2020

Novel Random Forest Methods and Algorithms for Autism Spectrum Disorders Research

Afrooz Jahedi
Claremont Graduate University

NOVEL RANDOM FOREST METHODS AND ALGORITHMS FOR AUTISM SPECTRUM DISORDERS RESEARCH

A Dissertation

Presented to the Faculty of

Claremont Graduate University

and

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in

Computational Science - Statistics

---

by

AFROOZ JAHEDI

Spring 2020

APPROVAL OF THE DISSERTATION COMMITTEE

This dissertation has been duly read, reviewed, and critiqued by the Committee listed below,

which hereby approves the manuscript of

Afrooz Jahedi

as fulfilling the scope and quality requirements for meriting the degree of

Doctor of Philosophy

Juanjuan Fan, Chair
Department of Mathematics and Statistics, San Diego State University
Professor

Ralph-Axel Müller
Department of Psychology, San Diego State University
Professor

Barbara Bailey
Department of Mathematics and Statistics, San Diego State University
Associate Professor

John Angus
Institute of Mathematical Sciences, Claremont Graduate University
Professor

Allon Percus
Institute of Mathematical Sciences, Claremont Graduate University
Professor

ABSTRACT OF THE DISSERTATION

NOVEL RANDOM FOREST METHODS AND ALGORITHMS
FOR AUTISM SPECTRUM DISORDERS RESEARCH

by
AFROOZ JAHEDI
Doctor of Philosophy in Computational Science - Statistics
Claremont Graduate University and San Diego State University, 2020

Random Forest (RF) is a flexible, easy to use machine learning algorithm that was proposed by Leo Breiman in 2001 for building a predictor ensemble with a set of decision trees that grow in randomly selected subspaces of data. Its superior prediction accuracy has made it the most used algorithms in the machine learning field. In this dissertation, we use the random forest as the main building block for creating a proximity matrix for multivariate matching and diagnostic classification problems that are used for autism research (as an exemplary application).

In observational studies, matching is used to optimize balance between treatment groups. Although many matching algorithms can achieve this goal, in some fields, matching could face its own challenges. Datasets with small sample sizes and limited control reservoirs are prone to this issue. This problem may apply to many ongoing research fields, such as autism spectrum disorder (ASD). We are interested in eliminating the effect of undesirable variables using two types of algorithms, 1:k nearest matching, and full matching. Therefore, we first introduced three different types of 1:k nearest matching algorithms and two full matching based methods to compare group-wise matching vs. pairwise matching for creating an optimal balance and sample size. These proposed methods were applied to a data set from the Brain Development Imaging Lab (BDIL) at San Diego State University. Next, we introduce the iterMatch R package. This package finds a 1:1 matched subsample of the data that is balanced on all matching variables while incorporating missing values in an iterative manner. Missing variables in dataset need to be imputed or only complete cases can be

considered in matching. Losing data because of the limitations in a matching algorithm can decrease the power of the study as well as omit important information. Other than introducing the iterMatch package, tuning the input parameters of this package is discussed, using medium and large datasets from the Autism Brain Imaging Data Exchange (ABIDE).

We then propose two mixed-effects random forest-based classification algorithms applicable to multi-site (clustered data) using resting-state fMRI (rs-fMRI) and structural MRI (sMRI). These algorithms control the random effects of the confounding factor of the site and fixed effect of phenotype variable of age internally while building the prediction model. On top of controlling the effects of confounding variables, these algorithms take away the necessity of utilizing a separate dimension reduction algorithm for high dimensional data such as functional connectivity in a non-linear fashion. We show the proposed algorithms can achieve prediction accuracy over 80 percent using test data.

DEDICATION

I would like to dedicate this dissertation to my family. A special thanks to my husband, Dr. Reza Banirazi, for his constant love and support. I would also like to thank my family, especially my parents, for their endless emotional support and encouragement during my studies in the US.

It's tough to make predictions, especially about future.

– Yogi Berra

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to both advisors, Professor Juanjuan Fan and Professor Ralph-Axel Müller, for their continuous guidance, mentorship, and support. I have greatly enjoyed our meetings and getting to know them over these past few years. Dr. Fan is a wonderful mentor and friend. The combination of her knowledge and expertise made her a great mentor for me during the last five years. I enjoyed many suggestions and ideas that helped make this dissertation possible. My unique thanks to Dr. Müller for the 5-year Financial support granted by NIH that made my research possible. His character and research experience was always a great asset for me to become a better researcher and I hope our collaboration continues for many years to come.

I would also like to express my thanks to my doctoral committee: Dr. Barbara Bailey, Dr. John Angus, and Dr. Allon Percus. Their advice and feedback over the years has been tremendously helpful, and I am indebted to them for taking the time to review this dissertation.

I would also like to thank Dr. James Otto for his help in providing timely support on computational resources at CSRC. Special thanks to Dr. Jose Castillo for providing help to fund the cost associated with the joint SDSU-CGU Ph.D. program. Finally, I would like to thank BDIL faculties Dr. Inna Fishman and Dr. Ruth Carper and other lab members for providing help with data acquisitions.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

PAGE

CHAPTER 1
INTRODUCTION

1.1 GROUP-WISE VS. PAIR-WISE MATCHING

Autism spectrum disorder (ASD), as the name suggests, include a range of highly heterogeneous disorder diagnosed based on behavioral criteria. Despite consensus on the neurobiological nature of ASD, brain biomarkers remain unknown. To decipher the imaging data from functional connectivity MRI (fcMRI) scans, for example, one must first eliminate the undesirable effect of important background variables before any pattern recognition procedures are performed. Head motion in the MRI scanner is considered as one of the most important contamination sources in the resting-state fcMRI. Head motion systematically alters correlations in resting-state functional connectivity MRI [50]. Therefore, it is crucial to eliminate the effect of head motion before proceeding with any further analysis. Age and non-verbal IQ (NVIQ) are considered two other main variables that could affect autism. Bishop et al. observed a declining trend of NVIQ in individuals with ASD between toddler-hood and young adulthood [6].

One of the existing solutions for controlling the effect of confounding variable is matching. Matching ensures that the two exposure groups have similar distributions with respect to confounding and other risk factors, both of which we will refer to as covariates. In an observational study, treated participants are often matched to control participants with similar values of covariates in an effort to form treatment groups that are comparable [16]. While matching is generally used to estimate causal effects, it is also used for non-causal questions. For example, matching was used to investigate non-causal questions regarding racial disparities [60]. Though there are many matching algorithms that could be used to select treated and control groups that are comparable in terms of covariates, to the best of our

knowledge, none of these could readily address the situation where the size of the control group is not much bigger than that of the treatment group, and both samples are small or modest. To obtain good matching results, most matching algorithms require a large number of control participants or a ”control reservoir”. Therefore, matching algorithm that can handle dataset with limited control reservoir is desired.

Random forest (RF) [8] has been shown to have excellent predictive accuracy compared to many statistical and machine learning methods, including logistic regression, boosting, support vector machine, and artificial neural network, for example, [14]; [63]; [13] and [12].

The random forest can also be used to readily produce a proximity matrix among participants using categorical covariates with more than two categories without any difficulties, while the Mahalanobis distance used by traditional matching algorithms is well defined only for continuous variables and categorical variables with only two categories. Since RF is a model ensemble tool that is non-parametric in nature, it can be used to provide a more accurate and less model-dependent estimate for the propensity score. In particular, random forest provides automatic variable selection, can handle complicated relationships without the need to manually include non-linear transformations or high-order interactions among input variables. Though not faced in our specific application, random forest can also deal with missing data via surrogate splits without the need for data imputation beforehand [68]. For these reasons, the propensity score and distance measures for the proposed matching algorithms will be calculated based on random forest.

In Chapter 2, Section 2.2 and Section 2.3 will summarize random forest as relevant to this chapter and describe the calculation of propensity score and proximity matrix based on random forest. Sections 2.4.1 to Section 2.4.5 will describe the proposed three matching methods, including three group matching and two iterative matching algorithms. Section 2.5.1 will present an autism data set collected by the Brain Development Imaging Lab (BDIL) at

the San Diego State University (SDSU). Section 2.5.2 will present the matching results of the BDIL data using the proposed matching methods.

1.2 iterMatch R Package

After introducing different matching algorithms and evaluating the performance of each one in Chapter 3, *iterMatch* is chosen as the best performing algorithms for R package implementation with some new features. The detail is discussed in Chapter 3 with an introduction given below.

Observational studies require matching to eliminate the effect of undesirable variables to achieve randomized design. The design for observational studies is based on the comparison between two groups of participants. Matching is an effective tool to create a balance among multiple confounding variables, and has been used for many different research fields such as Psychology, Social Science, Medicine, Economics, and Political science. Although many matching algorithms can fulfill the purpose of balancing confounding variables, many research studies continue to rely on manual matching [48], [46], and [15]. Several problems are associated with using manual matching such as leading to suboptimal subjective and non-reproducible results. Although manually matching is not complicated on small numbers of variables, it rapidly becomes a complicated, time-consuming problem as the number of variables increases. Therefore,it is necessary to have matching algorithm when more matching variables are involved.

In recent years some research groups started to use the existing matching algorithms and some groups started to create their customized matching algorithm to create a well-balanced sample in terms of reducing the effect of confounding variables. For example, riddle et al. [52] matched each participant on age and sex within each brain site, using the Case-control matching feature on SPSS. Both Uddin et al. [65] and Supekar[64] used a customized group matching algorithm to match each ASD and Typically Developing (TD) participant on three variables. Although the proposed algorithms could address reproducibility, complexity, and time consuming issues of the manual matching, the challenge

of including participants with a partial missing values remains unsolved. Moreover, the customized algorithms might not be publicly available for other research groups to use.

As we stated earlier, one of the main challenges to most matching algorithms is dealing with missing values. The recommended way of dealing with missing values in datasets is either to remove observations with more than one missing instance entirely or impute those missing values with one of various methods. For example, Iacus et al. [35] proposed a matching algorithm that can only input a complete dataset. However, the issue with using only complete data is that it may result in the loss of crucial information and decreasing the statistical power of the study.

Choosing the appropriate imputation method is a challenging problem. Missing value imputation requires user's knowledge of several imputation techniques and understanding of finding the appropriate method. Furthermore, there are variables that can not be imputed, such as ethnicity or gender, cycling back to the removal of an observation.

To overcome these problems, we adjusted the *IterMatch* algorithm introduce in Chapter 2 to iterative matching with missing values. The *IterMatch* algorithm uses the OPTMATCH package [31] to find a one-to-one match where a distance matrix is fitted as an input object. The main goal of Chapter 3 is to introduce a publicly available R package, iterMatch, as a one of the matching methods for observational data that can handle missing values internally without using a separate imputation technique. We further discussed the main factors that could affect finding matched samples such that for the desirable matching threshold, the maximum number of retained samples from both groups can be retained.

In Chapter 3, we will first introduce the building blocks of this package — random forest — in the Section 3.2. Following this, the *iterMatch* algorithm that has the capability of handling missing values will be explained in Section 3.3. Section 3.4 will introduce two methods of calculating balance measure or Standardized Mean Difference (SMD). Methods for handling missing values will be explained in Section 3.5. Next, in section 3.6.1, two datasets with medium and large sample sizes will be selected from the Autism Brain Imaging

Data Exchange (ABIDE-I & ABIDE-II) on five matching variables. Finally, results from tuning input parameters from the *IterMatch* algorithm will be provided in Section 3.7.

The package is ready to be uploaded at Comprehensive R archive Network (CRAN) [51].

1.3 Mixed-Effects Random Forest-Based Classification Algorithms for Clustered Data

Autism Spectrum Disorder (ASD) includes an array of neurodevelopmental disorders that are characterized by social and communicative deficits and repetitive behavior [3]. The prevalence of autism is increasing as it has been reported to affect approximately 1:68 of the population [4].

In recent years, there has been a growing trend in designing neuroimaging-based prognostic/diagnostic tools. As a result, there have been many efforts using neuroimaging tools to discriminate patients with brain disorders from healthy control automatically [40]. Many of these studies have reported promising models that claim a robust, accurate, and rapid diagnostic prediction in an automatic fashion. Despite the promising results on specific research datasets, these tools have yet to be integrated into the clinical realm. We believe multiple issues should be addressed before any further progress can be made.

The first issue is controlling the variation between imaging sites, specifically when observations within each imaging site have their own data acquisition protocol such as scanner parameter acquisition and MRI scanner type. Differences in data acquisition protocol can increase the heterogeneity of the sources between imaging sites. Moreover, it is necessary to increase the data sample in which observations are nested within imaging sites or clusters. For example, multi-site databases such as Autism Brain Imaging Data Exchange (ABIDE) are considered clustered data which is frequently used for many autism research studies. Clustered data are often obtained by multistage sampling with observations nested within higher-level units (clusters). These data may include two types of covariates, observation-level and cluster-level covariates, and involve two sources of variation, within and

between clusters. The within-cluster variation is controlled by a fixed or population-averaged component of the model, and between-cluster is handled by the random part. Usually, observations that belong to the same cluster tend to be more similar to each other than observations from different clusters. Therefore, it is crucial to control the induced source of variation from the different imaging sites in ABIDE dataset.

A common way to handle the clustered nature of multi-site data and the correlation between imaging sites is to use Linear Mixed-Effects (LME)[42] models. Although utilizing mixed models is one of the possible solutions to handle the correlation between imaging sites, the non-linear and high-dimensional nature of imaging data eliminates the applicability of such models. Using tree-based models could address the non-linear and high-dimensional nature of complex data such as neuroimaging data.

In recent years, tree-based models have been very successful in handling the non-linear models. Tree-based models are a set of algorithms that empower predictive models with high accuracy, stability, and ease of interpretation. Several methods have been proposed to modify decision tree and random forest models for longitudinal and clustered continuous outcomes [1], [19],[44], [45], [61], [62], [67]. Abdolell et al. first proposed the idea of combining the tree-based model and mixed-effects model in 2002 by proposing longitudinal recursive partitioning [1], which permits a non-linear mixed-effects model, in addition to a linear mixed-effects model in each node. In 2009, [38], and in 2011, [2] two methods were proposed in which the sampling method for clustering was adjusted, but random effects in the predictions were not incorporated. In 2012, Sela et al. [62], proposed *REEM* trees where a single tree incorporated with mixed-effects was proposed for a Gaussian outcome. They also released an R package called REEMtree. Later Hajjem et al. [28] proposed a similar tree approach to handle cluster-level predictors and individual-level predictors simultaneously. These methods incorporated mixed-effects within the tree framework to account for only clustering effects, using an iteration between the fixed part and the random part to estimate the parameters through an Expectation-Maximization (EM) algorithm, analogous to

expectation-maximization described by Wu and Zhang [66]. Hajjem [29] extended the decision tree method to the random forest setting for clustered and longitudinal continuous outcomes referred to as Mixed Effects Random Forest (MERF). In 2018, Calhoun [**?**] proposed the repeated measures random forest (RMRF) algorithm that extends the standard random forest implementation to handle longitudinal designs. In 2019, Capitaine [11] added a stochastic component to MERF model, which could be used for high-dimensional and longitudinal data. Given the provided rich history of semi-parametric mixed-effects models, we extend the tree-based models by proposing two new algorithms that can handle the heterogeneity of clustered data uniquely which will be explained in later sections.

In addition to the aforementioned issues, extended sample and controlling the variability between imaging sites, research with a feature-rich data requires utilizing one form of dimension reduction algorithm separately before building a classification model. Including multiple modalities in a dataset increases the coherence of the brain picture. Reducing the dimensionality internally without losing information in a consistent way becomes more critical when multiple modalities contain different numbers of variables in building a model. In our study, we included two MRI modalities, Structural MRI (sMRI) and functional MRI (fMRI). MRI related techniques such as sMRI and fMRI have the benefit of providing localized spatial information about brain structure and functional connectivity, respectively. Including both of these modalities is important because resting-state functional Magnetic Resonance Imaging (rs-fMRI) holds the promise to reveal functional biomarkers of ASD by measuring the brain activity using blood oxygenation changes [21];[47]. fMRI makes it possible to study functional regions and networks of the brain as well as temporal associations among them. On the other hand, sMRI, as a high-resolution image of the brain, has made it possible to detect physical abnormalities, lesions, and damage. Hence, it is important to deal with the issue of dimension reduction of multi-modal data in a less complex and time-consuming way that can preserve important variables from thousands of variables in the brain.

Therefore, in Chapter 4, the two mentioned shortcomings will be addressed by proposing two binary classification algorithms for unbalanced clustered structure data. These algorithms internally handle the undesirable effect of covariates using a mixed-effects random forest-based dissimilarity matrix. While all other similar existing algorithms for clustered data only use random effect and the splitting node at each non-terminal node, we will separately control the impact of confounding phenotypic variables such as age at each splitting node internally. Moreover, this algorithm has the capability of handling high-dimensional data without using any dimension reduction algorithm. In Section 4.2, we will provide an overview of data and imaging modalities. Section 4.3 will focus on providing two binary classification algorithms. Result Section 4.4 will illustrate the use of this algorithm using empirical data from ABIDE.

All existing and proposed algorithms have been implemented in an R package called RFMEClass which will be available to be uploaded to the CRAN [51].

CHAPTER 2
MATCHING METHODS FOR OBSERVATIONAL DATA WITH SMALL GROUP SIZES

2.1 OVERVIEW

In the observational studies, matching is used to optimize balance and sample size. Although many matching algorithms can achieve this goal, in some fields, matching could face its own challenges. Datasets with small sample sizes and limited numbers of control reservoirs are prone to this issue. This problem may apply to many ongoing research fields such as autism spectrum disorder (ASD). In this study, we are interested in eliminating the effect of undesirable variables using two types of algorithms, 1:k nearest matching and full matching. Three different types of 1:3 nearest matching algorithms were implemented and tested to check whether group-wise matching could potentially create an optimal balance and sample size. Moreover, two other full matching based methods were proposed: Iteration optimal matching (IterMatch) and iterative optimal coarsened exact matching (iterMatchCEM). Both methods incorporate an optimal matching algorithm OPTMATCH that was iteratively based on network flow theory [31]. This algorithm has the flexibility of feeding the proximity matrix as the main input. The main input is an RF-based proximity matrix which is then defined and calculated to find a well-matched sample. IterMatchCEM mixed the idea of both optimal matching and coarsened exact matching [35]. These proposed methods were applied to a data set of 47 ASD and 57 typically developing (TD) participants from San Diego State Brain Development Imaging Lab (BDIL). In this sample, there were not enough TD participants to be matched with ASD. Among all methods, an iterative optimal matching algorithm (IterMatch) performed the best in terms of preserving more participants

and creating balance on all variables eliminating ASD participants individually. IterMatchCEM algorithm achieved similar results with a smaller sample size.

2.2 RANDOM FOREST

Random forest (RF) is an ensemble method based on decision (classification or regression) trees. The main principle behind the ensemble approach is to average many weak learners to form a strong learner. To "decorrelate" the trees, each split in RF is chosen based on only a randomly selected subset of covariates while optimizing a target function. The result of recursive partitioning is that the nodes become increasingly pure (in terms of outcome) as data move from the root node to terminal nodes. Individual trees in an RF are typically grown until some loose stopping criteria are met without pruning. Though each tree in the standard random forest [8] is constructed using a bootstrap sample (drawn with replacement from the training data), we use all of the data to build each tree so that propensity score and proximity measure can be calculated for each observation and between any pair of observations respectively, based on each tree.

2.3 CALCULATION OF PROPENSITY SCORE AND PROXIMITY MATRIX BASED ON RF

In our application of the RF, three variables were randomly selected to choose each split. This value remains fixed during forest construction. If a selected covariate is a continuous variable, every possible cutpoint is considered. For each proposed split, a 2×2 table is formed based on the split (left versus right child nodes) and the treatment indicator (ASD versus TD in our application), and the best split is chosen based on the smallest *p*-value from the chi-square test for the 2×2 table. Splitting continues until either the tree reaches its maximum depth of 10, the node size becomes less than 20 participants or minimum number of 7 participants at a terminal node. We have explored various numbers of trees in the random forest and found that the matching results become stable after 1,000 trees. Therefore, all the results reported in this chapter are based on random forest of 1,000 trees.

The propensity score is defined as the conditional probability of treatment given covariates, or

$$P(Z = 1|X) = E(Z|X) \tag{2.1}$$

where Z is the treatment indicator (or the ASD status in our application) and X denotes all covariates excluding the treatment indicator. The propensity score provides a scalar summary of all the covariates: under the assumption of strong ignoreability, the distribution of X given the propensity score is balanced between the treated and control groups [56]. This is the basis for propensity score based matching. Thanks to random forest's outstanding predictive performance and ease of implementation, there is a growing literature on using tree-based methods to calculate the propensity score, see, e.g., [14] and [43]. As described below, both the propensity score and the proximity matrix can be easily calculated once a random forest is constructed.

Given we use all the data for constructing each tree, all participants can be tracked down in a terminal node of the tree. For each tree, all participants in a terminal node are assigned the proportion of the treated participants in that terminal node as their propensity score. Note that the proportion itself, not the majority vote, is saved for each tree for better predictive performance (cite). To calculate the distance between two terminal nodes, form a 2×2 contingency table based on the treatment indicator and terminal node membership, calculate the chi-square *p*-value of the contingency table, and assign the distance of 1 – (chi-square *p*-value) to all the pairs of participants belonging to different terminal nodes. The distance between two participants within the same terminal node is zero. The propensity score and distance from the RF is the average propensity score and average distance over all the tree in the forest. Algorithm 1 presents these steps.

Initialize: set $S_i = 0$ *and* $D_{ij} = 0$ *for i,j = 1,. . . ,n.*

for b *= 1,. . . ,B,* **do**

- Using all data, grow a binary tree with the treated and control labels as the outcome and use all other covariates as input.
- At each split, search over m randomly selected inputs. No pruning.
- Repeat for all terminal nodes of the tree to obtain a proportion $s_i, i = 1, \ldots, n$ for all participant.
- Update: $S_i = S_i + s_i$
- Proximity matrix calculation: For each pair of terminal nodes of the tree, form a 2×2 table based on the treatment indicator and terminal node membership (i.e., which node of the pair a participant is from), and calculate a p-value for the 2×2 table based on the chi-square test.
- Repeat for all pairs of terminal nodes of the tree.
- Assign d_{ij} to be 0 if the ith and j-th participants belong to the same terminal node of the tree, and 1- (p-value) otherwise.
- Update: $D_{ij} = D_{ij} + d_{ij}$ for $i, j = 1, \ldots, n$.
- Average: $S_i = \frac{S_i}{n}$ and $D_{ij} = \frac{D_{ij}}{n}$ for i,j = 1,. . . ,n.

end

Algorithm 1: Algorithm for calculating propensity score and proximity matrix

2.4 GROUP MATCHING

Given that we have only 47 ASD (treated) and 52 TD (control) participants, the goal of this matching project is to keep as many control participants as possible in the matched samples. In addition, we do not need the two groups to be of the same sample size, as long as the distributions of the ASD and TD groups are balanced with respect to the five background covariates. With these goals in mind, the first method we propose is group matching. Since we do not have a "control reservoir", we will keep matched control participants still in the pool to be possibly matched by other participants. In order to find good matches while keeping bad ones out, we use a nonexclusive 1:3 match based on propensity score only, proximity measure only, and proximity measure within calipers defined by the propensity score. These methods are similar to those in D'Agostino (1998)[17]. In order to facilitate the comparison among all the methods proposed in this chapter, the treated group is randomized once and then the order stays the same for all matching methods.

2.4.1 Group matching based on propensity score

For each treated participant, three control participants were selected based on the closest propensity score with the treated participant. That is, the three selected control participants were the ones with the smallest absolute difference in propensity score with the treated participant. The matched control participants were left in the pool as possible matches for other treated participants. This process was repeated for all treated participants. The final matched samples include all the treated participants and only those control participants who were selected at least once.

2.4.2 Group matching based on distance

For each treated participant, three control participants with the smallest proximity measure to the treated participant were selected. The matched control participants were left in the pool as possible matches for other treated participants. This process was repeated for all

treated participants. Subsequently, the matched samples consisted of all treated participants and only those control participants who were selected at least once.

2.4.3 Group matching based on distance within calipers defined by the propensity score

For each treated participant, only those control participants whose logit transformed propensity scores were within a small range, i.e. $\frac{3}{4}$ standard deviation of the logit propensity scores of all participants, were examined further. Within this small set of control participants, the three participants with the smallest distance from the treated participant were selected. The matched control participants were left in the pool as possible matches for other treated participants. This process was repeated for all treated participants. Subsequently, the matched samples consisted of all treated participants and only those control participants who were selected at least once.

2.4.4 Iterative matching without missing value handling

One problem with not having a large control reservoir is that not all treated subjects may be able to find a good match from the limited control pool. To this end, we propose an iterative matching algorithm such that the treated participant that has the poorest match with a control may be excluded from the sample. We will use the traditional 1:1 matching and remove only one treated participant at a time. With limited control participants to choose from, it becomes imperative that, within each iteration, the algorithm has the ability to release matched control participants if they can become better matches for other treated participants. Unlike the nearest matching method where the order of treated participants can change the quality of matching, in the optimal matching [31], the overall set of matches minimize the global distance measure [55]. We implement the optimal matching, borrowing the syntax of OPTMATCH, an add-on package to R while using the proximity matrix as our distance measure.

Iterative matching is a multi-step matching algorithm that aims for matched samples

of desired quality while keeping as many participants in the matched samples as possible. In the first iteration, optimal one-to-one matches are obtained for all the treated participants based on the proximity matrix. Further iterations aim to improve the quality of matched samples, for example Standardized Mean Difference (SMD) of the matched samples. To do so, in each iteration the treated participant with the farthest distance to its matched control is chosen for exclusion. The exclusion process continues until a certain threshold on the quality of matching is reached. Note that in each iteration the proximity matrix is reduced to correspond to all control participants and only those treated participants who have not been excluded due to the SMD criteria. The threshold used for the SMD criteria is that its absolute value is less than 0.10. All steps are provided in Algorithm 2.

Denote the sample sizes for the treated and control groups by n_1 and n_2, respectively. Without the loss of generality, assume $n_1 < n_2$.

Start with all n_1 treated participants.
Building the binary random forest.
Building the proximity matrix.
Find a 1:1 optimal matches from the n_2 controls based on the proximity matrix $D_{ij_{n_1 \times n_2}}$.
if *SMD threshold for all matching vars* $<$ *SMD thresholds* **then**
    Return SMD, *p*-values for all matching variables and matched sample data frame.
else
    while *SMD threshold is not satisfied* **do**
        Retrieve the distance between the matched treated vs. control participants.
        Find the farthest distance and exclude the corresponding treated participant.
        Take subset of distance matrix using all control and remaining treated participants.
        Find 1:1 optimal matches from the n_2 controls based on the current proximity matrix.
        Creates matched data frame in iteration-i using the retrieved distance matrix.
        Check the balance between matched samples.
    end
end

Algorithm 2: Iterative matching algorithm without missing data handling

2.4.5 Iterative matching incorporating coarsened exact matching

While exact matching provides a perfect balance between two samples, it typically produces few matches because of curse-of-dimensionality issues [7]. The idea of coarsened exact matching (CEM) is to temporarily coarsen each variable into substantively meaningful groups, exact match on these coarsened data, and then retain only the original (uncoarsened) values of the matched data. Though this method is fast, easy to understand, requires fewer assumptions, and possesses more attractive statistical properties for many applications than the existing matching methods [7], "meaningful groups" is subjective and could alter the resulting matched samples. In particular, "binning" in CEM requires user-defined cut points, with the shortcoming that neighboring values of a covariate may end up being grouped into different bins. On the other hand, a distance based threshold can avoid such undesirable scenarios and hence lead to more valid groups. Since the proximity matrix is defined for each pair of treated and control participants, it is possible to determine whether any pair of participants are close based on their values for a given covariate directly.

To begin, a distance threshold needs to be set for each covariate so that participants with absolute difference below the threshold are considered having the coarsened exact value for the covariate. Two participants are considered coarsened exact matches of each other if they have coarsened exact values for all covariates. This is used to modify the proximity matrix from Section 2.3 such that the proximity between a pair of treated and control participants would remain the same if they are coarsened exact matches, otherwise it would be set to a large number (for example, the value of 1 is sufficiently large). Finally, the iterative matching described in Section 2.4.4 is performed based on the modified proximity matrix to obtain matched samples. These steps are provided in Algorithm 3.

For our application, we set distance thresholds to be 4 years for age, 0.05 mm for motion, and 20 units for NVIQ. This part may be automated by using thresholds that are multiples of standard deviations for each covariate, for example, half or one standard

deviation. Note that how generous the thresholds can be depends on the sample sizes (especially the size of the control sample) as well as the desired sample sizes of the matched samples. Very large thresholds will return the same results as from iterative matching Section 2.4.4, while small thresholds may result in many unmatched treated participants.

1 *Defining being chosen:*

Define what constitutes "being close" for each covariate. We use the absolute difference and set a threshold for each covariate.

Modifying proximity matrix:

if *If all covariate values for two participants are "close",* **then**

    keep its distance;

else

    set the distance to a large number (for example, the value of 1)

end

Applying iterative matching:

Perform iterative matching based on the modified proximity matrix.

Algorithm 3: Algorithm for iterative matching incorporating coarsened exact matching

2.5 SIMULATION EXAMPLE

2.5.1 Data

The sample consisted of an SDSU Brain Development Imaging Lab in-house dataset with 98 children and adolescents with ASD and 80 TD peers. Many participants were excluded from the study due to a variety of reasons, including excessive head motion, low Autism Diagnostic Observation Schedule (ADOS) score for ASD participants, misdiagnosis, known genetic (e.g., Fragile-X or Rett syndrome) and neurological (e.g., epilepsy) conditions associated with ASD, or a personal or family history of ASD for TD participants. These exclusion criteria resulted in a final sample of 47 ASD and 52 TD participants. Informed consent was obtained from all participants and their caregivers in accordance with the University of California, San Diego, and San Diego State University Institutional Review Boards.

Since the purpose of matching is to obtain balanced ASD and TD groups for the study of functional connectivity patterns in search for autism brain markers, the treatment variable in our matching study is the ASD indicator defined as having autism spectrum disorder (1) and typically developing (0). Available background covariates include three continuous variables: motion, age, and non-verbal IQ (NVIQ); as well as two categorical variables: gender and handedness (left- or right-handed). As can be seen from Table 2.1, there are 9 girls among 47 ASD participants and 10 girls among 52 TD participants. The age of the combined ASD and TD groups ranges from 8 to 18 years with a mean of 13. For the retained samples after inclusion criteria are applied, the range of motion is from 0.02 mm to 0.17 mm. The non-verbal IQ ranges from about 62 to 140, with the ASD group having a slightly lower average NVIQ with a mean of 102.4, compared to a mean of 107.3 for the TD group. There are 9 and 7 left-handed participants in the ASD and TD groups, respectively.

Table 2.1. Participants characteristics and summary statistics before matching.

Full sample	ASD, M $\pm SD$ [range]	TD, M $\pm SD$ [range]	*p*-value (2-sample t-test/Chi-square)	Standardized Mean Difference (SMD)*
N (Female)	47 (9)	52 (10)	1	$<$0.01
Age (years)	13.82 $\pm$2.46 [9.2-18.0]	13.44 $\pm$2.72 [8.1-17.6]	0.47	0.14
Motion (mm)	0.07 $\pm$0.03 [0.02-0.15]	0.63 $\pm$0.35 [0.02-0.17]	0.36	0.18
Non-Verbal IQ	102.36 $\pm$17.46 [67-140]	107.34 $\pm$13.07 [62-137]	0.11	0.32
Handedness (Left)	47 (9)	52 (7)	0.62	0.15

*The standardized difference is given by $\mid \bar{X}_T - \bar{X}_C \mid / \sqrt{(S_T^2 + S_C^2)/2}$, where and denote sample mean for treated and control and sample variance respectively.

One of the key steps in evaluating matching methods is to assess the balance of the resulting matched samples, or similarity of the empirical distributions for all covariates in the matched treated and control groups. The most common measures used in the literature are two-sample t-test *p*-values for continuous variables and chi-square *p*-values for categorical variables. However, Imai et al. [36] show that *p*-values that incorporate information on the sample size should not be used as a measure of balance. In this dissertation, standardized mean difference (SMD), defined as Standardized Mean Difference

$$(SMD) = \frac{\mid \bar{X}_T - \bar{X}_C \mid}{\sqrt{\frac{S_T^2 + S_C^2}{2}}} \tag{2.2}$$

, where X and s^2 denote sample mean and sample variance respectively, is used as the main yardstick to gauge matching quality, though *p*-values are also presented alongside SMD's. Note that a large *p*-value and a small SMD indicate better matching results. Since *p*-values are sample size dependent, we will use an SMD of below 0.10 as a rough guideline for good matching performance [36]. In Table 2.1, participants' characteristics and summary statistics before matching is presented. SMD values for all matching variables except for sex are above 0.1.

2.5.2 Results

2.5.3 Group matching results

Group matching algorithms were applied to SDSU Brain Development Imaging Lab in-house dataset with 47 ASD and 52 TD participants. A random forest of 1000 trees was built to estimate the propensity score and proximity matrix. Matching results from 1:3 matching based on all three methods, namely group matching using propensity score alone, distance alone, and distance within calipers defined by the propensity score, are summarized in Table 2.2. Though the sample sizes of the matched samples from the three methods are not the same, the performance of the matching method based on propensity score alone is relatively poor compared to the other two methods. Recall that head motion is considered the most important variable in this matching project and the matching performance based on propensity score alone is the worst for head motion.

The matching methods based on distance alone and on distance within calipers defined by the propensity score lead to matched samples with similar sample sizes for the ASD and TD groups, with the method based on distance alone producing slightly better overall matching results. Though both of these algorithms performed OK with respect to head motion, age, handedness and gender, the SMD's for NVIQ are much larger than 0.10. Although all ASD participants were retained by these methods, close to 20 TD participants were discarded by each method.

In summary, the three group matching methods neither can provide a balance between the number of ASD and TD participants, nor does retain enough TD participants. This is not desirable since the power of the subsequent study of brain imaging data will suffer as a result. Moreover, matching results for NVIQ are poor for all three group matching methods.

2.5.4 1:1 Iterative matching results

Based on relative strong performance of the matching method using distance alone (see Table 2.2) as well as its easy implementation, the two iterative algorithms are based on

Table 2.2. Group comparisons after matching using Summary statistics after matching using 1:3 nearest matching based on propensity score, distance and distance within calipers defined by propensity score.

Covariates	Propensity score		Distance		Distance within calipers of propensity score	
Balance Measures	*p*-values	Std diff	*p*-values	Std diff	*p*-values	Std diff
Gender	1	0.04	0.82	0.12	1	0.05
Age	0.49	0.14	0.85	0.04	0.86	0.04
Motion	0.39	0.18	0.69	0.09	0.61	0.12
Non-Verbal IQ	0.14	0.31	0.26	0.25	0.22	0.28
Handedness	0.6	0.16	0.82	0.12	0.87	0.1
	ASD	TD	ASD	TD	ASD	TD
N (female)	47 (9)	46 (8)	47 (8)	34 (9)	47 (9)	33 (7)

proximity matrices only.

Table 2.3 presents summaries and matching results from the iterative matching. Since absolute standardized mean differences (SMD's) of 0.10 or below was used as the algorithm stopping criteria, we expect that the SMD's for all five background covariates would be below 10 percent. Table 2.3 shows that the iterative routine is able to preserve high number of participants, 43 ASD and 43 TD participants in the matched samples, while meeting the desired level of matching in SMD. These are successful matching results both in terms of retained sample sizes and matching quality as measured by SMD.

Note in Table 2.3 that, though the covariate distributions in the matched samples are balanced as judged by high *p*-values and small SMDs, the matched samples are not matched exactly, as obvious for gender and handedness. In the matched samples, there are 8 female participants among 43 ASD participants as compared to 7 female participants among 43 TD participants. Similarly, there are 8 (7) left-handed participants among 43 ASD (TD) participants.

Table 2.4 presents summaries and matching results from the iterative matching method incorporating coarsened exact matching. Again absolute standardized mean differences (SMD's) of 0.10 or below was used as the stopping criteria. The algorithm stopped after seven iterations resulting in 40 ASD participants and 40 TD participants in the matched samples.

Table 2.3. Summary statistics after matching using iterative matching.

Full sample	ASD, M $\pm SD$ [range]	TD, M $\pm SD$ [range]	p-value (2-sample t-test/Chi-square)	Standardized Mean Difference (SMD)
N (Female)	43 (8)	43 (7)	1	0.06
Age (years)	13.64 ± 2.43 [9.2-17.8]	13.81 ± 2.53 [8.6-17.6]	0.75	0.07
Motion (mm)	0.07 ± 0.04 [0.017-0.15]	0.07 ± 0.04 [0.02-0.168]	0.85	0.04
Non-Verbal IQ	104.81 ± 16.06 [67-140]	106.09 ± 13.67 [62-137]	0.69	0.09
Handedness (Left)	43 (8)	43 (7)	1	0.06

Table 2.4. Summary statistics after matching using Iterative matching incorporating Coarsened exact matching.

Full sample	ASD, M $\pm SD$ [range]	TD, M $\pm SD$ [range]	p-value (2-sample t-test/Chi-square)	Standardized Mean Difference (SMD)
Gender (Female)	40 (6)	40 (6)	1	0
Age (years)	13.83 ± 2.44 [9.2-17.8]	13.97 ± 2.46 [8.1-17.6]	0.81	0.05
Motion (mm)	0.07 ± 0.03 [0.02-0.15]	0.07 ± 0.04 [0.02-0.17]	0.89	0.03
Non-Verbal IQ (NVIQ)	105.18 ± 15.06 [70-140]	106.43 ± 13.41 [62-129]	0.7	0.09
Handedness (Left)	40 (5)	40 (5)	1	0

This algorithm outperformed all the other ones in terms of both p-value and SMD. However, the total number of participants in the final matched samples was reduced to a total of 80 participants, or 40 participants in each group. Note that the ASD and TD participants in Table Table 2.4 are exact matches in a coarsened sense, by design.

The quality of matching may also be assessed by the distributions of propensity scores in the two groups, given by propensity score is a scalar summary of all the covariates. To this end, the logit transformed propensity scores of the ASD and TD groups are plotted in Figure 2.1 before matching, Figure 2.2 after iterative matching, and Figure 2.3 for after iterative coarsened exact matching. As can be seen from these figures, there is much more overlap between the propensity score distributions for the two groups after the two iterative matching algorithms. Note that the matching quality here may not be compared to other matching

studies since we have limited sample size and the goal is to retain as many participants as possible for the subsequent study of brain imaging data.

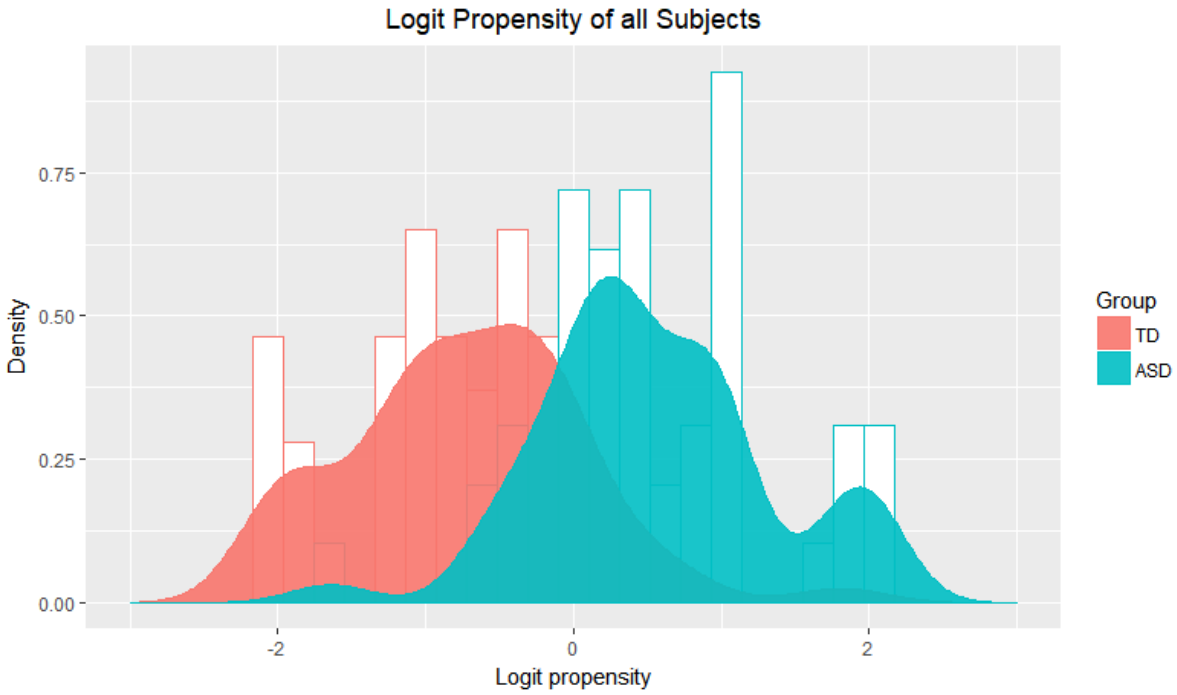


Figure 2.1. Logit propensity score of all participants before matching.

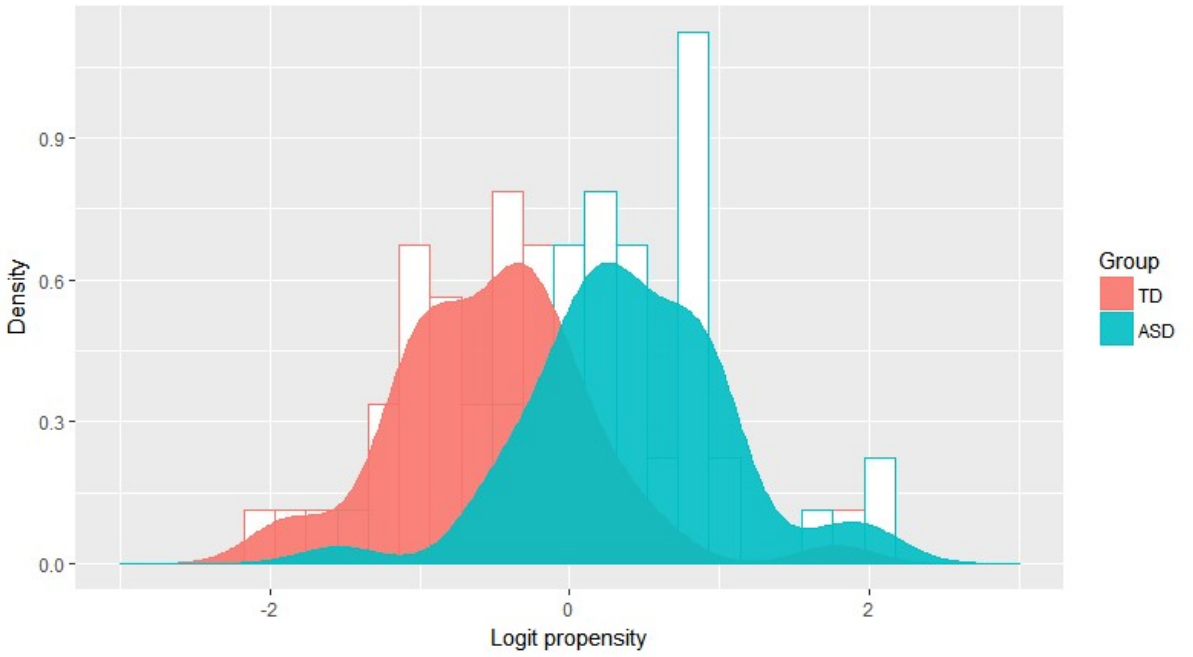


Figure 2.2. Logit propensity score of all participants after using *IterMatch* matching algorithm.

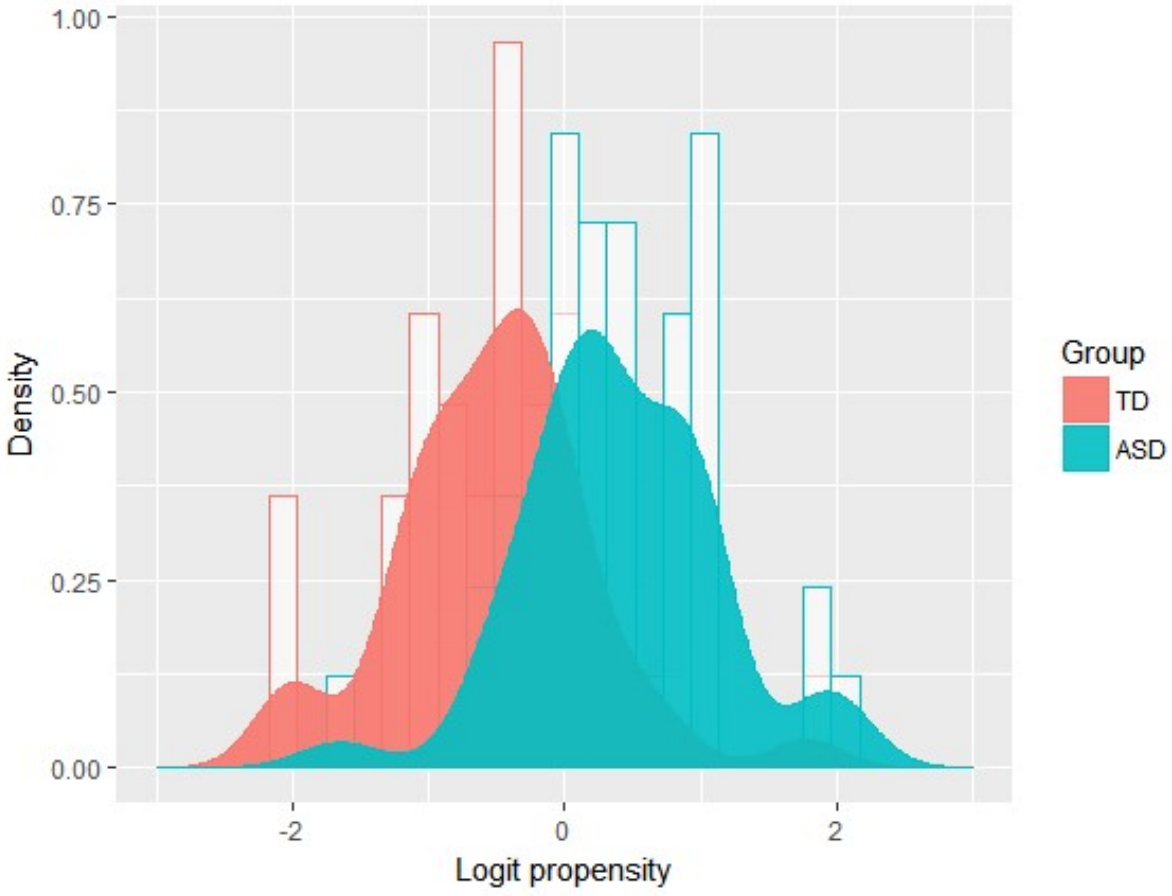


Figure 2.3. Logit propensity score of all participants after using *IterMatchCEM* matching algorithm.

CHAPTER 3

ITERATIVE MULTIVARIATE MATCHING PACKAGE FOR SAMPLE WITH MISSING DATA: THE ITERMATCH PACKAGE FOR R

3.1 OVERVIEW

Matching two groups on multiple confounding variables is one of the primary steps in conducting observational studies. All existing matching algorithms can function only with a complete dataset, while none can function with missing data. Datasets with these problems must impute or only complete cases can be considered in matching.

Losing data because of the limitations in a matching algorithm can decrease the power of the study as well as omit important information. In this chapter, we introduce the R package iterMatch that tackles these shortcomings. This package finds a one-to-one subsample of the data that is balanced on all matching variables while incorporating missing values in an iterative manner. Random forest is used as a crucial tool to handle missing values when constructing a distance matrix to be fitted to an optimal matching algorithm proposed by [31]. We measure the robustness of the matching results by injecting levels of missing values across two medium and large datasets for comparison. More detail is provided in this chapter.

3.2 RANDOM FOREST

This section, describes the Random Forest as the main building blocks of the iterMatch package. Although general framework and parameter tuning of the Random forest (RF) was described in Section 2.2, additional features were added to the package version of the *iterMatch* algorithm. These features included handling missing values and handling categorical confounding variables with more than two categories.

Missing data handling is described in more details in Section 3.5. Hence, only a summary of features and the RF parameters are provided as follow.

A binary classification RF model was created using the following steps. First, all participants were sent down the root node to form two child nodes. At each child node, a randomly selected subset of covariates were selected to decorelate the trees. In our application, we selected three covariates. A greedy search based cutpoints selection was performed on all selected covariates. The Gini index was utilized to select the best covariate and its corresponding cutpoint at each node. Based on the mentioned criteria, participants were splitted into further child nodes until some loose stopping criteria was met. For example, child nodes splitting stopped if a child node had less than 20 participants or tree size exceeded the length of 10 or less than five participants remained in a terminal node.

3.3 Iterative Matching Algorithm with Missing Data Handling

To handle missing data, we propose an iterative matching algorithm that can handle missing values such that participants with partial missing values will be used in creating a matched sample. We used the traditional 1:1 matching and remove only one treated participant at a time. Within each iteration, the algorithm has the ability to release matched control participants if they can become a better match for other treated participants. Unlike the nearest matching method where the order of treated participants can change the quality of matching, in the optimal matching [31], the overall set of matches minimizes the global distance measure [54]. We borrowed the syntax of OPTMATCH, an add-on package to R. We used the proximity matrix as our distance measure in OPTMATCH to create optimal matched samples.

Iterative matching is a multi-step matching algorithm that aims for matched samples of a desired quality while keeping as many participants in the matched samples as possible. In this chapter, *data-1* and *data-2* have more control participants than treated participants. Hence a subset of control participants has to be selected in an iterative manner to reach the desired

matching quality by excluding one treated participant at a time. In the iteration-0, OPTMATCH creates a one-to-one matched sample for all the treated participants based on the proximity matrix. Further iterations aim to improve the quality of matched samples or the SMD values of the matched samples. However, with the presence of missing values, the SMD calculation changes the retained matched sample. Therefore, at each iteration, different SMD calculation methods can be used which were discussed in more details in section 3.4.

The following framework is the general framework of iterative matching when there are missing values involved.

Inputs: *data*, *formula*, *nTree*, *distance*, *minsplit*, *rfmtry*, *methodSMD*, thresholds

Outputs: A list contains of SMD values and *p*-values for all matching covariates and final matched sample

Data: A data frame consists of a Binary response variable and participant ID and matching variables

Data preparation: Check data prerequisites for the matching with n_1 and n_2 sample from treated and control group

Building the random forest implemented with surrogate splits

Building the proximity matrix

Find a 1:1 optimal matched sample from the n_2 controls based on the proximity matrix in iteration-0

if *SMD threshold for all matching vars* $<$ *SMD threshold* **then**

    Return SMD, *p*-values for all Matching Variables and Matched Sample Data Frame

else

    while *SMD threshold is not satisfied* **do**

        Retrieve the distance between the matched treated vs. control participants

        Create a data frame paired participant ID and the corresponding distance

        Find the max distance bet

        Exclude the farthest participant among the participants with smaller group size

        Retrieve the distance matrix with remaining smaller participant group and all participants from larger group

        Creates matched data frame in iteration-i using the retrieved distance matrix

        Result: Calculate SMD excluding only participant from smaller group which had farthest distance

    end

end

Algorithm 4: The *IterMatch* algorithm with missing data handling

3.4 METHODS OF CALCULATING SMD

Calculating SMD at the presence of missing values could affect matched sub-samples. Multiple factors could play a role in creating a matched sample with the most retained participants in the presence of missing data. An obvious factor is the method of calculating SMD for a matched sample.

For iterMatch R package purposes, only two options are available for calculating SMD values, *method-1* and *method-2*. These methods are explained in Section 3.4.1 and Section 3.4.2. However, for empirical purposes and to better compare results from these two methods, another method of calculating SMD is introduced which we refer to as ”Gold Standard”.

In calculating SMD values using the *Gold standard* method, we started with a dataset that had no missing values, then a certain amount of ”NA” values were randomly inserted into the dataset. After creating a data frame with artificial missing values, *IterMatch* algorithm was run using the created data frame. Then using the *Gold standard* method, original values were retrieved from the original dataset before assigning NA values in calculating SMD. We expected that results from this analysis would match results from the analysis of data before injecting missing values.

3.4.1 Method-1

In this method, for SMD calculation, we considered all non-missing values and omitted cases that had ”NA” values. Therefore, participants with a missing variable were omitted in the calculation of standardized mean difference. This method is referred to as *method-1* in the iterMatch package. It is obvious that the number of non-missing values for variables that contain missing values in each group is different. The unbalanced number of participants for variables with missing values could potentially create bias in calculating SMD. This problem is addressed in the following section with *method-2*.

3.4.2 Method-2

In *method-1*, the SMD is calculated based on an unequal number of non-missing variable values from treated and control groups. To address this problem, we consider only treated or control matched participants for whom all variables are available. This means that, if a treated participant's variable value is non-missing but its corresponding control pair variable value is missing, both the treated and control participant will be discarded for SMD calculation. Meaning that in each iteration, only pairs of participants whose treated participants are selected for matching and have no missing values, will be considered for SMD calculation. In other words, if one of the pairs have any missing values, both pairs will be discarded for the SMD calculation. This exclusion criteria of this participant will help SMD to be unbiased. Note that this exclusion will affect the SMD calculation, not the current matching participants in that iteration.

3.5 Dealing with Missing Values

Dealing with missing data is always challenging. This challenge becomes more important if a large portion of the sample is crucial to the study. In matching observational studies, besides creating a well-balanced sample, one would like to maximize the number of participants. In this section, we investigate the effect of missing values in medium (*data-1*) and large (*data-2*) samples.

Most of the literature on matching and propensity scores assumes fully observed covariates so that models such as logistic regression can be utilized to estimate the propensity scores. However, there are often missing values in the covariates, which complicates matching and propensity score estimation. A key consideration when thinking about missing covariate values is that the pattern of missing covariates can be prognostically important, and in such cases, the methods should be based on the observed values of the covariates and on the observed missing-data indicators. In 2004, Hill [34] finds that methods using multiple imputation work better than complete-data or complete-variable methods (which use either only units with complete data, or variables with complete data).

3.5.1 Surrogate Split

Surrogate splits [9] are a technique where we find good replacements to the primary split at any given node in a classification/regression tree; in this case we use it over multiple trees in a RF. This technique is advantageous because when an observation has a missing value on the primary split, we can delineate to a list of surrogate splits to act as a replacement rule wherever missing data is present.

When specified in iterMatch, surrogate splits are evaluated at each node based on the agreement with the primary split during the RF construction. To be able to compare the primary split to a given surrogate split, we define a rule that ensures we are comparing nodes correctly. Therefore, we use the CART standard of computing the proportion of treated in each child node to the parent node and then placing the split sample with the least treated proportion in the left node. Conversely, the right node receives the split data with the highest proportion of treated. In this way, agreement is calculated by counting up the proportions of treated in the children nodes of the primary split, and comparing these proportions to a candidate surrogate split that also enforces the aforementioned rule. We effectively find how close the surrogate split mimics the primary split. We then maximize this agreement to find the best split for a particular surrogate variable.

In evaluating the effectiveness of the surrogate split, we compute the adjusted correlation to a simple majority rule [9] in handling missing data. If this value is greater than zero, we are better than majority rule and therefore keep the candidate surrogate split.

Depending on the number of *mtry* in constructing the RF, we create a list of up to five surrogate splits, by default. When applied to RF, the literature shows we can maintain similar results with increasing levels of missingness and over different methods of imputation [53] and [32]. In our analysis, surrogate splits come at the cost of an increase in computation time, but we gain a lot more being able to naturally match while handling missing data.

3.6 EMPIRICAL EXAMPLE

3.6.1 Data

The data considered for this analysis is selected from Autism Imaging Data Exchange-I (ABIDE-I)[23] and Autism Imaging Data Exchange-II (ABIDE-II) [22] and SDSU's proprietary data. This dataset is publicly available at `http://fcon_1000.projects.nitrc.org/indi/abide/`. The dataset combining two versions of the Autism Brain Imaging Exchange include 1276 total participants from ABIDE I and 996 participants from ABIDE II from 20 international research institutes with over 70 variables. Since ABIDE-I and ABIDE-II contain SDSU data, we excluded them from these two datsets and include them with more participants later. Therefore, these statistics exclude participants from SDSU's site. A full cohort of proprietary data consists of 241 participants added to the pool of data. For the purpose of this chapter, we choose a subset of data consist of only 5 variables of head motion, age, Performance IQ (PIQ)/Full Scale IQ (FIQ), handedness scores, and sex. Two different sets of criteria are utilized to examine the effects of sample size with the proposed algorithm. The first sample has stricter criteria to ensure high quality data for producing more accurate neuroimaging results. Criteria for the first sample are: head motion during scanning (RMSD) less than 0.20 mm, percent good time points greater than 0.80, and the age is between 6 to 18 years and open eyes while scanning. These criteria were applied to only sites that have at least 20 ASD and 20 TD from both versions of ABIDE. For example, participants that are labeled as ABIDEII-NYU-2 are discarded due to small sample of ASD (11) and TD (0).

Our second dataset uses a less conservative criteria which aims to increase the sample size. The criteria for the second data are: head motion (RMSD) less than 0.27 mm, percent good time points greater than 0.65 time points, and the age is between 5 to 65 years. Theses two datasets are available in the package as toy data named *data-1* and *data-2*.

After applying exclusion criteria, the remaining participants for the first dataset

consisted of 327 ASD and 437 TD participants and in the second dataset 463 ASD and 554 TD. For these two datasets, the number of ASD is greater than TD participants. Therefore, the goal of our 1:1 matching algorithm is to subset of TD participants whose standardized mean difference for all 5 matching variables is below a specific threshold compared to ASD participants. Note that both data sets have missing values in PIQ and handedness scores. The summary statistics for the following variables is shown in Table 3.1 Table 3.2. Variables used for *data-1* and *data-2* are defined as follow:

- SITE-ID. Site abbreviation
- SUB-ID. ABIDE unique ID number
- RMSD. Root Square Mean Sliding Difference or head motion in the scanner (mm)
- GOOD-TP. Number of good time points acquired from scanning
- PERCENT-GOODTP. Percent of good time points acquired from scanning
- DX-GROUP. Diagnostic Group, 1=Autism, 0=Control
- AGE-AT-SCAN. Age at time of scan (years)
- SEX. Gender, 1 = male; 2 = female
- HANDEDNESS-SCORES. Handedness Scores, right-handed: scores ≥ 50, left-handed:scores ≤ -50, mixed handed: scores between -50 and 50
- PIQ. Performance IQ Standard Score (**data-1**)- FIQ. Full IQ Standard Score (*data-2*)
- EYE-STATUS-AT-SCAN. Eye Status During Rest Scan, 1 = open, 2 = closed

Table 3.1. Participants characteristics and summary statistics before matching for *data-1*.

data-1	ASD, M±SD [range]-[missing]	TD, M±SD [range]-[missing]	*p*-value (2-sample t-test/chi-square)	Standardized Mean Difference*(SMD)
N (Female)	327 (87)	437 (168)	0.00	0.25
Motion (mm)	0.08± 0.17 [0.01-0.26]-[327]	0.07± 0.14 [0.01-0.22]-[437]	0.01	0.19
Age (years)	11.32 ± 2.53 [6.16 - 17]-[327]	11.08 ± 2.19 [6.36 - 17]-[437]	0.16	0.10
Performance IQ (PIQ)	104.94 ± 15.14 [53 - 149]-[64]	110.48 ± 13.57 [62 - 147]-[29]	0.00	0.35
Handedness Scores	55.36 ±50.41 [-100 - 100]-[154]	70.33 ±40.02 [-100 - 100]-[156]	0.00	0.32

*The standardized difference is given by $|\bar{X}_{ASD} - \bar{X}_{TD}|/\sqrt{(S^2_{ASD} + S^2_{TD})/2}$, where and denote sample mean for treated and control and sample variance respectively.

Table 3.2. Participants characteristics and summary statistics before matching for *data-2*.

data-2	ASD, M±SD [range]-[missing]	TD, M±SD [range]-[missing]	*p*-value (2-sample t-test/chi-square)	Standardized Mean Difference*(SMD)
N (Female)	463 (120)	554 (203)	0.00	0.23
Motion (mm)	0.08 ± 0.17 [0.00 - 0.26]-[463]	0.08 ± 0.17 [0.00 - 0.26]-[554]	0.03	0.13
Age (years)	13.02 ± 7.30 [5.12 - 62]-[463]	12.71 ± 6.08 [5.88 - 64]-[554]	0.47	0.04
Full Scale IQ (FIQ)	105.26 ± 17.46 [59 - 149]-[25]	113.61 ± 12.73 [79 - 149]-[12]	0.00	0.55
Handedness Scores	57.58 ± 47.9 [-100 - 100]-[224]	70.59 ± 38.14 [-100- 100]-[203]	0.00	0.30

*The standardized difference is given by $|\bar{X}_{ASD} - \bar{X}_{TD}|/\sqrt{(S^2_{ASD} + S^2_{TD})/2}$, where and denote sample mean for treated and control and sample variance respectively.

The following R command is an example of input parameters for iterMatch function,

```
R> form <-DX_GROUP ~ RMSD + AGE_AT_SCAN + PIQ + SEX + HANDEDNESS_SCORES
R> iterMatch (data = data-1, formula = form, nTree,distance = "p-value",
+             ID = "SUB_ID", thresh = c(0.15, 0.15, 0.15, 0.15, 0.51),
+             methodSMD , surrogates, rfmtry, match.tol = 0.001, iseed = 1,
+              outputfile = "output")
```

iterMatch parameters are defined as follow:

- **data** Dataframe of participants and variables
- **formula** Formula which defines the response and matched variables. Matching variables have to be separated by ”+”, and response variable is separated by” ”.
- **nTree** An integer number of trees in the random forest.
- **distance** Calculating distance between two participants i and j. Possible values is ”0-1” where distance of participants in the same terminal nodes are set to be zero, one, otherwise.
- **ID** Unique participant ID.
- **thresh** A vector of real values defining in SMD threshold for matching variables in the order that they appear in formula.
- **methodSMD** Two methods are proposed to calculate SMD, ”method-1” and ”method-2”.
- **surrogates** Creates surrogate splits if set to *T*, otherwise the default is *F*. FALSE option, assign observations with missing values randomly to a child node.
- **rfmtry** Number of random variables to split at each node.
- **match.tol** Specifies the extent to which fullmatch’s output is permitted to differ from an optimal solution to the original problem. This parameter is taken from fullmatch.
- **outputfile** An output file containing results from all iteration and their balance measures.

3.7 RESULTS

3.7.1 Effect of Number of Trees on Matched Sample

In this section, we illustrate the ability of the iterMatch package to optimally pick a matched sub-sample of treated versus control participants. For this purpose, some of the input variables were picked to illustrate the effects on the final matched sub-sample. For example, Table 3.3 indicates the balance measures for all five matching variables when the number of trees was changing between 100, 250, 500, 750, and 1000 trees while other input parameters were fixed. These parameters included *methodSMD* with ”*method-1*” as an arbitrary option, *thresh = 0.1* for all five matching variables, *surrogates = T*, and *distance* equal to ”*p*-value”.

Table 3.3. Effect of number of trees in *data-1*.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
Number of trees = 100						
SMD	0.07	0.07	0.01	0.04	0.08	291
p-value	0.41	0.39	0.89	0.62	0.44	291
Number of trees = 250						
SMD	0.09	0.09	0.00	0.06	0.09	295
p-value	0.28	0.25	0.90	0.48	0.39	295
Number of trees = 500						
SMD	0.09	0.05	0.02	0.07	0.09	297
p-value	0.32	0.51	0.73	0.43	0.40	297
Number of trees = 750						
SMD	0.03	0.01	0.01	0.05	0.09	263
p-value	0.77	0.84	0.83	0.57	0.42	263
Number of trees = 1000						
SMD	0.08	0.09	0.05	0.09	0.06	284
p-value	0.36	0.24	0.56	0.33	0.60	284

Even though, the number of trees in random forest is determined based on the stability of classification, in our application, to retain more participants from the matched sample, it is recommended to change number of trees to ensure retaining more participants on top of results stability. For example, results become stable for *data-1* after creating 250 trees. But if a user chooses 1000 trees, 284 participants can be retained while a matched sample can retain 295 participants per group when using 250 trees. As we see in Table 3.4 results for *data-2* become stable after 250 trees as well. Therefore, for the rest of this chapter, we used 250 trees for both *data-1* and *data-2*.

3.7.2 Effect of SMD Methods on Matched Sample

When a dataset contains missing values, depending on the amount of missing, we may need to use different SMD methods to ensure the desired balance and maximum number of retained participants per group. Table 3.5 and Table 3.6 demonstrates the effects of applying different SMD methods using *data-1* and *data-2*. All analysis used 0.15 as the SMD threshold for all variables with 250 trees, and surrogates set to *T*.

Table 3.4. Effect of number of trees in *data-2*.

Experiment Balance measure	Sex	RMSD	Age (Years)	FIQ	Handedness Scores	N (Treated) N (Control)
Number of trees = 100						
SMD	0.07	0.01	0.03	0.09	0.01	365
P-value	0.33	0.79	0.60	0.19	0.87	365
Number of trees = 250						
SMD	0.09	0.02	0.01	0.05	0.07	362
P-value	0.22	0.77	0.80	0.43	0.47	362
Number of trees = 500						
SMD	0.09	0.05	0.08	0.09	0.01	307
P-value	0.28	0.65	0.53	0.48	0.92	307
Number of trees = 7500						
SMD	0.09	0.01	0.05	0.02	0.06	348
P-value	0.24	0.79	0.49	0.74	0.55	348
Number of trees = 1000						
SMD	0.09	0.05	0.09	0.03	0.02	323
P-value	0.26	0.50	0.24	0.67	0.84	323

For the analysis of these tables, we started from the complete cases of *data-1*/*data-2* (n=343/n=585). Then similar to missingness patterns in *data-1* and *data-2*, missing values were injected to the complete data. In other words, we randomly injected 0.15/0.003, 0.37/0.39, and 0.03/0.03 NA's to PIQ/FIQ and handedness, and both variables to complete cases of *data-1*/*data-2*. Then, a matched sample based on data with artificially created missing values was created. That is, data set of 343/585 participants with artificially generated missing values was fed into iterMatch algorithm.

Results in both Table 3.5 and Table 3.6 represent SMD values before matching and after matching using different SMD methods for both *data-1* and *data-2* at iteration-0 and final iteration. In iteration-0, after feeding the proximity matrix to OPTMATCH, a 1:1 matched sub-sample of data was selected that had an equal number of treated and control participants. For the comparison purposes of SMD methods, three SMD methods were calculated: *method-1* , *method-2*, and *Gold Standard*.

For *Gold Standard* method, SMD was calculated for data with no missing values. In other words, we re-injected the actual values for the artificially missing values. As explained

in Section 3.4.1, *Method-1* is calculated based on all the values regardless of having missing values or not. Whereas for *method-2*, SMD is calculated based on only the pairs of treated and control participants were used when both participants had non-missing values for the variable under evaluation. In other words, for *method-2*, the actual number of pairs used were different from variable to variable.

The first row of results from Table 3.5 shows the status of SMD and *p*-values of the explained data above with simulated artificial missing variables before matching. The second, third, and fourth rows indicate the performance of *Gold Standard*, *method-1* and *method-2* in iteration-0, respectively. In this iteration, comparing SMD values from *method-1* and *method-2* versus *Gold Standard*, similar performance was observed. However, an obvious superiority of *method-1* was observed in terms of retaining maximum number of participants given 0.15 SMD value threshold in the last iteration. Therefore, for the next two sections, we fixed the *methodSMD* to *method-1* for *data-1*.

This comparison however, is different for *data-2* with a larger sample and fewer missing values. Meaning that in iteration-0 and the final iteration, the number of reattained participants and SMD results are very similar for both *method-1* and *method-2* with slight superiority of SMD values for *method-2* for all matching covariates. Therefore, *method-2* is selected as the preferred SMD method for *data-2* for the next two sections.

Table 3.5. Effect of SMD methods on *data-1*.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
Before matching						
SMD	0.41	0.00	0.21	0.41	0.56	129
p-value	0.00	0.38	0.07	0.00	0.00	214
Gold standard, iteration-0						
SMD	0.28	0.21	0.24	0.23	0.30	129
p-value	0.04	0.10	0.05	0.06	0.01	129
method-1, iteration-0						
SMD	0.28	0.21	0.24	0.26	0.31	129
p-value	0.04	0.10	0.05	0.06	0.05	129
method-2, iteration-0						
SMD	0.28	0.23	0.04	0.20	0.38	129
p-value	0.04	0.32	0.85	0.39	0.10	129
Gold Standard, Last-iteration						
SMD	0.15	0.11	0.09	0.10	0.13	106
p-value	0.37	0.44	0.49	0.48	0.34	106
method-1, Last-iteration						
SMD	0.15	0.11	0.08	0.09	0.03	106
p-value	0.37	0.44	0.56	0.54	0.88	106
method-2, Last-iteration						
SMD	0.00	0.28	0.07	0.12	0.21	4
p-value	1.00	0.80	0.70	0.91	0.85	4

Table 3.6. Effect of SMD methods on *data-2*.

Experiment Balance measure	Sex	RMSD	Age (Years)	FIQ	Handedness Scores	N (Treated) N (Control)
Before matching						
SMD	0.38	0.29	0.06	0.63	0.29	234
p-value	< 0	0.00	0.42	< 0	0.01	351
Gold Standard, iteration-0						
SMD	0.31	0.15	0.15	0.49	0.23	234
p-value	0.00	0.09	0.10	< 0	0.01	234
method-1, iteration-0						
SMD	0.31	0.15	0.15	0.47	0.18	234
p-value	0.00	0.09	0.10	< 0	0.14	234
method-2, iteration-0						
SMD	0.31	0.06	0.15	0.40	0.11	234
p-value	0.00	0.66	0.27	0.00	0.42	234
Gold Standard, Last-iteration						
SMD	0.09	0.07	0.08	0.01	0.13	114
p-value	0.61	0.60	0.54	0.96	0.34	114
method-1, Last-iteration						
SMD	0.14	0.04	0.03	0.07	0.11	159
p-value	0.27	0.72	0.77	0.52	0.46	159
method-2, Last-iteration						
SMD	0.14	0.01	0.09	0.14	0.04	159
p-value	0.27	0.96	0.58	0.37	0.80	159

3.7.3 Effect of Missing Values

To evaluate the performance of surrogates argument in the iterMatch package, We fixed all input variables and changed the surrogates to *T* and *F*, respectively for *data-1* and *data-2*. As it is illustrated in Table 3.7, given the SMD threshold of 0.15, twelve more participants were retained total using surrogates as *T* versus *F* option. Note that the percent missing values in *data-1* was 55% and setting surrogates = *T* shows better performance of RF surrogate versus random node assignment option for the *surrogates* argument.

This scenario is opposite for *data-2* with 42% missing values. Setting *surrogates* = *F* retained more participants compare to *T* option for *data-2*. Setting surrogates to *F*, implies random assignment of participants with NA values to a random child node when sending the data down the tree.

To evaluate the performance of *surrogates* argument for handling the amount of missing data, we repeated the same experiment as mentioned in Section 3.7.2 and changing the amount of NA's injected to completed data. For example, 50% missing means multiplying the proportion of missing values for all missing variables, handedness, PIQ/FIQ, and both variables, by 0.5. In other words, we decrease the number of missing values by half for all variables with missing values. Note that to evaluate the effect of amount of missing values on *data-1* and *data-2*, we set *surrogates* equal to *F* and *T*, respectively.

As results in the first four rows indicate in Table 3.8, as the number of missing values increases, given the fixed number of participants per group, SMD values improved for all matching variables. This implies the importance of surrogate option in the iterMatch package despite the increasing amount of missing values. To preserve SMD threshold of 0.15 in the last iteration, it is expected to retained more participants as the number of missing values decreases.

For *data-2*, changing the missing values of 15%, 50%, and 100%, yielded a similar pattern as in *data-1*. As the amount of missing value increases, given the fixed number of participants in the iteration-0, improved SMD values were observed (see Table 3.9). This

Table 3.7. Effect of surrogates argument on *data-1* and *data-2*.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
data-1, surrogates = T, *method-1*						
SMD	0.10	0.11	0.01	0.15	0.08	308
p-value	0.25	0.17	0.93	0.10	0.46	308
data-1, surrogates = F, *method-1*						
SMD	0.10	0.07	0.02	0.05	0.14	302
p-value	0.25	0.33	0.81	0.56	0.20	302
data-2, surrogates = T, *method-2*						
SMD	0.15	0.05	0.21	0.15	0.01	398
p-value	0.04	0.61	0.84	0.15	0.92	398
data-2, surrogates = F, *method-2*						
SMD	0.15	0.01	0.06	0.07	0.08	430
p-value	0.04	0.93	0.59	0.57	0.52	430

implies the efficiency of surrogates option of *F* for *data-2* in terms of utilizing data with partial missing values in the final matched sample.

Table 3.8. Effect of amount of missing value on *data-1* with surrogates = F.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
Gold Standard, iteration-0						
SMD	0.20	0.21	0.14	0.22	0.23	129
p-value	0.14	0.09	0.26	0.08	0.07	129
15%, *method-1*, iter-0, surr = F						
SMD	0.20	0.19	0.15	0.18	0.18	129
p-value	0.14	0.13	0.23	0.15	0.17	129
50%, *method-1*, iter-0, surr = F						
SMD	0.22	0.17	0.21	0.13	0.17	129
p-value	0.11	0.16	0.09	0.33	0.23	129
100%, *method-1*, iter-0, surr = F						
SMD	0.28	0.03	0.06	0.07	0.12	129
p-value	0.04	0.78	0.66	0.58	0.43	129
Gold Standard, last-iter, surr = F						
SMD	0.06	0.02	0.02	0.03	0.13	94
p-value	0.84	0.89	0.89	0.83	0.34	94
15%, *method-1*, Last-iter, surr = F						
SMD	0.14	0.13	0.09	0.00	0.10	118
p-value	0.38	0.30	0.46	0.95	0.46	118
50%, *method-1*, Last-iter, surr = F						
SMD	0.28	0.03	0.06	0.07	0.12	107
p-value	0.04	0.78	0.66	0.58	0.43	107
100%, *method-1*, Last-iter, surr = F						
SMD	0.15	0.11	0.08	0.09	0.03	106
p-value	0.37	0.44	0.56	0.54	0.88	106

Table 3.9. Effect of amount of missing value on *data-2* with surrogates = T.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
Gold Standard, iteration-0						
SMD	0.31	0.15	0.15	0.49	0.23	234
p-value	0.00	0.09	0.10	< 0	0.01	234
15%, *method-2*, iteration-0						
SMD	0.26	0.10	0.12	0.48	0.21	234
p-value	0.01	0.29	0.22	< 0	0.03	234
50%, *method-2*, iteration-0						
SMD	0.25	0.13	0.12	0.51	0.17	234
p-value	0.01	0.17	0.21	< 0	0.12	234
100%, *method-2*, iteration-0						
SMD	0.31	0.06	0.15	0.40	0.11	234
p-value	0.00	0.66	0.27	0.00	0.42	234
Gold Standard, *method-2*, Last-iteration						
SMD	0.09	0.07	0.08	0.01	0.13	114
p-value	0.61	0.60	0.54	0.96	0.34	114
15%, *method-2*, Last-iteration						
SMD	0.14	0.03	0.06	0.12	0.03	183
p-value	0.23	0.80	0.60	0.29	0.79	183
50%, *method-2*, Last-iteration						
SMD	0.14	0.02	0.05	0.02	0.04	164
p-value	0.27	0.87	0.62	0.87	0.76	164
100%, *method-2*, Last-iteration						
SMD	0.14	0.01	0.09	0.14	0.04	159
p-value	0.27	0.96	0.58	0.37	0.80	159

CHAPTER 4
MIXED-EFFECTS RANDOM FOREST-BASED CLASSIFICATION ALGORITHMS FOR CLUSTERED DATA

4.1 OVERVIEW

To date, a variety of classification schemes have been proposed, and the accuracy of classification has reached as high as 95 percent for many disorders, including Autism Spectrum Disorder (ASD). However, to build a reliable and robust classification model for ASD, it is necessary to incorporate a large dataset which is often obtained from multi-site imaging data. In addition to the extended sample size, including multiple MRI modalities can increase the coherency of the brain picture.

However, two challenges are associated with an extended sample size and multi-modal dataset. The first issue is controlling the source of variation that is imposed by multiple imaging sites. Second, it is necessary to use a dimensional reduction algorithm to cope with the computational complexity of multi-modal data. Controlling the multi-site variability is particularly important as it can be mixed with the heterogeneous nature of ASD to build a robust and accurate classification model.

We addressed both concerns by proposing two mixed-effects random forest-based classification algorithms, applicable to multi-site (clustered) data using rs-fMRI and structural MRI (sMRI) modalities. These algorithms control the random effects of the confounding factor of the imaging site. Additionally, the algorithms internally control the fixed effect of the phenotypic variables such as age while building classification model. Moreover, they eliminate the necessity of utilizing a separate dimension reduction algorithm for high-dimensional data such as functional connectivity in a non-linear fashion.

In our empirical data study, we used an unseen external validation set including resting-state fMRI and sMRI from the ABIDE dataset. The RFME-based classification algorithms showed an accuracy of over 80 percent to distinguish ASD participants from typically developing (TD) participants. We concluded that the RFME-based classification model is a promising tool to build a more reliable and efficient diagnostic model for multi-site and multi-modal datasets.

4.2 DATA

The data for this analysis were selected from two versions of Autism Imaging Data Exchange, (ABIDE-I) [23] and (ABIDE-II) [22], and SDSU's proprietary data. ABIDE dataset is publicly available at $http://fcon_1000.projects.nitrc.org/indi/abide/$. The dataset contains 70 variables from 1276 total participants from ABIDE-I and 996 participants from ABIDE-II across 41 international research institutes. Since ABIDE-I and ABIDE-II contain limited SDSU data, we excluded SDSU data from these two datasets and included them with more participants in the final dataset. Therefore, these statistics exclude participants from the SDSU site. Additional 236 participants from the proprietary data were added to the pool of data. Therefore, the initial sample from ABIDE and SDSU added up to 2508 participants.

However, to include high-quality anatomical data with Local Gyrification Index (lGI) features, the data sample was limited to 95 ASD and 95 typically developing (TD) participants that was selected from a study by Kohli et al. [41]. The selected sample was comprised of three imaging sites, SDSU, NYU-1-ABIDE-II, and NYU-ABIDE-I. However, among these selected participants, 47 participants were excluded due to low fMRI data quality. Therefore, a total of 143 participants consisting of 72 ASD and 71 TD participants were picked for further analysis.

In Table 4.1, we labeled data from the NYU-1-ABIDE-II sample as NYU-II and NYU-ABIDE-I as NYU-I. A summary of demographic data and matching status between the

Table 4.1. Demographic summary of data per imaging sites.

Dataset: SDSU		
	ASD, M $\pm Var$ [range]	TD, M $\pm Var$ [range]
N (Female)	41 (8)	41 (7)
Motion (mm)	0.07$\pm$0.00 [0.02-0.15]	0.07$\pm$0.00 [0.01-0.15]
Age (years)	14.18$\pm$5.4 [9.2-17.8]	13.7$\pm$7.24 [6.90-17.6]
Handedness (Left/Right)	(6/35)	(6/35)
Dataset: NYU-II		
	ASD, M $\pm Var$ [range]	TD, M $\pm Var$ [range]
N (Female)	10 (0)	7 (0)
Motion (mm)	0.09$\pm$0.02 [0.04-0.18]	0.07$\pm$0.00 [0.04-0.12]
Age (years)	10.8$\pm$8.7 [7.2-17.9]	10.2$\pm$1.1 [9.03-11.3]
Handedness (Left/Right)	(4/6)	(4/6)
Dataset: NYU-I		
	ASD, M $\pm Var$ [range]	TD, M $\pm Var$ [range]
N (Female)	21 (0)	23 (0)
Motion (mm)	0.09$\pm$0.00 [0.03-0.17]	0.05$\pm$0.00 [0.02-0.12]
Age (years)	11.8$\pm$7.9 [7.2-18.6]	12.3$\pm$7.4 [9.03-11.3]
Handedness (Left/Right)	All missing	All missing

ASD group vs. the TD group per site is provided in Table 4.1.

4.2.1 Structural MRI (sMRI)

High resolution T1-weighted MRI sequences were obtained on a Siemens Allegra MR 2004a scanner (repetition time = 2530 ms, echo time = 3.25 ms, flip angle = 7°, 171 slices, $1.0 \times 1.0 \times 1.0\ mm^3$ resolution). FreeSurfer version 5.3.0 was employed to perform semi-automated cortical reconstruction on data from both study samples [18]. All FreeSurfer output was examined on a slice-by-slice basis to identify any inaccuracies in surface placement. Inaccuracies were corrected with white matter control points as needed, and then reassessed for accuracy. Scans that still showed inaccuracies were excluded. Scans with major artifacts, such as ghosting or ringing or inaccuracies deemed unlikely to be ameliorated by manual edits (based on past experience), were excluded. Cortical Surface Area (CSA) and Cortical Thickness (CT) were automatically measured at each FreeSurfer surface vertex, and Local Gyrification Index (lGI) was measured using an added flag to the FreeSurfer reconstruction processing stream [59]. lGI is a 3D surface-based method for calculating the ratio of cortical surface area within the sulcal folds (pialsurface) relative to the amount of cortex on the outer visible cortex (cortical hull). This calculation was made within a sphere with a 25 mm radius around the pial surface vertex. This automated reconstruction feature has been validated as a reliable measure of gyrification against manual measurement [58]. After anatomical pre-processing, we extracted 34 ROIs per hemisphere with 3 measurements of cortical surface area, cortical thickness, and local gyrification index plus 2 mean cortical thickness measures. Therefore, a total of 206 anatomical variables were extracted to be used toward building a classification model.

4.2.2 Functional Connectivity MRI (fcMRI)

The pipeline followed to process and analyze the fMRI data was similar to Jahedi et al. [37], using AFNI and FSL software. The first 5-time points from each scan were discarded to allow for T1 equilibrium. The remaining 180-time points underwent pre-processing steps such as motion, slice-time, and field-map correction. Functional data was aligned to

anatomical data using FLIRT (six degrees of freedom), resampled to 3.0 mm isotropic voxels (sinc interpolation), and standardized to the MNI-152 template (FSLs nonlinear registration tool), all in a single transformation step. Data was spatially smoothed with an isotropic 6 mm full-width at half-maximum Gaussian filter. Residual time-series of the resting-state images were then band-pass filtered $(0.008 < f < 0.08Hz)$ using a second-order Butterworth filter [58]; [57].

Head displacements in scanners for each brain volume were computed as the Root Mean Square Deviation (RMSD), calculated from 3 translational and 3 rotational motion parameters. Signal averaged over white matter and ventricles (from FreeSurfer segmentation), as well as their first derivatives, were removed by regression of nuisance variables. The nuisance regressors were bandpass filtered using the same procedure as for BOLD time series [30]. The time points with head motion > 0.5 mm, and the 2 successive time points were censored. If censoring occurred twice within 10-time points, all the time points between them were removed.

Due to known impacts of even minuscule amounts of head motion on BOLD fluctuations [50], strict inclusion criteria were set regarding motion and only participants with at least 80 % of volumes were retained after censoring volumes with too much head-motion (defined at > 0.5 mm) were included, as were in the final analyses. Additionally, individuals with root-mean-square displacement (RMSD, a summary measure of head motion throughout the scan) > 0.2 mm were also excluded. After careful quality control for functional data, the participants that had both modalities was selected.

ROIs considered for this analysis were obtained from Harvard Oxford [20] and from Jülich [24]. Initially, a brain coverage mask is generated which has a BOLD signal in 80% of participants. ROIs were excluded if they had less than 80% of the volume inside the brain mask. The final analysis included 342 ROIs obtained from 313 cortical, 14 subcortical, and 15 cerebellar regions (Table 4.2). The data was organized in a diagonally symmetrical 342×342 feature matrix with each cell in the matrix representing a Fisher z-transformed correlation

Table 4.2. Summary table for Regions of Interests (ROIs).

	Number of ROIs	Surface area (mm2)	Number of Voxels
Cortex	313 (92%)	273.3±255.0 [15.96-1935.34]	52.7±45.18 [4-336]
Subcortex	14 (4.1%)	13.4±5.95 [5-21]	NA
Cerebellum	15 (4.4%)	10.6±7.09 [2-27]	NA

between two ROIs. Consequently, there were 58311 ($\frac{342\times341}{2}$) unique features for each participant.

4.3 METHODS

In this section, we propose two binary classification models using Mixed-Effects Random Forest (RFME). Recall that we are interested in building a classification model such that the confounding effects of variables can be handled internally for high-dimensional clustered data without using any external dimension reduction algorithm. Note that it is essential to validate the results of this method using a new sample.

4.3.1 Random Forest Using Mixed-Effects for Clustered Data

Random Forest using Mixed-Effects (RFME) algorithm derived its origin from combining the concept of linear mixed-effects modeling with random forest. Similar to tree-based methods like Classification and Regression Trees (CART) [9], RFME ensembles many weak learners, as in Mixed-Effects Regression Trees (MERT) [28], to build a stronger learner. The basic idea behind MERT is to dissociate the fixed effects from the random effects in clustered data. This disassociation is particularly vital in clustered-structure data due to the similarity of participants that belong to the same cluster, as opposed to those from different clusters. In MERT, a standard regression tree [28] is used to model the fixed effects with a node-invariant linear structure at each terminal node. In our proposed RFME model, not only

are observations split within clusters, but the confounding effect of a phenotypic variable can be controlled at each splitting child node.

The basis of this chapter is an extension of a random forest-based dissimilarity matrix proposed in Chapter 2. This incorporates mixed models to build more reliable classification models. To this end, our proposed algorithm has the following steps:

First step is to handle the high-dimensionality of datasets. For the high-dimensional clustered data (multi-site data), only a random subset of variables participate in building the tree. This means that the user can choose to pick any arbitrary number of variables less than or equal to the number of variables in the dataset. This parameter is referred to as *SelVar* in the algorithm. To avoid bias in building the classification model using a random set of variables, the selected number of variables are randomly picked with replacement for each tree. Note that for low-dimensional data, this parameter should be equal to the number of variables in the dataset.

Unlike standard random forest [8] in which a bootstrap sample (drawn with replacement from the training data) constructs each tree, we use all of the data to build each tree. This distinction is necessary to build a proximity matrix for all pairs of observations in the proposed algorithm which are introduced as in Algorithm 5 and Algorithm 6.

After reducing the dimensionality, at each splitting node, $\sqrt{SelVar}$ are picked to "decorrelate" the trees for further splitting. For all randomly selected variables, a random selected cutpoint from all possible cut point values, which is referred to as Extremely Randomized trees (ER)[26], is chosen. The rationale behind choosing ER for cutpoint selection is to reduce the computational complexity of picking the cutpoint procedure. Also, it has been shown that ER has the best overall accuracy and performance for classification problems [10]. It not only has the lowest misclassification rates, but it also reduces or eliminates the variable selection bias, and is the fastest algorithm.

Next, for the selected variables along their cutpoints, an evaluation is conducted at

each child node using a mixed-effects model as follows:

$$p(Y_{ij} = 1|X) = \beta_0 + \beta_1 P_{ij} + \eta_i + \beta_2 I(X_i \leq C) \tag{4.1}$$

In this equation, the random component of the mixed model is denoted by η_i, that controls the cluster effect and a separate fixed effect component of P_i for phenotype variable. $I(X_i \leq C)$, which controls the population-averaged tree (forest), consists of imaging modality variables. The random intercept part, η_i, is still assumed to be linear. The goal of using mixed-effects is to predict the response for a new observation j that belongs to a cluster i. As previously mentioned, the advantage of having a phenotype variable as a separate fixed-effect is to partition the variable space controlling for the fixed effect of phenotype at each splitting node. Note that clusters do not need to be balanced, i.e., each cluster can have a different number of observations.

To estimate the parameters of the mixture model, lme4 package was used. The lme4 package separates the efficient computational linear algebra required to compute profiled likelihoods and deviances for a given value of θ from the nonlinear optimization algorithms, which use general-purpose nonlinear optimizers [5]. We incorporated two built-in optimization choices of the Nelder-Mead simplex algorithm [39] and a wrapper for Powell's *BOBYQA* algorithm [49] with ten points per axis. This one was done to evaluate the adaptive Gauss-Hermite approximation to the log-likelihood or nAGQ = 10.

Variables with a small *p*-value are selected for node splitting. In estimating the LME model, sometimes parameters cannot converge; therefore, models face a convergence problem. For these cases, a logistic model is used, which includes the splitting variable along with cluster and phenotype variables. As data split into further child nodes or get closer to terminal nodes, the clustering variable as a factor variable can become purer with only one level. For these cases, we continued by excluding the problematic variable (in our case cluster variable) in the logistic model. Individual trees in RFME are typically grown until some loose stopping criteria are met without pruning. These criteria include the maximum depth of the tree, the minimum number of participants to split a node, or the minimum number of

participants at a terminal node.

In this algorithm, before constructing each tree, only complete data were considered, i.e., observations with missing values were discarded from the cohort. To ensure the stability of the results, various number of trees were explored. To calculate the classification accuracy of the proposed RFME model, a validation data sample was sent down through the built RFME model. Consequently, a two by two confusion matrix of actual diagnostic labels was compared with the predicted terminal nodes for each observation. The algorithm for the classification RFME model is as follows:

Inputs: *data*, *formula*, *nTree*, *distance*, *randEf*, *fixedEf*, *selVar*, *minsplit*, *rfmtry* , *iSeed*

Output: RFME classification accuracy

Data: Training data

Dimension reduction

for *i in 1:nTree* **do**

  Binary labels use binary labels of the data to train the model

  while *Splitting criteria* **do**

    Variable selection

    Cutpoint selection

    Child node splitting evaluation Use mixed effect model

    $$p(Y_{ij} = 1|X) = \beta_0 + \beta_1 P_{ij} + \eta_i + \beta_2 I(X_i \leq C)$$

    Fit training data to the built model

  end

end

Terminal node prediction for test data

Calc classification accuracy of the RFME classification using a 2x2 confusion matrix

Result: Classification accuracy using RFME Model

Algorithm 5: RFME classification algorithm

4.3.2 RFME-Based Dissimilarity Matrix Algorithm

This method borrows its basis from the RFME model presented in Section 4.3.1 and adds a couple of additional steps to improve the classification accuracy using the two most common clustering methods and a defined RFME-based proximity matrix. To the best of our knowledge, none of the existing binary classification algorithms used clustering techniques to enhance the classification accuracy controlling for the effect of confounding variables internally. In this algorithm, the importance of the RFME model is to define a proximity matrix to be fed in clustering methods. As stated in Section 4.3.1, unlike typical random forest, in which bootstrap sample of data would be selected to participate in building each tree, for the proposed RFME model, all data were selected to participate in building the model. After building the RFME model using a trained sample, we used unseen data referred to as an external validation set to predict the terminal node for each participant based on the mean propensity score. The propensity score is defined as the conditional probability of treatment given covariates, where treatment is considered as treated participants, or ASD, while X denotes all covariates, excluding the treatment indicator.

$$P(Z = 1|X) = E(Z|X)$$

Our defined proximity matrix assigned a distance of zero if any pair of participants are in same terminal node, and one, if participant pairs are predicted in different terminal nodes. Thus,

$$d_{ij} = \begin{cases} 0, & \text{In the same terminal node} \\ 1, & \text{Otherwise} \end{cases}$$

The proximity matrix for each tree is then averaged across all trees. After this step, the proximity matrix is fed into agglomerative Hierarchical Clustering (HC) and K-means [33] with two predefined cluster groups. A two by two confusion matrix consists of predicted group membership for the external validation set, and actual binary labels are formed to calculate the binary classification accuracy using RFME-based proximity matrix created by

K-means and hierarchical clustering algorithms. We referred to them as RFME-KC and RFME-HC, respectively. The algorithm 6 explains the major steps of this classification method.

Inputs: *data*, *formula*, *nTree*, *distance*, *randEf*, *fixedEf*, *selVar*, *minsplit*, *rfmtry* , *iSeed*
Outputs: RFME classification accuracy from RFME-based proximity matrix
Data: Training data
Dimension reduction
Building RFME model
Predict test data using the built RFME model
if *participants i and j are in the same node* **then**
    d_{ij}=0
end
else
    d_{ij}=1
end
Feed-in distance matrix to HC and K-means clustering for 2 clusters
Cluster membership prediction for external validation set for both K-means and HC clustering
Calc cluster accuracy using confusion matrix for 2 clusters using HC and K-means clustering
Result: Classification accuracy using RFME-based dissimilarity matrix

Algorithm 6: RFME-based dissimilarity matrix classification algorithm (RFME-KC and RFME-HC)

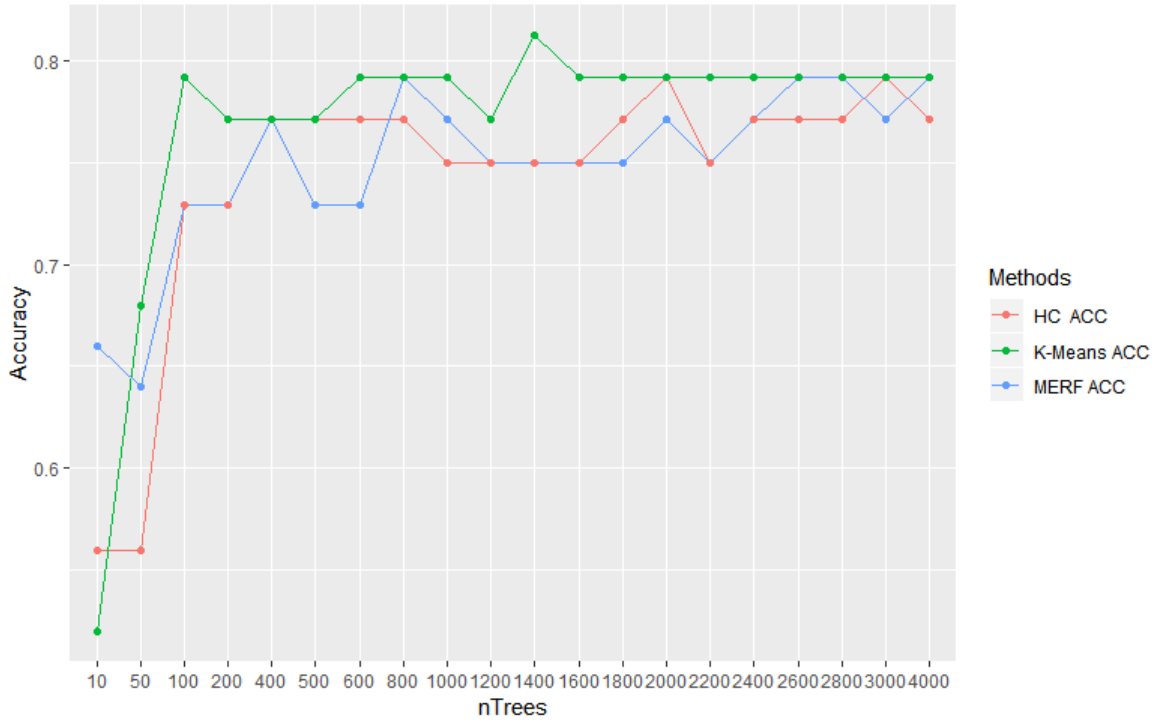
4.4 RESULTS

In creating RFME, multiple parameters had to be tuned. These parameters are the number of trees that we referred to as *nTree* and three stopping criteria. Three stopping criteria were chosen to prevent further splitting the data in a tree. These include the minimum number of observations for further splitting, set to 20 (defined as *minsplit*), the minimum number of observations in each terminal node to stop tree building, set to 5 (*minbucket*), and the maximum depth of the tree, set to 10 (*maxdepth*).

For tree-based models, various numbers of trees have to be explored to assure the stability of results. In our experiments, stable results started to emerge with 2000 trees (see Figure 4.1.) Therefore, an *nTree* of 2000 was used for all empirical experiments.

In this chapter, three empirical experiments were designed to answer the following questions. How does the proposed Algorithm 5 perform vs. Algorithm 6 in terms of accuracy? Also, how is the accuracy performance for these algorithms when using clustered data?

Figure 4.1. Number of trees vs. test classification accuracy.



To answer these questions, we created three different data samples extracted from data in Section 4.2. In one, we included training data from SDSU and NYU-I and left NYU-II for external validation sample shown as *Train:SDSU+NYU-I* in Table 4.3. In the second dataset, to increase the number of samples in the validation set, SDSU and NYU-II were combined for training, and NYU-I was used for validating the models, which was referred to as *Train:SDSU+NYU-II* in Table 4.3. In the third sample, however, to increase variability within sites in both train and validation samples, we used stratification sampling. Thus, $2/3$ of participants from each site were randomly picked to be used for training, and the $1/3$ remaining data per site was utilized for the validation sample. We call this sample *Train:2/3StratFromAllSites*.

As we mentioned in both proposed algorithms, to avoid computational complexity and to provide the possibility of including all features without external dimension reduction, we randomly selected variables with replacements before building any tree. In our algorithms, this value was referred to as *selVar*. Depending on the type of modality, the *selVar* value was 15000 or 206. For example, in the analysis provided in Section 4.4.2, 15000 random variables were selected if FC modality was chosen.

One of the other interesting questions that has been recently addressed by Eill et al. [25] is which imaging modality is more informative to detect the classification label for autism. Empirical results to answer this question are provided in Section 4.4.2. To investigate the informativeness of anatomical modality, all 206 anatomical variables were used for parameter *selVar*, whereas *selVar* was 15000 for combined modality or separate functional modality.

4.4.1 Impact of Mixed-Effects Modeling

In this section, we seek to answer which proposed algorithms can achieve higher classification accuracy and how including the mixed-effects modeling in each terminal node affects diagnostic classification accuracy. To this end, we experimented with six different combinations where age as a fixed effect and site as a random effect were either included or

not included for model fitting at each splitting node. For example, in the first row of Table 4.3 an experiment was conducted named *siteRandomAgeFixed*, which means the site was in the random effect component of LME and age took part in the fixed effect. The second column of the same table indicates the type of model that was used for node splitting. Possible models, as stated in the algorithm, were mixed LME and logistic models.

The next column determines if the site was used in the random component or not. A column with RFME indicates the accuracy of the RFME model from Algorithm 5. The following columns represent the accuracy of Algorithm 6. To ensure the validity of results three datasets were tested, *Train:SDSU+NYU-I*, *Train: SDSU+NYU-II*, and *Train: 2/3StratFromAllSites*.

Although results for the external validation set of *Train:SDSU+NYU-I* were promising and were close to 100% for both algorithms, a relatively small validation set could potentially explain the overestimated accuracy from these models.

In the second experiment, with a relatively larger sample of 44 participants from *Train:SDSU+NYU-II* from one site, two points are noticeable. First, the consistent higher prediction accuracy of the *RFME-KC* and the *RFME-HC* algorithms compared to the RFME algorithm was found. Second, the accuracy differences between the Algorithm 6 vs. Algorithm 5 reached their maximum of 15% for various models, such as *siteFixedAgeFixed*, *siteNotincludedAgeFixed*, *siteNotincludedAgeFixed*, and *siteNotIncludedAgeNotIncluded*. However, in the third dataset, *TrainedData:2/3StratfromAllSites*, the maximum accuracy differences of 13% between models appeared for the *siteRandomAgeNotIncluded* experiment which only had imaging site as the random effect. This indicates the role of including the random effect component when building the model for the clustered data. Thus, as the variation in the training and the external validation sample increased, RFME-KC outperformed the *RFME* and *RFME-HC* algorithms.

The last column of this table under ”Conv Rate,” referred to as the convergence rate for the LME parameter estimation when the random effect was included in the model.

Although there were some instances in which the LME parameter estimation had a convergence problem, over 98% of the time no convergence problem had occurred.

We concluded accuracy results from the Algorithm 6 outperformed the accuracy results of the Algorithm 5 for all three datasets. Moreover, the *RFME-KC* algorithm provided a higher-performance classification accuracy compared to the *RFME* and the *RFME-HC* algorithms using the multi-site dataset, ABIDE. Therefore, our proposed algorithm has the capability of increasing clustering accuracy for the clustered data. Regardless of which clustering method was used, RFME-based dissimilarity matrix classification algorithms could improve the accuracy results of a classification model compared to the RFME model.

Table 4.3. Effect of Fixed and Random Effects on two samples using functional and anatomical modality.

Experiment	Model	Age	Site	RFME	HC	K-means	Conv
TrainedData: *SDSU+NYU-I*							
siteRandomAgeFixed	Mixed	Fixed	Random	1	0.94	0.88	0.99
	Logistic	Fixed	Fixed				
siteFixedAgeFixed	Mixed	Fixed	Fixed	1	0.94	0.94	Irr
siteNotIncludedAgeFixed	Mixed	Fixed	-	1	0.94	0.94	Irr
siteRandomAgeNotIncluded	Mixed	-	Random	0.94	0.94	1	0.99
	Logistic	-	Fixed				
siteFixedAgeNotIncluded	Mixed	-	Fixed	1	0.94	0.94	Irr
siteNotIncludedAgeNotIncluded	Mixed	-	-	1	0.94	0.94	Irr
	Logistic	-	-				
TrainedData: SDSU+NYU-II							
siteRandomAgeFixed	Mixed	Fixed	Random	0.69	0.78	0.8	0.98
	Logistic	Fixed	Fixed				
siteFixedAgeFixed	Mixed	Fixed	Fixed	0.67	0.78	0.82	Irr
siteNotincludedAgeFixed	Mixed	Fixed	-	0.67	0.78	0.82	Irr
siteRandomAgeNotIncluded	Mixed	-	Random	0.69	0.82	0.78	0.99
	Logistic	-	Fixed				
siteFixedAgeNotIncluded	Mixed	-	Fixed	0.67	0.67	0.82	Irr
	Logistic	-	Fixed				
siteNotIncludedAgeNotIncluded	Mixed	-	-	0.67	0.78	0.82	Irr
	Logistic	-	-				
TrainedData: 2/3StratfromAllSites							
siteRandomAgeFixed	Mixed	Fixed	Random	0.75	0.79	0.79	0.99
	Logistic	Fixed	Fixed				
siteFixedAgeFixed	Mixed	Fixed	Fixed	0.79	0.79	0.83	Irr
siteNotIncludedAgeFixed	Mixed	Fixed	-	0.79	0.79	0.83	Irr
siteRandomAgeNotIncluded	Mixed	-	Random	0.64	0.75	0.77	1
	Logistic	-	Fixed				
siteFixedAgeNotIncluded	Mixed	-	Fixed	0.79	0.79	0.83	Irr
siteNotIncludedAgeNotIncluded	Mixed	-	-	0.79	0.79	0.83	Irr
	Logistic	-	-				

4.4.2 Effect of Imaging Modalities

The empirical analysis in this section aimed to investigate what imaging modality is more informative to detect classification labels for autism. As described in Section 4.4, three datasets were tested using only anatomical, only functional, and combined functional and anatomical modalities. The accuracy fluctuation of including these modalities may contribute to feature-richness and informativeness of the modality.

In the *Train:SDSU+NYU-I* with a small validation set, the *RFME* model with an accuracy of 1 is slightly higher than 94% accuracy for the combined modality or 88% accuracy for only the functional modality. However, this result cannot be trusted as the validation sample is very small. Regardless of the data experiment, including only anatomical modality did not provide a high classification accuracy whereas including only functional or combined modalities could achieve a high classification accuracy.

Another point to mention from Table 4.4 is the overall higher accuracy of the *RFME-KC* algorithm compared to the *RFME* and the *RFME-HC* algorithms. Comparing results from *Train:SDSU+NYU-II* and the *Train:2/3StratFromAllSites*, which provided a higher range of variability within sites, we observed a similar classification accuracy when both modalities were included. Overall, results from this section show the informativeness of combined modalities for predicting the diagnostic status of ASD.

Table 4.4 shows more information on this comparison when different modalities are presented.

4.4.3 Computational Complexity

All computations were conducted on CPU clusters at the Computational Science Research Center (CSRC) at San Diego State University. The code was designed to use multiple processes in parallel, which employed the power of 16 nodes of 48GB of RAM with two quad-core processors per node. All the analysis was performed in the R language and

Table 4.4. Effects of imaging modalities on new sample test accuracy with age as fixed effect and site as random effect.

Data	Experiment	RFME ACC	HC ACC	K-means ACC	Converge Rate
Train:SDSU+NYU-I	FC+Anat	1	0.94	0.88	0.99
	Anat	0.85	1	1	0
	FC	1	0.88	0.88	0.99
Train:SDSU+NYU-II	FC+Anat	0.69	0.78	0.8	0.98
	Anat	0.53	0.51	0.55	0.97
	FC	0.6	0.84	0.77	0.98
Train:2/3StratFromAllSites	FC+Anat	0.75	0.79	0.79	0.99
	Anat	0.58	0.62	0.62	0.99
	FC	0.73	0.73	0.72	0.99

environment for statistical computing. A significant limitation of constructing RFME was the computational intensity. The major time-consuming part of the analysis was related to building the random forest part. Factors that affected computation speed in building RFME were: number of trees in building the forest, fitting a mixed-effects model at each node, the number of categorical and continuous predictors, and the minimum node size to continue splitting the branch.

To assess computation time, we recorded time in seconds varying the number of trees using stratification sampling data explained in the results section with a *minsplit* = 20, and *minbucket* of 5. While running the analysis did not utilize more efficient software programs such as C++ or Fortran, most of the computation time was devoted to calculating the splitting statistic. Even though we restricted the number of cut points for a limited number of variables, the high-dimensionality of our data increased the computation time.

CHAPTER 5

CONCLUSION

5.1 SUMMARY AND FUTURE WORK

In this dissertation, multiple algorithms and methods were proposed to improve sample matching and the classification modeling. The main building blocks that was used in both methods was random Forest (RF). Although these RF-based methods and algorithms can be used for different fields of research, the specific focus of dissertation was to apply these methods to autism research.

In Chapter 2, multivariate matching was discussed. Matching is a nonparametric method of controlling for the influence of confounding variables in observational studies. In other words, observations from the data are pruned so that the remaining data has a better balance between the treated and control groups. This means that the empirical distributions of the covariates (X) in the groups are more similar.

Five different matching algorithms were proposed that incorporated the idea of 1: k nearest matching and full matching algorithms. The performance of all these algorithms was evaluated using a small sample of BDIL data, which had a limited number of control individuals to be used for matching with treated participants. Among all, the 1:3 nearest method did not perform well due to a sparse proximity matrix. This is aligned with the previous finding from Gu et al.[27]. These researchers showed nearest neighbor matching performs poorly when there is intense competition for control participants. However, it performs well when there is little competition.

However, full matching algorithms performed well for creating a balance between the number of matched participants in both groups and the distribution of covariates. One reason is that the participant sequence order in which the treated participants are matched is not

important. Instead, optimal matching takes individual matches and minimizes their global distance measure [55].

Comparing SMD results from *IterMatch* and *IterMatchCEM* algorithms, the latter slightly outperformed the former method; however, the *IterMatch* algorithm preserved more participants for the matching set. Preserving participants is more important when the sample size is small. One shortcoming associated with *IterMatchCEM* that worked against small sample sizes was arbitrary bin user-defined cutpoint. User-defined cutpoint which was addressed by using a distance-based bin, could decrease the sample in each group. Therefore, *IterMatchCEM* might not be a suitable algorithm when a limited sample is available for matching. Both algorithms had flexibility of choosing a tighter threshold that might affect the analysis more severely compared to other covariates. For example, in our study, we could be more interested in a tighter match on head motion, and a less restrictive match on NVIQ.

Besides the mentioned characteristics, one of the main shortcomings associated with the presented *IterMatch* and *IterMatchCEM* algorithms was incapability of handling missing values. Given the strong RF background on handling missing data, this issue was addressed in the implementation phase of this algorithm in Chapter 3.

Researchers in applied fields often have to deal with observations with missing data, and thus this motivated us to address matching challenges. *IterMatch* algorithm was chosen over other proposed algorithms introduced in Chapter 3 due to superior performance and flexibility of choosing the desired threshold while maintaining the maximum number of possible participants from both groups.

To the best of our knowledge, iterMatch is the only available package in R to match two groups of data with missing values internally. The package is easy-to-use and its key feature is setting a user-based threshold for finding a matched sample. It allows users to control the desired matching balance threshold for variables separately. The output file provides insightful information about the role of each participant in finding an optimally matched sample. This means, in each iteration, a list of matched participants' IDs, along with

the excluded participant's ID, is provided for the user to decide if any input variable can be altered to obtain a better-matched sample.

Having the option of a surrogate as one of the input parameters allows the user to handle missing values within the process of matching without the necessity of using any imputation techniques. This is particularly important for a matching package as all other existing matching packages can provide matched samples with complete cases. This implies that the user does not have to look for the missing values that need to be imputed separately. In addition, the matching package can handle data with a large number of predictors with mixed types (continuous, ordinal, and categorical with two or more levels) without the need for variable transformation, dummy variable creation, or variable selection. Users should note that this package only works for matching two groups. In other words, the response variable should be binary.

To retain more participants and desired balance in a sample, a user should tune the input parameters. To this end, we conducted multiple empirical simulations as a way to illustrate how to tune variables for different datasets and scenarios. For example, users should vary the number of trees to make sure results are stable. After checking the stability of results, it is recommended to try a similar number of trees around the stability value to retain more participants per group.

Depending on the number of missing values, users are given two choices for SMD methods to calculate the balance of the matched sample. In the empirical study, we showed that *method-1* and *method-2* could be selected depending on the available number of participants from both groups and the number of missing values.

One of the expected outcomes from a matched data with missing values is that as the proportion of missing values increases, the number of retained participants per group decreases. However, usage of the *surrogates* option in *iterMatch* enables retaining more subjects as the number of missing values increases. In our empirical study, we showed that for data with fewer missing values, setting the *surrogates* to *T* could maintain more subjects per

group. Conversely, data with fewer missing values could retain more subjects with *surrogates* = *F* while maintaining the same SMD thresholds.

The building blocks from the *iterMatch* algorithm were then borrowed to build a binary classification method for high-dimensional clustered data in Chapter 4. In other words, the idea of random forest and mixed-effects models were combined to handle the effects of correlated variables within clustered data, which we referred to as Random Forest using Mixed-Effects (RFME). In this chapter, we proposed three classification algorithms such as *RFME*, *RFME-HC*, and RFME-KC.

All algorithms can tolerate high-dimensional data such as fMRI without the necessity of externally using a dimension reduction algorithm. The undesirable effects of confounding variables are controlled internally in the model using the Linear Mixed-Effects (LME) models at each node per tree. Proposed algorithms can allow observations from the same cluster to be separated during the splitting process, both for random and fixed effects with an unbalanced number of observations in each cluster.

The new contribution of Chapter 4 was to design an RFME-based proximity matrix to be fed into a common clustering algorithms, which provided a higher classification accuracy. Results from our proposed algorithms showed over 80% prediction accuracy for an unseen sample. The prediction accuracy result was particularly important because, in most high-accuracy classification models, validation is done using k-fold cross-validation or Out-Of-Bag (OOB) error rate, where the test sample was already used to train the model. Utilizing external validation data based on *RFME-KC* indicates the model successfully managed the variation between imaging sites.

Moreover, we confirmed the greater informativeness of combined functional and sMRI modalities using the three proposed algorithms. Results indicate the informativeness of fMRI modality compared to sMRI modality for diagnostic classification of ASD vs. TD participants. Our results are consistent with the existing literature by Eill et al. [25].

One possible pitfall of our study in Chapter 4 was the inclusion of high-functioning

participants whose fMRI scans had low head motion. However, to capture the full picture of ASD, lower functioning participants with ASD should be included as well. Thus, findings may not generalize to the entire functional spectrum of ASD.

As previously mentioned, missing values for the given participant may result in the loss of that participant from the analysis. In other words, to analyze datasets with missing values, the recommended way of dealing with missing values is either to remove observations with more than one missing instance entirely or impute those missing values with one of various methods. In the current implementation of the algorithms in Chapter 4, we ignored cases with missing values. However, handling the missing values can easily be incorporated into the future R package implementation.

Another limitation was to choose a small sample from only three imaging sites which potentially decrease the actual variability between sites. Thus, increasing samples from multiple sites can contribute to build a robust and accurate prediction model for ASD in the future.

BIBLIOGRAPHY

[1] M. ABDOLELL, M. LEBLANC, D. STEPHENS, AND R. HARRISON, *Binary partitioning for continuous longitudinal data: categorizing a prognostic variable*, Statistics in medicine, 21 (2002), pp. 3395–3409.

[2] W. ADLER, S. POTAPOV, AND B. LAUSEN, *Classification of repeated measurements data using tree-based ensemble methods*, Computational Statistics, 26 (2011), p. 355.

[3] A. P. ASSOCIATION ET AL., *Diagnostic and statistical manual of mental disorders (DSM-5®)*, American Psychiatric Pub, 2013.

[4] J. BAIO, *Prevalence of autism spectrum disorders: Autism and developmental disabilities monitoring network, 14 sites, united states, 2008. morbidity and mortality weekly report. surveillance summaries. volume 61, number 3.*, Centers for Disease Control and Prevention, (2012).

[5] D. BATES, M. MÄCHLER, B. BOLKER, AND S. WALKER, *Fitting linear mixed-effects models using lme4*, arXiv preprint arXiv:1406.5823, (2014).

[6] S. L. BISHOP, C. FARMER, AND A. THURM, *Measurement of nonverbal iq in autism spectrum disorder: scores in young adulthood compared to early childhood*, Journal of autism and developmental disorders, 45 (2015), pp. 966–974.

[7] M. BLACKWELL, S. IACUS, G. KING, AND G. PORRO, *cem: Coarsened exact matching in stata*, The Stata Journal, 9 (2009), pp. 524–546.

[8] L. BREIMAN, *Random forests*, Machine learning, 45 (2001), pp. 5–32.

[9] L. BREIMAN, J. FRIEDMAN, R. OLSHEN, AND C. STONE, *Classification and regressiontrees, wadsworth statistics*, Probability Series, Belmont, California: Wadsworth, (1984).

[10] P. CALHOUN, M. J. HALLETT, X. SU, G. CAFRI, R. A. LEVINE, AND J. FAN, *Random forest with acceptance–rejection trees*, Computational Statistics, (2019), pp. 1–17.

[11] L. CAPITAINE, R. GENUER, AND R. THIÉBAUT, *Random forests for high-dimensional longitudinal data*, arXiv preprint arXiv:1901.11279, (2019).

[12] R. CARUANA, N. KARAMPATZIAKIS, AND A. YESSENALINA, *An empirical evaluation of supervised learning in high dimensions*, in Proceedings of the 25th international conference on Machine learning, 2008, pp. 96–103.

[13] R. CARUANA AND A. NICULESCU-MIZIL, *An empirical comparison of supervised*

learning algorithms, in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 161–168.

[14] H. N. CHAM, *Propensity score estimation with random forests*, PhD thesis, Arizona State University, 2013.

[15] C. P. CHEN, C. L. KEOWN, A. JAHEDI, A. NAIR, M. E. PFLIEGER, B. A. BAILEY, AND R.-A. MÜLLER, *Diagnostic classification of intrinsic functional connectivity highlights somatosensory, default mode, and visual regions in autism*, NeuroImage: Clinical, 8 (2015), pp. 238 – 245.

[16] W. G. COCHRAN AND S. P. CHAMBERS, *The planning of observational studies of human populations*, Journal of the Royal Statistical Society. Series A (General), 128 (1965), pp. 234–266.

[17] R. B. D'AGOSTINO JR, *Propensity score methods for bias reduction in the comparison of a treatment to a non-randomized control group*, Statistics in medicine, 17 (1998), pp. 2265–2281.

[18] A. M. DALE, B. FISCHL, AND M. I. SERENO, *Cortical surface-based analysis: I. segmentation and surface reconstruction*, Neuroimage, 9 (1999), pp. 179–194.

[19] G. DE'ATH, *Multivariate regression trees: a new technique for modeling species–environment relationships*, Ecology, 83 (2002), pp. 1105–1117.

[20] R. S. DESIKAN, F. SÉGONNE, B. FISCHL, B. T. QUINN, B. C. DICKERSON, D. BLACKER, R. L. BUCKNER, A. M. DALE, R. P. MAGUIRE, B. T. HYMAN, ET AL., *An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest*, Neuroimage, 31 (2006), pp. 968–980.

[21] E. A. DEYOE, P. BANDETTINI, J. NEITZ, D. MILLER, AND P. WINANS, *Functional magnetic resonance imaging (fmri) of the human brain*, Journal of neuroscience methods, 54 (1994), pp. 171–187.

[22] A. DI MARTINO, D. O'CONNOR, B. CHEN, K. ALAERTS, J. S. ANDERSON, M. ASSAF, J. H. BALSTERS, L. BAXTER, A. BEGGIATO, S. BERNAERTS, ET AL., *Enhancing studies of the connectome in autism using the autism brain imaging data exchange ii*, Scientific data, 4 (2017), p. 170010.

[23] A. DI MARTINO, C.-G. YAN, Q. LI, E. DENIO, F. X. CASTELLANOS, K. ALAERTS, J. S. ANDERSON, M. ASSAF, S. Y. BOOKHEIMER, M. DAPRETTO, ET AL., *The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism*, Molecular psychiatry, 19 (2014), p. 659.

[24] S. B. EICKHOFF, T. PAUS, S. CASPERS, M.-H. GROSBRAS, A. C. EVANS, K. ZILLES, AND K. AMUNTS, *Assignment of functional activations to probabilistic cytoarchitectonic areas revisited*, Neuroimage, 36 (2007), pp. 511–521.

[25] A. EILL, A. JAHEDI, Y. GAO, J. S. KOHLI, C. H. FONG, S. SOLDERS, R. A. CARPER, F. VALAFAR, B. A. BAILEY, AND R.-A. MÜLLER, *Functional connectivities are more informative than anatomical variables in diagnostic classification of autism*, Brain connectivity, 9 (2019), pp. 604–612.

[26] P. GEURTS, D. ERNST, AND L. WEHENKEL, *Extremely randomized trees*, Machine learning, 63 (2006), pp. 3–42.

[27] X. S. GU AND P. R. ROSENBAUM, *Comparison of multivariate matching methods: Structures, distances, and algorithms*, Journal of Computational and Graphical Statistics, 2 (1993), pp. 405–420.

[28] A. HAJJEM, F. BELLAVANCE, AND D. LAROCQUE, *Mixed effects regression trees for clustered data*, Statistics & probability letters, 81 (2011), pp. 451–459.

[29] A. HAJJEM, F. BELLAVANCE, AND D. LAROCQUE, *Mixed-effects random forest for clustered data*, Journal of Statistical Computation and Simulation, 84 (2014), pp. 1313–1328.

[30] M. N. HALLQUIST, K. HWANG, AND B. LUNA, *The nuisance of nuisance regression: spectral misspecification in a common approach to resting-state fmri preprocessing reintroduces noise and obscures functional connectivity*, Neuroimage, 82 (2013), pp. 208–225.

[31] B. B. HANSEN AND S. O. KLOPFER, *Optimal full matching and related designs via network flows*, Journal of computational and Graphical Statistics, 15 (2006), pp. 609–627.

[32] A. HAPFELMEIER, T. HOTHORN, AND K. ULM, *Recursive partitioning on incomplete data using surrogate decisions and multiple imputation*, Computational Statistics & Data Analysis, 56 (2012), pp. 1552–1565.

[33] J. A. HARTIGAN AND M. A. WONG, *Algorithm as 136: A k-means clustering algorithm*, Journal of the Royal Statistical Society. Series C (Applied Statistics), 28 (1979), pp. 100–108.

[34] J. HILL, *Reducing bias in treatment effect estimation in observational studies suffering from missing data*, (2004).

[35] S. M. IACUS, G. KING, AND G. PORRO, *Causal inference without balance checking: Coarsened exact matching*, Political analysis, 20 (2012), pp. 1–24.

[36] K. IMAI, G. KING, AND E. A. STUART, *Misunderstandings between experimentalists and observationalists about causal inference*, Journal of the royal statistical society: series A (statistics in society), 171 (2008), pp. 481–502.

[37] A. JAHEDI, C. A. NASAMRAN, B. FAIRES, J. FAN, AND R.-A. MÜLLER, *Distributed*

intrinsic functional connectivity patterns predict diagnostic status in large autism cohort, Brain connectivity, 7 (2017), pp. 515–525.

[38] Y. V. KARPIEVITCH, E. G. HILL, A. P. LECLERC, A. R. DABNEY, AND J. S. ALMEIDA, *An introspective comparison of random forest-based classifiers for the analysis of cluster-correlated data by way of rf++*, PloS one, 4 (2009), p. e7087.

[39] K. KLEIN AND J. NEIRA, *Nelder-mead simplex optimization routine for large-scale problems: A distributed memory implementation*, Computational Economics, 43 (2014), pp. 447–461.

[40] S. KLÖPPEL, A. ABDULKADIR, C. R. JACK JR, N. KOUTSOULERIS, J. MOURÃO-MIRANDA, AND P. VEMURI, *Diagnostic neuroimaging across diseases*, Neuroimage, 61 (2012), pp. 457–463.

[41] J. S. KOHLI, M. K. KINNEAR, C. H. FONG, I. FISHMAN, R. A. CARPER, AND R.-A. MÜLLER, *Local cortical gyrification is increased in children with autism spectrum disorders, but decreases rapidly in adolescents*, Cerebral Cortex, 29 (2018), pp. 2412–2423.

[42] N. M. LAIRD, J. H. WARE, ET AL., *Random-effects models for longitudinal data*, Biometrics, 38 (1982), pp. 963–974.

[43] B. K. LEE, J. LESSLER, AND E. A. STUART, *Improving propensity score weighting using machine learning*, Statistics in medicine, 29 (2010), pp. 337–346.

[44] S. K. LEE, *On generalized multivariate decision tree by using gee*, Computational Statistics & Data Analysis, 49 (2005), pp. 1105–1119.

[45] W.-Y. LOH, W. ZHENG, ET AL., *Regression trees for longitudinal and multiresponse data*, The Annals of Applied Statistics, 7 (2013), pp. 495–522.

[46] J. A. NIELSEN, B. A. ZIELINSKI, P. T. FLETCHER, A. L. ALEXANDER, N. LANGE, E. D. BIGLER, J. E. LAINHART, AND J. S. ANDERSON, *Abnormal lateralization of functional connectivity between language and default mode regions in autism*, Molecular Autism, 5 (2014), p. 8.

[47] S. OGAWA, T.-M. LEE, A. R. KAY, AND D. W. TANK, *Brain magnetic resonance imaging with contrast dependent on blood oxygenation*, proceedings of the National Academy of Sciences, 87 (1990), pp. 9868–9872.

[48] D. PINTO, A. T. PAGNAMENTA, L. KLEI, R. ANNEY, D. MERICO, R. REGAN, J. CONROY, T. R. MAGALHAES, C. CORREIA, B. S. ABRAHAMS, ET AL., *Functional impact of global rare copy number variation in autism spectrum disorders*, Nature, 466 (2010), p. 368.

[49] M. J. POWELL, *The bobyqa algorithm for bound constrained optimization without*

derivatives, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, (2009), pp. 26–46.

[50] J. D. POWER, A. MITRA, T. O. LAUMANN, A. Z. SNYDER, B. L. SCHLAGGAR, AND S. E. PETERSEN, *Methods to detect, characterize, and remove motion artifact in resting state fmri*, Neuroimage, 84 (2014), pp. 320–341.

[51] R CORE TEAM, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. ISBN 3-900051-07-0.

[52] K. RIDDLE, C. J. CASCIO, AND N. D. WOODWARD, *Brain structure in autism: a voxel-based morphometry analysis of the autism brain imaging database exchange (abide)*, Brain imaging and behavior, 11 (2017), pp. 541–551.

[53] A. RIEGER, T. HOTHORN, AND C. STROBL, *Random forests with missing values in the covariates*, (2010).

[54] P. R. ROSENBAUM ET AL., *Covariance adjustment in randomized experiments and observational studies*, Statistical Science, 17 (2002), pp. 286–327.

[55] P. R. ROSENBAUM ET AL., *Design of observational studies*, vol. 10, Springer, 2010.

[56] P. R. ROSENBAUM AND D. B. RUBIN, *Reducing bias in observational studies using subclassification on the propensity score*, Journal of the American statistical Association, 79 (1984), pp. 516–524.

[57] T. D. SATTERTHWAITE, M. A. ELLIOTT, R. T. GERRATY, K. RUPAREL, J. LOUGHEAD, M. E. CALKINS, S. B. EICKHOFF, H. HAKONARSON, R. C. GUR, R. E. GUR, ET AL., *An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data*, Neuroimage, 64 (2013), pp. 240–256.

[58] M. SCHAER, M. B. CUADRA, N. SCHMANSKY, B. FISCHL, J.-P. THIRAN, AND S. ELIEZ, *How to measure cortical folding from mr images: a step-by-step tutorial to compute local gyrification index*, JoVE (Journal of Visualized Experiments), (2012), p. e3417.

[59] M. SCHAER, M. B. CUADRA, L. TAMARIT, F. LAZEYRAS, S. ELIEZ, AND J.-P. THIRAN, *A surface-based approach to quantify local cortical gyrification*, IEEE transactions on medical imaging, 27 (2008), pp. 161–170.

[60] E. C. SCHNEIDER, A. M. ZASLAVSKY, AND A. M. EPSTEIN, *Use of high-cost operative procedures by medicare beneficiaries enrolled in for-profit and not-for-profit health plans*, New England Journal of Medicine, 350 (2004), pp. 143–150.

[61] M. R. SEGAL, *Tree-structured methods for longitudinal data*, Journal of the American Statistical Association, 87 (1992), pp. 407–418.

[62] R. J. SELA AND J. S. SIMONOFF, *Re-em trees: a data mining approach for longitudinal and clustered data*, Machine learning, 86 (2012), pp. 169–207.

[63] E. A. STUART, *Matching methods for causal inference: A review and a look forward*, Statistical science: a review journal of the Institute of Mathematical Statistics, 25 (2010), p. 1.

[64] K. SUPEKAR AND V. MENON, *Sex differences in structural organization of motor systems and their dissociable links with repetitive/restricted behaviors in children with autism*, Molecular autism, 6 (2015), p. 50.

[65] L. Q. UDDIN, K. SUPEKAR, C. J. LYNCH, A. KHOUZAM, J. PHILLIPS, C. FEINSTEIN, S. RYALI, AND V. MENON, *Salience network–based classification and prediction of symptom severity in children with autism*, JAMA psychiatry, 70 (2013), pp. 869–879.

[66] H. WU AND J.-T. ZHANG, *Nonparametric regression methods for longitudinal data analysis: mixed-effects modeling approaches*, vol. 515, John Wiley & Sons, 2006.

[67] Y. YU AND D. LAMBERT, *Fitting trees to functional data, with an application to time-of-day patterns*, Journal of Computational and graphical Statistics, 8 (1999), pp. 749–762.

[68] P. ZHAO, X. SU, T. GE, AND J. FAN, *Propensity score and proximity matching using random forest*, Contemporary clinical trials, 47 (2016), pp. 85–92.

Table 1. Effect of amount of missing value on *data-1* with surrogates = T.

Experiment Balance measure	Sex	RMSD	Age (Years)	PIQ	Handedness Scores	N (Treated) N (Control)
Gold standard, iteration-0						
SMD	0.28	0.21	0.24	0.23	0.30	129
p-value	0.04	0.10	0.05	0.06	0.01	129
15%, *method-1*, iter-0, surr = T						
SMD	0.30	0.22	0.18	0.12	0.17	129
p-value	0.03	0.08	0.15	0.34	0.17	129
50%, *method-1*, iter-0, surr = T						
SMD	0.40	0.22	0.15	0.18	0.18	129
p-value	0.00	0.08	0.24	0.17	0.20	129
100%, *method-1*, iter-0, surr = T						
SMD	0.28	0.21	0.24	0.26	0.31	129
p-value	0.04	0.10	0.05	0.06	0.05	129
Gold Standard, last-iter, surr = T						
SMD	0.15	0.11	0.09	0.10	0.13	106
p-value	0.37	0.44	0.49	0.48	0.34	106
15%, *method-1*, last-iter, surr = T						
SMD	0.13	0.12	0.14	0.11	0.00	40
p-value	0.77	0.59	0.52	0.62	0.99	40
50%, *method-1*, last-iter, surr = T						
SMD	0.14	0.00	0.15	0.02	0.05	77
p-value	0.53	1.00	0.36	0.88	0.76	77
100%, *method-1*, last-iter, surr = T						
SMD	0.15	0.11	0.08	0.09	0.03	106
p-value	0.37	0.44	0.56	0.54	0.88	106