) is added to a constant \( c \). In order to guarantee the objective function is convex and the optimization problem is feasible, we add a constant to all \( Y_i \) to make sure the smallest response is positive. The constant is chosen so that the smallest shifted response is 0.01. POWL is implemented using convex toolbox in MATLAB. We compute the IPW estimator using Monte Carlo simulations with 1000 replicates and select the tuning parameter \( \lambda \) with the largest \( \hat{Y}_{opt} \).

To evaluate the estimated decision rule, we report the mean outcome following the esti-
mated optimal treatment regime (Estimated Value) and the percentage of correct decision (PCD) of the estimated optimal treatment regime. The mean of value function following estimated treatment regime is calculated by plugging estimated decisions in the real model using Monte Carlo simulations with 1000 replicates. The mean of value function following the true optimal treatment regime (True Value) is also listed. In addition, we report the mean square error of $\hat{\beta}$. For variable selection, we report correct number of zero coefficients (Corr0) and incorrect number of zero coefficients (Incorr0) compared to the true optimal treatment regime. Results are evaluated and compared under various settings. The associated sample standard deviations are included in the parentheses.

2.4.1 Low Dimension

We follow the first simulation scenario in [Zha12c]: $X_i1, X_i2, \ldots, X_{i50}$ are generated independently from a uniform distribution on $[-1, 1]$, $i = 1, \cdots, n$. The treatment indicator $A$ is generated from Bernoulli distribution with $p = 0.5$. The conditional density of the response $Y$ given $X$ and $A$ is normal, with mean $Q_0(X_i) = 1 + 2X_i1 + X_i2 + 0.5X_i3 + 0.442(1 - X_i1 - X_i2)(2A_i - 1)$ and variance 1. Here only $X_i1$ and $X_i2$ have linear interaction with treatment. We ran 100 simulations with $n=30, 100$ and 200 respectively to estimate the individualized treatment rule using SCAL and POWL. Table 1 summarizes the results.

From Table 1 we have the following observations. First, sparse concordance-assisted learning leads to more accurate estimates of $\beta$ and better variable selection results. Sparse concordance-assisted learning achieves smaller mean square error (MSE) smaller Incorr0 and smaller Corr0 than that of penalized outcome weighted learning. Although the model size of POWL is smaller and closer to the real model size, its value function estimation is
smaller. This further demonstrates that SCAL can select covariates that have strong interaction with treatment and compensate for the influence of model complexity.

The mean of value function following the estimated treatment regime gets closer to the real optimal value as sample size increases. We also notice that SCAL estimator does not vary much from sample to sample: both PCD and value function estimated by SCAL have smaller variance.

In general two methods lead to comparable results. The difference between methods are small, and this is especially true when the sample size is large. When n=30, SCAL leads to much closer value function estimation to true optimal value function estimation than that of POWL. But when n=200, the difference between value functions estimated from SCAL and POWL is only 1.39% of the true value function estimation. It is not surprising since compared to concordance-assisted learning, outcome weighted learning uses information less efficiently, which, can be made up of by increasing available information.

### 2.4.2 High Dimension

We consider the following six models to generate simulation data:

Model I: \( Y = X \gamma_1 + X \beta A + \epsilon, \gamma_1 = (3, -1, 1, 0_{p-2})^T, \beta = (2, 1.8, 0, 0, -1.6, 0_{p-6})^T. \)

Model II: \( Y = 3 - 0.5(X \gamma_1)^2 + 0.625(X \gamma_2)^2 + X \beta A + \epsilon, \gamma_1 = (1, 0.5, 0_{p-2})^T, \gamma_2 = (0, 1, 0_{p-2})^T, \beta = (2, 1.8, 0, 0, -1.6, 0_{p-6})^T. \)

Model III: \( Y = 1 - \sin(X \gamma_1) + \sin(X \gamma_2) + X \beta A + \epsilon, \gamma_1 = (1, 0_{p-1})^T, \gamma_2 = (0, 1, 0_{p-2})^T, \beta = (2, 1.8, 0, 0, -1.6, 0_{p-6})^T. \)
Model IV: 

\[ Y = X\gamma_1 + (X\beta)^3 A + \epsilon, \quad \gamma_1 = (3, -1, 1, 0_{p-2})^T, \beta = (1, 0.9, 0, 0, 0, -0.8, 0_{p-6})^T. \]

Model V: 

\[ Y = 3 - 0.5(X\gamma_1)^2 + 0.625(X\gamma_2)^2 + (X\beta)^3 A + \epsilon, \quad \gamma_1 = (1, 0.5, 0_{p-2})^T, \]
\[ \gamma_2 = (0, 1, 0_{p-2})^T, \beta = (1, 0.9, 0, 0, 0, -0.8, 0_{p-6})^T. \]

Model VI: 

\[ Y = 1 - \sin(X\gamma_1) + \sin(X\gamma_2) + (X\beta)^3 A + \epsilon, \quad \gamma_1 = (1, 0_{p-1})^T, \]
\[ \gamma_2 = (0, 1, 0_{p-2})^T, \beta = (1, 0.9, 0, 0, 0, -0.8, 0_{p-6})^T. \]

There are three baseline functions: Models I and III share the same linear baseline function; Models II and IV share the same higher order polynomial baseline function; Models III and VI share the same complex baseline function. In the first three models there is a linear interaction between covariates and treatment; in the last three models treatment and a cubic function of prescriptive index are interacted with each other. All six models have the same important variables \( X_{i1}, X_{i2} \) and \( X_{i6} \). Covariates \( X_i = (X_{i1}, X_{i2}, \ldots, X_{ip})^T \) are generated from a multivariate normal distribution: each entry is standard normal and the correlation between covariates is \( \text{Corr}(X_{ij}, X_{ik}) = \rho^{|j-k|} \) for \( 1 \leq j \neq k \leq p \). \( \rho \) is chosen to be 0 and 0.2 respectively. The error term \( \epsilon \) is generated from standard normal distribution.

We ran 100 simulations for each scenario with \( n=100 \) and \( p=500, 1000 \) respectively.

We consider randomized studies where \( A \) is generated from Bernoulli distribution with \( p = 0.5 \). The performance of variable selection and treatment regime estimation of both methods are summarized in Tables 2.2 and 2.3. Conclusions are similar to low-dimensional case. SCAL selects more important variables and fewer unimportant variables than POWL. SCAL also provides more accurate decision rule estimation. Its value function estimate is 0.73 higher than POWL, which is 18.3% of true optimal value function estimate and its PCD is 22.4% higher. The comparison between MSE of \( \beta \) estimate further supports the advantage of SCAL.
The performance of SCAL continues to improve as magnitude of interaction between treatment and covariates increases. However we are unable to see this trend from POWL. In general, SCAL can recover important variables better under cubic prescriptive index and treatment interaction than under linear interaction. When $\rho = 0$ and $p = 500$, comparing Model 1 with Model 4, we can see that MSE and Incorr0 dropped a lot. On the contrary, for POWL under the same circumstances, MSE remains almost the same and Incorr0 even increases. Results of Corr0 as well as PCD agree with this statement.

SCAL has demonstrated its performance under high dimension. It also shows the potential to identify important variables when $p$ goes even larger. Performance of variable selection and optimal treatment regime estimation become slightly worse when $p$ increases from 500 to 1000. In reality it is common that hundreds of covariates are available for each patient. Reliable results can still be obtained by SCAL under such circumstances.

The PCD slightly increases when the correlation between covariates increases, suggesting that correlated covariate structure can reduce the impact of falsely selected unimportant variables and missing important variables. When $\rho$ increases from 0 to 0.5, $p = 500$, the PCD of Model 1 increases 4.2% for SCAL and 0.9% for POWL. Due to the fact that correlation exists in most of the real-world data, SCAL proves itself to be a desirable approach.

Next, we consider observational studies where the propensity score is estimated from data. To be specific, the treatment indicator $A$ is generated from Bernoulli($\frac{1}{1+e^{-u}}$), where $u(X) = 0.01 - 0.5 \times X_1 + 0.4 \times X_{10}$. Here, we consider the same high-dimensional settings with $p = 500$ and $\rho = 0.2$, and the propensity score is estimated using the $l_1$-penalized logistic regression. The results of SCAL and POWL are summarized in Table 3.2. Overall,
SCAL outperforms POWL in terms of variable selection and estimating optimal treatment regime in all cases as observed for randomization studies.

2.5 Application to STAR*D Study

We apply the proposed method to STAR*D Study, the largest and longest study ever conducted to assess effectiveness of depression treatments. 4041 outpatients who are diagnosed with major depressive disorder (MDD), representing of various ethnic and socioeconomic groups are collected. There are four levels in this clinical trial and at each level different treatments are evaluated and compared. See [Fav03] for design and measurement details of STAR*D study.

In the data analysis, we focus on patients who received bupropion (BUP) or sertraline (SER) in the second level to illustrate our method. Among the 309 selected subjects, 153 of them received bupropion (BUP) and 166 received sertraline (SER). In order to be consistent with our previous notation, we use 0 to represent SER and 1 to represent BUP. We consider all 305 covariates collected from enrollment, IVR call, ROA interviews, clinic visit and other events (such as suicide, non-serious adverse event and protocol deviation) to recommend individualized treatment for each patient. We choose negative 16-item Quick Inventory of Depressive Symptomatology-Clinician-Rated (QIDS-C16) as our response variable. QIDS-C16 is reverse coded so that it satisfies larger outcome indicates better treatment effect. The negative QIDS-C16 is in the range of -24 to 0.

We apply SCAL using Spingarn's method to this data set. $\lambda_{opt}$ is tuned using 5-fold cross validation. A pre-defined range $(0, 4)$ is searched and $\lambda$ is chosen based on the IPW estima-
Propensity score is estimated by proportion of subjects who receive treatment 1 in the training data set. For comparison, we also evaluate the performance of POWL using 5-fold cross validation. The response is shifted with the smallest value to be 0.01. Note that the adjusted response is only used to optimize objective function; Value function is estimated using original response.

To compare the estimated treatment regimes on STAR*D data, we draw bootstrap samples over 1,000 times and estimate the 95% confident interval of difference between expected outcome following estimated treatment regime from SCAL and the non-dynamic treatment regimes. The expected outcome difference between SCAL and POWL is also calculated. See Table 2.5 for estimate of value function based on estimated optimal treatment regime and 95% confident interval of the differences.

The estimated value function by SCAL is significantly larger than either of the non-dynamic treatment regimes. It is also significantly larger than the expected outcome following the optimal treatment regime based on POWL at $\alpha = 0.05$. Compared to POWL, SCAL achieves a reasonable model size. It keeps good balance of including important features and controlling the complexity of the model. Estimate based on POWL is too sparse and many covariates which have interaction with treatment effects are missed. Table 2.6 and Table 2.7 are summaries of treatment actually received versus estimated optimal treatment. We can see that estimated optimal treatment regime based on SCAL tends to be more balance. Based on SCAL, among all 171 subjects who were assigned to SER, 92 of them should stay in the same treatment and 79 of them should be assigned to BUP. While the result of POWL indicates 111 of them were assigned to the optimal treatment and only 56 patients should switch to BUP.
2.6 Conclusion

We propose a variable selection method based on concordance-assisted learning for estimating optimal treatment regime. Our method can minimize the weighted misclassification rate and select prescriptive index simultaneously. The proposed method gives much more accurate decision rule and value function estimation than existing popular methods under various simulation settings. Moreover, inputs that are correlated with treatments effects are also successfully identified. Sparse concordance-assisted learning achieves promising result in constructing real-world decision. We also study the error bound of SCAL in ultra-high dimension.

The proposed method does not require model specification except for propensity score. It is based on an estimate of contrast function which can be defined easily under binary treatment circumstance. SCAL can solve problems in other fields as well. One popular example is to determine the best move in a game. SCAL can be implemented to choose the best action. In the future, one interesting direction of our study would be to extend the definition of contrast function when more than two treatment arms are available. We may also replace linear support vector machine by other kernels to see whether a better treatment regime can be found.
Table 2.1 Simulation results of sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL): low-dimensional case

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>MSE</th>
<th>Incorr0(0)</th>
<th>Corr0(48)</th>
<th>PCD</th>
<th>Estimated Value</th>
<th>True Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POWL</strong></td>
<td>30</td>
<td>1.60</td>
<td>1.70</td>
<td>42.23</td>
<td>0.615(0.02)</td>
<td>1.09(0.02)</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.27</td>
<td>1.94</td>
<td>46.64</td>
<td>0.768(0.02)</td>
<td>1.27(0.02)</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1.09</td>
<td>1.99</td>
<td>47.78</td>
<td>0.786(0.02)</td>
<td>1.30(0.03)</td>
<td>1.44</td>
</tr>
<tr>
<td><strong>SCAL</strong></td>
<td>30</td>
<td>1.40</td>
<td>0.73</td>
<td>35.79</td>
<td>0.659(0.01)</td>
<td>1.16(0.01)</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.52</td>
<td>0.11</td>
<td>41.97</td>
<td>0.764(0.01)</td>
<td>1.31(0.01)</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.19</td>
<td>0.01</td>
<td>46.03</td>
<td>0.749(0.01)</td>
<td>1.32(0.01)</td>
<td>1.44</td>
</tr>
</tbody>
</table>
Table 2.2 Simulation results of sparse concordance-assisted learning (SCAL): high dimensional case

<table>
<thead>
<tr>
<th>p</th>
<th>ρ</th>
<th>Model</th>
<th>MSE</th>
<th>Incorr0(0)</th>
<th>Corr0(497/997)</th>
<th>PCD</th>
<th>Estimated Value</th>
<th>True Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0</td>
<td>Model 1</td>
<td>0.61</td>
<td>0.75</td>
<td>482.62</td>
<td>0.744(0.01)</td>
<td>3.80(0.02)</td>
<td>4.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>0.56</td>
<td>0.57</td>
<td>485.34</td>
<td>0.763(0.01)</td>
<td>3.79(0.03)</td>
<td>4.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>0.44</td>
<td>0.49</td>
<td>488.12</td>
<td>0.786(0.01)</td>
<td>1.92(0.02)</td>
<td>2.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>0.35</td>
<td>0.35</td>
<td>486.81</td>
<td>0.801(0.01)</td>
<td>5.67(0.04)</td>
<td>5.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>0.32</td>
<td>0.25</td>
<td>487.00</td>
<td>0.810(0.01)</td>
<td>5.63(0.05)</td>
<td>5.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>0.29</td>
<td>0.20</td>
<td>485.33</td>
<td>0.820(0.01)</td>
<td>3.74(0.04)</td>
<td>3.94</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Model 1</td>
<td>0.50</td>
<td>0.71</td>
<td>487.37</td>
<td>0.783(0.01)</td>
<td>4.00(0.02)</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>0.47</td>
<td>0.68</td>
<td>488.52</td>
<td>0.788(0.01)</td>
<td>3.86(0.02)</td>
<td>4.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>0.38</td>
<td>0.46</td>
<td>490.32</td>
<td>0.816(0.01)</td>
<td>2.08(0.02)</td>
<td>2.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>0.28</td>
<td>0.25</td>
<td>487.45</td>
<td>0.831(0.01)</td>
<td>6.53(0.02)</td>
<td>6.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>0.28</td>
<td>0.23</td>
<td>487.86</td>
<td>0.832(0.06)</td>
<td>6.36(0.05)</td>
<td>6.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>0.25</td>
<td>0.18</td>
<td>485.98</td>
<td>0.845(0.01)</td>
<td>4.56(0.02)</td>
<td>4.68</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>Model 1</td>
<td>0.67</td>
<td>0.89</td>
<td>981.92</td>
<td>0.741(0.01)</td>
<td>3.81(0.02)</td>
<td>4.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>0.56</td>
<td>0.70</td>
<td>985.66</td>
<td>0.755(0.01)</td>
<td>3.84(0.03)</td>
<td>4.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>0.49</td>
<td>0.58</td>
<td>988.22</td>
<td>0.782(0.01)</td>
<td>1.93(0.01)</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>0.38</td>
<td>0.34</td>
<td>986.01</td>
<td>0.797(0.01)</td>
<td>5.70(0.03)</td>
<td>5.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>0.34</td>
<td>0.20</td>
<td>985.83</td>
<td>0.800(0.01)</td>
<td>5.68(0.04)</td>
<td>5.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>0.32</td>
<td>0.20</td>
<td>985.25</td>
<td>0.813(0.01)</td>
<td>3.76(0.03)</td>
<td>3.98</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Model 1</td>
<td>0.56</td>
<td>0.94</td>
<td>989.91</td>
<td>0.762(0.01)</td>
<td>4.01(0.02)</td>
<td>4.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>0.51</td>
<td>0.64</td>
<td>986.95</td>
<td>0.779(0.01)</td>
<td>3.88(0.02)</td>
<td>4.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>0.43</td>
<td>0.54</td>
<td>989.14</td>
<td>0.805(0.01)</td>
<td>2.09(0.02)</td>
<td>2.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>0.31</td>
<td>0.27</td>
<td>987.40</td>
<td>0.819(0.01)</td>
<td>6.55(0.02)</td>
<td>6.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>0.30</td>
<td>0.24</td>
<td>985.37</td>
<td>0.827(0.01)</td>
<td>6.41(0.02)</td>
<td>6.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>0.26</td>
<td>0.18</td>
<td>987.92</td>
<td>0.832(0.01)</td>
<td>4.59(0.02)</td>
<td>4.73</td>
</tr>
</tbody>
</table>
Table 2.3 Simulation results of penalized outcome weighted learning (POWL): high dimensional case

<table>
<thead>
<tr>
<th>p</th>
<th>ρ</th>
<th>Model</th>
<th>MSE</th>
<th>Incorr0(0)</th>
<th>Corr0(497/997)</th>
<th>PCD</th>
<th>Estimated Value</th>
<th>True Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0</td>
<td>Model 1</td>
<td>1.81</td>
<td>2.37</td>
<td>449.06</td>
<td>0.521(0.01)</td>
<td>3.03(0.02)</td>
<td>4.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>1.76</td>
<td>2.40</td>
<td>456.11</td>
<td>0.520(0.01)</td>
<td>2.98(0.01)</td>
<td>4.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>1.76</td>
<td>2.33</td>
<td>450.64</td>
<td>0.521(0.01)</td>
<td>1.04(0.01)</td>
<td>2.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>1.81</td>
<td>2.53</td>
<td>450.63</td>
<td>0.516(0.01)</td>
<td>3.00(0.05)</td>
<td>5.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>1.77</td>
<td>2.53</td>
<td>453.97</td>
<td>0.517(0.01)</td>
<td>2.95(0.05)</td>
<td>5.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>1.78</td>
<td>2.53</td>
<td>454.53</td>
<td>0.517(0.01)</td>
<td>1.02(0.06)</td>
<td>3.94</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Model 1</td>
<td>1.76</td>
<td>2.30</td>
<td>456.63</td>
<td>0.528(0.01)</td>
<td>3.08(0.02)</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>1.76</td>
<td>2.38</td>
<td>458.01</td>
<td>0.528(0.07)</td>
<td>2.92(0.02)</td>
<td>4.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>1.75</td>
<td>2.24</td>
<td>458.19</td>
<td>0.530(0.01)</td>
<td>1.09(0.02)</td>
<td>2.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>1.80</td>
<td>2.48</td>
<td>453.13</td>
<td>0.520(0.01)</td>
<td>3.13(0.07)</td>
<td>6.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>1.81</td>
<td>2.53</td>
<td>456.69</td>
<td>0.519(0.01)</td>
<td>2.98(0.07)</td>
<td>6.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>1.79</td>
<td>2.52</td>
<td>453.29</td>
<td>0.521(0.01)</td>
<td>1.15(0.08)</td>
<td>4.68</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>Model 1</td>
<td>1.79</td>
<td>2.67</td>
<td>948.08</td>
<td>0.512(0.01)</td>
<td>3.04(0.01)</td>
<td>4.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>1.82</td>
<td>2.68</td>
<td>947.05</td>
<td>0.512(0.01)</td>
<td>2.99(0.01)</td>
<td>4.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>1.84</td>
<td>2.66</td>
<td>949.14</td>
<td>0.514(0.01)</td>
<td>1.06(0.01)</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>1.83</td>
<td>2.78</td>
<td>950.22</td>
<td>0.508(0.01)</td>
<td>2.88(0.04)</td>
<td>5.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>1.83</td>
<td>2.78</td>
<td>947.72</td>
<td>0.508(0.01)</td>
<td>2.86(0.04)</td>
<td>5.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>1.82</td>
<td>2.78</td>
<td>947.83</td>
<td>0.507(0.01)</td>
<td>0.91(0.04)</td>
<td>3.98</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>Model 1</td>
<td>1.83</td>
<td>2.68</td>
<td>950.87</td>
<td>0.517(0.01)</td>
<td>3.06(0.02)</td>
<td>4.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>1.82</td>
<td>2.66</td>
<td>946.94</td>
<td>0.516(0.01)</td>
<td>2.91(0.01)</td>
<td>4.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>1.78</td>
<td>2.66</td>
<td>951.12</td>
<td>0.517(0.01)</td>
<td>1.08(0.02)</td>
<td>2.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>1.79</td>
<td>2.77</td>
<td>948.49</td>
<td>0.508(0.01)</td>
<td>2.93(0.05)</td>
<td>6.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>1.81</td>
<td>2.76</td>
<td>947.55</td>
<td>0.507(0.01)</td>
<td>2.76(0.05)</td>
<td>6.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>1.76</td>
<td>2.77</td>
<td>952.16</td>
<td>0.510(0.01)</td>
<td>0.96(0.05)</td>
<td>4.73</td>
</tr>
</tbody>
</table>
Table 2.4 Simulation results for observational studies: sparse concordance-assisted learning (SCAL) and penalized outcome weighted learning (POWL):

<table>
<thead>
<tr>
<th>p=500</th>
<th>ρ</th>
<th>Model</th>
<th>MSE</th>
<th>Incorr0(0)</th>
<th>Corr0(497)</th>
<th>PCD</th>
<th>Estimated Value</th>
<th>True Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCAL</td>
<td>0.2</td>
<td>Model 1</td>
<td>0.82</td>
<td>0.85</td>
<td>484.96</td>
<td>0.732(0.01)</td>
<td>3.83(0.04)</td>
<td>4.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>0.73</td>
<td>0.77</td>
<td>487.52</td>
<td>0.744(0.01)</td>
<td>3.72(0.04)</td>
<td>4.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>0.67</td>
<td>0.70</td>
<td>489.03</td>
<td>0.768(0.01)</td>
<td>1.93(0.04)</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>0.41</td>
<td>0.47</td>
<td>483.77</td>
<td>0.807(0.01)</td>
<td>6.42(0.04)</td>
<td>6.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>0.32</td>
<td>0.31</td>
<td>483.05</td>
<td>0.815(0.01)</td>
<td>6.36(0.04)</td>
<td>6.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>0.36</td>
<td>0.41</td>
<td>486.40</td>
<td>0.818(0.01)</td>
<td>4.46(0.05)</td>
<td>4.71</td>
</tr>
<tr>
<td>POWL</td>
<td>0.2</td>
<td>Model 1</td>
<td>1.64</td>
<td>2.49</td>
<td>460.97</td>
<td>0.505(0.01)</td>
<td>2.99(0.01)</td>
<td>4.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 2</td>
<td>1.56</td>
<td>2.70</td>
<td>467.86</td>
<td>0.509(0.01)</td>
<td>2.87(0.01)</td>
<td>4.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 3</td>
<td>1.62</td>
<td>2.56</td>
<td>467.20</td>
<td>0.512(0.01)</td>
<td>1.04(0.01)</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 4</td>
<td>1.52</td>
<td>2.66</td>
<td>470.31</td>
<td>0.500(0.01)</td>
<td>2.92(0.04)</td>
<td>6.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 5</td>
<td>1.51</td>
<td>2.70</td>
<td>472.09</td>
<td>0.501(0.01)</td>
<td>2.79(0.04)</td>
<td>6.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Model 6</td>
<td>1.48</td>
<td>2.66</td>
<td>471.58</td>
<td>0.500(0.01)</td>
<td>0.93(0.05)</td>
<td>4.71</td>
</tr>
</tbody>
</table>
Table 2.5 Estimated values, difference in estimated values and its 95% CI

<table>
<thead>
<tr>
<th>Treatment Regime</th>
<th>Estimated Value</th>
<th>Diff</th>
<th>95% CI on Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Regime (SCAL)</td>
<td>-6.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal Regime (POWL)</td>
<td>-9.46</td>
<td>2.69</td>
<td>(1.18, 4.24)</td>
</tr>
<tr>
<td>BUP</td>
<td>-10.52</td>
<td>3.75</td>
<td>(2.38, 5.19)</td>
</tr>
<tr>
<td>SER</td>
<td>-10.74</td>
<td>3.97</td>
<td>(2.57, 5.50)</td>
</tr>
</tbody>
</table>
Table 2.6 Summary of treatment recommended by SCAL: ET stands for estimated treatment; RT stands for randomized treatment.

<table>
<thead>
<tr>
<th></th>
<th>ET: SER</th>
<th>ET: BUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT: SER</td>
<td>92</td>
<td>75</td>
</tr>
<tr>
<td>RT: BUP</td>
<td>79</td>
<td>73</td>
</tr>
</tbody>
</table>
Table 2.7 Summary of treatment recommended by POWL: ET stands for estimated treatment; RT stands for randomized treatment.

<table>
<thead>
<tr>
<th></th>
<th>ET: SER</th>
<th>ET: BUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT: SER</td>
<td>111</td>
<td>56</td>
</tr>
<tr>
<td>RT: BUP</td>
<td>106</td>
<td>46</td>
</tr>
</tbody>
</table>
3.1 Introduction

Optimal treatment regime aims to tailor medical treatment by taking into account patients’ heterogeneity. Compared to the traditional “one-size-fits-all” approach, optimal
treatment regime can individualize treatment and get optimal output for each patient. Different treatments include differences in treatment type and dosage level variation. For some diseases, treatment adjustment is required through the entire treatment process and multiple treatment selections are needed. Dynamic treatment regime (DTR) aims to select a sequence of treatments at multiple time points for each patient based on patient’s characteristics. By following these rules, the best (maximal) response over the entire population can be achieved. One difficulty in DTR estimation is for each patient at each decision point, we only observe the response of one treatment option. The potential outcomes of other treatments are missing. Many approaches have been proposed to solve this problem such as Q-learning ([Wat89], [WD92]) and A-learning ([Mur03]). In this chapter, we propose a new method to estimate optimal dynamic treatment regime using deep A-learning.

Deep learning has been widely used in many fields such as game playing ([Mni13]), finance ([Din15]), robotics ([Len15]), control and operations research ([Mni15]), and language processing ([CW08]). There are many successful examples of implementing DNN to solve challenging problems. Google Deepmind’s AlphaGo, a program using deep neural networks to play the Go game, won 99.8% of the games against other computer programs and won all 5 games against the European champion ([Sil16]). NVIDIA have implemented deep learning technique to achieve self-driving car ([Boj16]). Using CNN and end-to-end learning, goals like simultaneous localization and mapping and movement planning can be achieved. Besides implementations, theoretical results have been obtained. Pinkus [Pin99] and Hornik et al. [Hor89] discussed theoretical results of multilayer feedforward perceptron (MLP) approximation. Choromanska et al. [Cho15] have shown that as long as the size of neural network is large enough, the performances of any local minimum and global minimum are very similar on testing datasets.
Convolutional neural networks (CNN) has its own advantages due to parameter sharing and local connectivity. The parameters of each filter are shared across patches. It greatly reduces the number of parameters. LeCun et al. [LeC98], first successfully implemented CNN in handwritten digit recognition and lowercase words recognition. Since then many work have been done on CNN. In ImageNet LSVRC-2012 contest, Krizhevsky et al. [Kri12] proposed a deep CNN with dropout and GPU implementation. It successfully classifies more than 1,000,000 images into more than 1,000 categories ([LeC15]). Its top-5 test error rate is 11% lower than second-place solution. Zhang et al. [Zha16b] used low rank matrix to represent the filter of CNN in the reproducing kernel Hilbert space (RKHS). They further proposed CCNN by relaxing the rank constraint to nuclear norm constraint. The authors proved that under binary classification case, its generalization error has oracle inequality. The author also compared several variants of CCNN on image classification.

In this chapter we combine deep convolutional neural network with A-learning. The value functions estimated by CCNN and CNN are compared with an A-learning based penalized least square estimator. The rest of the chapter is organized as follows: In Section 2 we briefly discuss popular DTR techniques in literature. We formulate the A-learning based CCNN and CNN in Section 3. The detailed CCNN and CNN algorithms are included. Deep A-learning is extended to DTR estimation using backward induction. We apply the proposed methods to a data from the STAR*D clinical trial in Section 4. Network architecture and model training details are discussed. Section 5 gives conclusion and lists of avenues for future research.
3.2 Literature Review

3.2.1 Notation and assumption

Denote the predictor vector available at the k-th time point by $X_k$, the treatment at the j-th time point as $A_k$, $k = 1, 2, ..., K$, the final observed outcome as $Y$. The bar notation represents a sequence of past information, e.g., $\bar{A}_k = \{A_1, A_2, ..., A_k\}$. Let $d_k$ denote treatment regime at k-th time point and the asterisk notation represents optimal decision rules. There are several common assumptions needed in estimating optimal dynamic treatment regime, for example, see [Bas80], [Rob97] and [Sch14]. To be specific, we need

1. No unmeasured confounders assumption:

\[ A_k \perp Y_k^*(a)|\{X_1, A_1, X_2, A_2, ..., X_k\}, a \in \psi_j(X_k, \bar{A}_{k-1}). \]

Here $Y_k^*(a)$ denotes the potential outcome given treatment a is received. $\psi_k(\bar{X}_k, \bar{A}_{k-1})$ is all possible treatments given medical and treatment history. This assumes that $X_k$ contains sufficient information thus all predictors that interact with treatment have been observed. No unmeasured confounders assumption holds for sequentially randomized experiments.

2. Positivity assumption: the treatment sequences following dynamic treatment regime can occur. Positivity assumption can be summarized as:

\[ P[\prod_{k=1}^{K} P\{A_k \in \psi_k(\bar{X}_k, \bar{A}_{k-1})|\bar{X}_k, \bar{A}_{k-1}\} > 0] = 1. \]
3. Stable unit treatment assumption (SUTVA): It assumes that the outcome of a patient is only influenced by the treatment(s) he or she receives. There is no interference between subjects. It also assumes that for each treatment there is one unique version. SUTVA can be summarized as:

\[ Y_k = \sum_{a \in \psi_k(X_k, A_{k-1})} Y_k^*(a) I(A_k = a). \]

When the above assumptions hold, the optimal DTR can be estimated based on observed data. Next we will introduce two popular DTR estimation techniques.

### 3.2.2 Q-learning

Watkins [Wat89] proposed Q-learning. It uses incremental dynamic programming to learn optimal action. Q-function reflects the expected outcome if at the k-th time point treatment \( a_k \) is received and at any later time points the optimal treatments are received. Value function represents the expected outcome if at the k-th and any later time points the optimal treatments are received. It can be estimated by solving estimating equations. The optimal decision rule \( d^* \) can be estimated as follows:

\[
d^*_k(X_k, A_{k-1}) = \arg \max_{A_k : p_k(A_k | X_k, A_{k-1}) > 0} Q_k(X_k, A_k), \text{ for } k = K, K-1, ..., 1.
\]

Multi-stage treatment regime estimation is based on backward induction proposed by Cowell et al. [Cow06]. One drawback of Q-learning is that it not consistent if Q-function is misspecified. In the next section, we will review A-learning, which is less sensitive to model misspecification.
3.2.3 Advantage learning

Murphy [Mur03] proposed Advantage learning. A-learning explicitly model the contrast function/regret function. Regret function is the difference in potential outcome between actually received and optimal treatment. Under the A-learning framework, optimal decision rule can be derived directly. From now on, we only consider the case where binary treatment choices are available. The two treatments are denoted as 0 and 1 respectively. Contract function $C$, and optimal treatment $d^*$ are defined as follows:

$$C_k(\bar{X}_k, \bar{A}_{k-1}) = Q_k(\bar{X}_k, \bar{A}_{k-1}, A_k = 1) - Q_k(\bar{X}_k, \bar{A}_{k-1}, A_k = 0),$$
$$d^*_k(\bar{X}_k, \bar{A}_{k-1}) = I\{C_k(\bar{X}_k, \bar{A}_{k-1}) > 0\}.$$

The contrast function can be estimated using g-estimation proposed by Moodie et al. [Moo07].

A-learning has the double-robustness property which makes it suffer less from model misspecification. A-learning makes it possible to build a complex model for baseline function and an easy-to-interpret model of contrast function. It reduces in the influence of model misspecification and generates a relative simple decision rule.

3.2.4 DTR estimation with variable selection

One way to improve decision accuracy of dynamic treatment regime is via variable selection. Qian & Murphy [QM11] extended Q-learning using $l_1$-penalized least square (PLS). However it still suffers problems that rises from Q-learning. The estimator derived from the two-step procedure may not be consistent if the conditional mean model is misspecified. Lu et al. [Lu11] considered model selection for estimating optimal treatment regime via
penalized least square. Shi et al. [Shi16] extended Lu’s method to cases where the propensity score is unknown. They studied the theoretical properties of the proposed estimator given the number of covariates is of the non-polynomial (NP) order of the sample size. Shi et al. [Shi17] studied penalizing A-learning estimation equations for dynamic treatment regime. Besides Q- and A- learning framework, Zhao et al. [Zha12c] proposed outcome weighted learning (OWL). The optimal decision rule is derived by maximizing the value function estimator. Song et al. [Son15a] proposed penalized outcome weighted learning (POWL) which adds a variable selection module to OWL. Penalty functions include lasso ([Tib96]) and SCAD ([FL01]).

3.3 Method

3.3.1 Inverse probability weighted estimator

We start with one stage optimal treatment regime estimation and extend it to multi-stage in later section. Zhang et al. [Zha12b] proposed the inverse probability weighted estimator (IPWE). The IPWE estimator of $E Y(d^*)$ is:

$$C_{\eta,i} = A_i d^*(X_i, \eta) + (1 - A_i)[1 - d^*(X_i, \eta)]$$

$$IPWE(\eta) = \frac{1}{n} \sum_{i=1}^{n} \frac{C_{\eta,i} Y_i}{\pi(X_i)^{A_i}(1 - \pi(X_i))^{1-A_i}}.$$

$\eta$ is the parameter in decision function $d^*$. $\pi(X_i)$ is the known propensity score of patient $i$ receives treatment 1. Optimal treatment regime is estimated by maximizing IPWE, which is
equivalent to estimating the contrast function:

\[ \hat{C}_{IPWE}(X_i) = \frac{A_i}{\pi(X_i)} Y_i - \frac{1 - A_i}{1 - \pi(X_i)} Y_i. \]

We now show that given \( X_i \), \( \hat{C}_{IPWE}(X_i) \) is an unbiased estimator of contrast function. Specifically,

\[
E\left\{ \frac{A_i}{\pi(X_i)} Y_i - \frac{1 - A_i}{1 - \pi(X_i)} Y_i \mid X_i \right\} = E\left[ Y_i \mid A_i = 1, X_i \right] - E\left[ Y_i \mid A_i = 0, X_i \right].
\]

\( \hat{C}_{IPWE} \) is the adjusted observed outcome base on propensity score. It does not posit any parametric assumptions on contrast function. A-learning requires model specification of baseline function. The IPWE does not make any assumptions on those nuisance parameters either. Therefore it suffers less from model misspecification issues. Next we propose a class of algorithms which integrate IPWE with convolutional neural network.

### 3.3.2 Deep convolutional neural network for advantage learning

#### 3.3.2.1 Convolutional neural network

LeCun et al. [LeC98] proposed convolutional neural network. A CNN usually consists of convolutional layers, pooling layers and fully-connected layers:

1. **convolutional layer**: The convolutional layer takes in multi-dimensional features. In the convolutional layer, weighted summation over each region is calculated and a
non-linear transformation (activation function) is operated on top of the summation. Weight matrices (filters) are shared across regions so that the number of parameters is reduced. As a result, CNN is easier to train compared to fully-connected neural network with similar number of neurons. Common activation functions include Rectified Linear Units (ReLUs): \( f(x) = \max(x, 0) \), polynomial functions and hyperbolic tangent function. Krizhevsky et al. [Kri12] showed that ReLU can reduce the training time than other common activation functions.

2. pooling layer: Pooling layer is usually placed between convolutional layers. Local pooling can summarize information within a neighborhood. It reduces the dimension of feature space and avoid overfitting. Two common pooling operations are maximum pooling and average pooling. The pooling operation is operated on non-overlapping or small-proportion-overlapping regions, which, controls the correlation between hidden neurons. Smaller pooling size can keep enough information and it is observed that for maximum pooling, the best pooling stride are 2 and the best patch size are 2 or 3 ([Kar17]).

3. fully-connected layer: The fully-connected layer maps reshaped previous output to the final output of neural network. It guarantees that all neurons are connected to the output. Convolutional layer and fully-connected layer are mutually convertible. For any convolutional layer, the corresponding fully-connected layer has sparse weight matrices. Their non-zero blocks share similar patterns.

Recent research has shown that multilayer stack architecture enhances both the abilities of capturing discriminating and ignoring irrelevant aspects. Compared to its shallow counterpart, deep neural network is more capable of automatic feature representation and capturing complicated relationships. Parameters are estimated by minimizing the empiri-
cal risk. Here we assume the loss function is convex and L-Lipschitz in the output given any value of the target. Backpropagation is used for parameter update. It uses relationship of gradient between parameters of consecutive layers to train a neural network.

### 3.3.2.2 A-learning based Convolutional neural network

In this section, we proposed a new approach, which integrates convolutional neural network with IPWE of contrast function. The input is all available information of each patient and the output is an estimate of IPWE of contrast function \( \frac{Y_i[A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} \). We use least square loss to measure the prediction performance of CNN. The details of CNN is covered in Algorithm 1.

#### Algorithm 1: m-layer advantage learning based CNN

<table>
<thead>
<tr>
<th>Input : ((X_i, A_i, y_i))_{i=1}^{n}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: predictor ({H_m(X_i)}<em>{i=1}^{n}) and optimal treatment regime ({I[H_m(X_i) &gt; 0]}</em>{i=1}^{n})</td>
</tr>
<tr>
<td>1 (1) (H_0(X_i) = X_i, i = 1, ..., n);</td>
</tr>
<tr>
<td>2 (2) for (j \leftarrow 1) to (m) do</td>
</tr>
<tr>
<td>3 if (j &lt; m) then</td>
</tr>
<tr>
<td>4 Apply the convolution filter (f_j^{conv}) and the pooling filter (f_j^{pool}) on (H_{j-1}(X_i), i = 1, ..., n) to get (H_j(X_i), i = 1, ..., n):</td>
</tr>
<tr>
<td>5 end</td>
</tr>
<tr>
<td>6 else</td>
</tr>
<tr>
<td>7 Apply the fully connected layer (f_m) on (H_{m-1}(X_i), i = 1, ..., n) to get (H_m(X_i));</td>
</tr>
<tr>
<td>8 end</td>
</tr>
<tr>
<td>9 end</td>
</tr>
<tr>
<td>10 (3) Use backpropagation to estimate all parameters in ((f_j^{conv}, f_j^{pool}), j=1, 2, ..., m-1) and (f_m):</td>
</tr>
<tr>
<td>[ \arg\min_{H_m} \frac{1}{n} \sum_{i=1}^{n} \left{ \frac{Y_i[A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} - H_m(X_i) \right}^2. ]</td>
</tr>
</tbody>
</table>
3.3.3 Deep convexified convolutional neural network for advantage learning

3.3.3.1 Convexified convolutional neural network

Zhang et al. [Zha16b] proposed convexified convolutional neural network (CCNN). If the activation function of CNN is smooth enough, the filter can be represented using RKHS. Some good choices of kernel functions include Gaussian kernel and inverse polynomial kernel. The parameter sharing properties of convolutional neural network results in low rank constraint, which can be relaxed to nuclear norm constraint. The network can be learned using convex optimization techniques. Compared to regular convolutional neural network, CCNN is computationally efficient and has ideal theoretical properties.

Denote \( X_i \in \mathbb{R}^{d_0} \) as input and \( y_i \) as output of convolutional neural network, \( i=1,2,...,n \). \( crop_i(X_i),..., crop_p(X_i) \) are \( P \) functions that create patches of size \( d_1 \) from input, \( X \) is the \( n \times p \) observation matrix with training sample \( X_i^{T} \) as its \( i \)-th row. \( \{ w_j \in \mathbb{R}^{d_1} \}_{r=1}^{r} \) are weight vectors where \( r \) is number of filters. \( \beta_{r \times p} \) is the filter-patch weight matrix. \( g(X) \) is the output of 2-layer convolutional neural network:

\[
g(X_i) = \sum_{j=1}^{r} \sum_{p=1}^{p} \beta_{j,p} \sigma(crop_p(X_i)^{T} w_j). \tag{3.1}
\]

Under proper choices of activation function, there exists \( \varphi \) s.t.

\[
\sigma(<w_j,z>) = <\hat{w}_j, \varphi(z)>
\]

43
holds where the reproducing kernel Hilbert space induced by kernel function $\kappa$ contains filters $z \rightarrow \sigma(<w, z>)$. The corresponding feature map $\varphi$ satisfies $\kappa(z, z') = <\varphi(z), \varphi(z')>$. $<>$ stands for inner product. $\bar{w}_j \in L_2(N)$ is a countable-dimensional vector. Since the parameters are estimated using only the training dataset, without loss of generality, we assume $\bar{w}_j \in \text{span}(\text{crop}_p(X_i)))_{i=1,2,\ldots,n}$. Denote the linear coefficients as $\gamma_j$. $Q(X) \in R^{nP \times s}$ is the factorization of kernel matrix $K$ of pairwise patches from training dataset, i.e. $K = Q(X)Q(X)^T$. Here $s$ is dimension of random feature approximation, which is explained in details in the next session. $Q(X_i) \in R^{P \times s}$ is the sub-matrix of $Q(X)$ corresponding to $X_i$. It can be shown that $<\bar{w}_j, \varphi(z)> = <Q^*(X)\nu(z), Q(X)^T \gamma_j>$, where $Q^*(X)$ is the pseudo-inverse of $Q(X)$, and $\nu(z)$ is a vector with each position as $\kappa(z, \text{crop}_p(X_i))$. Denote $Z(X_i)_{P \times nP} = \begin{pmatrix} Q^*(X)v(\text{crop}_1(X_i)) \\ Q^*(X)v(\text{crop}_2(X_i)) \\ \vdots \\ Q^*(X)v(\text{crop}_P(X_i)) \end{pmatrix}$. We have:

$$g(X_i) = \sum_{j=1}^{r} \beta_j^T Z(X_i)Q(X)^T \gamma_j = tr(Z(X_i)(\sum_{j=1}^{r} Q(X)^T \gamma_j \beta_j^T)) = tr(Z(X_i)B).$$

Two-layer CNN can be summarized as:

$$\hat{g} = \arg \min_{g \in G} \sum_{i=1}^{n} L(g(X_i); y_i)$$

$$G = \{g : \max_{j \in [r]} ||w_j||_2 \leq C_\sigma(B_1) \text{ and } \max_{j \in [r]} ||\beta_j||_2 \leq B_2 \text{ and } \text{rank}(B) = r \}.$$ 

where $C_\sigma$ is a monotonically increasing function that depends on the activation function, $L$ represents the loss function. To relax the non-convex constraints in (3.2), Zhang et al.
[Zha16b] considered the following class with the nuclear norm constraint:

\[
\hat{g}^B = \arg\min_{g^B \in G^B} \sum_{i=1}^{n} L(g^B(X_i); y_i)
\]

\[
G^B = \{ g^B : ||B||_* \leq C_\sigma (B_1 B_2 r) \}
\]

Since the optimization problem is transferred to a convex version, it is easier to compute and the resulting estimator has better theoretical properties.

### 3.3.3.2 A-learning based convexified convolutional neural network

We proposed a new approach which integrates CCNN with advantage learning. The contrast function is estimated by CCNN. For multi-layer CCNN, each layer is estimated in the bottom-up order. The low rank output of previous layer is fed to the next layer as input. For the current layer, a two-layer network with output \( \frac{Y_i[A_i - \pi(X_i)]}{\pi(X_i) [1 - \pi(X_i)]} \) is trained. If the number of channel is greater than 1, the processed patches are concatenated into one vector. This multi-channel extension technique makes it possible for extending 2-layer CCNN to multi-layer CCNN. Denote number of layers as \( m \), nuclear norm regularization parameters as \( R \). The algorithm is summarized in Algorithm 2.

\( Q(\mathcal{X}) \) can be calculated using Random Fourier Transformation proposed by Rahimi & Recht [RR08]:

\[
f_{RFT}: R^s \rightarrow R^s
\]

\[
f_{RFT}(X_i) = \begin{pmatrix}
\sqrt{\frac{\pi}{2}} \cos(w_{1}^T X_i + b_1) \\
\quad \vdots \\
\sqrt{\frac{\pi}{2}} \cos(w_{s}^T X_i + b_s)
\end{pmatrix}.
\]
We choose Gaussian kernel with parameter $\gamma$, then $w_1, \ldots, w_s \in R^s_0$ are i.i.d. samples from $N(0, 2\gamma I_{s \times s_0})$ and $b_1, \ldots, b_s$ are i.i.d. samples from Uniform$[0, 2\pi]$. Before training all weights and biases are randomly initialized. During the training process, we use least-square loss function to measure the difference between IPWE and output of neural network. The optimization with constraints in step 3 are achieved by projected gradient descent proposed by Duchi et al. [Duc08]. Parameters are updated using stochastic gradient descent followed by a projection onto the nuclear norm ball.

Algorithm 2: m-layer advantage learning based CCNN

<table>
<thead>
<tr>
<th>Input</th>
<th>$(X_i, A_i, y_i)_{i=1}^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>predictor ${tr(Z(H_{m-1}(X_i))\hat{B})}<em>{i=1}^n$ and optimal treatment regime ${I{tr(Z(H</em>{m-1}(X_i))\hat{B}) &gt; 0}}_{i=1}^n$</td>
</tr>
</tbody>
</table>

1. $H_1(\mathcal{X}) = \mathcal{X}$;

2. for $j \leftarrow 2$ to $m$ do

3. (1) Generate $P(H_{j-1}(\mathcal{X}))$, patches of $H_{j-1}(\mathcal{X})$.

4. (2) Use the random feature approximation to get $Q(H_{j-1}(\mathcal{X}))$, which corresponds to the decomposition of kernel matrix of $P(H_{j-1}(\mathcal{X}))$ and $Z(H_{j-1}(X_i))$.

5. (3) Use projected gradient descent to iteratively update $\hat{B}$:

$$\hat{B} = \arg\min_{\|\hat{B}\| \leq R} \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{Y_i[A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} - tr(Z(H_{j-1}(X_i))B) \right\}^2. \quad (3.3)$$

6. (4) Use singular value decomposition $\hat{B} = U \Lambda V^T$ to get the output $H_j(X_i)$ with $r$ filters. Here $\hat{U}$ is the first $r$ columns of $U$, $H_j(X_i) = \hat{U}^T (Z(H_{j-1}(X_i)))^T$. 

end
3.3.4 Multi-stage CCNN and CNN

This algorithm can be extended to dynamic treatment regime using backward induction. The framework is identical for CNN and CCNN: When estimating the decision rule for one stage, the outcome is adjusted as if during any later stages the optimal treatments have been received. The potential outcome is shifted based on contrast function estimation at each stage. Algorithm 3 is the procedure of multi-stage optimal treatment regime estimation with $K$ decision points. Note we use double subscripts here: the first subscript represents stage and the second one represents subject.

Algorithm 3: K-stage advantage learning based CCNN/CNN

<table>
<thead>
<tr>
<th>Input</th>
<th>$(\bar{A}<em>{K,i}, \bar{X}</em>{K,i}, y_i)_{i=1}^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>DTR $d_k^*(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}), i = 1, ..., n, k = 1, ..., K$</td>
</tr>
<tr>
<td>$V_{(k+1)i}$</td>
<td>$y_i$;</td>
</tr>
<tr>
<td>for $k \leftarrow K$ to 1 do</td>
<td></td>
</tr>
<tr>
<td>(1) Train CNN following Algorithm 1 or CCNN following Algorithm 2, inputs are $(\bar{A}<em>{ki}, \bar{X}</em>{ki}, V_{(k+1)i})_{i=1}^n$. The estimated contrast function at stage $k$</td>
<td></td>
</tr>
<tr>
<td>$f_k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) = \left{ \begin{array}{ll} H_k^k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) &amp; \text{for CNN} \ t_r(Z_k^k(H_{m-1}^k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki})) \hat{B}^k) &amp; \text{for CCNN} \end{array} \right.$ Here</td>
<td></td>
</tr>
<tr>
<td>superscripts are used to distinguish different stages. The estimated optimal treatment regime at stage $k$ is:</td>
<td></td>
</tr>
<tr>
<td>$d_k^*(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) = I{f_k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) &gt; 0}$.</td>
<td></td>
</tr>
<tr>
<td>(2) Update value function:</td>
<td></td>
</tr>
<tr>
<td>$V_k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) \leftarrow V_{k+1}(\bar{A}<em>{ki}, \bar{X}</em>{(k+1)i}) + f_k(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) \ast (d_k^*(\bar{A}<em>{(k-1)i}, \bar{X}</em>{ki}) - A_{ki})$.</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>
3.4 Simulation

We ran simulation studies to compare advantage learning based CCNN with existing popular methods. Our comparison is based on two stages situation. In training, validation and testing datasets, covariates $X_1$ and $X_2$, randomized treatment $A_1$ and $A_2$ are generated using STAR*D data (Details of the dataset is covered in the next section). This guarantees we simulate data that is close to true distribution. The response variable $Y$ is generated as follows:

$$y = A_1 A_2 + A_2 \sin(\beta_2^T [X_1; X_2]) + A_1 \sin(\beta_1^T X_1) + \epsilon,$$

where random error $\epsilon$ is generated independently from normal distribution with mean zero and standard deviation 0.1. $\bar{X}_2$ is obtained by stacking new information at each stage according to chronological order, i.e. $[X_1; X_2]$. We considered four scenarios with fixed coefficients generated from different distribution combinations:

Case 1: $\beta_{1i} \sim N(0, 1) \forall i=1,2,...,\text{dim}(\beta_1^T)$, $\beta_{2j} \sim N(0, 1) \forall j=1,2,...,\text{dim}(\beta_2^T)$;

Case 2: $\beta_{1i} \sim U[0, 1] \forall i=1,2,...,\text{dim}(\beta_1^T)$, $\beta_{2j} \sim N(0, 1) \forall j=1,2,...,\text{dim}(\beta_2^T)$;

Case 3: $\beta_{1i} \sim N(0, 1) \forall i=1,2,...,\text{dim}(\beta_1^T)$, $\beta_{2j} \sim U[0, 1] \forall j=1,2,...,\text{dim}(\beta_2^T)$;

Case 4: $\beta_{1i} \sim U[0, 1] \forall i=1,2,...,\text{dim}(\beta_1^T)$, $\beta_{2j} \sim U[0, 1] \forall j=1,2,...,\text{dim}(\beta_2^T)$.

Here $\text{dim}(X_1)$ and $\text{dim}([X_1; X_2])$ are dimensions of $\beta_1^T$ and $\beta_2^T$ respectively and each dimension follows independent and identical distribution. U stands for uniform distribution.

We compare the results of $l_1$ penalized least squares ($l_1$-pls). Lasso penalty is added to
least square loss (3.4) to avoid overfitting. The objective function (3.5) is optimized using scikit-learn module in python ([Ped11]). For both methods, tuning parameters are selected by maximizing the value function estimation on validation dataset.

\[
L(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{Y_i[A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} - \beta^T X_i \right\}^2, \quad (3.4)
\]

\[
\min_{\beta} L(\beta) + \lambda \sum_{j=1}^{p+1} |\beta_j|. \quad (3.5)
\]

For both methods, we assume the propensity score is a constant and estimate it by sample mean. We ran simulation using 50 Monte Carlo data sets and reported the mean value function on testing dataset based on different estimated decision rules. Figure 1 summarized simulation results.

Compared to \textit{l}_1-pls, CCNN has better performance in terms of overall potential outcome: Value function based on CCNN is larger than that of \textit{l}_1-pls: in case 1, the value ratio for CCNN and \textit{l}_1-pls is 1.34. We also notice that the difference between two methods in stage two is less than that of stage one. There are several reasons: the optimal decision rule at stage one is more intricate than that of stage two. Therefore neural network outperforms \textit{l}_1-pls in approximation of this highly non-linear function. Optimal decision rule at stage two can be written as:

\[
d_2^*(X_2, A_1) = I(A_1 + sin(\beta_2^T[X_1; X_2]) > 0).
\]

While optimal decision rule at stage one is more complicated. Another possible explanation may be optimal decision rule at stage one involves fewer covariates, resulting in less noise
and more signal. Nevertheless, when the true decision function can be approximated well enough by linear regression, it is possible that $l_1$-pls may achieve better results.

### 3.5 Application to STAR*D Study

In this section we demonstrate the performance of the proposed deep A-learning methods using the STAR*D clinical trial dataset. We compared mean population outcome following the estimated DTR using the deep A-learning and penalized least square estimator based on a linear decision rule.

#### 3.5.1 Dataset

Sequenced Treatment Alternatives to Relieve Depression (STAR*D) is the largest and longest clinical trial to compare the effectiveness of treatments for major depressive disorder. It has four levels. Each level lasts for 12 weeks. The severity of depression is measured by Quick Inventory of Depressive Symptomatology (QIDS) score. Participants without adequate clinical response at the end of each level would continue to the next stage. For level 1 all participants received citalopram. For each level of 2 - 4, patients received one randomized treatment. Covariates are collected from enrollment, IVR call, ROA interviews, clinic visit and other events (such as suicide, non-serious adverse event and protocol deviation). See [Fav03] for design and measurement details of STAR*D study.

#### 3.5.2 Processing

In general there are two types of options: switch to a different medication or adding on to their existing medication. Since all patients received the same treatment at level 1 and
the number of patients who entered level 4 is too small, we only focus on the 299 patients that has complete information at level 2 and level 3. Table 1 is the list of treatment switch and treatment augmentation options at the two levels. We take the negative level 3 16-item QIDS (QIDS-C16) as response variable $Y$. The propensity score is assumed to be constant and is estimated using sample mean. At each level, we remove a few covariates with small variance and reshape the covariate vector to a square matrix. The inputs of CNN/CCNN for level 2 are a $17 \times 17$ matrix and a $19 \times 19$ matrix for level 3 respectively.

Local normalization and zero-phase component analysis (ZCA-whitening) are incorporated for the input data. ZCA-whitening proposed by Krizhevsky & Hinton [KH09] is a popular technique for pre-processing of CNN. It can preserve local properties. ZCA-whitening can produced sphered and less-correlated covariates while transforming the data as little as possible. ZCA is summarized in (3.6) where $\Sigma$ is the covariance matrix, whose eigenvalues are $s_1, s_2, ..., s_r$ and close to the original variable linearly independent eigenvectors are column vectors of matrix $P$. The regularization parameter $\epsilon$ is added to avoid numerically
unstable situations.

\[
\Sigma = P \begin{pmatrix}
s_1 \\
s_2 \\ \\
\vdots \\
s_r \\
\end{pmatrix} P^T, \\
\]

\[
W = P \begin{pmatrix}
\frac{1}{\sqrt{s_1 + \epsilon}} \\
\frac{1}{\sqrt{s_2 + \epsilon}} \\
\vdots \\
\frac{1}{\sqrt{s_r + \epsilon}} \\
\end{pmatrix} P^T, \\
\]

\[X_{\text{transform}} = X W.\]

### 3.5.3 Architecture

The architecture of 2-layer CNN is as follows: the filter size of convolutional layer is 3 × 3 for stage 2 and 5 × 5 for stage 3 since number of covariates at level 3 is larger. The average pooling is implemented in the pooling layer with pooling patch size 2 × 2 and pooling stride 2. We do not use any padding techniques in order to control overfitting. Then a fully-connected layer maps all neurons to a scalar. For m-layer CNN, it has m-1 convolutional layers followed by one fully-connected layer. The number of filters is tuned by IPW estimator of value function:

\[
\hat{V}_{IPW} = \frac{1}{n} \sum_{i=1}^{n} \frac{y_i I\{A_{1,i} = d_1^*(X_{1i}), A_{2,i} = d_2^*(\bar{X}_{2i}, A_{1i})\}}{(1 - \pi_1(X_{1i}))^{1-A_{1,i}} \pi_1(X_{1i})^{A_{1,i}} (1 - \pi_2(\bar{X}_{2i}))^{1-A_{2,i}} \pi_2(\bar{X}_{2i})^{A_{2,i}}}
\]

The architecture of CCNN is very similar to CNN except that pooling is taken place before convolution operation. This can reduce the number of parameters in the neural network. The number of dimension of random feature approximation, the scale parameter of Gaus-
sian kernel and the nuclear norm constraint are tuned. Both CNN and CCNN are trained using mini-batch gradient descent. The learning rate of CNN and CCNN are 2e-3 and 2e-4, respectively. CNN is implemented using tensorflow ([Aba15]).

### 3.5.4 Results

We compare the results based on neural network with $l_1$-pls. Shrinkage parameter $\lambda$ is tuned based on BIC. To measure the performance of estimated DTR, we split the data into 3 parts: 60% of data set to be training data, 20% as validation and 20% as testing. Since parameters are randomly initialized, the accuracy of dynamic treatment regime varies each time. We trained each architecture 100 times using the training dataset and choose the best network based on the maximum value function estimation on the validation dataset. We report the value function estimation on the testing dataset. Table 2 summarized results of different architectures.

Based on the results, we have the following observations: Both CNN and CCNN can learn a dynamic treatment regime that outperforms $l_1$-pls. They use intricate functions to learn a decision rule. It is more accurate than that of penalized least square estimator. For fixed number of layers, CCNN has better performance than CNN. We also notice that processing has a great influence on decision accuracy. For example, without zca-whitening, both CNN and CCNN would have smaller estimated value function.

### 3.6 Conclusions and future work

In this chapter, we propose a deep A-learning that integrates CNN and CCNN with A-learning for optimal treatment regime estimation. Here our contrast function is estimated
based on IPW estimator. The new methods have the following advantages: It uses intricate functions to learn decision rules, which, allows for model complexity. The parameter sharing mechanism makes it computational efficient and controls overfitting. This method can be used in situations where medical images are available. The results indicate that the deep A-learning is competitive and outperforms penalized least square estimator. Although deep neural network has shown its competency in intricate scenarios, linear approximation based Q-learning or A-learning may still lead to a more accurate and computational-efficient solution when the true decision rule is simple.

In the future, more sophisticated algorithms such as RMSProp and Adagrad can be implemented to better estimate parameters in the neural network. Adaptive methods for hyperparameter specification is also worth studying. Complicated architectures which consist of multiple subnetworks have demonstrated themselves to be very powerful. An extension for the near future is the use of these architectures. Last but not least, the performance of neural network would improve if training sample size increases. In computer vision, techniques like rotation and flipping are widely used for data augmentation. Another line of future work is to investigate whether incorporating those techniques can help further improve the performance of optimal DTR estimation.
Figure 3.1 Value function given patients received estimated optimal treatments for both stages
Table 3.1 Lists of STAR*D treatment options at level 2 and level 3

<table>
<thead>
<tr>
<th>Type</th>
<th>Level</th>
<th>Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>2</td>
<td>Bupropion SR, Sertraline, Venlafaxine XR</td>
</tr>
<tr>
<td>Augmentation</td>
<td>2</td>
<td>Citalopram plus Bupropion SR, Citalopram plus Buspirone</td>
</tr>
<tr>
<td>Switch</td>
<td>3</td>
<td>Nortriptyline, Mirtazapine</td>
</tr>
<tr>
<td>Augmentation</td>
<td>3</td>
<td>Lithium augmentation, Triiodothyronine augmentation</td>
</tr>
</tbody>
</table>
Table 3.2 Evaluation results for estimated values on STAR*D dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of layers</th>
<th>Value function estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCNN</td>
<td>2</td>
<td>-7.22</td>
</tr>
<tr>
<td>CCNN</td>
<td>3</td>
<td>-4.57</td>
</tr>
<tr>
<td>CNN</td>
<td>2</td>
<td>-7.58</td>
</tr>
<tr>
<td>CNN</td>
<td>3</td>
<td>-5.92</td>
</tr>
<tr>
<td>CNN</td>
<td>4</td>
<td>-4.16</td>
</tr>
<tr>
<td>$l_1$-pls</td>
<td>-</td>
<td>-8.38</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


In this appendix we first state and prove two fundamental lemmas.
Proof of Lemma 0.1

**Lemma 0.1:** If condition (A1) and (A2) are satisfied, $\beta^*$ exists.

**Proof.** of Lemma 0.1:

$$
E \{ (w_i - w_j) I(w_i - w_j > 0) [1 - \beta^* (X_i^T - X_j^T)]_+ \}
$$

$$
= E \{ (w_i - w_j) I(w_i - w_j > 0) [1 - \beta^* (X_i^T - X_j^T)]_+ | X_i^T - X_j^T \}
$$

$$
L(\beta) = \int E \left[ (w_i - w_j) I(w_i - w_j > 0) | z \right] (1 - z^T \beta)_+ f^*(z) d z
\geq \int I(z^T \beta \leq 0) E \left[ (w_i - w_j) I(w_i - w_j > 0) | z \right] (1 - z^T \beta) f^*(z) d z
\geq \int I(z^T \beta \leq 0) E \left[ (w_i - w_j) I(w_i - w_j > 0) | z \right] (-z^T \beta) f^*(z) d z
\geq C_3 \int_{B(0, \delta_0)} I(z^T \beta \leq 0) (-z^T \beta) d z
= C_3 ||\beta|| \int_{B(0, \delta_0)} I(z^T w \leq 0) (-z^T w) d z
\geq C_3 ||\beta|| \text{vol}(B(0, \delta_0) \cap \{ -z^T w \geq \epsilon \}) \epsilon,
$$

where $w = \beta / ||\beta||$ and vol is short for volume. Note that $\text{vol}(B(0, \delta_0) \cap \{ -z^T w \geq \epsilon \}) > 0$ for some $\epsilon < \delta_0$ and $\text{vol}(B(0, \delta_0) \cap \{ -z^T w \geq \epsilon \})$ is independent of $\beta$. Thus $L(\beta) \to \infty$ as $||\beta|| \to \infty$. Since $L(\beta)$ is convex in $\beta$, the solution $\beta^*$ exists. $\blacksquare$
Proof of Lemma 0.2

Lemma 0.2: Condition (A3) implies $\beta^* \neq 0$.

Proof. of Lemma 0.2:
Without loss of generality, suppose $\int E[(w_i - w_j)I(w_i - w_j > 0)|z]z_k f^*(z)dz > 0$, then for $\beta_k^* > 0$,

$$L(0, ..., \beta_k^*, 0, ... 0) = \int E[(w_i - w_j)I(w_i - w_j > 0)|z](1 - \beta_k^* z_k)f(1 - \beta_k^* z_k > 0)f^*(z)dz$$

$$= \int E[(w_i - w_j)I(w_i - w_j > 0)|z]I(z_k < 1/\beta_k^*)f^*(z)dz$$

$$- \beta_k^* \int E[(w_i - w_j)I(w_i - w_j > 0)|z]z_k I(z_k < 1/\beta_k^*)f^*(z)dz.$$

The second term is non-negative for a sufficient small $\beta_k^* > 0$.

Therefore,

$$L(0, ..., \beta_k^*, 0, ... 0) < \int E[(w_i - w_j)I(w_i - w_j > 0)|z]f^*(z)dz = L(0, ..., 0).$$
Proof of Lemma 1

Proof of Lemma 1: Recall that

\[ \hat{S}(\beta) = \frac{-2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0] (X_i - X_j) \].

Note that

\[ \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0] (X_{ik} - X_{jk}) \]

\[ = \frac{1}{n(n-1)} \sum_{i \neq j} q_1((w_i, X_i), (w_j, X_j)), \]

where

\[ q_1((w_i, X_i), (w_j, X_j)) = \begin{cases} 
(w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0] (X_{ik} - X_{jk}) & \text{if } i < j, \\
q_1((w_j, X_j), (w_i, X_i)) & \text{if } i > j. 
\end{cases} \]

By Hoeffding's inequality for u-statistics of order 2, which can be found in [Pee10]:

\[ P(\sqrt{32A(\alpha)\log p}^3/n \leq \frac{2}{n(n-1)} | \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0] (X_{ik} - X_{jk}) |) \]

\[ \leq 2e^{-\frac{32A(\alpha)\log p^3}{2n(4M_0\alpha)^3M^3}} + ne^{-\frac{w_M}{M_3}}. \]

Let \( w_M = A(\alpha)^{1/3} \log p M_0^{-2/3} M_3^{1/3} \), then
\[ e^{-\frac{32\alpha\log p}{32nM_n^2 \beta_M^2}} = e^{-\frac{w_M}{3\beta_M^2}}, \]

\[ P\left( \frac{\sqrt{32}\alpha(\log p)^{3/2}}{n} \leq \frac{2}{n(n-1)} \left| \sum_{i=1}^{n} \sum_{j=1+1}^{n} (w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0](X_{ik} - X_{jk}) \right| \right) \]
\[ \leq (2 + n) p^{-\frac{A(\alpha)}{2}M_n^{-2/3}M_3^{-2/3}}, \]

\[ P\left( c \frac{\sqrt{32}\alpha(\log p)^{3/2}}{n} \leq c \| \hat{S}(\beta^*) \|_\infty \right) \]
\[ \leq \sum_{k=1}^{p} P\left( \frac{\sqrt{32}\alpha(\log p)^{3/2}}{n} \leq \frac{2}{n(n-1)} \left| \sum_{i=1}^{n} \sum_{j=1+1}^{n} (w_i - w_j) I[1 - (X_i^T - X_j^T)\beta \geq 0](X_{ik} - X_{jk}) \right| \right) \]
\[ \leq (2 + n) p^{-\frac{A(\alpha)}{2}M_n^{-2/3}M_3^{-2/3}+1} \leq \alpha. \]
Proof of Lemma 2

Proof of Lemma 2:
Let \( q_2((w_i, X_i), (w_j, X_j)) = \)
\[
\begin{cases}
(w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ - (w_i - w_j)[1 - (X_i^T - X_j^T)\beta^*]_+ & \text{if } i < j, \\
q_2((w_j, X_j), (w_i, X_i)) & \text{if } i > j.
\end{cases}
\]

and \( \hat{U}_n(w_n, X_n) = \frac{1}{n(n-1)} \sum_{i \neq j} q_2((w_i, X_i), (w_j, X_j)) \). It is a u-statistics of order 2.

\[
\left|(w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ - (w_i - w_j)[1 - (X_i^T - X_j^T)\beta^*]_+ \right| \leq 2w_M(X_i^T - X_j^T)h
\]
holds with at least probability \( 1 - ne^{-\frac{w_M}{M}} \), \( \forall 1 \leq i, j \leq n \).

By Hoeffding’s inequality,
\[
P\left( \frac{B(h)}{\|h\|_2} \geq \frac{t}{\sqrt{n}} |X| \right) \leq 2e^{\left(-\frac{t^2}{32M_0^2 w_M^2 q}\right)}
\]
holds with at least probability \( 1 - ne^{-\frac{w_M}{M}} \), \( \forall 1 \leq i, j \leq n \). Let \( t = C\sqrt{32\log p q w_M} \), then
where $N$ is an $\varepsilon$-net to cover \( \{ h \in R^p, ||h||_0 \leq q, ||h||_2 \neq 0 \} \), for any \( h_1, h_2 \) within the same $\varepsilon$-ball and $||h_1||_2 \neq 0$ and $||h_2||_2 \neq 0$, \( \frac{h_1 - h_2}{||h_1||_2} \leq \varepsilon \) holds.

We also have

\[
P\left( \sup_{h \in N} \frac{B(h)}{||h||_2} \geq C \sqrt{32 \log \frac{p}{n}} q M \right) \leq 2p^{-\frac{c^2}{36}},
\]

\[
P\left( \sup_{h \in R^p, ||h||_0 \leq q, ||h||_2 \neq 0} \frac{B(h)}{||h||_2} \geq C \sqrt{32 \log \frac{p}{n}} q M \right) \leq 2 \left( \frac{3}{\varepsilon} p^{1-\frac{c^2}{36}} \right)^q,
\]

holds with probability at least $1 - ne^{-\frac{wM}{M_1}}$.

\[
\frac{B(h)_{||h||_2}}{||h||_2 \leq q, ||h||_2 \neq 0} \leq \frac{B(h)_{||h||_2}}{h \in N} + 8wM\sqrt{M_1} \varepsilon
\]

holds with probability at least $1 - ne^{-\frac{wM}{M_1}}$.
Let $\varepsilon = q \sqrt{\frac{32 \log p}{n}} \frac{1}{8 \sqrt{M_1}}$, we have that

$$P\left( \sup_{\|h\| \leq q, \|h\|_2 \neq 0} \frac{B(h)}{\|h\|_2} \geq C q \sqrt{\frac{32 \log p}{n}} w_M \right) \leq P\left( \sup_{h \in N, \|h\|_2 \neq 0} \frac{B(h)}{\|h\|_2} \geq (C - 1) q \sqrt{\frac{32 \log p}{n}} w_M + 2 ne^{\left( \log 2 - \frac{w_M}{M_3} \right)} \right).$$

Since $p > n$ and take $C = 1 + C_2 M_0$ for some $C_2 \geq \sqrt{2}$, for sufficiently large $n$:

$$P\left( \sup_{\|h\| \leq q, \|h\|_2 \neq 0} \frac{B(h)}{\|h\|_2} \geq (1 + C_2 M_0) q \sqrt{\frac{32 \log p}{n}} w_M \right) \leq 2p^{\frac{w_M}{M_3}} - q^{(C_2^2 - 2)} \log 2n - \frac{w_M}{M_3}.$$

Take $w_M = M_3 q(C_2^2 - 2) \log p + M_3 \log 2n$, we have that

$$P\left( \sup_{\|h\| \leq q, \|h\|_2 \neq 0} \frac{B(h)}{\|h\|_2} \geq (1 + C_2 M_0) q \sqrt{\frac{32 \log p}{n}} \left[ M_3 q(C_2^2 - 2) \log p + M_3 \log 2n \right] \right) \leq 3p^{\frac{w_M}{M_3}} - q^{(C_2^2 - 2)}.$$
Proof of Lemma 3

Proof of Lemma 3.

By the definition of \( \beta^* \), we have

\[
\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1-(X_i^T - X_j^T)\hat{\beta}]_+ + \lambda ||\hat{\beta}||_1 \leq \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1-(X_i^T - X_j^T)\beta^*_+] + \lambda ||\beta^*||_1
\]

\[
\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1-(X_i^T - X_j^T)\beta^*_+] \leq \lambda ||\beta^*||_1 - \lambda ||\hat{\beta}||_1.
\]  

(7)

We can also show that

\[
||\beta^*||_1 - ||\hat{\beta}||_1 \leq ||h_T||_1 - ||h_{Tr}||_1,
\]

(8)

\[
\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1-(X_i^T - X_j^T)\hat{\beta}]_+ - \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1-(X_i^T - X_j^T)\beta^*_+] \geq \hat{S}^T(\beta^*)h \geq -||h||_1||\hat{S}(\beta^*)||_\infty \geq -\frac{\lambda}{c}(||h_T||_1 + ||h_{Tr}||_1).
\]

(9)
Based on (7), (3.6) and (9), we have

$$\|h_T\|_1 \geq \frac{c - 1}{c + 1} \|h_{T^c}\|_1.$$
Proof of Lemma 4

Proof. of Theorem 4:
Assume $|h_1| \geq |h_2| \geq \ldots \geq |h_p|$, then for the partition: $S_0 = \{1, 2, \ldots, q\}, S_1 = \{q+1, q+2, \ldots, 2q\}, \ldots,$

$$||h_{S_0}||_1 \geq ||h_{T'}||_1 \geq \bar{c}||h_{T'C}||_1 \geq \bar{c}||h_{S_0^c}||_1$$

holds. We have

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ -$$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)\beta^*]_+$$

$$= \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h_{S_0})]_+ -$$

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)\beta^*]_+$$

$$+ \frac{2}{n(n-1)} \sum_{l \geq 1} \left\{ \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=0}^{l-1} h_{S_k})]_+ \right\}-$$

$$- \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=0}^{l-1} h_{S_k})]_+. $$
By Lemma 2 we have with at least probability \(1 - 3p^{-q(C_2^2 - 2)}\),

\[
E\left\{ (w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=1}^l h_{S_k})]_+ \right\}
- E\left\{ (w_i - w_j)I(w_i - w_j > 0)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=1}^{l-1} h_{S_k})]_+ \right\}
\leq \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=1}^l h_{S_k})]_+ 
- \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - \sum_{k=1}^{l-1} h_{S_k})]_+ 
+ C_4 q^2 \sqrt{\frac{(\log p)^3}{n}} ||h_{S_0}||_2.
\]

It holds for every \(l\), therefore

\[
\frac{2}{n(n-1)} E\left\{ \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ \right\} 
- \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)\beta^*]_+ 
\leq \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ 
- \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n (w_i - w_j)[1 - (X_i^T - X_j^T)(\beta^* - h)]_+ 
+ C_4 q^2 \sqrt{\frac{(\log p)^3}{n}} \sum_{j=0}^n ||h_{S_j}||_2
\leq \lambda(||h_T||_1 - ||h_{TC}||_1) + C_4 q^2 \sqrt{\frac{(\log p)^3}{n}} ||h_{S_0}||_2 + \sum_{j=1}^n C_4 q^2 \sqrt{\frac{(\log p)^3}{n}} ||h_{S_j}||_2
\leq \lambda \sqrt{q} ||h_{S_0}||_2 + C_4 q^2 \sqrt{\frac{(\log p)^3}{n}} \left(\frac{5}{4} + \frac{1}{c}\right) ||h_{S_0}||_2
\]

holds with probability at least \(1 - 3p^{-q(C_2^2 - 2)}\).
The last inequality holds by,

\[
\sum_{j \geq 1} \| h_{S_j} \|_2 \leq \sum_{j \geq 1} \frac{\sqrt{q}}{\sqrt{q}} \frac{\| h_{S_j} \|_1}{4} \leq \frac{\| h_{S_j} \|_1}{4} + \frac{\| h_{S_j} \|_1}{4 \sqrt{q}} \\
\leq \left( \frac{1}{\sqrt{q} \bar{c}} + \frac{1}{4 \sqrt{q}} \right) \| h_{S_0} \|_1 \leq \frac{1}{4} \| h_{S_0} \|_2.
\]

And more details can be found in [Cai10].

By Taylor expansion of \( L(\beta) \) around \( \beta^* \):

\[
\frac{2}{n(n-1)} E \left\{ \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j) [1 - (X_i^T - X_j^T)(\beta^* - h)]_+ - \sum_{i=1}^{n} \sum_{j=i+1}^{n} (w_i - w_j) [1 - (X_i^T - X_j^T)\beta^*]_+ \right\} \\
= \frac{1}{2} h^T H(\beta^*) h + o_p(\|h\|_2^2) \geq \frac{1}{2} M_2 \| h \|_2^2 + o_p(\|h\|_2^2).
\]

Then we have

\[
\frac{1}{2} M_2 \| h \|_2^2 + o_p(\|h\|_2^2) \leq \lambda \sqrt{q} \| h_{S_0} \|_2 + \sqrt{ \frac{\log p}{n} } \left( \frac{5}{4} + \frac{1}{\bar{c}} \right) \| h_{S_0} \|_2.
\]

On the other hand,
\[ \| h \|_2^2 = \| h_{S_0} \|_2^2 + \sum_{j \geq 1} \| h_{S_j} \|_2^2 \leq \| h_{S_0} \|_2^2 + |h_q| \sum_{j \geq 1} \| h_{S_j} \|_1 \leq \| h_{S_0} \|_2^2 + \frac{1}{c} \| h_{S_0} \|_1 |h_q| \leq (1 + \frac{1}{c}) \| h_{S_0} \|_2^2. \]

Therefore,

\[ \| h_{S_0} \|_2^2 \leq \| h \|_2^2 \leq (1 + \frac{1}{c}) \| h_{S_0} \|_2^2, \]

\[ o(\| h \|_2^2) = o(\| h_{S_0} \|_2^2). \]

Hence

\[ \frac{1}{2} M_s \| h_{S_0} \|_2 + o_p(\| h_{S_0} \|_2) \leq \lambda \sqrt{q} + C q^2 \sqrt{\frac{(\log p)^3}{n}} \left( \frac{5}{4} + \frac{1}{c} \right), \]

\[ \| h \|_2 + o_p(\| h \|_2) \leq \sqrt{1 + \frac{1}{c} \left[ \frac{2(\lambda \sqrt{q})}{M_s} + \frac{2C q^2}{M_s} \sqrt{\frac{(\log p)^3}{n}} \left( \frac{5}{4} + \frac{1}{c} \right) \right]}. \]

That is,

\[ \| \hat{\beta} - \beta^* \|_2 \leq \sqrt{1 + \frac{1}{c} \left[ \frac{2(\lambda \sqrt{q})}{M_s} + \frac{2C q^2}{M_s} \sqrt{\frac{(\log p)^3}{n}} \left( \frac{5}{4} + \frac{1}{c} \right) \right]}. \]

with probability at least \( 1 - 3p^{-q(C_2^2 - 2)+1} \).
Appendix B.

We now show that

$$\min \sum_{i \neq j} (w_i - w_j)I(\beta^T X_i < \beta^T X_j),$$

subject to $||\beta|| = 1$.

is equivalent to

$$\min \sum_{w_i > w_j} (w_i - w_j)I(\beta^T X_i < \beta^T X_j),$$

subject to $||\beta|| = 1$.

Suppose $\beta^T X_i \neq \beta^T X_j, \forall i, j$, we have

$$\sum_{i \neq j} (w_i - w_j)I(\beta^T X_i < \beta^T X_j)$$

$$\sum_{w_i > w_j} (w_i - w_j)I(\beta^T X_i < \beta^T X_j) + \sum_{w_i > w_j} (w_j - w_i)I(\beta^T X_j < \beta^T X_i)$$

$$\sum_{w_i > w_j} (w_i - w_j)[I(\beta^T X_i - \beta^T X_j < 0) - I(\beta^T X_i - \beta^T X_j > 0)]$$

$$\sum_{w_i > w_j} (w_i - w_j)[I(\beta^T X_i - \beta^T X_j < 0) - 1 + I(\beta^T X_i - \beta^T X_j < 0)]$$

$$\sum_{w_i > w_j} (w_i - w_j)[2I(\beta^T X_i - \beta^T X_j < 0) - 1].$$
Appendix C.

We now show that given $X_i$, $w_i$ is an unbiased estimator of $D(X_i)$. Specifically,

$$E\left\{ \frac{[Y_i - \nu(X_i)][A_i - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} \mid X_i \right\}$$

$$= E \left\{ \frac{[Y_i - \nu(X_i)][1 - \pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} \mid A_i = 1, X_i \right\} \pi(X_i)$$

$$+ E \left\{ \frac{[Y_i - \nu(X_i)][-\pi(X_i)]}{\pi(X_i)[1 - \pi(X_i)]} \mid A_i = 0, X_i \right\} [1 - \pi(X_i)]$$

$$= E[Y_i|A_i = 1, X_i] - E[Y_i|A_i = 0, X_i] = D(X_i).$$