Statistical Learning Methods for Electronic Health Record Data

by

Evan Lee Reynolds

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2019

Doctoral Committee:

Research Professor Mousumi Banerjee, Chair
Professor Thomas Braun
Professor Brian Callaghan
Professor Brisa Sánchez

Evan Lee Reynolds
evanlr@umich.edu
ORCID id: 0000-0002-0138-8436

To my wife and family.

ACKNOWLEDGEMENTS

There have been so many individuals that have supported me in my path towards my dissertation. This guidance, advice, support and love has shaped me into to who I am today. I would like express my genuine appreciation to these people.

First, I would like to express my sincere gratitude to my wonderful advisor, Dr. Mousumi Banerjee. Through her guidance, I have learned to become not only a statistician, but a researcher and scientist. Her passion and excitement for teaching, research and advancing human health through statistics is remarkable. I am so thankful to have had the opportunity to work with you over the last several years.

I would also like to thank Drs. Brisa Sanchez, Tom Braun and Brian Callaghan for serving on my dissertation committee. You have all provided me with valuable insight and support towards my dissertation.

During the last several years I have had the incredible opportunity to serve as a graduate student research assistant with Dr. Brian Callaghan. Your invaluable advice has sharpened my skills as an applied statistician and given me confidence to become a researcher. I am lucky to have worked with a collaborator that enjoys talking sports as much as research.

Additionally, I want to express my appreciation to the other collaborators I have worked on specific projects with during my time in graduate school: Drs. Michael Gaies, Brahamajee Nallamothu, James Burke and Cathy Van Poznak.

Being a member of STATCOM (Statistics in the Community) has been one of

my favorite parts of Graduate School. I would like to thank the STATCOM faculty advisors: Drs. Michael Elliott and Cathie Spino for the consistent support and guidance they have provided me. I would like to thank current and former STATCOM members that have helped grow the group into what it is today. Specifically I would like to thank Timothy NeCamp, Lauren Beriont and the rest of the leadership team.

I have been fortunate to be surrounded by a group of friends who have been some of my biggest supporters. The care and love you has all given me have allowed the years in graduate school to be some of the best of my life.

Most importantly, I would like to thank my family. Without their love, support and encouragement, I would not be the person I am today. I have been incredibly fortunate to have lived in the same city as my entire family during graduate school. I can honestly say, I am blessed to have two of the best parents in the world: Jean and Roger Reynolds. They have given and taught me more than I can ever put into words. They are my role models and number one supporters. I am also lucky to have incredible support from my sister and brother-in-law: Nina and Ryan Strang.

Lastly, I want to thank my wife, Riley. You has been my number one supporter from day one through the ups and downs. You are my best friend and love of my life. I cannot imagine completing this journey without you and am lucky to have you by my side no matter where life takes us.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

In the current era of electronic health records (EHR), use of data to make informed clinical decisions is at an all-time high. Although the collection, upkeep and accessibility of EHR data continues to grow, statistical methodology focused on aiding real-time clinical decision making is lacking. Improved decision making tools generally lead to improved patient outcomes and lower healthcare costs. In this dissertation, we propose three statistical learning methods to improve clinical decision making based on EHR data.

In the first chapter we propose a new classifier: SVM-CART, that combines features of Support Vector Machines (SVM) and Classification and Regression Trees (CART) to produce a flexible classifier that outperforms either method in terms of prediction accuracy and ease of use. The method is especially powerful in situations where the disease-exposure mechanisms may be different across subgroups of the population. Through simulation, under settings with high levels of interaction, the SVM-CART classifier resulted in significant prediction accuracy improvements. We illustrate our method to diagnose neuropathy using various components of the metabolic syndrome. In predicting neuropathy, SVM-CART outperformed CART in terms of prediction accuracy and provided improved interpretability compared to SVM.

In the second chapter, we develop regression tree and ensemble methods for multivariate outcomes. We propose two general approaches to develop multivariate regression trees by: (1) minimizing within-node homogeneity, and (2) maximizing between-node separation. Within-node homogeneity is measured using the average Mahalanobis distance and the determinant of the covariance matrix. For between-node separation, we propose using the Mahalanobis and Euclidean distances. The proposed multivariate regression trees are illustrated using two clinical datasets of neuropathy and pediatric cardiac surgery. In high variance scenarios or when the dimension of the outcome was large, the Mahalanobis distance split trees had the best prediction performance. The determinant split trees generally had a simple structure and the Euclidean distance metrics performed well in large sample settings. In both applications, the resulting multivariate trees improve usability and validity compared to predictions made using multiple univariate regression trees.

In the third chapter we develop a sequential method to make prediction using shallow (large-scale EHR) data in tandem with deep (health system specific) patient data. Specifically, we utilize machine learning based methods to first give prediction based on a large-scale EHR, then for a select group of patients, refine prediction based on the deep EHR data. We develop a novel framework that is time and cost-effective, for identifying patient subgroups that would most benefit from a second-stage prediction refinement. Final tandem prediction is obtained by combining predictions from both the first and second stage classifiers. We apply our tandem approach to predict extubation failure for pediatric patients that have undergone a critical cardiac operation using shallow data from a national registry and deep continuously streamed data captured in the intensive care unit. Using these two EHR data sources in tandem increased our ability to identify extubation failures in terms of the area under the ROC curve (AUC: 0.639) compared to using just the national registry (AUC:

0.607) or physiologic ICU data (AUC: 0.634) alone. Additionally, identifying a specific patient subgroup for second stage prediction refinement resulted in additional prediction improvement, as opposed to giving each patient a deep-data prediction (AUC: 0.682).

CHAPTER I

Introduction

In healthcare delivery settings, accurate and informed decisions can improve patient care and reduce unnecessary healthcare cost. For the past several years, the collection, upkeeping and accessibility of electronic health records (EHR) has grown, allowing clinicians to utilize a breadth of data for patient diagnosis and treatment [1,2]. Although the ability to use EHR in developing clinical decision support tools is at an all-time high, often the statistical methodology falls short in terms of developing tools that have clinical validity and applicability.

The ideal prediction tools for clinical decision support must satisfy a few criteria. First, they must be *practical*, easy to implement, and allow clinicians to make real-time predictions. Secondly, they must possess strong predictive attributes and yield *reliable* predictions for clinical use. Finally, resulting classifiers must balance the aforementioned predictive ability with *clinical validity* and *clinical interpretability.* Tools that lack *clinical validity and interpretability* are less likely to be trusted by clinicians and therefore: less likely to be implemented in practice.

In this dissertation, we propose three statistical learning methods to improve prediction in the clinical setting using EHR data. In Chapters 2 and 3, we develop methods that can be used to produce *clinically valid*, *practical* and *reliable* tools.

In Chapter 4, we develop an approach to leverage multiple data sources to improve the *reliability* when making prediction with EHR. The overarching goal of this dissertation is to develop statistical methods that aid in clinical decision making and subsequently improve patient care and outcomes.

Tree based methods such as Classification and Regression Trees (CART) are statistical learning tools that have become very popular in biomedical research [3-7]. Tree based methods have the ability to uncover hidden interactions in complex data scenarios, and often yield *practical* and easy to use tools for diagnostic and prognostic classification. However, due to the rectangular splits implemented in CART, there are scenarios in which tree based methods yield classifiers with poor predictive performance. In Chapter 2 of the dissertation, we propose a novel method (SVM-CART) that combines features of support vector machines (SVM) and CART to produce a more flexible classifier that has the potential to outperform either method in terms of accuracy (*reliability*) and simplicity. SVM-CART performs especially well when the disease-exposure mechanism is different across subgroups of the population. In such settings, SVM-CART results in improved prediction accuracy and *clinical interpretability.* Using extensive simulations, we demonstrate the predictive performance of the SVM-CART classifier under various clinical scenarios. Finally, we illustrate our method to diagnose neuropathy using various components of the metabolic syndrome.

Simultaneous prediction of multivariate outcomes is critical in many clinical applications. Multivariate prediction is needed when a diagnosis is based on multiple measures of the same intrinsic condition or when there are multiple outcomes from the same clinical domain. An example is in neurology, where in order for a clinician to give an assessment of neuropathy, they would have to take into account multiple measures of the disease (e.g. many nerve conduction measures). Current approaches to analyzing such outcomes rely on creating several univariate trees or creating multivariate trees that fail to account for the inherent correlation structure. These approaches may have several limitations. Specifically, as the dimension of the multivariate outcome grows, the trees lose *practical* applicability in the clinic setting. Additionally, univariate trees may have disparate sets of covariates, affecting *clinical interpretation*. In Chapter 3, we propose two general tree methods for multivariate outcomes. First, we grow regression trees by maximizing within-node homogeneity. Specifically, at each step of the splitting process, we use the determinant and the average Mahalanobis distance of the empirical covariance matrix as within-node impurity measures. Our second approach focuses on maximizing the between node separation in the daughter nodes resulting from a split. We measure between node separation using the Mahalanobis, Euclidean and standardized Euclidean distances. The proposed methods are assessed using an extensive simulation study. Finally, we illustrate our methods using data from two clinical studies. In the first application, we build a multivariate regression tree to predict 3 measures of nerve conduction that are used as a battery of tests to diagnose neuropathy. In the second application, we predict post-operative length of stay in 3 phases of hospitalization for pediatric patients who undergo a cardiac operation.

Large scale, multi-center, electronic health records (EHR) offer an incredible opportunity to assess the quality of health services and improve patient outcomes [8-15]. Examples of such EHR include hospital insurance claims data and multi-center clinical registries. Since these data are collected across multiple sites/institutions, the EHRs typically capture broad-range (shallow) data that may be lacking physiological and disease-specific clinical variables. Therefore, prediction models built on large scale EHR may only yield modest predictive attributes. The latter can be enhanced by taking advantage of deep data available in a single health center, e.g. after a patient undergoes an operation or treatment. In Chapter 4, we develop a tandem prediction approach, taking into account both shallow and deep EHR resources to improve prediction. First, we utilize a toolbox of machine learning (ML) classifiers to give an initial risk prediction based on the large scale EHR data. We then develop a novel framework that is time and cost-effective for identifying subgroups that would most benefit from a second stage prediction based on deep data. Final tandem prediction is obtained by combining predictions from both the first and second stage classifiers. We illustrate our proposed approach to predict extubation failure for pediatric patients following a critical cardiac operation.

CHAPTER II

SVM-CART for Disease Classification

Classification and regression trees (CART) and support vector machines (SVM) have become very popular statistical learning tools for analyzing complex data that often arise in biomedical research. While both CART and SVM serve as powerful classifiers in many clinical settings, there are some common scenarios in which each fails to meet the performance and interpretability needed for use as a clinical decision-making tool. In this paper, we propose a new classification method, SVM-CART, that combines features of SVM and CART to produce a more flexible classifier that has the potential to outperform either method in terms of interpretability and prediction accuracy. Furthermore, to enhance prediction accuracy we provide extensions of a single SVM-CART to an ensemble, and methods to extract a representative classifier from the SVM-CART ensemble. The goal is to produce a decision-making tool that can be used in the clinical setting, while still harnessing the stability and predictive improvements gained through developing the SVM-CART ensemble. An extensive simulation study is conducted to asses the performance of the methods in various settings. Finally, we illustrate our methods using a clinical neuropathy dataset.

2.1 Introduction

Statistical learning methods such as decision trees and support vector machines have become very popular tools for analyzing complex data that often arise in biomedical research [3-7,16-19]. Classification and Regression Trees (CART) are useful statistical learning tools because they allow for intuitive and simple disease classification by recursively partitioning the covariate space [3-7]. Support vector machines (SVMs) are non-probabilistic supervised learning procedures that create a multi-dimensional hyperplane to partition the covariate space into two groups allowing for classification [5,16-19].

While both CART and SVM serve as powerful classifiers in many clinical settings, there are some common scenarios in which both fail to meet the performance and interpretability needed for application as a decision-making tool. These scenarios often occur when there are different disease-exposure mechanisms in subgroups of the population. The following scenarios describe some pathological examples where SVM and CART fail to meet the above criteria.

2.1.1 Scenario 1: Disease Outcome, Patient Gender and two Continuous Exposure Variables

In Figure 2.1a, the exposure-disease mechanism is very different between males and females. The gender-outcome subgroups are represented by shape and the continuous exposure variables are in the x and y axis of the plot.

The dashed line in Figure 2.1a represents the split from a linear SVM and the solid line represents the single split from CART. The SVM splits the data down the middle of continuous covariate 1 and has a 46% misclassification rate. CART performs the same with a 46% misclassification rate. Visually, it is simple to classify the patients into disease and control groups, but both methods fail to perform this simple task.

2.1.2 Scenario 2: Disease Outcome, Patient Gender, Smoking Status and two Continuous Exposure Variables

In the second example, in addition to the gender groups and two continuous covariates, we have the additional binary covariate: smoking status. Figure 2.1b shows the CART and SVM classifiers: the solid lines represent the CART splits in the two-dimensional continuous exposure covariate space and the dashed line represent the hyperplane from the SVM classifier.

The CART splits perform slightly better this time with a misclassification rate of 34%. SVM still has a misclassification rate of 46%. We also see that the CART tree becomes quite complicated quickly, but in the end, still produces a relatively poor performing classifier.

Classification scenarios such as the two presented above provide motivation for our research. In this paper, we propose a new classification method that combines features of SVM and CART to allow a more flexible classifier that has the potential to outperform either method in terms of interpretability and prediction accuracy. Ultimately our goal is to develop a tool that can be used in the clinical setting for decision-making.

The literature on combination classifiers is somewhat sparse. Xu et al. (1992) described methods of combining classifiers to improve handwriting recognition. These authors propose a combination classifier that aggregates predictions across many different types of classifiers [20-22]. Our proposed method differs from Xu et al. in that we exploit specific aspects of each classifier in tandem to create a new single classifier [20-22]. While many different classifiers could be considered for use in tandem, the choice of CART and SVM was motivated by the clinical study in our context.

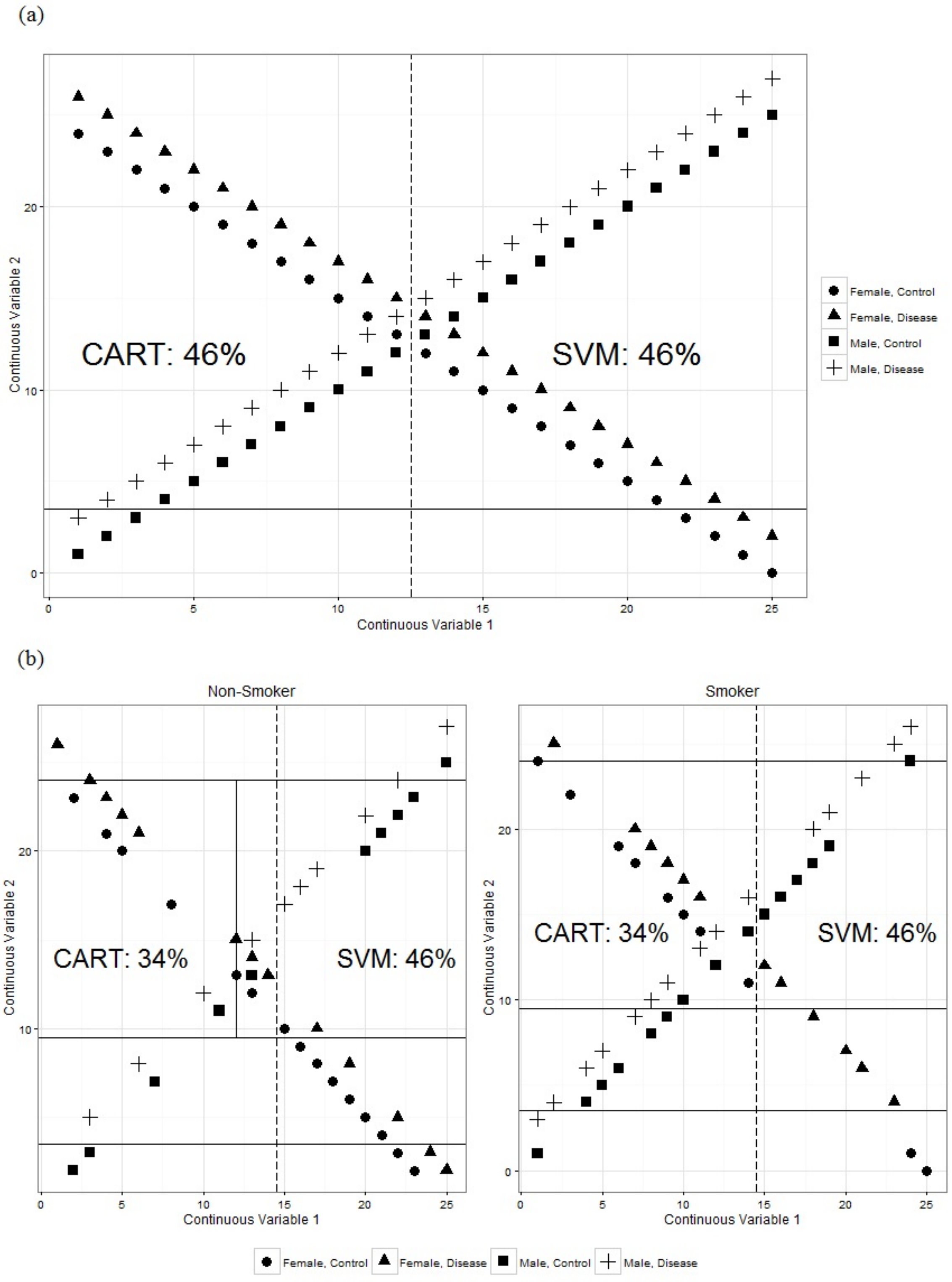

Figure 2.1: Results from SVM and CART for Simulated Scenarios

In neurology, the mechanistic pathway towards the disease polyneuropathy can be different amongst gender/glycemic subgroups of the population. CART was chosen as the first classifier because it offers a very natural way of subgrouping patients and SVM was specifically chosen since it is non-probabilistic and complements CART by overcoming issues with rectangular splits that often plague tree based methods. Additionally, because CART and SVM are two of the most well known non-parametric approaches in the clinical setting, the methods will be more approachable by clinicians who will ultimately use this method to make clinical decisions. The binary decision rule generated by CART is attractive to clinicians; This is how clinicians 'think' and it is therefore easy for them to bin patients in the fashion that CART works.

There is a growing literature for combining classification trees with parametric models, often implemented at the terminal nodes. Additionally, methods have been developed for growing trees to find treatment-subgroup interactions. Examples include GUIDE (Loh), CRUISE (Kim and Loh), LOTUS (Chan and Loh), MOB (Zeileis), STIMA (Dusseldorp), PALM (Seibold), PPTree (Lee) and Interaction Trees (Su) [23-30]. In certain scenarios, these methods take a significant step to improve prediction accuracy compared to a typical classification tree by overcoming issues with perpendicular splits, finding important interaction subgroups and applying parametric models for inference. Our proposed method differs from the earlier works in that we use a fully non-parametric approach combining two classifiers to capture likely different disease-exposure mechanisms amongst subgroups of the population. In our proposed method, we elicit clinical information for covariate inputs into the CART portion of the classifier, without having to evaluate every pairwise interaction. By focusing on a priori knowledge-driven interactions, our resulting classifier is more nuanced in its application. The overall goal of the proposed method is to develop a valid, interpretable, and easily usable tool for prediction in the clinical setting.

This paper is organized as follows: Section 2.2 describes the proposed methodology, SVM-CART. Section 2.3 describes ensemble methods for SVM-CART. In Section 2.4, we perform an extensive simulation to asses the prediction performance of our proposed SVM-CART classifier under various scenarios. Section 2.5 illustrates an application of our methodology to create a classifier for neuropathy. Lastly, concluding remarks and discussion are provided in Section 2.6.

2.2 Methodology

2.2.1 Classification and Regression Trees

First, we introduce some terminology that will be used to describe a classification tree. A classification tree T has multiple nodes where observations are passed down the tree. The tree starts with a root node at the top and continues to be recursively split to yield the terminal nodes at which stage no further split is prescribed. The intermediate nodes in the tree between the root node and terminal nodes are referred to as internal nodes. We specifically denote the set of terminal nodes as $\tilde{T}$ and the number of terminal nodes is denoted as $|\tilde{T}|$. In CART, a class prediction is given to each observation based on which terminal node it falls into.

In growing a tree, the natural question that arises is how and why a parent node is split into daughter nodes. Trees use binary splits, phrased in terms of the covariates, that partition the covariate space recursively. Each split depends upon the value of a single covariate. The partitioning is intended to increase within-node homogeneity. Goodness of a split must therefore weigh the homogeneities in the two daughter nodes. The extent of node homogeneity is measured using an 'impurity' function.

Potential splits for each of the covariates are evaluated, and the covariate and split value resulting in the greatest reduction in impurity is chosen.

The impurity at proposed node h is denoted as $i(h)$ and the probability that a subject falls into node h is $P(h)$, where $P(h)$ is estimated from the sample proportions in the training data. Specifically, for a split $s \epsilon S$ at node h, the left and right daughter nodes are denoted as h_L and h_R respectively. Where S is the set of all possible splits. The reduction in impurity is calculated as follows: $\Delta I(s, h) = i(h) - P(h_L)i(h_L) - P(h_R)i(h_R)$. For binary outcomes, $i(h)$ is measured in terms of entropy or Gini impurity [3,4]. The splitting rule that maximizes $\Delta I(s, h)$ over the set S of all possible splits is chosen as the best splitter for node h.

2.2.2 Support Vector Machines

Support Vector Machines create separating hyperplanes to give class-level predictions. The SVM hyperplane takes a small or large number of covariates to create a hyperplane that can be used to classify patients into outcome groups [16-19].

Let y_i be the binary outcome for patient i, and x_i the $p \times 1$ vector of covariates for the ith patient. Then we denote $\boldsymbol{x}$ as the $p \times n$ matrix of continuous covariates.

SVMs create a hyperplane of the form:

$$\mathbb{H} = \boldsymbol{x} : \boldsymbol{w}'\boldsymbol{x} + b = \mathbf{0}$$

where $\boldsymbol{w} \epsilon \mathbb{R}^p$ and $b \epsilon R$ are the set of optimal weights corresponding to each continuous covariate that construct the hyperplane. SVMs create the separating hyperplane by maximizing the margin between the nearest p-dimensional data points on each side of the hyperplane. In clinical data, we often do not have linearly separable data. To deal with non-separable data we use the optimal soft-margin hyperplane which introduces slack variables to penalize classification errors based on some predetermined weights

[5,16,17]. The optimal soft-margin hyperplane is found by minimizing the following objective function:

$$\min_{\boldsymbol{w},b,\psi} \tfrac{1}{2}||\boldsymbol{w}||^2 + \tfrac{C}{n}\textstyle\sum_{i=1}^{n}\psi_i$$

$$\text{subject to: } y_i(\boldsymbol{w'x}_i + b) \geq 1 - \psi_i\forall i \text{ and } \psi_i \geq 0\forall i$$

The solution to the above minimization problem represents the optimal hyperplane. C represents the cost penalty assigned for a misclassified subject and ψ_i are the slack variables that allow for this misclassification. Using Lagrangian multipliers, $\alpha_i \geq 0$, the optimal hyperplane is obtained as [5,16,17]:

$$\hat{\boldsymbol{w}} = \textstyle\sum_{i=1}^{n}\hat{\boldsymbol{\alpha}}_i\boldsymbol{y}_i\boldsymbol{x}_i$$

$$\hat{b} = y_i - \hat{\boldsymbol{w}}'\boldsymbol{x}_i$$

2.2.3 SVM-CART

In the traditional CART method, all covariates of interest are considered for tree building. For SVM-CART, we propose to employ CART to split based on only the categorical covariates. The terminal nodes from CART are used to pass along patients to subgroups. Support Vector Machines are developed on each of the subgroups using the continuous covariates, thereby generating $|\tilde{T}|$ separating hyperplanes.

The optimal hyperplane solution can now be written within the SVM-CART framework as follows:

$$\hat{\boldsymbol{w}}_{\tilde{T}_j} = \textstyle\sum_{i=1}^{n_j}\hat{\boldsymbol{\alpha}}_i\boldsymbol{y}_i\boldsymbol{x}_i$$

$$\hat{b} = y_i - \hat{\boldsymbol{w}}_{\tilde{T}_j}{}'\boldsymbol{x}_i$$

where we have a unique solution for each of the $\tilde{T}_j$ terminal nodes created from CART. For each patient i in terminal node $\tilde{T}_j$, the classifier evaluates hyperplane

j to determine classification: if $\hat{\boldsymbol{w}}_{\tilde{T}_j}{}'\boldsymbol{x}_i + b > 0$ we assign patient i to group 1. Alternatively, if $\hat{\boldsymbol{w}}_{\tilde{T}_j}{}'\boldsymbol{x}_i + b < 0$ we assign patient i to group 0. With a strong subgroup selection by CART, the terminal nodes may not be well mixed. If any of the terminal nodes are pure and contain only patients from one outcome group, no SVM is generated. For future prediction, patients that fall into these nodes are given the same predicted outcome.

Results from the single SVM-CART classifier offer a clinician friendly and easy to use tool by first distributing patients into different subgroups and then assigning an outcome class prediction based on an array of continuous data features.

2.2.4 Hyperparameter Tuning

Class Weights for CART

The proposed SVM-CART allows for implementation of class weights within the CART part of the method. For a rare disease, a user can put higher weight to the disease cases to assist in the most useful classification. Using inverse proportions of the cases and controls is a simple way to include weight for the CART part of the SVM-CART classifier.

SVM Cost Parameter

The cost parameter C allows the user to control how costly misclassification is in the creation of the SVM hyperplane. Large values of C generally result in a smaller and harder margin hyperplane and conversely smaller cost results in a larger, softer margin hyperplane. The cost parameter is chosen through a data-driven search. We select the cost parameter by examining the test error using a cross validation procedure or by examining the out-of-bag error estimates from a bootstrapped sample. In either case, a reasonable grid search over the range 10^{-5} to 10^{4} allows the user to select the proper cost parameter for building the SVM within the SVM-CART procedure.

Class Weights for SVM

There exist many scenarios in which we want to assign different misclassification costs to each outcome group. We propose assigning an outcome-class specific cost parameter by assigning weights to each of the outcome groups. Assigning weights to the two outcome classes allows us to re-write the hyperplane minimization problem as:

$$\min_{\boldsymbol{w}_{\tilde{T}_j},b,\psi} \tfrac{1}{2}||\boldsymbol{w}_{\tilde{T}_j}||^2 + \tfrac{C_+}{n_{j+}} \textstyle\sum_{i=1}^{n_{j+}} \psi_{i+} \tfrac{C_-}{n_{j-}} \textstyle\sum_{i=1}^{n_{j-}} \psi_i$$

where $C_+ = r_+ * C$ and $C_- = r_- * C$. In other words, to assign a higher cost to the misclassification of the disease outcomes, we do so by using the weights r_- and r_+.

In SVM-CART, the inverse proportion of the cases and controls in each of CART's terminal nodes is chosen to be that node's SVM class weights. Then, a data driven search is performed to determine a weight multiplier, m, for the inverse proportion weights (r_-, r_+). The final chosen weights for the SVM-CART are: $C_+ = m * r_+ * C$ and $C_- = r_- * C$. The multiplier allows more flexibility in the class weights for the support vector machines in each terminal node.

2.2.5 SVM-CART as a Clinical Tool

Clinical applicability is an important goal of our proposed method. For certain data applications, our methodology is expected to provide a better performing yet simpler classifier due to the ability to create non-rectangular splits.

Using the SVM-CART classifier is very straightforward. First a patient is passed down the CART tree based on his/her characteristics until they are placed in one of the CART terminal nodes. Then, a linear equation is evaluated to classify patients into the final terminal classification nodes. For a SVM-CART with k continuous covariates, we evaluate an equation of the form:

$$I_{\tilde{T}_j}(x_1 * w_1 + ... + x_k * w_k + b > 0)$$

where the indicator function $I_{\tilde{T}_j}$ assigns patients to outcome 1 if $x_1 * w_1 + . + x_k * w_k + b > 0$ in terminal node j and to outcome 0 otherwise. There is a terminal node classification equation for each of the subgroup terminal nodes $\tilde{T}_j$ created by the initial CART. The exception would be the scenario where terminal node $\tilde{T}_j$ is pure; in that case, we have a class prediction without a SVM classifier.

2.2.6 SVM-CART for Simulated Scenarios

For both simulated examples in Section 2.1 , SVM-CART yields a perfect classifier. The CART portion in the first scenario splits on gender allowing us to create a perfect linear SVM based on the two continuous covariates. In Scenario two, we first split on smoking status and then by gender. Each of the four node groups from CART then yields itself to a linear SVM classifier that produces perfect classification. The results from the SVM-CART classifier in Scenario 2 are displayed in Figure 2.2.

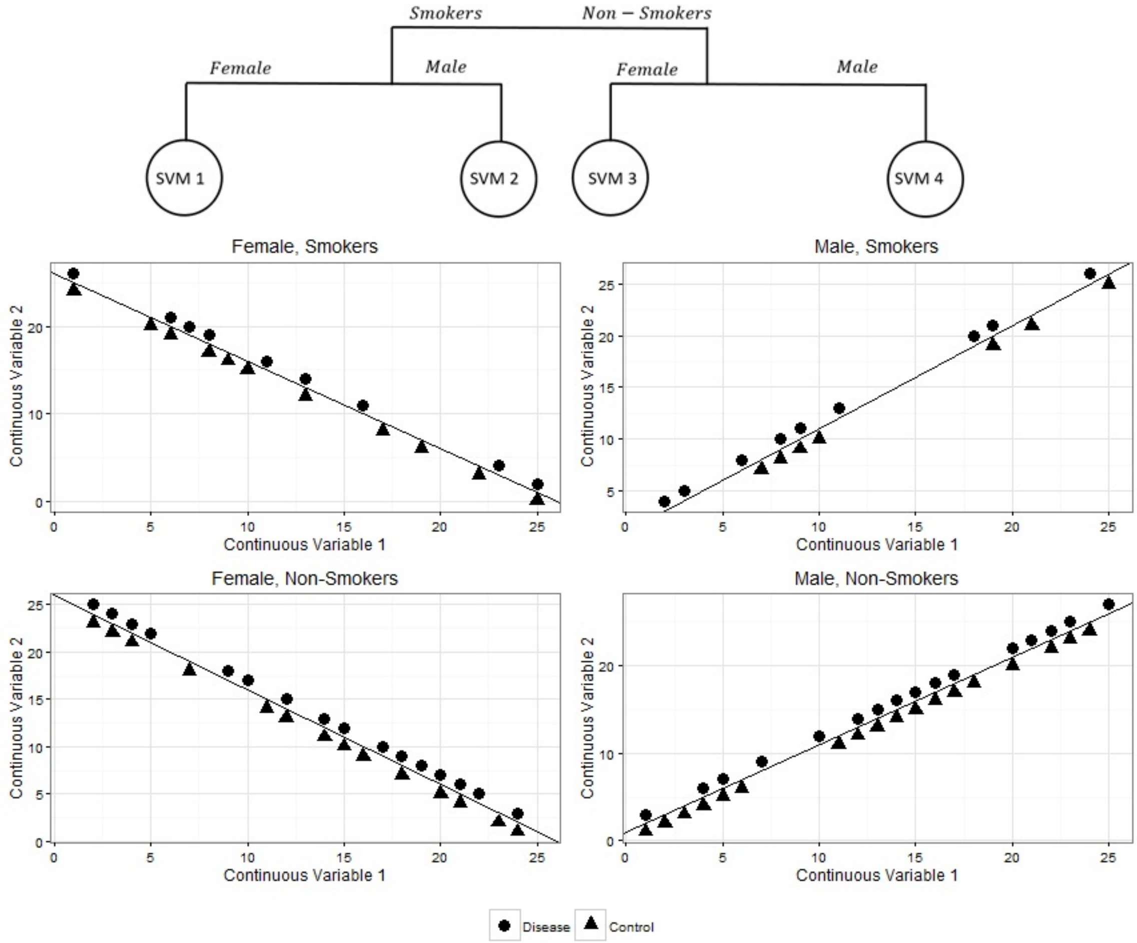

Figure 2.2: SVM-CART for Simulated Scenarios

2.3 SVM-CART Ensemble

2.3.1 Creating the SVM-CART Ensemble

Ensemble methods have become very popular in tree based applications, allowing for creation of more stable trees that often lead to improved predictions [31-34]. An ensemble method proposed in 1994 by Breiman involved bootstrap aggregating or Bagging [31]. The premise of this method is to generate many bootstrap samples of the data and create an individual classification tree from each of the bootstrap samples. A final classification is determined by voting across all trees in the ensemble [31-34].

We develop an SVM-CART ensemble to enhance prediction accuracy. Specifically, we generate b bootstrap samples by sampling uniformly n observations with replacement from the entire data of size n. On each of the b samples, we generate a SVM-CART classifier. For each patient, the most common class prediction across the b classifiers is the predicted outcome.

To obtain honest estimates of prediction accuracy, we derive error rates based on the out-of-bag sample. For the jth SVM-CART classifier, the out-of-bag sample consists of patients that were not included in the specific bootstrap sample used to create the jth classifier. For each SVM-CART classifier in the ensemble, we make a class prediction for only the patients in the out-of-bag sample. Finally, the out-of-bag prediction for each patient is the most common predicted class across all $c \leq b$ samples for which that patient was out-of-bag. The out-of-bag error rate for a single bootstrap sample is calculated as the percentage of misclassified patients in the out-of-bag sample. The out-of-bag error rate for the ensemble is calculated as the percentage of misclassified patients based on the out-of-bag prediction.

2.3.2 Selecting the Most Representative Classifier from the Ensemble

Bagging the SVM-CART improves stability and prediction ability; however, we lose the interpretability of a single classifier. This is a significant loss from a clinical standpoint because ultimately, we want to produce a decision-making tool that can be used in the clinic setting.

This section describes how to harness the stability and predictive improvements gained through the SVM-CART ensemble while still producing a clinician friendly tool. To obtain a usable prediction tool for the clinical setting, we attempt to extract the single most representative classifier from the ensemble. We think of each SVM-CART classifier as a point in a high-dimensional space and cluster the classifiers according to some measure of proximity. Note that this space is much more complex than the Euclidean space, and distances between the classifiers can be quantified in several ways. Any classifier in the above space can be identified by a finite set of parameters, and these parameters could include the partition of the covariate space and the predictions from each terminal classification node. We propose two such metrics that are extensions of Banerjee et al. [35].

The first metric focuses on prediction proximity (i.e. similarity). Two classifiers are similar if the predictions from them are the same for all subjects. Without loss of generality, the distance between SVM-CART 1 and SVM-CART 2 is measured using the metric:

$$d_1(T_1, T_2) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_{1i} - \hat{y}_{2i})^2$$

where $\hat{y_{1i}}$ is the class prediction for patient i from SVM-CART classifier 1.

The second metric focuses on how closely (i.e. similarly) the covariate space is partitioned by two classifiers. Classifiers that are similar will place the same subjects together in a terminal classification node and separate the same subjects in different terminal classification nodes (i.e. SVM-CART 1 and 2 are similar if two patients that are placed in the same terminal node by classifier 1 are also placed in the same terminal node by classifier 2). Towards that end, we define a metric that captures how subjects are clustered in the terminal classification nodes from SVM-CART. For all $\binom{n}{2}$ pairs of subjects, let $I(T_1(i, j))$ be the indicator that patient i and patient j are in the same terminal classification node from SVM-CART 1.

The distance metric is then defined as:

$$d_2(T_1, T_2) = \frac{\sum_{i>j}\sum_{j}|I_{T_1}(i,j) - I_{T_2}(i,j)|}{\binom{n}{2}}$$

The factor $\binom{n}{2}$ scales the metric to the range $(0, 1)$ such that 0 indicates perfect agreement. A pair of subjects contributes a positive amount to d_2 if and only if one SVM-CART classifier places the subjects together and the other SVM-CART classifier places them apart. Thus d_2 is 0 if the two classifiers partition the covariate space in exactly the same way.

The score $D(T)$ for a SVM-CART classifier T is computed by averaging the individual distance metrics between the classifier T and all other classifiers in the ensemble. This is the average distance between T and all other classifiers in the ensemble. So, a low score for a classifier indicates its similarity to all other classifiers in the ensemble. The score $D(T)$ is computed for each of the distance metrics (i.e. d_1, d_2 giving rise to scores $D_1(T)$ and $D_2(T)$) and the representative classifiers in the ensemble are chosen based on the smallest $D(T)$ values.

2.4 Simulation Study

We compare performance of the SVM-CART classifier using several criteria over a variety of simulation scenarios. We generated a binary outcome y based on a logistic regression model with two categorical covariates: $x_1 \sim bernoulli(0.3)$ and $x_2 \sim multinomial(0.45, 0.15, 0.4)$, and four continuous covariates: $x_3 \sim uniform(0, 5)$, $x_4 \sim N(7, 5)$, $x_5 \sim weibull(0.5)$ and $x_6 \sim N(1, 5)$.

Several tuning parameters were used to assess prediction performance under the various simulation scenarios. First, we varied the sample size: $n = \{100, 500, 1000\}$. Next, we assessed the impact that different levels of continuous-categorical covariate interactions may have on prediction performance. Specifically, we assess prediction performance under small to large interaction effect sizes as well as in the absence of any interaction between the categorical and continuous covariates. Lastly, we examine how varying degrees of the main effects of the categorical covariates influence prediction performance.

Specifically, we generate data using the following underlying model: $logit(\boldsymbol{p}) = \mathbf{X}\boldsymbol{\alpha} + \mathbf{W}\boldsymbol{\beta}$, where $p = P(y = 1|\mathbf{X}, \mathbf{W})$, where $\boldsymbol{X}$ is the design matrix for the main effects and $\mathbf{W}$ is the matrix of interactions. The fixed $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values are listed in Table 2.1. These give rise to varying main and interaction effects as described above. The binary outcome is generated as $\boldsymbol{y} \sim bernoulli(\boldsymbol{p})$.

In a separate simulation, we generated data from a true underlying SVM-CART type structure. In this set-up, the tree first splits patients by x_1. Patients with $x_1 = 1$ were further split by x_2 (1 vs. 2,3). For patients with $x_1 = 0$, those with $x_2 = 3$ were split from $x_2 = 1$ or $x_2 = 2$. For each of these four terminal nodes created by the true underlying tree, we generate data from node-specific logistic regression models. The data generating structure is displayed in Figure 2.3. We examine scenarios where the β coefficients were different across the four terminal nodes (Figure 2.3a) and similar across the four terminal nodes (Figure 2.3b).

Table 2.1: Beta Coefficients for the Data Generated from Logistic Regression Models Under Various Scenarios

		Interaction Effect							
		None		Low		Moderate		High	
		Main Effects of Categorical Covariates							
Beta Values	Covariate	High	Low	High	Low	High	Low	High	Low
α_0	Intercept*	α_0	α_0	α_0	α_0	α_0	α_0	α_0	α_0
α_1	x_1	4.7	2.7	4.7	2.7	4.7	2.8	4.7	2.7
α_2	x_{2_1}	8.05	4.05	8.05	4.05	8.05	-4.05	8.05	4.05
α_3	x_{2_2}	-8.35	-4.35	-8.36	-4.36	-8.05	-4.36	-8.36	-4.36
α_4	x_3	-1	-1	-0.7184	-0.7184	-1	-1	8	8
α_5	x_4	-.5	-.5	0.317	0.317	0.5	0.5	2	2
α_6	x_5	2	2	0.215	0.215	2	2	5	5
α_7	x_6	4	4	0.695	0.695	4	4	-7	-7
β_1	$x_1 * x_{2_1}$	0	0	-2.2	-2.2	-2.2	-2.2	-2.2	-2.2
β_2	$x_1 * x_{2_2}$	0	0	-2.9	-2.9	-2.9	-2.9	-2.9	-2.9
β_3	$x_3 * x_1$	0	0	0.4934	0.4934	6	6	-9	-9
β_4	$x_3 * x_{2_1}$	0	0	1.5764	1.5764	-4	-4	-16	-16
β_5	$x_3 * x_{2_2}$	0	0	0.6203	0.6203	-2	-2	-4	-4
β_6	$x_3 * x_1 * x_{2_1}$	0	0	-1.21143	-1.21143	0	0	24	24
β_7	$x_3 * x_1 * x_{2_2}$	0	0	-1.1303	-1.1303	0	0	14	14
β_8	$x_4 * x_1$	0	0	0.229	0.229	-2	-2	-4	-4
β_9	$x_4 * x_{2_1}$	0	0	-0.591	-0.591	1	1	-1	-1
β_{10}	$x_4 * x_{2_2}$	0	0	-0.215	-0.215	0.5	0.5	-1	-1
β_{11}	$x_4 * x_1 * x_{2_1}$	0	0	-0.909	-0.909	0	0	1	1
β_{12}	$x_4 * x_1 * x_{2_2}$	0	0	0.0572	0.0572	0	0	1	1
β_{13}	$x_5 * x_1$	0	0	0.772	0.772	-3	-3	-8	-8
β_{14}	$x_5 * x_{2_1}$	0	0	-0.318	-0.318	1	1	-4	-4
β_{15}	$x_5 * x_{2_2}$	0	0	-1.189	-1.189	-5	-5	-5	-5
β_{16}	$x_5 * x_1 * x_{2_1}$	0	0	-0.658	-0.658	-2	-2	9	9
β_{17}	$x_5 * x_1 * x_{2_2}$	0	0	0.5689	0.5689	7	7	14	14
β_{18}	$x_6 * x_1$	0	0	-1.423	-1.423	-5	-5	11	11
β_{19}	$x_6 * x_{2_1}$	0	0	-0.15	-0.15	-3	-3	15	15
β_{20}	$x_6 * x_{2_2}$	0	0	-0.895	-0.895	-8	-8	10	10
β_{21}	$x_6 * x_1 * x_{2_1}$	0	0	0.974	0.974	1	1	-14	-14
β_{22}	$x_6 * x_1 * x_{2_2}$	0	0	1.652	1.652	12	12	-19	-19

*Where α_0 centers the data at 0 to ensure a well mixed sample

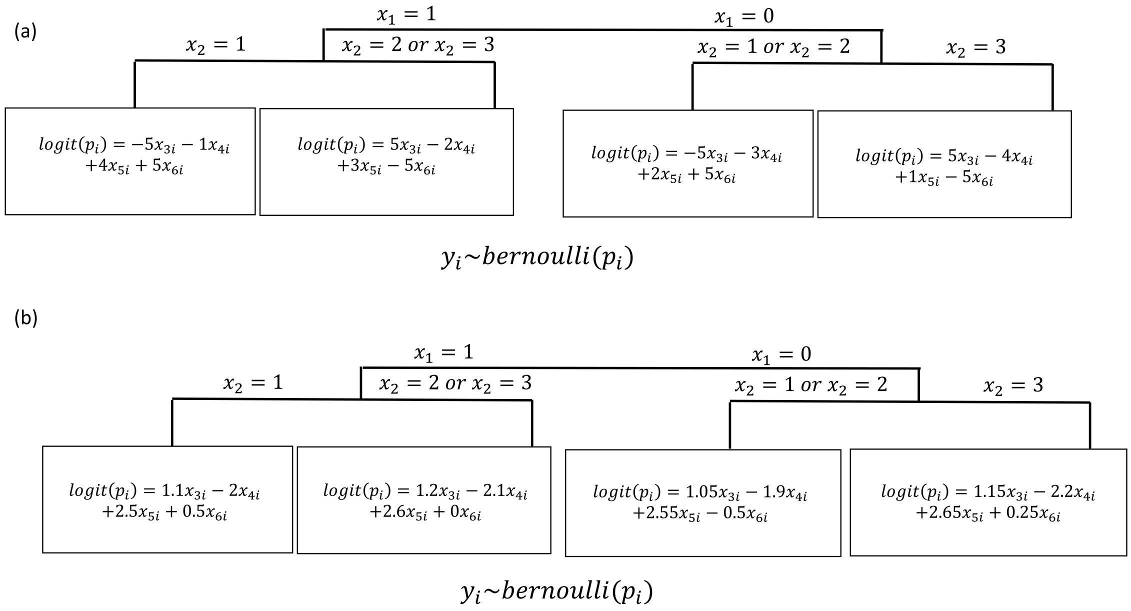
(a)
$x_1 = 1$
$x_1 = 0$
$x_2 = 1$
$x_2 = 2 \; or \; x_2 = 3$
$x_2 = 1 \; or \; x_2 = 2$
$x_2 = 3$
$logit(p_i) = -5x_{3i} - 1x_{4i} + 4x_{5i} + 5x_{6i}$
$logit(p_i) = 5x_{3i} - 2x_{4i} + 3x_{5i} - 5x_{6i}$
$logit(p_i) = -5x_{3i} - 3x_{4i} + 2x_{5i} + 5x_{6i}$
$logit(p_i) = 5x_{3i} - 4x_{4i} + 1x_{5i} - 5x_{6i}$
$y_i \sim bernoulli(p_i)$
(b)
$x_1 = 1$
$x_1 = 0$
$x_2 = 1$
$x_2 = 2 \; or \; x_2 = 3$
$x_2 = 1 \; or \; x_2 = 2$
$x_2 = 3$
$logit(p_i) = 1.1x_{3i} - 2x_{4i} + 2.5x_{5i} + 0.5x_{6i}$
$logit(p_i) = 1.2x_{3i} - 2.1x_{4i} + 2.6x_{5i} + 0x_{6i}$
$logit(p_i) = 1.05x_{3i} - 1.9x_{4i} + 2.55x_{5i} - 0.5x_{6i}$
$logit(p_i) = 1.15x_{3i} - 2.2x_{4i} + 2.65x_{5i} + 0.25x_{6i}$
$y_i \sim bernoulli(p_i)$

Figure 2.3: Data Simulated from Underlying SVM-CART Like Structure

For each of the above scenarios, we generated 1,000 simulated datasets. We split each dataset into a training and testing set by randomly assigning 70% of the sample for training and 30% for testing. To assess predictive performance, we calculated overall prediction accuracy (ACC), sensitivity (TPR), specificity (TNR), positive predictive value (PPV) and negative predictive value (NPV) based on the testing set. Lastly, we obtained average size of the classifiers based on the number of terminal nodes for CART, and the number of non-orthogonal dimensions of the hyperplane for SVM. For SVM-CART, we obtained both the number of terminal nodes created by the CART part of the classifier, and total number of SVM dimensions created for each of the CART terminal nodes.

2.4.1 Simulation Results

Simulation results for data generated from logistic regression models (Table 2.1 coefficients) are displayed in Table 2.2. As sample size increases, each of the three methods show improvement in prediction performance. CART and SVM-CART demonstrate significant prediction gains (SVM-CART: 6.9% and CART: 8.9% average ACC increase from $n = 100$ to $n = 1000$) while SVM has only modest gains (3.9% average ACC increase from $n = 100$ to $n = 1000$). Additionally, the largest improvements in prediction performance for SVM-CART occurred when sample sizes increased from $n = 100$ to $n = 500$, with only minor improvements occurring when sample size increased from $n = 500$ to $n = 1000$. SVM-CART and CART classifiers also increased in size as n increased.

SVM-CART generally had better prediction performance when the main effects of the categorical covariates were large. When the main effects of the categorical covariates were large, the CART part of the SVM-CART classifier builds slightly larger trees. SVM performed similarly in the scenarios with high/low main effects of the categorical covariates and CART had somewhat modest improvement in the setting with high main effects.

In the presence of interactions, the SVM-CART classifier outperforms SVM or CART alone. The prediction gains increase as the interaction effect sizes increase. When there were no interactions, SVM-CART performs similar to CART but worse than SVM alone in terms of prediction ability.

When there are distinctly different disease-exposure mechanisms in different subgroups of the population (Figure 2.3 setting), SVM-CART demonstrates the best prediction performance. This was consistently true across all sample sizes when the the disease-exposure effects varied substantially between the 4 subgroups (generated from Figure 2.3a), where SVM-CART outperformed CART or SVM alone in terms of ACC, PPV, NPV, TPR and TNR. When the continuous disease-exposure effects did not vary much between the 4 subgroups (generated from Figure 2.3b): SVM-CART still performed better than CART but almost identically to SVM alone.

In conclusion, there are clinical scenarios in which SVM or CART alone may have low predictive performance. These typically arise when there are large and complex interactions that exist in the data, specifically, when there are different disease-continuous exposure mechanisms amongst subgroups of the population. In simulations, we observed that as these interaction effects increase, SVM-CART may have modest to substantial prediction gains compared to SVM or CART alone.

While prediction performance is an important aspect to consider when assessing the performance of SVM-CART, with the ultimate goal of aiding in clinical decision support: interpretability and clinical validity are also important considerations. In scenarios with complex interactions, SVM-CART may provide enhanced interpretability compared to SVM or CART alone. The improved interpretability is demonstrated in our application to build a classifier for polyneuropathy in the following section.

2.5 Polyneuropathy Classification in an Obese Cohort

2.5.1 Data Collection and Background Information

Polyneuropathy is a painful condition affecting 2-7% of the adult population [36,37]. The most common etiology of the disease is diabetes. However, it is hypothesized that other components of metabolic syndrome can play a role in the etiology of polyneuropathy [15,38,39]. In this section, we take an in-depth look at the classification of neuropathy using patient measures from the metabolic syndrome.

Data were collected from obese patients recruited to the University of Michigan Investigational Weight Management Cohort. There were 115 patients recruited between November 2010 and December 2014. Inclusion criteria included age 18 years or older and a body mass index of at least (BMI) $35kg/m^2$ or $32kg/m^2$ if they had one or more medical conditions in addition to obesity.

Table 2.2: Results Based on Data Generated from a Logistic Regression Model

			Main Effects of the Categorical Covariates											
			Low						High					
Interaction Effect	Sample Size (N)	Classifier	ACC	PPV	NPV	TPR	TNR	Size	ACC	PPV	NPV	TPR	TNR	Size
High	100	CART	0.72	0.69	0.74	0.70	0.73	3.5	0.70	0.70	0.69	0.70	0.70	3.6
	100	SVM	0.75	0.75	0.74	0.72	0.77	7.0	0.75	0.76	0.74	0.75	0.76	7.0
	100	SVM-CART	0.75	0.76	0.74	0.71	0.79	11.6,2.9	0.75	0.77	0.73	0.72	0.77	12.1,3.0
	500	CART	0.82	0.83	0.81	0.80	0.84	4.0	0.82	0.83	0.82	0.82	0.83	10.5
	500	SVM	0.78	0.79	0.78	0.76	0.81	7	0.80	0.79	0.81	0.81	0.80	7.0
	500	SVM-CART	0.87	0.89	0.86	0.84	0.90	16.1,4.0	0.82	0.85	0.79	0.78	0.86	17.0,4.3
	1000	CART	0.84	0.85	0.83	0.82	0.86	13.0	0.85	0.85	0.85	0.85	0.85	13.1
	1000	SVM	0.84	0.84	0.83	0.81	0.86	7.0	0.79	0.80	0.78	0.78	0.80	7.0
	1000	SVM-CART	0.90	0.92	0.88	0.87	0.92	16.6,4.2	0.82	0.85	0.80	0.78	0.86	18.4,4.6
Moderate	100	CART	0.72	0.71	0.73	0.65	0.78	3.3	0.73	0.81	0.63	0.74	0.72	3.3
	100	SVM	0.82	0.80	0.83	0.79	0.84	7.0	0.82	0.85	0.80	0.84	0.80	7.0
	100	SVM-CART	0.80	0.78	0.81	0.77	0.82	9.4,2.4	0.82	0.85	0.77	0.83	0.80	8.7,2.1
	500	CART	0.80	0.79	0.80	0.75	0.84	9.8	0.80	0.85	0.74	0.81	0.80	9.6
	500	SVM	0.84	0.83	0.85	0.82	0.86	7.0	0.84	0.87	0.80	0.86	0.82	7.0
	500	SVM-CART	0.84	0.84	0.84	0.81	0.87	11.9,3.0	0.86	0.89	0.82	0.87	0.85	10.0,2.5
	1000	CART	0.82	0.82	0.82	0.77	0.86	13.2	0.83	0.88	0.76	0.83	0.83	13.4
	1000	SVM	0.84	0.83	0.86	0.83	0.86	7.0	0.85	0.87	0.82	0.86	0.84	7.0
	1000	SVM-CART	0.86	0.85	0.86	0.83	0.88	12.0,3.0	0.87	0.90	0.84	0.88	0.86	10.1,2.5
Low	100	CART	0.69	0.69	0.70	0.70	0.69	3.8	0.68	0.68	0.67	0.69	0.66	3.8
	100	SVM	0.79	0.79	0.79	0.79	0.79	7.0	0.78	0.78	0.78	0.79	0.77	7.0
	100	SVM-CART	0.61	0.60	0.61	0.61	0.60	11.5,3.0	0.7	0.7	0.7	0.72	0.68	10.7,3.2
	500	CART	0.81	0.81	0.82	0.82	0.81	11.4	0.81	0.81	0.81	0.82	0.8	10.8
	500	SVM	0.81	0.81	0.82	0.82	0.81	7.0	0.81	0.8	0.81	0.83	0.79	7.0
	500	SVM-CART	0.73	0.74	0.72	0.70	0.75	14.9,4.0	0.71	0.7	0.72	0.76	0.66	15.2,3.9
	1000	CART	0.84	0.84	0.85	0.85	0.84	14.1	0.84	0.83	0.84	0.85	0.82	12.8
	1000	SVM	0.83	0.83	0.82	0.82	0.83	7.0	0.81	0.8	0.82	0.83	0.79	7.0
	1000	SVM-CART	0.74	0.74	0.73	0.72	0.75	15.7,4.0	0.71	0.66	0.77	0.81	0.61	16.0,4.0
None	100	CART	0.85	0.85	0.84	0.86	0.83	2.4	0.88	0.88	0.88	0.87	0.9	2.3
	100	SVM	0.94	0.94	0.94	0.94	0.93	7.0	0.94	0.93	0.94	0.94	0.93	7.0
	100	SVM-CART	0.85	0.88	0.83	0.84	0.87	9.8,2.5	0.9	0.9	0.9	0.88	0.91	9.0,2.3
	500	CART	0.90	0.89	0.90	0.91	0.87	6.8	0.91	0.9	0.91	0.9	0.92	5.5
	500	SVM	0.97	0.97	0.97	0.97	0.96	7.0	0.97	0.96	0.97	0.97	0.97	7.0
	500	SVM-CART	0.87	0.90	0.84	0.85	0.90	11.5,2.9	0.94	0.94	0.95	0.94	0.95	7.6,1.9
	1000	CART	0.91	0.91	0.91	0.92	0.89	8.6	0.92	0.92	0.92	0.9	0.93	6.9
	1000	SVM	0.97	0.97	0.97	0.97	0.97	7.0	0.97	0.97	0.97	0.97	0.97	7.0
	1000	SVM-CART	0.87	0.90	0.84	0.85	0.90	11.5,2.9	0.95	0.95	0.95	0.94	0.95	7.1,1.8

Table 2.3: Results Based on Data Generated from an Underlying SVM-CART Like Structure

		Disease-Exposure Effects											
		Vary Across Subgroups						Similar Across Subgroups					
Sample Size (N)		ACC	PPV	NPV	TPR	TNR	Size	ACC	PPV	NPV	TPR	TNR	Size
100	CART	0.83	0.81	0.85	0.75	0.89	2.8	0.64	0.64	0.65	0.61	0.67	4.2
100	SVM	0.89	0.86	0.91	0.86	0.91	7.0	0.69	0.68	0.70	0.69	0.70	7.0
100	SVM-CART	0.88	0.85	0.90	0.84	0.91	6.6,1.7	0.75	0.74	0.75	0.73	0.76	10.8,2.7
500	CART	0.88	0.88	0.88	0.81	0.93	6.8	0.81	0.81	0.80	0.78	0.83	10.9
500	SVM	0.92	0.90	0.93	0.89	0.94	7.0	0.73	0.72	0.74	0.73	0.73	7.0
500	SVM-CART	0.91	0.89	0.92	0.88	0.93	4.9,1.2	0.84	0.86	0.83	0.80	0.88	12.0,3.0
1000	CART	0.89	0.90	0.89	0.82	0.94	9.1	0.84	0.86	0.83	0.81	0.87	13.3
1000	SVM	0.92	0.90	0.93	0.89	0.94	7.0	0.74	0.72	0.75	0.74	0.74	7.0
1000	SVM-CART	0.91	0.90	0.92	0.88	0.93	4.3,1.1	0.87	0.91	0.84	0.82	0.92	12.1,3.0

Five components make up the metabolic syndrome: glycemic status, waist circumference, high-density lipoprotein (HDL), triglycerides and systolic blood pressure (SBP). These five components along with patient's gender were used to create a classification tool for polyneuropathy in these obese patients.

Previous research has found that many of the relationships between metabolic syndrome factors vary depending on patient glycemic status. The varying disease mechanisms within different glycemic subgroups make SVM-CART an ideal methodology in this setting. SVM-CART learns the distinct subgroups that may exist within the metabolic syndrome-neuropathy mechanism to build a predictive tool.

Creating a strong clinician friendly classifier of polyneuropathy allows for greater detection of neuropathy in certain patient subgroups and subsequently may improve patient care. A neurologist has the greatest expertise to diagnose neuropathy. However, most patients with neuropathy are followed by their primary care physician who may not have specific expertise in making this diagnosis. In contrast, the metabolic syndrome components are easily measured by a wide range of clinicians. A good classification tool based on the metabolic syndrome could target certain patient subgroups that are highly likely to have neuropathy based on their demographics and metabolic profile. These patients could be referred for additional testing or consultation with a neurologist.

The following section compares SVM-CART, SVM alone and CART alone (single classifier as well as ensemble). The methods were compared based on both prediction accuracy and interpretability.

Covariates in the study were gender (binary: male/female), glycemic status (categorical: normoglycemic, pre-diabetes, diabetes), and four continuous variables: systolic blood pressure (SBP, units=mmHg), triglyceride levels (TRIG, unit=mg/dL), high-density lipoprotein levels (HDL, unit=mg/dL) and waist circumference (WC, unit=cm). The primary outcome measure was the Toronto consensus definition of probable polyneuropathy (two or all of the following: neuropathy symptoms, abnormal sensory examination, and abnormal reflexes) as determined by a neuromuscular specialist [38]. In this cohort, 27 patients were diagnosed with neuropathy while 88 patients were determined not to have neuropathy.

2.5.2 Determining Optimal Tuning Parameters

In this application, careful examination of the tuning parameters is especially important because neuropathy is a rare event, even in this at risk, obese cohort. For the CART part of the methodology we implemented inverse weights for the neuropathy cases. Implementing these inverse weights gave proportionally higher importance to the correct classification of the neuropathy cases in the dataset.

An empirical grid search was used to determine the optimal hyperparameters for the SVM part of the SVM-CART classifier. Cost parameters ranging from 10^{-5} to 10^4 were considered. Within each terminal node created from the CART part, the SVM cost parameter weights were chosen as the inverse proportion within that terminal node subgroup. The cost weights were further extended by considering cost weight multipliers from 0.1 to 10 by 0.1 to either amplify or reduce the class weights on the cost parameters.

The optimal tuning parameters were selected by comparing prediction accuracy for the out-of-bag estimates across 1,000 bootstrapped samples for the 900 different cost/cost-weight multiplier scenarios. The classifiers were compared based on two statistics: % correct case classification and % correct control classification. Based on these statistics, the optimal cost parameter for SVM-CART was 100 and the class weight multiplier was 1.9. Figure 2.4 shows the % correct classification for

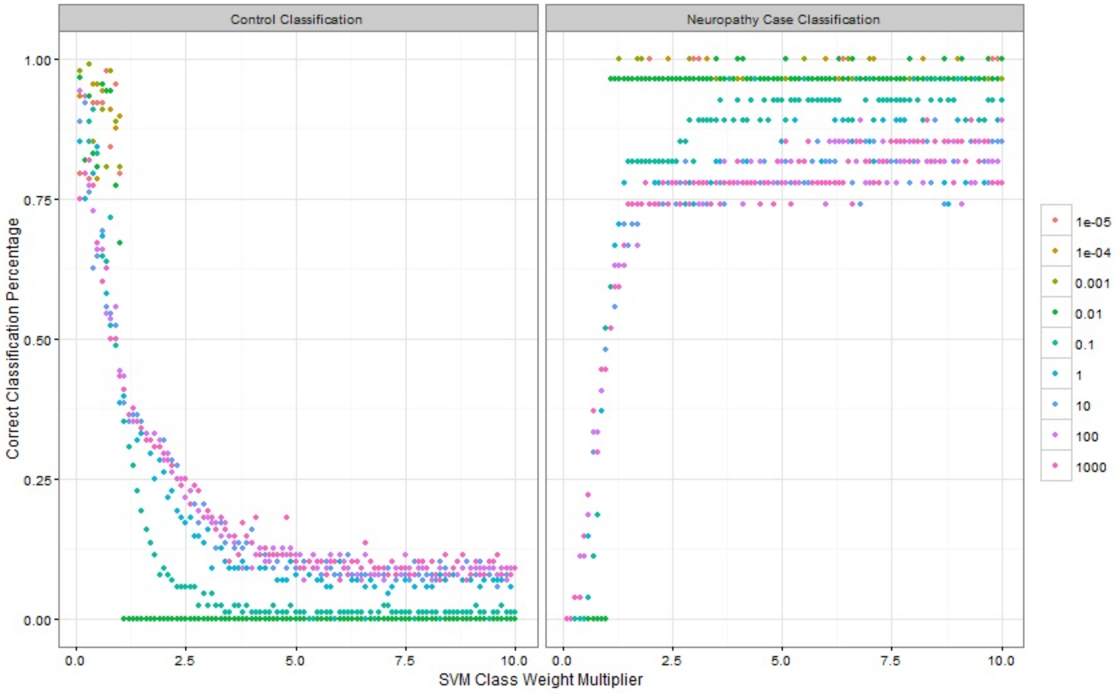

Figure 2.4: % Correct Classification Percentage for Tuning Parameter Selection

neuropathy and non-neuropathy patients based on SVM-CART. For SVM alone, the weight multiplier of 1.5 and the cost parameter of 100 were chosen as the optimal tuning parameters (figure not shown).

2.5.3 Comparison of SVM-CART, SVM and CART Single Classifiers

In this section, we compare the SVM-CART, SVM and CART classifiers based on prediction accuracy, interpretability and simplicity. Previous research lead us to believe that there are distinct subgroups based on glycemic status in this obese population. In the SVM-CART methodology, we first create a tree based on the patient gender and glycemic status. The resulting classifier is displayed in Figure 2.5.

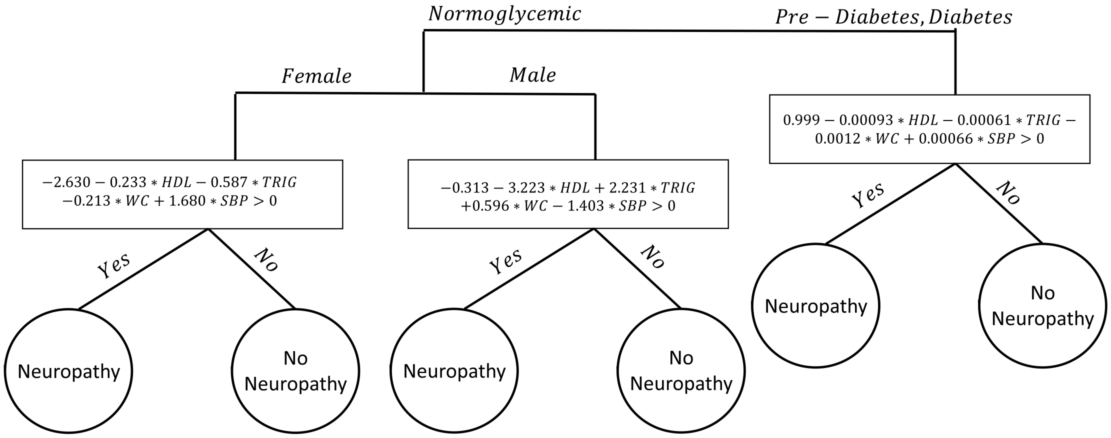
Normoglycemic
Pre − Diabetes, Diabetes
Female
Male
−2.630 − 0.233 ∗ HDL − 0.587 ∗ TRIG −0.213 ∗ WC + 1.680 ∗ SBP > 0
−0.313 − 3.223 ∗ HDL + 2.231 ∗ TRIG +0.596 ∗ WC − 1.403 ∗ SBP > 0
0.999 − 0.00093 ∗ HDL − 0.00061 ∗ TRIG − 0.0012 ∗ WC + 0.00066 ∗ SBP > 0
Yes
No
Yes
No
Yes
No
Neuropathy
No Neuropathy
Neuropathy
No Neuropathy
Neuropathy
No Neuropathy

Figure 2.5: Single SVM-CART Classifier for Neuropathy

The first split was by glycemic status: normoglycemic patients were separated from the pre-diabetes/diabetes patients. The normoglycemic patients were then split based on gender, resulting in three distinct subgroups: normoglycemic male, normoglycemic female and pre-diabetes/diabetes. These three groups of patients were then passed along to create three distinct four dimensional hyperplanes based on a linear soft margin SVM. The hyperplane was generated using each patient's waist circumference size, HDL, triglyceride and SBP levels.

CART-alone considered all five metabolic components and gender as a categorical variable, and produced a complicated tree with 10 terminal nodes. We determined the prediction accuracy of each method using a 10-fold cross validation.

The results are presented in Table 2.4. In terms of clinical relevance, the CART tree misses the most important predictor of neuropathy: glycemic status. In the tree created by CART alone, glycemic status only enters the tree at a deep split (figure not shown). Neuropathy case prediction accuracy (66.7% vs 70.4%) and overall prediction accuracy (53.9% vs 48.7%) were similar between SVM-CART and SVM alone respectively. SVM correctly classifies 19 of the 27 neuropathy patients while SVM-CART correctly classifies 18. SVM-CART correctly classifies 62/115 patients overall compared to SVM which only correctly classifies 56/115.

Though the prediction accuracy was similar between SVM and SVM-CART, SVM-CART was able to identify clinically meaningful subgroups which SVM alone was not able to create. For the neuropathy classifier demonstrated in this application, there are different mechanistic pathways to the disease within different subgroups of the population. Specifically, it was hypothesized that the neuropathy-metabolic syndrome relationship was different across glycemic subgroups. SVM-CART's ability to identify these subgroups give it enhanced interpretability compared to SVM alone.

2.5.4 Ensemble Classifier

Next, we attempt to improve prediction ability by creating an ensemble of classifiers. To compare the three methods, we examined the out-of-bag error rates. Ensembles of the classifiers were built using 1,000 bootstrap samples from the entire data.

Each of the three methods experience a boost in neuropathy classification performance when predictions were averaged over the 1,000 classifiers in the ensemble. SVM-CART gains 11.1% improvement in correct case classification, however there is a 11.3% decrease in overall correct classification. In conclusion, the SVM-CART ensemble outperforms the CART ensemble (77.8% vs 44.4% correct neuropathy classification) but is comparable to bootstrapped SVM correct neuropathy classification (77.8% for both).

Table 2.4: Prediction Accuracy Comparison of Various Classifiers

Method	% Correct Neuropathy Classification	% Correct Overall Classification
CART	51.9	59.1
SVM	70.4	48.7
SVM-CART Single Classifier	66.7	53.9
SVM-CART Ensemble	77.8	42.6

2.5.5 Representative Classifier

The ensemble of SVM-CART classifiers allows for a significant gain in neuropathy prediction accuracy compared to a single SVM-CART classifier, but we lose some of the interpretability. In this section, we select the most representative classifier from the ensemble based on two similarity metrics. The representative classifier can be used as a clinical decision-making tool. The first metric is based on the similarity in class prediction. The SVM-CART classifier that was most representative in this respect, is depicted in Figure 2.6. It is slightly different than the single classifier; we split first by all three glycemic categories and then further divide the pre-diabetes group by gender. We have four terminal nodes but only create three linear SVMs because the normoglycemic group is pure with a class prediction of no neuropathy.

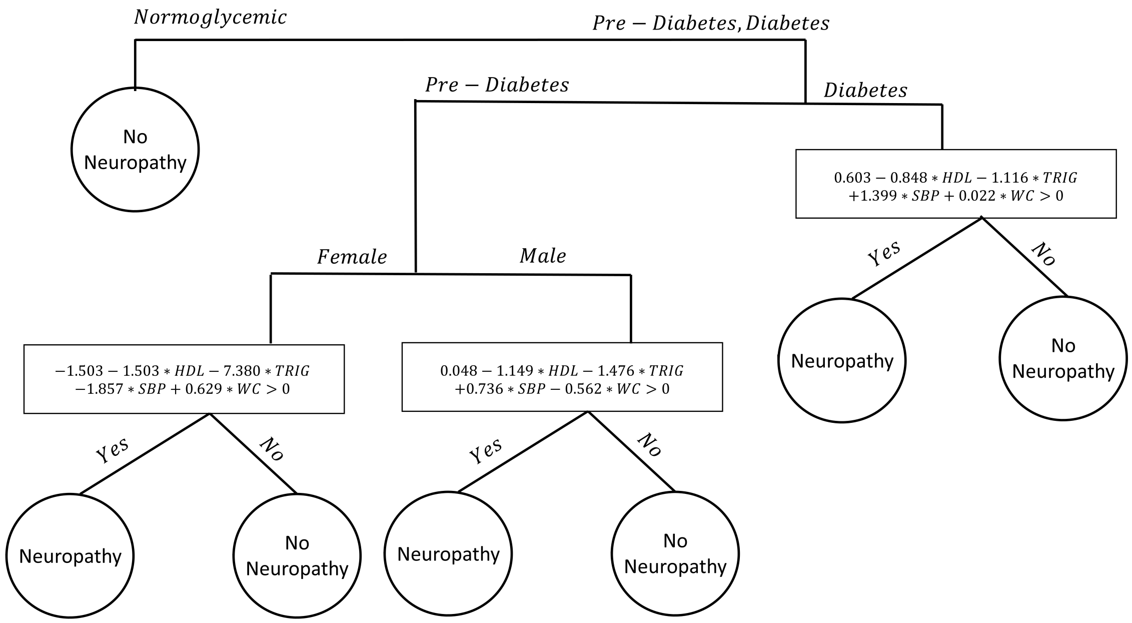
Normoglycemic
Pre − Diabetes, Diabetes
No Neuropathy
Pre − Diabetes
Diabetes
0.603 − 0.848 * HDL − 1.116 * TRIG +1.399 * SBP + 0.022 * WC > 0
Yes
No
Neuropathy
No Neuropathy
Female
Male
−1.503 − 1.503 * HDL − 7.380 * TRIG −1.857 * SBP + 0.629 * WC > 0
0.048 − 1.149 * HDL − 1.476 * TRIG +0.736 * SBP − 0.562 * WC > 0
Yes
No
Yes
No
Neuropathy
No Neuropathy
Neuropathy
No Neuropathy

Figure 2.6: Most Representative SVM-CART Classifier from the Ensemble

The second similarity metric focuses on how patients are clustered within terminal nodes. The most representative classifier in this respect (figure not shown) creates six subgroups based on each gender by glycemic status subgroup.

2.5.6 Neuropathy Study Conclusions

In conclusion, a strong classifier for neuropathy using patient metabolic measures has the potential to improve patient care. SVM-CART produces an ideal classifier that identifies different neuropathy-metabolic relationships across different gender/glycemic subgroups.

SVM-CART outperforms CART and performs similarly to SVM in terms of prediction accuracy both for the single classifier and ensemble. Using two similarity metrics, we selected the two most representative classifiers from the SVM-CART ensemble. The representative classifier provides a useful clinical tool while harnessing the improved predictive ability of the ensemble.

2.6 Discussion

Classification of disease continues to be an important aspect of analyzing human health data. Classification trees and support vector machines have both become popular tools for classifying patients into different disease groups. There are many clinical scenarios where each of these two methods fail to meet standards in terms of performance and applicability. Some of these common scenarios were highlighted in Figure 2.1 and in the simulation study. We propose a new classifier SVM-CART, that combines features of SVM and CART to allow for a more flexible classifier that has the potential to improve prediction accuracy and model interpretability.

CART offers an intuitive and interpretable method for classification. However, one significant drawback of CART is that it only allows rectangular splits that are perpendicular to the covariate space. There are many clinical scenarios where a non-rectangular split is more appropriate. In such scenarios, CART must create a very complicated tree in order to achieve reasonable prediction accuracy. SVM-CART can achieve similar or improved prediction performance with generally a more parsimonious structure. The more parsimonious structure created with SVM-CART provides a more interpretable decision making tool for clinicians.

The flexibility of SVM-CART allow it to uncover complex interactions among the covariates. In simulations, in the presence of interaction, SVM-CART outperforms SVM or CART alone. The structure created by SVM-CART makes it a very intuitive predictive tool in clinical scenarios where the disease-exposure mechanism may be very different across patient subgroups. This was the case in our neuropathy application, where it made clinical sense to create distinct classifiers based on gender and glycemic status subgroups. SVM-CART's potential ability to find clinically meaningful subgroups can lead to enhanced interpretability compared to SVM alone. When there is a priori clinical evidence to believe there are unique disease-exposure relationships between subgroups of the population, SVM-CART will likely have enhanced performance.

In settings where there is weak interaction, results from the simulation study were mixed. Therefore, we do not recommend using SVM-CART as a complete replacement for SVM or CART alone. In practice, it will be important to utilize the expertise of clinicians to determine if complex interactions likely exist amongst subgroups of the population for the clinical problem of interest. In such scenarios, SVM-CART will improve prediction ability and interpretability.

One important goal of our methodology was to develop a practical and usable tool for clinical decision making. We developed SVM-CART ensemble to improve prediction accuracy and stability of the classifiers. Though the ensemble method improved prediction accuracy, we lost the interpretability of a single SVM-CART classifier. We proposed two metrics that were used to identify the most representative classifier from the ensemble of SVM-CART classifiers. The resultant representative SVM-CART provided an interpretable, easy to use and flexible classification tool.

While linear SVMs provided a simple extension in our case, using more intricate kernel function SVM classifiers at each node has the potential to provide a powerful boost in certain scenarios. These kernelized extensions of support vector machines may allow for more flexible implementation of the SVM-CART method in non-linear problems that often exist with high dimensional feature space. Due to the small sample size of our neuropathy dataset, the kernel extensions did not perform as well as the linear SVM splits in the presented application. The methodology presented here can also be easily extended to multi-class outcomes.

In this paper, we make a distinction between categorical vs. continuous covariates in terms of how they are used in the SVM-CART classifier. In practice, a continuous covariate may behave like a categorical variable or could be categorized into an ordinal variable. If there is a priori evidence in the literature that different levels of a continuous covariate result in subgroups where the disease-exposure mechanism is different, then it should be added to the CART part of the classifier. One example of this might be age; as different age groups might lead to very different exposure mechanisms for various diseases.

The SVM-CART methodology presented here is one example of a composite classifier. In various clinical scenarios such as the ones presented in this research, using a wide range of classifiers in tandem might achieve better performance. In this research, CART was chosen as the first member of the composite classifier as it provided the most intuitive and flexible tool to separate patients into subgroups. Since CART is non-probabilistic, we decided to combine CART with another member of the general class of non-probabilistic classifiers. SVMs complement the rectangular splits created by CART and are arguably the most popular method within the class of non-probabilistic classifiers. Hence, we chose SVMs as the second member of the composite classifier. Extending the general approach to include other classifiers is an area of future research.

CHAPTER III

Regression Tree and Ensemble Methods for Multivariate Outcomes

Tree-based methods have become one of the most flexible, intuitive, and powerful analytic tools for exploring complex data structures. The best documented, and arguably most popular uses of tree-based methods are in biomedical research, where multivariate outcomes occur commonly (e.g. diastolic and systolic blood pressure, periodontal measures in dental health studies, and nerve density measures in studies of neuropathy). Existing tree-based methods for multivariate outcomes do not appropriately take into account the correlation that exists in such data or do not have the flexibility needed to make accurate prediction with complex data. In this paper, we develop goodness of split measures for multivariate tree building for continuous outcomes. We propose two general approaches to develop multivariate regression trees: by minimizing within-node homogeneity and by maximizing between-node separation. Specifically, we measure within-node homogeneity using the average Mahalanobis distance and the determinant of the covariance matrix. To measure between-node separation, we propose using the Mahalanobis distance, Euclidean distance and standardized Euclidean distance. Furthermore, to enhance prediction accuracy we extend the single multivariate regression tree to an ensemble of trees. Extensive simulations are presented to examine the properties of our goodness of fit measures. Finally, the proposed methods are illustrated using two clinical datasets of neuropathy and pediatric cardiac surgery.

3.1 Introduction

Prediction of multivariate outcomes is essential in many biomedical research problems. A multivariate prediction tool is required in scenarios where there are multiple outcomes from the same clinical domain, or when a diagnosis is based on multiple measures of the same intrinsic condition.

Currently, researchers wishing to use tree based methods to predict multivariate outcomes have very limited options. The first and most obvious approach is to build a series of univariate regression trees for each of the multivariate outcome components, whereby prediction is made by using each univariate tree [3-5]. There are two problems with this approach. First, as the dimension of the multivariate outcome increases, the trees lose their usability as a clinical decision-making tool. This is because to make predictions, a clinician has to evaluate a large number of trees, which is both time consuming and complicated. Second, we fail to take advantage of the correlation that likely exists between the multivariate outcomes. As a result, we may fail to uncover a subset of common covariates that are correlated with the multivariate outcome for the same underlying disease entity.

Earlier, De'Ath (2002) proposed a multivariate regression tree method using a general impurity function [40]. At each split, De'Ath demonstrates the method by building trees based on the covariate split that reduces the impurity in terms of the average variance across each of the outcomes [40]. Reducing the average variance of the outcomes is equivalent to choosing the split value that yields the maximum impurity reduction in terms of the covariance matrix trace. Thus, De'Ath's method only uses partial information from the covariance matrix and does not fully account for the correlation between the multivariate outcomes. Later, Larsen (2004) extended the work by De'Ath and proposed building multivariate regression trees by measuring node impurity as the median square-root of the Mahalanobis distances between the multivariate outcomes in that given node [41].

In this paper, we develop methodology for growing trees for multivariate continuous outcomes. Our proposed methods fall under two broad types of approaches. The first approach focuses on finding splits that reduce within node impurity [3-5]. We propose using the determinant of the covariance matrix and a flexible Mahalanobis distance as measures of within node impurity. The second approach focuses on finding splits that maximize between node separation in terms of the outcome [42,43]. We propose measuring between node separation using the Euclidean distance, standardized Euclidean distance and Mahalanobis distance.

This paper is organized as follows: Section 3.2 describes the methodology for growing a multivariate tree using the above goodness of split measures. Section 3.3 describes methods for growing an ensemble of multivariate regression trees. Section 3.4 provides a simulation study to assess the performance of our proposed multivariate tree methods. In Section 3.5, we illustrate our methodology to create a classifier for neuropathy and then to predict pediatric ICU visit outcomes. Lastly, in Section 3.6, we present concluding remarks and discussion.

3.2 Methodology

3.2.1 Univariate Classification and Regression Trees

We begin by introducing some terminology: a classification tree T has multiple nodes where observations are passed down the tree. The tree starts with a root node at the top and continues to be recursively split to yield the terminal nodes. At each stage of the splitting process, a binary decision rule is used to split parent nodes into two daughter nodes. The intermediate nodes in the tree between the root node and terminal nodes are referred to as internal nodes. We specifically denote the set of terminal nodes as $\tilde{T}$ and the number of terminal nodes is denoted as $|\tilde{T}|$. In classification trees, a class prediction is given to each observation based on which terminal node it falls into. In regression trees, the predicted value is the average outcome of the respondents in each terminal node [3-5].

There are three basic elements for constructing a tree under the classification and regression tree (CART) paradigm. These are: (1) tree growing, (2) finding the 'right-sized' tree and (3) testing [3-5]. In growing a tree, the natural question that arises is how and why a parent node is split into daughter nodes. Trees use binary splits, phrased in terms of the covariates, that partition the covariate space recursively. Each split depends upon the value of a single covariate. There are two general approaches to tree building. The first and most common approach follows that of Breiman, whereby a parent node is partitioned into two daughter nodes to increase within-node homogeneity [3]. Goodness of a split must therefore weigh the homogeneities in the two daughter nodes. The extent of node homogeneity is measured using an 'impurity' function. Potential splits for each of the covariates are evaluated, and the covariate and split value resulting in the greatest reduction in impurity is chosen [3-5].

For a split $s \epsilon S$ at node h, the left and right daughter nodes are denoted as h_L and h_R respectively. The impurity at given node h is denoted as $i(h)$ and the probability that a subject fall into node h is $P(h)$. $P(h)$ is estimated from the sample proportions in the training data. The reduction in impurity is calculated as follows: $\Delta I(s,h) = i(h) - P(h_L)i(h_L) - P(h_R)i(h_R)$. For binary outcomes, $i(h)$ is measured in terms of entropy or Gini impurity [3,4]. For continuous outcomes, $i(h)$ is typically the mean residual sum of squares [3-5]. The splitting rule that maximizes $\Delta I(s,h)$ over the set S of all possible splits is chosen as the best splitter for node h.

The second approach to tree building attempts to maximize the between node separation at each stage of the splitting process. Previous methods for building trees using between node separation have been popular with survival outcomes [42,43]. In this approach we measure between-node separation using a 'distance' function. As before, all potential splits are evaluated using the distance function and the split value resulting in the largest between node separation is chosen. The distance function between proposed daughter nodes h_L and h_R is denoted as $D(s, h_L, h_R)$ for the specific split $s \epsilon S$.

An important feature in CART is growing a large tree and then pruning it back to find the right-sized tree. Our proposed methods focus on the tree growing. For the within-node impurity based trees, one could borrow the original CART machinery to obtain pruned and cross-validated trees [3-5].

3.2.2 Multivariate Regression Tree

Extending univariate regression trees to continuous multivariate outcomes retains much of the same tree terminology. Moving from the univariate to multivariate framework, we propose several metrics that can be used to assess goodness of split in the multivariate setting. Our proposed metrics are based on either summary measures of the empirical covariance matrix at a given node or the distance between the proposed daughter nodes.

Let $\mathbf{Y}_i$ be the $r \times 1$ vector of outcomes for subject i. Define $\mathbf{Y} = (\mathbf{Y}_1', \mathbf{Y}_2', ..., \mathbf{Y}_n')'$ as the $n \times r$ outcome matrix. At a proposed split $s \epsilon S$, the patients that fall into the left and right daughter nodes are denoted by h_L and h_R respectively. Let $\mathbf{Y}_{(h)}$ be the outcome matrix and n_h the sample size (i.e. number of patients) in internal node h. At any internal node h, let $c\hat{o}v(\mathbf{Y}_{(h)})$ be the $r \times r$ matrix of observed variance and covariances.

3.2.3 Within Node Homogeneity Splitting

3.2.4 Determinant

In our first approach, we propose to build multivariate trees by reducing within node homogeneity. In the multivariate setting, to account for the correlation, we first propose to use generalized variance as a measure of impurity. The generalized variance was proposed as a one-dimensional measure of multidimensional scatter and is obtained using the determinant of the covariance matrix [45-48]. A node would be considered relatively impure if the determinant of the empirical covariance matrix was large and relatively pure if the multidimensional scatter (determinant) was small. Building on this, we define our impurity function at a given internal node h as:

$$i(h) = determinant(\hat{Cov}(\mathbf{Y}_{(h)})) \tag{3.1}$$

At each stage of the splitting process, the split value that results in the greatest impurity reduction in terms of the determinant of the empirical covariance matrix is chosen.

The method demonstrated by De'Ath, builds multivariate trees with an impurity function that can be written: $i(h) = trace(\hat{Cov}(\mathbf{Y}_{(h)}))$, where a node is considered relatively impure if the trace of the covariance matrix is large and relatively pure if the trace of the covariance matrix is small [40]. The trace impurity function's inability to utilize all information from the empirical covariance matrix is a potential limitation, especially in the high correlation setting.

3.2.5 Mahalanobis Distance

The second metric measures impurity by calculating the average distance amongst the subjects in a given node. Mahalanobis distance is a measure of the distance between a point and a distribution that is often used with multivariate data [49]. At a given node h, we first calculate the Mahalanobis distance between $\mathbf{Y}_i$ for patient i (in node h) and the distribution of $\mathbf{Y}_{(h)}$. Let $\bar{\mathbf{Y}}_{(h)}$ and $\hat{Cov}(\mathbf{Y}_{(h)})$ be the sample mean and empirical covariance matrix for the patients in node h. The Mahalanobis distance for patient i is calculated as:

$$Mah_1(\mathbf{Y}_i) = \sqrt{((\mathbf{Y}_i - \bar{\mathbf{Y}}_{(h)})'\hat{Cov}^{-1}(\mathbf{Y}_{(h)})(\mathbf{Y}_i - \bar{\mathbf{Y}}_{(h)})}$$

We define impurity at node h based on the average Mahalanobis distances:

$$i(h) = \frac{1}{n_h}\sum_{i=1}^{n_h} Mah_1(\mathbf{Y}_i) \tag{3.2}$$

where n_h is the number of patients in node h.

This method has similarities with that proposed by Larsen [41]. The primary difference between the methods is that while Larsen uses the covariance matrix of the full data, at each step of the tree growing process, we update the covariance matrix using patients at the given node for which the split is proposed. Our method uses empirical variances at each proposed split, thereby accounting for variations in the relationships across different subgroups of the population. In contrast, Larsen's method may be computationally efficient due to fewer parameter estimates.

3.2.6 Between Node Separation Splitting

In this section, tree growing proceeds by selecting the covariate value split that maximizes the between node separation in terms of the outcome. Specifically we wish to maximize the distance between the outcome centroid of the proposed daughter nodes. In the multivariate outcome setting, distance can be measured in a variety of ways. In this section, we propose three distance metrics that can be used for tree building using multivariate outcomes.

Euclidean Distance Splitting

The first proposed method measures the distance between the mean outcome in the two proposed daughter nodes using the Euclidean distance. The Euclidean distance function is formalized as:

$$D(s, h_L, h_R) = \mathbf{1}'(\bar{\mathbf{Y}}_{(h_R)} - \bar{\mathbf{Y}}_{(h_L)})'(\bar{\mathbf{Y}}_{(h_R)} - \bar{\mathbf{Y}}_{(h_L)}) \tag{3.3}$$

The covariate and covariate value the leads to the greatest $D(s, h_L, h_R)$ is chosen at that stage of the splitting process.

Standardized Euclidean Distance Splitting

The second proposed between-node splitting mechanism measures node separation using the standardized Euclidean distance. Standardizing the Euclidean distance by the empirical variances of the outcomes in the parent node results in a similar relative scale for each outcome component. This may be preferable in a setting where the scale of the multivariate outcome components are different. The Standardized Euclidean distance function is written as:

$$D(s, h_L, h_R) = \mathbf{1}'(\bar{\mathbf{Y}}_{(h_R)} - \bar{\mathbf{Y}}_{(h_L)})' diag(\hat{Cov}(\mathbf{Y}_{(h_R)}))^{-1}(\bar{\mathbf{Y}}_{(h_R)} - \bar{\mathbf{Y}}_{(h_L)}) \tag{3.4}$$

Mahalanobis Distance Splitting

A limitation of the two Euclidean based distance functions is they are unable to take into account the correlated nature of the multivariate outcomes. The final proposed distance function uses the Mahalanobis distance. This distance function is slightly more complicated than simply finding the distance between the centroids in the previous two distance functions. For this splitting method, we must evaluate the total Mahalanobis distance between each subject's outcome and the alternative daughter node outcome distribution. For a given patient i in h_L, the Mahalanobis distance between patient i and alternative daughter node h_R is defined as:

$$Mah_R(\mathbf{Y}_i) = \sqrt{((\mathbf{Y}_i - \bar{\mathbf{Y}}_{(h_R)})'\hat{Cov}^{-1}(\mathbf{Y}_{(h_R)})(\mathbf{Y}_i - \bar{\mathbf{Y}}_{(h_R)})} \tag{3.5}$$

and similarly, we denote the Mahalanobis distance between patient j in h_R and the alternative daughter node h_L as:

$$Mah_L(\mathbf{Y}_j) = \sqrt{((\mathbf{Y}_j - \bar{\mathbf{Y}}_{(h_L)})'\hat{Cov}^{-1}(\mathbf{Y}_{(h_L)})(\mathbf{Y}_j - \bar{\mathbf{Y}}_{(h_L)})}$$

where $\hat{Cov}(\mathbf{Y}_{(h_R)})$ and $\bar{\mathbf{Y}}_{(h_R)}$ are the empirical covariance matrix and sample mean for the subjects in the proposed right daughter node. Then the total Mahalanobis distance for a proposed split s, is defined as:

$$D(s, h_L, h_R) = \textstyle\sum_{i=1}^{n_L}(Mah_R(\mathbf{Y}_i)) + \sum_{j=1}^{n_R} Mah_L(\mathbf{Y}_j)$$

The split s, in the set of all splits S, that results in the largest $D(s, h_L, h_R)$ is chosen at that stage of the splitting process.

3.2.7 Multivariate Tree Prediction

For prediction using the multivariate tree, patients are first passed down the tree based on their covariates until they land in one of the $|\tilde{T}|$ terminal nodes. Prediction is made by taking the average of the outcome from all patients that fall into that node. For a specific terminal node $\tilde{T}_p$, the predicted $r \times 1$ outcome vector is given by:

$$\hat{\mathbf{y}}_i = \tfrac{1}{n_h}\mathbf{1}'\mathbf{Y}_{(\tilde{T}_p)}$$

3.3 Ensemble Methods for Multivariate Outcomes

The two biggest shortcomings of tree-based methods is modest prediction performance and instability in the tree structure. Instability in building the single tree occurs because small perturbations of the data can have very dramatic effects on the structure of the created tree. Ensemble methods are often used with tree-based methods to help remedy the modest prediction performance and instability of the single trees. An ensemble method proposed in 1994 by Breiman involved bootstrap aggregating or Bagging [31-34]. The premise of this method is to generate many bootstrap samples of the data and create an individual regression tree from each of the bootstrap samples. The final predicted outcome is given as the average prediction across each individual tree in the ensemble.

To increase stability and improve prediction accuracy, we propose growing ensembles of the multivariate regression trees. First, we generate b bootstrap samples by sampling uniformly n observations from the entire data of size n. For each of the b samples, we create a multivariate regression tree. For each patient, prediction from the ensemble is made by averaging the predictions from each of the b multivariate trees.

For each tree in the ensemble, the out-of-bag sample is the set of patients that were not included in the bootstrap sample used to create the tree [31]. The out-of-bag prediction is the average outcome across the $m \leq b$ samples in which the patient was in the out-of-bag sample. To fairly assess the prediction accuracy of the ensemble methods, we calculate the mean squared error (MSE) based on the out-of-bag prediction which is referred to as the out-of-bag error.

3.4 Simulation Study

3.4.1 Simulation Methods

Data Generated from Multivariate Linear Regression

To compare and evaluate the proposed goodness of split measures, we performed an extensive simulation study. Covariates were generated by $x_1 \sim Bernoulli(0.3)$,

$x_2 \sim N(50, 15)$ and $x_3 \sim multinomial(0.45, 0.15, 0.40)$. Additionally, we generated two noise variables: $x_4 \sim N(1.5, 1.1)$ and $x_5 \sim uniform(1, 10)$, that were included in tree building but not in the true underlying model.

The outcome was generated using the following multivariate linear regression model with 3 covariates:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{Y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_r)$ is the $n \times r$ outcome matrix, $\boldsymbol{X} = (\boldsymbol{1}, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_{3_1}\boldsymbol{x}_{3_2})$ is the $n \times 5$ design matrix, $\boldsymbol{B} = (\boldsymbol{b}_1, ..., \boldsymbol{b}_r)$ is the $5 \times r$ effect matrix with $\boldsymbol{b}_j = (\beta_{0j}, \beta_{1j}, \beta_{2j}, \beta_{3j}, \beta_{4j})$ and $\boldsymbol{\varepsilon} = (\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_n)'$ is the $n \times r$ error matrix with $\boldsymbol{\epsilon_i} \sim MVN(\boldsymbol{0}_r, \Sigma)$. The β coefficients are fixed across the simulation settings.

We varied the dimension of the outcome: $r\epsilon\{2, 5\}$, the sample size: $N\epsilon\{100, 500, 1000\}$ and considered three different structures for Σ (unstructured: Σ_1, compound symmetry (CS): Σ_2 and heterogeneous compound symmetry (CSH): Σ_3). The unstructured covariance is defined:

$$\Sigma_1 = \begin{bmatrix} \sigma^2_{1,1} & \dots & \sigma^2_{1,r} \\ \vdots & \ddots & \vdots \\ \sigma^2_{r,1} & \dots & \sigma^2_{r,r} \end{bmatrix}$$

We calculate $\Sigma_1 = \boldsymbol{A}'\boldsymbol{A}$, by generating $\boldsymbol{A}$ with values $a_{ij} \sim uniform(-\sigma, \sigma)$, where choice of $\sigma\epsilon\{1, 5, 10\}$, creates low, moderate and large magnitude covariance scenarios.

The compound symmetry covariance is defined:

$$\Sigma_2 = \sigma^2 \begin{bmatrix} 1 & \dots & \rho \\ \vdots & \ddots & \vdots \\ \rho & \dots & 1 \end{bmatrix}$$

The heterogeneous compound symmetry covariance is defined:

$$\Sigma_3 = \begin{bmatrix} \sigma^2 \times m_1^2 & \dots & \sigma^2 \times \rho \times m_1 \times m_r \\ \vdots & \ddots & \vdots \\ \sigma^2 \times \rho \times m_r \times m_1 & \dots & \sigma^2 \times m_r^2 \end{bmatrix}$$

For $\{\Sigma_2, \Sigma_3\}$, we also varied the correlation: $\rho\epsilon\{0.05, 0.50, 0.95\}$ and the variance: $\sigma^2\epsilon\{1, 25, 100\}$. Additionally, to create the CSH covariance matrix, we set $(m_1, m_2) = (1, 2)$ and $(m_1, m_2, m_3, m_4, m_5) = (1, 2, 3, 4, 5)$ when $r = 2$ and $r = 5$ respectively. For each design scenario, we generated 1000 simulations.

The tuning parameter used for tree building was minimum terminal node size, which was set as {5,15,25} when $n = \{100, 500, 1000\}$ respectively. Our interest lies in contrasting the splitting rules for the tree growing. Therefore, we do not perform any cost-complexity pruning.

In a separate simulation, we generated data from a true underlying multivariate regression tree structure. The data generating structure is displayed in Figure 3.1. In this set-up, the tree first splits patients by x_1, x_2 and x_3. For each of the four terminal nodes in the true underlying tree, we generate outcomes from a node-specific multivariate normal distribution with covariance, Σ_2 and mean, $\boldsymbol{\mu}$. For this set of simulations, we assessed performance under various scenarios where: $r = 5$, $N\epsilon\{100, 500, 1000\}$, $\rho\epsilon\{0.05, 0.50, 0.95\}$ and $\sigma^2\epsilon\{1, 25, 100\}$.

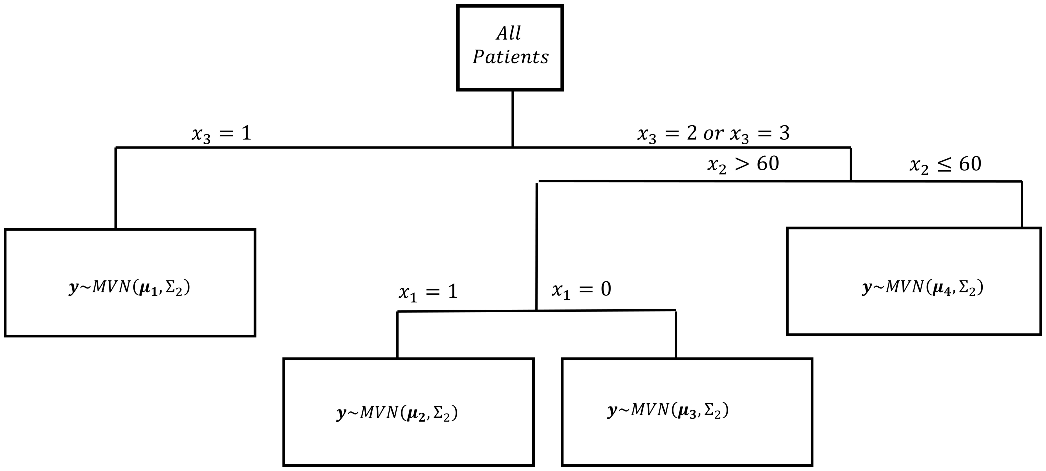
All Patients
$x_3 = 1$
$x_3 = 2$ or $x_3 = 3$
$x_2 > 60$
$x_2 \leq 60$
$\boldsymbol{y} \sim MVN(\boldsymbol{\mu_1}, \Sigma_2)$
$x_1 = 1$
$x_1 = 0$
$\boldsymbol{y} \sim MVN(\boldsymbol{\mu_4}, \Sigma_2)$
$\boldsymbol{y} \sim MVN(\boldsymbol{\mu_2}, \Sigma_2)$
$\boldsymbol{y} \sim MVN(\boldsymbol{\mu_3}, \Sigma_2)$

Figure 3.1: True Underlying Multivariate Regression Tree

3.4.2 Simulation Results

We assess the performance of the splitting rules by examining a variety of measures that focus on the tree structure and predictive ability. For each simulation run, we split the simulated data into a training and testing set by randomly selecting 70% of the sample for training and 30% for testing. The sample size in the testing set is n_{test}. The goodness of split measures are compared in terms of the following:

1. $MSE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (\mathbf{y}_i - \hat{\mathbf{y}}_i)'(\mathbf{y}_i - \hat{\mathbf{y}}_i)$
2. % of times true signal variables are used for splitting
3. % of times the noise variables are used for splitting
4. Average number of terminal nodes

Table 3.1-Table 3.4 describe the simulation results.

Metric based on Within-Node Homogeneity, $r = 5$

The prediction accuracy for the regression trees were similar across the different within-node metrics. When $r = 5$, under Σ_2 and Σ_3: the Larsen method had the best performance in terms of MSE, except under Σ_3 with $\sigma^2 = 100$, where the determinant trees had the smallest MSE. When simulating the outcome from an unstructured covariance setting (Σ_1), the Larsen metric had the smallest MSE (average MSE: 15,748), followed by the the determinant (16,979), Mahalanobis distance (17,006) and trace trees (21,423).

Generally, as σ^2 increased, the prediction accuracy of each metric worsened. Holding all other parameters constant, under Σ_2, the determinant trees were least effected in terms of MSE (1.43% MSE increase) when comparing $\sigma^2 = 100$ to $\sigma^2 = 1$, followed by the Mahalanobis trees (1.55% MSE increase), trace trees (1.61% MSE increase) and Larsen trees (2.10% MSE increase). Under Σ_1, comparing $\sigma^2 = 1$ to $\sigma^2 = 100$, there was a 4.8% MSE increase for determinant trees, 5.7% increase for trace trees, 6.4% increase for Mahalanobis trees and 6.8% increase for Larsen trees. A similar pattern can be observed under Σ_3, where the Larsen metric had the largest increases in terms of MSE when comparing $\sigma^2 = 100$ to $\sigma^2 = 1$. This suggests that the Determinant trees are well-suited in a high variance setting and the Larsen trees are not well-suited.

Holding all other settings constant, there were slight improvements in MSE for the Mahalanobis trees when comparing $\rho = 0.95$ to $\rho = 0.05$ (0.5% improvement under Σ_2 and 0.9% improvement under Σ_3). The trace, determinant and Larsen trees had mixed performance when comparing varying levels of ρ. These results suggest the Mahalanobis trees are ideal when the correlation amongst the outcome is high.

Each of the methods improved prediction accuracy as the sample size increased.

Under each covariance structure: Σ_1, Σ_2 and Σ_3, the trace trees had the largest MSE reduction (25.7%, 27.2% and 10.1% respectively) when comparing $N = 1000$ to $N = 100$. The Mahalanobis trees had the second largest MSE reduction (21.5%, 20.0% and 10.1% respectively). The determinant tree had the smallest prediction accuracy improvements (19.0%, 19.8% and 6.5% respectively), suggesting that the metric may be the well suited to build trees in the small sample setting.

All methods perform poorly in terms of separating the signal variables from the noise variables, though, the Mahalanobis trees included the fewest noise variable splits of the within-node metrics. In terms of tree size, the determinant metric yielded the simplest trees under Σ_1, Σ_2 and Σ_3, (average number of nodes: 10.7, 11.0 and 11.3) followed by the Mahalanobis trees (11.7, 11.9 and 12.0), Larsen (11.7, 12.1 and 12.1) and the trace trees (11.4, 12.2 and 12.3). There were no clear patterns in terms of which specific covariates were used as splits in the resulting trees.

Metric based on Within-Node Homogeneity, $r = 2$

Under Σ_2, when $N = 100$, the Larsen method had the smallest MSE and when $N = 500$ or $N = 1000$, the trace trees had the best prediction accuracy. Under Σ_3, the smallest MSE was from the Mahalanobis trees when $\sigma^2 = 100$ and the trace trees when $\sigma^2 = 25$ or $\sigma^2 = 1$. Under Σ_1, the Larsen tree had the best prediction accuracy (average MSE: 4,937), followed by Mahalanobis distance (5,078), determinant (5,128) and trace trees (6,792).

All of the metrics had increased MSE when comparing $\sigma^2 = 100$ to $\sigma^2 = 1$. Unlike what was observed when $r = 5$, when $r = 2$, there were no clear trends to which metric was most resistant to MSE increases as the variance increased. Under Σ_2, the Larsen trees were most resistant to MSE increases (1.66% MSE increase), followed by the determinant (1.79%), trace (2.08%) and Mahalanobis (2.71%). Under Σ_3, the Mahalanobis distance had the smallest percent increase in MSE (415% increase), followed by the determinant (428%), Larsen (428%) and trace trees (436%). Lastly, under Σ_1, the trace trees had the smallest MSE gain (2.2%), followed by Larsen (3.3%), Mahalanobis distance (3.3%) and determinant trees (5.0%). Holding all other simulation settings constant, there were no clear trends in prediction accuracy when comparing $\rho = 0.95$ to $\rho = 0.05$ under any covariance structure.

Each of the metrics had similar improvement in MSE as sample size increased from $N = 100$ to $N = 1000$ under each of the three covariance structures. The metrics performed similarly in terms of which covariates were used as splits in the tree across the different settings. Contrary to when $r = 5$, under Σ_1, Σ_2 and Σ_3 when $r = 2$, the Mahalanobis trees had the smallest average number of terminal nodes (11.3, 10.7 and 10.9), followed by the determinant trees (11.2, 11.7 and 11.5), Larsen (11.7, 11.9 and 8.9) and trace trees (11.8, 12.3 and 12.1).

Metric based on Between-Node Separation, $r = 2$

Of the between-node separation metrics, under both Σ_2 and Σ_3, the Mahalanobis distance trees had the smallest MSE when $N = 100$ and the standardized Euclidean trees had the smallest MSE when $N = 500$ and $N = 1000$. Under Σ_1, the Euclidean distance metrics had the best prediction accuracies.

The Mahalanobis trees had a smaller increase in MSE than the Euclidean metrics when comparing $\sigma^2 = 100$ to $\sigma^2 = 1$ (Σ_1: 3.0% vs to 27.4% and 26.6%, Σ_2: 7.9% vs 4.0% and 3.2%, Σ_3: 447% vs 453% and 456%). The Mahalanobis trees also had the largest reduction in MSE when comparing $\rho = 0.95$ to $\rho = 0.05$ (3.6% improvement under Σ_2 and 0.2% improvement under Σ_3). The two Euclidean distance metrics had no clear MSE improvement across levels of ρ.

Each of the between-node metrics improved prediction accuracy as sample size increased from $N = 100$ to $N = 1000$. Under Σ_1, the Euclidean trees had the greatest reduction in MSE (28.0%) followed by standardized Euclidean (27.1%) and Mahalanobis distance (12.6%). Similarly, under Σ_2, the Euclidean distance trees had largest reduction in MSE (26.5%) compared to the standardized Euclidean (26.0%) and Mahalanobis distance trees (13.6%). Under Σ_3, each of the three between-node separation metrics led to similar improvements in MSE (11%) comparing $N = 100$ to $N = 1000$.

While each of the metrics incorrectly split on noise variables at a high rate, the Mahalanobis distance trees included noise covariates at a slightly lower rate compared to the Euclidean distance and standardized Euclidean distance metrics (Σ_2: 99.2% v 100% and 99.9% and Σ_3: 99.6% v 100% and 100%). The Mahalanobis trees were also slightly larger than the Euclidean and standardized Euclidean trees (Σ_1: 14.3 v 14.1 and 14.2, Σ_2: 14.9 v 14.4 and 14.7 and Σ_3: 17.3 v 15.1 and 15.1).

Metric based on Between-Node Separation, $r = 5$

Similarly to when $r = 2$, the Mahalanobis distance metric resulted in the best prediction accuracy (MSE) when $N = 100$ under Σ_2. However, under Σ_2 with moderate and large sample size or Σ_1, the Euclidean and standardized Euclidean distance split trees had the smallest MSE. The resulting MSEs under Σ_3 were similar across the three metrics.

There were no trends in terms of which between node metric was most resistant to increased MSE as the σ^2 increased. Under Σ_1, the standardized Euclidean trees had the smallest MSE increase (6.9%), under Σ_2, the Euclidean trees had the smallest average MSE increase (3.0%) and under Σ_3, the Mahalanobis trees had the smallest MSE increase (371%). There were also no trends when comparing the three metrics across different levels of ρ.

Under and Σ_1 and Σ_2, as sample size increased from $N = 100$ to $N = 1000$, the Euclidean and standardized Euclidean metrics had large MSE reductions compared to the Mahalanobis distance (Σ_1: 20.8% and 21.9% vs. 5.1%. Σ_2: 24.1% and 24.1% vs. 3.2%). The MSE reductions were similar amongst the three metrics as sample size increased under Σ_3.

Similar to when $r = 2$, compared to the Euclidean trees, the Mahalanobis trees had a larger average number of terminal nodes (Σ_1: 16.6 vs. 14.5 and 14.6, Σ_2: 15.3 vs. 14.7 and 14.7, Σ_3: 17.3 vs. 15.0 and 15.4). The Mahalanobis distance trees included noise covariates as splits at a slightly lower rate than the Euclidean and standardized Euclidean distance trees (Σ_1: 99.6% vs. 100% and 100%, Σ_2: 99.3% vs. 100% and 100%, Σ_3: 99.6% vs. 100% and 100%)

Simulation Results with Data Generated from Underlying Tree Structure

The determinant trees had the best prediction accuracy (MSE) of the within-node metrics when data were generated from a true underlying tree structure. Specifically, the determinant trees had the best prediction accuracy when the underlying variance was large or when $\rho = 0.05$. When the variance was small and ρ was large, the Larsen method had the best prediction accuracy. Each of the metrics performed similarly across different levels of ρ while holding all other parameters constant. Each metric had a similar MSE inflation when comparing the large variance setting to the small variance setting (each around 10% increase).

Of the between-node separation metrics, the Euclidean distance trees were often the best performing in terms of prediction accuracy. There was little differences across different ρ and each method had similar MSE inflation as σ increased.

Summary of Simulations

Results from these simulations provide valuable insight. Generally, the prediction accuracies (MSE) are very similar across the metrics, but under certain scenarios, specific metrics may be preferable to build a decision making tool. Amongst the within-node metrics, when the dimension of the outcome is small ($r = 2$), the trace trees are often preferable whereas, when the outcome dimension is moderate ($r = 5$), the Larsen trees are often preferable. For high variance scenarios, the Mahalanobis or determinant metrics are preferred. In particular, the determinant trees were the most resistant to MSE inflation as the σ increased. The Mahalanobis trees (when $r = 5$) and determinant trees (when $r = 2$) often result in the most simple tree structure. We also observed that the Mahalanobis trees had improved performance when the correlation amongst the outcome is large (when $r = 5$) and included the noise covariates as a slightly smaller rate. Finally, the determinant trees may be well suited in a small sample setting ($N = 100$).

In comparing the simulation results for the between-node metrics we found that generally, the Mahalanobis splits have the best performance when the sample size is small ($N = 100$). The Euclidean and standardized Euclidean distance metrics may have better performance when the sample is larger ($N = 500$ or $N = 1000$), or when the data follows an unstructured covariance. The Mahalanobis distance metric generally built slightly larger trees, but included noise covariates at a slightly lower rate. Across levels of σ and ρ, there were no consistent trends amongst the metrics.

When data was generated from a true underlying multivariate tree structure, the between-node Euclidean and the within-node determinant were the preferred metrics.

Table 3.1: Simulation Results (MSE) with Data Generated from Unstructured Covariance Matrix (Σ_1)

			r=2			r=5		
			$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$
Metrics based on Between Node Separation	$\sigma = 1$	Mah	5678.86	4892.57	4913.11	17049.05	14602.27	16672.81
		Euc	3238.46	2527.64	2336.02	17373.12	14184.74	13880.65
		Std Euc	3227.04	2514.86	2359.57	17742.23	14307.36	13818.96
	$\sigma = 25$	Mah	5705.92	4932.90	4991.62	17669.40	15127.35	16567.20
		Euc	3276.09	2514.82	2340.50	17790.97	14607.83	14009.75
		Std Euc	3250.72	2517.19	2368.44	17859.53	14626.98	13989.46
	$\sigma = 100$	Mah	5798.45	5035.98	5112.64	18652.27	15970.87	17405.26
		Euc	3338.97	2564.52	2415.35	18723.15	15558.15	14793.80
		Std Euc	3313.82	2554.62	2409.24	18832.90	15503.27	14693.95
Metrics based on Within Node Homogeneity	$\sigma = 1$	Trace	8018.25	6537.63	5605.61	24597.36	19911.87	18235.98
		Det	5863.88	4814.66	4340.05	19178.29	15250.25	15467.58
		Mah	5874.15	4722.82	4414.00	19165.89	15285.87	15037.50
		Larsen	5830.13	4607.95	4172.71	17726.04	14552.56	13803.69
	$\sigma = 25$	Trace	8134.50	6576.25	5645.50	25014.90	20149.48	18577.44
		Det	6039.48	4828.28	4486.38	19487.75	15461.39	15656.93
		Mah	5997.93	4704.48	4480.74	19701.68	15698.96	15492.30
		Larsen	5827.02	4647.29	4258.08	18051.59	14649.72	13727.97
	$\sigma = 100$	Trace	8238.45	6659.45	5714.78	26003.54	20984.62	19332.33
		Det	6142.84	4979.21	4653.04	19723.65	16403.68	16180.35
		Mah	6035.49	4898.40	4574.00	20303.15	16421.20	15947.89
		Larsen	6044.70	4737.42	4310.75	19250.06	15462.23	14504.03

3.5 Illustrative Examples

3.5.1 Application: Predicting Nerve Conduction Measures

Data Collection and Background Information

Polyneuropathy is a painful condition affecting an estimated $2-7\%$ of the adult population [36, 37]. The most common predictor of the disease is diabetes; however, it is hypothesized that other components in the metabolic syndrome can play a role in the etiology of polyneuropathy [15,38,39]. In this section, we attempt to build a regression tree to predict three neuropathy outcomes using patient measures from the metabolic syndrome.

The data for this application comes from participants in the Health, Aging and Body composition study (Health ABC): a prospective cohort study of 70-79 year olds [39]. The cohort is a simple random sample of age eligible patients in Pittsburgh or Memphis that planned to remain in the study area for at least three years. The participants had to report difficulty with walking $\frac{1}{4}$ mile, climbing 10 steps or any basic

Table 3.2: Simulation Results (MSE) for Within-Node Homogeneity Metrics with Data Generated from Compound Symmetry and Heterogeneous Compound Symmetry (Σ_2 and Σ_3)

			r=2						r=5					
			CS			CSH			CS			CSH		
			$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$
$\rho = 0.05$	$\sigma = 1$	Trace	3317.13	2621.11	2433.09	3247.62	2660.79	2485.61	25107.16	19311.34	18238.41	25200.43	19413.03	18334.50
		Det	3415.11	2693.47	2533.25	3300.77	2705.09	2515.44	19176.32	15530.14	15325.65	19365.00	15557.62	15821.49
		Mah	3355.37	2739.59	2607.96	3444.95	2672.69	2695.01	19020.25	15139.97	14999.51	18668.48	15234.31	15198.28
		Larsen	3300.87	2676.79	2513.84	3274.30	2672.79	2524.56	18493.34	14143.63	13425.67	18270.94	14143.52	13504.21
	$\sigma = 25$	Trace	3370.59	2637.00	2454.40	11009.37	9922.00	9589.37	25310.93	19459.88	18388.41	63201.57	55261.38	53553.23
		Det	3451.92	2688.12	2523.31	11056.70	9937.21	9624.67	18958.29	15541.72	15723.25	56819.46	50673.86	49528.32
		Mah	3398.97	2749.99	2707.00	11110.93	9948.86	9701.00	18950.26	15438.62	15020.68	58246.95	50311.10	49436.13
		Larsen	3252.63	2650.84	2556.66	11013.81	9882.75	9630.45	18687.30	14260.81	13656.43	56075.82	50266.10	48776.95
	$\sigma = 100$	Trace	3442.14	2678.85	2541.14	127406.09	118400.46	116189.14	25566.87	19845.40	18771.29	636080.83	588982.76	580768.59
		Det	3499.14	2758.93	2612.79	127493.88	118506.77	115928.96	19284.36	1565.32	15946.72	610609.91	581197.27	574717.29
		Mah	3485.32	2806.41	2737.35	126999.57	118270.69	115893.33	19002.85	16070.78	15170.24	638881.13	580701.19	574050.45
		Larsen	3339.13	2716.38	2595.56	127495.23	118388.00	116864.93	18999.74	14548.14	13735.57	613026.66	581225.47	574899.89
$\rho = 0.5$	$\sigma = 1$	Trace	3313.28	2612.72	2432.07	3256.04	2655.78	2484.72	25132.11	19322.34	18286.36	25217.18	19403.25	18316.70
		Det	3417.81	2696.14	2531.24	3313.18	2700.11	2515.38	19042.76	15581.85	15394.18	19322.00	15384.16	15603.21
		Mah	3343.94	2725.23	2609.84	3449.79	2652.57	2692.17	19032.30	15292.48	14804.78	18861.96	15511.99	15194.40
		Larsen	3296.64	2670.53	2525.26	3284.07	2677.74	2530.65	18402.26	14183.23	13527.22	18548.54	14152.39	13619.20
	$\sigma = 25$	Trace	3353.22	2620.40	2451.69	11092.87	9887.72	9615.50	25316.58	19462.06	18363.91	63013.38	54745.17	53372.57
		Det	3448.91	2691.83	2535.19	11060.11	9922.53	9623.59	19885.84	15631.54	15437.96	57115.60	50563.09	49675.75
		Mah	3411.89	2738.95	2678.87	11090.11	9922.84	9711.48	18822.72	15455.47	15156.14	57890.10	50274.75	49845.85
		Larsen	3281.82	2648.05	2559.17	10979.87	9896.14	9624.12	18697.69	14489.27	13629.09	55835.71	49861.31	49226.46
	$\sigma = 100$	Trace	3435.73	2681.46	2538.17	127726.12	127726.12	116177.19	25745.96	19833.42	18740.74	636267.76	588096.12	581034.77
		Det	3505.65	2746.64	2597.11	127331.57	127331.57	116092.70	20027.68	16253.63	15764.11	609861.40	580267.00	574697.03
		Mah	3499.16	2814.61	2729.14	127241.77	127241.77	115937.35	18892.85	15834.80	15401.03	634861.40	579170.99	574169.91
		Larsen	3349.35	2723.92	2579.65	127819.61	127291.61	116107.64	18802.84	14870.20	14125.43	610455.52	580736.69	574687.01
$\rho = 0.95$	$\sigma = 1$	Trace	3323.89	2606.67	2436.30	3254.67	2651.54	2477.40	25128.94	19292.27	18273.61	25191.78	19378.21	18284.57
		Det	3414.43	2693.75	2531.47	3327.94	2709.40	2508.73	18930.39	15441.15	15506.42	19392.95	15628.25	15579.63
		Mah	3343.45	2718.22	2613.60	3474.57	2668.22	2696.10	18806.30	15160.87	15021.37	18849.15	15504.30	15116.65
		Larsen	3284.86	2666.65	2521.20	3289.28	2658.83	2550.78	18505.37	14102.55	13651.06	18578.17	14180.56	13434.94
	$\sigma = 25$	Trace	3337.13	2614.20	2438.01	11302.10	9940.88	9576.11	25213.15	19461.86	18336.47	63273.82	54857.27	53274.92
		Det	3439.08	2697.45	2544.64	11180.20	9942.72	9653.72	19681.72	15808.19	15600.28	56633.25	50804.97	49595.29
		Mah	3378.43	2728.64	2654.51	11491.20	9885.99	9694.90	18468.09	153237.47	14943.13	58018.05	50315.12	49614.46
		Larsen	3294.58	2671.09	2554.56	126188.07	12492.68	12442.16	18442.36	143513.35	13621.94	56139.43	49886.41	48894.00
	$\sigma = 100$	Trace	3403.09	2668.82	2516.74	128305.26	118589.95	116122.32	25591.25	19896.20	18756.71	638739.35	588636.56	580575.46
		Det	3534.98	2741.05	2603.51	126816.14	118335.68	116011.59	20072.38	15585.43	15650.85	604155.48	578045.26	573928.00
		Mah	3529.61	2815.33	2714.39	132607.53	118306.69	115720.18	18948.89	15897.69	15486.01	627083.18	576355.01	572944.43
		Larsen	3369.54	2738.33	2648.00	118360.71	119470.52	116827.73	18918.52	14888.48	13953.45	604432.12	577816.39	574255.39

Table 3.3: Simulation Results (MSE) for Between Node Metrics with Data Generated from Compound Symmetry and Heterogeneous Compound Symmetry (Σ_2 and Σ_3)

			r=2						r=5					
			CS			CSH			CS			CSH		
			$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$	$N = 100$	$N = 500$	$N = 1000$
ρ=.05	$\sigma = 1$	Mah	3149.55	2663.53	2575.53	3114.82	2603.79	2602.77	16891.37	14753.89	16550.03	17078.48	14824.76	16517.76
		Euc	3171.92	2517.49	2356.86	3177.96	2525.39	2369.20	17749.37	14161.64	13494.27	17737.36	14345.52	13607.32
		Std Euc	3148.44	2496.83	2347.83	3155.14	2503.71	2351.53	17802.68	14107.26	13416.12	17616.48	14192.72	13515.85
	$\sigma = 25$	Mah	3160.23	2690.10	2624.02	11252.23	9896.30	9737.01	17233.63	15118.61	16616.03	56328.01	50516.18	50891.59
		Euc	3217.62	2533.15	2370.98	11166.67	9813.48	9518.77	17740.3	14222.29	13552.73	56874.86	50202.92	49287.29
		Std Euc	3203.64	2523.08	2367.12	11132.02	9796.03	9453.18	17771.88	14386.37	13551.09	56886.66	50175.01	49293.66
	$\sigma = 100$	Mah	3234.14	2773.19	3707.67	132463.85	119457.35	116978.87	17715.29	15228.36	16847.86	632133.83	590923.09	580721.23
		Euc	3301.77	2610.24	2420.13	130677.23	119165.11	116504.84	18446.58	14728.07	13823.23	627710.94	5872283.19	578879.61
		Std Euc	3271.30	2562.65	2413.83	129955.21	119279.77	116504.52	18398.89	14674.42	13884.81	627871.24	586997.88	579083.66
ρ=.5	$\sigma = 1$	Mah	3149.71	2665.03	2581.99	3109.99	2610.99	2606.11	17026.33	14744.87	16644.23	17148.54	14892.85	16592.14
		Euc	3171.61	2515.13	2349.05	3191.36	2525.42	2374.65	17776.33	14193.89	13522.11	17737.99	14351.27	13556.01
		Std Euc	3152.29	2500.57	2343.08	3161.70	2506.20	2356.76	17615.06	14181.73	13456.05	176621.30	14276.28	13557.09
	$\sigma = 25$	Mah	3157.64	2693.16	2635.08	11279.46	9941.60	9780.57	17087.73	14935.99	163289.3	55828.94	50924.26	50972.16
		Euc	3224.48	2529.83	2360.57	11181.97	9821.05	9520.54	17837.35	14282.78	13455.97	56600.99	50160.24	49013.78
		Std Euc	3189.26	2519.72	2365.60	11106.60	9786.09	9496.07	17798.02	14277.11	13503.49	56833.71	50156.49	48933.36
	$\sigma = 100$	Mah	3234.77	2759.45	2698.01	132101.00	132100.97	116516.52	17773.38	15216.15	16960.62	626286.74	590080.40	580167.74
		Euc	3316.97	2596.22	2432.56	130073.40	119221.40	116601.31	18143.28	14708.58	13902.96	625631.47	585200.50	578231.62
		Std Euc	3267.31	2564.66	2424.24	130534.50	119185.66	116562.60	18212.45	14648.77	13897.18	625662.90	586126.01	578080.41
ρ=.95	$\sigma = 1$	Mah	3142.97	2659.54	2585.09	3129.21	2610.18	2603.06	17038.17	14874.58	16517.92	16903.52	14788.85	16645.33
		Euc	3174.09	2511.30	2350.90	3181.78	2525.86	2356.52	17887.18	14088.42	13524.85	17830.16	14281.77	13575.26
		Std Euc	3160.07	2496.76	2370.95	3170.19	2505.18	2354.22	17667.01	14191.33	13575.28	17784.79	14237.49	13616.64
	$\sigma = 25$	Mah	3160.28	2686.33	2635.16	11181.73	9801.62	9743.53	16935.99	14556.23	16600.37	55838.01	50479.96	50410.29
		Euc	3230.42	2534.32	2349.12	11197.98	9824.68	9489.43	17642.65	14397.54	13502.41	56575.93	50137.91	48930.84
		Std Euc	3213.23	2517.72	2359.38	11193.30	9811.15	9469.78	17891.58	14429.63	13424.46	56628.66	50105.62	48962.37
	$\sigma = 100$	Mah	3273.55	2757.36	2732.90	130762.40	120510.61	117107.61	17558.45	15057.14	17093.33	627083.2	586840.09	579616.33
		Euc	3338.23	2629.38	2441.29	129890.60	119183.31	116656.80	18307.39	14638.21	13764.87	615152.3	582087.91	577114.12
		Std Euc	3311.83	2562.57	2408.87	130546.30	119242.93	116553.33	18329.64	14626.77	13840.52	615872.8	581726.15	577247.82

Table 3.4: Simulation Results (MSE) with Data Generated from Underlying Tree Structure

			$\rho = 0.05$	$\rho = 0.50$	$\rho = 0.95$
Metrics based on Between Node Separation	Small Variance	Btw Mah	6196.72	6209.90	6203.40
		Euc	5842.06	5870.91	5865.22
		Std Euc	5861.00	5880.67	5863.29
	Moderate Variance	Btw Mah	6337.95	6357.51	6365.73
		Euc	5967.54	5978.80	5974.60
		Std Euc	5988.35	6014.60	5985.18
	Large Variance	Btw Mah	6726.85	6755.70	6747.68
		Euc	6407.92	6408.07	6354.14
		Std Euc	6404.17	6417.67	6371.50
Metrics based on Within Node Homogeneity	Small Variance	Trace	5554.61	5551.22	5562.98
		Det	5045.30	5075.94	5107.04
		Mah	5117.79	5090.53	5099.47
		Larsen	5089.99	5062.18	5087.08
	Moderate Variance	Trace	5690.51	5713.12	5701.27
		Det	5187.48	5210.39	5262.94
		Mah	5300.89	5256.88	5290.32
		Larsen	5231.84	5210.98	5253.26
	Large Variance	Trace	6065.26	6073.31	6048.91
		Det	5572.38	5600.90	5580.40
		Mah	5734.49	5674.52	5708.91
		Larsen	5597.24	5603.63	5618.92

activity of daily living. Participants have been followed since 1997-1998 and have had a variety of neuropathy testing performed. This application is a cross-sectional analysis of the data in 2000-2001. There were three continuous nerve conduction study (NCS) outcomes that were measured for patients in this cohort: peroneal motor nerve conduction velocity (CV, m/s), peroneal CMAP (mV) and nerve vibration threshold (μm). These three nerve conduction study measurements are used to aid in the diagnosis of neuropathy [39]. There were 1,748 patients that had nerve conduction study testing and therefore included in this application.

Creating a strong predictive tool for neuropathy based on the metabolic syndrome components would allow for earlier testing and detection of neuropathy which could subsequently improve patient care. Nerve conduction studies are generally performed by a neurologist who has the highest expertise in making a diagnosis of neuropathy. However, each of the metabolic syndrome factors are typically measured by a range of clinicians with diverse experience and expertise. Therefore, strong prediction of these multivariate NCS measures could target patients with a heightened risk of neuropathy for expedited NCS testing by a neurology specialist [14].

There are five components of the metabolic syndrome: glycemic status, waist circumference, high-density lipoprotein (HDL), triglycerides and systolic blood pressure (SBP). These five components along with patient's gender and age are used in building regression trees to predict the NCS outcomes.

In the next section, we present results from the proposed multivariate tree methods using both a single classifier and ensemble. Results are compared using prediction accuracy and tree structure. Prediction accuracy was assessed using MSE and tree structure is assessed by examining the number of terminal nodes. Lastly, we assess whether the trees split patients based on the glycemic status: the most clinically jus-

Table 3.5: Prediction Results for Neuropathy Outcomes

Tree Method	10-Fold Cross Validation	Out of Bag Error from Ensemble	Number of Terminal Nodes	Number of Covariates Included
3 Univariate Trees	1206.5	1171.4	26	7
Trace	1220.7	1199.0	19	7
Determinant	1221.7	1216.0	5	4
Mahalanobis Distance	1193.1	1229.5	8	5
Mahalanobis Distance (Larsen)	1186.3	1206.6	16	4
Between Node Mahalanobis	1238.9	1235.9	4	4
Between Node Euclidean	1197.5	1211.9	22	6
Between Node Standardized Euclidean	1198.8	1210.2	22	6

tified predictor of neuropathy. We calculate the MSE using a 10-fold cross validation for the single trees and using the out-of-bag error for the ensembles.

Comparison of Multivariate Regression Trees for Neuropathy Measures

The correlation between the peroneal CV and peroneal CMAP is 0.40, between the vibration threshold and peroneal CV is -0.20 and between the vibration threshold and peroneal CMAP is -0.18. When building the regression trees, we set the minimum terminal node size to be 50.

The trace-based goodness of split resulted in the third worst tree in terms of prediction accuracy (MSE=1212.27), a complicated structure with 19 terminal nodes, but split on each of the seven covariates of interest. The determinant-based goodness of split resulted in a simple tree with 5 terminal nodes but the second worst prediction accuracy (MSE=1239.33). The determinant tree split patients based on glycemic status, waist circumference, SBP and HDL levels. This tree included the most important covariate, namely glycemic status, and split on 4 of the 5 metabolic syndrome components while maintaining simplicity. The determinant tree is displayed in Figure 3.2.

The Mahalanobis distance-based metric resulted in a tree with the second smallest MSE and is displayed in Figure 3.3. This tree was moderately simple with 8 terminal nodes, splitting patients based on gender, waist circumference, SBP and

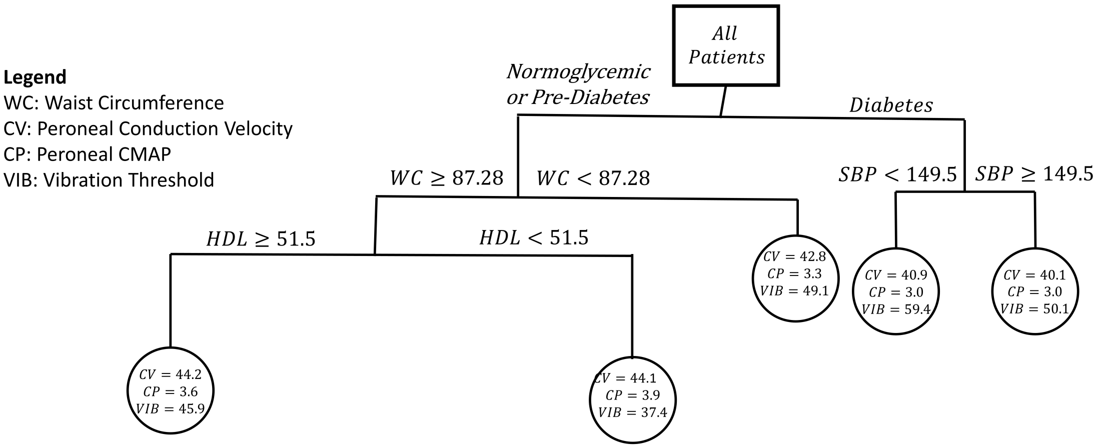
Legend
WC: Waist Circumference
CV: Peroneal Conduction Velocity
CP: Peroneal CMAP
VIB: Vibration Threshold
All Patients
Normoglycemic or Pre-Diabetes
Diabetes
$WC \geq 87.28$
$WC < 87.28$
$SBP < 149.5$
$SBP \geq 149.5$
$HDL \geq 51.5$
$HDL < 51.5$
$CV = 42.8$ $CP = 3.3$ $VIB = 49.1$
$CV = 40.9$ $CP = 3.0$ $VIB = 59.4$
$CV = 40.1$ $CP = 3.0$ $VIB = 50.1$
$CV = 44.2$ $CP = 3.6$ $VIB = 45.9$
$CV = 44.1$ $CP = 3.9$ $VIB = 37.4$

Figure 3.2: Multivariate Neuropathy Tree Using Determinant Impurity

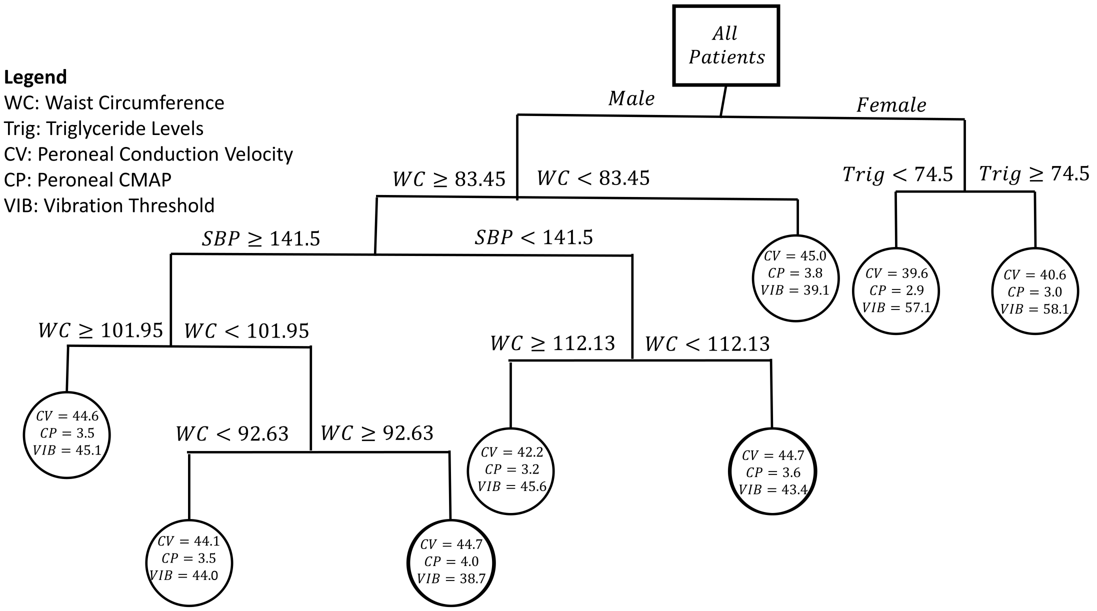
Legend
WC: Waist Circumference
Trig: Triglyceride Levels
CV: Peroneal Conduction Velocity
CP: Peroneal CMAP
VIB: Vibration Threshold
All Patients
Male
Female
$WC \geq 83.45$
$WC < 83.45$
$Trig < 74.5$
$Trig \geq 74.5$
$SBP \geq 141.5$
$SBP < 141.5$
$CV = 45.0$ $CP = 3.8$ $VIB = 39.1$
$CV = 39.6$ $CP = 2.9$ $VIB = 57.1$
$CV = 40.6$ $CP = 3.0$ $VIB = 58.1$
$WC \geq 101.95$
$WC < 101.95$
$WC \geq 112.13$
$WC < 112.13$
$CV = 44.6$ $CP = 3.5$ $VIB = 45.1$
$WC < 92.63$
$WC \geq 92.63$
$CV = 42.2$ $CP = 3.2$ $VIB = 45.6$
$CV = 44.7$ $CP = 3.6$ $VIB = 43.4$
$CV = 44.1$ $CP = 3.5$ $VIB = 44.0$
$CV = 44.7$ $CP = 4.0$ $VIB = 38.7$

Figure 3.3: Multivariate Neuropathy Tree Using Mahalanobis Impurity

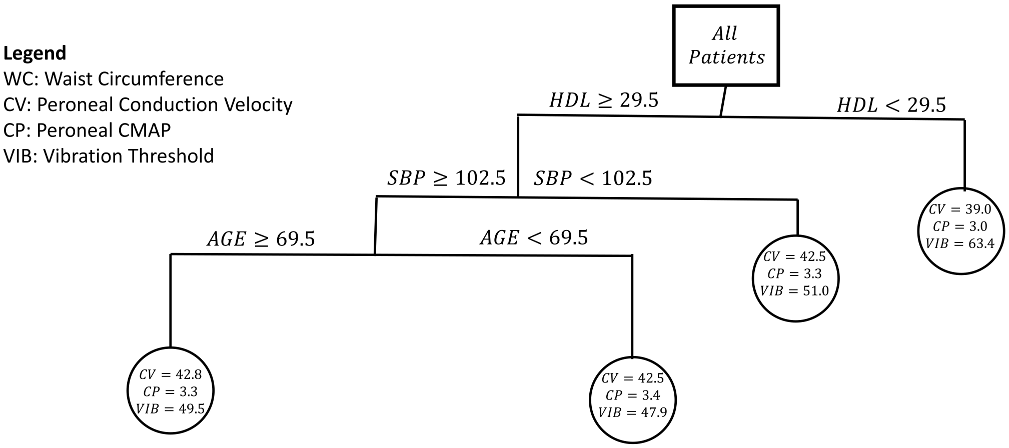
Legend
WC: Waist Circumference
CV: Peroneal Conduction Velocity
CP: Peroneal CMAP
VIB: Vibration Threshold
All Patients
$HDL \geq 29.5$
$HDL < 29.5$
$SBP \geq 102.5$
$SBP < 102.5$
$AGE \geq 69.5$
$AGE < 69.5$
$CV = 39.0$ $CP = 3.0$ $VIB = 63.4$
$CV = 42.5$ $CP = 3.3$ $VIB = 51.0$
$CV = 42.8$ $CP = 3.3$ $VIB = 49.5$
$CV = 42.5$ $CP = 3.4$ $VIB = 47.9$

Figure 3.4: Multivariate Neuropathy Tree Using Between Node Mahalanobis Distance

triglyceride levels. While the resulting tree from the Mahalanobis metric split on 5/7 covariates of interest, it failed to split on glycemic status. Using the full sample covariance estimate in the Mahalanobis distance resulted in the best prediction accuracy (MSE=1186.3), but a complex, 16 terminal nodes tree structure. Though it split on 6/7 covariates, the Larsen Mahalanobis tree missed the most clinically proven predictor: glycemic status.

The between node Mahalanobis tree (Figure 3.4) resulted in the worst prediction performance (MSE=1238.9) and the simplest tree stucture with only 4 terminal nodes. The between-node Mahalanobis distance tree split on HDL, SBP and age. The Euclidean and standardized Euclidean distance splits performed slightly better in terms of prediction (MSE=1197.5 and MSE=1198.8) but were the most complicated of the multivariate methods with 22 terminal nodes. The two Euclidean distance trees were identical in structure.

The three univariate trees (8, 5 and 13 terminal nodes), perform worse than the within node Mahalanobis and Euclidean metrics in terms of prediction accuracy (MSE=1206.5). To predict each of the three neuropathy outcomes, a clinician would have to use 3 different trees, thereby limiting clinical usability as a decision-making tool. Another issue with the resulting univariate trees is the clinical reliability of the tree structure. That is, these methods lose validity with clinicians when different covariates are used to predict different measures from the same clinical construct: neuropathy. Each of the three univariate trees include gender, age, triglycerides and waist circumference, however, only two of the trees include SBP and pre-diabetes, and only one of the trees include diabetes status. Since each of the multivariate components measure the same disease, ideally, the same clinical predictors would be included in each of the univariate trees.

Comparison of Multivariate Tree Ensemble Methods

We attempted to stabilize the tree structure and improve prediction accuracy by evaluating a multivariate tree ensemble. The regression tree ensembles were built using 1000 bootstrap samples from the entire data. The out-of-bag prediction error from the ensemble is displayed in Table 3.5. The prediction performance improves slightly for the trace, determinant, between node Mahalanobis distance and univariate trees. Interestingly, there is a larger improvement in prediction accuracy for the univariate, determinant and trace ensembles compared to the between-node distance and within node Mahalanobis metrics.

Conclusions for Nerve Conduction Prediction

The results from this Neuropathy application follow many of the trends that were observed in the simulation study for a low/moderate ρ, large N and moderate r setting. In assessing the single trees, the within node Mahalanobis distance metrics performed the best in terms of prediction accuracy and the between-node Mahalanobis distance metric performed the worst. The between-node Mahalanobis distance metric resulted in the simplest tree structure while the trees developed from the trace and Euclidean distance metrics were the most complicated. The method that identified the most clinically relevant structure was the determinant tree: identifying diabetes as the most important risk factor.

3.5.2 Application: Predicting Pediatric Post-Operative Length of Hospitalization

Data Collection and Background information

Improving the quality of care given to pediatric patients is an important goal of health service researchers and clinicians alike. There is room to improve the quality of care given to pediatric patients undergoing critical cardiac care, as cardiac arrest occurs in an estimated 2.6-6% of children with cardiac disease [8,9]. This subsequently results in a large mortality rate [8,9]. Additionally, there is significant variation in congenital cardiac disease outcomes amongst hospitals in the United States [10]. The length of time patients stay in a hospital has been linked to poor outcomes and high healthcare costs [11]. In this section, we build a multivariate regression tree to predict the length of time pediatric heart disease patients spend in different aspects of their hospitalization after thoracic surgery.

The data in this application comes from the Pediatric Cardiac Critical Care Consortium (PC4). Formed in 2009, PC4 is a collaboration of clinical leaders from 32 institutions across the United States. The overarching goal of PC4 is to improve the quality of care given to pediatric patients with critical cardiovascular disease.

A tool that accurately predicts various aspects of postoperative hospitalization time has many useful applications. First, finding patient factors that accurately predict the length of hospitalization could allow clinicians to intervene earlier in the stay to help troubleshoot the complications that might arise. The goal would be to intervene with patients at risk of a long length of stay to reduce the risk of mortality and reduce overall healthcare costs. Second, the resulting prediction could help hospitals with resource utilization and staffing. Last, having a precise prediction tool could be used in evaluating clinician performance; meaning clinicians can attempt to shorten these predicted lengths of stay by improving patient care.

For this application we included 7,066 pediatric patients from the PC4 database that had a STS (Society of Thoracic Surgeons) defined operation. This included both Cardiopulmonary bypass (CPB) surgery and non-CPB surgery. The goal of this application is to take patient demographic and clinical information at the time of operation to predict three aspects of hospitalization time. The typical hospitalization timeline of a pediatric patient in our data is displayed in Figure 3.5. The three outcomes for this application are the length hospitalization post operation (not including time in the ICU), the amount of time in the ICU (not while on a mechanical ventilator), and the amount of time on a mechanical ventilator. If patients had multiple stints in the ICU or on a mechanical ventilator, the times were added together.

The covariates obtained from the PC4 registry describe a variety of clinical and demographic factors collected at the time of operation. Demographic factors include: patient age, gender, race and ethnicity. Clinical measurements from the child's birth include: birth weight, birth length, head circumference, gestational age at birth and whether the child was born prematurely. Other clinical covariates include: the amount of time the patient was hospitalized before the operation, whether the patient had previous cardiothoracic operations, whether there was an antenatal diagnosis of congenital heart disease, whether the patient had a chromosomal abnormality

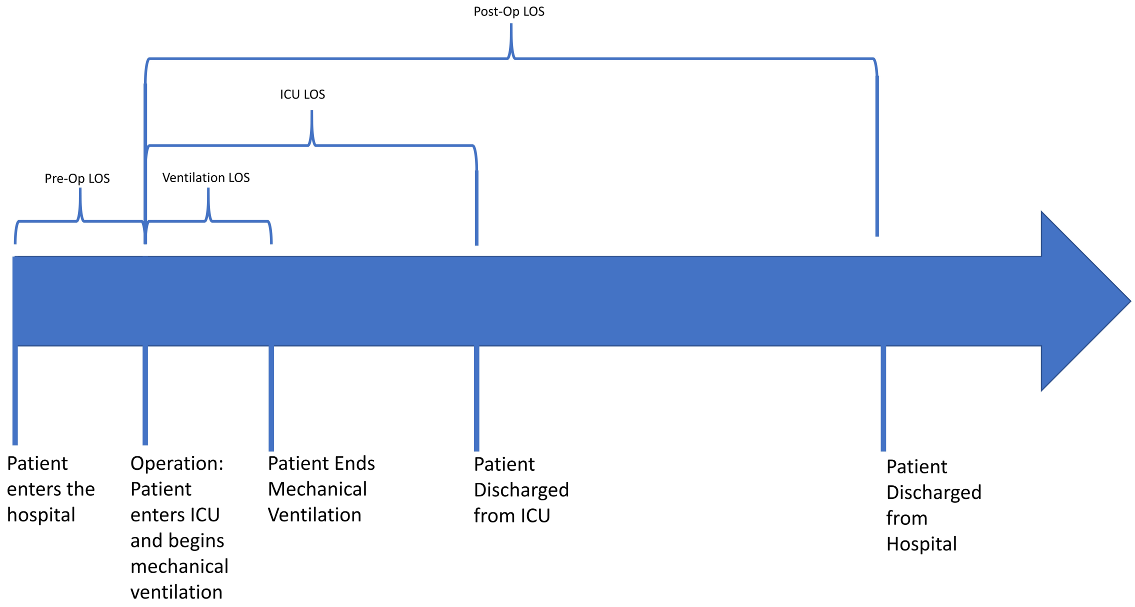
Post-Op LOS
ICU LOS
Pre-Op LOS
Ventilation LOS
Patient enters the hospital
Operation: Patient enters ICU and begins mechanical ventilation
Patient Ends Mechanical Ventilation
Patient Discharged from ICU
Patient Discharged from Hospital

Figure 3.5: Typical Pediatric Hospitalization Time line for Congenital Heart Disease Patients

or identified syndrome, whether there was a presence of extracardiac anomaly, the patient's STAT score, whether they had vasoactive support at the time of surgery and whether the patient had renal failure or stroke.

In the following section, results from the goodness of split metrics are presented and compared. As in the neuropathy application, prediction accuracy and simplicity are assessed using the MSE and the number of terminal nodes respectively. The MSE is calculated from a 10-fold cross validation for the single trees and the out-of-bag error rate for the ensemble methods.

Comparison of Multivariate Regression Trees for Pediatric Length of Stay

The correlation between the times spent in the CICU and on mechanical ventilation is 0.35, between the CICU and in the non-CICU inpatient setting is 0.30 and between the times spent with mechanical ventilation and in the non-CICU inpatient setting is 0.25. When building the regression trees, we set the minimum terminal node size to be 750.

Table 3.6: Prediction Results for Pediatric Length of Stay

Tree Method	10-Fold Cross Validation	Out of Bag Error from Ensemble	Number of Terminal Nodes	Number of Covariates Included
3 Univariate Trees	460.5	432.0	13	2
Trace	459.5	470.5	5	3
Determinant	443.7	469.7	6	3
Mahalanobis Distance	490.5	490.5	3	2
Mahalanobis Distance (Larsen)	499.8	501.0	4	2
Between Node Mahalanobis	441.8	455.5	7	6
Between Node Euclidean	446.7	459.5	5	3
Between Node Standardized Euclidean	446.7	461.3	5	3

The prediction results are displayed in Table 3.6. The resulting trace, Mahalanobis distance, determinant and between-node Mahalanobis distance trees are displayed in Figures 3.6, 3.7, 3.8 and 3.9.

With a MSE of 443.7, the determinant tree was the best performing within-node tree in terms of prediction accuracy: splitting by birthweight, length of stay in the hospital pre-operation and STAT score, resulting in 6 terminal nodes. The trace tree had the second best prediction accuracy of the within-node trees with 5 terminal nodes, splitting by pre-operation hospitalization length, by antenatal heart disease diagnosis and STAT score (MSE=459.5). The two within-node Mahalanobis distance trees performed the worst in terms of prediction accuracy (MSEs of 490.5 and 499.8), but produced very simple trees (3 and 4 terminal nodes).

The between-node Mahalanobis distance resulted in the most accurate tree (MSE=441.8), split by the broadest number of clinical covariates and had 7 terminal nodes. The two Euclidean trees had the same structure, splitting by hospitalization time before the surgery, antenatal diagnosis and birthweight (MSE=446.7 for both). The Euclidean trees had 5 terminal nodes.

The univariate trees performed similarly to the multivariate trees in terms of

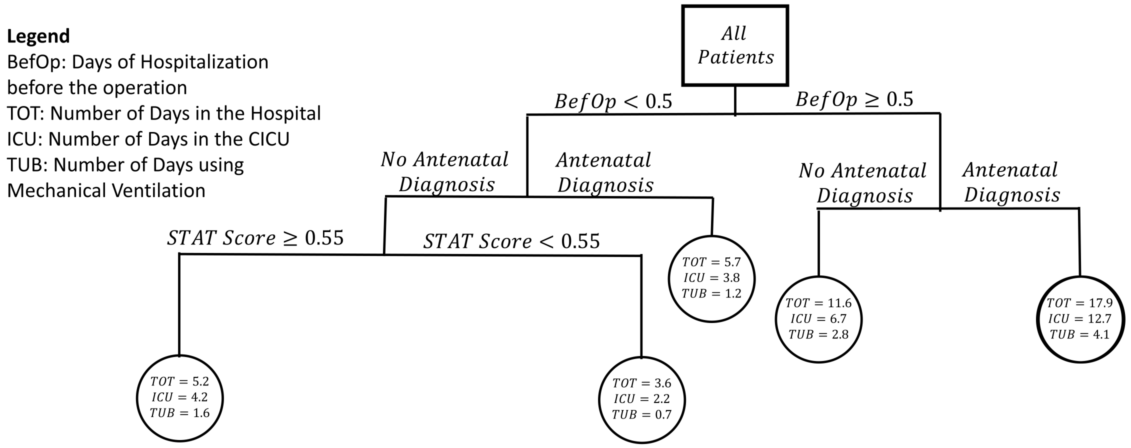

Figure 3.6: Multivariate Tree for Pediatric Hospitalization Time Using Trace Impurity

prediction accuracy (MSE=460.5), but produced a more complicated prediction tool (5, 4 and 4 terminal nodes). Since the multivariate trees perform similarly in terms of prediction accuracy, the simplicity of only evaluating a single tree to predict the multivariate outcome gives the multivariate trees an advantage over the univariate trees.

When evaluating the out-of-bag error from the ensemble, each of the multivariate methods perform slightly worse in terms of prediction accuracy. Only the univariate trees improve prediction accuracy, improving the MSE from 460.5 to 432.0.

Pediatric Post-Operative Length of Hospitalization Conclusion

The results from this application closely followed the trends observed in simulation. The PC4 registry data used in this application was large sample (N=7,066), had moderate outcome size (r=3) and a large variance ($mean(diag(\hat{\Sigma})) = 35.7$ and $mean(\hat{\Sigma} - diag(\hat{\Sigma})) = 166.8$). Similar to what was observed in simulation, the determinant tree performed well in this high-variance scenario, having the best prediction

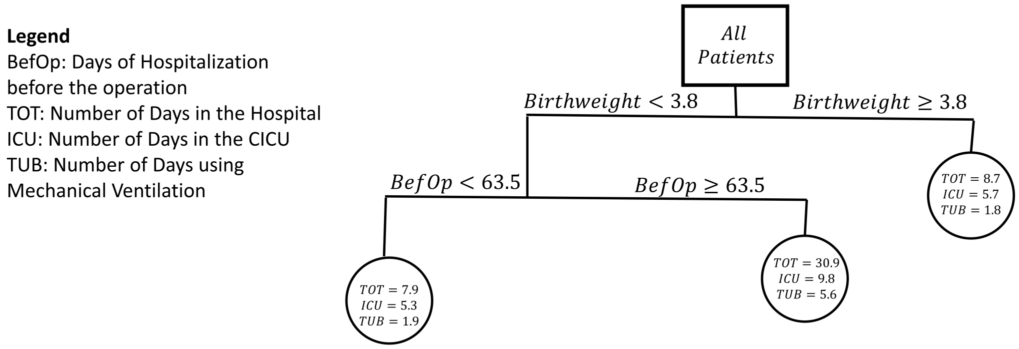
Legend
BefOp: Days of Hospitalization before the operation
TOT: Number of Days in the Hospital
ICU: Number of Days in the CICU
TUB: Number of Days using Mechanical Ventilation
All Patients
$Birthweight < 3.8$
$Birthweight \geq 3.8$
$BefOp < 63.5$
$BefOp \geq 63.5$
$TOT = 8.7$ $ICU = 5.7$ $TUB = 1.8$
$TOT = 7.9$ $ICU = 5.3$ $TUB = 1.9$
$TOT = 30.9$ $ICU = 9.8$ $TUB = 5.6$

Figure 3.7: Multivariate Tree for Pediatric Hospitalization Time Using Mahalanobis Distance Impurity

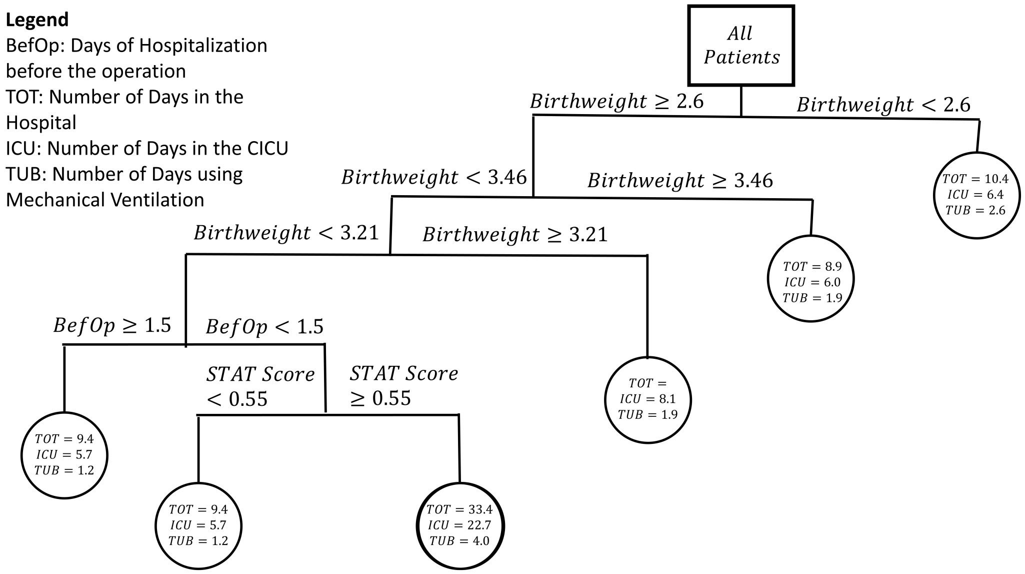
Legend
BefOp: Days of Hospitalization before the operation
TOT: Number of Days in the Hospital
ICU: Number of Days in the CICU
TUB: Number of Days using Mechanical Ventilation
All Patients
$Birthweight \geq 2.6$
$Birthweight < 2.6$
$Birthweight < 3.46$
$Birthweight \geq 3.46$
$TOT = 10.4$ $ICU = 6.4$ $TUB = 2.6$
$Birthweight < 3.21$
$Birthweight \geq 3.21$
$TOT = 8.9$ $ICU = 6.0$ $TUB = 1.9$
$BefOp \geq 1.5$
$BefOp < 1.5$
$STAT\ Score < 0.55$
$STAT\ Score \geq 0.55$
$TOT =$ $ICU = 8.1$ $TUB = 1.9$
$TOT = 9.4$ $ICU = 5.7$ $TUB = 1.2$
$TOT = 9.4$ $ICU = 5.7$ $TUB = 1.2$
$TOT = 33.4$ $ICU = 22.7$ $TUB = 4.0$

Figure 3.8: Multivariate Tree for Pediatric Hospitalization Time Using Determinant Impurity

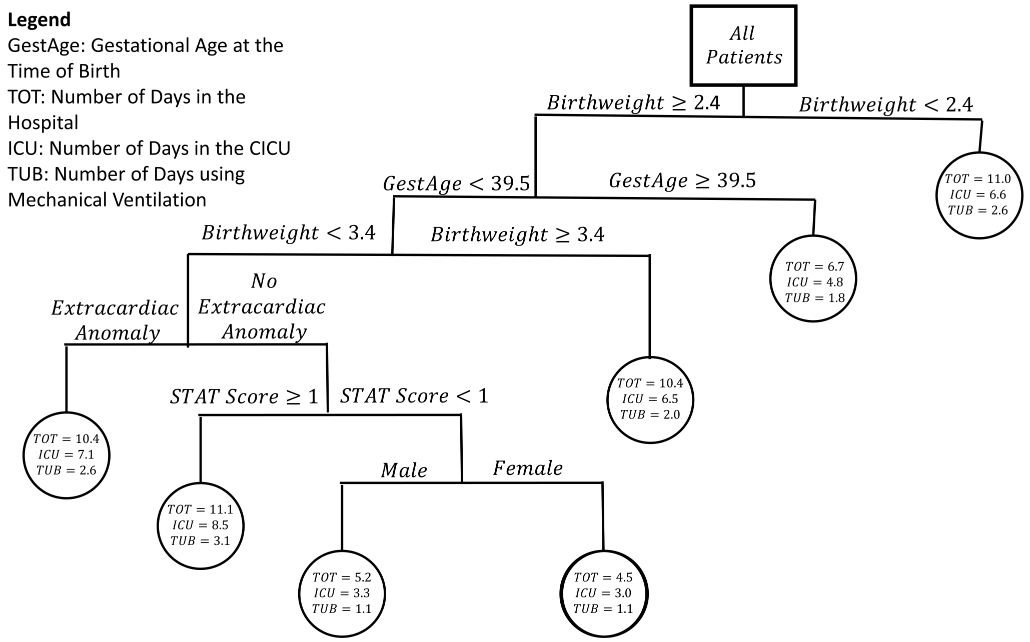
Legend
GestAge: Gestational Age at the Time of Birth
TOT: Number of Days in the Hospital
ICU: Number of Days in the CICU
TUB: Number of Days using Mechanical Ventilation
All Patients
Birthweight ≥ 2.4
Birthweight < 2.4
TOT = 11.0 ICU = 6.6 TUB = 2.6
GestAge < 39.5
GestAge ≥ 39.5
TOT = 6.7 ICU = 4.8 TUB = 1.8
Birthweight < 3.4
Birthweight ≥ 3.4
TOT = 10.4 ICU = 6.5 TUB = 2.0
Extracardiac Anomaly
No Extracardiac Anomaly
TOT = 10.4 ICU = 7.1 TUB = 2.6
STAT Score ≥ 1
STAT Score < 1
TOT = 11.1 ICU = 8.5 TUB = 3.1
Male
Female
TOT = 5.2 ICU = 3.3 TUB = 1.1
TOT = 4.5 ICU = 3.0 TUB = 1.1

Figure 3.9: Multivariate Tree for Pediatric Hospitalization Time Using the Between-Node Mahalanobis Distance

accuracy of the within-node metric trees. The between-node Mahalanobis tree had the best prediction accuracy overall, which was consistent with what was observed in the simulation with large N and moderate r. Each of the multivariate methods remained somewhat simple, each having 7 terminal nodes or less. In comparison to the univariate trees, the presented multivariate methods achieve slightly improved prediction accuracy with a more simple prediction structure.

3.6 Discussion and Conclusion

The improving collection, upkeeping and accessibility of electronic health records (EHR) gives physicians the opportunity to make informed decisions in the clinical setting at an unprecedented rate [1,2,50-53]. Strong prediction of multiple correlated outcomes is vital to make these informed decisions in many biomedical settings. Ideally, statistical methods used to aid clinical decision support are accurate, easy to use and clinically valid. Assessing neuropathy or targeting pediatric heart disease patients most likely to have long hospitalizations are just two clinical scenarios where strong prediction of multivariate outcomes could lead to improved patient care and lower healthcare costs.

In this research, we propose five splitting rules that can be implemented to build multivariate trees. Our first approach uses summary quantities of the empirical covariance matrix at each stage of the tree building process. This allows us to identify which covariate value leads to the greatest reduction in impurity. Specifically, calculating the determinant of the outcome empirical covariance matrix and by calculating the average Mahalanobis distance in each node are the two ways we propose evaluating node impurity. Compared to the previous multivariate tree methods, our proposed impurity functions take into account the correlated nature of the multivariate outcomes and allow the estimated covariance to be different amongst subgroups of the population.

Our second approach builds off the ideas of LeBlanc and Crowley, who built survival trees by maximizing the between node distance at each stage of the splitting process [42-43]. We proposed three distance functions that can be used to build trees in the multivariate setting. While this research focused on the tree growing step, developing a pruning algorithm remains as future work.

Results from the multivariate ensemble methods were mixed, often not improving prediction accuracy compared to the single tree counterparts. This might be because multivariate trees are hitting a ceiling in terms of prediction accuracy. It is also possible that the multivariate nature of the trees, built by balancing variances of multiple outcomes, has given the single tree lessened variability in terms of structure. A nice feature of this result is that we are left with a interpretable, usable tool to make prediction in the clinical setting. Unfortunately, it is unlikely that we can improve prediction accuracy by developing an ensemble. Assessing the performance of ensemble methods under specific scenarios through simulation is left as future work.

The prediction accuracy results of the multivariate methods were very similar in both applications. In practice, we recommend using the results from the presented simulation study to help guide the selection of a specific multivariate tree building metric. In simulation, the methods perform similarly in terms of MSE, however under certain scenarios, specific metrics may be preferable. Amongst the within-node metrics, the trace trees are preferable when $r = 2$ and the Larsen method is preferable when $r = 5$. However, in high-variance and high-correlation scenarios, the determinant and Mahalanobis distance trees are preferable and maintain a simple tree structure. In comparing the between-node metrics, when the sample is small the Mahalanobis distance trees have the best prediction accuracy and when the sample is large the Euclidean distance metrics have the best prediction accuracy.

An important comparison is between the multivariate tree methods and a series of univariate classifiers. In practice, clinicians would have to choose between these two approaches for prediction in the clinical setting. Building many univariate trees generally outperforms the multivariate trees in simulation. These two approaches are not completely comparable: the univariate trees can focus on each outcome separately, allowing them to maximize their performance individually. The multivariate tree methods must leverage performance on an individual outcome to achieve better performance in the overall collection of outcomes. The proposed multivariate tree method had mixed results when compared to multiple univariate trees in our applications. In the single neuropathy trees, the within node Mahalanobis distance and Euclidean distance trees outperformed the univariate trees, however, the series of univariate trees outperformed each of the multivariate trees in the ensemble. In predicting pediatric hospitalization times, all of the multivariate trees except the within node Mahalanobis trees outperformed the 3 univariate trees.

Prediction accuracy (in terms of MSE) is only one aspect to evaluate when comparing tree methods for the prediction of multivariate outcomes. When comparing the univariate and multivariate tree performance, it is important to consider clinical reliability and usability. When predicting many outcomes from the same clinical domain, a clinically reliable method would utilize similar covariates as predictors for each outcome. As was the case in the application for neuropathy, each of the three univariate trees had a different structure and contained some different covariates. The differences in covariates used for prediction in the univariate trees might lead a clinician to believe that the predictive tool is just an artifact of the data and therefore loses generalizability and reliability.

Clinical usability is another important consideration between the proposed multivariate and univariate trees. Prediction of patient outcomes in the clinical setting is a primary goal of the proposed tree based method. Tree based methods are an ideal tool for prediction in the clinical setting since they are easy to use and interpret. As the number of outcomes increases, a series of predictions using univariate trees reduces practical clinical usability. It is much more realistic for a clinician to use a single tree that predicts the entire multivariate outcome.

Development of statistical methods that are not only accurate, but also clinically valid and usable is vital as the use of EHR data to aid in clinical decision making becomes more prevalent. In this research, we offer a clinician-friendly solution for prediction of multivariate outcomes using regression trees. Our proposed multivariate regression trees attempt to bridge the gap between methods with strong prediction performance and real-time clinical decision support.

CHAPTER IV

Improving Patient Prediction Using Patient Specific Data in Tandem with Well Validated Predictive Tools

Electronic Health Records (EHR) possess a powerful breadth of data allowing biomedical and health service researchers alike to solve numerous research questions. Predictive tools resulting from large EHR data sources have been validated for use across many health centers but generally only use 'shallow' patient data that is likely to be commonly collected. Often, data collected locally at specific health care centers contain rich information on continuously monitored physiologic parameters that have the potential to enhance outcome prediction. In this chapter, we propose methods to combine large-scale, shallow EHR data with small-scale, deep patient data to improve prediction. The idea is to perform sequential classification: first using widely available covariates for risk stratification and subsequently refining prediction using deep data for a subgroup of patients. We propose three approaches to select which patients move onto a second stage prediction refinement using deep data. Our approaches select patients with poor predictive properties and intermediate risk based on the first stage classification. At each step of the sequential classification, we use a toolbox of machine learning methods. The predictions from the first two steps are combined to produce a tandem prediction. The methods are illustrated using a pediatric cardiac study.

4.1 Introduction

The improved collection, upkeep and accessibility of electronic health records (EHR) give physicians the opportunity to make informed decisions in the clinic setting at an unprecedented rate [1,2]. Electronic Health Records such as large, multi-center disease registries and nation-wide insurance claim databases provide a powerful breadth of data allowing clinicians and health service researchers alike to solve numerous research questions [8-15].

Predictive tools resulting from large EHR data sources have been validated for use across many health centers and as a result, have strong predictive properties. In practice, the resulting tools are used by clinicians to predict patient outcomes. These large EHR databases typically capture 'shallow' patient data that are commonly collected across a wide spectrum of health centers. In a specific health center, extra patient information is collected in addition to that contained in the large EHR.

Making clinical decisions based on the large-scale (shallow) EHR may miss important attributes that are collected locally at the patient level in a specific health care center. In this paper, we hypothesize that using the rich, locally collected data can improve prediction ability when used in tandem with the initial prediction based on large scale EHR data in certain patient subgroups. Specifically, we develop an approach to use deep (health-center specific) data to complement shallow (large-scale) EHR data with the goal of improving prediction while maintaining practicality.

We hypothesize that using just the large scale EHR-based prediction tools will provide accurate prediction for patients at the extreme ends of the risk spectrum. Generally, there is more heterogeneity and therefore, less prediction accuracy in the intermediate range. This is the group that would benefit from added information using deep data to refine prediction. Building on this, we develop three general approaches for identifying patient subgroups.

Pediatric cardiac critical care providers are often challenged with the equally important but often conflicting goals of minimizing patients exposure to mechanical ventilation and preventing extubation failure [8-11]. Extubation failures have been associated with adverse outcomes including increased duration of hospital stay, cardiac arrest, and mortality [8-11]. Patients that experience extubation failure may also suffer downstream complications such as airway injury, prolonged mechanical ventilation, and the numerous consequences of prolonged exposure to critical care therapies [10]. As such, efforts to reduce extubation failure events may lead to great benefits for patients.

Reliable measures of extubation readiness, while validated in adult patients, remain elusive in pediatric cardiac critical care. Patients in the cardiac intensive care unit (CICU) have heterogeneous pathophysiology. Failure to breathe without assistance from a ventilator can be the result of primary respiratory failure, cardiac failure, or a mixed etiology. Our previous work also demonstrates wide variation in case mix-adjusted extubation failure rates across pediatric CICUs, further suggesting that practice and outcomes vary due to existing knowledge gaps [11]. Physicians and nurses need new prediction tools to help with clinical decision making when assessing children in the CICU for extubation readiness.

Previous research has investigated how patient, disease, and hospital factors associate with the risk of extubation failure using data from a large clinical registry of over 32 institutions from North America (Pediatric Cardiac Critical Care Consortium: PC4) [8-11]. Using traditional regression approaches, previous work identified several patient and disease characteristics that are associated with extubation failure, but failed to develop a tool capable of informing clinical practice at the time of deciding whether a patient is safe to extubate. In the delivery of medical and surgical care, often complex interactions between patient, physician, and hospital factors influence practice patterns [50]. Machine learning methods could generate a decision algorithm based on these complex factors that could lead to reduced extubation failure rates.

It is likely that more granular physiologic data collected during a patient extubation readiness assessment, are crucial for improving the prediction of extubation failure. This is especially true when patients have intermediate risk based on their broad spectrum of clinical characteristics. An innovative software platform currently in use in the CICU at the University of Michigan C.S. Mott Children's hospital captures patient information from 76 cardiology, respiratory and physiologic variables from CICU monitors and devices at 1 minute intervals. This data source allows us the opportunity to study physiologic parameters during the key period when patients are evaluated for extubation readiness. Machine learning methods that utilize large-scale shallow data from the PC4 registry in tandem with small-scale deep physiologic data from CICU monitors hold the possibility of unlocking patterns that even the most experienced clinicians may fail to recognize.

This paper is organized as follows: Section 4.2 describes the methodology for making prediction with large-scale, shallow EHR based data in tandem with small, deep healthcare specific data. In Section 4.3, we illustrate the methodology to make prediction for pediatric patients who have undergone a critical cardiac operation. Lastly, in Section 4.4, we present concluding remarks and discussion.

4.2 Methodology

4.2.1 Tandem Prediction

First Stage Prediction

We begin by introducing some notation to describe the tandem approach. Let n_1, k_1 and n_2, k_2, be the sample size and number of predictors for the shallow (large-sample EHR) and deep (health system specific) datasets respectively. In this setting, $n_1 >> n_2$, $n_2 \subset n_1$ and $k_2 >> k_1$. The goal is to predict the binary outcome, $\mathbf{y}$, using $\mathbf{X}$, the matrix of predictors. Typically, clinicians attempt to predict $\mathbf{y}$ utilizing either $\mathbf{X_1}$, the $n_1 \times k_1$ covariate matrix from the large, shallow EHR or $\mathbf{X_2}$, the $n_2 \times k_2$ covariate matrix from the small, deep patient data. In this method, we propose making a tandem prediction, by first stratifying patients based on risk using $\mathbf{X_1}$, then selecting a subgroup of patients with intermediate risk and poor predictive properties to pass on to the second stage, and lastly, refining prediction for the selected subgroup using $\mathbf{X_2}$.

Motivated by our clinical example in pediatric heart disease, we utilize machine learning (ML) methods to make predictions. Machine learning methods are especially useful in heterogeneous patient populations where standard regression methods may fail to uncover complex patterns in the data. Specifically, we will select a prediction tool, $f{*}_1 : \mathbf{X_1} \rightarrow \mathbf{Y}$, from $\mathcal{F}$, the toolbox of ML methods. The ML toolbox will include: classification and regression trees (CART), Bagging, Random Forests, Boosting, Support Vector Machines (SVM), logistic regression and Naive Bayes [3,4,5,31,32,33,34,56]. Classification and Regression Trees (CART), will be the ideal prediction method to emerge from the ML toolbox. CART are useful statistical learning tools because they allow for intuitive and simple disease classification by recursively partitioning the covariate space. Since CART is built with the goal of increasing node homogeneity, we will be able to easily identify subgroups of patients that are heterogeneous and therefore likely to improve prediction. They are also very popular in the clinical setting because they are easy to use, implement and understand from a clinical perceptive.

The first stage ML method: $f*_1$ is selected based on predictive performance using a 10-fold cross validation. Each potential first stage classifier assigns a predicted outcome probability: $\hat{\mathbf{p_1}} = Pr(f*_1(\mathbf{X_1}) = 1|\mathbf{X_1})$ to each patient. For each ML method, to assess predictive performance we calculated overall prediction accuracy (ACC), area under the ROC curve (AUC), sensitivity (TPR), specificity (TNR), positive predictive value (PPV) and negative predictive value (NPV) based on the cross validation. The classifier that possessed the best predictive attributes under the 10-fold cross validation is used for the first stage prediction.

Second Stage Patient Selection

With the end goal of developing a tool to aid in clinical decision making, we need to balance predictive improvements with simplicity, computational efficiency and usability. There are two important considerations when determining which patients should be selected for prediction refinement. First, the collection, storage and quality control of deep data is presumably more expensive and time intensive. Examples include genetic testing and continuously streamed data (e.g. monitoring of physiologic data). Testing and collecting such data for an entire health system may simply not be feasible. Recommendation for specific patient subgroups, that may benefit most from the additional data collection, will be cost-effective and timely.

Second, we anticipate that patients with very high or low outcome risk from the first stage classifier may already be homogeneous enough and may not have considerable discrimination on the basis of additional deep data. In the event that prediction attributes are similar between the tandem prediction and first stage classification alone, we would prefer the first stage classifier since it is based on a larger, well-validated data source.

We propose three general approaches to determine which patients will be passed along to the second stage prediction. These approaches focus on finding subgroups of patients that have (1) intermediate risk only (2) poor predictive properties only and (3) intermediate risk and poor predictive properties.

1. Patient Selection Based on Risk

In this approach we select patients with intermediate risk to pass on to the second stage. Lower and upper risk thresholds are pre-specified based on clinical input and are denoted τ_L and τ_U. These thresholds can be tuned to select a specific size subgroup for the second stage. Algorithm 1 is described below.

Algorithm 1 Risk Driven Subgroup Selection

```
Select $f_1$ from $\mathcal{F}$ using 10-Fold CV
$\hat{\mathbf{p_1}} = Pr(f_1(\mathbf{X_1}) = 1|\mathbf{X_1})$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \quad s.t. \quad \tau_L < \hat{p}_{1i} < \tau_U\}$
```

2. Patient Selection Based on Predictive Performance

We develop four techniques to find the subgroup of patients with poor predictive properties. In some small sample settings, all patient combinations could be examined to find the subgroup that has the worst predictive attributes. In this approach, the clinician would pre-specify the subgroup size: v, and then calculate the AUC for each ${}_{n_1}C_v$ subgroup combination. The subgroup combination that minimizes the AUC would be chosen for the second stage prediction. In many applications, the sample size is large enough to where calculating the AUC for each patient subgroup of size v is not computationally feasible. This approach is detailed in Algorithm 2.

A second option is to utilize the natural subgroups created by CARTs terminal nodes. If CART is selected as f_1, we can utilize impure terminal nodes to identify subgroups of patients with poor predictive performance. We denote $\mathbf{T}$ as the set of all terminal nodes from $f_1(\mathbf{X_1})$ and $i(h)$ as the impurity for a given node h. The patients that fall into the terminal node with the largest impurity would be passed on for second stage prediction. The specifics of this approach are described in Algorithm 3.

The third approach involves finding patients that when added to a candidate patient subgroup, reduces the calculated AUC for the subgroup in which the patient was added. The ideal upper-bound second stage sample size is pre-specified and denoted n_*. We first rank order patients based on the predicted outcome probabilities from the first stage prediction: $\hat{\mathbf{p_1}}$. We denote the rank-ordered probabilities as $\{\hat{p}_{11}, \hat{p}_{12}, ..., \hat{p}_{1n_1}\}$ where $\hat{p}_{11}$ is lowest predicted risk and $\hat{p}_{1n_1}$ is the highest predicted risk. Next, we define $\hat{\mathbf{p}}_{\mathbf{1(j)}} = \{\hat{p}_{1i}\}_{i=1,...,j}$ as the subset of patients of size j, with the lowest $\hat{p_1}$ and similarly, $\hat{\mathbf{p}}_{\mathbf{1(-j)}} = \{\hat{p}_{1i}\}_{i=n_1-j,...,j}$ as the size j patient subset with the largest predicted risk ($\hat{p_1}$). Similarly, we denote $\mathbf{X_{1(j)}}$ and $\mathbf{y_{1(j)}}$ as the covariate matrix and outcome for the j lowest risk patients based on the first stage classifier. We calculate the $AUC(f(\mathbf{X_{1(j)}}), \mathbf{y_{1(j)}})$, for sequentially increasing patient subgroups beginning with the lowest risk patient until the entire sample is included. Next, we subset the patients, that when added to the sequentially larger test group, reduced the calculated AUC. We repeat this process for the the backwards direction, starting with those with highest risk, and increasing subgroups sequentially until the full sample is included. The patients that reduced the AUCs when added in both directions are taken as the first candidate subgroup. Instead of the intersection of the forward and backward selected groups, one could consider the union as well. Next, the patients in the candidate subgroup are re-ordered by risk and sequential AUCs are re-calculated. This process is iterated until a sufficiently small ($\leq n_*$) group of patients is selected. This approach is described in Algorithm 4.

A fourth approach is described in Algorithm 5. This approach is similar to Algorithm 4, except patients are randomly ordered, instead of rank-ordered by probability. Then, we sequentially add patients based on the random order, calculate AUCs and track which patients caused AUC decreases. This process iterates several times and we determine the poor predictive subgroup by selecting patients that reduced AUC most often as a candidate subgroup. We denote B as the number of iterations, and for each patient, we can add up the number of times that patient decreases subgroup AUC. We introduce ψ, as a tuning parameter cutoff, to which we compare the proportion of times (out of B), that a patient decreased subgroup AUC when sequentially added.

Algorithm 2 Prediction Driven Subgroup Selection- 1

```
Select $f_1$ from $\mathcal{F}$ using 10-Fold CV
$v \Leftarrow$ Pre-Specified Subgroup Size
$S_v \Leftarrow$ Set of all $v$ size subsets in $n_1$
Select $s \in S_v \quad s.t. \quad min(AUC(S_v)) = AUC(s)$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \in s\}$
```

Algorithm 3 Prediction Driven Subgroup Selection- 2

```
$f_1 \Leftrightarrow CART$
Define $\mathbf{T}$ as the set of all terminal nodes created by $f_1$
Define $i(h)$ as the impurity function for a given node $h$
Select $T_* \in \mathbf{T} \quad s.t. \quad max(i(\mathbf{T})) = i(T_*)$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \in T_*\}$
```

3. Patient Selection Based on Predictive Performance and Risk

In order to identify patients with poor prediction attributes, an AUC threshold γ, is pre-specified. We recommend working with expert clinicians to elicit the γ value that would be considered acceptable for the specific clinical outcome (e.g. 0.65, 0.90). Rank-ordering the predicted outcome probabilities from the first stage prediction

Algorithm 4 Prediction Driven Subgroup Selection- 3

```
Select $f_1$ from $\mathcal{F}$ using 10-Fold CV
$n_* \Leftarrow$ Pre-Specified ideal sample size
$n' = n_1$
$\mathbf{X_1'} = \mathbf{X_1}$
while $(n_* \leq n')$:
    Define $\mathbf{X_{2forward}} = \{\mathbf{X}_{2j} \forall j \in \{2, ..., n_1\}$ where $AUC(f_1(\mathbf{X'_{1(j)}}), \mathbf{y'_{1(j)}}) - AUC(f_1(\mathbf{X'_{1(j-1)}}), \mathbf{y'_{1(j-1)}}) < 0\}$
    Define $\mathbf{X_{2backward}} = \{\mathbf{X}_{2j} \forall j \in \{2, ..., n_1\}$ where $AUC(f_1(\mathbf{X'_{1(-j)}}), \mathbf{y'_{1(-j)}}) - AUC(f_1(\mathbf{X'_{1(-j+1)}}), \mathbf{y'_{1(-j+1)}}) < 0\}$
    $\mathbf{X_2} = \mathbf{X_{2forward}} \cap \mathbf{X_{2backward}}$
    Define $\mathbf{X_{1forward}} = \{\mathbf{X}_{1j} \forall j \in \{2, ..., n_1\}$ where $AUC(f_1(\mathbf{X'_{1(j)}}), \mathbf{y'_{1(j)}}) - AUC(f_1(\mathbf{X'_{1(j-1)}}), \mathbf{y'_{1(j-1)}}) < 0\}$
    Define $\mathbf{X_{1backward}} = \{\mathbf{X}_{1j} \forall j \in \{2, ..., n_1\}$ where $AUC(f_1(\mathbf{X'_{1(-j)}}), \mathbf{y'_{1(-j)}}) - AUC(f_1(\mathbf{X'_{1(-j+1)}}), \mathbf{y'_{1(-j+1)}}) < 0\}$
    $\mathbf{X_1'} = \mathbf{X_{1forward}} \cap \mathbf{X_{1backward}}$
    $n' = length(\mathbf{X_2})$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \in \mathbf{X_1'}\}$
```

Algorithm 5 Prediction Driven Subgroup Selection- 4

```
Select $f_1$ from $\mathcal{F}$ using 10-Fold CV
$B \Leftarrow$ Number of Iterations
$\mathbf{Q} \Leftarrow n_1 \times B$ results matrix
for (b in 1:B):
    $\mathbf{X_1'} = reorder(\mathbf{X_1})$
    $res = I\{AUC(f_1(\mathbf{X'_{1(-j)}}), \mathbf{y'_{1(-j)}}) - AUC(f_1(\mathbf{X'_{1(-j+1)}}), \mathbf{y'_{1(-j+1)}}) < 0\}$
    $\mathbf{Q}_b = unorder(res)$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \quad s.t. \quad \sum_{b=1}^{B} \mathbf{Q}_i/B \geq \psi\}$
```

($\hat{\mathbf{p_1}}$), allows us to identify those patients with intermediate risk. We utilize the same notation from Algorithm 4.

Our strategy is to find the intermediate risk group with poor predictive attributes by increasingly moving j to create sequentially larger and larger patient subgroups with the j lowest (or highest) risks based on the first stage predictions. For each value of j, we calculate $AUC(f(\mathbf{X_{1(j)}}), \mathbf{y_{1(j)}})$ for the sequential subgroups of patients. We increase j until that subgroup loses its strong predictive performance ($AUC(f(\mathbf{X_{1(j)}}), \mathbf{y_{1(j)}}) < \gamma$). The sequential subgroups start at the lowest risk (or highest risk) in which the outcomes are not pure. For example, if the patient with the 25th smallest risk is the first that experiences the event, j will start at 25. We introduce ω as a tuning parameter, where low values can give stricter and larger subgroups and high values result in more liberal, smaller subgroups. The ω can be thought of as a leniency parameter, allowing the calculated AUC to fall below γ exactly ω number of times before we create our lower and upper bound subgroups. We introduce ω for two reasons. First, when j is small, the calculated AUCs can be noisy (since we are building an ROC curve based on such few patients). This noise may lead us to incorrectly find a large intermediate subgroup of patients. Secondly, when data is expensive to collect, we provide some control on the size of the intermediate subgroup that would be recommended for second stage prediction. After the ωth consecutive subgroup where $AUC(f(\mathbf{X_{1(j)}}), \mathbf{y_{1(j)}}) < \gamma$, we define $\tau_L = p_{1(j-\omega)}$ as the lower risk threshold. This process is repeated, beginning with the highest risk patients, resulting in $\tau_U = p_{1(-j+\omega)}$ as the upper risk threshold. The calculated thresholds result in $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \quad s.t. \quad \tau_L < \hat{p}_{1i} < \tau_U\}$, that are then used for the second stage prediction. The details of this approach are described in Algorithm 6.

Algorithm 6 AUC and Risk Driven Subgroup Selection

```
Select $f_1$ from $\mathcal{F}$ using 10-Fold CV
$auc \leftarrow \gamma$
$\omega' \leftarrow \omega$
$j = 1$
$m \leftarrow n_1$
while ($j \leq m$ & $\omega' > 0$):
    $auc_{test} = AUC(f_1(\mathbf{X_{1(j)}}), \mathbf{y_{1(j)}})$
    if $auc_{test} \geq auc$:
        $\omega' \leftarrow \omega$
    else:
        $\omega' = \omega' - 1$
    $j = j + 1$
$\tau_L = \hat{p}_{1(j-1-\omega')}$

$\omega' \leftarrow \omega$
$j = 1$
$m \leftarrow n_1$
while ($j \leq m$ & $\omega' > 0$):
    $auc_{test} = AUC(f_1(\mathbf{X_{1(-j)}}), \mathbf{y_{1(-j)}})$
    if $auc_{test} \geq auc$:
        $\omega' \leftarrow \omega$
    else:
        $\omega' = \omega' - 1$
    $j = j + 1$
$\tau_U = \hat{p}_{1(-j+1+\omega')}$
Define $\mathbf{X_2} = \{\mathbf{X}_{2i} \quad \forall i \quad s.t. \quad \tau_L < \hat{p}_{1i} < \tau_U\}$
```

Subgroup Selection in the 'Cheap' Deep Data Scenario

There exists certain clinical scenarios in which the deep data collection actually has minimal costs. Cheap deep data sources could include text frequency from clinical notes or examining ICD-9 codes from medical claims data. In each of these examples, data is already collected, thus reducing the costs greatly of using 'additional' information. An alternative approach in a setting where additional data collection costs are minimal, would involve giving a second stage prediction for all patients, using $\hat{\mathbf{p}}_\mathbf{1}$, as an extra covariate in the second stage classifier. As a comparison, we also include the initial predicted risk as an additional covariate for the second stage classifiers based on the selected subgroups.

Second Stage Prediction Refinement

In the second stage of the tandem approach, we refine the prediction for the selected patients using $\mathbf{X_2}$. Once again we consider the toolbox of ML approaches, $\mathcal{F}$. In the second stage prediction, we choose $f{*_2} : \mathbf{X_2} \rightarrow \mathbf{Y}$, from $\mathcal{F}$ based on prediction results calculated from a 5-fold cross validation. The resulting prediction made from the second stage refinement is denoted: $\hat{\mathbf{y_2}} = f *_2 (\mathbf{X_2})$ with probability, $\hat{\mathbf{p}}_\mathbf{2} = Pr(f *_2 (\mathbf{X_2}) = 1|\mathbf{X_2})$. For notational convenience, we define the complete tandem prediction as $T = f_2(f_1(\mathbf{X}))$ where final prediction is given as: $\hat{\mathbf{p}} = Pr(T(\mathbf{X}) = 1)$.

4.2.2 Making Real-Time Prediction

To ensure this method can be easily implemented for real time clinical decision support, we detail the process of making prediction for a new patient i. First, an initial prediction is given as $\hat{p_{1i}}$ using f_1. The clinician would have pre-determined whether to select patients for prediction refinement using risk, predictive performance or both. The clinician would need to pre-specify tuning parameters such as what would be considered satisfactory predictive performance (γ) or what is considered high and low risk (τ_U and τ_L). Then using the selected algorithm, we determine whether patient i will be moved to second stage classification.

Algorithms 1, 2 and 6 result in well-defined risk cutoffs or terminal nodes that would be used for subgroup selection. Conversely, Algorithms 3, 4 and 5 select subgroups of patients with the worst prediction attributes, based on already collected data. Unfortunately, when determining whether a new patient falls into the subgroups selected by Algorithms 3, 4 and 5: we do not yet know the full outcome information.

To circumvent this issue in practice, we treat whether the patient was included in the refinement subgroup as a binary outcome. We find the group of patients that patient i is most similar to in terms of known covariates using a k-nearest neighbors algorithm. For a new patient i, if the k-nearest patients are included in the prediction refinement subgroup (based on Algorithms 3, 4 or 5), then she/he is also passed on to second stage prediction. Final prediction for these patients will take into account both $\hat{\mathbf{p}}_\mathbf{1}$ and $\hat{\mathbf{p}}_\mathbf{2}$. We propose two general approaches to produce a final tandem prediction: $\hat{p_i}$. Consider producing a tandem predicted probability as:

1. $\hat{p_i}$=$\hat{p_{2i}}$

2. $\hat{p_i}$=$(r_1\hat{p_{1i}}+r_2\hat{p_{2i}})$

where r_1 and r_2 are weights scaled between (0,1) with $r_1+r_2 = 1$. One common choice for the weights would be $r_1 = r_2 = 0.5$ which would equally weight the two predicted probabilities. A second choice would be to weight the predicted probabilities based on the proportion of patients that were passed to second stage: c, resulting in $r_1 = 1-c$ and $r_2 = c$. A third choice of weights can be derived through the fold change in AUC improvement for patients that are given prediction refinement. We define $d = AUC(\mathbf{X_2})/AUC(\mathbf{X_1})$ as the AUC ratio for the deep and shallow data. Then, define $r_1 = 1/(1+d)$ and $r_2 = d/(1+d)$. We once again assess the overall performance of the tandem approach by examining AUC, ACC, PPV, NPV, TPR, TNR based on a 5-fold cross validation.

4.2.3 Making Prediction with Complex Data

When using EHR data to make prediction in the clinical setting, we often have data that is so complex that standard methods are unable to produce a prediction performance needed to make decisions in the clinical setting. In this section we propose an approach to deal with the complex longitudinal data in our motivating application.

The deep data collected in the CICU at the University of Michigan contains continuously streamed cardiology, respiratory and physiologic variables collected every minute during the time period of the extubation readiness evaluation. Specifically we collect patient systolic blood pressure (SBP), diastolic blood pressure (DBP), mean blood pressure (mean BP), respiratory rate (RR), heart rate (HR), SpO2 levels, FiO2 levels, SpO2/FiO2 Ratio, mean airway pressure (MAP), delivered volume (DV), dynamic compliance (DYN), positive end expiratory pressure (PEEP), and central venous pressure (CVP).

The complex longitudinal nature of this data makes prediction challenging for three reasons. First, the number of covariates may be larger than our sample size, depending on the subgroup selection. Second, this type of continuously streamed information can be quite noisy. Finally, since the patients are each followed for a variable length of time, the data is very unbalanced. If one were to try and extract the entire physiologic profile, the analysis may be plagued by missing data. Dimension reduction techniques such as Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Generalized Discriminant Analysis (GDA) can deal with dimension reduction, but lose the easy clinical interpretability that is often necessary for clinicians to make patient decisions.

To deal with the high-dimensionality of the deep data, we propose several approaches, each time adding features to the previously derived covariate set. The idea is to 'magnify' in on the patient covariate profiles at different resolutions in an attempt to capture as much relevant information as possible while maintaining simplicity. Our objective is to balance computational complexity with predictive accuracy, generalizability and clinical interpretability in order to develop a tool for clinical decision-making. Towards this end, we want to 'zoom' in to gain as much information as possible (and therefore predictive ability), while maintaining a level of clinical applicability.

A simple dimension reduction technique that retains the interpretability of the covariates is to take summary measures of the covariate trajectories for each patient. For example, making a decision based on SBP slope reduces the dimension of the SBP measurements to a single covariate while maintaining clinical interpretability. For each of variables measured during the extubation readiness evaluation, we extract basic profile information (e.g. minimum, maximum, standard deviation, slope and average piece-wise slope) for each patient profile. This approach of feature selection yielded 53 summary covariates for each patient.

We further this technique by fitting polynomial regression models with increasing order for each patient profile. The regression coefficients are then extracted for use as inputs in the ML based analysis for the second stage prediction. Since our physiologic deep data source is a relatively small sample, as an exploratory analysis, we create indicators for each of these effects based on whether the patient effect was above the average effect for the full data.

4.3 Application: Predicting Extubation Failure for Pediatric Patients Following Critical Cardiac Operation

4.3.1 Data Collection and Background

Improving the quality of care given to pediatric patients is an important goal of health service researchers and clinicians alike. In this section, we demonstrate our tandem approach to predict extubation failure amongst 174 pediatric patients that have undergone a critical cardiac operation with measured physiological attributes. Utilizing the PC4 registry, we identified 12,244 pediatric patients that had undergone a critical STS (Society of Thoracic Surgeons) defined heart operation from 32 North American institutions. During CICU follow-up, 8.6% of the patients experienced an extubation failure. In the first stage of the tandem prediction, we attempt to predict patient extubation failure based on 22 covariates available to clinicians at the time patients enter the CICU following the operation.

Patient covariates in the PC4 registry included characteristics of the CICU visit (whether there was an operation included with the ventilator run, number of previous extubations, length of time on the ventilator, whether the patient was ventilated during the operation), patient airway characteristics (whether the patient had an airway anomaly, non-airway anomaly or chromosomal abnormality), patient clinical characteristics and demographics (patient length, gender, age and weight), three patient risk scores and lastly, patient comorbidities (whether the patient had hypertension, stroke, sepsis, acute decompensated heart failure, cardiac arrest, ECMO, vocal cord dysfunction or paralyzed diaphragm).

4.3.2 First Stage Prediction, Risk Stratification and Patient Refinement Selection

Table 4.1 presents results from the first stage prediction. Specifically, we present prediction performance based on a 10-fold cross validation for each method in the machine learning toolbox. Boosting performed the best with an AUC of 0.689. The variable importance plot for the boosted classifier is displayed in Figure 4.1. The length of time spent on the ventilator, whether the patient had a paralyzed diaphragm and patient weight were the three variables that resulted in the largest average impurity decrease. Based on the predicted probabilities from the boosted

Table 4.1: 10-Fold Cross Validation Prediction Results for First Stage Classification using PC4 Registry

First Stage Prediction using PC4	AUC	ACC	NPV	PPV	TPR	TNR
CART	0.5	0.914	0.914	0.4	0	1
Logistic Regression	0.677	0.914	0.915	0.367	0.00948	0.998
Random Forest	0.66	0.912	0.915	0.199	0.0116	0.996
Bagging	0.51	0.914	0.914	0	0	1
Boosting	0.689	0.913	0.915	0.318	0.0146	0.997
SVM	0.533	0.914	0.915	0.525	0.00565	0.999
Naive Bayes	0.655	0.817	0.927	0.163	0.272	0.869

classifier, we select patients to pass to the second stage.

1. Patient Selection Based on Risk

We first select intermediate risk patients by pre-specifying the lower and upper risk cutoffs (Algorithm 1). Setting the upper and lower risk cutoffs to be the 20th and 80th percentiles, the corresponding predicted probability cutoffs are 0.116 and 0.267 respectively. The extubation failure rates in the low, intermediate and high risk groups were 1.9%, 6.9% and 20.3% respectively. Of the 174 patients for which we have second stage data, 105 fall into the intermediate risk group.

2. Patient Selection Based on Predictive Performance

Next, we find candidate subgroups with poor predictive performance in the first stage classifier. Due to the large sample in this application, we are unable to examine the out-of-sample predictive performance of all patient combinations (Algorithm 2). Since the CART classifier based on the PC4 data yielded just a single node stump, we are also unable to identify heterogeneous patients using the tree (Algorithm 3).

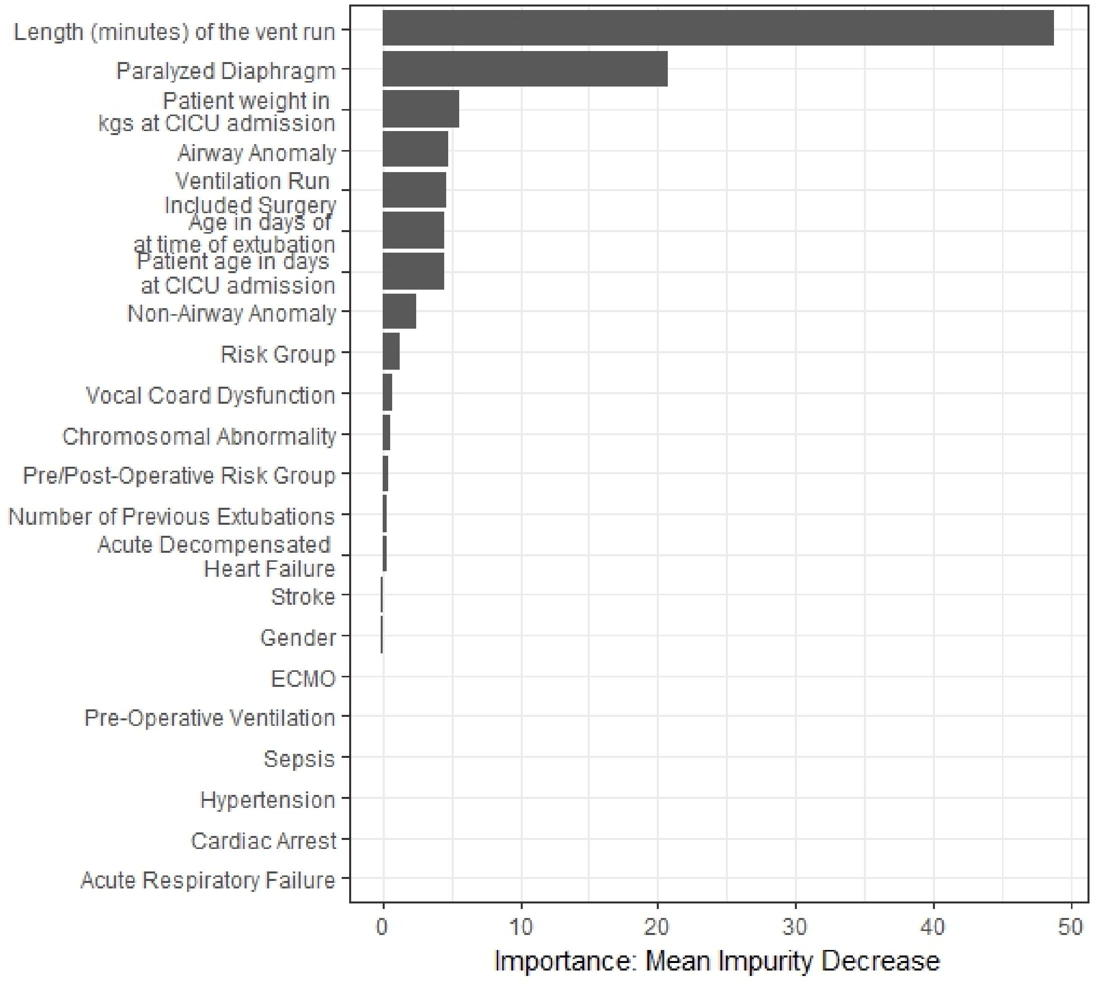

Figure 4.1: Variable Importance for Boosted Classifier based on PC4

Alternatively, we use the calculated AUCs from the sequentially larger subgroups based on risk to select the subgroup that decreased subgroup AUC when added (Algorithm 4). The AUCs from the sequentially larger subgroups are displayed in Figure 4.2. There were 10,528 patients when added, reduced the calculated AUC while sequentially moving forward with larger risks. When sequentially moving backward with smaller risks, there were 1,038 patients that when added, reduced the calculated AUC. The were 10,680 patients (151 failures) that reduced AUC when added in the forward or backward sequential calculations which included 155/174 patients with physiologic measurements. The 10,680 patients were re-ordered and sequential AUCs were re-assessed. There were 9,338 patients (36 failures) that reduced AUCs in the second iteration which resulted in 143/174 patients. Unfortunately only 2/155 and 1/143 of the proposed subgroup had failures. So while we may have selected a subgroup with poor predictive properties, this subgroup selection did not result in a usable subgroup for second stage prediction.

In the final approach to find a subgroup of patients with poor predictive attributes, we randomly ordered the 12,244 patients, created sequentially larger subgroups and calculated AUC differences (Algorithm 5). We repeated this process ten times ($B = 10$) and then added the number of times (out of 10) the AUC decreased when that patient was added to the subgroup. There were 1,212, 2,981, 3,468 and 2,500 patients that caused AUC decreases 10, 9, 8 and 7 times out of ten times respectively. We choose $\psi = 0.8$ and selected the 7,661 patients as the subgroup that resulted in AUC decreases at least 80% of the time. The subgroup had an 8.6% failure rate and resulted in 107/174 patients for second stage prediction.

3. Patient Selection Based on Predictive Performance and Risk

To select patients based on both predictive ability and risk (Algorithm 6), we set the predictive threshold as $\gamma = 0.65$ and initially set the tuning parameter to $\omega = 122$, or 1% of the sample size. After rank ordering the patients by the risk calculated from the boosted classifier, the lowest calculated risk for a patient with an extubation failure was 0.077, which was the 666th patient. The highest calculated risk for a patient without a failure was 0.695, which was the 12,243rd patient. The starting point for the forward and backward portions of the algorithm are therefore the 666th and 12,243rd patient respectively. The sequentially calculated AUCs used in finding the lower and upper risk cutoffs are displayed in Figure 4.2.

The tuning parameter ω was varied to create candidate subgroups with differing sizes. By varying levels of ω we can evaluate the different subgroup sizes for the 174 patients with both PC4 and CICU physiologic data. Not each increase in ω results in a meaningful subgroup for this application, since it will not always result in changing which patients of the 174 are recommended for the second stage. Keeping $\gamma = 0.65$ fixed, we find 6 different subgroup splits at $\omega = 1, 2, 18, 53, 368, 424$ resulting in subgroup sizes of 174,151,150,146,130,129. The ω can be chosen to select an ideal subgroup size in terms of cost. If data collection costs are minimal, we could potentially evaluate the predictive performance of the generated potential subgroups as a hold out set in the first stage classifier. In this application, we stay with our initial assignment of $\omega = 122$.

Algorithm 6 found the risk cutoffs to be 0.120 and 0.418 respectively resulting in 2,709 patients in the low risk group, 9,345 patients in the intermediate risk group and 190 patients in the high risk group. The extubation failure rate was 39.5% in the high risk group, 9.9% in the intermediate risk group and 1.9% in the low risk group. Patients with predicted risk between 0.120 and 0.418 are selected for prediction refinement. Of the 174 patients with second stage data, 24 fell into the

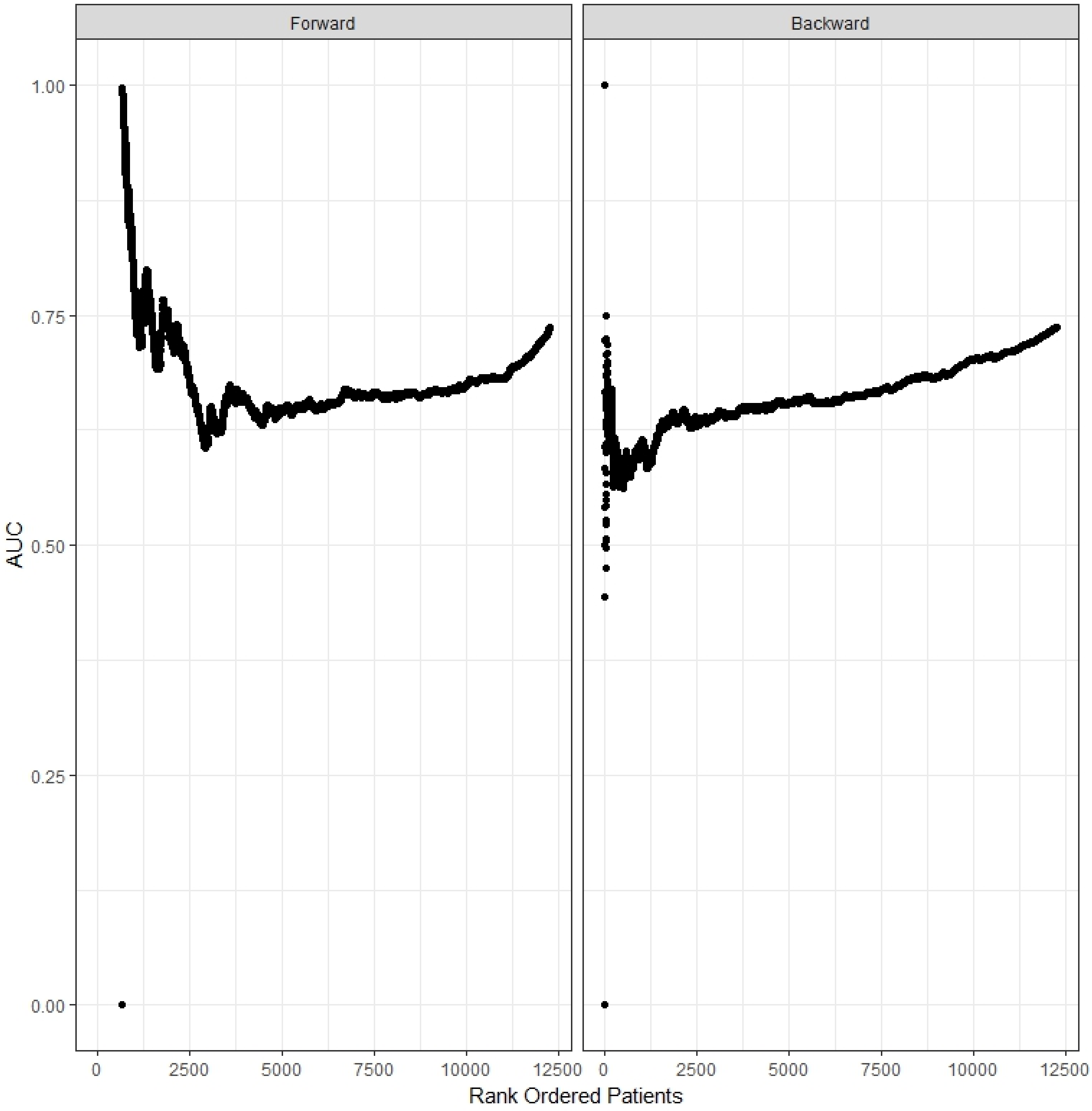

Figure 4.2: Sequentially Calculated AUCs

Table 4.2: Summary of Subgroup Selection Algorithms

	Patients in Subgroup		Patients with Physiologic Data Data in Subgroup		Prediction Results from Boosting Algorithm Using Holdout Set					
Algorithm	N	% Failure	N	% Failure	AUC	ACC	NPV	PPV	TPR	TNR
Risk and Prediction Performance	9345	9.9%	146	12.3%	0.529	0.863	0.875	0	0	0.985
Risk Only	7345	6.9%	105	7.6%	0.341	0.924	0.924	0	0	1
Prediction Performance (Algorithm 5)	10680	1.4%	152	1.3%	0.57	0.974	0.987	0	0	0.987
Prediction Performance (Algorithm 6)	7661	8.6%	107	9.4%	0.688	0.897	0.906	0	0	0.99

low risk group (0/24 had extubation failure) and 4 fell into the high risk group (1/4 had an extubation failure). The remaining 146 patients were passed on to the second stage.

The six algorithms resulted in three potential subgroups recommended for prediction refinement. Table 4.2, summarizes the subgroups and prediction attributes when used as a holdout testing set for the first stage boosted classifier. The predictive performance for the 2/3 subgroups was significantly weaker than the full data. The calculated AUCs were 0.341, 0.688 and 0.529 for the subgroups selected based on risk, prediction and both risk and predictive attributes respectively. These testing set predictive performances using the PC4 data alone represent a baseline that can be potentially improved with second stage deep data.

4.3.3 Second Stage Prediction

We evaluate predictive performance for each of the ML based classifiers on second stage data. Table 4.3 lists the predictive performance of each classifier for the selected patient subgroups. We compare the second stage prediction results to the baseline first stage results (with the subgroups used as a hold out set). First, we compare to the results of the selected subgroup using just the first stage prediction. There is clear improvement in prediction accuracy using the second stage classifiers for the subgroups selected by risk and the risk/predictive attributes. Each of the classifiers improved predictive performance for the risk (stage one AUC=0.341) and risk/predictive subgroup (stage one AUC=0.529). The subgroup selected by predictive attributes (algorithm 6) was less effective: none of the classifiers improved performance for the 107 patients compared to the first stage classifier alone.

The best performing classifier for the predictive and risk-based patient subgroup, was the Naive Bayes classifier with an AUC of 0.63. Boosting was the best second stage classifier for the risk based subgroup (AUC=0.64). Random forest resulted in the best predictive performance (AUC=0.69) for the predictive based subgroup. The variable importance plots for three selected subgroups are displayed in Figures 4.3-4.5. There were somewhat different covariates responsible for the largest average impurity decrease in the three subgroups. However, minimum SpO2 was amongst the most important covariates in each group. The prediction accuracies for the second stage prediction based on the full $n = 174$ cohort are slightly improved compared to the 3 subgroups which is unsurprising, as the likely homogeneous patients are included in the full data.

Next, we attempted to incorporate more information from the CICU physiologic database by including the coefficients from increasing polynomial trends as features for the ML classifiers. We also included results from classifiers where each feature was an indicator to whether the patient had above average values for each of the extracted polynomial features. The results from each of the subgroups is included in Table 4.4. Prediction accuracy did not consistently improve as the number of features increased.

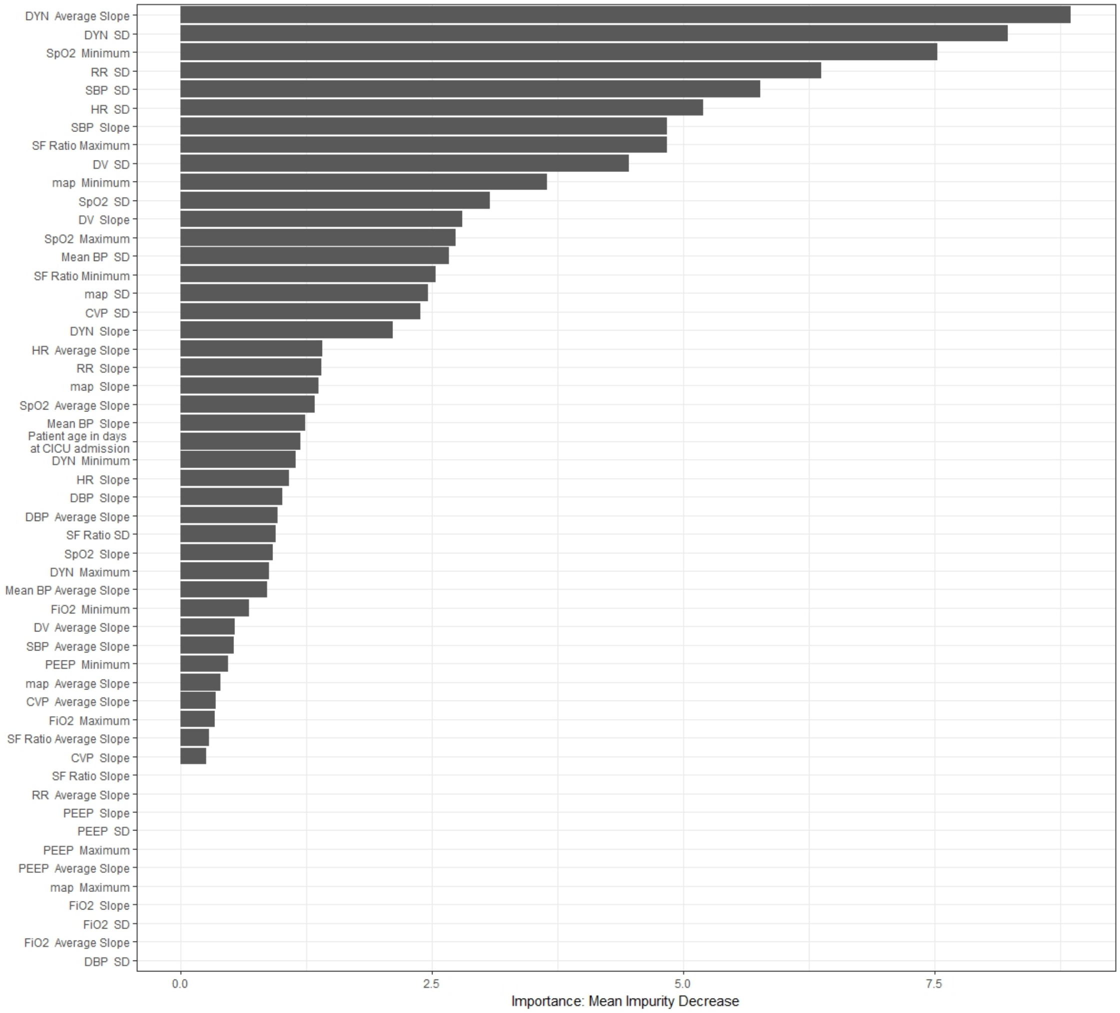

Figure 4.3: Variable Importance for Random Forest based on CICU Physiologic Data for Risk Subgroup

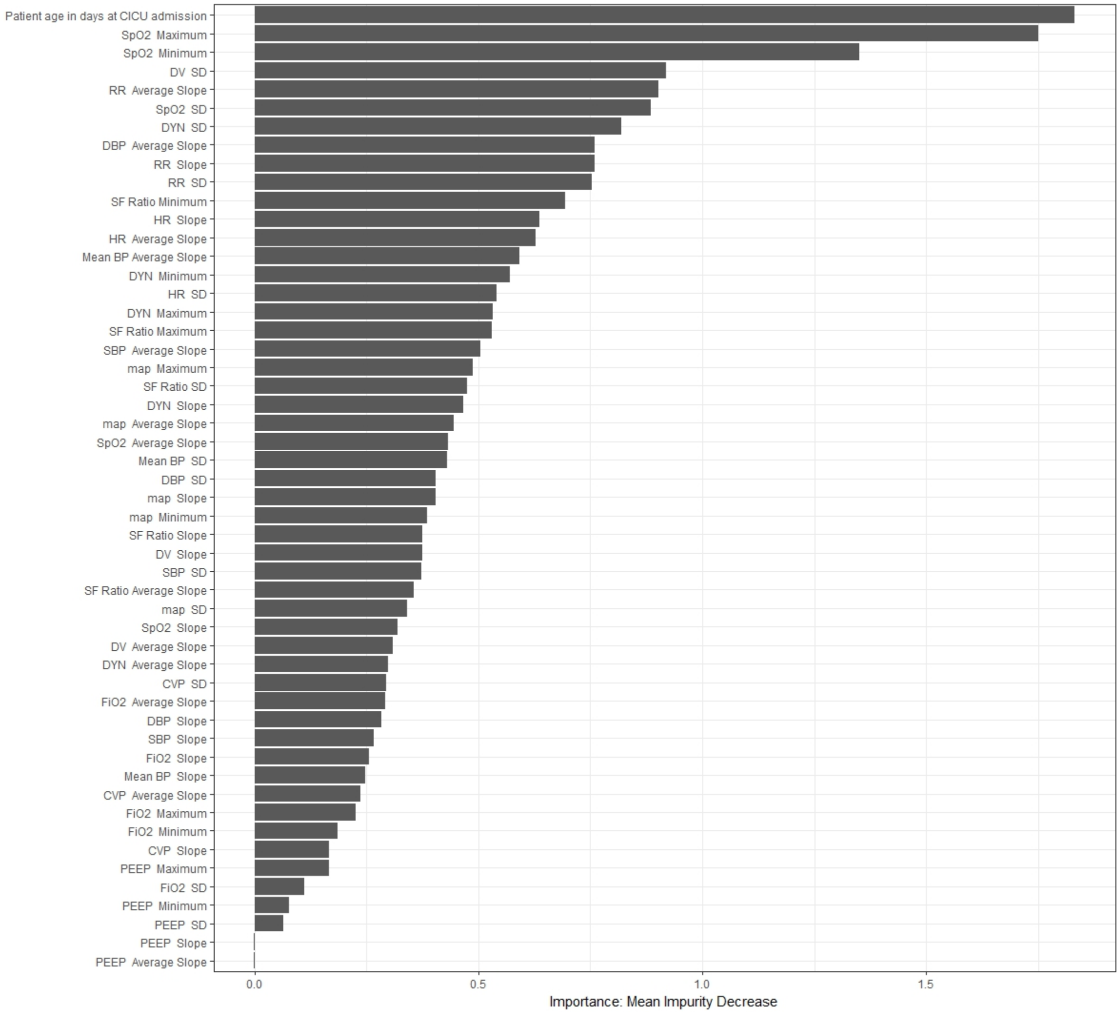

Figure 4.4: Variable Importance for Random Forest based on CICU Physiologic Data for Predictive Subgroup

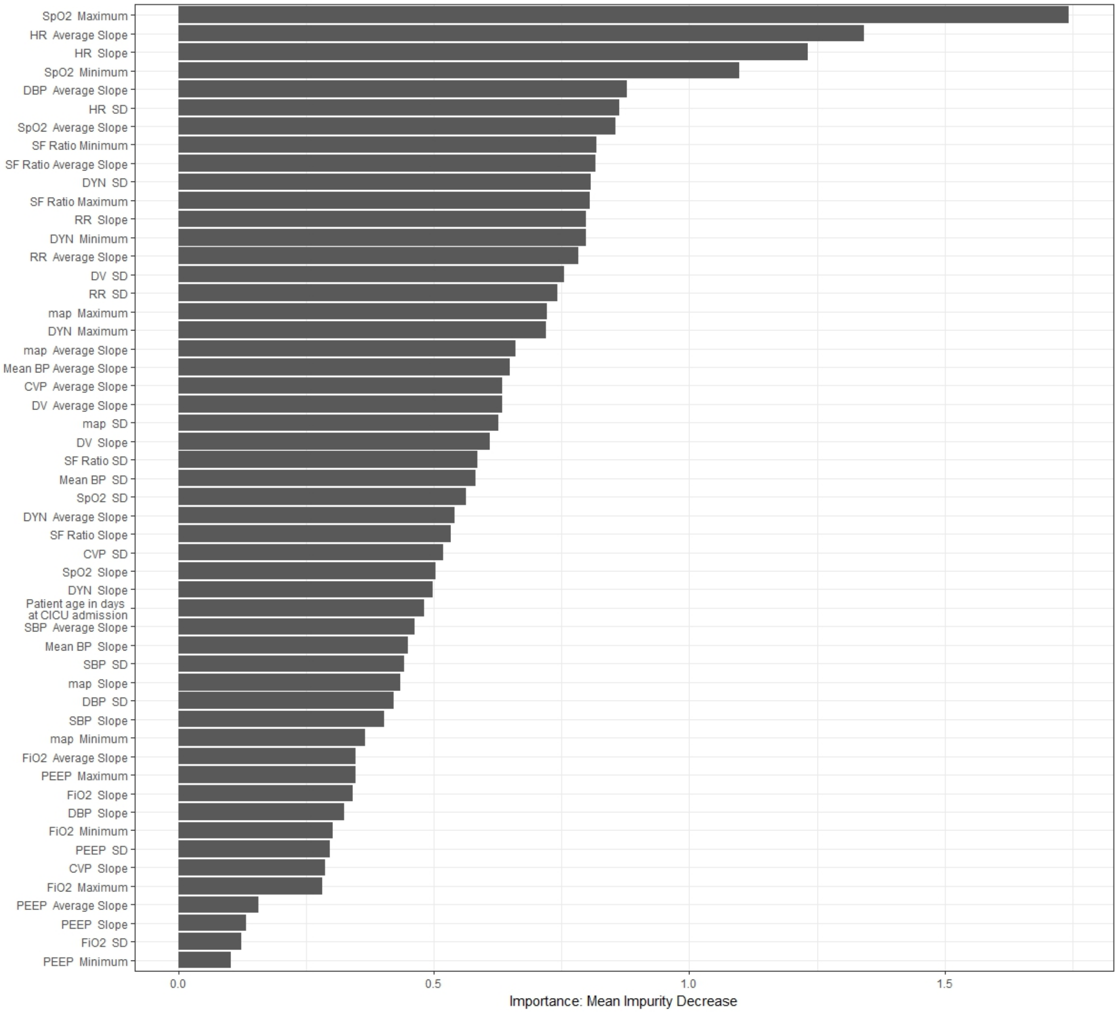

Figure 4.5: Variable Importance for Random Forest based on CICU Physiologic Data for Predictive and Risk Subgroup

Table 4.3: Second Stage Prediction Results

		AUC	ACC	NPV	PPV	TPR	TNR
Subgroup Based on Risk and Prediction	Boosting	0.602	0.835	0.873	0.200	0.000	0.955
	Bagging	0.624	0.870	0.877	0.000	0.000	0.933
	CART	0.573	0.829	0.896	0.187	0.330	0.915
	Random Forest	0.628	0.871	0.877	0.000	0.000	0.992
	Support Vector Machines	0.546	0.877	0.877	0.200	0.000	1.000
	Nave Bayes	0.634	0.698	0.886	0.093	0.290	0.784
Subgroup Based on Risk	Boosting	0.644	0.914	0.923	0.200	0.000	0.990
	CART	0.680	0.895	0.942	0.350	0.400	0.950
	Random Forest	0.603	0.924	0.924	0.200	0.000	1.000
	Support Vector Machines	0.295	0.914	0.923	0.000	0.000	0.990
	Nave Bayes	0.550	0.581	0.925	0.143	0.367	0.605
	Logistic Regression	0.501	0.705	0.911	0.040	0.100	0.751
Subgroup Based on Prediction	Boosting	0.603	0.897	0.906	0.000	0.000	0.990
	Bagging	0.537	0.935	0.935	0.000	0.000	1.000
	CART	0.500	0.907	0.907	0.000	0.000	1.000
	Random Forest	0.687	0.888	0.906	0.200	0.000	0.982
	SVM	0.615	0.906	0.906	0.000	0.000	1.000
	Nave Bayes	0.459	0.786	0.920	0.111	0.200	0.834
	Logistic Regression	0.508	0.766	0.925	0.105	0.333	0.816
Full Data ($n = 174$)	Boosting	0.676	0.868	0.891	0.200	0.050	0.966
	Bagging	0.631	0.891	0.891	0.500	0.000	1.000
	CART	0.559	0.850	0.891	0.200	0.067	0.948
	Random Forest	0.643	0.891	0.891	0.000	0.000	1.000
	SVM	0.476	0.890	0.890	0.000	0.000	1.000
	Nave Bayes	0.486	0.607	0.884	0.099	0.300	0.656
	Logistic Regression	0.567	0.758	0.891	0.132	0.150	0.834

Table 4.4: Prediction Results with Polynomial Effect Covariates

			Random Forest		Boosting	
	Degree Polynomial Fit	Number of Covariates	AUC	AUC with Indicators	AUC	AUC with Indicators
Approach 1 Subgroup (n=146)	1	54	0.628	0.605	0.602	0.699
	2	91	0.558	0.597	0.512	0.605
	3	130	0.547	0.631	0.497	0.647
	4	182	0.498	0.526	0.588	0.64
	5	265	0.474	0.496	0.5	0.699
Approach 2 Subgroup (n=105)	1	54	0.603	0.667	0.644	0.459
	2	91	0.303	0.505	0.356	0.323
	3	130	0.291	0.379	0.5	0.423
	4	182	0.231	0.453	0.393	0.3
	5	265	0.328	0.387	0.408	0.367
Full Data (n=174)	1	54	0.643	0.681	0.676	0.691
	2	91	0.632	0.660	0.652	0.604
	3	130	0.612	0.657	0.571	0.555
	4	182	0.571	0.633	0.569	0.633
	5	265	0.547	0.614	0.566	0.622

4.3.4 Final Tandem Prediction

While there is some obvious improvement in two of the three subgroups, we next examine the performance of the full tandem classifier in predicting extubation failure. The calculated AUCs from the tandem prediction are displayed in Table 4.5. We compare the prediction accuracies of the final tandem predictions (in the last 6 rows of the table) to that of just using the first stage prediction alone (in row two of the table). The subgroups selected by approach 1 and 3 saw significant improvements in prediction accuracies. Conversely, there was no prediction improvement when refining prediction using approach 2. The risk based subgroup increased AUC from 0.607 to 0.682 in the best scenario. The prediction/risk subgroup increased AUC from 0.607 to 0.647. The AUC improvements were less dramatic when assigning second stage prediction for each patient, increasing AUCs from 0.607 to 0.639 in the best case scenario. Using just physiologic data for all 173 patients, we have an AUC of 0.634. Thus, our tandem approach improved predictive ability compared to using either EHR alone. Additionally, the fact that the best case tandem AUC increase (when using the full data) was only 0.634 to 0.639 may support our hypothesis that we are unable to add discrimination to the already homogeneous population within the full data.

When fully adjusting for first stage prediction, there is a prediction gain in each of the three subgroups. Interestingly, there was no boost in prediction for the full data when adjusted for the initial predicted risk for the second stage classifier.

4.3.5 Application Conclusions

Improving our ability to identify pediatric patients most likely to experience an extubation failure can reduce risk of cardiac arrest, mortality and length of hospital-

Table 4.5: Tandem Prediction Results

	Subgroup Based on Risk and Prediction ($n_2 = 146$)	Subgroup Based on Risk ($n_2 = 105$)	Subgroup Based on Prediction ($n_2 = 107$)	Full Data ($n_2 = 174$)
$\hat{p_{1i}}$ (holdout: n_2)	0.529 AUC_1	0.341 AUC_1	0.688 AUC_1	0.607 AUC_1
$\hat{p_{1i}}$ (holdout: 174)	0.607	0.607	0.607	0.607
$\hat{p_{2i}}$ (5-Fold Cross Validation n_2)	0.604 AUC_2	0.662 AUC_2	0.604 AUC_2	0.676 AUC_2
$\hat{p_i}=\hat{p_{2i}}*\hat{p_{2i}}$	0.539	0.647	0.604	0.650
$\hat{p_i}=0.5*\hat{p_{1i}}+0.5*\hat{p_{2i}}$	0.647	0.663	0.589	0.637
$\hat{p_i}=\hat{p_{2i}}$	0.618	0.682	0.580	0.634
$\hat{p_i}=\frac{174-n_2}{174}*\hat{p_{1i}}+\frac{n_2}{174}*\hat{p_{2i}}$	0.634	0.668	0.585	0.637
$\hat{p_i}=\frac{1-\frac{AUC_2}{AUC_1}}{1+\frac{AUC_2}{AUC_1}}*\hat{p_{1i}}+\frac{\frac{AUC_2}{AUC_1}}{1+\frac{AUC_2}{AUC_1}}*\hat{p_{2i}}$	0.647	0.671	0.590	0.639
$\hat{p_2}$ adjusting for $\hat{p_1}$	0.665	0.723	0.605	0.599

ization. This application demonstrates how broad-scale EHR data can be combined with institution specific deep data for risk prediction for extubation failures. The collection and quality control of continuously streamed physiologic data is expensive and therefore, not collected on each patient. Using the proposed algorithms, we derived 3 potential patient subgroups that were passed to the second stage. In this application, selecting patients based on their predicted risk allowed us to take a deeper dive into the complex features that make a patient more or less likely to have a extubation failure. In the best case setting, we had an AUC improvement of 0.075.

To understand the impact of the tandem prediction, we make two key comparisons. First, we compare the prediction attributes of the tandem classifier to that of just using the PC4 registry or the CICU physiologic database alone for the 174 patients with full data. Using PC4 alone resulted in $AUC = 0.607$ and that for the CICU physiologic alone was $AUC = 0.634$. Using Algorithm 6 to select patients, we obtained an AUC of 0.647 and using Algorithm 1, we achieved AUC of 0.682. Second, we assess whether selecting specific patients for prediction refinement resulted in better predictive attributes compared to that of giving all patients a second stage prediction. Utilizing the PC4 registry to identify the heterogeneous second stage subgroup resulted in improved predictive attributes and required less data collection (0.647 and 0.682 vs. 0.639). Adding effects from polynomial regression models for covariates as features for the second stage classifiers did not consistently improve predictive ability for any of the subgroups.

The results of the tandem prediction could be implemented quite easily at health centers within the PC4 collaborative. As part of the extubation readiness trial, we could assign a patient an initial risk based on our boosted classifier. Then, if the patient fell into intermediate risk group, the physiologic data during the readiness trial would be collected and summarized. The second stage classifier would then utilize the physiologic data to give a final prediction. This tandem approach will allow us to identify patients with elevated risk of extubation failure more accurately, thereby improving downstream patient outcomes.

4.4 Discussion

Developing prediction tools that can be used in real-time has the potential to improve healthcare delivery and decrease healthcare costs. Ideally, such prediction tools should be accurate, practical and easy to use. In our proposed tandem approach, we first obtain risk prediction using a well-validated, large sample EHR that contains only shallow patient covariates. Next, we utilize the initial risk to select patients that have intermediate risk and/or poor predictive properties using one of six proposed algorithms. Finally, the selected patients are given an updated prediction based on a second stage ML based classifier that utilizes deep, physiologic data. The resulting prediction tool may offer clinicians risk cutoffs to inform when deep patient data should be collected and assessed to refine risk prediction. While adding more information to aid in predictions can be valuable in certain clinical scenarios, our method is built on the premise that targeting patient subgroups where there is maximum heterogeneity and potential for improvement, will have the greatest impact.

The collection and quality control of deep health system specific data can be expensive and time consuming. Deep data examples include genetic testing, continuously streamed ICU data, metabalomics, lipidomics or functional MRI. A useful feature of the proposed subgroup selection algorithms is their ability to be tuned; giving the user the ability to select from subgroups with differential sample sizes. In real-time prediction, one would determine whether a patient lands in our selected subgroup and then perform further testing to collect the deep data needed to refine prediction. The framework of selecting patients for additional testing based on their initial predicted risk make it possible to upscale the tandem approach widely.

Statisticians must attempt to balance computational and quantitative complexity with predictive accuracy, generalizability and clinical meaning. Frequently, the complexity of the methodology must be increased to meet the complexity of the underlying outcome etiology. This is especially true in our tandem approach with the second stage deep data. When data is complex, we need 'magnify' our view when extracting covariates from the deep data. In this paper, we propose extracting polynomial effects from regression models for each patients physiologic data trajectories to use as features in the second stage prediction. The level of 'magnification' should be chosen based on predictive ability through a cross validation in the training data to ensure we do not overfit by extracting covariates that are too patient specific.

We demonstrated our method by predicting extubation failure for pediatric patients who have undergone a critical heart operation. We identified 3 potential subgroups that could be used for prediction refinement. There was a significant improvement in prediction accuracy when comparing the tandem prediction to that from the first stage or second stage prediction alone. Additionally, the prediction attributes were improved for the tandem approaches that refined prediction for specific patient subgroups compared to making second stage prediction with the full data. Though our application did not allow us to utilize certain algorithms to select patient subgroups, they may be useful in other applications. Due to our small sample of patients with collected deep data, we were unable to gain consistent additional information by including polynomial effects as predictors. In larger sample applications, extracting important trends through polynomial trajectories may result in increased prediction accuracy.

An important consideration that must be made on a case-by-case clinical basis is determining what constitutes meaningful increases in predictive ability. In some clinical scenarios with extreme adverse outcomes, any predictive improvement will be 'worth' the increased time, costs and analysis of deep patient data. Using a training cohort of patients with already collected data, AUC improvement could be estimated under different tuning parameter settings. Under this approach, clinicians can estimate what predictive improvements could be gained based on differential subgroup sizes. If only a small AUC increase is observed, tuning parameters can be chosen to select a smaller subgroup. Once the desired AUC improvement and corresponding subgroup size is selected, the patient selection parameters could be 'locked in' for future prediction.

Our tandem approach could easily be extended to other outcome types. For continuous outcomes, predictive ability could be assessed using MSE and the patient subgroup could be selected by rank-ordering the predicted outcome. The ML literature for survival and clustered outcomes is less developed. Implementing the tandem approach to different outcomes remains as future work.

Due to the tandem nature in the data, we expect instability in the structure of the developed classifier. Instability in building a classifier occurs because small perturbations of the data can have very dramatic effects on the structure of created classifier (especially with tree-based methods). If instability occurs in the first stage prediction, the classifier could alter which patients are delivered to second stage prediction. Therefore, in certain cases, the instability caused by small perturbations in the data could have multiplicative consequences. To overcome potential instability in our tandem approach, ensemble methods could be implemented. Specifically, we could develop an ensemble in the first stage prediction to provide added stability in terms of subgroup selection. A final second stage subgroup could be chosen based on the group of patients that were selected to the subgroup most often in the ensemble. Important experimentation needs to be implemented to asses whether or not in each stage of the ensemble, the specific ML method will be locked in or allowed to change with each bootstrapped sample. Exploration of tandem prediction ensembles are left as future work.

In this paper, we demonstrate how our tandem approach can be implemented to two EHR data sources, however this method could be generalized to make prediction using more than two data sources. The idea for a generalized-tandem prediction is that after each sequential prediction, we would continue to re-assess which patients would most benefit from further classification. Combining more than two data sources could be applied in a patient diagnostic setting, where clinicians iteratively order more testing in order to assign a patient diagnosis. With each additional test, prediction is updated and the subgroup selection algorithms determine whether more testing should be ordered, or a diagnosis assigned.

CHAPTER V

Discussion

Although the availability of data to inform clinical decisions is at an all time high, predictive tools often lack certain attributes to aid in real-time clinical decision making. Statistical methods are needed to produce *practical*, *reliable*, *clinical valid* and *interpretable* decision making tools. In this dissertation, we proposed three statistical learning methods with the above attributes based on data from electronic health records. The ultimate goal of each method was to improve patient care and reduce unnecessary healthcare costs.

In the first chapter, we proposed a new classification tool: SVM-CART, that combined features of SVM and CART. This work was motivated by clinical scenarios where neither SVM nor CART alone could address complex features of the data reasonably, such as scenarios when there were different disease-exposure mechanisms across subgroups of the population. Through simulations, we demonstrated that under these scenarios, SVM-CART outperformed SVM or CART alone in terms of prediction accuracy. Furthermore, to improve prediction accuracy and stability of the SVM-CART classifier, we developed an ensemble. We proposed a method to select the most representative single classifier from the ensemble as a practical tool for decision making. Results from simulations were mixed. Therefore, we do not recommend using SVM-CART as a complete replacement for SVM or CART alone. Statisticians must take advantage of the knowledge from subject matter experts to determine if the hypothesized disease exposure mechanism differs across subgroups of the population. This was the case with our clinical application in neurology, where there were different neuropathy-metabolic syndrome relationships amongst patients with different glycemic-gender status. In addition to the modest improvements in prediction accuracy (*reliability*), our SVM-CART classifier also had enhanced *clinical interpretability* for the neuropathy application. Compared to CART and SVM alone, the SVM-CART classifier was relatively simple and allowed us to make prediction based on glycemic groups separately. In future work, it may be interesting to develop a general form of this composite classifier using other machine learning methods. Additionally, SVM-CART applied to large EHR data, will have the ability to implement more intricate kernel functions in the SVM portion of the classifier.

In Chapter 3, we proposed methods to build regression trees and ensembles for multivariate outcomes. We developed two general approaches to tree growing where goodness of split was evaluated based on maximizing (1) within-node homogeneity and (2) between node separation. Within-node homogeneity was measured using the average Mahalanobis distance and determinant of the empirical covariance matrix. Between node separation was assessed using Mahalanobis and Euclidean distances. In general, the prediction accuracies of trees resulting from these goodness of split measures were similar, with some variation under certain scenarios. Specifically, when true variance was large, the within-node Mahalanobis and determinant metrics resulted in trees with the best prediction accuracy. Ensemble of multivariate trees yielded mixed results. We illustrated the proposed methods using two applications to: (1) predict patients who were at risk for long lengths of stay in various phases of hospitalization and (2) predict nerve conduction measures to diagnose neuropathy. Compared to univariate trees, our multivariate trees were better in terms of clinical validity and applicability, while maintaining similar predictive attributes.

As more and more EHR data sources become available, it will be important to leverage data from multiple sources to enhance prediction. In Chapter 3, we developed an approach to make tandem predictions using large-scale shallow data and deep physiologic data from two separate EHRs. First, patients were assigned an initial risk from a classifier based on the large (shallow) EHR data. Then, we developed a framework to find a subgroup of patients that were most likely to benefit from a second stage prediction refinement. The selected subgroup was given a second stage prediction based on the small (deep) EHR. Final tandem prediction was based on combining predictions from both steps. We illustrated our method to predict extubation failure for pediatric patients that have undergone a critical cardiac operation. We used the PC4 registry to assign patients an initial risk. For a selected subgroup of these patients, we utilized continuously streamed data on physiologic variables to update risk prediction. Extending this tandem approach to continuous and survival outcomes is an area of future research. Additionally, it would be interesting to extend our approach to other clinical applications. Common shallow data sources include insurance claims. Using our approach, such data complemented with word frequencies from clinical notes, metabalomic, lipidomic or data from genetic testing has the potential to yield tools for classification. Finally, it would be interesting to see our tandem approach extended to more than two EHR sources; each time re-evaluating whether patients would benefit from additional data (and therefore additional testing).

As high complexity EHR-based data becomes more prevalent, it will be important to develop methods that creatively address nuances of the specific dataset. An example of this was in Chapter 1, where the development for SVM-CART was motivated from the inability of SVM or CART to produce a desired prediction tool for implementation in specific disease setting. Methods should be data-adaptive and require careful incorporation in terms of tuning parameters to obtain prediction tools with maximum potential impact.

Development of these methods require an intricate balance between *practical* applicability, *reliability* and *clinical validity*. In this dissertation we developed three methods attempting to optimize this balance. In each chapter, we proposed methods that improved *reliability* compared to existing methods. In Chapters 2 and 3, we developed tree-based methods to maximize *practical* applicability while improving *clinical validity* compared to the existing methods. In Chapter 4, we proposed subgroup selection approaches that takes into account time and financial considerations to improve the *practical* applicability of the tandem approach. Our overarching goal was to develop statistical learning methods that keep this balance in mind, to improve patient outcomes and reduce unnecessary healthcare costs.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Cimino JJ. “Improving the electronic health record: getting what we wished for”. *Journal of the American Medical Association.* 2013;309:991992.

2. Khairat S, Coleman GC, Russomagno S, Gotz D. “Assessing the status quo of EHR accessibility, usability, and knowledge dissemination”. *eGEMs: Generating Evidence & Methods to improve patient outcomes.* 2018;6:9

3. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone,C.J. *Classification and Regression Trees.* Belmont, California: Wadsworth. 1984.

4. Zhang, H. and Singer, B. *Recursive Partitioning in the Health Sciences.* Springer: New York. 1999.

5. Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning.* Springer: New York. 2001.

6. Cutler, A., Cutler, D.R. and Stevens, J.R. “Tree-based methods”. *High-Dimensional Data Analysis in Cancer Research*, 2009; 24, 123-140.

7. Banerjee, M.. “Tree-based model for thyroid cancer prognostication”. *The Journal of Clinical Endocrinology & Metabolosim.* 2014;99(10),3737-3745.

8. Gaies M., Donohue J.E., Willis G.M., Kennedy A.T., Butcher J., Scheurer M.A., Alten J.A., Gaynor J.W., Schuette J.J., Cooper D.S., Jacobs J.P., Pasquali S.K.,

Tabbutt S. “Data integrity of the Pediatric Cardiac Critical Care Consortium (PC4) clinical registry”. *Cardiology in the Young*. 2016;26(6):1090-1096

9. Gaies M., Cooper D.S., Tabbutt S., Schwartz S.M., Ghanayem N., Chanani N.K.. Costello J.M., Thiagarajan R.R., Laussen P.C., Shekerdemian L.S., Donohue J.E., Willis G.M., Gaynor J.W., Jacobs J.P., Ohye R.G., Charpie J.R., Pasquali S.K., Scheurer M.A. “Collaborative Quality Improvement in the Cardiac Intensive Care Unit: Development of the Paediatric Cardiac Critical Care Consortium (PC4)”. *Cardiology in the Young*. 2015;25(5):951-957

10. Tabbutt S, Schuette J, Gaynor JW, Ghanayem N, Jacobs JP, Alten JA, Dimick JB, Zhang W, Donohue JE, Pasquali S, Banerjee M, Cooper D, Gaies M. “A Novel Model Demonstrates Variation in Case Mix Adjusted Mortality in Pediatric Cardiac Intensive Care Units after Cardiac Surgery: A First Step to Disentangling Surgical from CICU Quality of Care”. *Pediatric Critical Care Medicine*. 2018.

11. Gaies M, Werho DK, Zhang W, Donohue JE, Tabbutt S, Ghanayem NS, Scheurer MA, Costello JM, Gaynor W, Pasquali SK, Dimick JB, Banerjee M, Schwartz SM. “Duration of postoperative mechanical ventilation as a quality metric for pediatric cardiac surgical programs”. *Annals of Thoracic Surgery*. 2018;105:615-621.

12. Kolli S, Reynolds E, Chakrabarthi A, Banerjee M, Nallamothu B. “Hospital-Level Differences in Use of Do Not Resuscitate Orders in PaWith Pneumonia, Acute Myocardial Infarction, and Congestive Heart Failure in California.”. *Circulation: Cardiovascular Quality and Outcomes*. 2017;10 Supplement 3.

13. Healy, MA, Reynolds E, Banerjee M, Wong SL. “Lymph Node Ratio Is Less

Prognostic in Melanoma When Minimum Node Retrieval Thresholds Are Not Met." *Annals of surgical oncology*. 2017;24(2):340-346.

14. Callaghan BC, Xia R, Reynolds E, Banerjee M, Burant C, Rothberg A, Pop-Busui R, Villegas-Umana E, Feldman E. "Better diagnostic accuracy of neuropathy in obesity: A new challenge for neurologists.". *Clinical Neurophysiolgy*. 2018;129:654-662.

15. Callaghan BC, Gao L, Li Y, Zhou X, Reynolds E, Banerjee M, Ji L. "Diabetes and obesity are the main metabolic drivers of peripheral neuropathy". *Annals of clinical and translational neurology*. 2018;5(4):397-405.

16. Vapnik, V.. "The Nature of Statistical Learning Theory". *Data Mining and Knowledge Discovery*. 1995.

17. Boser, B.E., Guyon, I.M. and Vapnik, V.N. "A Training Algorithm for Optimal Margin Classifiers". *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. 1992.

18. Hung, F. and Chiu, H. "Cancer subtype prediction from a pathway-level perspective by using a support vector machine based on integrated gene expression and protein network". *Analysis in Cancer Research*, 2017; 141, 27-34.

19. Zhang, J., Xu J., Hu X., Chen, Q., Tu L., Huang J. and Cui J.. "Diagnostic Method of Diabetes Based on Support Vector Machine and Tongue Images". *BioMed Research International*, 2017.

20. Xu L., Krzyzk A. and Suen C.Y. "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition". *IEEE Transactions of Systems, Man and Cybernetics*. 1992;22(3):418-435

21. Zhu M., Philpotts D., Sparks R.S. and Stevenson, M.J. “Approach to Combining CART and Logistic Regression for Stock Ranking”. *The Journal of Portfolio Management.* 2011;38:100-109

22. Guo H.M., Shyu Y.I. and Chang H.K. “ Combining logistic regression with classification and regression tree to predict quality of care in a home health nursing data set”. *Studies in Health Technology and Informatics.* 2006;122:891

23. Loh WY. “Regression Trees with Unbiased Variable Selection and Interaction Detection”. *Statistics Sinica.* 2002;12:361-386

24. Kim H. and Loh WY. “Classification Trees with Unbiased Multiway Splits”. *Journal of the American Statistical Association.* 2001;96:598-604

25. Chan KY. and Loh WY. “LOTUS: An Algorithm for Building Accurate and Comprehenisble Logistic Regression Trees”. *Journal of Computational and Graphical Statistics.* 2004;13(4):826-852

26. Zeileis A., Hothorn T. and Hornik K. “Model-Based Recursive Partitioning”. *Journal of Computational and Graphical Statistics.* 2008;17(2):492-514

27. Dusseldorp E., Conversano C. and Van Os BJ. “Combining and Additive and Tree-Based Regression Model Simultaneously: STIMA”. *Journal of Computational and Graphical Statistics.* 2010;19(3):514-530.

28. Seibold H., Hothorn T. and Zeileis A. “Generalised Linear Model Trees with Global Additive Effects”. *Conference Proceedings.* 2017.

29. Lee YD., Cook D., Park JW. and Lee EK. “PPtree: Projection Pursuit Classification Tree”. *Electronic Journal of Statistics.* 2013;7:1369-1386

30. Su X., Tsai CL., Wang H., Nickerson D. and Li B. “Subgroup Analysis via Recursive Partitioning”. *Journal of Machine Learning Research.* 2009;10:141-158

31. Breiman, L. “Bagging predictors”. *Machine Learning.* 1999;24, 123-140.

32. Breiman, L. “Random forests”. *Machine Learning.* 2001;45, 5-32.

33. Ishwaran, H., Blackstone, E.H., Pothier, C.E., and Lauer, M.S. “Relative risk forests for exercise heart rate recovery as a predictor of mortality”. *Journal of the American Statistical Association.* 2004;99, 591-600.

34. Quinlan, J. “Bagging, boosting, and C4.5”. *Proceedings Thirteenth American Association for Artificial Intelligence National Conference on Artificial Intelligence.* Menlo Park, CA., AAAI Press. 1996;725-730.

35. Banerjee, M., Ding, Y. and Noone, AM. “Identifying Representative Trees from Ensembles”. *Statistics in Medicine.* 2012;31(15):1601-16.

36. Bharucha N.E., Bharucha A.E. and Bharucha E.P. “Prevalence of peripheral neuropathy in the Parsi community of Bombay”. *Neurology.* 1991;41(8):1315-1317. 591-600.

37. Savettieri G., Rocca W.A., Salemi G., Meneghini F.,Grigoletto F., Morgante L., Reggio A., Costa V., Coraci M.A. and Di Perri R. “Prevalence of diabetic neuropathy with somatic symptoms: a door-to-door survey in two Sicilian municipalities”. *Neurology.* 1993;43(6):1115-1120.

38. Callaghan B.C., Xia R., Reynolds E., Banerjee M., Rothberg A.E. and Burant C.F. “Association between metabolic syndrome components and polyneuropathy in an obese population”. *JAMA Neurology.* 2016; 73(12):1468-1476

39. Callaghan, B.C., Xia, R., Banerjee, M.,de Rekeneire N., Harris T.B., Satterfield S., Schwartz A.V., Vinik A.I., Feldman E.L. and Strotmeyer E.S. "Metabolic syndrome components are associated with symptomatic polyneuropathy independent of glycemic status". *Diabetes Care*. 2016; 39(5):801-807

40. De'Ath, G. "Multivariate Regression Trees: A New Technique for Modeling Species-Environment Relationships". *Ecology*. 2002;83(4):1105-1117

41. Larsen, D., Speckman, P.L. "Multivariate Regression Trees for Analysis of Abundance Data". *Biometrics*. 2004;60(2):543-549

42. Segal, MR. "Regression trees for censored data". *Biometrics*. 1988;35-47.

43. LeBlanc, M. & Crowley, J. "Survival trees by goodness of split". *Journal of the American Statistical Association*. 1993;88(422):457-467.

44. Loh, W.Y., Zheng, W. "Regression Trees for Longitudinal and Multiresponse Data". *The Annals of Applied Statistics*. 2013;7(1):495-522

45. Wilks, S.S. "Certain Generalizations in the Analysis of Variance". *Biometrika* 1932;24:471-494.

46. Wilks, S.S. "Multidimensional Statistical Scatter.". *Contributions fo Probability and Statistics, I. Olkin et al., ed. Stanford University Press*. Stanford, Calif. 486-503

47. Wilks, S.S. "Mathematical Statistics". John Wiley and Sons, New York.

48. Wilks, S.S. "Muldimensional Statistical Scatter". *Collected Papers,Contributions to Mathematical Statistics, T.W. Anderson, ed. John Wiley and Sons* New York, NY. 1967:597-614

49. Mahalanobis, PC. "On the Generalized Distance in Statistics". 1936

50. Noone AM, Banerjee M. “Machine Learning Methods for Cancer Diagnosis and Prognostication”. *Computational Methods in Biomedical Research*. 2008;77101.

51. Johnson AEW, Ghassemi MM, Nemati S, Niehaus KE, Clifton DA, Clifford GD. “Machine learning and decision support in critical care”. *Proceedings of the IEEE*. 2016;104:444466.

52. Deo RC. “Machine learning in medicine”. *Circulation*. 2015;132:1920-1930.

53. Ambale-Venkatesh B, Yang X, Wu CO, Liu K, et al. “Cardiovascular Event Prediction by Machine Learning The Multi-Ethnic Study of Atherosclerosis”. *Circulation*. 2017;121:10921101.

54. Banerjee M, Filson C, Xia R, Miller DC. “Logic regression for provider effects on kidney cancer treatment delivery”. *Computational and Mathematical Methods in Medicine*. 2014.

55. Abdolell M, LeBlanc M, Stephens D, Harrison R. “Binary partitioning for continuous longitudinal data: categorizing a prognostic variable”. *Statistics in Medicine*. 2002;21(22):33953409.

56. Hand DJ, Yu K. “Idiot’s Bayesnot so stupid after all?”. *International statistical review*. 2001;69(3):385-398.