ABSTRACT

AGOR, JOSEPH KAPENA. Feature Selection and Score Development Methods with Health Care Applications. (Under the direction of Osman Y. Özaltın).

The applications of data driven models in the operations research industry have flourished over the past 30 years. Due to the advancement of data housing technologies, larger data sets are being accumulated. While this is an excellent resource for researchers, problems appear in the identification what is "important" in these extensive data frames. Even after the extraction of those critical features and the solving of data driven models, the issue of representing the output in an interpretable fashion for decision makers arises. This dissertation presents frameworks and methodologies to address (a) the problem of feature selection in prediction models and (b) the development of scoring systems to assist decision makers in health care.

We begin by presenting a framework used in the development of a score to capture physician workload in an effort balance workload among provider teams when triaging patients into a hospital. An optimization model is implemented to develop the score and a simulation is built to validate the use of the score. Our results demonstrate that if the proposed score were to be used to determine which hospital unit to assign an incoming patient to, then workload would not only be better balanced amongst provider teams, but would decrease overall. While this score is unique in that it provides a representation of information from the physicians point of view, many of the current scores used in medical practice are severity of illness scoring systems representing information from a patient's point of view. Therefore, the remainder of the dissertation is centered around the study of these types of scoring systems.

In the development of severity of illness scores used to track patient acuity, the value of missing information has not been throughly addressed. Therefore, we study the value of missing information in machine learning models used to predict patient outcomes. Furthermore, we quantify this value in the use of severity of illness scoring systems. Our results indicate that there is clinical value in the knowledge that information is missing and along with appropriate imputation, this knowledge improves predictive power.

When prediction models are used to construct severity of illness scores, the features are assumed to already be known. However, the selection of which features should contribute to the score is not extensively explored. We propose a bilevel programming approach to feature selection for prediction models. Due to the computational complexities associated with bilevel programs, we develop a tailored genetic algorithm as a solution approach. We implement this model in three

separate case studies demonstrating that the bilevel approach will identify those features which are *most important* for use in the prediction.

The current state of the art in the development of data driven scoring systems, specifically those in health care, is to take weights that are generated by statistical learning models such as logistic regression and round them to obtain integer point values for the scores. However, this is could potentially eliminate important variables and reduce calibration by shifting scores to extremes. It is also well known in the optimization community that rounding to get integer solutions usually leads to sub-optimality. Therefore, we propose a mixed integer programming framework for the development of severity of illness scores. To validate the use of our proposed method, we apply the framework to construct a score that can be used to track the acuity of patients who are susceptible to sepsis and compare it to some of the current scores in the literature. We find that our model produces an *interpretable* and accurate score compared to others in the current literature.

This work contributes to the fields of operations research in health care and machine learning. We demonstrate how optimization models, specifically bilevel and mixed integer programming models, can be used to develop data driven solution tools. While most of this work is applied in a health care setting, we believe that the frameworks and models presented in this dissertation are applicable in many other domains that utilize large data sets to learn features and build prediction models. Furthermore, if problems require the results of models to be presented in a concise way (e.g. as a singular score), then one could use the methods developed in this dissertation to do so.

Feature Selection and Score Development Methods with Health Care Applications

by
Joseph Kapena Agor

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

Operations Research

Raleigh, North Carolina

2019

APPROVED BY:

Julie Ivy	Maria Mayorga
Michael Kay	Osman Y. Özaltın Chair of Advisory Committee

DEDICATION

I dedicate this dissertation to my parents, David and Diana Agor.

BIOGRAPHY

Joseph Kapena Agor was born and raised in Pearl City, Hawaii, United States of America. After graduating from the Kamehameha Schools in 2006, he attended the University of Nevada, Reno where he received his bachelor's degrees in Civil Engineering (Cum Laude) and Applied Mathematics in 2011. He continued his studies at the University of Nevada to obtain his master's degree in Mathematics in 2013. In August, 2014, he joined North Carolina State University to pursue his Ph.D. degree in Operations Research under the direction of Dr. Osman Y. Özaltin. During his free time he enjoys online gaming, board games, surfing, spikeball, racquetball, golfing and simply hanging out with friends and family.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Osman Y. Özaltin for his great guidance during my PhD studies. I would also like to thank my committee members Dr. Julie Ivy, Dr. Maria Mayorga, and Dr. Michael Kay. They have provided me opportunities through internships, student supervising, and teaching that have allowed me to build the skills I will need in my upcoming career. I would also like to acknowledge that this work was supported by the National Science Foundation (Award Numbers: 1522072, 1522106, and 1522107).

I am also grateful too all of the friends I have made during my PhD career. The relationships I developed during my time at NC State is one of the key aspects that got me through this program and for this I am forever thankful for them all. I would also like to thank my family for their loving support through my academic studies. My grandparents Sue Agor, Sub Agor, Ronald Fo, and Charlotte Kawazoe have provided me with an immense amount of support over the years. I am also grateful for my sister, Chelsea Agor, who has been the most loving sibling a person could ask for. She has calmed me down in times of stress and picked me up in times of need during my undergraduate and graduate careers and for this I would like thank her. Last, but certainly not least, I would like to acknowledge my mother and father, Diana and David Agor. They have been more than just parents throughout this process, they have been rocks that I could always count and lean on. They have provided me with both emotional and financial support over my entire life and I am forever in their debt. My goal entering this PhD program was to make them proud and that will continue to be a priority of mine throughout the rest of my life. Thank you mom and dad, I love you very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

1.1 Motivation

Data driven solutions have become a successful problem solving methodology in the operations research industry. However, two major challenges that are consistently met are (1) determining which features of the data are *important* to solving problems and (2) using those features to provide *interpretable* results for management to use in the decision making process. In this dissertation, we address these issues through the use of optimization models, specifically bilevel optimization models. In this chapter, we introduce the problems that are addressed in this dissertation. We then give the outline of this proposal at the end of the chapter.

1.2 Statistical and machine learning models for classification, feature selection and score development

1.2.1 Classification

There has been much development machine learning and classification via mathematical optimization models over the past 30 years. Kotsiantis et al. [125] give a review of supervised classification

algorithms such as logic based algorithms, perceptron-based techniques, statistical learning algorithms, and support vector machines. They go on to discuss experimental results in the literature along with how to select appropriate classifiers. More recently, Shabanzadeh and Yusof [182] describe a new heuristic algorithm (Biogeography-Bsed Optimization - BBO) for unsupervised data classification. They test BBO on medical datasets and compare the performance to recent unsupervised data classification algorithms, which gives some perspective on the efficiency of BBO.

Classification in supervised learning may sometimes be referred to as “discriminant analysis" and there has been much work in both theory and application in this area. Initially, Freed and Glover proposed the use of linear and goal programming as alternative approaches to the discriminant analysis problem [70, 72]. In [71], they address some difficulties, such as placement of the origin, in these classification models, specifically in linear programming models and propose a normalization procedure to handle this issue. They state in [71] that normalization through the introduction of linear constraints can remedy issues, such as placement of the origin, and they prove the stability of the optimal solution for a specific type of normalization. As computational power began to flourish, integer and mixed-integer models in classfication problems did as well [81–83, 194, 207, 222, 226]. Optimization based classification methods can be used in a wide variety of applications spanning across many fields of research including medical and disease prediction [1, 14, 50, 132, 133, 136, 144, 146, 179, 199], sentiment classification [161, 204, 228], and natural language processing and text classification [98, 111, 161, 224], to name a few. For a comprehensive review of the literature on multicriteria classification, discriminant analysis and applications, the reader is referred too [132, 149, 232].

1.2.2 Feature selection

Classification models assign data points to predefined groups based on their features [133]. These models are typically built through cross validation to prevent overfitting. The simplest cross validation, namely the holdout method, partitions the data into two sets: the training set and the validation set. The classification model is built based on the training set, and then applied to the data points in the validation set. The model features are selected to achieve adequate out-of-sample classification performance on the validation set. There are two main approaches to feature selection. If feature selection is done independent of the learning algorithm that is used to construct the classifier, the technique is said to follow a *filter* approach [93, 163, 170]. Otherwise, it is said to follow a *wrapper* approach [124]. While the filter approach is generally computationally more efficient than the wrapper approach, its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm. The wrapper approach, on the other hand involves the computational overhead of evaluating each candidate

feature set by executing the learning algorithm.

1.2.3 Score development

Early prediction of disease progression is an on-going challenge in health care. Patient acuity and disease progression are frequently captured through the use of severity of illness scoring systems such as Acute Physiology and Chronic Health Evaluation (APACHE) classification system [2, 122], Sequential Organ Failure Assessment Score (SOFA) [213], Mortality in Emergency Department Scores (MEDS) [185], and Predisposition, Infection, Response, Organ dysfunction (PIRO) scoring system [35, 44, 102]. The development of these scores is mainly accomplished by combining medical expertise with statistical analysis of Electronic Health Record (EHR) data [35, 44, 102, 117, 122, 142, 185, 213]. However, missing information is ubiquitous in EHR data, and if not handled properly, this may lead to bias in the computation and interpretation of severity of illness scoring systems [31, 195]. There are established methods for handling missing EHR data such as neglecting patient records with missing values (complete case analysis), eliminating all variables with missing values from the analysis, or value imputation [51, 131, 176, 209]. Imputation methods have been shown to result in less biased conclusions [51, 61, 65, 108, 119, 155, 171]. Harel and Zhou [96] reviewed key ideas that form the basis of value imputation. Masconi et al. [148] presented a systematic review on the reporting of missing data and imputation methods in clinical studies. They highlighted the inexperience of investigators in their disregard of the effect of missing data and found that 62.5% of the selected studies did not even mention how missing data were handled. We quantify the value of missing information in severity of illness score development.

1.3 Bilevel optimization

Bilevel programs [46, 54, 152] model the hierarchical relationship between two autonomous decision makers: the leader and the follower. Each decision maker controls a distinct set of variables and the decisions are made hierarchically: the upper-level decisions are made first by the leader, after which the lower-level decisions are made by the follower. The follower's decisions in return affect the leader's performance. In the context of feature selection, the upper-level problem selects model features to maximize the out-of-sample classification performance on the validation set, while the parameters of the selected features are learned by the lower-level problem using the training dataset.

A general bilevel program can be formally stated as follows:

$$\max_{x \in \mathbb{X}} \quad F(x, y^*) \quad \text{subject to} \quad y^* \in \operatorname*{argmax}_{y} \{ f(x, y) \mid g(x, y) \leq 0,\ y \in \mathbb{Y} \}, \tag{1.1}$$

where variables are divided into two distinct classes, namely the leader's (upper-level) variables $x \in \mathbb{X} \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ and the follower's (lower-level) variables $y \in \mathbb{Y} \subseteq \mathbb{R}^{a_1} \times \mathbb{Z}^{a_2}$. Functions $F : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}^1$ and $f : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}^1$ are the leader's and the follower's objective functions, respectively. Furthermore, $g : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}^{m_2}$ corresponds to the follower's constraints that are impacted by the leader's decisions. In problem (1.1), it is assumed that the follower breaks ties between multiple optimal solutions in favor of the leader.

Bilevel optimization models have been considered in hazardous material transportation [92], network design [145], revenue management [40], traffic planning [151], energy [10] and computational biology [30, 166] to name a few. A broad class of problems which can be recast as a bilevel programs is the network interdiction problems [87, 107, 186, 223]. For surveys on bilevel programming methodology and applications the reader is referred to [9, 46, 54, 152].

Any linear mixed 0-1 programming problem can be reduced to a bilevel linear program (BLP) [6]. Therefore, BLPs, and so BPs in general are strongly NP-hard [95]. Furthermore, Vicente et al. [210] showed that even checking local optimality of BPs is NP-hard. Jeroslow [110] proves that algorithmic approaches to solving multi-level programs will require at least exponential time with $p \geq 3$ players leading to the result that value questions for bilevel programs are NP-hard. Due to the computational complexities associated with bilevel programs, we develop a genetic algorithm as a solution approach to the proposed bilevel models.

1.4 Dissertation Outline

The rest of this dissertation is organized as follows. In Chapter 2, we present the use of a linear programming model to develop a score to that captures physician workload to assist in the routing of patients admitted to an internal medicine department. A simulation is model is constructed to validate the effectiveness of the developed score. In Chapter 3, we quantify the value of missing information for use in the development of a scoring system to track severity in patients susceptible to sepsis. In Chapter 4, we develop a bilevel model for feature selection, propose a tailored genetic algorithm as a solution approach, and implement this model on three distinct case studies. Chapter 5 proposes a mixed integer framework for severity of illness score development with an implementation of this framework for the development of a score that can be used for those patient susceptible to sepsis. Finally, Chapter 6 concludes this dissertation by providing a summary of insights resulting from each chapter and the overall contributions. Furthermore, we discuss planned

future work that will be continued to further refine this research.

CHAPTER

2

SCORE TO CAPTURE PHYSICIAN TEAM WORKLOAD

The work reported in this chapter was accomplished over a summer internship at Mayo Clinic in Rochester, MN. This was a collaborative effort amongst myself, the Hospital Internal Medicine Department at Mayo Clinic and another PhD student in the North Carolina State University College of Design, Kendall McKenzie. The high-level translation of patient flow within the simulation, survey results and outcomes of interest that were tracked was a collaborative effort amongst the group. The contributions of this dissertation include the development and execution of the optimization model proposed in Section 2.3 and the construction of the simulation using the software `SIMIO` discussed in Section 2.4. These contributions delivered the results seen in Section 2.5 and facilitated the discussion of Section 2.6.

2.1 Introduction

Since 1990, the health care literature has seen a substantial increase in publications regarding the workload experienced by health care professionals. Research has shown that the amount of workload placed on nurses directly affects patient outcomes (e.g. survival or death) [63], as well as

nurse satisfaction and resilience in the workplace [90]. In order to prevent negative consequences associated with high workload, methods must be created to manage workload. Necessarily, the initial steps in developing a method for managing workload are, first, to define workload and, then, to quantify it. However, a review of the health care literature pertaining to workload shows that the medical community has not reached a consensus on the definition and quantification of workload.

There has been a substantial amount of research conducted on nurse workload and staffing. Since nurses are shift-based employees with a limited scope of tasks, this is a natural application area for those researching workload. Nursing workload has been defined in many ways, often by first determining a set of important factors that affect it. For example, Myny et al. [62] performed a cross-sectional study to identify the factors, outside of patient acuity, that contribute to nurse workload, including high numbers of work interruptions, high patient turnover rate, and high numbers of mandatory government registrations. A plethora of approaches exist for quantification of nurse workload. Kwiecien et al. [128] reviewed tools used for quantifying nurse workload in the ICU and classified them into five groups, which included patient classification, the Therapeutic Intervention Scoring System-28 (TISS-28), the Nine Equivalents of Nurse Manpower Use Score (NEMS), the Nursing Activities Score (NAS), and experimental methods.

Physicians workload has also been defined and quantified in the literature, usually pertaining to either (i) ED/ICU physicians or (ii) primary care providers / general practitioners. Gedmintas et al. [7] employed the Australian Triage Scale to develop a tool for managing staffing requirements and understanding resource use in the ED. Levin et al. [135] also tracked ED physician workload using a human factors approach that included time-motion task analysis and load index. Doerr et al. [60] used electronic health records and time studies to measure the time and complexity involved in the workload tasks of primary care physicians between their patient visits. While this research can serve as a foundation for defining and quantifying the workload of hospital providers, it is not a direct representation of the daily tasks and decisions experienced in an internal medicine environment.

Over the past few decades, the problem of balancing workload equitably among healthcare providers has been steadily gaining attention. However, few recommendations exists that are tailored to hospital workload. Researchers have primarily focused on developing methods for (i) determining staffing needs based on workload requirements and (ii) distributing workload equitably across providers at key decision points.

Many studies in engineering and management disciplines look at workload for the purpose of staffing. Bard and Purnomo [11], for example, developed a methodology for nurse scheduling that has the ability to dynamically adjust hospital-wide staffing recommendations based on supply and demand considerations. Additionally, Thorwarth et al. [201] created a simulation model to represent the dynamic workload experienced by ED providers for use by health care workload management

personnel. Punnakitikashem et al. [164] took a stochastic integer programming approach to solve nurse staffing and assignment problems. The objective of the stochastic program was to minimize both excess nurse workload and staffing costs. Other methods for balancing workload in health care settings focus on key decision points, such as admission decisions and patient allocation decisions. Tseytlin [205] developed a queueing model for the process of routing patients from the ED to internal wards. The author searched for routing policies that resulted in fairness and good operational performance. Hulshof et al. [105] used approximate dynamic programming to create a robust framework for allocating new patient admissions to health care resources. The literature is saturated with nurse-focused publications, but is lacking in specifics about physicians, especially as it pertains to hospitalists because this is a relatively new role for physicians.

2.2 Feature selection through surveys

At the Mayo Clinic in Rochester, MN, patients arrive daily needing general inpatient care by Hospital Internal Medicine (HIM) providers. When a patient arrives, he or she is immediately assigned to a care team, which consists of one doctor, one nurse practitioner or physician's assistant, and one clinical administrative assistant. This care team will be responsible for servicing all the needs of this patient until he or she is discharged from the hospital. However, the decision-making process for determining which patients get assigned to which care teams is sub-optimal. In fact, it often results in ill-will between care teams and decreased employee satisfaction. Currently, an incoming patient is assigned to whichever care team has the fewest number of patients in their charge, but this leaves ample room for imbalanced workloads between care teams, since the care team with "the fewest number of patients" does not necessarily equate to the care team with the lightest existing workload. The goal of our research is to create a workload score for each care team that is more accurately representative of the amount of work the team is being tasked with. By assigning incoming patients to whichever team has the lowest workload score, we can then be assured that the patient workload will be more equitably balanced among HIM care teams at the Mayo Clinic.

In order to identify the broad categories that contribute most to provider team workload, the HIM department conducted a Delphi survey among the Mayo Clinic providers prior to the analysis reported in this dissertation. Beginning in March of 2015, focus group sessions were held to create an exhaustive list of categories that providers perceived as affecting their workload. Throughout April and May of 2015, more than 900 categories were reviewed and condensed into a set of recurring themes. Nine broad categories were identified: patient churn (i.e. patient admissions and discharges and their associated work), lack of autonomy, work interruptions (e.g. pages, switching tasks, phone calls), non-clinical responsibilities (e.g. documentation), uncertainty about when the work day

Patient churn	Lack of autonomy	**Work interruptions**
Nonclinical responsibilities	Uncertainty about end of day	**Complexity of patients**
Work inefficiency	Changing team members	Geographical location of patients

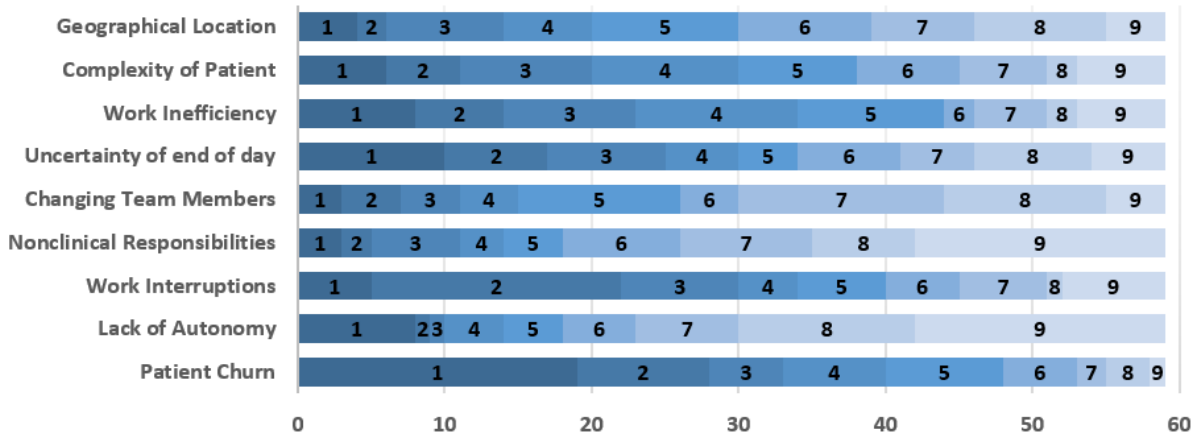
The four highlighted categories were selected for inclusion in the workload score

Figure 2.1 Nine Factors Contributing to Provider Workload

would end, complexity of patients, work inefficiency (e.g. poor communication between departments, awaiting patient arrivals), changing team members, and geographic location of patients. These categories are shown in Figure 2.1.

To assess the relative impact of category on provider workload, Mayo Clinic physicians created an online survey and disseminated it to Mayo Clinic providers in December of 2015. The survey asked providers to rank the nine categories in the order in which they contributed to workload (i.e. rank each of the categories from 1 to 9, where 1 identifies the category that contributes most heavily to workload and 9 identifies the category that contributes least to workload). A total of 59 providers responded to the survey and the results of those surveys were provided to us by the HIM department upon our arrival in May 2016. The distribution of ranks given to each category is shown in Figure 2.2. The mean rankings of each category (in decreasing order of contribution to workload) were 3.34 for patient churn, 4.27 for work interruptions, 4.37 for work inefficiency, 4.64 for uncertainty about the end of work day, 4.69 for patient complexity, 5.37 for geographical location of patients, 5.76 for changing team members, 6.39 for lack of autonomy, and 6.39 for non-clinical responsibilities.

It was decided that the four categories with the lowest mean rankings (indicating highest average contribution to workload) should be incorporated into the workload score. These were patient churn, work interruptions, work inefficiency, and uncertainty about the end of work day. After further discussion with Mayo Clinic providers and data experts, however, the fourth category (i.e. uncertainty about end of work day) was determined to be too difficult to quantify in a score. Thus, we replaced that category with the patient complexity, which could be easily quantified using diagnosis



The width of each bar segment is proportional to the number of physicians who assigned ranks 1-9 to each category, where 1 identifies the heaviest influence on workload and 9 identifies the lightest influence on workload

Figure 2.2 Distribution of rankings of nine categories contributing to provider workload

codes and was next in the ordered list of average rankings.

In order to develop a score to represent workload, the four primary categories were broken down into ten quantifiable factors (see Table 2.1) that would be reasonable to pull from the Mayo Clinic data systems. The ten factors included were number of admissions, number of discharges, number of patients in current census, number of low-complexity (level 1) patients, number of moderate-complexity (level 2) patients, number of high-complexity (level 3) patients, number of patients with families in town, number of behavioral patients, number of admissions confirmed by not yet physically arrived for care, and number of patient registrations in the ED in the last hour. All factors have the ability to be updated either daily or in real-time.

Table 2.1 Ten quantifiable factors used to represent the primary four categories of provider workload

Patient Churn	Patient Complexity	Work Interruptions	Work Inefficiency
# admissions	# Level 1 Patients	# patients with family in town	# admissions assigned & not arrived
# discharges	# Level 2 Patients	# behavioral patients	# ED registrations
# patients (census)	# Level 3 Patients		

All factors have the ability to be updated either daily or in real-time

To create a workload score using the ten factors in Table 1, we needed to assign appropriate weights to each factor. To accomplish this, Kendall McKenzie, a PhD student in the College of Design at North Carolina State University, designed and analyzed a secondary survey for the HIM providers using a choice-based conjoint analysis design. The survey was constructed using XLSTAT software, which

Which is worse?	Churn	Patient Complexity	Number of Interruptions	Amount of Indirect Work
Scenario 1:	High	Low	Average	High
Scenario 2:	High	Average	Low	High

Providers identified the scenario with higher workload in 15 such comparisons, and conjoint analysis was used to elicit utility values for each level (i.e. high, average, low) of the four broad categories

Figure 2.3 Example of comparison question presented in choice-based survey

used a D-optimal design to optimize the statistical significance produced by the survey results. In the survey, providers were presented with 15 comparison questions formatted as shown in Figure 2.3. The providers were asked to compare two potential scenarios, each with different levels (i.e. high, medium, low) of the four categories selected for inclusion in the workload score: patient churn, work interruptions, patient complexity, and work inefficiency. There were 19 respondents to our survey, and conjoint analysis was used to elicit relative utilities for each level of the four categories.

Next, we generated 1000 possible combinations of the ten workload factors in our score, using expert opinion to inform the allowable values for each factor. The values of all factors that represented a given broad category were given equal weight and added together, producing a single number to represent each of the four categories. Based on expert opinion, we defined ranges of values that constituted high, average, and low levels of each category, allowing us to condense each of the 1000 generated ten-factor combinations into a corresponding representation comprised of only the broad categories. We then attached a utility to each of the 1000 combinations by weighting its categories by their associated utility values and summing them. These steps resulted in a list of 1000 possible situations (i.e. ten-factor combinations), each with an associated utility value representing the workload that the situation might imply. Situations with higher utility values were considered to have heavier workloads, and those with lower utility values were considered to have lighter workloads. We ordered the 1000 combinations from heaviest to lightest workload, many of which had identical total utilities due to the limited number of four-category permutations. Combinations with identical utilities were placed into groups, resulting in 34 groups of situations. These groups served as input to our optimization model (Section 2.3), which attempted to minimize the number of deviations from these ordered groups.

2.3 Optimization model to develop the workload score

A linear optimization model was created to find the optimal weights for each of the ten factors included in the workload score. Essentially, the optimization model aimed to (i) minimize the number of situations that deviated from the ordered grouping obtained from the conjoint analysis results and (ii) keep the total of the factor weights in each broad category as close as possible to the relative category rankings from the Delphi survey results. The optimization model was trained using 1000 generated observations (i.e. ten-factor combinations). Each observation consisted of ten factors and was assigned a utility score based on the conjoint analysis results. Observations with identical utilities were grouped, and the groups were ranked in order from heaviest workload (highest utility) to lightest workload (lowest utility).

Let $\{C_i\}_{i=1}^n$ be a sequence of sets, e.g. categories that contribute to provider workload, where this sequence is in order of importance (i.e. set C_i has less utility than set C_j for each $i < j$). Let G be the number of groups of situations determined from the conjoint analysis (see section 3.1). Suppose that the groups are ordered from highest workload situations to lowest (the ten-factor combination in group i represents a higher workload situation than that of group j, for each $i > j$). We define the vectors $\boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^{\sum_i |C_i|}$ as the input observation vector and decision variable vector, respectively. The elements in $\boldsymbol{x}$ and $\boldsymbol{w}$ represent the numerical observations and weights on the elements within the sets $\{C_i\}_{i=1}^n$, respectively. We also define the decision variable e_{ij} to represent the error in workload score deviation when comparing observation $i \in G_k$ to observation $j \in G_{k+1}$ for $k = 1, \ldots, G-1$. Given an observation vector $\boldsymbol{x} \in \mathbb{R}^{\sum_i |C_i|}$, our goal is find weights for the $\sum_{i=1}^n |C_i|$ factors that result in a workload score calculation that accurately reflects the workload being experienced at the time when this observation is taken. We define our workload score, denoted S, as a weighted linear combination of these factors, $S = \boldsymbol{w}'\boldsymbol{x}$, such that S provides a numerical representation of workload at the time that observation $\boldsymbol{x}$ is taken. The linear program used to determine the weight vector $\boldsymbol{w}$ is as follows:

$$\text{Min} \quad \sum_{k=1}^{G-1} \sum_{i \in G_k} \sum_{j \in G_{k+1}} e_{ij} \tag{2.1a}$$

$$\text{s.t.} \quad \boldsymbol{w}'\boldsymbol{x}_i + e_{ij} \geq \boldsymbol{w}'\boldsymbol{x}_j \qquad i \in G_k, j \in G_{k+1}, k = 1, \ldots, G-1 \tag{2.1b}$$

$$\sum_{k \in C_i} w_k \leq \sum_{k \in C_{i+1}} w_k \qquad i = 1, \ldots, n-1 \tag{2.1c}$$

$$\boldsymbol{w} \geq \boldsymbol{1}, \quad \boldsymbol{e} \geq \boldsymbol{0} \tag{2.1d}$$

The objective function (2.1a) attempts to minimize the total error (i.e. deviation) from the results of

both the Delphi survey and the conjoint analysis. The constraints in (2.1b) ensures that the score preserves the order of the groups generated from the conjoint analysis survey. The constraints in (2.1c) ensure that the sum of the weights of the elements within each broad category (e.g. patient churn, patient complexity, work interruptions, and work inefficiency) are in order with respect to their average rankings from the Delphi survey. For instance, the sum of the weights of the elements within the work inefficiency category should be at least as high the sum of the weights of the elements in the patient complexity category, since the average rankings produced in the Delphi survey identified work inefficiency as contributing more to workload than patient complexity. The value $\boldsymbol{w}'\boldsymbol{x}_i$ is the workload score associated with observation i. The error term e_{ij} is added to constraint (2.1b) since preserving the ordered grouping may be infeasible (i.e. there may not exist a weight vector $\boldsymbol{w}$ such that all these constraints are satisfied).

We run an instance of model (2.1) in the context of our problem, where there are four categories in the sequence of categories, namely, patient churn (C_1), patient complexity (C_2), work interruptions (C_3), and indirect work (C_4). The number of elements in C_1, C_2, C_3, and C_4 are three, three, two and two, respectively, yielding $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^{10}$. The optimal weight vector $\boldsymbol{w}$ found after running the optimization model is shown in Table 2.2. To test how these weights performed, a second set of 1000 ten-factor combinations was created for use in validation. Using the same procedure as described in Section 2.2, we again assigned utility values to each of these observations and placed them in an ordered grouping. The performance measure used to evaluate the weight vector was the percentage of observations that were misplaced, i.e. did not fall into the ordered grouping correctly once the workload score was calculated. Thus, using the weights in Table 2.2, we output workload scores for the 1000 ten-factor combinations in the training set, as well as the 1000 ten-factor combinations in the validation set. We counted the number of observations that were not in their correct group locations and found that approximately 18.1% and 18.7% of the observations were misplaced in the training and validation sets, respectively. Our conclusion was that this score accurately captures the work being experienced within a medical service and that we could take this score to the next phase of verification through simulation.

2.4 Simulation model validation

A simulation model was built in `SIMIO 8` to represent the movement of patients and their associated workloads throughout the Mayo Clinic's HIM department. We modeled 11 provider teams (i.e. Med services), including services 1-4 (teams of multiple medical residents), services 5-9 and 11 (regular provider teams consisting of an MD, nurse practitioner or physician's assistant, and clinical assistant), and service 14 (a single provider who focuses more on doing rounds with existing patients than

Table 2.2 Optimal weights for each workload score factor

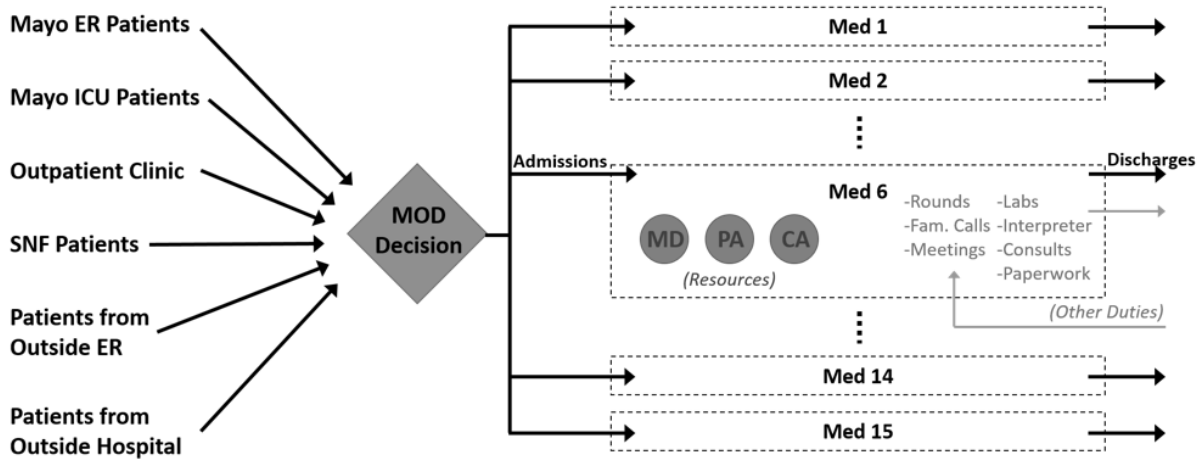
Factor Weight	Factor Variable	Factor Description	Category
1.0	w_1	# patients (census)	
3.5	w_2	# admissions	Patient Churn
2.5	w_3	# discharges	
1.0	w_4	# Level 1 Patients (low-complexity)	
2.0	w_5	# Level 2 Patients (moderate-complexity)	Patient Complexity
3.5	w_6	# Level 3 Patients (high-complexity)	
2.5	w_7	# patients with family in town	Work
4.0	w_8	# behavioral patients	Interruptions
1.0	w_9	# admissions assigned but not yet arrived	Indirect
5.5	w_{10}	# Ed registrations in last hour	Work

The weighted linear combination of these factors produces a resulting workload score

admitting new patients). The providers on each team were modeled as resources that needed to be seized to complete work. While the simulation model was primarily built by myself, during the simulation construction process, there was collaboration when translating the problem into elements that could be represented in the model. This required interdisciplinary discussions and understanding, facilitated by Kendall McKenzie with her background in Design.

Patients arrive to the system from six distinct sources, each with different distributions for diagnosis complexity and admission processing time. Three entity types were used to represent high, moderate, and low patient complexity (defined by diagnosis codes). The amount of daily work generated by each patient was different for each complexity level. Other patient attributes assigned upon creation included source location, transfer patient flags, arrival day and time, discharge day and time, etc.

Upon arrival, patients are sent to a decision node that represents the Medical Officer of the Day (MOD), who determines the provider team, or service, to which each patient will be assigned. After assignment to a service, a delay occurs (based on arrival source) to represent the amount of time usually required for the patient to physically arrive at the Mayo Clinic HIM department. After this delay, the patient entity is considered to be under the care of its assigned provider team until it is either transferred to a different provider team or discharged entirely from the hospital. Each service is modeled as a sub-model in the simulation, where the providers are modeled as workers that can be seized by jobs created by patients each day. A daily check within the simulation determines when each patient is scheduled to exit the sub-model (according to historical data). Work is produced on each day that a patient remains in the sub-model. Jobs (e.g. rounds, family visits, paperwork, etc.) are created at the start of each day, and the provider resources within the sub-model work to complete these jobs. A large portion of the work created deals with patient admissions and discharges. Figure 2.4 shows a high-level visualization of the simulation model. A sample sub-model flow diagram is displayed in Figure 2.5.



Each Med service is a sub-model that seizes resources (i.e. MD, physician's assistant, clinical assistant) who act as servers for generated tasks, such as admission work, rounds, paperwork, etc. Utilization of these resources is tracked.

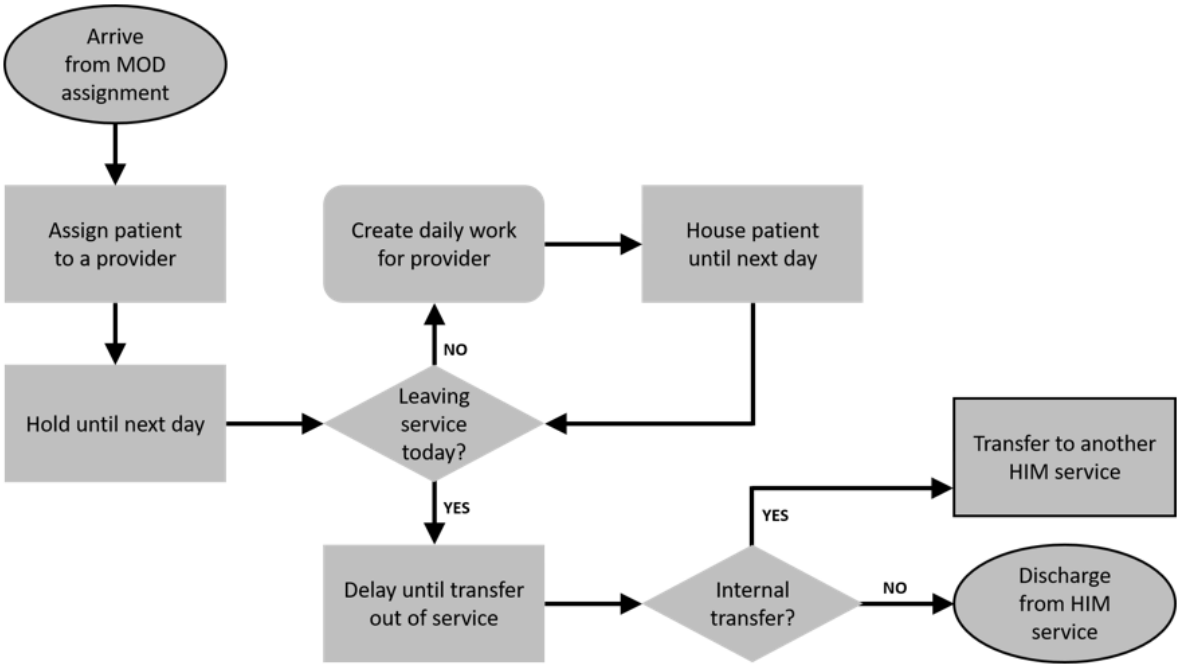
Figure 2.4 High-level flow diagram of simulation model

The simulation model samples from historical data in some cases, and it makes select assumptions in others. For example, patient arrival and discharge locations, arrival and discharge times, and patient complexity were matched to historical data. Distributions for provider job processing times, the number of behavioral patients, and the number of patients with families in town were developed from expert opinion. Using the historical data, we were able to first simulate the historical assignment locations of patients and track the resulting resource utilization. We were then able to re-run the model using our workload score to make patient assignment decisions and compare the resulting resource utilization with that of historical assignment policy.

2.5 Results from simulation model

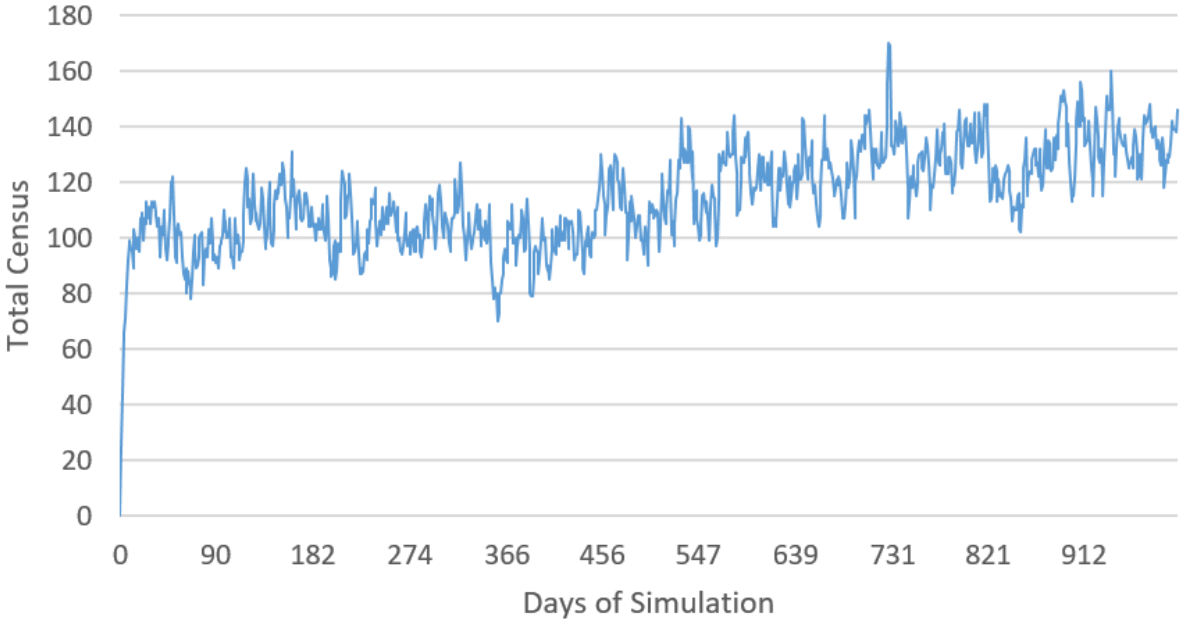
The simulation was run using data from January 1, 2013 through December 31, 2015. Since we did not have data prior to this time period, the model was run with a one-year warm-up period to allow the total patient census across all HIM services to stabilize (shown in Figure 2.6). After a one-year warm-up period, we began collecting data for calculating results. Figure 2.6 shows a steady increase in the total patient census throughout the entire three-year period, and this result was confirmed by HIM providers, displaying further evidence for the need to balance high workloads effectively across provider teams.

The experts at the Mayo Clinic evaluating the workload score were interested in comparing simulation output relating to a specific metric: the proportion of days per month that each service



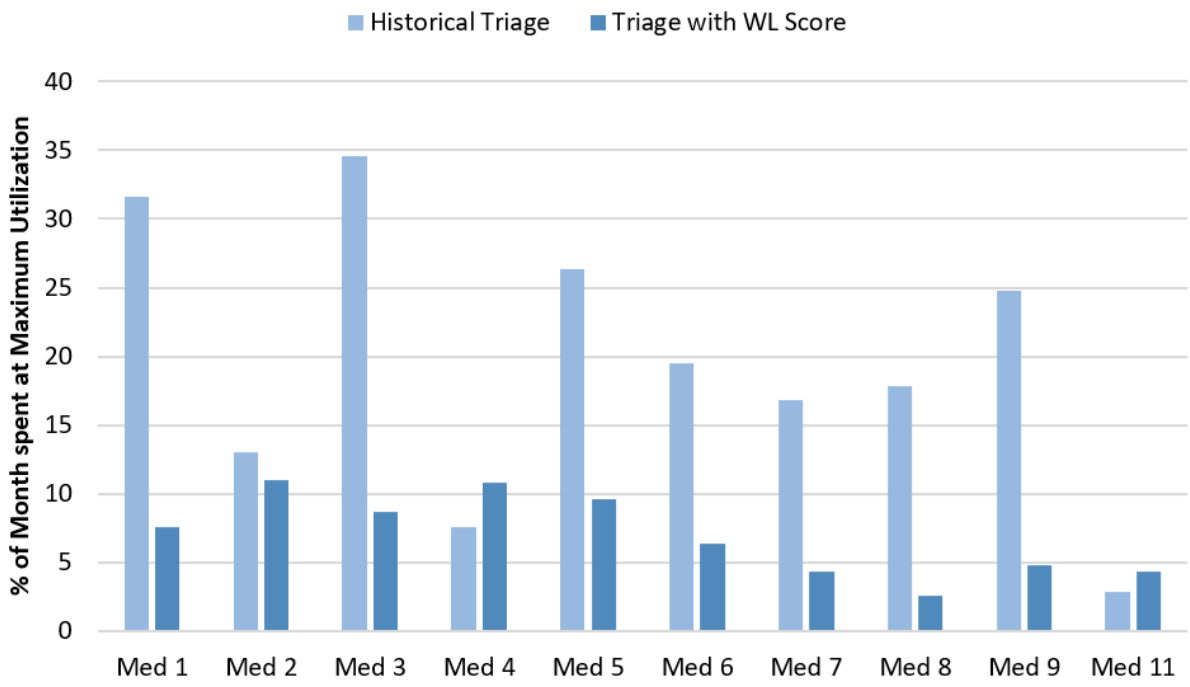
Outlined modules identify the entrance and exit points for the sub-model. All patients are assigned to a provider team by the MOD, thus entering the sub-model. Patients exit the sub-model by either (i) being transferred to another Med service sub-model (i.e. changing provider teams) or (ii) being discharged entirely from the HIM department.

Figure 2.5 Flow diagram of a single provider team (i.e. Med service) sub-model



After a one-year warm-up period, the census continued to increase steadily. This result was confirmed by HIM providers, displaying further evidence for the need to balance high workloads effectively across provider teams.

Figure 2.6 Simulated total patient census in HIM department over three years



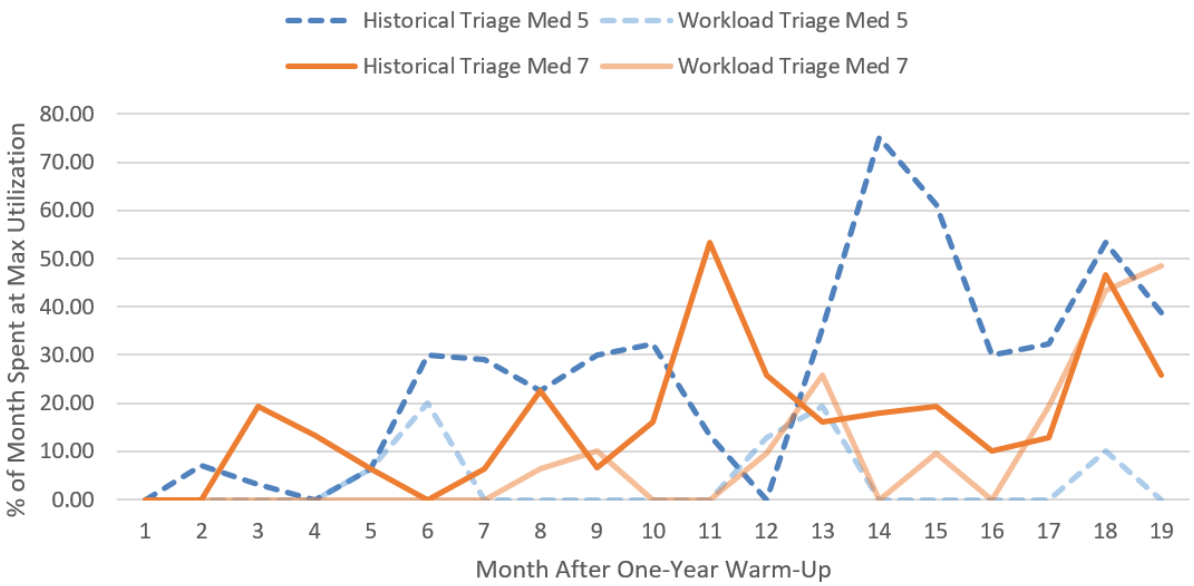
Maximum utilization defined by Mayo Clinic experts. Provider teams in services 1-4 were considered to have reached maximum utilization on any day that either (i) their census hit 12 or (ii) all of their admissions slots were used. Provider teams in services 5-9 and 11 were considered to have reached maximum utilization on any day that their census hit 14.

Figure 2.7 Proportion of month each Med service reached maximum utilization

spent any amount of time working at maximum utilization. The term maximum utilization was defined differently across provider teams. Provider teams in HIM services 1-4 were considered to have reached maximum utilization on any day that either (i) their census hit 12 or (ii) all of their admission slots were used. Provider teams in HIM services 5-9 and 11 were considered to have reached maximum utilization on any day that their census hit 14. The provider team in HIM service 14 was considered to have reached maximum utilization on any day that its census hit 12.

Figure 2.7 shows the proportion of time per month that each provider team reached its maximum utilization (on average) in the simulation. Using the proposed workload score to make patient assignment decisions resulted in a 12.1% decrease (on average) in the proportion of time per month provider teams spent at or above maximum utilization. The proportions in Figure 2.7 are calculated by averaging the monthly maximum utilization proportions across the entire 24-month observation period.

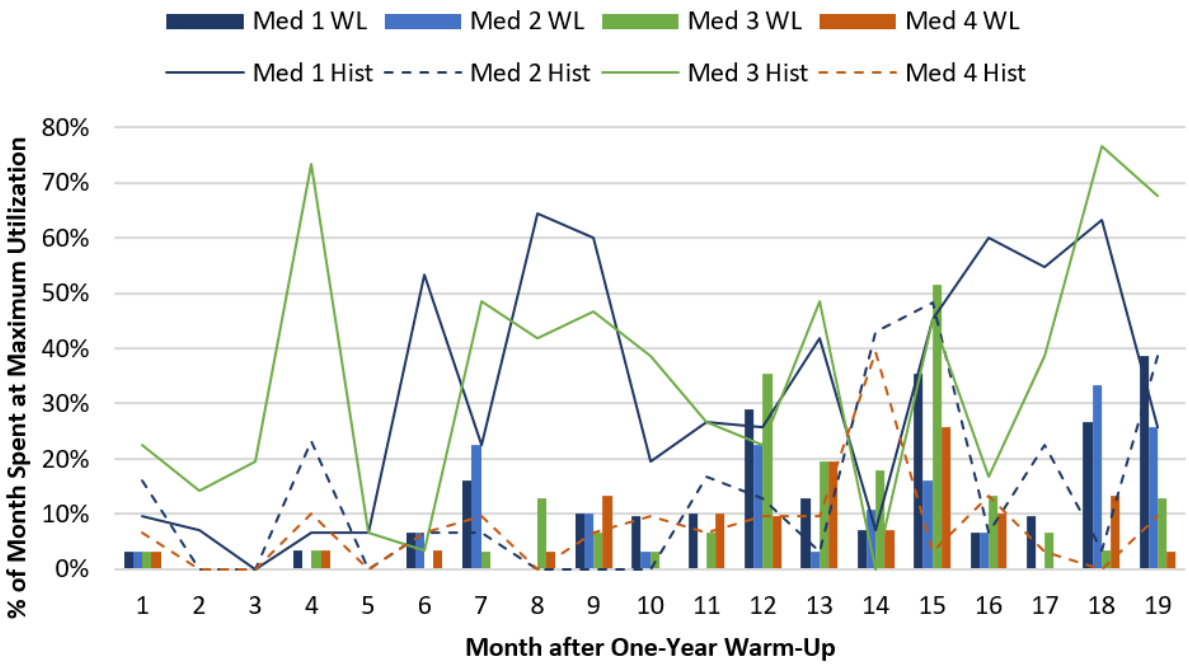
Mayo Clinic management was interested in understanding the difference in utilization between non-resident provider teams (HIM services 5-9, 11, and 14) and resident provider teams (HIM



Census level was at least 14 patients for Med services 5 (dashed line) and 7 (solid line) when comparing historical triage (darker color) and triage by simulated workload score (lighter color).

Figure 2.8 Proportion of month Med services 5 and 7 reached maximum utilization

services 1-4). Figures 2.8 and 2.9 provide comparisons of the simulated results for these two groups under historical triage assignments and triage assignments using our proposed workload score. Figure 2.8 shows the proportions of days in the month that two non-resident services (i.e. Med 5 and Med 7) reached a census level greater than 14. There is improvement when patients are triaged using the proposed workload score with respect to time spent at maximum utilization, along with the more desired outcome of more equitably-balanced workload between the two services throughout the displayed time window. Figure 2.9 shows distribution of the proportions of time spent at max utilization throughout the months of data collection pertaining to the resident services, which must follow strict rules that prevent them from taking on too much work, and shows the proportion of days per month that each service did not reach maximum utilization. It is clear that the proportion of time that the resident services are not fully utilized decreases when patients are triaged using the proposed workload score. This is an improvement that HIM providers are looking to achieve, since there is no risk of overworking the resident teams. Again, Figure 2.9 also shows improvement in the balance of workload across the resident services when the proposed workload score is used to make assignment decisions instead of the current census.



Lines represent maximum utilization levels under historical triage, and bars show maximum utilization levels under triage by our workload score.

Figure 2.9 Proportion of days per month that resident services 1-4 reached maximum utilization

2.6 Discussion

Through the use of a combination of methodologies (i.e. Delphi survey, conjoint analysis, optimization), we were able to create a workload score that proved successful in our simulation model. The measure that the Mayo Clinic found most informative for workload comparison was an interesting one: the percentage of days each month that a care team reached maximum utilization. Our simulation yielded a 12.1% decrease (averaged across all teams) in this metric when the proposed workload score was used to determine the provider team that should receive each incoming patient, instead of simply assigning incoming patients to the team with the fewest patients currently in their care. The results also show a significant decrease in the variation between team workloads when our workload score is used for patient assignment decisions. Furthermore, the provider teams staffed by medical residents showed a reduction in time spent being under-utilized, which is an improvement that Mayo Clinic HIM management was hoping to achieve. This confirms that the proposed workload score has the potential to balance workload more equitably across provider teams. Not only did we achieve a more equitable workload between HIM care teams, but we were also able to provide the Mayo Clinic operations team with a workload score calculation that more accurately represents employees' perceptions of their own workloads. The score can be implemented by pulling only ten numbers from the hospital data systems. In the future, we hope to implement this score within the Mayo Clinic and test its performance. Our next steps are to refine the score with provider input and implement this workload score as a triaging tool within the Mayo Clinic system for a beta test.

CHAPTER

3

THE VALUE OF MISSING INFORMATION IN SEVERITY OF ILLNESS SCORE DEVELOPMENT

3.1 Introduction

Many of the models used to develop severity of illness scoring systems do not specifically consider which clinical variables are missing but assume that missing data occur at random [28, 195]. However, for clinical decision making, it is critical to know if information (vital or lab) has been measured or is missing. In this work, we quantify the impact of missing and imputed variables on the performance of various prediction models used in the development of sepsis-related severity of illness scoring systems with the ultimate goal of incorporating this information into scoring systems used in real-time clinical practice.

The handling of missing data (or lack thereof) in the development of clinical scores to track patient disease progression has been rarely addressed in the literature. Howell et al. [102] developed the PIRO score for sepsis staging based on clinical variables in the Predisposition, Infection, Response and Organ dysfunction categories. They stated as a limitation that missing data may have biased their

results. Afessa et al. [2] investigated the impact of missing values in the APACHE III score on patients in the intensive care units (ICU) and found that the risk of death was strongly associated with the number and type of missing variables. Keegan et al. [116] reviewed intensive care severity of illness scoring systems for adults. They suggested that missing data may compromise the performance of the prognostic models used in the development of these scores.

Many of the severity scores developed for critical care patients impute values for the missing variables from their normal ranges [2, 130, 185]. While we take a similar approach and assume normal values for the missing variables, we also incorporate indicators to inform the model about which variables are missing. Knol et al. [123] discussed that using indicators for missing variables in an etiological context can lead to biased results. Our analysis, however, differs from their work in that we assess the predictive power of severity of illness scoring systems using dynamically recorded data derived from the EHR. Moreover, Knol et al. [123] imputed zero for missing variables, whereas we impute values from their normal ranges.

The objective of this study is to investigate the hypothesis that using information about which variables are missing along with appropriate imputation improves the performance of severity of illness scoring systems used to predict critical patient outcomes. To achieve this objective, we quantify the value of knowing which information is missing based on prediction performance in models that use all variables as predictors compared to those that utilize summary variables as predictors. We consider five different machine learning models including logistic regression, random forests, stepwise regression, support vector machines, and the least absolute shrinkage and selection operator (LASSO) methods.

This study focuses on the value of missing information in sepsis-related severity of illness scoring systems. Specifically, we consider a score developed in the PIRO framework to predict mortality in sepsis patients [102]. While the results are specific to this sepsis-related score, the same analysis can be generalized to identify the value of missing information in the development of severity of illness scores for other diseases.

3.2 Methods

3.2.1 Design and setting

We utilize EHR data from Christiana Care Health System (CCHS) as part of a collaborative National Science Foundation grant with North Carolina State University (NCSU) and the Mayo Clinic entitled S.E.P.S.I.S.: Sepsis Early Prediction Support Implementation System. The Institutional Review Boards

at CCHS, Mayo, and NCSU approved the study.

CCHS is a not-for-profit healthcare system comprised of two hospitals with over 53,000 hospital admissions per year and 1,100 hospital beds. The dataset includes longitudinal EHR data for adult patients (age $\geq$ 18 years) hospitalized between July 2013 and December 2015 corresponding to 119,968 unique patients and 210,289 visits. The analysis is performed at the visit level, i.e., we consider each visit as a unique case from which observations are generated. Observations refer to routinely collected data elements such as vital signs, lab values and clinical assessments that are associated with sepsis diagnosis and response. The care locations in the dataset include the emergency department (ED), non-intensive care units and intensive care units (ICUs).

We aim to evaluate the importance of knowing which information is missing for sepsis-related severity of illness scoring system development. Sepsis, infection plus systemic manifestations of infection, is the leading cause of in-hospital mortality. About 700,000 people die annually in the US hospitals and 16% of them are diagnosed with sepsis (including a high prevalence of severe sepsis with major complications) [188]. In addition to being deadly, sepsis is the most expensive condition associated with in-hospital stay, resulting in a 75% longer stay than any other condition [94]. In 2011, the total burden of sepsis to the United States (US) healthcare system is estimated to be $20.3 billion, most of which is paid by Medicare and Medicaid [203]. This accounted for 5.2% of the total aggregate costs for hospitalizations in the US resulting as the single most expensive treated condition in that year [203].

We focus our analysis on the PIRO score developed by Howel et. al. [102] because it is specifically designed for sepsis and our dataset includes variables related to sepsis diagnosis and treatment. Table 3.1 presents the variables and calculation of the PIRO score.

3.2.2 Sampling case and control populations

We consider the prediction of two primary patient outcomes: in-hospital mortality and first transfer to the ICU. For each outcome, we start with the total population of 210,289 visits and identify a subset of those that had suspected infection. We define a visit as being suspected of infection if the patient was administered an anti-infective (antibiotic, antiviral, or antifungal) or if a positive PCR Fast Flu Culture test result was reported at some point throughout the hospitalization. From the suspected infection population, we exclude those visits that were related to pregnancy. We then split this refined set of visits into case and control groups. Figure 3.1 displays the criteria used to isolate case and control populations for each outcome of interest. Throughout the remainder of the paper, we refer to the first transfer to ICU and in-hospital mortality outcomes as “ICU transfer" and “mortality", respectively.

Table 3.1 PIRO score components*

Category	Variable		Points	Max Possible Score
(**P**)redisposition	Age	<65	0	9
		65-80	1	
		>80	2	
	COPD		1	
	Liver Disease		2	
	Nursing Home Resident		2	
	Malignancy	Without Metastases	1	
		With Metastases	2	
(**I**)nfection	Pneumonia		4	4
	Skin/Soft Tissue Infection		0	
	Any Other Infection		2	
(**R**)esponse	Respiratory Rate>20		3	6
	Bands>5%		1	
	Heart Rate>120		2	
(**O**)rgan Dysfunction	BUN>20		2	16
	Respiratory Failure/Hypoxemia		3	
	Lactate>4.0		3	
	Systolic Blood Pressure	<70	4	
		70-90	2	
		>90	0	
	Platelet Count>150K		2	

COPD: Chronic Obstructive Pulmonary Disease
Bands: Bandemia (refers to an excess of immature white blood cells released by the bone marrow into the blood
BUN: Blood Urea Nitrogen (measures amount of urea nitrogen in the blood)
Hypoxemia: Pulse oximetry 90% on room air or 95% while breathing supplemental oxygen of 4 L/min [102]
**Developed by Howell et. al. [102]*

Approximately 24.7% of the patients were directly admitted to ICU. Since we generate observations five hours prior to the event (see Figure 3.2), patients who entered ICU on admission do not have sufficient information for inclusion in the study (only one row of data at time zero). Therefore, we exclude a visit from the ICU transfer case group if the first transfer to ICU occurred within five hours of arrival. Moreover, in order to capture visits that truly did not require transfer to ICU, we exclude a visit from the ICU transfer control group if the patient died in the hospital. From a clinical perspective, a transfer to ICU should have occurred for those visits that ended with in-hospital mortality. For the mortality control group, we exclude those visits which resulted in a discharge to hospice. While these visits did not actually end with an in-hospital death, placing them in the mortality control group would introduce bias as they do not represent the population of patients who survived their hospitalization and were discharged as healthy. From the control group of each outcome, we randomly sample six times the number in the respective case group (uniformly over the entire control group). In order to ensure that the sampled control group is not statistically different than the original control group (i.e., it accurately represents the entire control group)

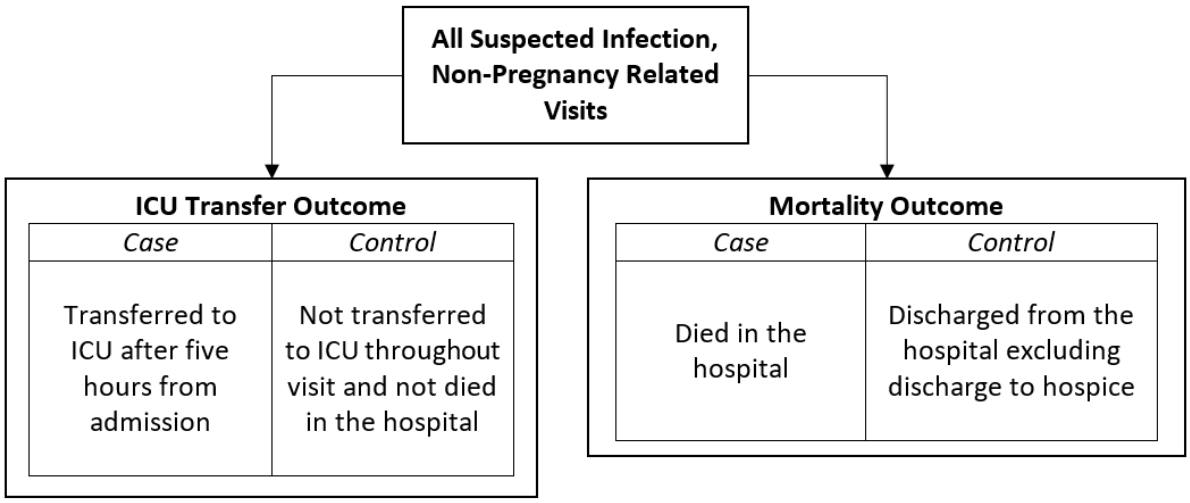


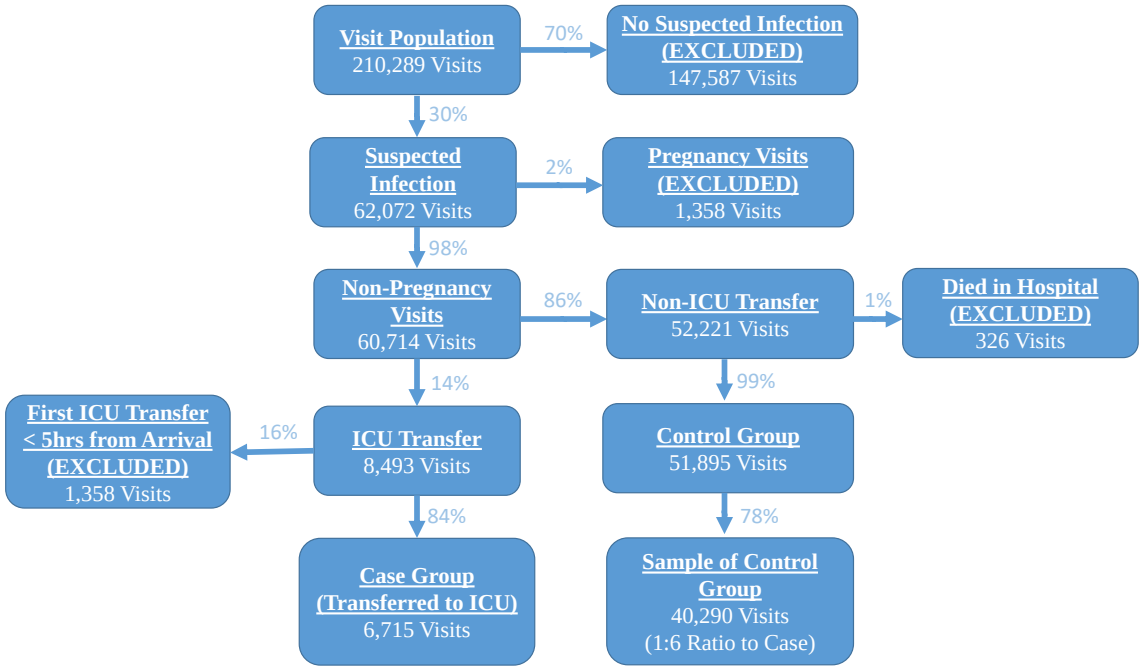
Figure 3.1 Selection criteria for case and control populations for each outcome of interest

with respect to age, race, and medical histories, we perform two-sided t-tests. Appendix Tables A.1 and A.2 give the results of these tests. We conclude that the sampled control groups statistically represent their corresponding control populations with respect to age, race and medical history. The final populations consist of 47,005 and 10,031 visits for the ICU transfer and mortality outcomes, respectively (Figure 3.2).

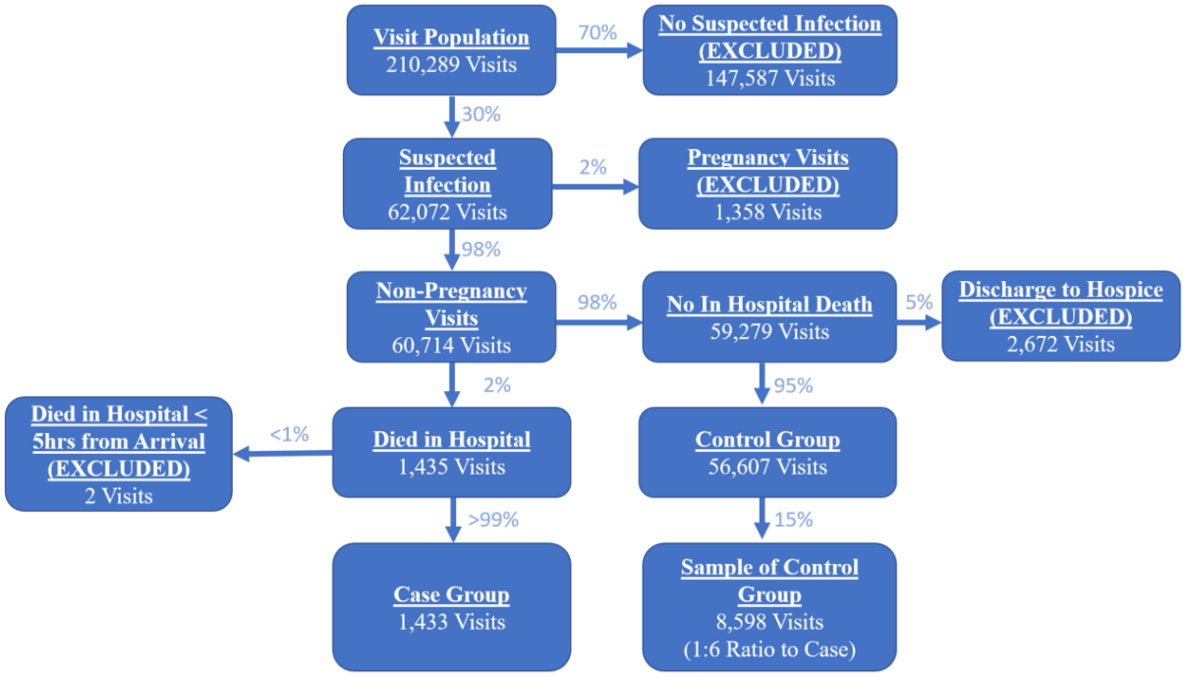
3.2.3 Observation generation

We define an *observation* as an array of real numbers representing each variable of the PIRO score. It has been shown that time periods of more than 4 hours can have a significant effect on mortality when responding to deteriorating patients [39, 100]. Hence, we choose five hours prior to the event as the observation time point for the case group of each outcome. For the control groups, we generate observations when the max PIRO score was observed during the visit to capture the most acute condition of the patient. If there are multiple times at which this happened, we take the earliest one. Figure 3.3 shows the time points at which observations are generated in the case and control groups.

The actual value of each variable is recorded in observations. In the PIRO score, age, for instance, is broken into three intervals. We do not use these intervals, and directly consider the age of the patient. We identify the Predisposition variables ”COPD", ”Liver Disease", and ”Malignancy" along with the Infection variable ”Pneumonia" through ICD9 diagnosis codes. The ICD9 codes related to the Predisposition category represent if there was any historical diagnosis of the condition at any point prior to the visit. The ICD9 codes used for identifying ”Pneumonia" represent if the patient was

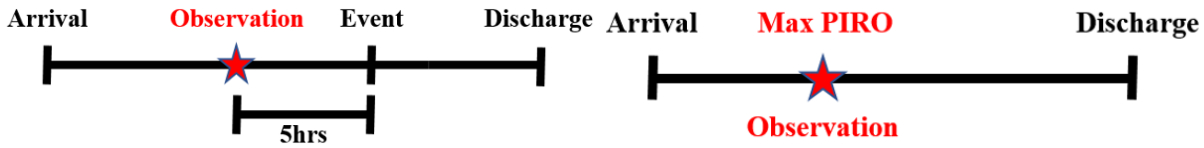


(a) ICU transfer outcome



(b) Mortality outcome
The percentage on each arrow represents proportion of the population group at the origin of the arrow

Figure 3.2 Generation of case and control groups for the ICU transfer and mortality outcomes



(a) Case observation generation **(b)** Control observation generation
Note that "Event" refers to either one of the outcomes of interest: ICU transfer or mortality

Figure 3.3 The observation time for (a) the case group and (b) the control group

diagnosed with pneumonia at some point during the visit. We were unable to distinguish between "skin/soft tissue infection" and "any other infection" in our data. However, we designate all visits in our analysis as being suspected of infection (Section 3.2.2). Therefore, we assume that not having a diagnosis of "Pneumonia" implies having "any other infection".

3.2.4 Missing variables

An additional row is generated in the EHR data with a corresponding time stamp whenever new information is gathered. This new row may not have a value for every variable of the PIRO score. In these instances, we use the latest available value [45]. If there is no such prior value, then we consider that variable as missing. That is, our definition of a missing variable is equivalent to having no measured value from the time of admission until the time at which the observation is generated.

The missing data mechanism is classified in three cases: (1) Missing Completely at Random (MCAR), (2) Missing at Random (MAR) and (3) Missing Not at Random (MNAR) [13]. These cases have been defined as follows [39, 45, 229]:

1. **MCAR**: The probability that a data point is missing does not depend on the value of that data point and any of the other variables. This is the desirable scenario for missing data
2. **MAR**: The probability that a data point is missing only depends on observed variables.
3. **MNAR**: The probability that a data point is missing depends on unobserved information, such as the value of the observation itself.

We utilize the R package BalyorEdPsych [12] and perform Little's test [138] to evaluate if the missing elements in our dataset is MCAR. We define a binary missing indicator for each variable of the PIRO score. This indicator equals to one when the corresponding variable is missing, and zero otherwise. We generate correlation coefficients between these missing indicators and observed variables (Figure A.1 in Appendix A). It is not possible to prove that missing data is MAR or MNAR

without collecting additional information on unobserved variables [13]. Additional information on missing variables is not available in this study because the data were collected retrospectively.

3.2.5 Value imputation

We employ two methods to impute missing variables: (1) Multiple imputation by chained equations (MICE) [32] and (2) normal range imputation. Multiple imputation has been shown to be effective in the analysis of longitudinal data with missing values[12, 32, 45, 48, 138]. We implement MICE through the R package mice [32]. Similar to other multiple imputation procedures, MICE assumes that missing data is MAR, and thus generates imputations based on the observed variables.

We investigate how the knowledge of missing information can impact the prediction performance of severity of illness scoring systems in a clinical setting. Therefore, we also employ a second imputation method by sampling values uniformly from their pre-defined normal ranges. We choose this imputation method to reflect clinical practice. If a vital or lab value is missing (not measured) up until a point in time during a patient's hospital stay, it is probable that the patient has not shown symptoms to prompt the measurement. Therefore, a physician would assume (appropriately or inappropriately) that this missing value is within a nonthreatening normal range. Many of the severity scores developed for critical care patients impute missing variables from their normal ranges [2, 48, 185]. The normal ranges used for respiratory rate, bands, heart rate, BUN, lactate, systolic blood pressure, and platelet count are: 12-20 bpm, 1-5%, 60-100 bpm, 7-20 mg/dL, 0-2 mmol/L, 90-120 mmHg, 150-450K per microliter, respectively. Hypoxia is a binary variable and the normal condition is when hypoxia is not present (i.e., it is equal to zero). These normal ranges for vital signs were determined using widely accepted values. We consulted our institutional laboratories to determine normal ranges for laboratory tests.

3.2.6 Experimental design

We first sample the population of visits as described in Section 3.2.3. Then, for each visit, we generate observations where there exist variables that have no value (are missing). Given these observations, we impute values for missing variables using the two methods described in Section 3.2.5 to create two complete sets of observations that have no missing data points. We repeat the imputation process 100 times generating 100 distinct complete observation sets. The difference across the 100 complete observation sets is the imputation values. For a given complete observation set, we generate four different observation vectors for each visit as shown in Table 3.2. Appendix Table A.3 presents an example of how these observation vectors are generated from a visit. For a given compete

observation set, we construct 20 distinct prediction models as described in Section 3.2.7 (5 models for each of the 4 observation vectors). For each prediction model, observation vector and imputation method combination, there will be 100 AUC values associated with the generated 100 complete observation sets (Figure 3.4). For example, there will be one hundred AUCs for the logistic regression model when the observations used to train the model include individual variables of the PIRO score with no missing indicators and the MICE procedure is used for imputation. We run paired t-tests to test if the mean difference between AUCs generated from observations with and without missing indicators are different.

Table 3.2 Four different observation vectors generated for each visit

	Individual Variables of the PIRO Score	**Four Variables Summarizing P,I,R,O Components**
Without Missing Indicators	Observation Vector 1	Observation Vector 3
With Missing Indicators	Observation Vector 2	Observation Vector 4

Note: Missing indicators refer to binary variables corresponding to whether or not a variable is missing and being imputed. PIRO: Predisposition, Infection, Response, Organ dysfunction scoring framework

3.2.7 Prediction models

Five different machine learning models for prediction are utilized: logistic regression [59], random forest [137], step-wise multiple regression [109], support vector machines (SVM) [133, 225], and the LASSO method [73, 202]. A main assumption of these models is that the predictor variables are independent of each other. Therefore, we calculate the correlation matrix, and exclude highly correlated variables from the model. We study two types of correlation: (1) correlation among variables and (2) correlation among missing indicators. While two variables may not be highly correlated (e.g. BUN and platelet count), their corresponding missing indicators could still be correlated due to measurement and documentation workflows (e.g., when BUN is not measured, platelet count is also not measured as they are commonly tested and documented together). The observations after removing correlated variables and indicators are used to generate the results.

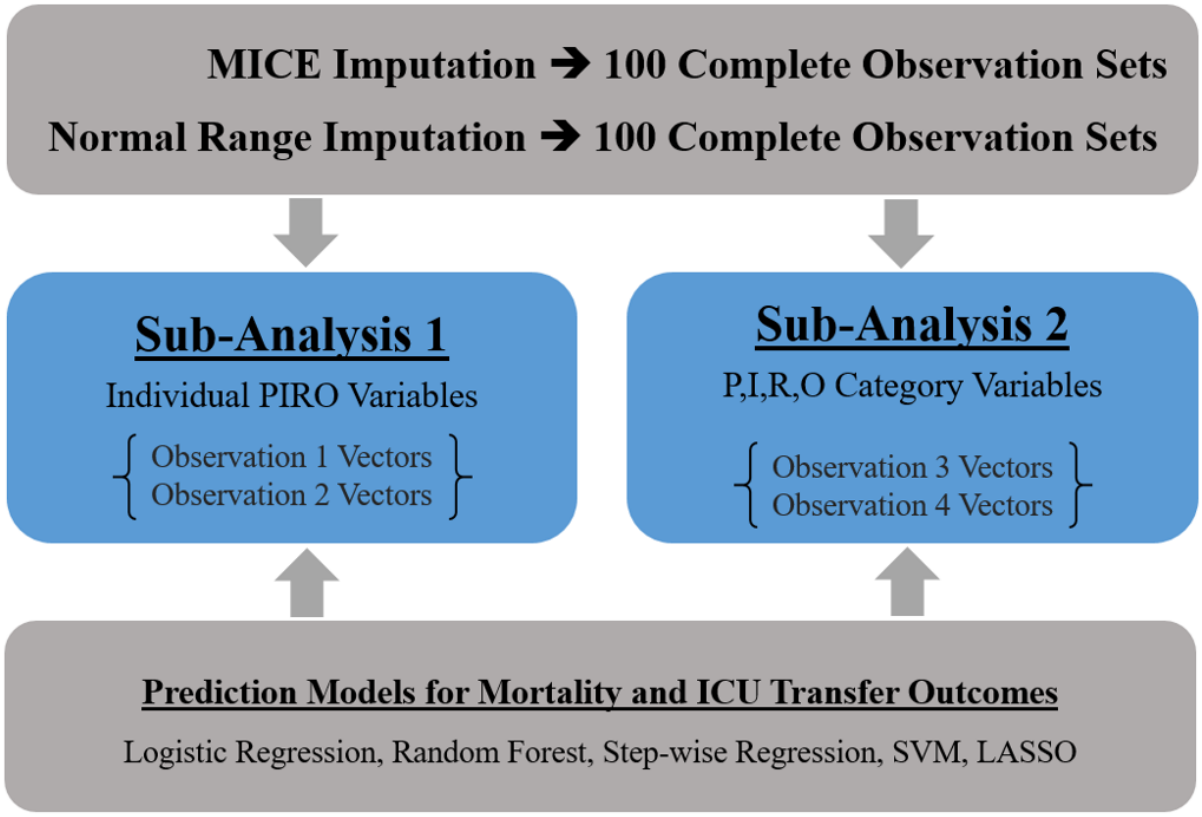


Figure 3.4 Summary of the experiment design.

3.3 Results

3.3.1 Correlation analysis results

There is no significant correlation found among the variables of the PIRO score. However, there is significant multi-collinearity present among the missing indicators. With respect to the mortality outcome, while the values of platelet and bands are not strongly correlated, there is strong positive correlation between their corresponding missing indicators (correlation coefficient ≈ 0.9939). The missing indicator for BUN is highly correlated with the indicators for bands (correlation coefficient ≈ 0.9088) and platelet (correlation coefficient ≈ 0.9028). Based on these findings, for the models predicting mortality, missing indicators for BUN and platelet count are removed from the observation vectors. Regarding the ICU transfer outcome, missing indicators for heart rate and respiratory rate are highly correlated (correlation coefficient ≈ 0.9472). The missing indicators for platelet and bands are highly correlated as well (correlation coefficient ≈ 0.9941). Therefore, for the ICU transfer outcome, missing indicators for heart rate and platelet count are removed from the observation vectors.

3.3.2 Missing data mechanism

Little's test [138] indicates that the missing variables in the observations (see Section 3.2.4) are not MCAR ($p < 0.001$). The magnitude of correlation between each missing indicator and any other observed variable is less than 0.5 (Figure A.1). This indicates that propensity of a missing data field is not strongly correlated with observed variables. We assume that the missing data mechanism is MAR for the MICE method. This assumption has been argued to be reasonable for longitudinal EHR data [196]. Departures from the MAR assumption may involve the MNAR mechanism under which the probability of any missingness pattern depends on data fields that are not observed under that pattern. Different forms of such dependence could represent the observed data equally well. Thus, a recurrent theme in the case of MNAR missing data is the need to make untestable identifiability assumptions based on domain knowledge [108, 209]. The normal range imputation motivated by clinical practice makes such an assumption. In particular, given a missing data field, we identify its value by assuming a uniform distribution within a normal range determined by clinicians. This method fits under the pattern mixture modelling approach, which is frequently used in the literature to handle MNAR missing data [61].

3.3.3 Distribution of the PIRO score

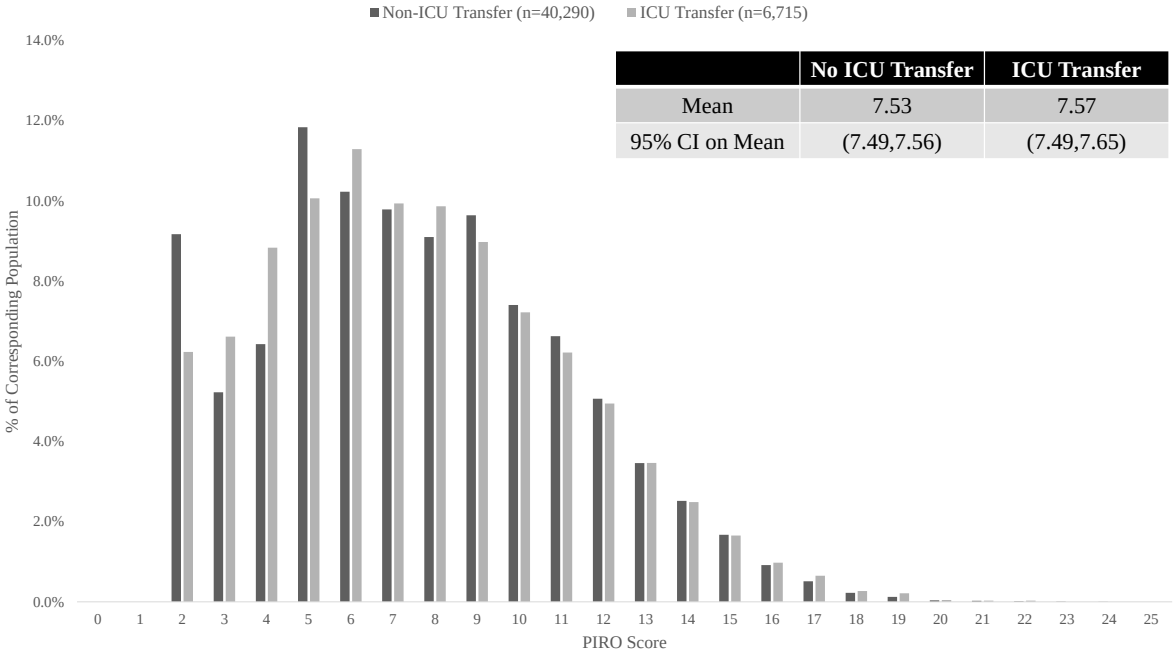
Figure 3.5 displays the distribution of the PIRO score for the ICU transfer and mortality outcomes. Observe that the PIRO score cannot differentiate between the case and the control groups of the ICU transfer outcome. Nevertheless, it captures acuity of patients who die in the hospital more accurately than those who are transferred to the ICU. This result is indicated by the significant difference in the mean PIRO scores between the case and control groups of the mortality outcome compared to the insignificant difference in the mean PIRO scores between the case and control groups of the ICU transfer outcome. The same result is further evidenced by the similarities of the PIRO score distributions in case and control groups of the ICU transfer outcome as shown in Figure 3.5a. Observe that the PIRO score distributions in the case and control groups of the mortality outcome are quite different with different centers and skewness as shown in Figure 3.5b.

3.3.4 Distribution of missing elements

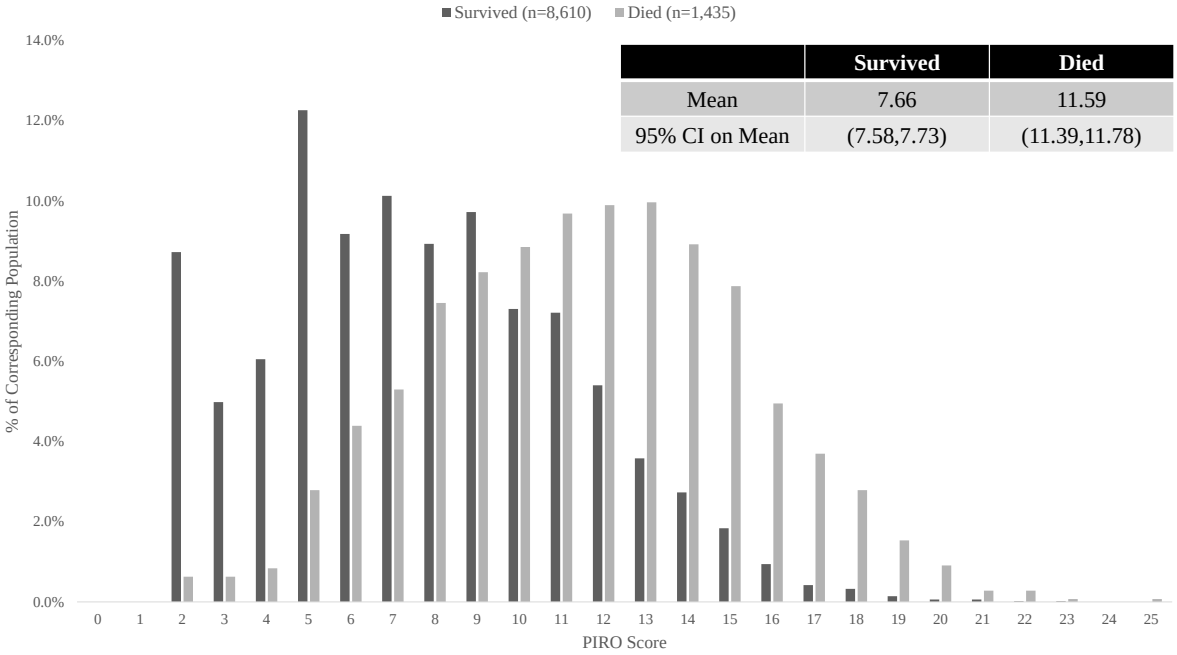
Figure 3.6 shows the frequency of missing variables in the Response and Organ dysfunction categories of the PIRO score for the two outcomes of interest. This figure indicates that there is generally more missing information in the ICU transfer observations than in the mortality observations. This abundance of missing information could contribute to the result that the PIRO score cannot differentiate between the case and the control groups of the ICU transfer outcome (Figure 3.5a).

3.3.5 Prediction model performance comparison

Our results are centered around two main sub-analyses comparing the models that have missing indicators to those that do not for each outcome. Sub-analysis 1 studies this comparison when all variables of the PIRO score are used as predictors, while sub-analysis 2 studies this comparison when the summary variables for the P, I, R, and O categories are used as predictors. We plot the receiver operator curve (ROC) and compare the area under the receiver operator curve (AUC) as a performance measure (Figure 3.7 and Tables 3.3, 3.4, 3.5 & 3.6). Tables 3.3 & 3.4 reports the AUC results and the relative gain in AUC when moving from models that do not include missing indicators to models that do when imputation of missing values is done via the MICE procedure. Tables 3.5 & 3.6 reports the same AUC results when imputed values are sampled from the pre-defined normal range. DeLong's algorithm [52, 198] is used to test the hypothesis that the AUC values of the models without missing indicators are different than the AUC values of the models with indicators. This algorithm uses theory developed for generalized U-statistics to compare two or more ROC curves and the resulting test statistic has a chi-square distribution asymptotically.

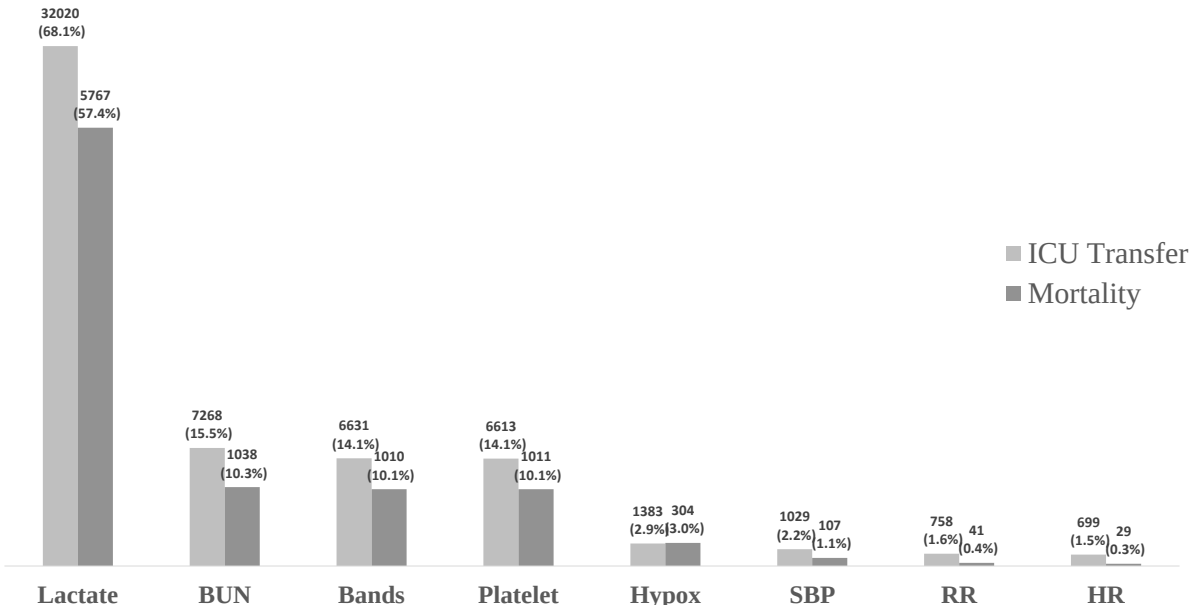


(a) ICU transfer outcome



(b) Mortality outcome

Figure 3.5 PIRO score distribution for the (a) ICU transfer and (b) mortality at the time observations are sampled



Labels: number of observations missing that covariate (% of observations)

Figure 3.6 Comparison of what is missing between ICU transfer and mortality outcomes

Due to the high prevalence of missing lactate measurements (Figure 3.6), we postulate that the missingness of lactate has a large impact on the results presented in Tables 3.3- 3.6. Therefore, we run a second set of experiments where only a missing indicator variable for lactate is used. We give the results of these experiments in Tables 3.7- 3.10. We observe that when the only addition to the model was a missing lactate indicator the relative AUC gain is less than when having missing indicators for all variables. However, we still observe that the relative AUC gain is larger when using summary category variables as predictors (Sub Analysis 2).

Figure 3.7 displays the ROCs of the logistic regression model for the two outcomes with and without missing indicators. These curves demonstrate that the models with indicators (solid lines) outperform the models without indicators (dotted lines). Moreover, the models with all variables of the PIRO score and missing indicators perform best. These results suggest that the value of missing information with respect to AUC is highest for the models with summary variables. They also indicate that the value of missing information can depend on the prediction model. In the random forest model where all variables of the PIRO score are used as predictors, the value of introducing missing indicators seems negligible (Appendix Figure A.2). Whereas in the logistic regression model (Figure 3.7), the value of missing information is much greater in both models (with all variables as predictors and with summary variables as predictors). The results for the remaining prediction

Table 3.3 Mean AUC results for models related to ICU transfer outcome when imputation is performed by MICE

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.77	0.80	4.6%	<0.001
Random Forest	0.85	0.87	2.9%	<0.001
Stepwise Regression	0.77	0.80	4.6%	<0.001
SVM	0.72	0.78	8.7%	<0.001
LASSO	0.77	0.80	4.7%	<0.001
Average	0.77	0.81	5.1%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.63	0.71	13.4%	<0.001
Random Forest	0.56	0.65	16.5%	<0.001
Stepwise Regression	0.63	0.71	13.3%	<0.001
SVM	0.51	0.63	24.6%	<0.001
LASSO	0.63	0.71	13.4%	<0.001
Average	0.59	0.69	16.2%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.4 Mean AUC results for models related to mortality outcome when imputation is performed by MICE

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.77	0.80	4.6%	<0.001
Random Forest	0.85	0.87	2.9%	<0.001
Stepwise Regression	0.77	0.80	4.6%	<0.001
SVM	0.72	0.78	8.7%	<0.001
LASSO	0.77	0.80	4.7%	<0.001
Average	0.77	0.81	5.1%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.72	0.85	18.6%	<0.001
Random Forest	0.64	0.76	18.9%	<0.001
Stepwise Regression	0.72	0.85	18.2%	<0.001
SVM	0.61	0.78	27.7%	<0.001
LASSO	0.72	0.85	18.5%	<0.001
Average	0.68	0.81	19.1%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.5 Mean AUC results for models related to ICU transfer outcome when imputed values sampled from a pre-defined normal range

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.69	0.77	10.3%	<0.001
Random Forest	0.83	0.85	2.8%	<0.001
Stepwise Regression	0.69	0.76	10.6%	<0.001
SVM	0.66	0.74	12.4%	<0.001
LASSO	0.69	0.77	10.3%	<0.001
Average	0.71	0.78	9.3%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.63	0.72	13.5%	<0.001
Random Forest	0.57	0.65	14.2%	<0.001
Stepwise Regression	0.63	0.72	13.4%	<0.001
SVM	0.50	0.61	22.0%	<0.001
LASSO	0.63	0.72	13.7%	<0.001
Average	0.59	0.68	15.4%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.6 Mean AUC results for models related to mortality outcome when imputed values sampled from a pre-defined normal range

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.89	0.92	3.3%	<0.001
Random Forest	0.92	0.94	1.6%	<0.001
Stepwise Regression	0.90	0.92	3.0%	<0.001
SVM	0.90	0.94	3.9%	<0.001
LASSO	0.89	0.92	3.2%	<0.001
Average	0.90	0.93	3.0%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.75	0.88	17.0%	<0.001
Random Forest	0.71	0.82	15.3%	<0.001
Stepwise Regression	0.75	0.87	16.6%	<0.001
SVM	0.70	0.82	18.5%	<0.001
LASSO	0.75	0.88	17.0%	<0.001
Average	0.73	0.85	16.9%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.7 Mean AUC results for models related to ICU transfer outcome when imputation is performed by MICE and only a missing indicator for lactate is used

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.77	0.78	0.9%	<0.001
Random Forest	0.85	0.85	0.6%	<0.001
Stepwise Regression	0.77	0.78	0.9%	<0.001
SVM	0.72	0.74	2.7%	<0.001
LASSO	0.77	0.78	0.9%	<0.001
Average	0.77	0.78	1.2%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.63	0.65	4.0%	<0.001
Random Forest	0.56	0.60	6.1%	<0.001
Stepwise Regression	0.63	0.65	4.1%	<0.001
SVM	0.51	0.51	0.0%	1
LASSO	0.63	0.65	3.9%	<0.001
Average	0.59	0.61	3.6%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.8 Mean AUC results for models related to mortality outcome when imputation is performed by MICE and only a missing indicator for lactate is used

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.88	0.91	3.7%	<0.001
Random Forest	0.92	0.93	1.9%	<0.001
Stepwise Regression	0.88	0.91	3.3%	<0.001
SVM	0.90	0.93	4.0%	<0.001
LASSO	0.88	0.91	3.7%	<0.001
Average	0.89	0.92	3.3%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.72	0.83	14.3%	<0.001
Random Forest	0.64	0.75	16.2%	<0.001
Stepwise Regression	0.72	0.83	14.8%	<0.001
SVM	0.61	0.78	27.7%	<0.001
LASSO	0.72	0.83	14.9%	<0.001
Average	0.68	0.80	17.7%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.9 Mean AUC results for models related to ICU transfer outcome when imputed values sampled from a pre-defined normal range and only a missing indicator for lactate is used

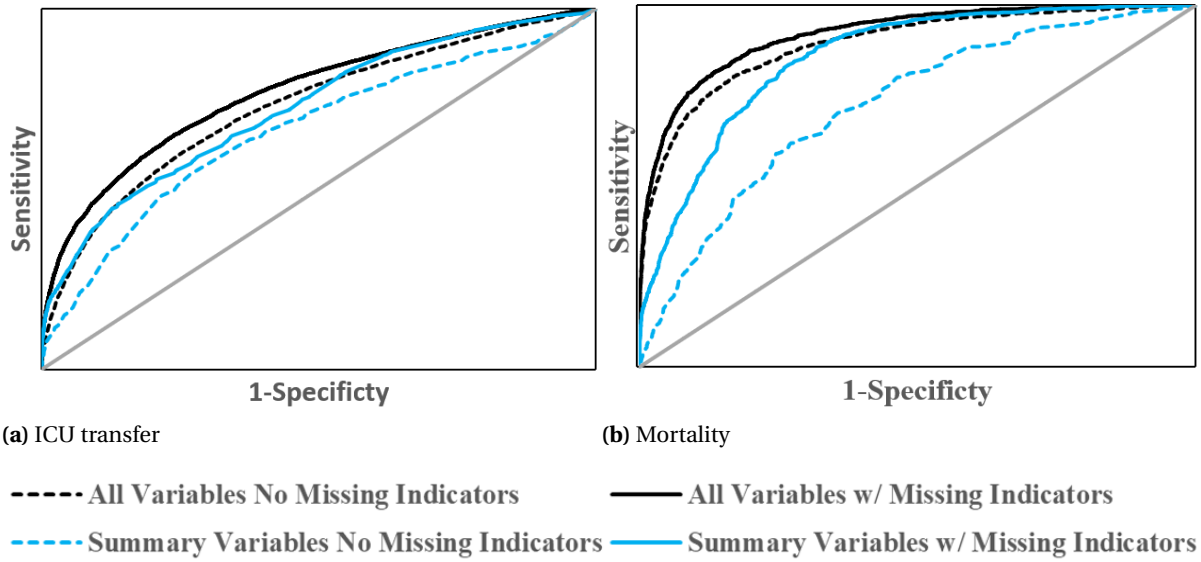
Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.69	0.69	0.0%	1
Random Forest	0.83	0.83	0.0%	1
Stepwise Regression	0.69	0.69	0.0%	1
SVM	0.66	0.68	2.5%	<0.001
LASSO	0.69	0.69	0.0%	1
Average	0.71	0.72	0.6%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.63	0.66	4.2%	<0.001
Random Forest	0.57	0.59	3.4%	<0.001
Stepwise Regression	0.63	0.66	4.1%	<0.001
SVM	0.50	0.55	10.2%	<0.001
LASSO	0.63	0.66	4.2%	<0.001
Average	0.59	0.62	5.2%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables

Table 3.10 Mean AUC results for models related to mortality outcome when imputed values sampled from a pre-defined normal range and only a missing indicator for lactate is used

Sub Analysis 1				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.89	0.91	1.6%	<0.001
Random Forest	0.92	0.93	0.6%	<0.001
Stepwise Regression	0.90	0.91	1.3%	<0.001
SVM	0.90	0.92	2.3%	<0.001
LASSO	0.89	0.91	1.5%	<0.001
Average	0.90	0.92	1.5%	
Sub Analysis 2				
	Without Missing Indicators	**With Missing Indicators**	**Relative Gain in AUC**	**p-value**
Logistic Regression	0.75	0.83	11.6%	<0.001
Random Forest	0.71	0.77	8.6%	<0.001
Stepwise Regression	0.75	0.83	11.6%	<0.001
SVM	0.70	0.78	11.4%	<0.001
LASSO	0.75	0.83	11.6%	<0.001
Average	0.73	0.81	10.9%	

Note: Relative gain in the area under the receiver operator curve (AUC) is computed by taking the difference in AUC between the models with and without indicators and dividing the result by the AUC from the model without indicators. All p-values calculated using De-Long's algorithm [52]
SVM: Support Vector Machine, LASSO: Least Absolute Shrinkage and Selection Operator
Sub Analysis 1 Models: Refers to models that use all PIRO variables as predictors
Sub Analysis 2 Models: Refers to models that use P, I, R, and O category scores as predictor variables



(a) ICU transfer **(b)** Mortality

Figure 3.7 Receiver Operator Curves created by logistic regression models for the (a) ICU transfer and (b) mortality outcomes

models are shown in Appendix Figures A.3 – A.5.

3.4 Discussion and conclusions

Our results show that models that use information about which clinical variables are missing can perform better than models that do not take that information into account because there is important clinical information in the fact that certain variables are missing.

Classical techniques related to handling missing data, such as deletion, substitution or imputation [28], deal with the “problem side" of missing data: finding the hidden value behind the missing data to make a better informed conclusion. In this paper, we present evidence on the “opportunity side" of missing data: the fact that data is not missing at random adds information that can be used to better predict an outcome of interest. While approaches have been developed for handling missing data using Bayesian network modeling [146], ours is one of the first to quantify the value of incorporating knowledge of missing information. Our findings are in line with our previous attempts to incorporate information about missing data in the health care setting, when data is not missing at random, but associated with certain clinical insight about the patient’s risk.

Our findings indicate that the main missing variable in our dataset is lactate i.e., knowing whether

lactate has been measured is most valuable for predicting mortality and ICU transfer. From a clinical standpoint, this is not surprising, as it is a laboratory test that is usually ordered when there is suspicion of tissue hypoxia or ischemia. This means that a missing blood lactate level indicates that the patient was considered low risk by the clinicians. This adds information about the patient's risk, even if the direction of that relationship is unclear. As clinicians, we would like to think that if clinicians considered the patient as low risk, it means the patient is low risk, but that is probably dependent on the clinical expertise. The next three most common missing variables are BUN, bands and platelets. They are missing in almost the same number of visits, which is expected since they are generally ordered together as part of a blood test. It is uncommon for a patient to not have a blood test in the first 24 hours after admission, so those test results are likely missing only at time points before a blood test is performed, generally in the first couple of hours after admission. This can also add important information: in some cases, a blood test may be delayed if the patient needs to be stabilized urgently. Also, the laboratory test may be ordered as "*stat*" in more severe patients.

In reviewing the results in Tables 3.3- 3.6, the more complex MICE procedure seems to result in better prediction performance compared to normal range imputation without missingness indicators. The normal range imputation, however, performs at the same level as MICE procedure after adding indicators. While we cannot make any assumptions whether the MICE procedure is capturing the same information that is captured by missingness indicators, this suggests that adding these indicators in effect creates a transformed variable that captures some information between the clinician's decision to measure that variable and the patient's risk.

As shown in Figure 3.5 the PIRO score performs better for the prediction of mortality than for the prediction of the ICU transfer. This is not surprising since the PIRO score was developed for the prediction of mortality in ICU patients. However, this result may also be influenced by differing ICU admission criteria between the institutions where the model was developed and our institution. The prevalence of missing data prior to the first transfer to ICU (Figure 3.6) may be another factor contributing to lower performance of the prediction models for this outcome.

As shown in Tables 3.5 & 3.6, models in which all variables of the PIRO score are used separately, allowing for multiple degrees of freedom in the model, perform much better than similar models using variables summarizing the P, I, R and O components. This is not uncommon with low-bias modeling methods like random forest and LASSO. While all of the models improve their performance by adding missing indicators, the relative gain is larger in the models that include the summary variables as opposed to all individual variables.

Our study has several strengths and some limitations. First, we do not separate our dataset into training and validation sets. This is a small limitation since we are interested in the informational value of missing data, and are not interested in the absolute performance of the models developed.

As we have described elsewhere, to evaluate the models for implementation, different metrics other than AUC can also be used [172]. We use a large dataset from multiple hospitals within a hospital system, which we would expect to have relatively heterogeneous clinical practices and documentation patterns. Additionally, our results are robust. The improvement in the model accuracy when adding the missing indicators is consistent across all of the modeling methods used, both in low-bias (random forest) and low-variance model types (logistic regression with and without stepwise selection of variables, or LASSO). The results are similarly robust for both the prediction of mortality and the ICU transfer outcomes. However, our results are limited by the fact that this is a single institution. Our dataset might not be representative of other hospitals or hospital systems and the attributes of their corresponding EHR. Future studies are needed to assess the degree to which the informational value of missing data is institution-specific. Finally, our findings are specific to one disease. The results will need to be replicated for other conditions and outcomes, and using different variables and missingness indicators.

We present evidence that the performance of models predicting mortality and ICU transfer for patients in the sepsis spectrum can be improved by incorporating information about which data elements are missing. Accurate predictions of such adverse outcomes could enable early intervention. It has been shown in the literature that early intervention can result in reduced mortality rate when responding to deteriorating patients [39, 229]. Therefore, it seems reasonable to expect that improving the predictive performance of models through inclusion of missing information would result in improved quality of care. While our results are specific to a sepsis-related severity of illness score, the same analysis can be generalized for other diseases. Our findings support the recommendation to explore the incorporation of missing variables, along with appropriate imputation, in the development of predictive models that use EHR data.

CHAPTER

4

BILEVEL MODELS FOR FEATURE SELECTION

4.1 Introduction

As discussed in chapter 1, the two main approaches to feature selection are the *filter* and *wrapper* methods. We focus on the wrapper approach in this chapter. A typical feature selection method with the wrapper approach would define a grid over candidate features, and then perform cross validation for each grid point [154]. This method, however, would suffer from combinatorial explosion of grid points in high dimensions. Problems with many features arise frequently in real life applications [19, 91]. For those problems, greedy strategies such as stepwise regression, backward elimination, filter methods, or genetic algorithms are used [103, 189].

We propose a bilevel programming approach to feature selection for classification models. In the literature, Kunapuli et al. [127] proposed a bilevel programming approach to classification model selection based on T-fold cross validation. They utilized a support-vector machine (SVM) model in the lower level as the follower's problem. Their upper-level problem chose lower and upper bounds on the parameters of the lower-level SVM model to minimize T-fold average misclassification error. Kunapuli et al. [127] reformulated this bilevel problem as a mathematical program with

equilibrium constraints (MPEC). They described a grid search procedure and solved a relaxed nonlinear programming reformulation of the MPEC using off-the-shelf nonlinear programming solvers. Our work is different from Kunapuli et al. [127] as we explicitly control the number of model features selected in the upper-level using binary variables. Kunapuli et al. [127] has only continuous variables in the upper-level problem. Furthermore, in addition to SVM, we implement a Lasso-based logistic regression model in the lower level. Finally, we develop a genetic algorithm solution approach and compare the performance to a derivative-free optimization method.

We implement the proposed bilevel feature selection approach in three different case studies where we classify influenza strains based on antigenic variety [136], distinguish between good and bad quality colposcopy images [69], and identify splice junction sites in genetic sequences [150]. Our results indicate that the proposed bilevel framework can be used to achieve similar, if not stronger, classification performance using fewer model features. There has been some investigation into solving general bilevel programs using metaheuristics in the literature [34, 77, 99, 217, 221]. however, any metaheuristic approach needs to be carefully designed for the specific problem under investigation to be able to produce good solutions efficiently. The main algorithmic contribution of our paper is to show that a tailored genetic algorithm can be used to solve the feature selection problem for complex machine learning methods such as SVM and Lasso-based logistic regression.

4.2 Machine learning models and the proposed bilevel model

We briefly describe the Lasso-based logistic regression and support vector machine models. We then present our bilevel feature selection model and outline the proposed genetic algorithm.

4.2.1 The Lasso-based logistic regression

The least absolute shrinkage and selection operator (Lasso) method performs both feature selection and regularization when building regression models [73, 97, 202]. In basic linear regression, a set of input observations are provided $(\boldsymbol{x_i}, y_i)$ for $i = 1, ..., m$ and a best-fit hyperplane $y = \pi_0 + \boldsymbol{\pi}'\boldsymbol{x}$ is generated by finding parameters $(\boldsymbol{\pi}, \pi_0)$ such that the sum of the squared deviations from the hyperplane is minimized. In the Lasso-based linear regression, the same objective is optimized subject to an ℓ_1-constraint $||\boldsymbol{\pi}||_{\ell_1} \leq t$, where t is a given regularization parameter [202]. The model is solved as an unconstrained optimization problem by appending a penalty term, i.e., $\lambda||\boldsymbol{\pi}||_{\ell_1}$, to the least-squares error in the objective function, where $\lambda \geq 0$ is a given Lagrangian parameter [97].

The response variables are continuous in regression models. For classification models with binary

response variables (e.g., $Y = 0$ or $Y = 1$), a logistic regression can be built in the Lasso framework. In particular, consider the following logistic regression model:

$$\Pr(Y = 0|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{1 + e^{-(\pi_0^* + \boldsymbol{x}'\boldsymbol{\pi}^*)}}$$
$$\Pr(Y = 1|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{1 + e^{+(\pi_0^* + \boldsymbol{x}'\boldsymbol{\pi}^*)}}$$

In the Lasso-based logistic regression, hyper-parameters $(\boldsymbol{\pi}^*, \pi_0^*)$ are estimated by maximizing the penalized log likelihood [73]:

$$(\boldsymbol{\pi}^*, \pi_0^*) \in \underset{(\boldsymbol{\pi}, \pi_0) \in \mathbb{R}^n \times \mathbb{R}}{\operatorname{argmax}} \left\{ \frac{1}{2m} \sum_{i=1}^{m} \mathbf{1}(y_i = 0) \log p(\boldsymbol{x}_i) + \mathbf{1}(y_i = 1) \log(1 - p(\boldsymbol{x}_i)) - \lambda \|\boldsymbol{\pi}\|_{\ell_1} \right\}, \tag{4.1}$$

where $p(\boldsymbol{x}_i) = \Pr(Y = 0|\boldsymbol{X} = \boldsymbol{x}_i)$ for $i \in \Omega$. Furthermore, the indicator function $\mathbf{1}(\cdot)$ is such that $\mathbf{1}(\texttt{true}) = 1$ and $\mathbf{1}(\texttt{false}) = 0$. After the optimal parameters $(\boldsymbol{\pi}^*, \pi_0^*)$ are found by solving (4.1), classification happens as follows:

$$\begin{array}{lcl} p(\boldsymbol{x}_i) \geq \eta & \Longleftrightarrow & Y = 0 \\ p(\boldsymbol{x}_i) < \eta & \Longleftrightarrow & Y = 1 \end{array} \qquad \forall i \in \Omega,$$

where η is a given threshold.

There has been much discussion of Lasso-based classification models in the literature with respect to applications and solution approaches. Ghosh and Chinnaiyan [79] combined classification and variable selection in the analysis of microarray data through a Lasso framework. Ma et al. [141] proposed a supervised group Lasso approach to cluster gene expression data for predictive models. They tested their models on two cancer and two lymphoma datasets to identify which gene clusters yield stronger diagnostic predictive power. Vincent and Hansen [215] proposed a sparse group lasso algorithm to be used on high dimensional, multi-class classification problems. Friedman et al. [73] developed a cyclic coordinate descent algorithm which they implemented in the R package `glmnet` to solve problem (4.1). They approximated the log-likelihood portion of the objective using Taylors expansion and utilized coordinate descent to solve problem (4.1) with this approximation in place.

4.2.2 Support vector machines

The support vector machine (SVM) is a classification model that produces hyperplanes to separate different classes of data. The SVMs apply regression methodology in a feature space H, where the observation space X is mapped to the feature space through a kernel function $\Phi : X \to H$ [115, 173].

In binary classification, the SVM takes observation vectors $\boldsymbol{x}_i \in \mathbb{R}^n$ and corresponding response variables $y_i \in \{-1, 1\}$ for $i = 1 \dots m$ as inputs, and returns the parameters $(\boldsymbol{\pi}, \pi_0) \in \mathbb{R}^n \times \mathbb{R}$ which form a separating hyperplane $\Phi(\boldsymbol{x})'\boldsymbol{\pi} + \pi_0 = 0$. The ℓ_2-norm soft margin problem for computing the SVM classifier is given by:

$$\min_{\boldsymbol{\pi}\in\mathbb{R}^n, \boldsymbol{\xi}\in\mathbb{R}^m_+, \pi_0\in\mathbb{R}} \quad \frac{\lambda}{2}||\boldsymbol{\pi}||_2^2 + \sum_{i=1}^m \xi_i \tag{4.2a}$$

$$\text{subject to} \quad y_i\left(\Phi(\boldsymbol{x}_i)^T\boldsymbol{\pi} + \pi_0\right) \geq \rho - \xi_i \qquad \text{for } i = 1 \dots m \tag{4.2b}$$

The objective function (4.2a) minimizes the deviation from the hyperplane $\Phi(\boldsymbol{x})'\boldsymbol{\pi} + \pi_0 = 0$ with $\boldsymbol{\xi}$ being an error vector. This model has soft margins because errors are allowed in constraints (4.2b). The regularization parameter $\lambda \geq 0$ and the threshold parameter $\rho > 0$ are specified by the user. Karatazoglou and Meyer [115] gave the framework for the development of the R package `e1071` to solve model (4.2). They showed that the solution can be written as $\boldsymbol{\pi} = \sum_{i=1}^m \alpha_i y_i \Phi(x_i)$ where coefficients α_i are found by solving a quadratic programming problem.

4.2.3 Bilevel feature selection

The parameters of features in classification models (4.1) and (4.2) are learned based on a training dataset. The out-of-sample performance should be assessed by applying the classifier to a validation dataset. Model features are selected to prevent overfitting. We propose a bilevel approach to feature selection for classification models. This approach selects model features to maximize the out-of-sample classification performance on the validation set.

Let Ω be the dataset. Moreover, let $\Omega_{\mathrm{T}} \subseteq \Omega$ be the training set and $\Omega_{\mathrm{V}} = \Omega \setminus \Omega_T$ be the validation set in the holdout cross validation procedure. We further partition Ω_{T} into $\{\Omega_{\mathrm{T}}^0, \Omega_{\mathrm{T}}^1\}$ and Ω_{V} into $\{\Omega_{\mathrm{V}}^0, \Omega_{\mathrm{V}}^1\}$, where Ω_{T}^k and Ω_{V}^k represents the set of observations from class $k \in \{0, 1\}$ in the training and validation sets, respectively. Let the upper-level binary variable u_j be equal to 1 if feature $j = 1, \dots, n$ is selected for the classification model, and 0 otherwise. The bilevel feature selection model with the Lasso-based logistic regression in the lower level is given by:

$$\min_{\boldsymbol{u}} \sum_{i\in\Omega_{\mathrm{V}}^0} \mathbf{1}\left(p(\boldsymbol{x}_i) < \eta\right) + \sum_{i\in\Omega_{\mathrm{V}}^1} \mathbf{1}\left(p(\boldsymbol{x}_i) \geq \eta\right) \tag{4.3a}$$

$$\text{s.t.} \sum_{j=1}^n u_j \leq \beta \tag{4.3b}$$

$$u_j \in \{0, 1\} \qquad \text{for } j = 1, \dots, n \tag{4.3c}$$

$$(\boldsymbol{\pi}^*, \pi_0^*) \in \underset{(\boldsymbol{\pi},\pi_0)\in\Pi(\boldsymbol{u})}{\operatorname{argmax}} \left\{ \frac{1}{2|\Omega_{\mathrm{T}}|} \sum_{i=1}^{|\Omega_{\mathrm{T}}|} \mathbf{1}(y_i = 0)\log p(\boldsymbol{x}_i) + \mathbf{1}(y_i = 1)\log(1 - p(\boldsymbol{x}_i)) - \lambda ||\boldsymbol{\pi}||_{l_1} \right\}.$$

The lower-level feasible region is given by $\Pi(\boldsymbol{u}) \triangleq \{(\boldsymbol{\pi}, \pi_0) \in \mathbb{R}^n \times \mathbb{R} \mid -M\boldsymbol{u} \leq \boldsymbol{\pi} \leq M\boldsymbol{u}\}$, where the big-$M$ is an arbitrarily large positive constant. The definition of $\Pi(\boldsymbol{u})$ ensures that only the features selected in the upper level, i.e., $u_j = 1$, are included in the Lasso-based logistic regression model optimized over the training set in the lower level. The upper-level objective function (4.3a) minimizes the number of misclassified observations in the validation set Ω_{V}. In other words, it maximizes the out-of-sample classification performance. Constraint (4.3b) restricts the model to select at most β features out of n candidate features.

Next, we present another bilevel feature selection model where the lower-level problem is an SVM with a linear kernel function, i.e., $\Phi(x) = x$.

$$\min_{\boldsymbol{u}} \sum_{i\in\Omega_{\mathrm{V}}^0} \mathbf{1}\left(\boldsymbol{\pi}^*\boldsymbol{x}_i' + \pi_0^* \geq -\rho\right) + \sum_{i\in\Omega_{\mathrm{V}}^1} \mathbf{1}\left(\boldsymbol{\pi}^*\boldsymbol{x}_i' + \pi_0^* < \rho\right) \tag{4.4a}$$

$$\text{s.t.} \sum_{j=1}^{n} u_j \leq \beta \tag{4.4b}$$

$$u_j \in \{0, 1\} \qquad \text{for } j = 1, \ldots, n \tag{4.4c}$$

$$(\boldsymbol{\pi}^*, \pi_0^*, \boldsymbol{\xi}^*) \in \underset{(\boldsymbol{\pi},\pi_0,\boldsymbol{\xi})\in X(\boldsymbol{u})}{\operatorname{argmin}} \left\{ \frac{1}{2}||\boldsymbol{\pi}||_2^2 + \sum_{i\in\Omega_{\mathrm{T}}} \xi_i \right\}, \tag{4.4d}$$

where $X(\boldsymbol{u}) \triangleq \left\{(\boldsymbol{\pi}, \pi_0, \boldsymbol{\xi}) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_+^{|\Omega_{\mathrm{T}}|} \mid y_i(\boldsymbol{\pi}'\boldsymbol{x}_i + \pi_0) \geq \rho - \xi_i \ \forall i \in \Omega_{\mathrm{T}},\ -M\boldsymbol{u} \leq \boldsymbol{\pi} \leq M\boldsymbol{u}\right\}$. In both bilevel models (4.3) and (4.4), the upper-level decisions (i.e., selection of features $\boldsymbol{u}$) are influenced by the lower-level decision variables (i.e., $\boldsymbol{\pi}$ and π_0 weights) and vice-versa. In a single-level approach features will be selected based on the training data without any consideration of the out-of-sample performance in the validation data. However, in the proposed bilevel feature selection approach the out-of-sample performance is explicitly considered in the upper-level problem.

4.3 Proposed genetic algorithm solution approach

We discuss the computational complexities associated with bilevel programs. We then propose a genetic algorithm as a solution approach. We conclude this section with a discussion on the mesh-adaptive direct search algorithm [5] which is used as a benchmark for the performance of our genetic algorithm.

4.3.1 Complexity of bilevel programs

Bilevel programs (BPs) are strongly NP-hard, even in their simplest form with only continuous variables in both levels [95]. For BPs consisting of mixed-integer variables in the upper and lower levels, Moore and Bard [156] proposed a branch-and-bound algorithm and DeNegre [56] improved this algorithm using cutting planes. Lozano and Smith [140] studied binary mixed integer programs (BMIPs) with integer upper-level variables and proposed an exact algorithm based on value function reformulation. Wang and Xu [218] presented an exact algorithm for the bilevel integer linear programming problem. Their algorithm uses a multiway disjunction cut to remove bilevel infeasible solutions from the search space. Vicente et al. [210] used penalty functions to reformulate BMIPs into bilinear programs. Audet et al. [6], Dempe [53] and Wen and Yang [220] proposed solution approaches for specific classes of BMIPs where either the leader's or the follower's variables are all continuous. Dempe and Richter [55], Özaltın et al. [160] and Brotcorne et al. [27] considered the bilevel knapsack problem, an extension of the classical knapsack problem to the bilevel framework. Detailed surveys on bilevel programming solution techniques were presented in [9, 54, 153]. Solution approaches for BPs with nonlinear objective functions were discussed in [20, 64, 67, 113, 175, 211]. Outside of the exact solution methods, there has been investigation into solving BPs through metaheuristics [34, 77, 99, 217, 221]. Due to computational complexities associated with solving bilevel programs and the size of our instances, we implement a genetic algorithm to solve bilevel models (4.3) and (4.4).

4.3.2 Genetic algorithm

A genetic algorithm (GA) is an evolutionary metaheuristic that generates candidate solutions from known competitive solutions. It represents solutions as "chromosomes", and just as in the natural evolution process, new generations of solutions are created through mutation and crossover [159, 183, 227]. In GAs, a fitness value needs to be defined to characterize the strongest chromosomes that will be used when creating the next generation of candidate solutions. We define the fitness value F for chromosome (upper-level solution) $\boldsymbol{u}$ as the total number of mis-classifications in the validation set, i.e., the upper-level objective function value.

We present a high-level description of the proposed GA in Table 4.1, and provide the implementation details in Appendix B.2. In Step 0, we initialize the algorithm by storing P_{initial} number of feasible chromosomes in set S, which denotes the current population. We randomly choose β features to generate each chromosome $\boldsymbol{u} \in \{0,1\}^n$ in the initial population and calculate their fitness values. Selecting critical features for certain observation types may be enforced at this step. For a given chromosome $\boldsymbol{u}$, let $\boldsymbol{v_u} \in \mathbb{R}^6$ denote the performance vector of agreement, specificity and sensitivity

Table 4.1 Genetic algorithm for solving the bilevel feature selection problem.

Step 0:	Given β, $\boldsymbol{L}$, and $\boldsymbol{w}$, generate P_{initial} chromosomes and store them in set S. Set probability of crossover and mutation, P_c and P_m, respectively. Set the iteration counter $k \leftarrow 1$, and let $G_{\max}$ be the maximum number of iterations.
Step 1:	If $k = G_{\max}$, report the strongest solutions in S and stop. Otherwise, go to Step 2.
Step 2:	Reduce S via `tournament selection`. Perform `crossover` and `mutation` operations to generate new feasible chromosomes.
Step 3:	Evaluate the fitness values of the new chromosomes and store them in S. Update $k \leftarrow k+1$ and go to Step 1.

in the training and validation sets. We say $\boldsymbol{u}$ is non-dominated if $\nexists \boldsymbol{j} \in S$ such that $v_{\boldsymbol{j}} \geq v_{\boldsymbol{u}}$ component-wise. Furthermore, let $\boldsymbol{L} \in \mathbb{R}^6$ be the desired performance vector. We define the weighted deviation from the desired level of performance as $W_{\boldsymbol{u}} = \boldsymbol{w}^T \min\{\boldsymbol{0}, \boldsymbol{v}_{\boldsymbol{u}} - \boldsymbol{L}\}$, where $\boldsymbol{w} \in \mathbb{R}^6$ is a set of weights specified by the user.

In Step 1, we identify all non-dominated chromosomes in S and store them in set N_D, representing the set of "strongest" chromosomes at the end of a given iteration. If the maximum number of generations has been reached, we terminate and return the solutions in N_D. Otherwise, we proceed to Step 2 where the population S is reduced via `tournament selection` so that chromosomes that have relatively low fitness values are removed. We then perform `crossover` and `mutation` on the reduced population to generate new feasible chromosomes. Finally, in Step 3, we evaluate the fitness values of the new chromosomes, store them in S, update the generation counter and repeat the whole process starting from Step 1. We implement and test the performance of the proposed GA in three separate case studies in our computational experiments.

4.3.3 Mesh-adaptive direct search

As a comparison to our genetic algorithm, we solve bilevel models (4.3) and (4.4) also by using mesh-adaptive direct search (MADS), a derivative-free optimization method, as implemented in open-source software NOMAD [5]. The MADS algorithm can be utilized to optimize functions that have no exploitable properties (e.g., derivatives) or are difficult to evaluate. Starting from an initial solution, this algorithm iteratively tries to improve the current best solution by generating trial points on a mesh, which is a discretization of the variable space. Each iteration is composed of two main steps: the search and poll steps.

The search step evaluates a number of trial mesh points. If an improved mesh point is found, then the next iteration is initiated with the new incumbent solution using a larger mesh size. Whenever

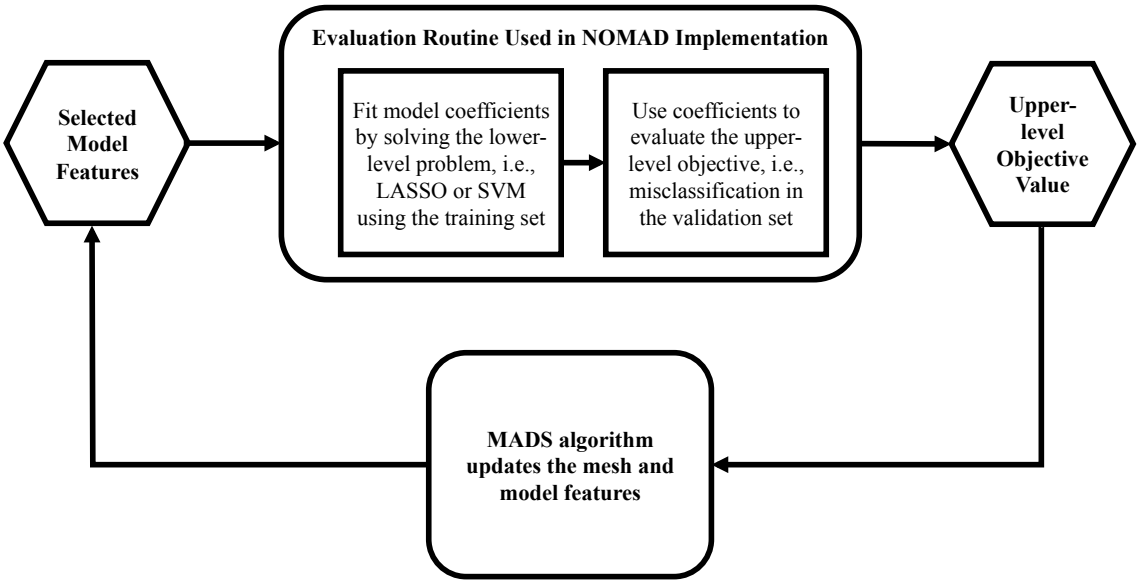


Figure 4.1 Solving the bilevel feature selection problem using the MADS algorithm implementation in NOMAD

the search step fails to generate an improved mesh point, then the poll step is invoked. The poll step explores the variable space near the current incumbent solution. If the poll step also fails to improve the current best solution, then the mesh size and poll size parameters are reduced in order to increase the search resolution. The MADS algorithm stops after a given number of iterations or when the mesh size reaches a precision limit. We refer the reader Le Digabel (2011) for more information about the MADS algorithm and the open-source software package NOMAD [129].

To use the MADS algorithm implementation in NOMAD, we create a routine to evaluate the upper-level objective function value for a given set of selected features. This routine calls the statistical software package R to fit coefficients for the selected features by solving the LASSO (4.1) or SVM (4.2) model using the training data in the lower-level. It then utilizes these coefficients to compute the upper-level objective function (4.3a) or (4.4a) based on the validation data. Figure 4.1 presents a flow-chart of our implementation to solve the bilevel feature selection problem using the MADS algorithm. We now discuss the implementation of the methods discussed on three distinct case studies beginning with the classification of antigenic variants in the influenza virus.

4.4 Case study: antigenic variants in influenza viruses

Seasonal influenza epidemics impact 5-15% of the world's population, resulting in 3-5 million cases of severe illnesses and up to 500,000 deaths annually [36]. The first line of defense is the influenza vaccine that contains A/H1N1, A/H3N2 and B virus strains [37]. Influenza A virus is of particular importance, as accumulated point mutations (antigenic drifts) in its hemagglutinin (HA) surface protein can generate immunologically different strains, i.e., antigenic variants, which require frequent updates in the annual influenza vaccine composition [158]. Modeling the antigenic distance between influenza strains can provide a rapid indication of the current flu vaccine's effectiveness against a recently emerged strain, and also facilitates the study of the virus' evolution in response to antibody pressure. In current practice, hemagglutinin inhibition (HI) assays are used to identify antigenic variants of influenza A strains. However, this assay does not identify the amino acid positions on the HA surface protein. The evolution of influenza virus integrates multiple antigenic properties with different molecular functions and this coupled dynamics can be captured through genetic models [3].

The goal of this case study is to build a binary classification model to categorize influenza virus strain pairs as antigenically different or similar using genetic data. Lee and Chen [134] studied the correlation between the number of amino acid mutations and antigenic variety (or distance). Liao et al. [136] explored the use of scoring methods, such as the construction of similarity classes and substitution matrices, to quantify scores for amino acid mutations. Based on these scoring methods, they employed iterative filtering, multiple and logistic regression and support vector machines to identify the antigenic difference between the two influenza virus strains. In these machine learning models, the pairwise comparison of amino acid sequences of strains are considered as independent variables and the antigenic distance between each strain pair is considered as dependent variable. The immunodominant positions or amino acid substitutions that can potentially lead to antigenic variety, i.e., positively associated with antigenic distance, are identified through feature selection.

4.4.1 Data and bioinformatics model

The HA surface protein of the influenza A virus induces antibody response. It consists of three identical subunits. Each subunit has two chains, HA1 and HA2, which are 329 and 175 amino acid residues long, respectively. The HA1 chain is used for studying antigenic variety because it mutates more frequently than the HA2 chain [134, 193]. Hence, by "sequence" of an influenza A strain, we refer to the amino acid sequence of its HA1 chain. Pairwise sequence alignment methods are used for assessing genetic difference. These methods compare the amino acid positions in a pair of

Table 4.2 Sample alignment vectors using two different groupings.

Sequence of strain 1	$\cdots$	A	B	C	D	K	V	Y	Q	F	$\cdots$
Sequence of strain 2	$\cdots$	F	B	Q	V	E	W	T	M	M	$\cdots$
GM1 alignment vector	$\cdots$	0	0	0	1	0	0	0	1	0	$\cdots$
GM2 alignment vector	$\cdots$	1	0	0	1	0	1	0	1	0	$\cdots$

sequences, and produce a vector denoting differences at each position. The simplest alignment method assigns zero or one to each position in the sequence based on whether two amino acids at that position are the same or not. More advanced alignment methods incorporate polarity, charge and structure of amino acids by grouping them into similarity classes [66]. These alignment methods assign zero to a position if the amino acids at that position are in the same similarity class, or one otherwise. We employ six different groupings introduced by Liao et al. [136]. The similarity classes in each grouping are given in Appendix Table B.1. Table 4.2 illustrates an example for the alignment of two sequences using alignment methods GM1 and GM2.

Let $\boldsymbol{x}_i \in \{0,1\}^n$ $(n = 329)$ be the alignment vector of strain pair i generated by comparing the sequences of strains s and ℓ. This vector will be used to explain the antigenic distance between strains s and ℓ, which is denoted as $g_{s\ell}$. The value of $g_{s\ell}$ is calculated by [121]:

$$g_{s\ell} = \left(\sqrt{c_{s\ell}\, c_{\ell s} / c_{ss}\, c_{\ell\ell}}\right)^{-1}. \tag{4.5}$$

In (4.5), $c_{s\ell}$ is the minimum antiserum concentration, raised against influenza strain s, that completely inhibits the agglutination of strain ℓ in the HI assay. Smith et al. [193] categorized strains s and ℓ as antigenic variants if $g_{s\ell} \geq 4$, and as similar strains if $g_{s\ell} < 4$. Accordingly, we set the response variable $y_i = 1$ if $g_{s\ell} \geq 4$, and $y_i = 0$ otherwise. We compile an HI assay data set containing 45 HA1 amino acid sequences of A/H3N2 viruses isolated between 1971 and 2002 from Lee and Chen [134]. Among these, 181 pairwise antigenic distances are computed and placed in the training set. We include 96 pairwise antigenic distances of 19 viruses isolated from 1999 to 2004 in the validation set.

4.4.2 Genetic algorithm implementation

We use the notation in Table 4.3 when describing the details of our GA implementation for the influenza virus classification case study. The upper-level feature selection variable $u_j = 1$ if amino acid position $j = 1 \ldots 329$ is selected for the classification model, and $u_j = 0$ otherwise. Let $P_k(j)$ be the proportion of observations (i.e., strain pairs) in $\Omega_{\mathrm{V}}^k \cup \Omega_{\mathrm{T}}^k$ where there is a "1" at position j of the

Table 4.3 Notation used in the influenza virus classification case study.

α :	min number of critical positions selected for each strain pair on average
β :	max number of features that can be selected
n :	number of positions in the amino acid sequence of a strain ($n = 329$)
Ω_{V} :	strain pairs in the validation set
Ω_{V}^1 :	antigenically different strain pairs in the validation set (antigenic distance ≥ 4)
Ω_{V}^2 :	antigenically similar strain pairs in the validation set (antigenic distance < 4)
Ω_{T} :	strain pairs in the training set
Ω_{T}^1 :	antigenically different strain pairs in the training set (antigenic distance ≥ 4)
Ω_{T}^2 :	antigenically similar strain pairs in the training set (antigenic distance < 4)
Ω :	set of all strain pairs ($\Omega = \Omega_{\mathrm{V}} \cup \Omega_{\mathrm{T}}$)
x_i :	alignment vector of strain pair $i \in \Omega$, $x_i \in \{0,1\}^n$
y_i :	1 if strains in pair i are antigenically different and 0 otherwise for $i \in \Omega$

alignment vector, for $k = 1$ (antigenically different strain pairs) and $k = 2$ (antigenically similar stain pairs). We define position j as "critical" for antigenically different strain pair $i \in \Omega_V^1 \cup \Omega_T^1$ if

a. $x_{ij} = 1$, i.e., there is a mutation at position j in the alignment vector of strain pair i, and $P_1(j) > P_2(j)$, i.e., there are proportionally more mutations at this position in the set of antigenically different strain pairs than in the set of similar ones,

b. $x_{ij} = 0$, i.e., there is no mutation at position j in the alignment vector of strain pair i, and $P_1(j) < P_2(j)$, i.e., there are proportionally less mutations at this position in the set of antigenically different strain pairs than in the set of similar ones.

The critical positions for pairs of similar strains are defined in the same fashion as presented in Table 4.4. Let parameter d_{ij} be equal to 1 if position j is critical for strain pair i, and 0 otherwise. In

Table 4.4 Definition of critical positions in the amino acid sequence. $d_{ij} = 1$ means critical position, $d_{ij} = 0$ means not critical position

	$P_1(j) > P_2(j)$		$P_1(j) < P_2(j)$		$P_1(j) = P_2(j)$	
d_{ij}	$x_{ij} = 1$	$x_{ij} = 0$	$x_{ij} = 1$	$x_{ij} = 0$	$x_{ij} = 1$	$x_{ij} = 0$
$i \in \Omega_{\mathrm{V}}^1 \cup \Omega_{\mathrm{T}}^1$	1	0	0	1	0	0
$i \in \Omega_{\mathrm{V}}^2 \cup \Omega_{\mathrm{T}}^2$	0	1	1	0	0	0

our GA implementation, we require that on average $\alpha \geq 0$ critical positions are selected for each strain pair in the dataset (i.e. satisfy the constraint $\sum_{i \in \Omega} \boldsymbol{u}^T \boldsymbol{d}_i \geq \alpha |\Omega|$). In the computational experiments, we

set $\alpha = 1$, and allowed for the selection of at most $\beta = 40$ positions in the upper-level constraints (4.3b) and (4.4b). Furthermore, we set the initial population size $P_{\text{initial}} = 200$. When calculating the fitness value of a given chromosome, we set the weight vector as $\boldsymbol{w} = [0.15, 0.15, 0.15, 0.25, 0.15, 0.15]$, and used the SVM results from Liao et al. [136] as the desired performance $\boldsymbol{L} \in \mathbb{R}^6$. We set the parameters of the genetic algorithm through preliminary computational experiments. One of the most critical parameters is β, i.e., maximum number of features that can be selected in the upper level. We present sensitivity analysis results on this parameter in Appendix Tables B.9 - B.11. Individual steps of our GA implementation are provided in Appendix B.2.

4.4.3 Results and discussion

We build a single-level Lasso model (4.1) using the training set. We then build another Lasso model using the proposed bilevel framework (4.3) which incorporates both the training and validation sets. The R package `glmnet` is called when the Lasso model (4.1) needs to be solved. Liao et al. [136] constructed a single-level SVM model (4.2) utilizing the same training set that we have. We build an SVM model using the proposed bilevel feature selection framework (4.4). We also solve each of the bilevel models using `NOMAD` (Section 4.3.3). The results obtained when pairwise strain comparisons are made via alignment method GM1 are displayed in Table 4.5. In the process of producing these results, the R package `e1071` is called when the SVM model (4.2) needs to be solved. Meta-parameters of the SVM and Lasso-based logistic regression models are automatically chosen by the corresponding R packages. Results from using other alignment methods described in Section 4.4.1 are given in Appendix Tables B.3-B.8.

As seen in Table 4.5, bilevel models are able to achieve better performance with the use of fewer features than their single-level counterparts. Comparing the results generated by solving the bilevel models using our proposed genetic algorithm to the results reported in Liao et al. [136], we observe that approximately 70% of the performance indicators are improved. In addition to SVM, Liao et al. [136] also used iterative filtering, multiple regression and logistic regression to build classification models. The number of positions selected by Liao et al. [136] ranges from 9 to 44, and they identified 14 and 16 amino acid positions as sufficient for detecting antigenic variety. This is comparable to the results of our bilevel models that select 14 positions when alignment method GM6 is implemented, see Appendix Table B.7. We now discuss our second case study in the classification of digital colposcopy images.

Table 4.5 Results for influenza A virus classification*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	35	93.4	96.0	87.5	80.2	94.7	76.6
LASSO	52	93.9	96.8	87.5	74.0	100.0	67.5
NOMAD SVM**	35	92.8	96.8	83.9	79.2	94.7	75.3
NOMAD LASSO**	38	93.9	96.8	87.5	75.0	100.0	68.8
GA SVM**	14	92.3	94.4	87.5	82.3	94.7	79.2
GA LASSO**	19	93.4	96.0	87.5	87.5	94.7	85.7

* The alignment method GM1 is used. Agreement, sensitivity and specificity are in percentages. SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].
**Results from the corresponding bilevel models.

4.5 Case study: digital colposcopy quality classification

Cervical cancer is a significant cause of mortality, with approximately one half million diagnosed cases per year and a quarter million of these cases resulting in death globally [18]. A main preventative measure to cervical cancer is screening for precancerous lesions using digital colposcopy [18]. This procedure involves viewing images of the cervix, i.e., cervigrams, under three different modalities: green filter, Hinselmann and Schiller. Opaque images may imply cervical lesions [18], therefore it is important to choose good quality images, i.e., those that assist physicians in making more accurate diagnosis [69]. In this case study, we apply the proposed bilevel framework to select features of a cervigram that determine its image quality.

4.5.1 Data

A dataset is downloaded from the University of California, Irvine Machine Learning Repository [58] containing approximately 100 cervigrams per viewing modality. There are 50 continuous predictor variables in the data encompassing the following features [69]:

1. Image area (cervix, external orifice and vaginal walls) and occluding objects (speculum and other artifacts)
2. Area of each region occluded by artifacts or by specular reflections
3. The maximum area difference between the four cervix quadrants
4. Goodness of fit of the cervix to a geometric model
5. Distance between image center and the cervix centroid/external orifice

6. Mean and standard deviation from each RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) channel in the cervix area and in the entire image

The response variable is whether or not the image constituted *good* or *bad* quality. For each observation (cervigram), six expert opinions are compiled and a consensus is drawn among the six on the caliber of each image. We considered this consensus to be the underlying truth of the image quality with the goal of identifying which features contributed most to this quality.

In our experiments, we predict the quality of the Hinselmann modality images. We do not include the data from the Schiller and green filter images. The modality is not considered in the feature space and therefore, we want to remove any bias that each specific modality may introduce into the response of the six experts. There are 97 observations in this data set from which 64 and 33 are placed into training and validation sets, respectively. Of the 64 observations in the training set, 54 are good quality images and 10 are bad quality. Of the 33 observations in validation set, 28 are good quality and 5 are bad quality images.

4.5.2 Results and Discussion

We compare our proposed bilevel model to their single level counter parts. We also compare our genetic algorithm solution to the solution generated by `NOMAD`. Table 4.6 displays the number of features selected (that have non-zero weight) as well as agreement, sensitivity, and specificity in the training and validation sets. We set the parameter $\beta = 20$ to generate the bilevel results seen in Table 4.6. The remaining parameters used for our genetic algorithm are the same as in the influenza A virus classification case study in Section 4.4.2. As seen in Table 4.6, although the performance in the training set remains unchanged when moving from single-level models to the bilevel models (in fact, there is a slight decrease in the performance of SVM models), the performance in the validation set increases. Also, irrespective of the solution approach, the bilevel models select fewer features than the corresponding single-level models.

An important characteristic of this data is that the ratio of positive to negative outcomes is large as opposed to the data used in the previous case study. Specifically, there are 82 good quality images (positive outcomes) and 15 bad quality images (negative outcomes). Although, this might create a bias toward positive outcomes and decrease out-of-sample specificity of the single-level models, observe that, in the context of the holdout cross validation, the proposed bilevel framework optimizes the out-of-sample performance using a minimal number of features. Therefore, we conclude that the proposed bilevel framework is robust with respect to the ratio of positive to negative outcomes in the data.

Table 4.6 Results for the quality assessment of digital colposcopy images*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	50	96.9	98.1	80	66.7	71.4	40.0
LASSO	30	100.0	100.0	100.0	69.7	75.0	40.0
NOMAD SVM**	19	85.9	98.1	20.0	90.9	96.4	60.0
NOMAD LASSO**	20	100.0	100.0	100.0	87.9	92.9	60.0
GA SVM**	17	95.3	98.1	80.0	93.9	96.4	80.0
GA LASSO**	18	100.0	100.0	100.0	84.8	85.7	80.0

*Agreement, sensitivity and specificity are in percentages.
*All SVM models solved using R package `e1071` [115]. All LASSO models solved using R package `glmnet` [73].
**Results from the corresponding bilevel models.

4.6 Case study: splice junction recognition

Genes are composed of alternated segments of *exons* and *introns*. Exons correspond to regions that are translated into proteins, and introns to regions that do not code for proteins. The process of protein coding from a gene is called gene expression. There are two phases of gene expression: transcription and translation. In the transcription phase, a mRNA (messenger Ribonucleic Acid) is synthesized from a DNA (Deoxyribonucleic Acid). The protein coding is performed in the translation phase using the mRNA sequence as a model. In this phase introns are spliced out from the mRNA. Splice junctions are the boundary points on the gene where splicing occurs. The mutation/deletion of junction sites in mRNA may lead to the production of abnormal proteins and many different types of diseases stem from these anomalies including breast and ovarian cancer [74], dementia [106], and some types of epilepsy [162]. For a more detailed description of splice junctions, the reader is referred to [49, 150, 180]. The splice junction recognition problem pertains to the identification of exon-intron (EI) and intron-exon (IE) sites on a genetic sequence [86, 150].

4.6.1 Data

We sample 1,535 DNA gene sequences downloaded from the University of California, Irvine Machine Learning Repository consisting of 767 EI sites and 768 IE sites [58]. Each sequence contains 60 nucleotides (features) represented by a set of letters. Since features are characters, the packages we use to solve the SVM and LASSO problems do not accept them as input. Therefore, we represent each of the characters using a binary encoding. Characters used for nucleotides and their binary representations are given in Table 4.7. Since we use three dimensional binary vectors to represent the nucleotides, our observations become a vector of length 180 (60×3). We give an example of this binary encoding in Table 4.8.

Table 4.7 Binary representation of nucleotides used when generating the observation vectors

Character	Nucleotide	Binary Representation
A	Adenine	[0,0,0]
G	Guanine	[1,0,0]
C	Cytosine	[0,1,0]
T	Thymine	[0,0,1]
D	A or G or T	[1,1,0]
N	A or G or C or T	[1,0,1]
S	C or G	[0,1,1]
R	A or G	[1,1,1]

Table 4.8 Example binary representation of a gene sequence.

Position	1-20	21			22			23			$24-60$
Gene sequence	$\cdots$	R			D			T			$\cdots$
Binary representation	$\cdots$	1	1	1	1	1	0	0	0	1	$\cdots$

Due to binary encoding of nucleotides, the solution to each of the models may have non-zero weights for each of the 180 elements in the observation vector. If any of the three elements that make up the binary encoding of a nucleotide have non-zero weight in the solution generated by the SVM or LASSO, then we say that nucleotide (or position on the gene sequence) is selected. Otherwise, if all three elements in the binary vector have zero weight, we say that nucleotide is not selected. We partition the 1,535 sampled sequences into training and validation sets containing 845 (537 EI sites and 308 IE sites) and 690 (230 EI sites and 460 IE sites) sequences, respectively.

4.6.2 Results and Discussion

As in the previous case studies, we compare the proposed bilevel models to their single level counter parts and our genetic algorithm solution to the solution generated by `NOMAD`. Table 4.9 displays the number of features selected (that have non-zero weight) as well as agreement, sensitivity, and specificity for the training and validation sets. We set the parameter $\beta = 20$ to generate the genetic algorithm bilevel results seen in Table 4.6. The remaining parameters of genetic algorithm are the same as in Section 4.4.2.

As seen in Table 4.9, the single level models perform well in classifying this data. In fact, with respect to the training set performance, both single level models slightly dominate their corresponding bilevel models when solved by both the `NOMAD` and the proposed genetic algorithm. Furthermore, observe

that the single level LASSO model outperforms the bilevel LASSO model with respect to agreement and specificity in the validation set. Although the performance seems to remain the same on average, an important point to note is the number of features (positions in the gene sequence) selected in each model. The bilevel models, irrelevant of the solution approach, use significantly fewer features. This strengthens our claim that, in the context of holdout cross validation, the proposed bilevel framework optimizes the out-of-sample classification while identifying the minimal set of features required to do so.

In the splice junction recognition problem, genes can also be identified as having neither an EI nor an IE site. Actually, these types of observations are also available in our data. A limitation to this specific case study is that we exclude these observations, because we focus our findings on those models that involve binary classification.

Table 4.9 Results for splice junction recognition*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	60	100.0	100.0	100.0	94.3	93.0	95.0
LASSO	50	100.0	100.0	100.0	95.5	93.5	96.5
NOMAD SVM**	17	98.0	98.7	96.8	94.9	95.7	94.6
NOMAD LASSO**	17	97.2	98.1	95.5	94.9	94.8	95.0
GA SVM**	20	99.8	100.0	99.4	97.4	97.8	97.2
GA LASSO**	19	100.0	100.0	100.0	94.5	97.0	93.3

*Agreement, sensitivity and specificity are in percentages.
*All SVM models solved using R package `e1071` [115]. All LASSO models solved using R package `glmnet` [73].
**Results from the corresponding bilevel models.

4.7 Summary

We propose a bilevel approach to feature selection for classification models. The upper-level problem selects model features to maximize the classification performance on the validation set, while the parameters of the selected features are learned by the lower-level problem based on the training set. We consider holdout cross validation for out-of-sample performance evaluation, however the proposed bilevel framework can be extended easily to consider a more general cross validation with multiple folds.

We implement a tailored genetic algorithm to solve bilevel models (4.3) and (4.4) and compared this solution approach to a derivative-free optimization solver `NOMAD`. Currently, the exact solution approaches for bilevel programs do not scale well for the size of the instances that we need

to solve. However, bilevel models (4.3) and (4.4) have only binary variables in the upper level. Zhang and Özaltın [231] have recently developed a value function approach for such bilevel programs and reported promising computational results with large instances up to 200 variables. Future research will investigate exact solution approaches to large-scale bilevel feature selection problems.

We present three different case studies where the proposed bilevel framework is applied for feature selection. In particular, we consider the classification influenza strains based on antigenic variety, the assessment of good and bad quality digital colposcopy images, and the identification of splice junction sites in genetic sequences. The computational results show that the proposed bilevel feature selection approach can improve the classification performance in terms of agreement, sensitivity and specificity with fewer selected features.

CHAPTER

5

MIXED INTEGER OPTIMIZATION FRAMEWORK FOR SEVERITY OF ILLNESS SCORING SYSTEMS

In this chapter, we propose a new mixed integer programming framework for the learning of *interpretable* severity of illness scoring systems from data. We apply the proposed framework to develop a score that can be used to track the acuity of patients who are susceptible to sepsis. Section 5.1 gives a brief introduction and review of the literature for the problem. We present the frame work and solution methodology in Section 5.2. Section 5.3 displays our results and we give a discussion of these results in Section 5.4.

5.1 Introduction

5.1.1 Background

Scoring systems are widely used across multiple application domains to assist decision makers in quickly measuring the state of a system [208]. In medicine, these *severity of illness scoring systems* are

used to assess a patients risk for a particular outcome. These scores can be used in an online fashion for tracking patient acuity during a hospital stay like the Acute Physiology and Chronic Health Evaluation (APACHE) classification system [2, 122], Sequential Organ Failure Assessment Score (SOFA) [213], Mortality in Emergency Department Scores (MEDS) [185], Multiple Organ Dysfunction Score (MODS) [147] and Predisposition, Infection, Response, Organ dysfunction (PIRO) scoring system [35, 44, 102]. They can also be used to guide diagnostic protocols to assess the risk of a patient developing certain conditions [120, 187, 230] or having adverse events given a particular condition [76, 80, 89, 143, 191, 200]. These scores are traditionally calculated by adding or subtracting small numbers that are a function of a patients personal information or medical status (patient demographics, historical diagnosis, comorbid conditions, vital and lab readings, etc.). These risk scores are highly favored in many applications due to the ease of implementation (i.e. decision makers need only to add a few numbers together to gauge the state of a patient or system). Despite the diversity of medical conditions, most of the scores focus on predicting the likelihood of a critical patient outcome, e.g., 10-year mortality or 30-day readmission. Vincent and Moreno [214] provide a review of those more general severity of illness scoring systems and their applications.

5.1.2 Sepsis score development

Sepsis, infection plus systemic manifestations of infection, is the leading cause of in-hospital mortality [139]. About 700,000 people die annually in the US hospitals and 16% of them are diagnosed with sepsis (including a high prevalence of severe sepsis with major complications) [188]. In addition to being deadly, sepsis is the most expensive condition associated with in-hospital stay, resulting in a 75% longer stay than any other condition [94]. In 2011, the total burden of sepsis to the United States (US) healthcare system is estimated to be $20.3 billion, most of which is paid by Medicare and Medicaid [203]. This accounted for 5.2% of the total aggregate costs for hospitalizations in the US resulting as the single most expensive treated condition in that year [203]. The monitoring of sepsis using clinical criteria for infection and organ dysfunction has been made possible through the emergence and use of electronic health records (EHRs) [167]. These clinical criteria relate to the observed physiology of a patient and they can often precede clinical deterioration for those susceptible to sepsis. This has prompted the development of early warning systems, many in the form of *scores*, that allow for the earlier identification of this physiological deterioration [47].

The construction of these severity of illness scores is accomplished by combining medical expertise with statistical learning from EHR data [35, 44, 102, 117, 122, 142, 185, 213]. The general process for the development of many of these scores is as follows [17, 35, 43, 68, 102, 122, 147, 184]:

1. Identify candidate predictor variables

2. Use statistical tests (χ^2, Fisher's exact test, etc.) to determine the significance level each predictor has with respect to predicting an outcome of interest

3. Select predictor variables to be used in a logistic regression model [101] based on significance

4. Apply a *transformation* to the regression-coefficients (e.g. multiplying by an arbitrary constant) and round this transformed value to the nearest integer to get the point value for that variable

A main disadvantage of this structure for developing scores is in step 4. First, it is well known in the optimization community that the rounding of relaxed solutions for integer programs will more than often lead to sub-optimal or infeasible solutions. Next, many times the resulting coefficients from statistical learning models (e.g. logistic regression) are small and rounding could eliminate important variables. Finally, when transformations are applied in an effort to keep those variables that have small coefficients in the score, this reduces the calibration resulting in extreme weights and scores. Some scores apply additional methodologies in this process such as combined variable weighting [122] and bootstrapping to prevent overfitting [68, 184]. Cardoso et al. [35] developed a score using the PIRO framework. Alongside using a logistic regression model, they utilized a decision tree to define cut-offs for the component scores. There are a few scores that do not implement statistical methodologies in the development phase and are constructed only through medical expertise [174, 213]. The Sequential Organ Failure Assessment Score (SOFA) [213] is one of these scores and is one of the most used in the assessment of sepsis acuity to-date [33, 112, 142, 165, 190, 214]. There are some scores that are created by combining other scores (variables and point values) such as the Sepsis in Obstetrics Score (SOS) [4] which utilizes predictors of the APACHE, REMS and SIRS scoring systems. It's unclear how the point values for these types of scores are determined. Calle et al. [33] provides a systematic review of severity scores in patients with suspected infection in the emergency department in an effort to determine the accuracy of severity of illness scores in predicting mortality in these patients. They conclude that the reviewed literature did not provide enough information to assess the accuracy of the prognostic models in patients with suspected infection admitted to the ED or hospital ward.

5.1.3 Mixed integer programming for score development

Although most of the aforementioned scores have been validated as well calibrated and accurate in the prediction of adverse outcomes, we believe that there is an opportunity for integer optimization to be used in the construction of stronger scoring systems, specifically in the sepsis spectrum. Allowing for the point values on each of the predictors to be integer combines interpretability (ease for practitioners to assess score performance) and implementation (use in hospital practice). Ustun and Rudin [208] propose a new machine learning approach to the development of risk scores. They

formulate the problem as a mixed integer nonlinear program and propose a new lattice cutting plane algorithm as a solution method. They apply their framework in different application settings (criminal justice, medicine and finance) concluding that their models yield strong calibration and high prediction performance while avoiding some of the pitfalls that can arise from the use of heuristic methods (e.g. rounding to obtain integer point values). Risk estimate outputs are another advantage to using the framework proposed by Ustun and Rudin. Given a score, they estimate a predicted risk through the use of a logistic link function [101].

5.2 The mixed integer framework

In this section, we propose the use of a mixed-integer programming model in the construction of an *interpretable* scoring system that can be used to assess the risk an individual has for a set of outcomes. While similar to that of the work done by Ustun and Rudin [208], we utilize a different mapping from score to estimated risk (as opposed to a logistic link function). We also build the framework with a progressive sepsis trajectory in mind. The model proposed by Ustun and Rudin was constructed for binary outcome prediction. However, it is well accepted that the sepsis disease moves along the spectrum of infection, inflammation, organ dysfunction, and septic shock characterized by low blood pressure [24, 181, 190, 206]. Therefore, our framework produces risk estimates for multiple classes of a disease (e.g., stages of sepsis). This is done by allowing the risk estimate to be a decision variable in the model itself as opposed to a logit model that can only account for binary outcomes. We implement a hybrid heurstic solution approach combining the Alternate Direction of Multipliers Method and coordinate decent, described in Section 5.2.2.

5.2.1 The model

We define the parameters and variables of our model in Tables 5.1 and 5.2, respectively. An observation i is a feature and response vector pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. Key contributions that we make with this model are the following:

1. Incorporation of a risk decision variable that is directly optimized (q_{sk})
2. Ability to provide risk estimates for multiple classes (e.g. considering many adverse outcomes or stages in a disease)
3. Freedom to apply operational constraints

- Clinical context: If systolic blood pressure is < 90 for two *consecutive* measurements, then patient is in septic shock
- Group features: If African American male with history of diabetes, add two points
- Adjustment in class prediction: The risk score should predict septic shock with X% confidence

Table 5.1 Parameters for Mixed Integer Score Development Model

N_i :	# of non-zero features in observation i
K :	# of classes
P :	Max point value for any feature in the final score
γ :	Regularization coefficient used to balance sparsity and model accuracy
S :	# of score intervals
$\boldsymbol{l}_s$:	Lower bound of score interval s
$\boldsymbol{u}_s$:	Upper bound of score interval s
Ω :	Set of observations $= \left\{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{N \times K}\right\}$
Ω^{T} :	Set of observations that will be used for model construction
Ω^{V} :	Set of observations held for validation $= \Omega \setminus \Omega^{\mathrm{T}}$
$\boldsymbol{y}_i$:	Response vector of observation $i \implies y_{ik} = \begin{cases} 1 & \text{if observation } i \in \Omega \text{ is in class } k \\ 0 & \text{otherwise} \end{cases}$
$\boldsymbol{x}_i$:	Feature vector of observation $i \implies \boldsymbol{x}_i \in \mathbb{R}^N$ for each $i \in \Omega$

Table 5.2 Variables for Mixed Integer Score Development Model

$\boldsymbol{\lambda}$:	vector of point values for each of the attributes $\implies \boldsymbol{\lambda} \in \{0, 1, \ldots P\}$
q_{sk} :	probability that an observation with score s is in class k for $k = 1, \ldots, K$
	Let $\boldsymbol{Q}$ be the matrix representation of these variables
z_{is} :	$\begin{cases} 1 & \text{if observation } i \text{ has score } s \text{ (i.e. } \boldsymbol{\lambda}'\boldsymbol{x}_i \in [l_s, u_s]) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in \Omega$
	Let $\boldsymbol{Z}$ be the matrix representation of these variables.

$$\max_{\boldsymbol{\lambda}, \boldsymbol{Z}, \boldsymbol{Q}} \frac{1}{|\Omega^{\mathrm{T}}|} \sum_k \sum_s \sum_{i \in \Omega^{\mathrm{T}}} y_{ik} z_{is} q_{sk} - \gamma \sum_{j=1}^{N} \lambda_j^2 \tag{5.1a}$$

$$\text{s.t. } 0 \leq \sum_i z_{is}(y_{ik} - q_{sk}) \quad \forall s, k \tag{5.1b}$$

$$l_s z_{is} \leq \boldsymbol{\lambda}' \boldsymbol{x}_i \leq u_s z_{is} + N_i P(1 - z_{is}) \qquad \forall i \in \Omega^{\mathrm{T}},\ \forall s \tag{5.1c}$$

$$\sum_k q_{sk} = 1 \qquad \forall s \tag{5.1d}$$

$$\sum_s z_{is} = 1 \qquad \forall i \in \Omega^{\mathrm{T}} \tag{5.1e}$$

$$q_{sk} \geq 0 \qquad \forall s, k \tag{5.1f}$$

$$\boldsymbol{\lambda} \in \{0, 1, \ldots P\} \tag{5.1g}$$

$$z_{is} \in \{0, 1\} \qquad \forall i \in \Omega^{\mathrm{T}},\ \forall s \tag{5.1h}$$

The objective function (5.1a) balances maximizing the average probability of correct classification and sparsity. Constraint (5.1b) states that the probability of an observation being in class k given that it is in score interval s (i.e. q_{sk}) does not exceed the proportion of class k observations that end up in score interval s, i.e., $q_{sk} \leq \frac{\sum_i y_{ik} z_{is}}{\sum_i z_{is}}$, $\forall s, k$. Constraint (5.1c) ensures that the variable $z_{i,s}$ takes on the appropriate value. Constraint (5.1d) states that for each score s, the sum of the probabilities of being in class k sum to 1 (make up an appropriate probability vector). Constraint (5.1e) ensures that each observation is placed in a single score interval. Constraints (5.1f) – (5.1h) are the variable restrictions. Objective function (5.1a) is non-linear in the variables $z_{i,s}$ and $q_{s,k}$. We linearize this by defining variable $w_{i,s,k} \equiv z_{i,s} q_{s,k}$ for $i \in \Omega$ and $\forall s, k$. Then, we rewrite model (5.1) as follows:

$$\max_{\boldsymbol{\lambda}, \boldsymbol{Z}, \boldsymbol{Q}, \boldsymbol{w}} \frac{1}{|\Omega^{\mathrm{T}}|} \sum_k \sum_s \sum_{i \in \Omega^{\mathrm{T}}} y_{i,k} w_{i,s,k} - \gamma \sum_{j=1}^{N} \lambda_j^2 \tag{5.2a}$$

$$\text{s.t. } w_{i,s,k} \leq z_{i,s} \quad \forall i \in \Omega^T, \forall s, k \tag{5.2b}$$

$$w_{i,s,k} \leq q_{s,k} \quad \forall i \in \Omega^T,\ \forall s \tag{5.2c}$$

$$w_{i,s,k} \geq q_{s,k} + z_{i,s} - 1 \tag{5.2d}$$

$$\sum_i w_{i,s,k} \leq \sum_i z_{i,s} y_{i,k} \quad \forall s, k \tag{5.2e}$$

$$w_{i,s,k} \in [0, 1] \quad \forall i \in \Omega^{\mathrm{T}},\ \ \forall s, k \tag{5.2f}$$

$$(5.1\text{c}) - (5.1\text{h})$$

For any given $i \in \Omega^{\mathrm{T}}$, $s \in \{1, \ldots, S\}$ and $k \in \{1, \ldots, K\}$, any tuple $(w_{i,s,k}, z_{i,s}, q_{s,k})$ satisfying constraints (5.2b)- (5.2d) also satisfy $w_{i,s,k} = z_{i,s} q_{s,k}$, and vice-versa. Using this transformation, we can rearrange terms in objective (5.1a) and constraint (5.1b) to obtain (5.2a) and (5.2e). Therefore, an optimal solution $(\boldsymbol{\lambda}^*, \boldsymbol{Z}^*, \boldsymbol{Q}^*, \boldsymbol{w}^*)$ to model (5.2) implies that the solution $(\boldsymbol{\lambda}^*, \boldsymbol{Z}^*, \boldsymbol{Q}^*)$ is optimal for model (5.1). We have now transformed problem (5.1) into a mixed integer linear program. There has been much work done in the formulation and solving of integer programs (IP), mixed integer

linear programs (MILP) and mixed integer nonlinear programs (MINLP). Chen et al. [42], Junger et al. [114] and Kumar et al. [126] give an overview of the theoretical, algorithmic, and software development in integer programming that has taken place over the last 50 years. Vielma [212] surveys the more recent advancements in MILP formulation techniques that can result in stronger and smaller formulations. The reader is referred to [23, 25, 29, 88, 197] for comprehensive reviews on methods in solving MINLPs. Heuristic methods have become a favorable choice as a solution approach in many applications that require the solving of large-scale mixed integer programs [21, 78, 84, 85].

One of main challenges in solving model (5.2) is that the number of binary variables (z_{is}) increases at a rate that is proportional to the size of our training observation set Ω^{T}. This implies that the computational complexity in solving model (5.2) is exponential in the size of the training set Ω^{T}. Solving model (5.2) with less observations would still yield a score that could be used in practice (i.e. a set of variables $\boldsymbol{\lambda}$ would still be an output irrelevant of the size of the set of observations used for training). Hence, we can partition the training set Ω^{T} into smaller subsets that could still be used to yield a score. With this partitioning idea in mind, we decide to employ the Alternate Direction Method of Multipliers (ADMM) as a solution approach to our proposed score development framework (5.2). The particular application of the ADMM algorithm that we are interested in is the *consensus* problems [16, 26]. These problems began as an exploration of parallel and distributed computing [16] and variants of this problem have been used in many different application settings (stochastic resource allocation [219], parameter estimation in wireless sensor networks [177], quadratically constrained quadratic programming [104], big data optimization [168], and stochastic mixed-integer programming [22], to name a few). The application of the consensus problem to our work is discussed in detail in Section 5.2.3. First, we give some of the background of the general ADMM method.

5.2.2 The alternate direction method of multipliers

Applied optimization has been overwhelmed with the concept of “Big Data". At any time around the globe, large volumes of data are generated by today’s ubiquitous communication, imaging, and mobile devices such as cell phones, surveillance cameras and drones, medical and e-commerce platforms, as well as social networking sites [192]. Incorporating these large data sets into the solution process has made it difficult for operations researchers and engineers to solve application focused optimization models. Those “off-the-shelf" techniques and technologies that we use to analyze these models cannot work efficiently and satisfactorily anymore [41]. Therefore, there has been recent advances in convex optimization methods for data driven optimization and big data problems [38, 192]. The alternating direction method of multipliers (ADMM) is one such method that utilizes *dual decomposition* and *augmented Lagrangian* methods for constrained optimization [26]. The

original idea was first proposed in 1976 by Gabay et al. [75] where they develop a dual method that decouples a nonlinear variational problem. The concept of these dual decent methods were studied throughout the 1980s [15, 16, 169] and many of the theoretical results were established by the 1990s. Boyd et al. [26] gives a review of this method as well as other precursors to the ADMM algorithm. They survey some of the key literature surrounding this algorithm while also discussing a wide variety of applications that it has been used in. Under certain assumptions such as strong convexity, Deng and Yin [57] show the global linear convergence of the ADMM algorithm. For other applications of the ADMM algorithm, the reader is referred to [8, 157, 178, 216].

5.2.3 An ADMM algorithm to solve the score development problem

As discussed in section 5.2.1, we will focus on a specific application of the ADMM algorithm, the *consensus problem*. The main idea of the consensus problem is to separate the original optimization problem, which we will refer to from now on as the *global problem*, into smaller *sub-problems* that are easier to solve (i.e., smaller in size). Each sub-problem consists of a set of *local variables* equivalent to those variables of the global problem and includes a constraint that states all local variables defined within each sub-problem should agree [26]. First, we give the sub-problem parameters and variables in Tables 5.3 and 5.4, respectively. Then we take the first term in objective (5.2a) and rewrite it as sum of sub-problem specific objective terms.

$$\sum_k \sum_s \sum_{i \in \Omega^{\mathrm{T}}} y_{ik} w_{isk} = \sum_{r=1}^{R} f^r \quad \text{where} \quad f^r = \sum_k \sum_s \sum_{i \in \Omega_r^{\mathrm{T}}} y_{ik} w_{isk} \tag{5.3}$$

Table 5.3 ADMM Algorithm Sub-Problem Parameters

R :	# subsets of Ω^{T}
Ω_r^{T} :	r^{th} subset of Ω^{T} for $r = 1 \ldots R$ such that $\bigcup_{r=1}^{R} \Omega_r^{\mathrm{T}} = \Omega^{\mathrm{T}}$ and $\Omega_i^{\mathrm{T}} \cap \Omega_j^{\mathrm{T}} = \emptyset$ for each $i \neq j$

Constraints (5.4b)- (5.4k) represents the constraint set for sub-problem r $(r = 1 \ldots R)$ and we the define set $\Lambda^r \equiv \{(\boldsymbol{\lambda}^r, \boldsymbol{Z}, \boldsymbol{Q}^r, \boldsymbol{w}^r) \mid (\boldsymbol{\lambda}^r, \boldsymbol{Z}, \boldsymbol{Q}^r, \boldsymbol{w}^r)$ satisfy (5.4b) - (5.4k)$\}$ as the feasible region sub-problem r.

$$\boldsymbol{\lambda}^r = \boldsymbol{\zeta}, \quad \boldsymbol{Q}^r = \hat{\boldsymbol{Q}} \tag{5.4a}$$

$$w_{i,s,k} \leq z_{i,s} \quad \forall i \in \Omega_r^T, \forall s, k \tag{5.4b}$$

Table 5.4 ADMM Algorithm Sub-Problem Variables

$\boldsymbol{\lambda}^r$:	solution vector of point values when training done using observations from set Ω_r^{T}
q_{sk}^r :	probability that an observation in Ω_r^{T} with score s is in class k for $k = 1, \ldots, K$
	Let $\boldsymbol{Q}^r$ be the matrix representation of these variables
$\boldsymbol{\zeta}$:	Global consensus variable for integer point values
$\hat{q}_{sk}$:	Global consensus variable representing probability that an observation in Ω^{T} with score s is in class k for $k = 1, \ldots, K$
	Let $\hat{\boldsymbol{Q}}$ be the matrix representation of these variables

$$w_{i,s,k} \leq q_{s,k}^r \quad \forall i \in \Omega_r^T,\ \forall s, k \tag{5.4c}$$

$$w_{i,s,k} \geq q_{s,k}^r + z_{i,s} - 1 \quad \forall i \in \Omega_r^T,\ \forall s, k \tag{5.4d}$$

$$\sum_{i \in \Omega_r^{\mathrm{T}}} w_{i,s,k} \leq \sum_{i \in \Omega_r^{\mathrm{T}}} z_{i,s} y_{i,k} \quad \forall s, k \tag{5.4e}$$

$$l_s z_{is} \leq (\boldsymbol{\lambda}^r)' \boldsymbol{x}_i \leq u_s z_{is} + nP(1 - z_{is}) \qquad \forall i \in \Omega_r^{\mathrm{T}},\ \forall s \tag{5.4f}$$

$$\sum_k q_{sk}^r = 1 \qquad \forall s \tag{5.4g}$$

$$\sum_s z_{is} = 1 \quad \forall i \in \Omega_r^{\mathrm{T}} \tag{5.4h}$$

$$q_{sk}^r \geq \mathbf{0} \qquad \forall s, k \tag{5.4i}$$

$$\boldsymbol{\lambda}^r \in \{0, 1, \ldots P\} \tag{5.4j}$$

$$z_{is} \in \{0, 1\} \qquad \forall i \in \Omega_r^{\mathrm{T}},\ \forall s \tag{5.4k}$$

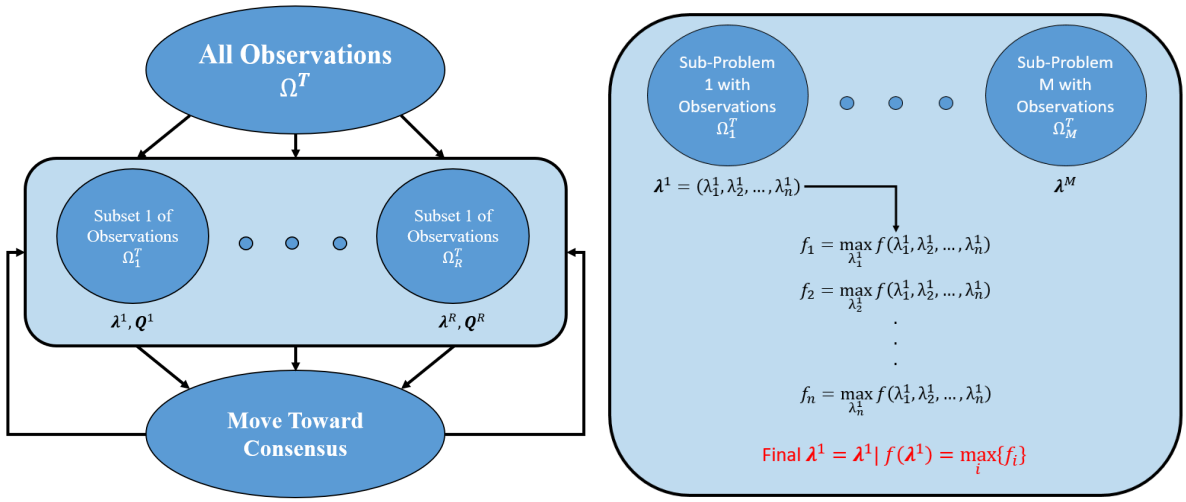
Constraint (5.4a) ensures that the local variables to each sub-problem agree. This allows us to rewrite problem (5.2) equivalently as model (5.5).

$$\max_{\{(\boldsymbol{\lambda}^r, \boldsymbol{Z}^r, \boldsymbol{Q}^r, \boldsymbol{w}^r) \in \Lambda^r\}_{r=1}^R} \left\{ \sum_{r=1}^R f^r - \gamma \sum_{j=1}^N \zeta_j^2 \ \text{ s.t. } \boldsymbol{\lambda}^r = \boldsymbol{\zeta} \ \text{ and } \ \boldsymbol{Q}^r = \hat{\boldsymbol{Q}}, \ \forall r \right\} \tag{5.5}$$

We can derive the augmented Lagrangian function [26] for problem (5.5) as

$$L_\rho\left(\boldsymbol{\lambda}^r, \boldsymbol{\alpha}^r, \boldsymbol{\phi}^r, \boldsymbol{\zeta}, \hat{\boldsymbol{Q}}\right) = \sum_{r=1}^R \left(f^r - \left\langle [\boldsymbol{\alpha}^r, \boldsymbol{\phi}^r], [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}] \right\rangle - \left(\rho/_2\right) \left|\left| [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}] \right|\right|_2^2 \right) \tag{5.6}$$

where $\rho > 0$ is a *penalty parameter*, $\boldsymbol{\alpha}^r$ and $\boldsymbol{\phi}^r$ are the Lagrangian multipliers for sub-problem r for constraints $\boldsymbol{\lambda}^r = \boldsymbol{\zeta}$ and $\boldsymbol{Q}^r = \hat{\boldsymbol{Q}}$, respectively. The symbol "$\langle \cdot, \cdot \rangle$" represents the inner product of two



(a) ADMM Algorithm for the Consensus Problem **(b)** `coordinateDecent` Procedure

Figure 5.1 High-level process for the (a) ADMM consensus problem algorithm and (b) `coordinateDecent` procedure for polishing sub-problem solutions

vectors. To simplify notation, we let $\boldsymbol{Q}^r - \hat{\boldsymbol{Q}}$ be a vector in $\mathbb{R}^{SK}$ defined as follows

$$\boldsymbol{Q}^r - \hat{\boldsymbol{Q}} = [q_{11}^r - \hat{q}_{11}, q_{12}^r - \hat{q}_{12}, \ldots, q_{1K}^r - \hat{q}_{1K}, \ldots, q_{S1}^r - \hat{q}_{S1}, q_{S2}^r - \hat{q}_{S2}, \ldots, q_{SK}^r - \hat{q}_{SK}]^{\mathrm{T}} \tag{5.7}$$

Letting ν be the iteration counter, we give the ADMM algorithm (5.8) for our consensus problem (5.5). Figure 5.1(a) displays a high-level visual representation of this ADMM algorithm.

$$\left(\boldsymbol{\lambda}_{\nu+1}^r, \boldsymbol{Q}_{\nu+1}^r\right) \equiv \underset{\boldsymbol{\Lambda}^r}{\operatorname{argmax}} \left\{ f^r - \left\langle [\boldsymbol{\alpha}_\nu^r, \boldsymbol{\phi}_\nu^r], [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}_\nu, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}_\nu] \right\rangle - \left(\rho/2\right) \left\| [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}_\nu, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}_\nu] \right\|_2^2 \right\} \tag{5.8a}$$

$$\boldsymbol{\zeta}_{\nu+1} \equiv \underset{\boldsymbol{\zeta} \in \{0,\ldots,P\}^N}{\operatorname{argmin}} \left\{ \gamma \sum_{j=1}^{N} \zeta_j^2 - \sum_{r=1}^{R} \left(\langle \boldsymbol{\alpha}_\nu^r, \boldsymbol{\zeta} \rangle - \left(\rho/2\right) \|\boldsymbol{\lambda}_{\nu+1}^r - \boldsymbol{\zeta}\|_2^2 \right) \right\} \tag{5.8b}$$

$$\hat{\boldsymbol{Q}}_{\nu+1} \equiv \underset{\hat{\boldsymbol{Q}} \in [0,1]^{SK}}{\operatorname{argmin}} \left\{ \sum_{r=1}^{R} \|\boldsymbol{Q}_{\nu+1}^r - \hat{\boldsymbol{Q}}\|_2^2 \right\} \tag{5.8c}$$

$$\boldsymbol{\alpha}_{\nu+1}^r \equiv \boldsymbol{\alpha}_\nu^r + \rho \left(\boldsymbol{\lambda}_{\nu+1}^r - \boldsymbol{\zeta}_{\nu+1} \right) \tag{5.8d}$$

$$\boldsymbol{\phi}_{\nu+1}^r \equiv \boldsymbol{\phi}_\nu^r + \rho \left(\boldsymbol{Q}_{\nu+1}^r - \hat{\boldsymbol{Q}}_{\nu+1} \right) \tag{5.8e}$$

When deciding upon the size of the sub-problems (i.e., the size of the sets Ω_r^{T}), there is a trade-off that needs to be considered between how difficult it is to solve problem (5.8a) and how many iterations it will take to find a consensus. The smaller the subsets Ω_r^{T}, the quicker it is to solve problem (5.8a).

However, this would create more sub-problems making it more difficult to obtain a consensus (i.e. it would take more iterations to terminate). We also note that the sub-problems (5.8a) are quadratic integer programming problems with both binary and pure integer variables. Therefore, the subsets defining these sub-problems will need to be relatively small in order to solve them with a commercial software. In an effort to avoid many sub-problems, we have decided to relax problem (5.8) by allowing the binary variables z_{is} to be continuous in the interval $[0,1]$ allowing the size of the subsets Ω_r^{T} to be much larger. Given an integer solution to problem (5.8), we can find a corresponding $(\boldsymbol{Z},\boldsymbol{Q})$ that is feasible for the global problem (5.2).

For any given sub-problem solution $\boldsymbol{\lambda}^r$, we can evaluate the performance of this solution on the entire training set to get the global problem objective value. First, we calculate the score for all observations in the training set Ω^{T} (i.e. we can calculate z'_{is} for each $i \in \Omega^{\mathrm{T}}$ and $s = 1 \dots S$). Then, we solve the following linear programming problem:

$$\max_{\boldsymbol{Q}} \sum_k \sum_s \sum_{i \in \Omega^{\mathrm{T}}} y_{ik} z'_{is} q_{sk} \tag{5.9a}$$

$$\text{s.t. } 0 \leq \sum_i z'_{is}(y_{ik} - q_{sk}) \quad \forall s, k \tag{5.9b}$$

$$\sum_k q_{sk} = 1 \qquad \forall s \tag{5.9c}$$

$$\boldsymbol{q}_s \geq \boldsymbol{0} \qquad \forall s \tag{5.9d}$$

The objective function value of problem (5.9) minus $\gamma \sum_j (\lambda_j^r)^2$ will give you the objective function value of problem (5.1) for the solution $\boldsymbol{\lambda}^r$. This process of finding an optimal probability matrix $\boldsymbol{Q}$ given a solution $\boldsymbol{\lambda}$ we call `qSolve` and we present the details of this process in Algorithm 2 given in Appendix C.

In an effort to find stronger integer solutions using the solution found by solving (5.8a), we implement a coordinate decent algorithm similar to what has been done by Ustun and Rudin [208]. For a given dimension of $\boldsymbol{\lambda}$, the algorithm fixes the other dimensions and attempts to find a feasible integer value along that dimension that improves the objective. This is repeated across all dimensions and the strongest updated solution found, if any, is returned. We call this procedure `coordinateDecent` and give it's details in Algorithm 4 in Appendix C. Figure 5.1(b) gives the visual representation for the `coordinateDecent` procedure. We define three different termination criteria for our ADMM algorithm: (1) *Finding a consensus*, (2) *No change in the target consensus*, or (3) *Maximum number of iteration reached*. The details of these termination criteria can be seen in Algorithm 1 in Appendix C. Table 5.5 displays the high-level procedure of our proposed tailored ADMM algorithm used to solve the score development problem (5.2) and Figure 5.2 gives the corresponding visual representation.

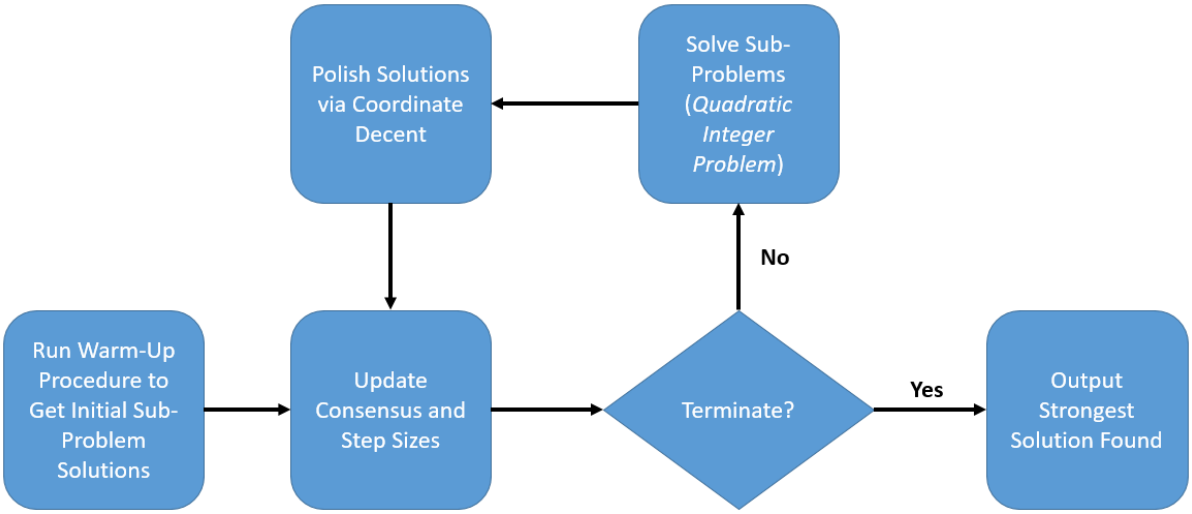


Figure 5.2 Tailored ADMM algorithm with coordinate decent to solve the score development problem (5.1)

We present the full detailed ADMM with coordinate decent procedure in Algorithm 1 in Appendix C.

5.3 Results

5.3.1 Experimental design

For the observation set Ω that would be used to develop our illness scores via model (5.1), we utilized the same observations that were generated for Chapter 3 of this dissertation. Recall that we had two main outcomes of interest (Figure 3.1): (1) First time to ICU transfer and (2) In-hospital mortality. For each of these outcomes, we identified case and control populations (Figure 3.2) from which we generate observation vectors (Figure 3.3). Unlike Chapter 3, we create binary observation vectors representing each of the variables within the categories of the PIRO score as described in Table 3.1. For example, if a patient was 67 years old, the value for variable one in the observations vectors used in Chapter 3 was 67. However, in the following experiments, we create three binary variables representing which age category a patient lies in. Therefore, for a 67 year old patient, the first three variables would be "$[0,1,0]$" representing that the patient lies in the age category 65–80. We used normal range imputation as was done in Section 3.2.5 of Chapter 3. We did not include the missing indicators to identify when values were imputed.

As we have described in Section 5.1, there have been many scores developed to track the acuity of those patients susceptible to sepsis [33, 102, 112, 142, 165, 190, 213, 214]. We also have discussed

Table 5.5 Alternate Direction Method of Multipliers with Coordinate Decent Algorithm

Step 0:	Given training set Ω^{T} and subset size M, partition Ω^{T} into subsets $\left\{\Omega^{\mathrm{T}}_r\right\}_{r=1}^{R}$ Create mutually exclusive score intervals $\{[l_s, u_s]\}_{s=1}^{S}$ of length L Set iteration counter $v \leftarrow 0$ and maximum number of iterations `maxIt`
Step 1:	Generate initial solutions $\left\{(\boldsymbol{\lambda}^r_0, \boldsymbol{Q}^r_0\right\}_{r=1}^{R}$ using `warmStart` (Appendix C) Initialize step sizes $\{\boldsymbol{\alpha}^r_0\}_{r=1}^{R}$ and $\{\boldsymbol{\phi}^r_0\}_{r=1}^{R}$ and consensus variables $\boldsymbol{\zeta}_0$ and $\hat{\boldsymbol{Q}}_0$
Step 2:	If $v =$ `maxIt` **OR** *consensus found* **OR** *no change in consensus variables*, **TERMINATE**. Otherwise, go to Step 3.
Step 3:	Solve each sub-problem to obtain sequence of solutions $\left\{\boldsymbol{\lambda}^r_{\mathrm{tmp}}\right\}_{r=1}^{R}$ Apply `coordinateDecent` (Appendix C) procedure on $\left\{\boldsymbol{\lambda}^r_{\mathrm{tmp}}\right\}_{r=1}^{R}$ to get $\left\{\boldsymbol{\lambda}^r_{v+1}\right\}_{r=1}^{R}$
Step 4	Use solutions $\left\{\boldsymbol{\lambda}^r_{v+1}\right\}_{r=1}^{R}$ to obtain optimal $\left\{\boldsymbol{Q}^r_{v+1}\right\}_{r=1}^{R}$ via `qSolve` procedure Evaluate objective (5.1a) for each solution updating best solution if appropriate
Step 5:	Update step sizes and consensus variables $\{\boldsymbol{\alpha}^r_{v+1}\}_{r=1}^{R}$, $\{\boldsymbol{\phi}^r_{v+1}\}_{r=1}^{R}$, $\boldsymbol{\zeta}_{v+1}$ and $\hat{\boldsymbol{Q}}_{v+1}$ $v \leftarrow v+1$ and go to Step 2.

another optimization based approach to the development of risk scores [208]. We compare the performance of our score in predicting the outcomes of interest against scores and frameworks in the current literature. Namely, we develop a risk score utilizing the model `riskSlim` developed by Ustun and Rudin [208]. They provide python code for implementation of their solution methodology and the parameters used in their package are given in Appendix C. As we are using the same variables and cut-off values provided by Howell et al. [102], we also compare our developed score to their already published PIRO score given in Table 3.1. The performance metric that we use to compare is the *average probability of correct classification*.

An ideal severity of illness scoring system would not only provide a score but also an associated probability that an adverse outcome (e.g., mortality) will occur in the near future (risk). Using retrospective data, we have ground truth on the outcome related to each observation (visit). Using this truth, we calculate the *average probability of correct classification* for a specific *method*, denoted by $\mathrm{P}_{\mathrm{method}}$, as the average probability that an observation is classified as having the correct outcome.

Average probability of correct classification for ADMM algorithm score

One of the variables of the solution is the probability that a particular visit will end up with a certain outcome (recall that the variable q_{sk} represents the probability that a patient will be in class k, given that they are in score interval s). Denoting our solution method as the "ADMM" method, we can calculate $\mathrm{P}_{\mathrm{ADMM}}$ as

$$\mathrm{P}_{\mathrm{ADMM}} = \frac{\sum_i q_{sk} \cdot \mathbf{1}\{\text{Observation } i \text{ in score interval } s \text{ and class } k\}}{\text{Total \# of Observations}} \tag{5.10}$$

Alongside the score created by solving our model (5.2) via our ADMM algorithm, we also utilized Cplex version 12.8 to solve this model. We specified twice the termination time of the ADMM algorithm as the stopping criteria for Cplex. The corresponding average probability of correct classification, denoted as $\mathbb{P}_{\mathrm{Cplex}}$, was calculated using equation (5.10).

Average probability of correct classification for `riskSlim` ***score***

The output of the framework `riskSlim` is an intercept/point vector pair, $(\beta, \boldsymbol{\lambda})$. Given an observation vector of a patient at any given point in time $\boldsymbol{x}$, the probability that the outcome of interest will occur within the specified time window (five hours in our case) will be given as $P(\text{Outcome}) = \frac{1}{1+\exp(\beta - \boldsymbol{\lambda}'\boldsymbol{x})}$. We calculate the `riskSlim` average probability of correct classification $\mathbb{P}_{\texttt{riskSlim}}$ as

$$\mathrm{P}_{\texttt{riskSlim}} = \frac{\sum_i P(\text{Outcome}) \cdot \mathbf{1}\{\text{Outcome}\} + (1 - P(\text{Outcome})) \cdot \mathbf{1}\{\text{No outcome}\}}{\text{Total \# of Observations}} \tag{5.11}$$

Average probability of correct classification for scored developed by Howell et al. [102]

For the already developed score by Howell et al. [102] (Table 3.1), there was no associated risk provided by their model. Recall, however, that we can find an optimal probability matrix $\boldsymbol{Q}$ given a solution $\boldsymbol{\lambda}$ using the `qSolve` procedure (Algorithm 2 in Appendix C). We implement this procedure using the point values $\boldsymbol{\lambda}$ provided by Howell et al. [102] and use equation (5.10) to calculate $\mathrm{P}_{\mathrm{Howell}}$.

5.3.2 Score results

For the set of mortality outcome observations (see Section 3.2.2), we partition the 10,031 observations (Ω) into 6,687 observations that would be used for training the score (Ω^{T}) and 3,344 that would be used for validation (Ω^{V}). The training and validation set are partitioned in such a way so as to keep the original ratio of case to control the same each set (a 1 to 6 ratio was for case to control. See Section 3.2.2). Figure 5.3 displays the *probability of correct classification* across the methods for both the training and validation sets. The values for the interval length (length of $[\boldsymbol{l}_s, \boldsymbol{u}_s]$), regularization parameter γ, and penalty parameter ρ that were used to produce these results were 1, $\frac{1}{100 \times \text{length of the interval}} = \frac{1}{100}$ and 0.5, respectively. Tables 5.6- 5.9 display the point values and calculated risks for each of the developed scores.

Table 5.6 Score obtained by solving `riskSlim` model ($R_{\max} = 5$)

Nursing Home Resident	1 point
History of Malignancy w/out Metastasis	1 point
Bands > 5%	2 point
BUN > 20	2 points
Lactate > 4	3 points

SCORE	0	1	2	3	4	5	6	≥ 7
RISK	1.8%	4.7%	11.9%	26.9%	50.0%	73.1%	88.1%	> 95%

Table 5.7 PIRO score developed by Howell et. al. 2011

Age < 65	0 points
Age $\in [65, 80]$	1 point
Age > 80	2 points
History of COPD	1 point
History of Liver Disease	2 points
Nursing Home Resident	2 points
History of Malignancy w/out Metastasis	1 point
History of Malignancy w Metastasis	2 points
Pneumonia	4 points
Skin/Soft Tissue Infection	0 points
Any Other Infection	2 points
Respiratory Rate > 20	3 points
Bands > 5%	1 point
Heart Rate > 120	2 points
BUN > 20	2 points
Respiratory Failure/Hypoxemia > 120	3 points
Lactate > 4	3 points
Systolic Blood Pressure < 70	4 points
Systolic Blood Pressure $\in [70, 90]$	2 points
Systolic Blood Pressure > 90	0 points
Platelet Count > $150K$	2 points

SCORE	[0,3]	[4,7]	[8,11]	[12,15]	[16,19]	[20,23]	[24,27]
RISK*	3.4%	8.1%	17.6%	33.0%	40.7%	53.8%	> 50.0%

**Risk calculated by taking the point values and calculating a $\boldsymbol{Z}'$ to solve problem* (5.9)

Table 5.8 Score obtained by solving model (5.2) using ADMM with sub-problem set size of 200

Nursing Home Resident	2 points
History of Malignancy w/out Metastasis	2 points
Respiratory Rate > 20	1 point
Bands > 5%	2 points
BUN > 20	3 points
Lactate > 4	4 points
Systolic Blood Pressure < 70	3 points
Systolic Blood Pressure $\in [70, 90]$	2 points

SCORE	[0,1]	[2,3]	[4,5]	[6,7]	[8,9]	[10,11]	≥ 12
RISK*	1.8%	3.2%	13.6%	31.7%	62.6%	83.0%	>90%

**Parameters used in solving* (5.2) *to obtain risk of death:* $\gamma = {}^{1}/_{100}$ & $\rho = 0.5$

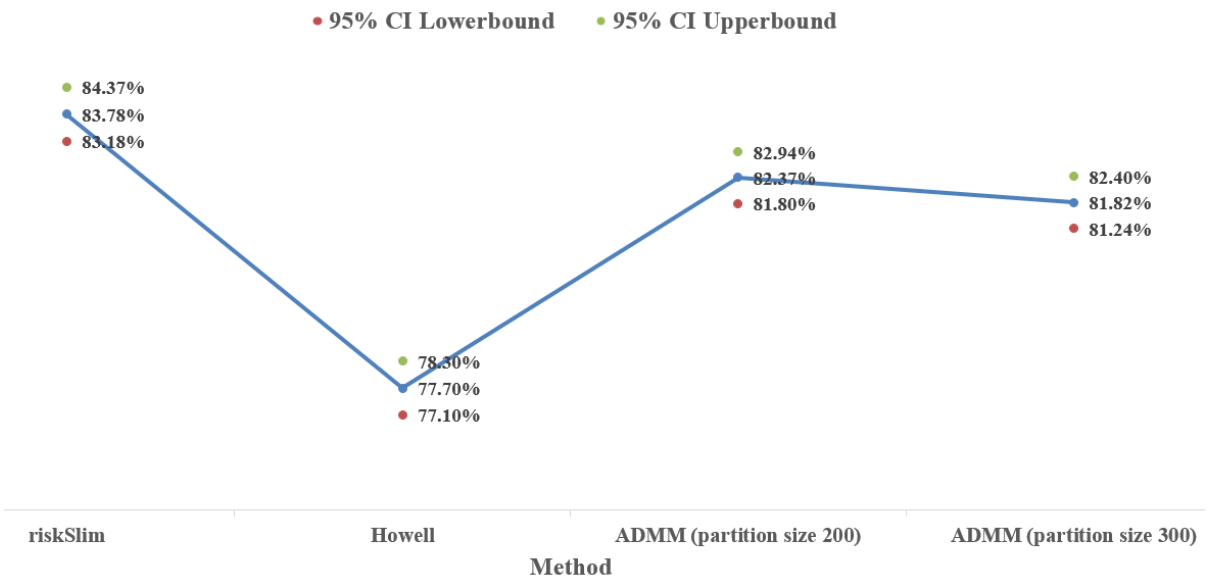
Table 5.9 Score obtained by solving model (5.2) using ADMM with sub-problem set size of 300

Age < 65	1 point
Age $\in [65, 80]$	2 points
Age > 80	4 points
History of COPD	1 point
History of Liver Disease	1 point
Nursing Home Resident	1 point
History of Malignancy w/out Metastasis	1 point
Hypoximea	1 point
Bands > 5%	3 points
Heart Rate > 120	1 points
BUN > 20	4 points
Lactate > 4	5 points
Systolic Blood Pressure < 70	3 points
Systolic Blood Pressure $\in [70, 90]$	2 points

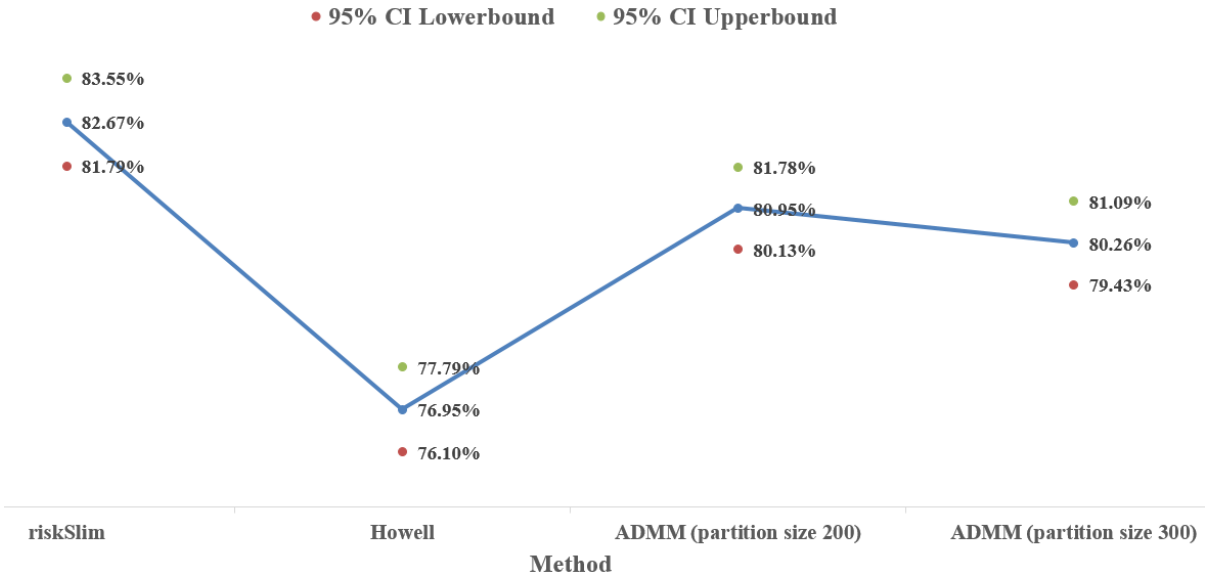
SCORE	[0,1]	[2,3]	[4,5]	[6,7]	[8,9]	[10,11]	[12,13]	[14,15]	[16,17]	≥ 18
RISK*	0.0%	1.4%	4.3%	10.3%	18.2%	30.5%	51.7%	76.8%	81.3%	>90%

**Parameters used in solving* (5.2) *to obtain risk of death:* $\gamma = {}^{1}/_{100}$ & $\rho = 0.5$

The parameters γ and ρ used in our ADMM algorithm (5.8) were defined through preliminary experiments. We run a sensitivity analysis on these parameters. We allowed γ and ρ to take on the values from the sets $\{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ and $\{0.1, 0.5, 1, 5, 10\}$, respectively and look at every combination of these possible values. Figures 5.4 and 5.5 display the results of the sensitivity analysis when γ is fixed at ${}^{1}/_{100}$ and when ρ is fixed at 0.5, respectively. The remaining results from the sensitivity analysis are given in Appendix C (Figures C.1- C.7).

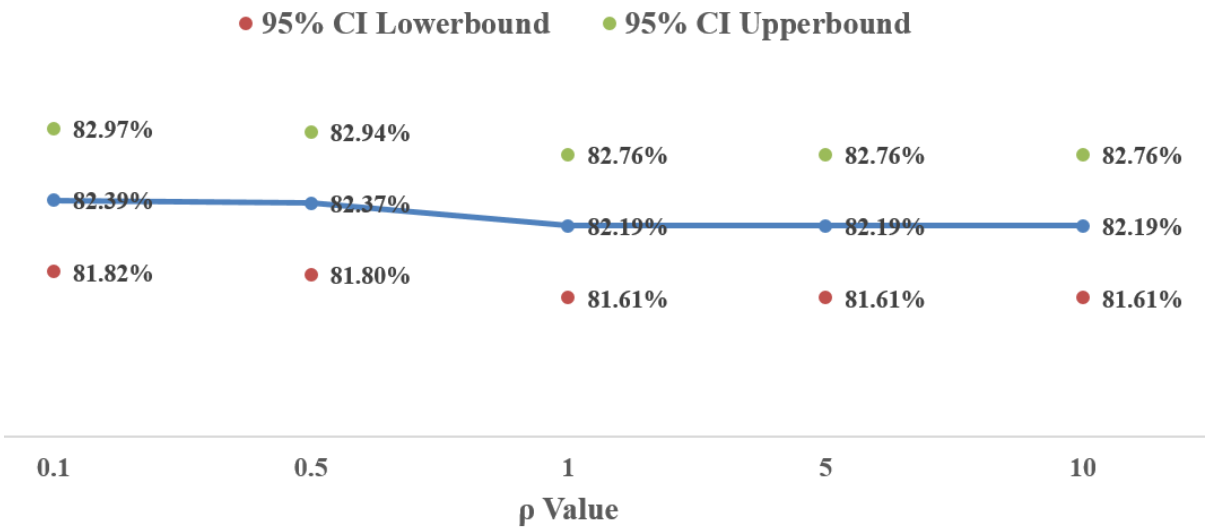


(a) Training set

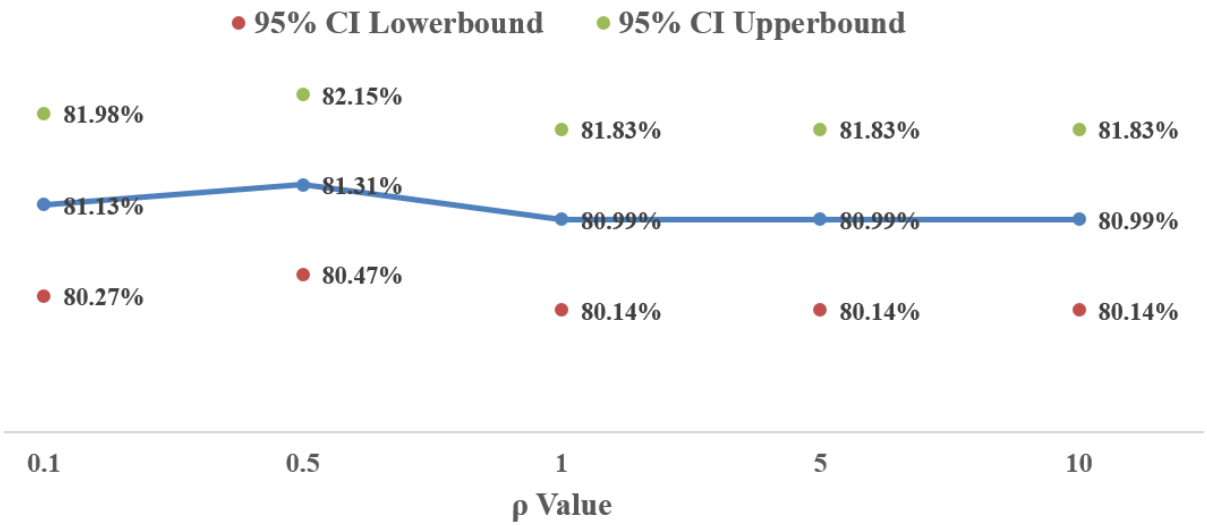


(b) Validation set

Figure 5.3 Average probability of correct classification for the mortality outcome in the (a) training set and (b) validation set

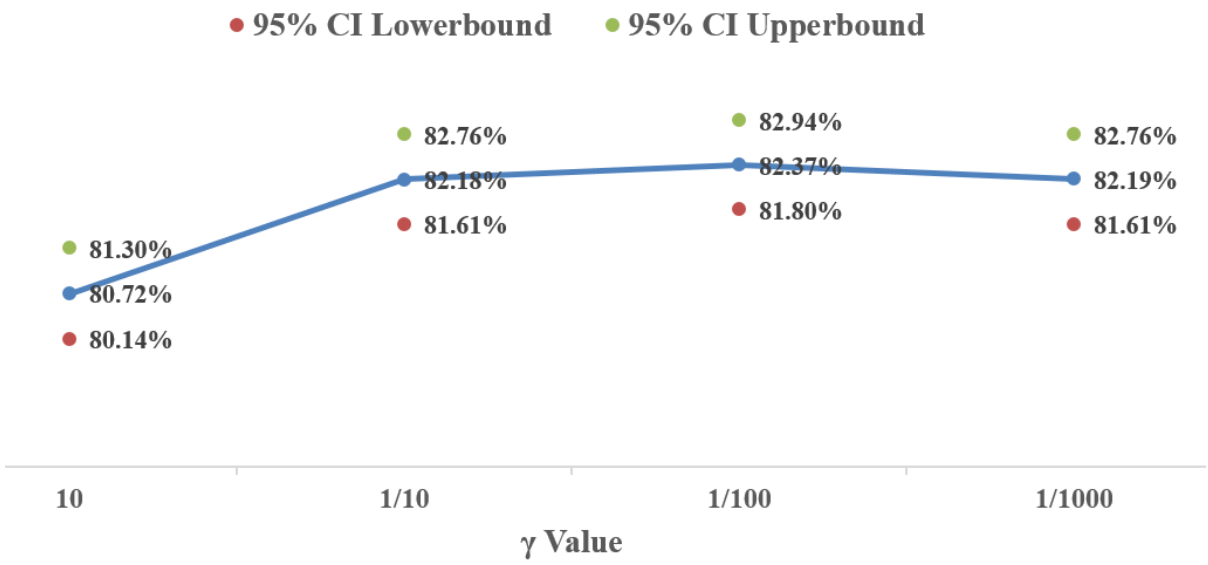


(a) Training set

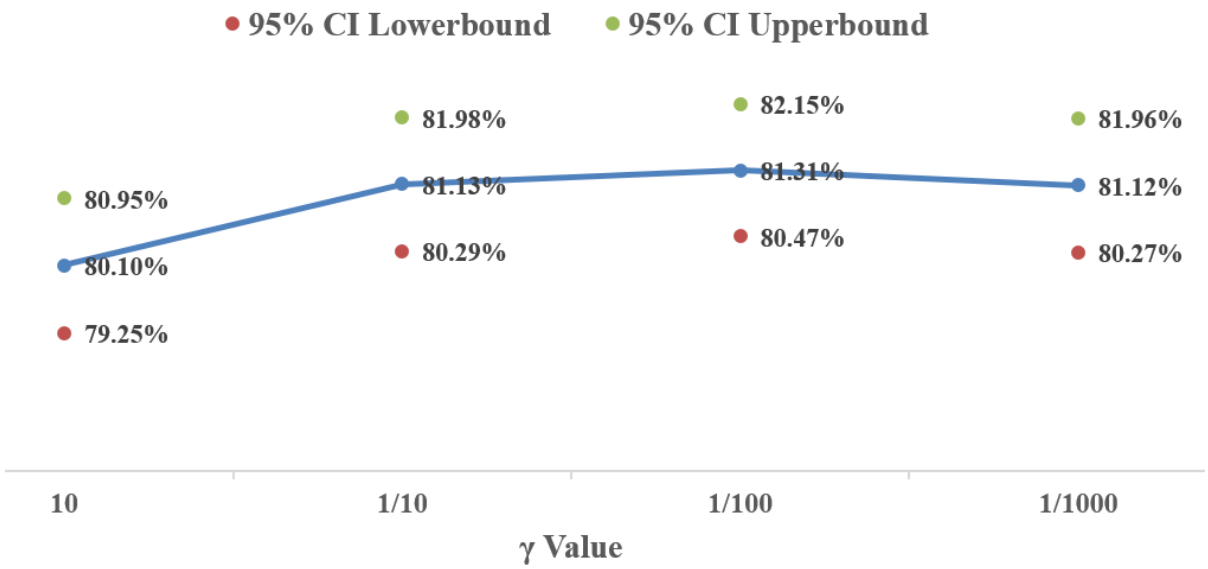


(b) Validation set

Figure 5.4 Sensitivity of probability of correct classification when $\gamma = {}^{1}/_{100}$ and $\rho \in \{0.1, 0.5, 1, 5, 10\}$ in the (a) training set and (b) validation set



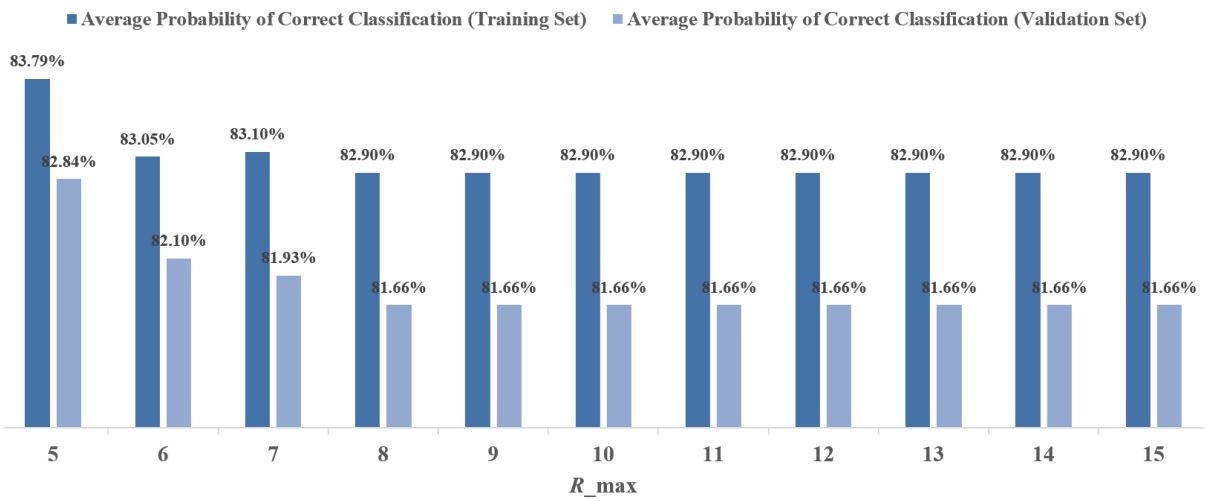
(a) Training set



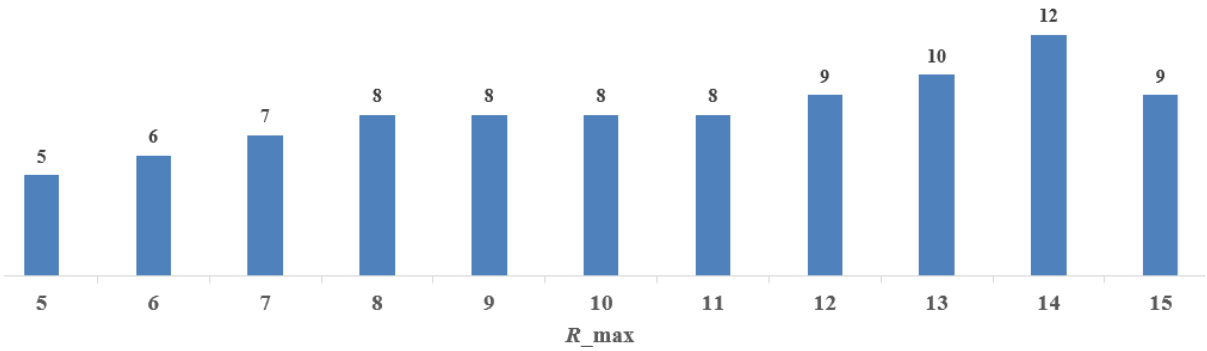
(b) Validation set

Figure 5.5 Sensitivity of average probability of correct classification when $\rho = 0.5$ and $\gamma \in \{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ in the (a) training set and (b) validation set

For the `riskSlim` method, there is an arbitrary parameter that was specified, $R_{\max}$. This parameter specifies the number of features that are allowed to be in the final score ($R_{\max} = 5$ is used for the output of the results seen in Figure 5.3). Figure 5.6 displays how the metric $\mathrm{P}_{\texttt{riskSlim}}$ changes alongside how many features are selected for the score as $R_{\max}$ is varied from 5 to 15.



(a) Average probability of correct classification



(b) Number of features selected

Figure 5.6 Sensitivity of (a) $\mathrm{P}_{\texttt{riskSlim}}$ and (b) the number of features selected as a function of the `riskSlim` parameter $R_{\max}$

5.4 Discussion

In this chapter, we have provided a new mixed-integer optimization framework for the development of *interpretable* severity of illness scoring systems. We have applied this framework to develop a

score that can be used to track acuity of patients in the sepsis trajectory. The main contributions of this framework are listed below:

1. *Ability to incorporate multiple classes/stages*
   Traditionally, severity of illness scores have been developed using prediction models that are focused on binary classification [102, 122, 184, 213]. Even in the optimization based risk score framework `riskSlim` [208], the derived risks are developed in a binary fashion. Since many diseases, such as sepsis, progress through stages, a strength of our proposed framework is that it has the ability to generate risks for multiple classes/stages of a disease.

2. *New derivation of risk from scores outside of logit/logistic regression function*
   As previously stated, many severity of illness scoring systems utilize logit/logistic regressions as a way to come up with weights associated with each of the predictor variables [17, 35, 43, 68, 102, 122, 147, 184]. We incorporate these risks directly into the model through the use of decision variables.

3. *Ability to incorporate clinical context in the form of constraints into the model*
   Clinical context and knowledge is extremely important in the deciphering of a patient's health status. Although retrospective EHR data has been a powerful tool in the analysis of health systems, it's much more challenging to utilize this data in the personalized medicine realm due to the difficulty of incorporating clinical context into our models. A mathematical programming framework is attractive because these clinical considerations can be taken into account via constraints of the model.

4. *New application area for the ADMM algorithm*
   As discussed in Section 5.2.2, the ADMM algorithm has been used and validated in many application areas. To our best knowledge, this is the first work that has used this methodology in the development of severity of illness scoring systems.

5.4.1 Score and performance comparison insights

Looking at the average probability of correct classification results displayed in Figure 5.3, we see that the score developed by the proposed model solved via the ADMM method does outperform the original PIRO score developed by Howell et al. [102]. This figure also indicates that the score developed by `riskSlim` performs only slightly better than that of the score that is constructed by our proposed ADMM algorithm by a small margin. However, we believe that since our ADMM algorithm is not terminating at the optimal solution whereas the `riskSlim` solution is, that our solution has the potential to exceed the performance of `riskSlim`. These results demonstrate that

the use of the proposed model to produce a severity of illness score can achieve a high probability of correctly placing patients into their appropriate class.

As stated in Section 5.3.2, the parameters γ and ρ in problem (5.8a) are arbitrarily defined so we run a sensitivity analysis the average probability of correct classification when we vary these parameters. The results of this analysis seen in Figure 5.4 indicate that the average probability of correct classification is not sensitive to the parameter ρ (penalty parameter for deviation from consensus). However, Figure 5.5 indicates that performance is slightly more impacted by the change of γ (sparsity parameter). This is explained by the fact that the parameter γ limits the magnitude and number of features that are allowed to be selected, whereas the parameter ρ does not have any effect on the magnitude of the integer point values. Our conclusions are that the sparsity parameter γ would need to be selected carefully in this model and future research could potentially explore the optimal selectiion of this parameter.

5.4.2 Limitations

One limitation of our results is that the features used in all models were restricted to those identified by Howell et al. [102]. Although these features have been identified for prediction in sepsis specifically, we believe that more features should be incorporated into the modeling process. However, because the number of general integer variables λ's increase linearly within the number of features, this implies that the problem complexity grows exponentially with the introduction of more features. Secondly, we do not provide any convergence guarantees for the ADMM algorithm. As stated in Section 5.2.2, convexity is a requirement for linear convergence of the ADMM algorithm [57]. Future work will refine the proposed ADMM algorithm to obtain convergence guarantees in non-convex cases, specifically for the propsed score development problem (5.2).

CHAPTER

6

CONCLUSIONS AND FUTURE WORK

6.1 Dissertation summary and contributions

We have transitioned to a world where many aspects of life are driven by the analysis of data. Electronic Health Record (EHR) systems have provided operations researchers, computer scientists, mathematicians, and engineers large amounts of data for research related to improving the efficiency and effectiveness of health systems. Learning from these data sets can drive significant science and engineering advances along with improvements in quality of life. However, big data comes with new challenges. Slavakis et al. [192] discuss many of these challenges such as storage, corrupted and inaccurate data, and online processing to keep up with the continuously generated data, to name a few. This dissertation presents frameworks and methodologies to address (a) the problem of feature selection and (b) the use of data in the development of interpretable scoring systems to assist decision makers in health care. This work contributes to the fields of Operations Research in health care and Machine Learning. We demonstrate how optimization models, specifically bilevel and mixed integer programming models, can be used to develop data driven support tools. While most of this work is applied in a health care setting, we believe that the frameworks and models presented in this dissertation are applicable in many other domains that utilize large data sets to learn features, build prediction models, and develop support tools for decision makers.

In Chapter 2, a framework for the construction of a score that can accurately represent workload

amongst teams of health care providers is proposed. This was accomplished collaboratively with the Hospital Internal Medicine (HIM) Department at Mayo Clinic in Rochester, MN and with another PhD student in the North Carolina State University College of Design, Kendall McKenzie. Together, we applied this framework to develop a score to assist in the triaging of patients into the HIM department at Mayo Clinic, Rochester, MN in an effort to balance workload amongst their provider teams. The HIM department executed the Delphi survey method to isolate factors contributing to the score and provided these results upon our arrival in May 2016. Kendall McKenzie designed the secondary survey through choice-based conjoint analysis that would provide inputs to the optimization model. The main contributions being claimed in this dissertation is the proposed optimization model for score development of Section 2.3 and the developed simulation model of Section 2.4. The results indicate a significant decrease in the variation between team workloads when our workload score is used for patient assignment decisions. Furthermore, the provider teams staffed by medical residents showed a reduction in time spent being under-utilized, which is an improvement that Mayo Clinic HIM management was hoping to achieve. This confirms that the proposed workload score has the potential to balance workload more equitably across provider teams. Not only did we achieve a more equitable workload among HIM care teams, but we were also able to provide the Mayo Clinic operations team with a workload score calculation that more accurately represents employees' perceptions of their own workloads. The score can be implemented by pulling only ten numbers from the hospital data systems. In the future, we hope to implement this score within the Mayo Clinic and test its performance.

In Chapter 3, we investigate the hypothesis that using information about which clinical variables are missing along with appropriate imputation improves the performance of prediction models for critical patient outcomes. To achieve this objective, we quantify the impact of missing and imputed variables on the performance of various prediction models used in the development of sepsis-related severity of illness scoring systems with the ultimate goal of incorporating this information into scoring systems used in real-time clinical practice. We quantify this value of knowing which information is missing based on prediction performance in models that use all variables as predictors compared to those that utilize summary variables as predictors. We consider five different machine learning models including logistic regression, random forests, step-wise regression, support vector machines, and the least absolute shrinkage and selection operator (LASSO) methods. Our results show that models that use information about which clinical variables are missing can perform better than models that do not take that information into account because there is important clinical information in the fact that certain variables are missing. When developing EHR-based prediction models, developers should consider incorporating indicators for missing variables.

In Chapter 4, we propose a bilevel programming approach to feature selection for classification models. As discussed in chapter 1, the two main approaches to feature selection are the *filter* and

wrapper methods. We focus on the wrapper approach in this chapter. A typical feature selection method with the wrapper approach would define a grid over candidate features, and then perform cross validation for each grid point [154]. This method, however, would suffer from combinatorial explosion of grid points in high dimensions. Problems with many features arise frequently in real life applications [19, 91]. For those problems, greedy strategies such as stepwise regression, backward elimination, filter methods, or genetic algorithms are used [103, 189]. The proposed bilevel approach improves upon the current methods by considering the subset held for validation in the model selection procedure. We also explicitly control the number of model features selected in the upper-level using binary variables in combination with a *feature importance parameter* that allows for the lower level optimization model to select the features that are most important to the classification. Finally, we develop a genetic algorithm solution approach and compare the performance to a derivative-free optimization method. We implement the proposed bilevel feature selection approach in three different case studies where we classify influenza strains based on antigenic variety [136], distinguish between good and bad quality colposcopy images [69], and identify splice junction sites in genetic sequences [150]. Our results indicate that the proposed bilevel framework can be used to achieve similar, if not stronger, classification performance using fewer model features.

In Chapter 5, we propose a mixed integer programming framework for the development of severity of illness scoring systems. We utilize this framework for the building of a score that can be used to track the acuity of patients who are susceptible to sepsis. These risk scores are highly favored in many applications due to the ease of implementation (i.e. decision makers need only to add a few numbers together to gauge the state of a patient or system). Our results validate the use of these scores by comparing it to the predictive ability of other scores in the literature. This framework offers a way to come up with *interpretable* scores by optimizing over integer point values while also allowing the incorporation of multiple class/stages of a disease to be factored into the modeling of the score. This also introduces a new application for the Alternate Direction Method of Multiplier (ADMM) method through the learning of these scores. Finally, we explore a new way to incorporate risk into development of these scores outside of the traditional logistic regression/logit function methodology that is currently used in the literature.

We summarize the the contributions of this dissertation below.

Contributions of this Dissertation

1. Proposal of optimization and simulation models to develop a score that represents workload so that decision makers can make well informed decisions without having to look through large sets of data
2. The quantification of the knowledge that information is missing in the development of severity of illness scores for real-time use in clinical practice and how to incorporate this knowledge in severity of illness scores
3. Proposal and validation of bilevel optimization for feature selection in prediction models
4. Development of a new mixed integer-programming framework for the construction of interpretable severity of illness scoring systems

6.2 Future work

6.2.1 Refinement of optimization methods for score development

We plan on continuing to refine the mixed integer model proposed in Chapter 5 of this dissertation for severity of illness score development. Clinically, *time* is an important factor that is currently not accounted for in the proposed framework. For instance, consecutive indications of deterioration within a certain time period can be a sign of being in a worse health state than a single indication alone would (e.g. two measurements of systolic blood pressure less than 70 that are greater than 30 minutes but less than a few hours apart from each other is a indication of a patient going into a shock state). This can be taken into account in the data pre-processing and observation generation steps but we would also like to explore how to account for this in the optimization model as well.

We believe that there is a natural bilevel optimization application in the development of severity of illness scoring systems. Those parameters that are selected prior to the solving of the model such as penalty parameters (γ and ρ) and interval bounds ($\boldsymbol{l}$ and $\boldsymbol{u}$) can be the variables for the leaders problem where they attempt to maximized the average probability of correct classification in a partition of the training set observations. Then given those parameters, the follower attempts to construct a score by solving our proposed score development problem via the ADMM procedure. There are other variants of this bilevel model that could be considered such as the leader decides upon the optimal probabilities $\boldsymbol{Q}$ given a set of point values decided upon by the follower, or vice-versa. We also plan on exploring the relationship of the proposed score development model to decision trees and random forests.

REFERENCES

[1] Abdi, M., Hosseini, S. & Rezghi, M. "A novel weighted support vector machine based on particle swarm optimization for gene selection and tumor classification". *Computational and Mathematical Methods in Medicine* (2012).

[2] Afessa, B. et al. "The influence of missing components of the Acute Physiology Score of APACHE III on the measurement of ICU performance". *Intensive Care Medicine* **31**.11 (2005), pp. 1537–1543.

[3] Agor, J. K. & Özaltın, O. Y. "Models for predicting the evolution of influenza to inform vaccine strain selection". *Human Vaccines & Immunotherapeutics* **14**.3 (2018), pp. 678–683.

[4] Albright, C. M. et al. "The Sepsis in Obstetrics Score : a model to identify risk of morbidity from sepsis in pregnancy". *The American Journal of Obstetrics & Gynecology* **211**.1 (2014), 39.e1–39.e8.

[5] Audet, C., Digabel, L. S. & Tribes, C. "NOMAD user guide". *Technical Report* (2009), pp. 267–288.

[6] Audet, C. et al. "Links between linear bilevel and mixed 0–1 programming problems". *Journal of Optimization Theory and Applications* **93** (1997), 273ï£¡–300.

[7] Audra, G. et al. "Emergency care workload units: A novel tool to compare emergency department activity". *Emergency Medicine Australasia* (2010).

[8] Aybat, N, Zarmehri, S & Kumara, S. "An ADMM Algorithm for Clustering Partially Observed Networks". *Proceedings of the 2015 SIAM international conference on data mining*. 2015.

[9] Bard, J. F. *Practical bilevel optimization*. Kluwer Academic Publishers, 1998.

[10] Bard, J. F., Plummer, J. & Sourie, J. C. "A bilevel programming approach to determining tax credits for biofuel production". *European Journal of Operational Research* **120** (2000), pp. 30–46.

[11] Bard, J. F. & Purnomo, H. W. "Hospital-wide reactive scheduling of nurses with preference considerations". *IIE Transactions* **37**.7 (2005), pp. 589–608.

[12] Beaujean, A. A. "Package 'BaylorEdPsych'" (2015), p. 16.

[13] Beaulieu-Jones, B. K. et al. "Characterizing and managing missing structured data in electronic health records: Data analysis". *JMIR Medical Informatics* **6**.1 (2018).

[14] Bertolazzi, P. et al. "Logic classification and feature selection for biomedical data". *Computers and Mathematics with Applications* **55** (2008), pp. 889–899.

[15] Bertsekas, D. P. & Eckstein, J. “Dual coordinate step methods for linear network flow problems”. *Mathematical Programming* **42** (1988), pp. 203–243.

[16] Bertsekas, D. P. & Tsitsiklis, J. *Parallel and Distributed Computation Numerical Methods*. 1989.

[17] Bewersdorf, J. et al. “The SPEED (sepsis patient evaluation in the emergency department) score: a risk stratification and outcome prediction tool”. *European Journal of Emergency Medicine* (2017), pp. 170–175.

[18] Bhatal, N et al. “Global Guidance For Cervical Caner Prevention and Control - Technical Report”. *WHO International Federation of Gynecology & Obstetrics* (2009).

[19] Bi, J. et al. “Dimensionality reduction via sparse support vector machines”. *Journal of Machine Learning Research* **3** (2003), pp. 1229–1243.

[20] Bi, Z., Calamai, P. & Conn, A. “An exact penalty function approach for the nonlinear bilevel programming problem”. *Technical Report #180-O-170591, Department of Systems Design Engineering, University of Waterloo* (1991).

[21] Blum, C., Roli, A. & Sampels, M. *Hybrid Metaheuristics*. 2008.

[22] Boland, N. et al. “Combining Progressive Hedging with a Frank-Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer”. *Preprint* (2016).

[23] Bonami, P. et al. “An algorithmic framework for convex mixed integer nonlinear programs”. *Discrete Optimization* **5** (2008), pp. 186–204.

[24] Bone, R. et al. “Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. The ACCP/SCCM Consensus Conference Committee. American College of Chest Physicians/Society of Critical Care Medicine.” *Chest* **101**.6 (1992), pp. 1644–1655.

[25] Boukouvala, F., Misener, R. & Floudas, C. A. “Global optimization advances in Mixed-Integer Nonlinear Programming , MINLP , and Constrained Derivative-Free Optimization , CDFO”. *European Journal of Operational Research* **252**.3 (2016), pp. 701–727.

[26] Boyd, S. et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. *Foundations and Trends in Machine Learning* **3**.1 (2010), pp. 1–122.

[27] Brotcorne, L., Hanafi, S. & Mansi, R. “A dynamic programming algorithm for the bilevel knapsack problem”. *Operations Research Letters* **37**.3 (2009), pp. 215–218.

[28] Buhi, E., Goodson, P. & Neilands, T. “Out of sight, not out of mind: Strategies for handling missing data”. *American Journal of Health Behavior* **32**.1 (2008), pp. 83–92.

[29] Burer, S. & Letchford, A. N. “Surveys in Operations Research and Management Science Non-convex mixed-integer nonlinear programming : A survey”. *Surveys in Operations Research and Management Science* **17**.2 (2012), pp. 97–106.

[30] Burgard, A. P., Pharkya, P. & Maranas, C. D. “Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization”. *Biotechnol. Bioeng.* **84**.6 (2003), pp. 647–657.

[31] Burton, J & Altman, D. “Missing covariate data within cancer prognostic studies: A review of current reporting and proposed guidelines”. *British Journal of Cancer* **91**.1 (2004), pp. 4–8.

[32] Buuren, S. van & Groothuis-Oudshoorn, K. “ **mice**: Multivariate Imputation by Chained Equations in *R*”. *Journal of Statistical Software* **45**.3 (2011).

[33] Calle, P et al. “Usefulness of Severity Scores in Patients with Suspected Infection in the Emergency Department: A Systematic Review”. *The Journal of Emergency Medicine* **42**.4 (2012), pp. 379–391.

[34] Calvete, H. I., Gale, C. & Mateo, P. M. “A new approach for solving linear bilevel problems using genetic algorithms”. *European Journal of Operational Research* **188** (2008), pp. 14–28.

[35] Cardoso, T. et al. “Predisposition, Insult/Infection, Response and Organ Dysfunction (PIRO): A Pilot Clinical Staging System for Hospital Mortality in Patients with Infection”. *PLoS ONE* **8**.7 (2013), pp. 1–10.

[36] Carrat, F. & Flahault, A. “Influenza vaccine: The challenge of antigenic drift”. *Vaccine* **25** (2007), pp. 6852–6862.

[37] CDC. *Influenza: The Disease*. Available at http://www.cdc.gov/flu/about/disease/index.htm. Accessed January 14, 2008. 2008.

[38] Cevher, V., Becker, S. & Schmidt, M. “Convex Optimization for Big Data: Scalable, randomized, and parallel algorithms for big data analytics”. *IEEE Signal Processing Magazine* **31** (2014), pp. 32–43.

[39] Chalfin, D. B. et al. “Impact of delayed transfer of critically ill patients from the emergency department to the intensive care unit”. *Critical Care Medicine* **35**.6 (2007), pp. 1477–1483.

[40] C*ôté*, J.-P. & Savard, G. “A bilevel modeling approach to pricing and fare optimization in the airline industry”. *Journal of Revenue and Pricing Management* **2** (2003), pp. 23–26.

[41] Chen, C. L. P. & Zhang, C.-y. “Data-intensive applications , challenges , techniques and technologies : A survey on Big Data”. *Information Sciences* **275** (2014), pp. 314–347.

[42] Chen, D., Batson, R. G. & Dang, Y. *Applied Integer Programming*. 2010, p. 490.

[43] Chen, K. et al. “Development and validation of a parsimonious and pragmatic CHARM score to predict mortality in patients with suspected sepsis”. *American Journal of Emergency Medicine* **35**.4 (2017), pp. 640–646.

[44] Chen, Y. & Li, C.-S. “Risk stratification and prognostic performance of the predisposition, infection, response, and organ dysfunction (PIRO) scoring system in septic patients in the emergency department: a cohort study”. *Critical Care* **18**.2 (2014), R74.

[45] Churpek, M. M. et al. “Multicenter development and validation of a risk stratification tool for ward patients”. *American Journal of Respiratory and Critical Care Medicine* **190**.6 (2014), pp. 649–655.

[46] Colson, B., Marcotte, P. & Savard, G. “An overview of bilevel optimization”. *Annals of Operations Research* **153**.1 (2007), pp. 235–256.

[47] Corfield, A. R. et al. “Utility of a single early warning score in patients with sepsis in the emergency department”. **31** (2014), pp. 482–487.

[48] Crameri, A. et al. “Sensitivity analysis in multiple imputation in effectiveness studies of psychotherapy”. *Frontiers in Psychology* **6** (2015), pp. 1–11.

[49] Crick, F. “Central dogma of molecular biology”. *Nature* **227**.5258 (1970), pp. 561–563.

[50] Dagliyan, O. et al. “Optimization based tumor classification from microarray gene expression data”. *PLoS ONE* **6**.2 (2011).

[51] “Dealing with missing predictor values when applying clinical prediction models”. *Clinical Chemistry* **55**.5 (2009), pp. 994–1001.

[52] Delong, E. R. & Carolina, N. “Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves : A Nonparametric Approach”. *Biometrics* **44**.3 (2016), pp. 837–845.

[53] Dempe, S. “Discrete bilevel optimization problems”. *Technical Report D-04109, Universitat Leipzig, Leipzig, Germany* (2001).

[54] Dempe, S. *Foundations of bilevel programming*. Dordrecht: Kluwer Academic, 2002.

[55] Dempe, S. & Richter, K. “Bilevel programming with knapsack constraints”. *Central European Journal of Operations Research* **8**.2 (2000), pp. 93–107.

[56] DeNegre, S. *Interdiction and discrete bilevel linear programming*. PhD thesis, Lehigh University, 2011.

[57] Deng, W. & Yin, W. “On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers”. *Journal of Scientific Computing* **66**.3 (2016), pp. 889–916.

[58] Dheeru, D. & K. T., E. *University of California, Irvine Machine Learning Repository*. Available at http://archive.ics.uci.edu/ml. Accessed April 11, 2018. 2017.

[59] Dobson, A. J. *An introduction to generalized linear models*. 2002, p. 225.

[60] Doerr, E. et al. "Between-Visit Workload in Primary Care". *Journal of General Internal Medicine* **25**.12 (2010), pp. 1289–1292.

[61] Donders, A. R. T. et al. "Review: A gentle introduction to imputation of missing values". *Journal of Clinical Epidemiology* **59**.10 (2006), pp. 1087–1091.

[62] Dries, M. et al. "Non-direct patient care factors influencing nursing workload: a review of the literature". *Journal of Advanced Nursing* **67**.10 (2011), pp. 2109–2129.

[63] Duffield, C. et al. "Nursing staffing, nursing workload, the work environment and patient outcomes". *Applied Nursing Research* **24**.4 (2011), pp. 244 –255.

[64] Edmunds, T. & Bard, J. "Algorithms for nonlinear bilevel mathematical programs". *IEEE Transactions on Systems, Man, and Cybernetics* **21** (1991), pp. 83–89.

[65] Enders, C. K., Mistler, S. A. & Keller, B. T. "Multilevel multiple imputation: A review and evaluation of joint modeling and chained equations imputation". *Psychological Methods* **21**.2 (2016), pp. 222–240.

[66] Espadaler, J. et al. "Prediction of protein-protein interactions using distant conservation of sequence patterns and structure relationships". *Bioinformatics* **21** (2005), pp. 3360–3368.

[67] Falk, J. & Liu, J. "On bilevel programming, Part I : general nonlinear cases". *Mathematical Programming* **70** (1995), pp. 47–72.

[68] Fanaroff, A. C. et al. "Risk Score to Predict Need for Intensive Care in Initially". *Journal of the American Heart Association* **7**.11 (2018).

[69] Fernandes, K., Cardoso, J. S. & Fernandes, J. "Transfer learning with partial observability applied to cervical cancer screening". *Lecture Notes in Computer Science* **10255** (2017), pp. 243–250.

[70] Freed N., G. F. "Simple but powerful goal programming models for discriminant problems". *European Journal of Operational Research* **7**.1 (1981).

[71] Freed N., G. F. "Resolving Certain Difficulties and Improving the Classification Power of LP Discriminant Analysis Formulations". *Decision Sciences* **17**.4 (2007).

[72] Freed, N. & Glover, F. "A Linear Programming Approach to the Discriminant Problem". *Applications and Implementation* (1981), pp. 68–74.

[73] Friedman, J., Hastie, T. & Tibshirani, R. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* **33**.1 (2010), pp. 1–22. arXiv: NIHMS201118.

[74] Friedman, L. S. et al. "Confirmation of BRCA1 by analysis of germline mutations linked to breast and ovarian cancer in ten families". *Nature Genetics* **8**.4 (1994), pp. 399–404.

[75] Gabay, D. & Mercier, B. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation". *Comps. & Maths. with Appls* **2**.1 (1976), pp. 17–40.

[76] Gage, B. F. et al. "Validation of Clinical Classification Schemes Results From the National Registry of Atrial Fibrillation". **285**.22 (2001), pp. 2864–2870.

[77] Gendreau, M., Marcotte, P. & Savard, G. "A Hybrid Tabu-Ascent Algorithm for the Linear Bilevel Programming Problem". *Journal of Global Optimization* **8** (1996), pp. 217–233.

[78] Gendreau, M. & Potvin, J. "Metaheuristics in Combinatorial Optimization". *Annals of Operations Research* **140** (2005), pp. 189–213.

[79] Ghosh, D. & Chinnaiyan, A. M. "Classification and selection of biomarkers in genomic data using LASSO". *Journal of Biomedicine and Biotechnology* **2005**.2 (2005), pp. 147–154.

[80] Gillespie, I. A. et al. "Development and validation of a predictive mortality risk score from a European hemodialysis cohort" (2015), pp. 996–1008.

[81] Glen, J. "Integer Programming Methods for Normalisation and Variable Selection in Mathematical Programming Discriminant Analysis Models". *The Journal of the Operational Research Society* **50**.10 (1999), pp. 1043–1053.

[82] Glen, J. J. "A comparison of standard and two-stage mathematical programming discriminant analysis methods". *European Journal of Operational Research* (2006).

[83] Glen, J. "Classification Accuracy in Discriminant Analysis : A Mixed Integer Programming Approach". *Journal of the Operational Research Society* **52**.3 (2001), pp. 328–339.

[84] Glover, F. & Laguna, M. "General Purpose Heuristics for Integer-Part I". *Journal of Heuristics* **358** (1997), pp. 343–358.

[85] Glover, F. & Laguna, M. "General Purpose Heuristics for Integer Programming-Part II". *Journal of Heuristics* **179** (1997), pp. 161–179.

[86] Gorzałczany, M. B. & Rudziński, F. "Classification of Splice-Junction DNA Sequences Using Multi-objective Genetic-Fuzzy Optimization Techniques". *Artificial Intelligence and Soft Computing*. Ed. by Rutkowski, L. et al. Cham, 2017, pp. 638–648.

[87] Granata, D., Steeger, G. & Rebennack, S. “Network interdiction via a critical disruption path: Branch-and-price algorithms”. *Computers & Operations Research* **40**.11 (2013), pp. 2689–2702.

[88] Grossmann, I. E. “Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques”. **3** (2002), pp. 227–252.

[89] Guazzi, M. et al. “Clinical Trial Development of a Cardiopulmonary Exercise Prognostic Score for Optimizing Risk Stratification in Heart Failure : The (P) e (R) i (O) dic (B) reathing During (E) xercise (PROBE) Study”. *Journal of Cardiac Failure* **16**.10 (2010), pp. 799–805.

[90] Gurses, A. P., Carayon, P & Wall, M. “Impact of performance obstacles on intensive care nurses’ workload, perceived quality and safety of care, and quality of working life”. *Health Services Research* **44**.2 (2009), pp. 422–443.

[91] Guyon, I. et al. “Gene selection for cancer classification using support vector machines”. *Machine learning* **46**.1 (2002), pp. 389–422.

[92] Gzara, F. “A cutting plane approach for bilevel hazardous material transport network design”. *Operations Research Letters* **41**.1 (2013), pp. 40–46.

[93] Hall, M. A. & Smith, L. A. “Feature subset selection: a correlation based filter approach”. *International Conference on Neural Information Processing and Intelligent Information Systems*. 1997, pp. 855–858.

[94] Hall, M. J. et al. “Inpatient care for septicemia or sepsis: a challenge for patients and hospitals.” *NCHS Data Brief* **62**.62 (2011), pp. 1–8.

[95] Hansen, P., Jaumard, B. & Savard, G. “New branch-and-bound rules for linear bilevel programming”. *SIAM Journal on Scientific and Statistical Computing* **13** (1992), pp. 1194–1217.

[96] Harel, O. & Zhou, X. H. “Multiple imputation: review of theory, implementation and software”. *Statistics in medicine* **26**.July 2007 (2007), pp. 3057–3077.

[97] Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity. The Lasso and Generalizations*. 2015.

[98] He, X. & Deng, L. “Speech-Centric Information Processing: An Optimization-Oriented Approach”. *Proceedings of the IEEE* **101**.5 (2013), pp. 1116–1135.

[99] Hejazi, S. R. et al. “Linear bilevel programming solution by genetic algorithm”. *Computers and Operations Research* **29** (2002), pp. 1913–1925.

[100] Hennessy, S. et al. “Factors influencing the optimal control-to-case ratio in matched case-control studies”. *American Journal of Epidemiology* **149**.2 (1999), pp. 195–197.

[101] Hosmer, D., Lemeshow, S. & Sturdivant, R. *Applied Logistic Regression*. John Wiley Sons, 2013.

[102] Howell, M. D. et al. "Proof of principle: The predisposition, infection, response, organ failure sepsis staging system*". *Critical Care Medicine* **39**.2 (2011), pp. 322–327.

[103] Huang, C. & Wang, C. "A GA-based feature selection and parameters optimization for support vector machines". *Expert Systems with Applications* **31**.2 (2006), pp. 231 –240.

[104] Huang, K. & Sidiropoulos, N. D. "Consensus-ADMM for General Quadratically Constrained Quadratic Programming". *IEEE Transactions on Signal Processing* **64**.20 (2016), pp. 5297–5310.

[105] Hulshof, P. et al. "Patient admission planning using Approximate Dynamic Programming". *Flexible services and manufacturing journal* **28**.1 (2016). Open access, pp. 30–61.

[106] Hutton, M. et al. "Association of missense and 5'-splice-site mutations in tau with the inherited dementia FTDP-17". *Nature* **393**.6686 (1998), pp. 702–704.

[107] Israeli, E. & Wood, R. K. "Shortest-path network interdiction". *Networks* **40**.2 (2002), pp. 97–111.

[108] Janssen, K. J. et al. "Missing covariate data in medical research: To impute is better than to ignore". *Journal of Clinical Epidemiology* **63**.7 (2010), pp. 721–727.

[109] Jennrich, R. I. "Stepwise regression". *Statistical Methods for Digital Computers* (1977), pp. 58–75.

[110] Jeroslow, R. "The Polynomial Hierarchy and a Simple Model for Competitive Analysis". *Mathematical Programming* **32** (1985), pp. 146–164.

[111] Joachims, T. "Text categorization with support vector machines: Learning with many relevant features." *Proceedings of 10th European Conference on Machine Learning (ECML-98)*. Springer, 1998, pp. 137–142.

[112] Jones, A., Trzeciak, S. & Kline, J. "The Sequential Organ Failure Assessment score for predicting outcome in patients with severe sepsis and evidence of hypoperfusion at the time of emergency department presentation". **37**.5 (2010), pp. 1649–1654.

[113] Judice, J. & Faustino, A. "The linear-quadratic bilevel programming problem". *Information Systems and Operational Research* **32** (1994), pp. 87–98.

[114] Junger, M. et al. *50 Years of Integer Programming 1958-2008*. 2010, p. 811.

[115] Karatzoglou, A., Meyer, D. & Hornik, K. *Support vector machines in R*. Department of Statistics and Mathematics, WU Vienna University of Economics and Business. 2005.

[116] Keegan, M. T., Gajic, O. & Afessa, B. “Severity of illness scoring systems in the intensive care unit”. *Critical Care Medicine* **39**.1 (2011), pp. 163–169.

[117] Keegan, M. T., Gajic, O. & Afessa, B. “Comparison of APACHE III, APACHE IV, SAPS 3, and MPM0III and influence of resuscitation status on model performance”. *Chest* **142**.4 (2012), pp. 851–858.

[118] Kendall, M. G. “A New Measure of Rank Correlation”. *Biometrika* **30**.1 (1938), pp. 81–93.

[119] Kenward, M. G. & Carpenter, J. “Multiple imputation: current perspectives.” *Stat Meth Med Res* **16**.3 (2007), pp. 199–218.

[120] Kessler, R. C. et al. “The World Health Organization adult ADHD self-report scale (ASRS): a short screening scale for use in the general population” (2005), pp. 245–256.

[121] Kilbourne, E. D., Johansson, B. E. & Grajower, B. “Independent and disparate evolution in nature of influenza A virus hemagglutinin and neuraminidase glycoproteins.” *Proceedings of the National Academy of Sciences of the United States of America* **87**.2 (1990), pp. 786–790.

[122] Knaus, W. et al. “The APACHE III prognostic system: Risk prediction of hospital mortality for critically III hospitalized adults”. **100** (1992), pp. 1619–36.

[123] Knol, M. J. et al. “Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example”. *Journal of Clinical Epidemiology* **63**.7 (2010), pp. 728–736.

[124] Kohavi, R. & John, G. H. “Wrappers for feature subset selection”. *Artificial intelligence* **97**.1-2 (1997), pp. 273–324.

[125] Kotsiantis, S. B., Zaharakis, I. D. & Pintelas, P. E. “Machine learning: A review of classification and combining techniques”. *Artificial Intelligence Review* (2006).

[126] Kumar, S. et al. “Fifty Years of Integer Programming : A Review of the Solution Approaches”. **VI**.3 (2010), pp. 5–15.

[127] Kunapuli, G. et al. “Classification model selection via bilevel programming”. *Optimization Methods and Software* **23**.4 (2008), pp. 475–489.

[128] Kwiecien-Jagus, K., Wujtewicz, M. & Medrzychka-Dabrowska, W. “Selected methods of measuring workload among intensive care nursing staff”. **25** (2012), pp. 209–17.

[129] Le Digabel, S. “NOMAD: Nonlinear Optimization with the MADS Algorithm”. *ACM Transactions on Mathematical Software* **37**.4 (2011), pp. 1–15.

[130] Le Gall, J.-R., Lemeshow, S. & Saulnier, F. “Simplified Acute Physiology Score (SAPS II) Based on a European / North American multicenter study”. *Jama* **270** (1993), pp. 2957–2963.

[131] Lee, C. et al. *Handbook of Large-Scale Distributed Computing in Smart Healthcare*. 2017, pp. 11–41.

[132] Lee, E. K. "Large-Scale Optimization-Based Classification Models in Medicine and Biology". *Annals of Biomedical Engineering* **35**.6 (2007), pp. 1095–1109.

[133] Lee, E. K. & Wu, T. L. "Classification and Disease Prediction via Mathematical Programming". *Handbook for Optimization in Medicine*. Springer Science+Business Media LLC, 2009, pp. 381–430.

[134] Lee, M. S. & Chen, J. "Predicting antigenic variants of influenza A/H3N2 viruses". *Emerging Infectious Diseases* **10**.8 (2004), pp. 1385–1390.

[135] Levin, S. et al. "Shifting Toward Balance: Measuring the Distribution of Workload Among Emergency Physician Teams". *Annals of Emergency Medicine* **50**.4 (2007), pp. 419–423.

[136] Liao, Y. C. et al. "Bioinformatics models for predicting antigenic variants of influenza A/H3N2 virus". *Bioinformatics* **24**.4 (2008), pp. 505–512.

[137] Liaw, a & Wiener, M. "Classification and Regression by randomForest". *R news* **2**.December (2002), pp. 18–22.

[138] Little, R. J. A. "A test of missing completely at random for multivariate data with missing values". *Journal of the American Statistical Association* **83**.404 (1988), pp. 37–41.

[139] Liu, V. et al. "Hospital Deaths in Patients With Sepsis From 2 Independent Cohorts". *Journal of the American Medical Association* **312**.1 (2014), pp. 90–92.

[140] Lozano, L. & Smith, J. C. "A value-function-based exact approach for the bilevel mixed integer programming problem". *Oper. Res., to appear* (2017).

[141] Ma, S., Song, X. & Huang, J. "Supervised group Lasso with applications to microarray data analysis." *BMC Bioinformatics* **8** (2007), p. 60.

[142] Macdonald, S. P. J. et al. "Comparison of PIRO, SOFA, and MEDS scores for predicting mortality in emergency department patients with severe sepsis and septic shock". *Academic Emergency Medicine* **21**.11 (2014), pp. 1257–1263.

[143] Malhotra, R. et al. "Original Articles A risk prediction score for acute kidney injury in the intensive care unit". April (2017), pp. 814–822.

[144] Mangasarian, O., Street, W. & Wolberg, W. "Breast Cancer Diagnosis and Prognosis Via Linear Programming". *Operations Research* **43** (1995), pp. 570–577.

[145] Marcotte, P. & Savard, G. "Bilevel programming: A combinatorial perspective". *Graph theory and combinatorial optimization*. Ed. by Avis, D., Hertz, A. & Marcotte, O. Kluwer Academic Publishers, Boston, 2005.

[146] Marinakis, Y. et al. “Intelligent and nature inspired optimization methods in medicine: the Pap smear cell classification problem”. *Expert Systems* **26**.5 (2009), pp. 433–457.

[147] Marshall, J. et al. “Multiple Organ Dysfunction Score : A reliable descriptor of a complex clinical outcome”. *Cricital Care Medicine* **23**.10 (1995), pp. 1638–1652.

[148] Masconi, K. L. et al. “Reporting and handling of missing data in predictive research for prevalent undiagnosed type 2 diabetes mellitus: A systematic review”. *EPMA Journal* **6**.1 (2015).

[149] McLachlan, G. *Discriminant Analysis and Statistical Pattern Recognition*. Vol. 544. John Wiley & Sons, 2004.

[150] Meher, P. K. et al. “A computational approach for prediction of donor splice sites with improved accuracy”. *Journal of Theoretical Biology* **404** (2016), pp. 285–294.

[151] Migdalas, A. “Bilevel programming in traffic planning: Models, methods and challenge”. *Journal of Global Optimization* **7** (1995), 381–ï£¡405.

[152] Migdalas, A., Pardalos, P. M. & Värbrand, P. *Multilevel optimization: algorithms and applications.* Norwell: Kluwer Academic Publishers, 1998.

[153] Migdalas, A., Pardalos, P. M. & Värbrand, P. *Multilevel Optimization: Algorithms and Applications*. Vol. 20. Springer Science & Business Media, 2013.

[154] Momma, M. & Bennett, K. P. “A pattern search method for model selection of support vector regression”. *Proceedings of the 2002 SIAM International Conference on Data Mining*. Ed. by Grossman, R. et al. SIAM. 2002, pp. 261–274.

[155] Moons, K. G. et al. “Using the outcome for imputation of missing predictor values was preferred”. *Journal of Clinical Epidemiology* **59**.10 (2006), pp. 1092–1101.

[156] Moore, J. T. & Bard, J. F. “The mixed integer linear bilevel programming problem”. *Operations Research* **38**.5 (1990), pp. 911–921.

[157] Mota, J. et al. “Distributed Basis Pursuit”. *IEEE Transactions on Signal Processing* **60**.4 (2012), pp. 1942–1956.

[158] Nelson, M. I. & Holmes, E. C. “The evolution of epidemic influenza.” *Nature reviews. Genetics* **8**.3 (2007), pp. 196–205.

[159] Oh, I., Lee, J. & Moon, B. “Hybrid genetic algorithms for feature selection.” *IEEE transactions on pattern analysis and machine intelligence* **26**.11 (2004), pp. 1424–1437.

[160] Özaltın, O. Y., Prokopyev, O. A. & Schaefer, A. J. “The bilevel knapsack problem with stochastic right-hand sides”. *Oper. Res. Lett.* **38**.4 (2010), pp. 328–333.

[161] Pang, P., Lee, L. & Vaithyanathan, S. “Thumbs up? Sentiment Classification using Machine Learning Techniques”. *Proceedings of the Converence on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2002, pp. 79–86.

[162] Pennachiao, L. A. et al. “Mutations in the gene encoding Cystatin B in progressive myoclonus epilepsy (EPM1)”. *Science* **271**.5256 (1996), pp. 1731–1734.

[163] Phuong, T. M., Lin, Z. & Altman, R. B. “Choosing SNPs using feature selection”. *2005 IEEE Computational Systems Bioinformatics Conference (CSB’05)*. 2005, pp. 301–309.

[164] Punnakitikashem, P., Rosenberber, J. M. & Buckley-Behan, D. F. “A stochastic programming approach for integrated nurse staffing and assignment”. *IIE Transactions* **45**.10 (2013), pp. 1059–1076.

[165] Raith, E. P. et al. “Prognostic accuracy of the SOFA score, SIRS criteria, and qSOFA score for in-hospital mortality among adults with suspected infection admitted to the intensive care unit”. *JAMA - Journal of the American Medical Association* **317**.3 (2017), pp. 290–300.

[166] Ren, S., Zeng, B. & Qian, X. “Adaptive bi-level programming for optimal gene knockouts for targeted overproduction under phenotypic constraints”. *BMC Bioinformatics* **14**.2 (2013), S17.

[167] Rhee, C. et al. “Incidence and Trends of Sepsis in US Hospitals Using Clinical vs Claims Data, 2009-2014”. **02215** (2017), pp. 2009–2014.

[168] Richtárik, P. & Takác, M. “Parallel coordinate descent methods for big data optimization”. *Mathematical Programming* **156** (2016), pp. 433–484.

[169] Rockafeller, R & Wets, R. “Scenarios and policy aggregation in optimization under uncertainty”. *International Institute for Applied Systems Analysis* (1987).

[170] Roffo, G., Melzi, S. & Cristani, M. “Infinite Feature Selection”. *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4202–4210.

[171] Romaniuk, H., Patton, G. C. & Carlin, J. B. “Multiple imputation in a longitudinal cohort study: A case study of sensitivity to imputation methods”. *American Journal of Epidemiology* **180**.9 (2014), pp. 920–932.

[172] Romero-Brufau, S. et al. “Why the C-statistic is not informative to evaluate early warning scores and what metrics to use”. *Critical Care* **19**.1 (2015), pp. 19–24.

[173] Saitta, L. “Support-Vector Networks”. **297** (1995), pp. 273–297.

[174] Sartelli, M. et al. “Global validation of the WSES Sepsis Severity Score for patients with complicated intra-abdominal infections : a prospective multicentre study (WISS Study)”. *World Journal of Emergency Surgery* (2015), pp. 1–8.

[175] Savard, G. & Gauvin, J. “The steepest descent direction for the nonlinear bilevel programming problem”. *Operations Research Letters* **15** (1994), 265ï£¡–272.

[176] Schafer, J. L. & Graham, J. W. “Missing data: Our view of the state of the art”. *Psychological Methods* **7**.2 (2002), pp. 147–177.

[177] Schizas, I. D., Ribeiro, A. & Giannakis, G. B. “Consensus in Ad Hoc WSNs With Noisy Links - Part I : Distributed Estimation of Deterministic Signals”. *IEEE Transactions on Signal Processing* **56**.1 (2008), pp. 350–364.

[178] Sedghi, H., Anandkumar, A. & Jonckheere, E. “Multi-Step Stochastic ADMM in High Dimensions: Applications to Sparse Optimization and Noisy Matrix Decomposition”. *Neural Information Processing Systems* (2015).

[179] Selker, H. et al. “A comparison of performance of mathematical predictive methods for medical diagnosis: identifying acute cardiac ischemia among emergency department patients”. *Journal of Investigative Medicine* **43**.5 (1995), 468âĂŤ476.

[180] Senapathy, P., Shapiro, M. B. & Harris, N. L. “Splice junctions, branch point sites, and exons: Sequence statistics, identification, and applications to genome project”. *Methods in Enzymology* **183**.C (1990), pp. 252–278.

[181] Seymour, C. et al. “Assessment of Clinical Criteria for Sepsis For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)”. *Journal of the American Medical Association* **315**.8 (2016), pp. 762–774.

[182] Shabanzadeh, P. & Yusof, R. “An efficient optimization method for solving unsupervised data classification problems”. *Computational and Mathematical Methods in Medicine* **2015** (2015).

[183] Shahamat, H. & Pouyan, A. A. “Feature selection using genetic algorithm for classification of schizophrenia using fMRI data”. *Journal of Artificial Intelligence and Data Mining* **3**.1 (2015), pp. 30–37.

[184] Shapiro, N. I. et al. “Mortality in Emergency Department Sepsis (MEDS) score: A prospectively derived and validated clinical prediction rule”. *Critical Care Medicine* **31**.3 (2003).

[185] Shapiro, N. I. et al. “Mortality in Emergency Department Sepsis (MEDS) score predicts 1-year mortality”. *Critical Care Medicine* **35**.1 (2007), pp. 192–198.

[186] Shen, S., Smith, J. & Goli, R. “Exact interdiction models and algorithms for disconnecting networks via node deletions”. *Discrete Optimization* **9**.3 (2012), pp. 172–188.

[187] Shieh, Y. et al. “Breast cancer risk prediction using a clinical risk model and polygenic risk score”. *Breast Cancer Research and Treatment* **159**.3 (2016), pp. 513–525.

[188] "Shock Index and Early Recognition of Sepsis in the Emergency Department: Pilot Study". *Western Journal of Emergency Medicine* **14**.2 (2013), pp. 168–174.

[189] Siedlecki, W. & Sklansky, J. "A note on genetic algorithms for large-scale feature selection". *Pattern Recognition Letters* **10**.5 (1989), pp. 335 –347.

[190] Singer, M. et al. "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". *Journal of the American Medical Association* **315**.8 (2017), pp. 801–810.

[191] Six, A. J., Backus, B. E. & Kelder, J. C. "Chest pain in the emergency room : value of the HEART score". **16**.6 (2008), pp. 191–196.

[192] Slavakis, K., Giannakis, G. B. & Mateos, G. "Modeling and Optimization for Big Data Analytics: (Statistical) learning tools for our era of data deluge". **31** (2014), pp. 18–31.

[193] Smith, D. J. et al. "Mapping the Antigenic and Genetic Evolution of Influenza Virus". *Science* **305**.5682 (2004), pp. 371–376.

[194] Stam, A. & Joachimsthaler, E. "A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem". *European Journal of Operational Research* **46**.1 (1990), pp. 113–122.

[195] Storlie, C. B. et al. "Prediction and Inference with Missing Data in Patient Alert Systems" (2017). eprint: 1704.07904.

[196] "Strategies for handling missing data in electronic health record derived data." *EGEMS (Washington, DC)* **1**.3 (2013), p. 1035.

[197] Su, L., Tang, L. & Grossmann, I. E. "Computational strategies for improved MINLP algorithms". *Computers and Chemical Engineering* **75** (2015), pp. 40–48.

[198] Sun, X. & Xu, W. "Fast Implementation of DeLong's Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves". *IEEE Signal Processing Letters* **21**.11 (2014), pp. 1389–1393.

[199] Tan, M., Pu, J. & Zheng, B. "Optimization of breast mass classification using sequential forward floating selection (SFFS) and a support vector machine (SVM) model". *International Journal of Computer Assisted Radiology and Surgery* (2014), pp. 1005–1020.

[200] Than, M. et al. "Development and validation of the Emergency Department Assessment of Chest pain Score and 2 h accelerated diagnostic protocol Objective : Methods : Results : Conclusion :" (2014), pp. 34–44.

[201] Thorwarth, M., Arisha, A. & Harper, P. "Simulation model to investigate flexible workload management for healthcare and servicescape environment". *Proceedings of the 2009 Winter Simulation Conference (WSC)*. 2009, pp. 1946–1956.

[202] Tibshirani, R. “Regression Selection and Shrinkage via the Lasso”. *Journal of the Royal Statistical Society* ().

[203] Torio, C. M. et al. “Statistical Brief # 160 National Inpatient Hospital Costs” (2013).

[204] Tripathi, G. & Naganna, S. “Feature Selection and Classification Approach for Sentiment analysis”. *Machine Learning and Applications: An International Journal* **2**.2 (2015).

[205] Tseytlin, Y. “Queuing systems with heterogeneous severs: On fair routing of patients in emergency departments”. *PhD thesis, Israel Institute of Technology* (2009).

[206] Tupchong, K., Koyfman, A. & Foran, M. “Sepsis, severe sepsis, and septic shock: A review of the literature”. *African Journal of Emergency Medicine* **5**.3 (2015), pp. 127–135.

[207] Üney, F. & Türkay, M. “A mixed-integer programming approach to multi-class data classification problem”. *European Journal of Operational Research* **173**.3 (2006), pp. 910–920.

[208] Ustun, B. & Rudin, C. “Optimized Risk Scores”. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2017.

[209] Vergouwe, Y. et al. “Development and validation of a prediction model with missing predictor data: a practical approach”. *Journal of Clinical Epidemiology* **63**.2 (2010), pp. 205–214.

[210] Vicente, L., Savard, G. & Judice, J. “Discrete linear bilevel programming problem”. *Journal of Optimization Theory and Applications* **89**.3 (1996), pp. 597–614.

[211] Vicente, L., Savard, G. & Judice, J. “Descent approaches for quadratic bilevel programming”. *Journal of Optimization Theory and Applications* **81** (1994), pp. 379–399.

[212] Vielma, J. “Mixed Integer Linear Programming Formulation Techniques”. *SIAM Review* **57**.1 (2015), pp. 3–57.

[213] Vincent, J et al. “The SOFA (sepsis-related organ failure assessment) score to describe organ dysfunction/ failure”. *Intensive Care Medicine* **22** (1996), pp. 707–710.

[214] Vincent, J.-l. & Moreno, R. “Clinical review : Scoring systems in the critically ill” (2010), pp. 1–9.

[215] Vincent, M. & Hansen, R. N. “Sparse group lasso and high dimensional multinomial classification”. *Computational Statistics and Data Analysis* **71** (2014), pp. 771–786.

[216] Wahlberg, B. et al. “An ADMM algorithm for a class of total variation regularized estimation problems”. *IFAC Symposium on System Identification*. Vol. 45. 16. IFAC, 2012, pp. 83–88.

[217] Wang, G. et al. “Genetic algorithm based on simplex method for solving linear-quadratic bilevel programming problem”. *Computers and Mathematics with Applications* **56** (2008), pp. 2550–2555.

[218] Wang, L. & Xu, P. *The watermelon algorithm for the bilevel integer linear programming problem*. To appear in *SIAM Journal on Optimization*. 2017.

[219] Watson, J.-P. & Woodruff, D. "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems". *Computational Management Sciences* **8** (2011), pp. 355–370.

[220] Wen, U. & Yang, Y. H. "Algorithms for solving the mixed integer two-level linear programming problem". *Comput. Oper. Res.* **17**.2 (1990), pp. 133–142.

[221] Wen, U. P. & Huang, A. D. "A simple Tabu Search method to solve the mixed-integer linear bilevel programming problem". *European Journal of Operational Research* **88** (1996), pp. 563–571.

[222] Wilson, J. "Integer programming formulations of statistical classification problems". *Omega* **24**.6 (1996), pp. 681–688.

[223] Wood, R. K. "Deterministic network interdiction". *Math. Comput. Model.* **17**.2 (1993), pp. 1–18.

[224] Wright, S. J. et al. "Optimization Algorithms and Applications for Speech and Language Processing". *IEEE Transactions on Audio, Speech, and Language Processing* **21**.11 (2013), pp. 2231–2243.

[225] Wu, T. L. & Lee, E. K. "Classification Models for Disease Diagnosis and Outcome Analysis PhD Dissertation" (2011).

[226] Xu, G. & Papageorgiou, L. "A mixed integer optimisation model for data classification". *Computers & Industrial Engineering* **56**.4 (2009), pp. 1205–1215.

[227] Yang, J. & Honavar, V. "Feature Subset Selection Using a Genetic Algorithm Feature Subset Selection Using A Genetic Algorithm Feature Subset Selection Using A Genetic Algorithm". *Computer Science Technical Reports. Paper* **156** (1997).

[228] Ye, Q., Zhang, Z. & Law, R. "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches". *Expert Systems with Applications* **36** (2009), pp. 6527–6535.

[229] Young, M. P. et al. "Inpatient transfers to the intensive care unit". *J Gen Intern Med* **18**.2 (2003), pp. 77–83.

[230] Yuen, M.-f. et al. "Independent risk factors and predictive score for the development of hepatocellular carcinoma in chronic hepatitis B q". *Journal of Hepatology* **50**.1 (2009), pp. 80–88.

[231] Zhang, J. & Özaltın, O. Y. *A branch-and-cut algorithm for discrete bilevel linear programs*. *Optimization Online*. Available at http://www.optimization-online.org/DB_HTML/2017/05/6012.html. Accessed April 11, 2018. 2017.

[232] Zopounidis, C. & Doumpos, M. *Multicriteria classification and sorting methods: A literature review*. 2002.

APPENDICES

APPENDIX

A

HANDLING OF MISSING DATA APPENDIX

Table A.1 Comparison of distributions for (a) age, (b) race, and (c) medical histories in the control and sample populations for the ICU transfer outcome

(a) Age Five Number Summary

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Control	18	50	63	61.8	77	90
Sample	18	50	63	62.0	77	90
pval	-	-	-	0.09	-	-

(b) Race Distribution

	White	American Indian/Alaska Nat.	Asian	Black/AA	Unknown	Other
Control	73.3%	0.2%	1.2%	22.3%	0.5%	2.6%
Sample	73.2%	0.1%	1.1%	22.4%	0.4%	2.6%
pval	0.60	0.78	0.99	0.64	0.71	0.64

(c) Medical Histories Distribution

	Control	Sample	pval
Aids	0.2%	0.2%	0.58
Alcohol Abuse	0.8%	0.8%	0.69
Anemic Disorders	6.2%	6.2%	0.94
Arrhythmia	1.1%	1.1%	0.82
Blood Loss	0.0%	0.0%	0.75
Congestive Heart Failure	4.0%	4.0%	0.88
Coronary Artery Disease	5.5%	5.5%	0.80
Chronic Pulmonary Disease	6.9%	6.9%	0.88
Coagulation Disorders	2.2%	2.2%	0.69
Depression	5.6%	5.6%	0.95
Diabetes without Complications	5.8%	5.8%	1.00
Diabetes with Complications	2.6%	2.6%	0.55
Illegal Drug Use	1.9%	1.9%	1.00
Hypertension	11.8%	11.8%	0.99
Hypothyroidism	3.3%	3.3%	0.94
Liver Disease	1.4%	1.4%	0.94
Lymph Disorders	0.3%	0.3%	0.71
Electrolyte Disorders	9.3%	9.4%	0.82
Malignancy Metastases	1.0%	1.0%	0.75
Neurologic Disorders	4.0%	4.0%	0.85
Obesity	6.0%	6.0%	0.89
Paralysis	1.1%	1.1%	0.81
Peripheral Vascular Disease	2.7%	2.7%	0.88
Psychological Disorders	2.2%	2.2%	0.78
Pulmonary Circulation Disorders	2.2%	2.2%	0.83
Renal Failure	4.0%	4.0%	0.96
Malignancy	2.2%	2.2%	0.97
Ulcers	0.0%	0.0%	0.83
Valve Disorders	2.4%	2.4%	0.64
Weight Loss	1.9%	1.9%	0.89
None	1.2%	1.2%	0.93

Table A.2 Comparison of distributions for (a) age, (b) race, and (c) medical histories in the control and sample populations for the mortality outcome

(a) Original observation of the dynamic variables with missing information

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Control	18	50	63	61.3	76	90
Sample	18	49	62	61.1	76	90
pval	-	-	-	0.09	-	-

(b) Race Distribution

	White	American Indian/Alaska Nat.	Asian	Black/AA	Unknown	Other
Control	73.1%	0.2%	1.2%	22.5%	0.5%	2.6%
Sample	72.8%	0.2%	1.1%	22.6%	0.4%	2.9%
pval	0.58	0.86	0.79	0.87	0.51	0.10

(c) Medical Histories Distribution

	Control	Sample	pval
Aids	0.2%	0.2%	0.25
Alcohol Abuse	0.9%	0.9%	0.34
Anemic Disorders	6.1%	6.1%	0.68
Arrhythmia	1.1%	1.1%	0.26
Blood Loss	0.0%	0.0%	0.65
Congestive Heart Failure	4.0%	4.0%	0.95
Coronary Artery Disease	5.5%	5.6%	0.46
Chronic Pulmonary Disease	6.9%	6.9%	0.80
Coagulation Disorders	2.3%	2.3%	0.91
Depression	5.6%	5.5%	0.45
Diabetes without Complications	5.9%	5.8%	0.62
Diabetes with Complications	2.6%	2.6%	0.76
Illegal Drug Use	1.9%	1.9%	0.84
Hypertension	11.6%	11.7%	0.50
Hypothyroidism	3.2%	3.3%	0.31
Liver Disease	1.5%	1.4%	0.42
Lymph Disorders	0.3%	0.3%	0.25
Electrolyte Disorders	9.5%	9.5%	0.57
Malignancy Metastases	0.9%	0.9%	0.33
Neurologic Disorders	4.0%	3.9%	0.33
Obesity	6.1%	6.1%	0.83
Paralysis	1.2%	1.1%	0.05
Peripheral Vascular Disease	2.7%	2.7%	0.85
Psychological Disorders	2.2%	2.2%	0.98
Pulmonary Circulation Disorders	2.2%	2.2%	0.74
Renal Failure	4.0%	3.9%	0.93
Malignancy	2.0%	2.1%	0.46
Ulcers	0.0%	0.0%	0.81
Valve Disorders	2.4%	2.6%	0.12
Weight Loss	1.9%	1.9%	0.49
None	1.2%	1.2%	0.33

Table A.3 Example for generating observation vectors*

(a) Original observation of the dynamic variables with missing information

RR	Bands	HR	BUN	Hypoxia	Lactate	SBP	Platelet
Missing	Missing	111	22	0	Missing	99	233

(b) Observation vector 1

RR	Bands	HR	BUN	Hypoxia	Lactate	SBP	Platelet
16	3	111	22	0	1.5	99	233

(c) Observation vector 2

RR	RR Indicator	Bands	Bands Indicator	HR	HR Indicator	BUN	BUN Indicator
16	1	3	1	111	0	22	0

Hypoxia	Hypoxia Indicator	Lactate	Lactate Indicator	SBP	SBP Indicator	Platelet	Platelet Indicator
0	0	1.5	1	99	0	233	0

(d) Observation vector 3

P	I	R	O
4	4	0	2

(e) Observation vector 4

P	I	R	O
4	4	0	2

RR Indicator	Bands Indicator	HR Indicator	BUN Indicator	Hypoxia Indicator	Lactate Indicator	SBP Indicator	Platelet Indicator
1	1	0	0	0	1	0	0

**see Table 3.2 for definitions of observation vectors*

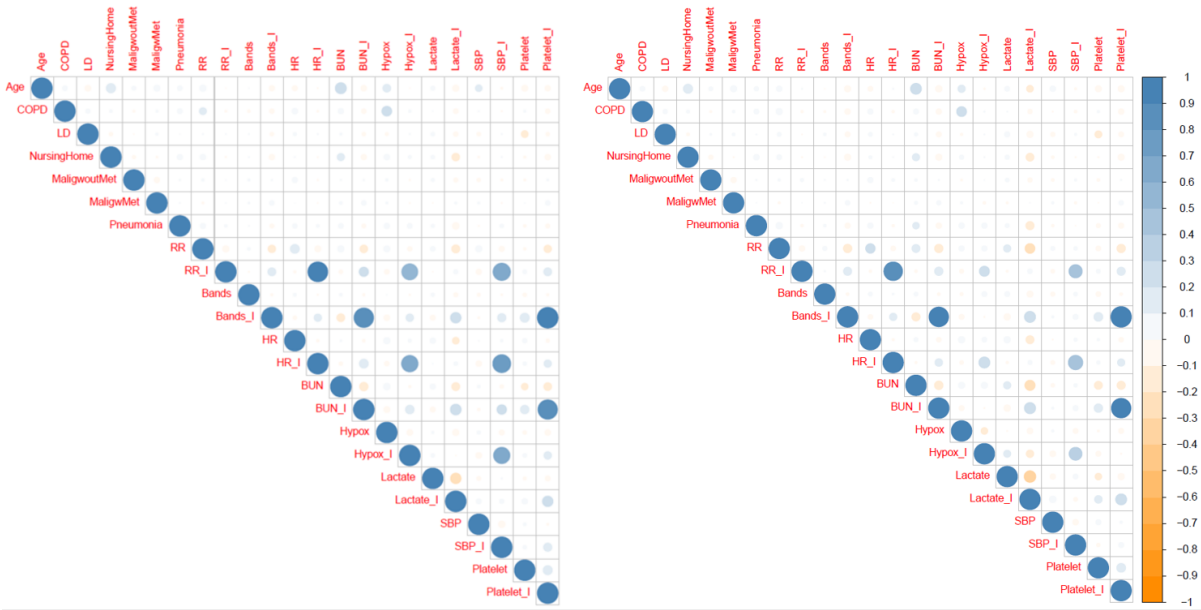
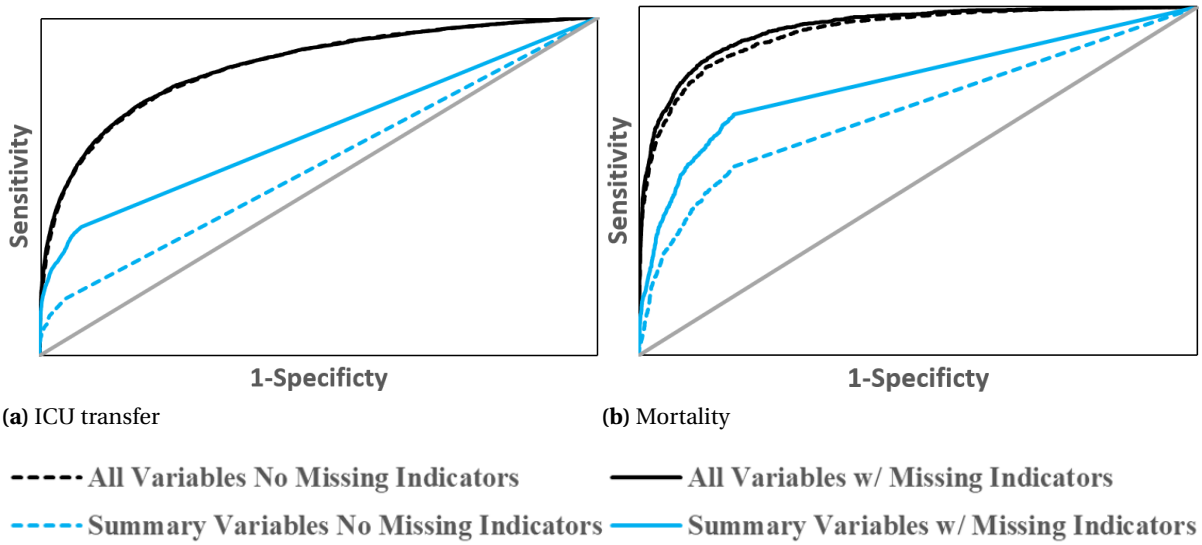
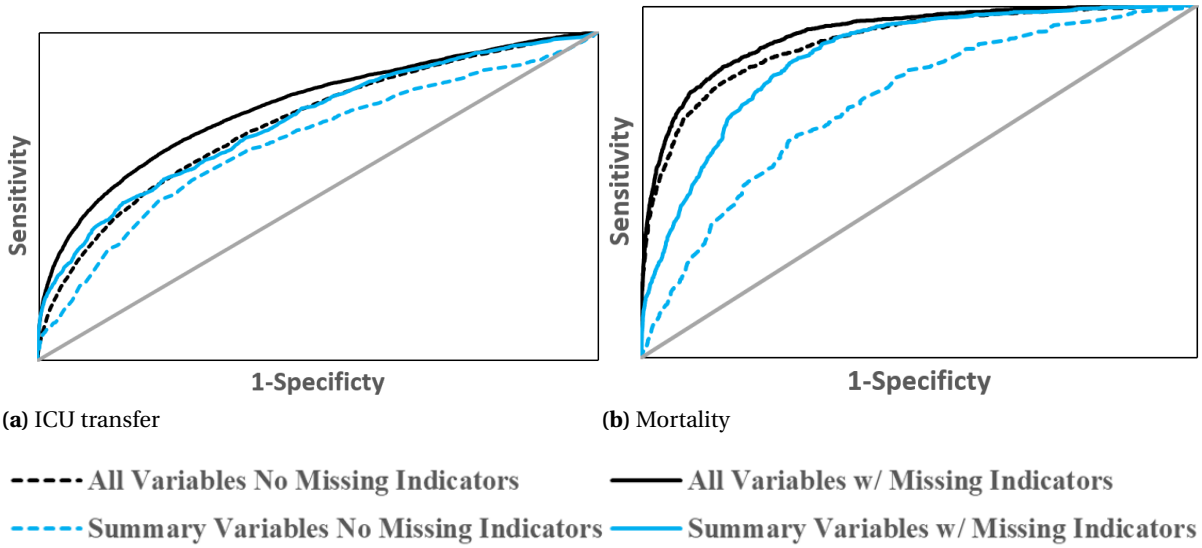


Figure A.1 Visual representation of correlation matrices for observations generated for the ICU Transfer outcome (left) and Mortality outcome (right). Kendall's rank correlation coefficient is reported [118]. "Variable_I" represents the missing indicator for the corresponding variable



(a) ICU transfer **(b)** Mortality

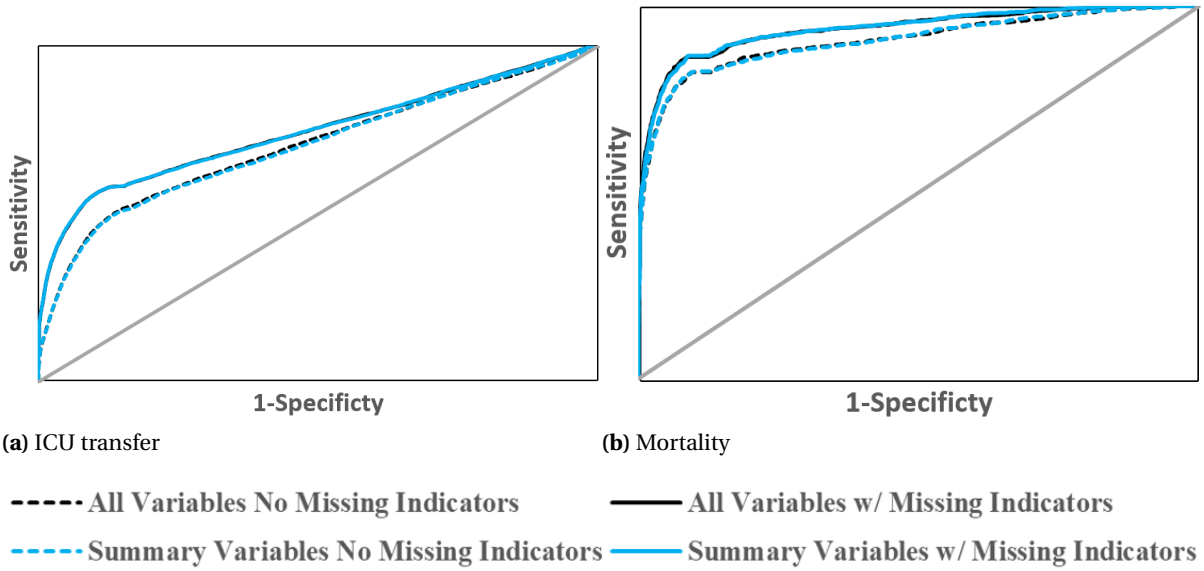
Figure A.2 Receiver Operator Curves created by random forest models for the (a) ICU transfer and (b) mortality outcomes



(a) ICU transfer **(b)** Mortality

All Variables No Missing Indicators
All Variables w/ Missing Indicators
Summary Variables No Missing Indicators
Summary Variables w/ Missing Indicators

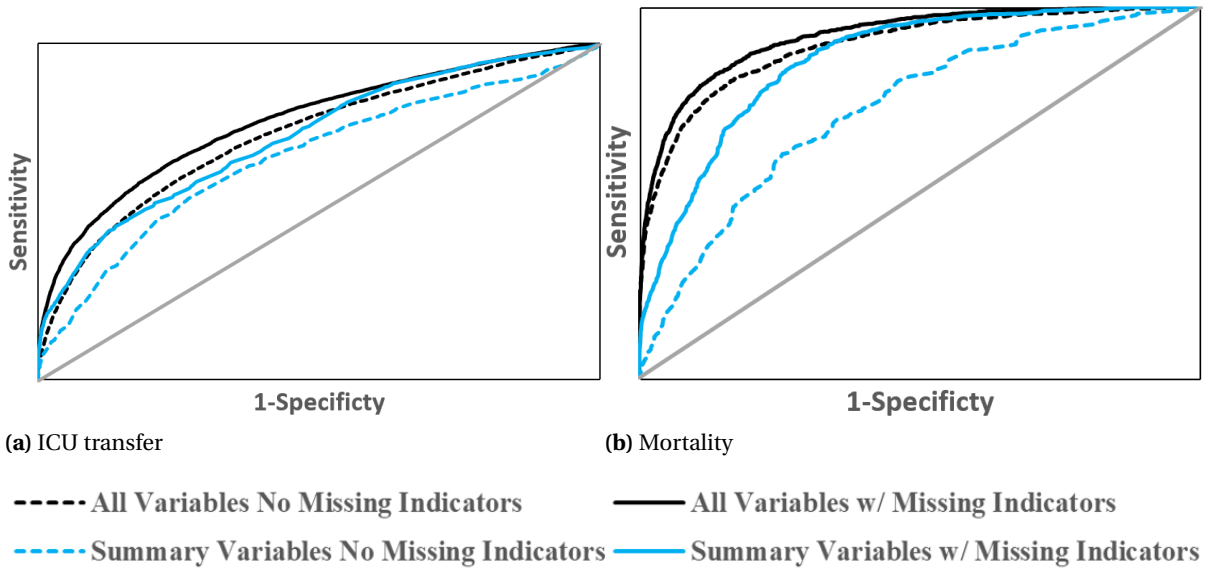
Figure A.3 Receiver Operator Curves created by stepwise regression models for the (a) ICU transfer and (b) mortality outcomes



(a) ICU transfer **(b)** Mortality

All Variables No Missing Indicators
All Variables w/ Missing Indicators
Summary Variables No Missing Indicators
Summary Variables w/ Missing Indicators

Figure A.4 Receiver Operator Curves created by SVM models for the (a) ICU transfer and (b) mortality outcomes



(a) ICU transfer **(b)** Mortality

Figure A.5 Receiver Operator Curves created by LASSO models for the (a) ICU transfer and (b) mortality outcomes

APPENDIX

B

BILEVEL FEATURE SELECTION APPENDIX

B.1 Amino acid sequence alignment

GM1 partitioned the 20 amino acid residues into the three classes: non-polar, polar and charged. GM2 and GM4 further partitioned the non-polar class into aliphatic and aromatic classes, and the charged class into positively and negatively charged classes. GM5 and GM6 are based on side chains of amino acid residues [66].

B.2 Steps of the genetic algorithm

We present individual steps of the GA implementation below.

Step 0 *Specify parameters for the genetic algorithm.*
- Specify minimum number of critical features that needs to be selected on average α (for

Table B.1 Groupings for amino acid sequence alignment [136]

GM0:	no grouping, i.e., each amino acid residue is a group by itself
GM1:	{non-polar: A,F,G,I,L,M,P,V,W}, {polar: C,N,Q,S,T,Y}, {charged: D,E,H,K,R}
GM2:	{non-polar aliphatic: A,G,I,L,M,V}, {non-polar aromatic: F,M,O,M,W}, {polar: C,N,Q,S,T,Y}, {charged: D,E,H,K,R}
GM3:	{non-polar: A,F,G,I,L,M,P,V,W}, {polar: C,N,Q,S,T,Y}, {positively charged: H,K,R}, {negatively charged: D,E}
GM4:	{non-polar aliphatic: A,G,I,L,M,V}, {non-polar aromatic: F,P,W}, {polar: C,N,Q,S,T,Y}, {positively charged: H,K,R}, {negatively charged: D,E}
GM5:	{non-polar aliphatic: A,I,L,M,P,V}, {non-polar aromatic: F,W,Y}, {polar: N,Q,S,T}, {positively charged: H,K,R}, {negatively charged: D,E}, {C}, {G}
GM6:	{non-polar aliphatic: A,I,L,M,P,V}, {non-polar aromatic: F,W,Y}, {polar: N,Q,S,T}, {charged: D,E,H,K,R}, {C}, {G}

influenza case study)

- Specify maximum number of feature positions that can be selected β
- Specify initial population level P_{initial}
- Set probability of crossover and mutation, P_c and P_m, respectively
- Set max number of generations $G_{\max}$
- Specify weight vector $\boldsymbol{w}$
- Set desired level of performance $\boldsymbol{L} \in \mathbb{R}^6$ (SVM results from Liao et al. [136] in Table 4.5 for all case studies)
- Let S be the population of chromosomes, initialize $S \leftarrow \emptyset$
- Let E be the set of already explored chromosomes, initialize $E \leftarrow \emptyset$
- Let N_D be the set of non-dominated chromosomes/solutions, initialize $N_D \leftarrow \emptyset$

Step 1 *Generate the initial population.*

(a) If $|S| = P_{\text{initial}}$, go to Step 2. Else, randomly create a chromosome (or a feature selection vector) $\boldsymbol{u} \in \{0,1\}^n$, where $\sum_{i=1}^{n} u_i = \beta$.

(b) If $\boldsymbol{u} \in E$, go to Step 1a. Else, if $\boldsymbol{u}$ upper-level infeasible, $E \leftarrow E \cup \{\boldsymbol{u}\}$ and go to Step 1a.

(c) Solve the lower-level training problem. Use solution to obtain vector $\boldsymbol{v}$ of performance measures associated with this chromosome and calculate fitness value $F(\boldsymbol{u})$ =# misclassified observations in out-of-sample validation set Ω_{V}. $S \leftarrow S \cup \{\boldsymbol{u}\}$, $E \leftarrow E \cup \{\boldsymbol{u}\}$ and go to Step 1a.

Step 2 *Set generation number* $g \leftarrow 1$. *Iterate through generations.*

(a) Remove non-dominated solutions from S, and store them in N_D. If $g = G_{\max}$, report the solutions with high fitness values in N_D and **STOP**.

(b) Set $s \leftarrow \lfloor |S|/2 \rfloor$ and $K \leftarrow \emptyset$. Perform `tournament selection`.

i. If $|K| = s$, then $S \leftarrow K \cup N_D$ and go to Step 2c.

ii. Randomly select two solutions $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ from S without replacement. Calculate fitness values $F(\boldsymbol{u}_1)$ and $F(\boldsymbol{u}_2)$. If $F(\boldsymbol{u}_1) \geq F(\boldsymbol{u}_2)$, then $K \leftarrow K \cup \{\boldsymbol{u}_1\}$. Else, $K \leftarrow K \cup \{\boldsymbol{u}_2\}$. Go to Step 2b(i).

(c) Set $N \leftarrow \emptyset$ (new chromosomes). Perform crossover and mutation.

i. If no more available pairs of chromosomes to crossover that have not been crossed-over yet, then go to Step 2c(iii). Otherwise, with probability P_c, take the next two available chromosomes in S that have not been attempted to be crossed over and perform `Crossover` to generate two new children $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. (see Appendix Table B.2)

ii. For $j = 1, 2$, if $\boldsymbol{u}_j \notin E$, then $N \leftarrow N \cup \{\boldsymbol{u}_j\}$, $E \leftarrow E \cup \{\boldsymbol{u}_j\}$ Go to Step 2c(i).

iii. If no more available chromosomes to mutate that have not been mutated yet in S, then go to Step 2c(v). Otherwise, with probability P_m, take the next available chromosome in S that has not been attempted to be mutated yet and perform `Mutation` to generate a new chromosome $\boldsymbol{u}_3$. (see Appendix Table B.2)

iv. If $\boldsymbol{u}_3 \notin E$, then $N \leftarrow N \cup \{\boldsymbol{u}_3\}$, $E \leftarrow E \cup \{\boldsymbol{u}_3\}$. Go to Step 2c(iii).

v. For each chromosome $\boldsymbol{u} \in N$, test for feasibility at the upper level (e.g. that it satisfies $\sum_{i \in \Omega} \boldsymbol{u}^T \boldsymbol{d}_i \geq \alpha |\Omega|$ in influenza case study). If feasible, then compute fitness value and $S \leftarrow S \cup \{\boldsymbol{u}\}$. Set $g \leftarrow g + 1$ and go to Step 2a.

B.3 Additional results

Table B.2 `Crossover` and `Mutation` procedures implemented in the GA

`Crossover`	`Mutation`
1) Take two parent chromosomes $\boldsymbol{u}^1$ and $\boldsymbol{u}^2$. Let $\Gamma^1 = \{j = 1,\ldots,n \mid u_j^1 - u_j^2 = 1\}$ and $\Gamma^2 = \{j = 1,\ldots,n \mid u_j^2 - u_j^1 = 1\}$. Randomly permute indices in Γ^1 and Γ^2.	**1)** Move through a chromosome and with probability P_m, change the element of the chromosome (i.e. from 1 to 0 or 0 to 1)
2) Go to first element in Γ^1 and swap values at that index. Go to first element in Γ^2 and swap values at that index. Continue in this fashion alternating between Γ^1 and Γ^2 until exhausting one of the sets.	**2)** If the sum of the elements in the chromosome is not equal to β, randomly delete or add 1's to make the sum β.
3) The two parents are now the two new children to be tested.	

Table B.3 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	35	92.8	96.0	85.7	80.2	94.7	76.6
LASSO	56	92.8	96.0	85.7	67.7	47.4	72.7
NOMAD SVM**	35	92.8	96.8	83.9	80.2	89.5	77.9
NOMAD LASSO**	39	91.2	94.4	83.9	61.5	31.6	68.8
GA SVM**	20	93.4	96.0	87.5	81.3	89.5	79.2
GA LASSO**	19	94.5	96.0	91.1	81.3	89.5	79.2

*The alignment method GM2 is used. Agreement, sensitivity and specificity are in percentages.

*SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.4 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	25	93.9	96.0	89.3	80.2	94.7	76.6
LASSO	51	93.9	96.0	89.3	76.0	100.0	70.1
NOMAD SVM**	25	92.3	96.0	83.9	75.0	100.0	68.8
NOMAD LASSO**	38	93.9	96.0	89.3	76.0	100.0	70.1
GA SVM**	16	93.4	94.4	91.1	82.3	94.7	79.2
GA LASSO**	23	94.5	96.0	91.1	87.5	94.7	85.7

*The alignment method GM3 is used. Agreement, sensitivity and specificity are in percentages.

*SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.5 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	35	93.4	95.2	89.3	79.2	84.2	77.9
LASSO	53	93.9	96.0	89.3	81.3	89.5	79.2
NOMAD SVM**	35	93.4	95.2	89.3	80.2	89.5	77.9
NOMAD LASSO**	37	89.0	92.8	80.4	66.7	42.1	72.7
GA SVM**	20	93.4	94.4	91.1	81.3	89.5	79.2
GA LASSO**	22	94.5	96.0	91.1	86.5	89.5	85.7

*The alignment method GM4 is used. Agreement, sensitivity and specificity are in percentages.

*SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.6 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	40	93.4	95.2	89.3	80.2	94.7	76.6
LASSO	48	92.8	96.0	91.1	81.3	89.5	79.2
NOMAD SVM**	40	94.5	96.0	91.1	82.3	89.5	80.5
NOMAD LASSO**	36	94.5	96.0	91.1	81.3	89.5	79.2
GA SVM**	15	93.4	94.4	91.1	81.3	89.5	79.2
GA LASSO**	20	94.5	96.0	91.1	81.3	89.5	79.2

*The alignment method GM5 is used. Agreement, sensitivity and specificity are in percentages.

*SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.7 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	28	93.9	95.2	89.3	80.2	94.7	76.6
LASSO	46	92.8	96.0	85.7	72.9	42.1	80.5
NOMAD SVM**	25	90.1	98.4	71.4	70.8	94.7	64.9
NOMAD LASSO**	36	93.4	97.6	83.9	67.7	31.6	76.6
GA SVM**	14	92.8	94.4	89.3	81.3	89.5	79.2
GA LASSO**	14	93.4	95.2	89.3	81.3	89.5	79.2

*The alignment method GM6 is used. Agreement, sensitivity and specificity are in percentages.

*SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.8 Results for influenza A virus classification case study*

		Training Set			Validation Set		
Grouping	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
SVM	45	93.9	96.8	87.5	78.1	100.0	72.7
LASSO	55	92.8	96.0	85.7	68.8	100.0	61.0
NOMAD SVM**	40	90.6	94.4	82.1	80.2	100.0	75.3
NOMAD LASSO**	39	93.4	96.8	85.7	60.4	89.5	53.2
GA SVM**	19	91.2	91.2	91.1	78.1	100.0	72.7
GA LASSO**	20	94.5	96.0	91.1	79.2	100.0	74.0

*Non-grouping alignment method is used. Agreement, sensitivity and specificity are in percentages.

*SVM models solved using R package `e1071` [115]. All LASSO models solved using R package `glmnet` [73].

**Results from the corresponding bilevel models.

Table B.9 Sensitivity of β parameter in influenza virus classification*

			Training Set			Validation Set		
Method	β	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
GA SVM	35	11	91.2	93.6	85.7	82.3	94.7	79.2
	36	12	91.2	93.6	85.7	82.3	94.7	79.2
	37	11	91.2	93.6	85.7	82.3	94.7	79.2
	38	13	91.2	93.6	85.7	82.3	94.7	79.2
	39	14	91.2	92.8	87.5	82.3	94.7	79.2
	40	14	92.3	94.4	87.5	82.3	94.7	79.2
GA LASSO	35	15	90.6	92.8	85.7	89.6	84.2	90.9
	36	12	87.3	84.8	92.9	93.8	84.2	96.1
	37	15	86.7	84.8	91.1	93.8	84.2	96.1
	38	15	89.0	91.2	83.9	93.8	84.2	96.1
	39	17	87.8	88.8	85.7	93.8	84.2	96.1
	40	19	93.4	96.0	87.5	87.5	94.7	85.7
NOMAD SVM	35	35	92.8	96.8	83.9	79.2	94.7	80.0
	36	31	90.6	97.6	75.0	68.8	100.0	61.0
	37	25	92.3	96.0	83.9	72.9	100.0	66.2
	38	31	90.6	97.6	75.0	68.8	100.0	61.0
	39	33	80.7	100.0	37.5	19.8	100.0	0.0
	40	35	92.8	96.8	83.9	79.2	94.7	80.0
NOMAD LASSO	35	33	90.6	93.6	83.9	89.6	84.2	90.9
	36	35	93.4	96.0	87.5	72.9	94.7	67.5
	37	30	85.1	96.0	60.7	25.0	31.6	23.4
	38	36	91.7	94.4	85.7	77.1	94.7	72.7
	39	36	89.5	92.0	83.9	87.5	89.5	87.0
	40	38	93.4	96.0	87.5	87.5	94.7	85.7

*The alignment method GM1 is used. Agreement, sensitivity and specificity are in percentages. SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

Table B.10 Sensitivity of β parameter in colposcopy image quality identification*

			Training Set			Validation Set		
Method	β	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
GA SVM	15	15	96.9	100.0	80.0	87.9	92.9	60.0
	16	16	96.9	100.0	80.0	87.9	92.9	60.0
	17	17	95.3	98.1	80.0	93.9	96.4	80.0
	18	18	96.9	100.0	80.0	87.9	89.3	80.0
	19	19	95.3	98.1	80.0	93.9	96.4	80.0
	20	17	95.3	98.1	80.0	93.9	96.4	80.0
GA LASSO	15	15	96.9	100.0	80.0	93.9	100.0	60.0
	16	16	95.3	100.0	70.0	93.9	96.4	80.0
	17	17	100.0	100.0	100.0	84.8	85.7	80.0
	18	18	100.0	100.0	100.0	84.8	85.7	80.0
	19	19	100.0	100.0	100.0	81.8	82.1	80.0
	20	18	100.0	100.0	100.0	84.8	85.7	80.0
NOMAD SVM	15	15	85.9	100.0	10.0	84.8	100.0	0.0
	16	16	90.6	98.1	50.0	87.9	96.4	40.0
	17	17	92.2	98.1	60.0	90.9	92.9	80.0
	18	18	92.2	100.0	50.0	90.9	96.4	60.0
	19	19	90.6	100.0	40.0	90.9	100.0	40.0
	20	19	85.9	98.1	20.0	90.9	96.4	60.0
NOMAD LASSO	15	15	90.6	98.1	50.0	78.8	85.7	40.0
	16	16	93.8	98.1	70.0	75.8	82.1	40.0
	17	16	87.5	98.1	30.0	87.9	96.4	40.0
	18	18	92.2	96.3	70.0	78.8	89.3	20.0
	19	18	87.5	96.3	40.0	90.9	100.0	40.0
	20	20	100.0	100.0	100.0	87.9	92.9	60.0

*Agreement, sensitivity and specificity are in percentages. SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

Table B.11 Sensitivity of β parameter in splice junction recognition*

			Training Set			Validation Set		
Method	β	# Features	Agreement	Sensitivity	Specificity	Agreement	Sensitivity	Specificity
	15	15	99.2	99.4	98.7	97.2	97.8	97.0
	16	16	99.2	99.4	98.7	97.8	97.8	97.8
GA	17	17	99.3	99.4	99.0	97.7	98.3	97.4
SVM	18	18	99.5	99.8	99.0	97.4	97.8	97.2
	19	19	99.8	100.0	99.4	97.4	97.8	97.2
	20	20	100.0	100.0	100.0	97.2	96.1	97.8
	15	15	97.0	98.3	94.8	93.9	96.5	92.6
	16	16	97.2	97.8	96.1	94.9	96.5	94.1
GA	17	17	97.2	98.5	94.8	94.6	95.2	94.3
LASSO	18	18	99.1	99.4	98.4	96.1	94.8	96.7
	19	19	99.1	99.4	98.4	94.1	94.8	93.7
	20	19	100.0	100.0	100.0	94.5	97.0	93.3
	15	15	94.8	98.7	88.0	92.5	97.4	90.0
	16	16	89.5	95.3	79.2	85.1	93.5	80.9
NOMAD	17	17	98.0	98.7	96.8	94.9	95.7	94.6
SVM	18	18	92.3	94.8	88.0	85.2	86.5	84.6
	19	19	95.7	96.6	94.2	93.6	96.1	92.4
	20	20	96.8	98.1	94.5	93.6	95.2	92.8
	15	15	95.0	98.0	89.9	92.8	96.5	90.9
	16	16	90.1	95.5	80.5	85.5	92.2	82.2
NOMAD	17	17	97.2	98.1	95.5	94.9	94.8	95.0
LASSO	18	18	85.2	90.9	75.3	81.3	88.7	77.6
	19	18	91.0	93.3	87.0	86.2	88.7	85.0
	20	20	96.1	97.0	94.5	91.9	94.3	90.7

*Agreement, sensitivity and specificity are in percentages. SVM models are solved using R package `e1071` [115]. LASSO models are solved using R package `glmnet` [73].

APPENDIX

C

MIXED-INTEGER PROGRAMMING FOR SCORE DEVELOPMENT APPENDIX

C.1 Algorithm Details

Algorithm 1: ADMM with Coordinate Decent

Input: $\Omega^{\mathrm{T}}, M, \epsilon_1, \epsilon_2, \rho, P, \gamma, c_t, c_{\mathrm{it}}, c_\epsilon$, L and `maxIt`

begin

  Initialization

  Create a sequence of score intervals $\{[l_s, u_s]\}_{s=1}^{S}$ using interval length L

  Partition Ω^{T} into R subsets $\left\{\Omega_r^{\mathrm{T}}\right\}_{r=1}^{R}$ consisting of M observations in each subset

  Initialize two sequences of zero vectors $\left\{\boldsymbol{\alpha}_0^r\right\}_{r=1}^{R}$ and $\left\{\boldsymbol{\phi}_0^r\right\}_{r=1}^{R}$

  $f' \longleftarrow \mathbf{0}, \quad f^* \longleftarrow -\infty, \quad \upsilon \longleftarrow 0, \quad \epsilon_1^{\mathrm{primal}}, \epsilon_2^{\mathrm{primal}} \longleftarrow \infty \quad \epsilon_1^{\mathrm{dual}}, \epsilon_2^{\mathrm{dual}} \longleftarrow \infty$

  Warm Start Solution Generation

  for $r = 1 \rightarrow R$ **do**

    $\boldsymbol{\lambda}_\upsilon^r \longleftarrow \texttt{warmStart}\left(\Omega_r^{\mathrm{T}}, \boldsymbol{l}, \boldsymbol{u}, \gamma, c_t, c_{\mathrm{it}}, c_\epsilon\right)$

    $[\boldsymbol{Q}', f^1] \longleftarrow \texttt{qSolve}\left(\boldsymbol{\lambda}_\upsilon^r, \Omega^{\mathrm{T}}, \boldsymbol{l}, \boldsymbol{u}\right)$

    $f_r' \longleftarrow f^1$

    if $f^1 > f^*$ **then**

      $f^* \longleftarrow f^1, \quad \boldsymbol{Q}^* \longleftarrow \boldsymbol{Q}', \quad \boldsymbol{\lambda}^* \longleftarrow \boldsymbol{\lambda}_\upsilon^r$

    end

  end

  Set Initial Consensus

  $\boldsymbol{\zeta}_\upsilon \longleftarrow \underset{\boldsymbol{\zeta}}{\operatorname{argmin}} \left\{ \gamma \sum_{j=1}^{N} \zeta_j^2 - \sum_{r=1}^{R} \left(\langle \boldsymbol{\alpha}_\upsilon^R, \boldsymbol{\zeta} \rangle - (\rho/2) \|\boldsymbol{\lambda}_\upsilon^r - \boldsymbol{\zeta}\|_2^2 \right) \right\}$

  $\hat{\boldsymbol{Q}}_\upsilon \longleftarrow \underset{\hat{\boldsymbol{Q}}}{\operatorname{argmin}} \left\{ \sum_{r=1}^{R} (\rho/2) \|\boldsymbol{Q}_\upsilon^r - \hat{\boldsymbol{Q}}\|_2^2 \right\}$

  Main Algorithm

  while $\epsilon_1^{primal} > \epsilon_1$ **and** $\epsilon_2^{primal} > \epsilon_1$ **and** $\epsilon_1^{dual} > \epsilon_2$ **and** $\epsilon_2^{dual} > \epsilon_2$ **and** $\upsilon \leq \texttt{maxIt}$ **do**

    for $r = 1 \rightarrow R$ **do**

      $\boldsymbol{\lambda}_{\mathrm{tmp}}^r \longleftarrow \underset{(\boldsymbol{\lambda}^r, \boldsymbol{Q}^r) \in \Lambda^r}{\operatorname{argmax}} \left\{ f^r - \left\langle [\boldsymbol{\alpha}_\upsilon^r, \boldsymbol{\phi}_\upsilon^r], [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}_\upsilon, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}_\upsilon] \right\rangle - (\rho/2) \left\| [\boldsymbol{\lambda}^r - \boldsymbol{\zeta}_\upsilon, \boldsymbol{Q}^r - \hat{\boldsymbol{Q}}_\upsilon] \right\|_2^2 \right\}$

      $\boldsymbol{\lambda}_{\upsilon+1}^r \longleftarrow \texttt{coordinateDecent}\left(\boldsymbol{\lambda}_{\mathrm{tmp}}^r, \Omega_r^{\mathrm{T}}, \boldsymbol{l}, \boldsymbol{u}, \gamma, c_t, c_{\mathrm{it}}, c_\epsilon\right)$

      $[\boldsymbol{Q}', f^1] \longleftarrow \texttt{qSolve}\left(\boldsymbol{\lambda}_{\upsilon+1}^r, \Omega^{\mathrm{T}}, \boldsymbol{l}, \boldsymbol{u}\right)$

      $f_r' \longleftarrow f^1$

      if $f^1 > f^*$ **then**

        $f^* \longleftarrow f^1, \quad \boldsymbol{Q}^* \longleftarrow \boldsymbol{Q}', \quad \boldsymbol{\lambda}^* \longleftarrow \boldsymbol{\lambda}_{\upsilon+1}^r$

      end

    end

    $\boldsymbol{\zeta}_{\upsilon+1} \longleftarrow \underset{\boldsymbol{\zeta}}{\operatorname{argmin}} \left\{ \gamma \sum_{j=1}^{N} \zeta_j^2 - \sum_{r=1}^{R} \left(\langle \boldsymbol{\alpha}_\upsilon^R, \boldsymbol{\zeta} \rangle - (\rho/2) \|\boldsymbol{\lambda}_{\upsilon+1}^r - \boldsymbol{\zeta}_\upsilon\|_2^2 \right) \right\}$

    $\hat{\boldsymbol{Q}}_{\upsilon+1} \longleftarrow \underset{\hat{\boldsymbol{Q}}}{\operatorname{argmin}} \left\{ \sum_{r=1}^{R} (\rho/2) \|\boldsymbol{Q}_{\upsilon+1}^r - \hat{\boldsymbol{Q}}\|_2^2 \right\}$

    $\boldsymbol{\alpha}_{\upsilon+1}^r \longleftarrow \boldsymbol{\alpha}_\upsilon^r + \rho\left(\boldsymbol{\lambda}_{\upsilon+1}^r - \boldsymbol{\zeta}_{\upsilon+1}\right), \quad \boldsymbol{\phi}_{\upsilon+1}^r \longleftarrow \boldsymbol{\phi}_\upsilon^r + \rho\left(\boldsymbol{Q}_{\upsilon+1}^r - \hat{\boldsymbol{Q}}_{\upsilon+1}\right) \quad \text{for } r = 1 \ldots R$

    $\epsilon_1^{\mathrm{primal}} \longleftarrow \frac{1}{R} \sum_{r=1}^{R} \|\boldsymbol{\lambda}_{\upsilon+1}^r - \boldsymbol{\zeta}_{\upsilon+1}\|_2^2, \quad \epsilon_2^{\mathrm{primal}} \longleftarrow \frac{1}{R} \sum_{r=1}^{R} \|\boldsymbol{Q}_{\upsilon+1}^r - \hat{\boldsymbol{Q}}_{\upsilon+1}\|_2^2$

    $\epsilon_1^{\mathrm{dual}} \longleftarrow \|\boldsymbol{\zeta}_{\upsilon+1} - \boldsymbol{\zeta}_\upsilon\|_2^2, \quad \epsilon_2^{\mathrm{dual}} \longleftarrow \|\hat{\boldsymbol{Q}}_{\upsilon+1} - \hat{\boldsymbol{Q}}_\upsilon\|_2^2$

    $\upsilon \longleftarrow \upsilon + 1$

  end

end

Output: Optimal point values and risk (probabilities) $(\boldsymbol{\lambda}^*, \boldsymbol{Q}^*)$

Algorithm 2: `qSolve`$(\boldsymbol{\lambda}, \Omega, \boldsymbol{l}, \boldsymbol{u})$

Input:
Solution vector of points $\boldsymbol{\lambda}$
Set of observations Ω
Vectors of lower and upper bounds on score intervals $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^S$

```
begin
    Put Observations in Score Intervals
    for i ∈ Ω do
        score ⟵ ⟨λ, x_i⟩,   s ⟵ 1
        while s ≤ S do
            if score ∈ [l_s, u_s] then
                z'_{is} ⟵ 1,   z'_{ij} ⟵ 0 for each j ≠ s
                s ⟵ S + 1
            else
                s ⟵ s + 1
            end
        end
    end
    Find Optimal Q
    Solve the following linear program for optimal solution and objective value pair (Q*, f*):
```

$$
\begin{aligned}
f^* = \max_{\boldsymbol{Q}} \; & \sum_k \sum_s \sum_{i \in \Omega} y_{ik} z'_{is} q_{sk} \\
\text{s.t. } 0 \le \; & \sum_i z'_{is} (y_{ik} - q_{sk}) \quad \forall s, k \\
& \sum_k q_{sk} = 1 \qquad \forall s \\
& \boldsymbol{q}_s \ge \boldsymbol{0} \qquad \forall s
\end{aligned}
$$

```
end
```

Output: Risk and objective value pair $(\boldsymbol{Q}^*, f^*)$

Algorithm 3: `warmStart`$\left(\Omega, \boldsymbol{l}, \boldsymbol{u}, \gamma, c_t, c_{\text{it}}, c_\epsilon\right)$

Input:
Set of observations Ω
Vectors of lower and upper bounds on score intervals $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^S$
Sparsity penalty parameter γ
Termination criteria $(c_t, c_{\text{it}}, c_\epsilon)$ for `coordinateDecent` procedure

```
begin
    Solve problem (5.2) setting Ω^T = Ω and relaxing the binary variables z_is by letting them being
      continuous (i.e. z_is ∈ {0,1} → z_is ∈ [0,1]) to obtain solution λ'
    λ* ⟵ coordinateDecent(λ', Ω, l, u, γ, c_t, c_it, c_ε)
end
```

Output: Vector of point values for score $\boldsymbol{\lambda}^*$

Algorithm 4: `coordinateDecent`$\left(\boldsymbol{\lambda}, \Omega, \boldsymbol{l}, \boldsymbol{u}, \gamma, c_t, c_{\text{it}}, c_\epsilon\right)$

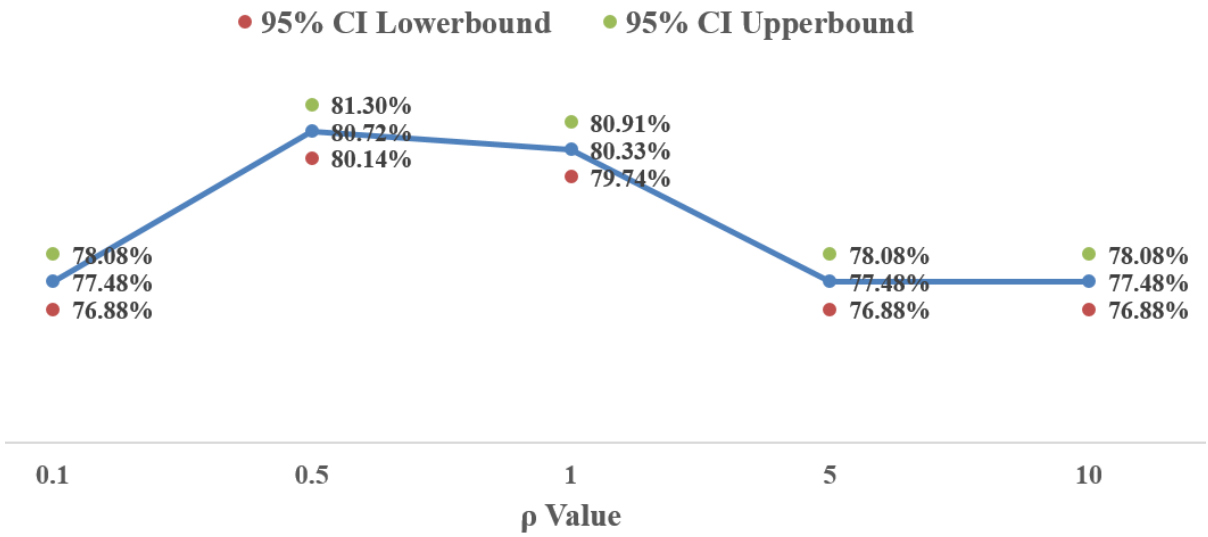
Input:
Solution vector of points $\boldsymbol{\lambda}$
Set of observations Ω
Vectors of lower and upper bounds on score intervals $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^S$
Sparsity penalty parameter γ
Termination criteria $(c_t, c_{\text{it}}, c_\epsilon)$

```
begin
    ε ⟵ ∞,   k ⟵ 1,   f* ⟵ −10^6
    while ε > c_ε and currentTime < c_t and k < c_it do
        for j = 1 → N do
            for i = 0 → P do
                λ_j ⟵ i
                [Q', f'] ⟵ qSolve(λ, Ω, l, u)
                if f' − γ Σ_{j=1}^N λ_j^2 > f* then
                    f* ⟵ f' − γ Σ_{j=1}^N λ_j,   Q* ⟵ Q',   λ* ⟵ λ
                end
            end
            reset λ_j to original value
        end
        ε ⟵ ||λ* − λ||_{l_1}
        λ ⟵ λ*
        k ⟵ k + 1
    end
end
```

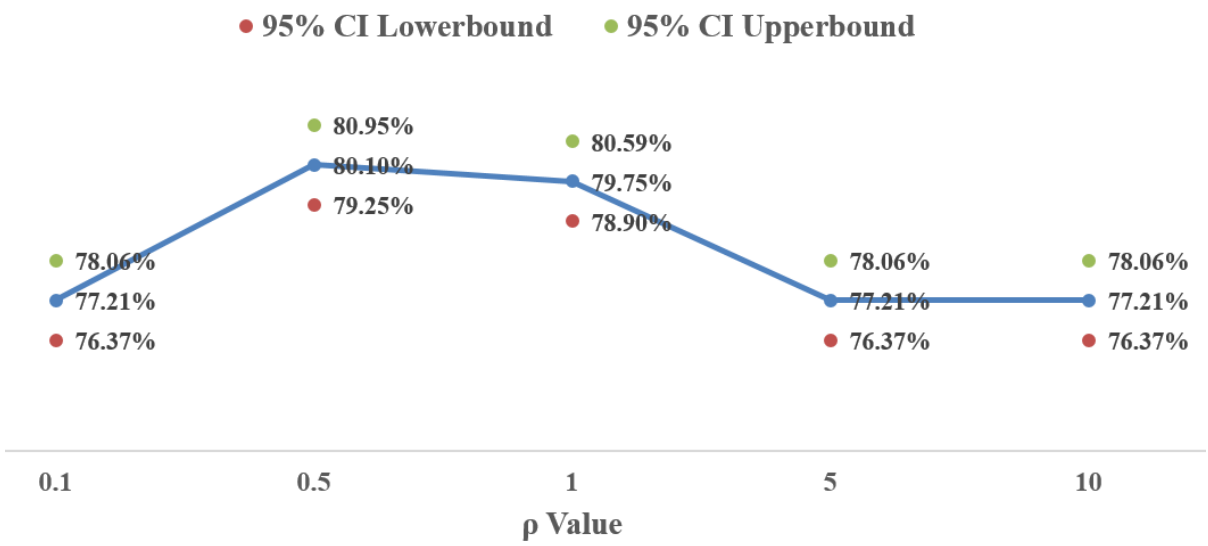
Output: Vector of point values for score $\boldsymbol{\lambda}^*$

C.2 `riskSlim` Parameters

C.3 Additional Sensitivity Analysis Results

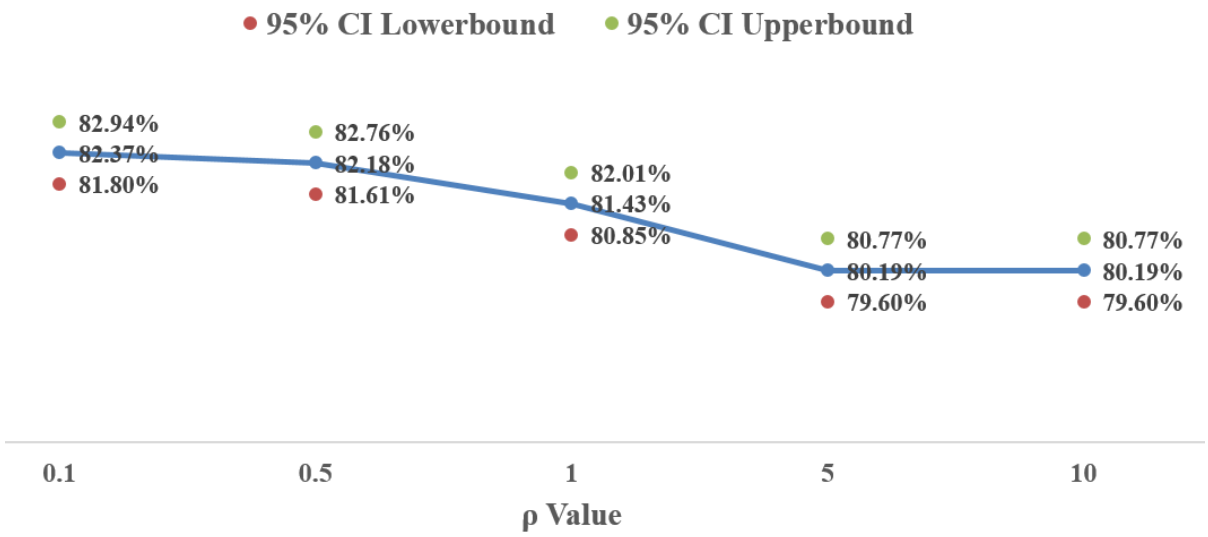


(a) Training set

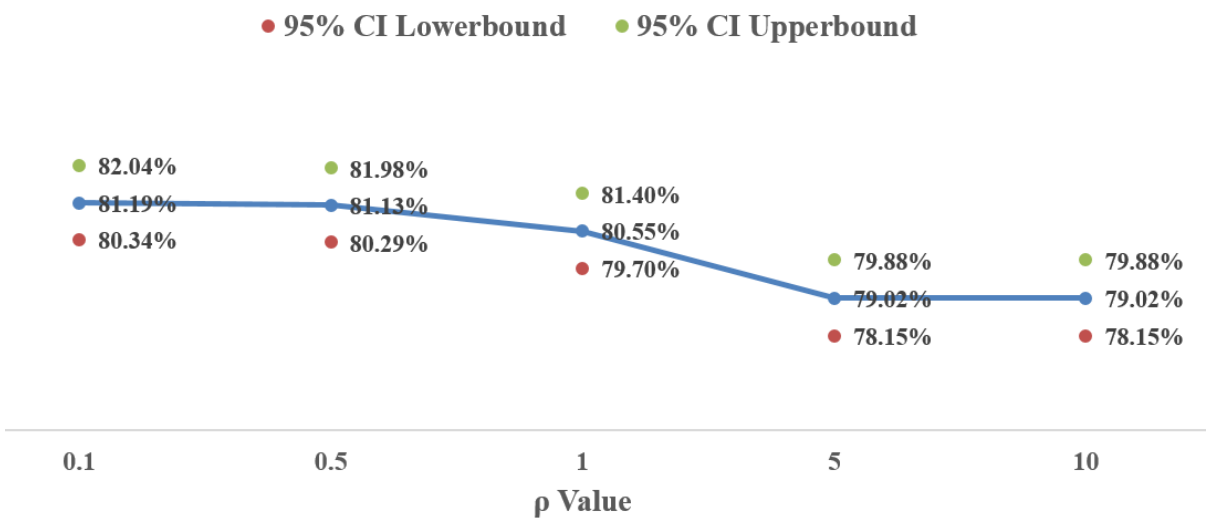


(b) Validation set

Figure C.1 Sensitivity of average probability of correct classification when $\gamma = 10$ and $\rho \in \{0.1, 0.5, 1, 5, 10\}$ in the (a) training set and (b) validation set

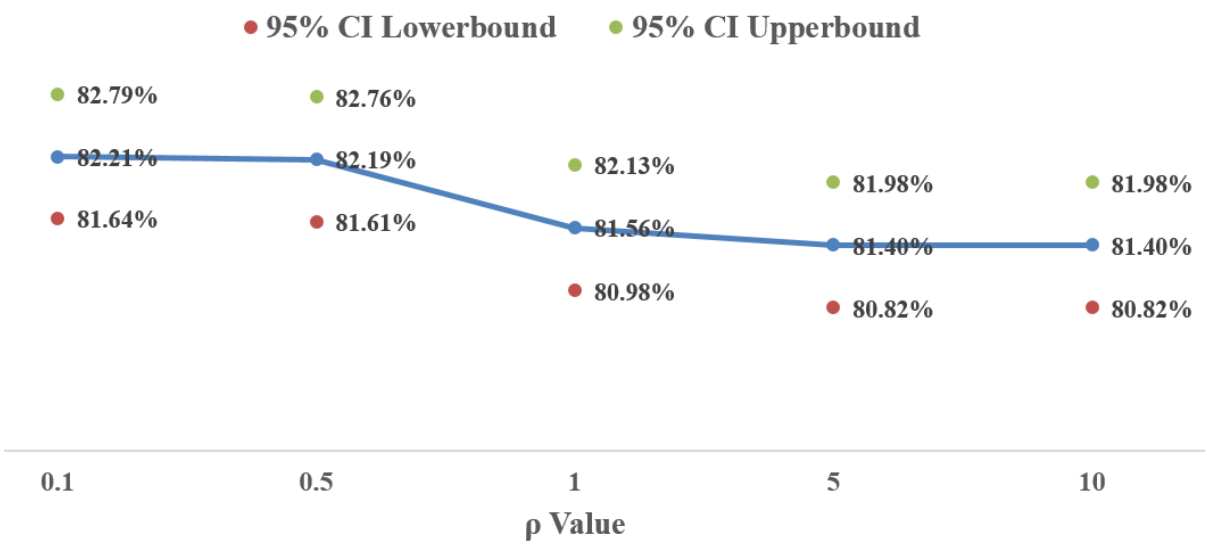


(a) Training set

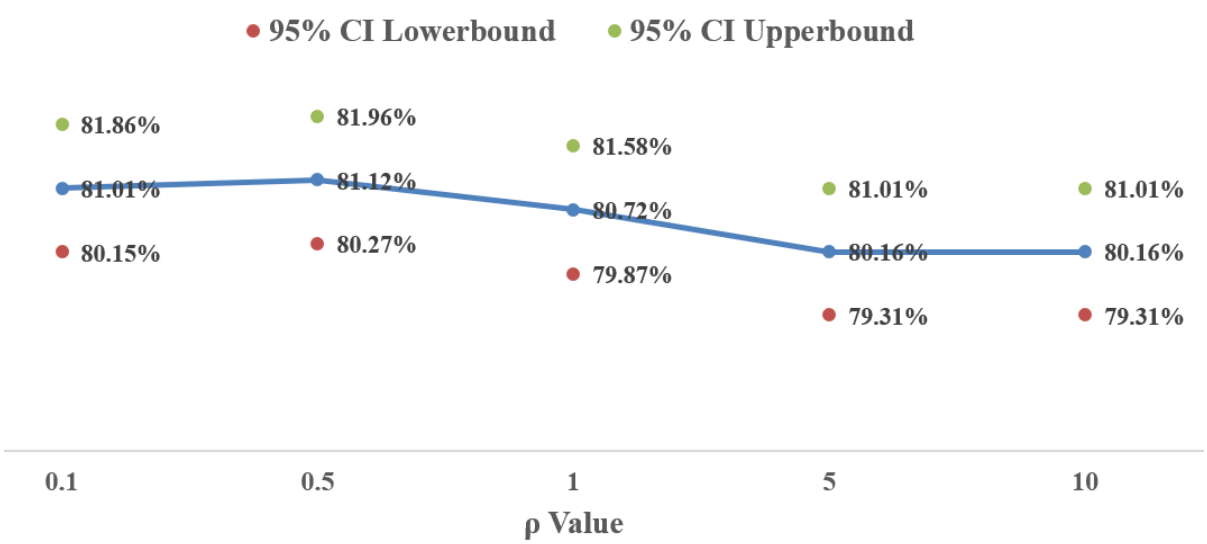


(b) Validation set

Figure C.2 Sensitivity of average probability of correct classification when $\gamma = {}^{1}/_{10}$ and $\rho \in \{0.1, 0.5, 1, 5, 10\}$ in the (a) training set and (b) validation set

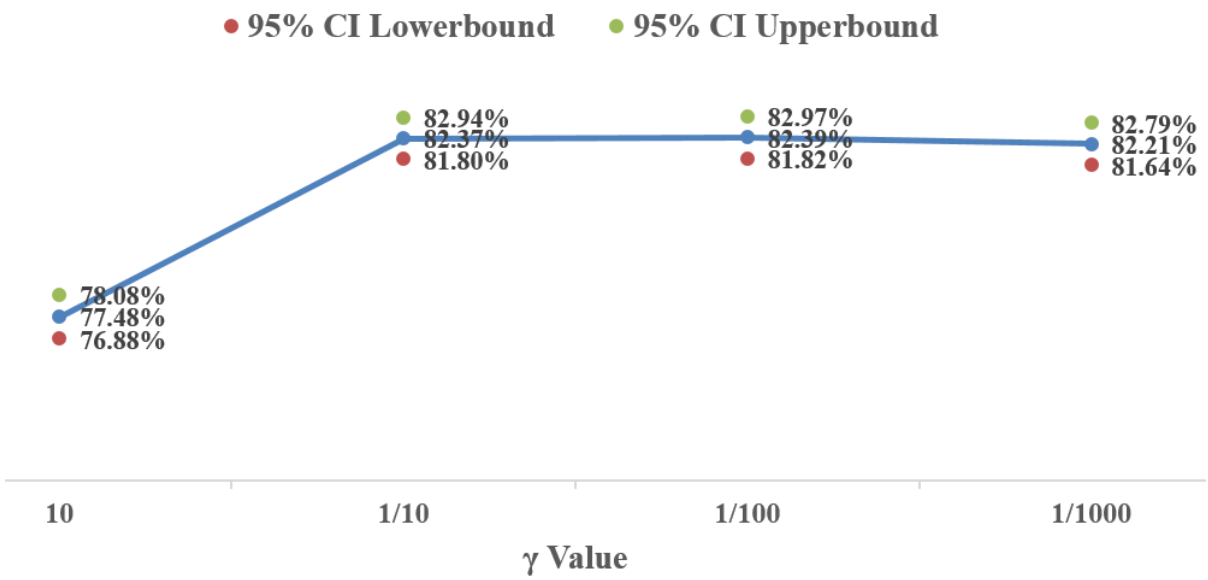


(a) Training set

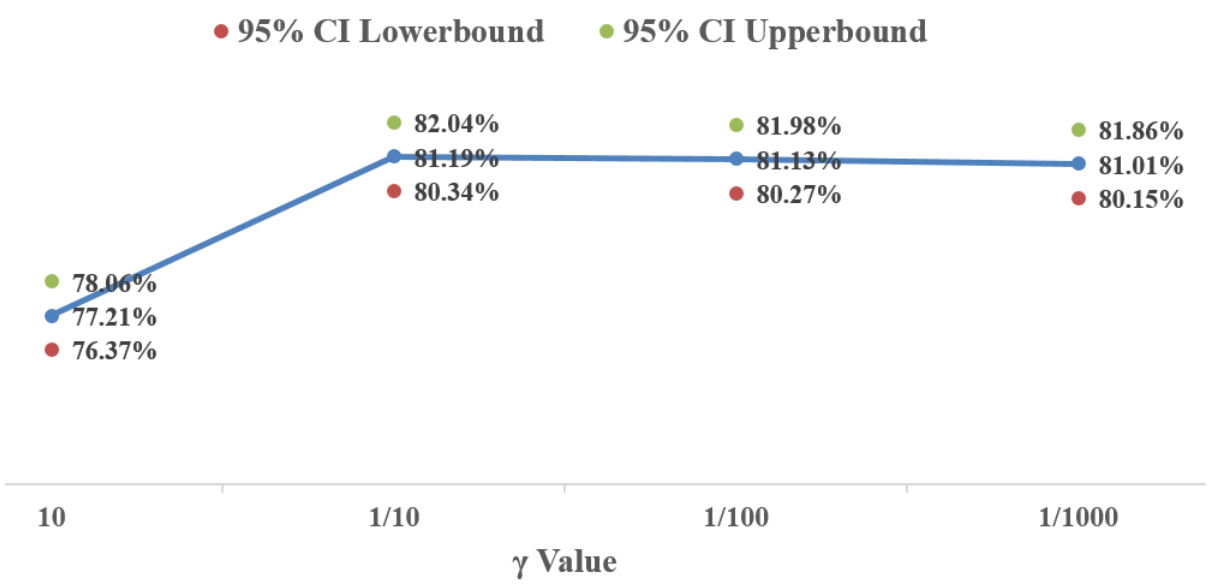


(b) Validation set

Figure C.3 Sensitivity of average probability of correct classification when $\gamma = {}^{1}/_{1000}$ and $\rho \in \{0.1, 0.5, 1, 5, 10\}$ in the (a) training set and (b) validation set

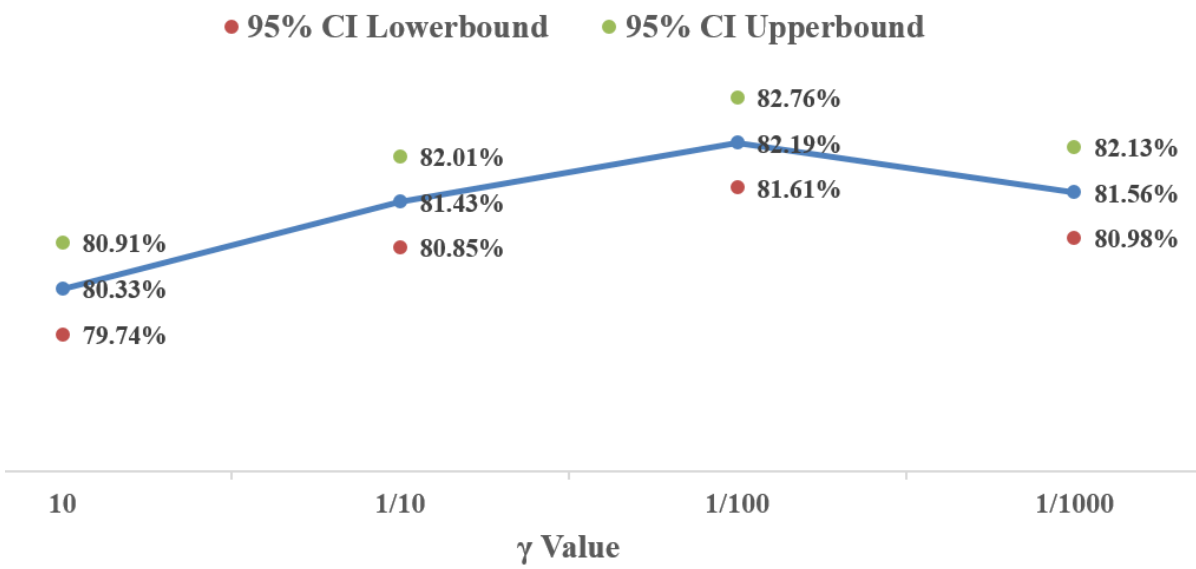


(a) Training set

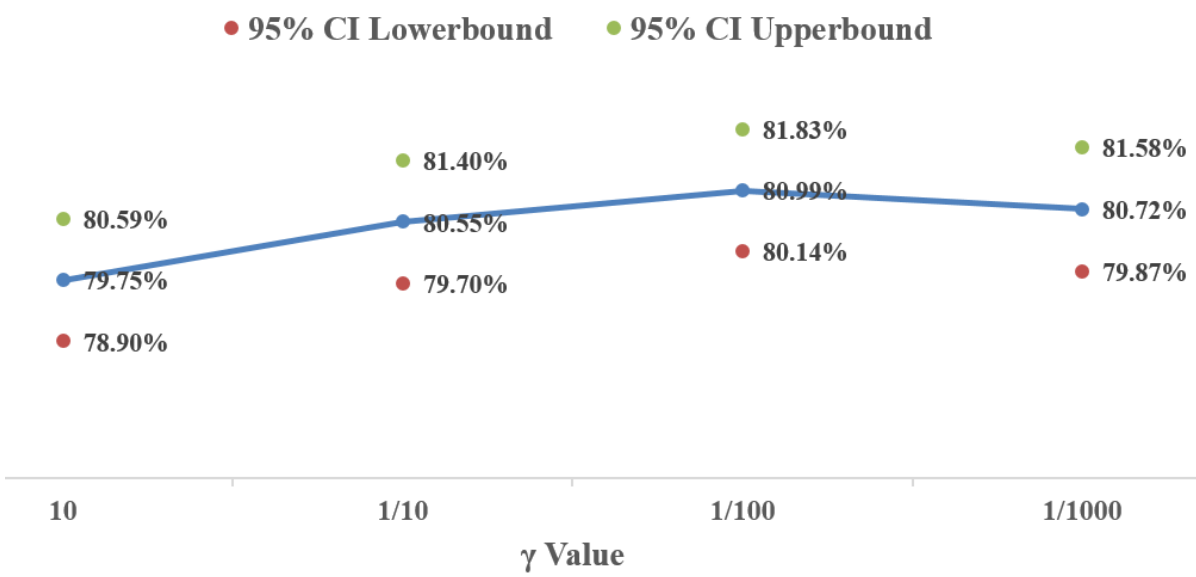


(b) Validation set

Figure C.4 Sensitivity of average probability of correct classification when $\rho = 0.1$ and $\gamma \in \{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ in the (a) training set and (b) validation set

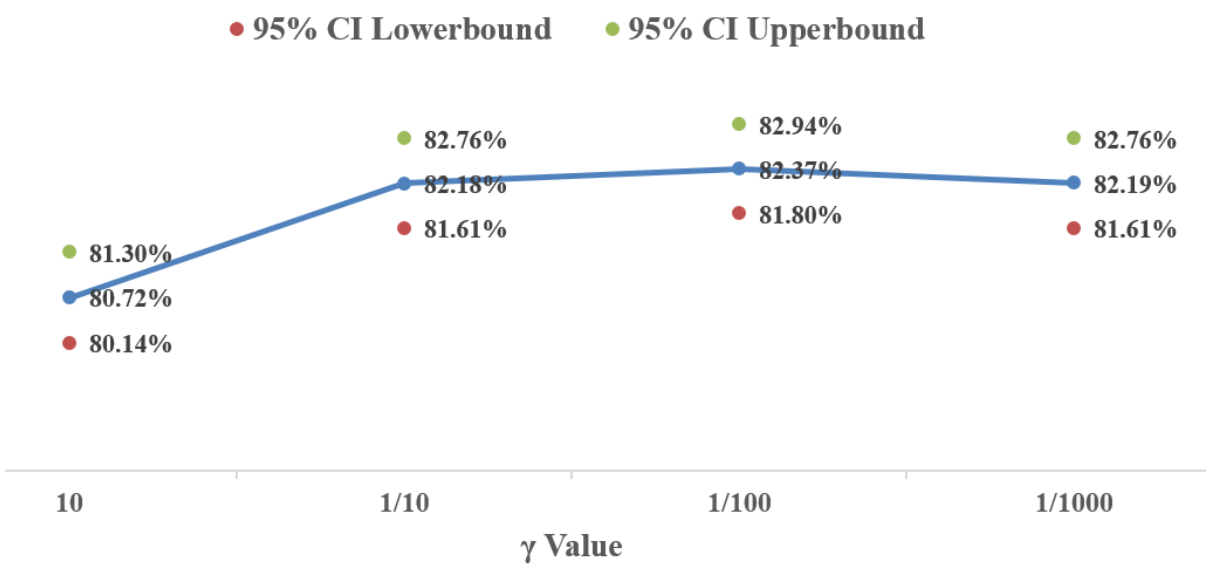


(a) Training set

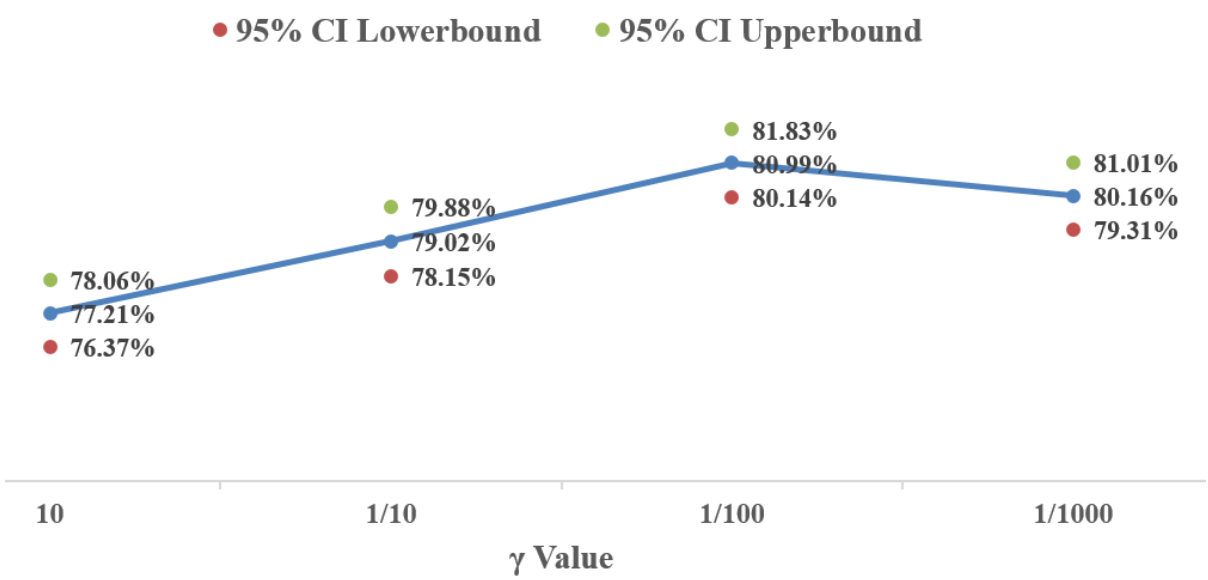


(b) Validation set

Figure C.5 Sensitivity of average probability of correct classification when $\rho = 1$ and $\gamma \in \{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ in the (a) training set and (b) validation set

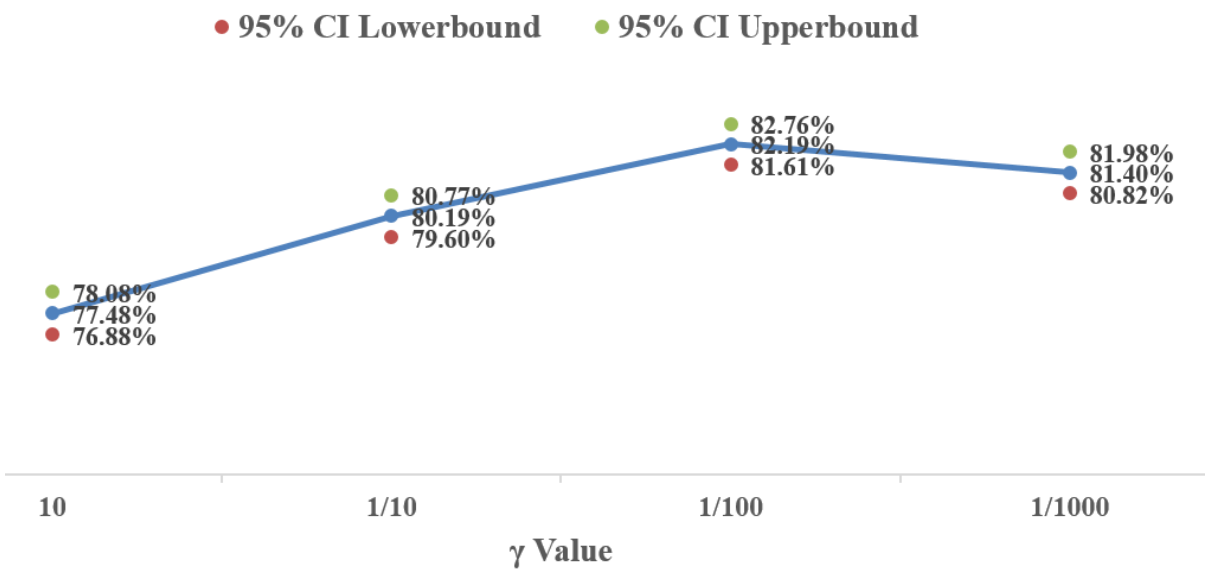


(a) Training set

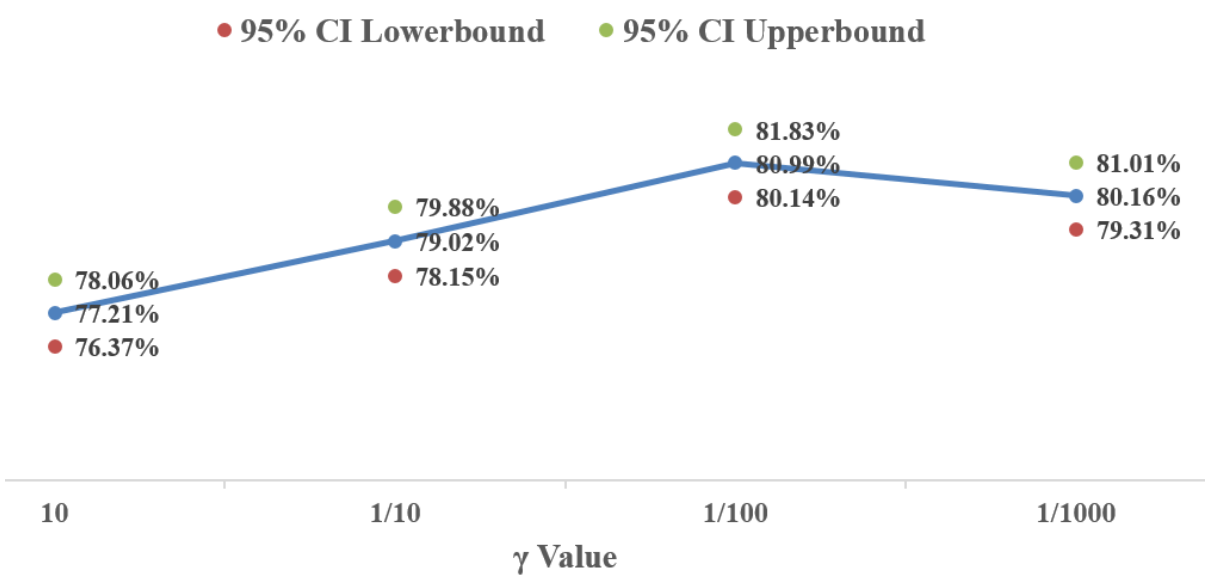


(b) Validation set

Figure C.6 Sensitivity of average probability of correct classification when $\rho = 5$ and $\gamma \in \{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ in the (a) training set and (b) validation set



(a) Training set



(b) Validation set

Figure C.7 Sensitivity of average probability of correct classification when $\rho = 10$ and $\gamma \in \{10, {}^{1}/_{10}, {}^{1}/_{100}, {}^{1}/_{1000}\}$ in the (a) training set and (b) validation set

Table C.1 `riskSlim` Parameters*

Category	Parameter Name	Value Used	Description
LCPA Settings	max_runtime	6050	Max runtime for LCPA
	max_tolerance	10^{-6}	Tolerance to stop LCPA
	display_cplex_progress	False	Set to 'True' to print CPLEX
	loss_computation	'lookup'	How to compute the loss
	chained_updates_flag	True	Use chained updates
	add_cuts_at_heuristic_solutions	True	Add cuts at integer feasible
Rounding Heuristic	round_flag	True	Round continuous solutions with `SeqRd`
	polish_rounded_solutions	True	Polish solutions rounded with `SeqRd` using DCD
	rounding_tolerance	∞	Only solutions with objective value $< (1 + \text{tol})$ are rounded
	rounding_start_cuts	0	Cuts needed to start using rounding heuristic
	rounding_start_gap	∞	Optimality gap needed to start using rounding heuristic
	rounding_stop_cuts	20000	Cuts needed to stop using rounding heuristic
	rounding_stop_gap	0.2	Optimality gap needed to stop using rounding heuristic
Polishing Heuristic	polish_flag	False	Polish integer feasible solutions with
	polishing_tolerance	0.1	Only solutions with objective value $(1 + \text{tol})$ are polished
	polishing_max_runtime	10.0	max time to run polishing each time
	polishing_max_solutions	5.0	Max # of solutions to polish each time
	polishing_start_cuts	0	Cuts needed to start using polishing heuristic
	polishing_start_gap	∞	Min optimality gap needed to start using polishing heuristic
	polishing_stop_cuts	∞	Cuts needed to stop using polishing heuristic
	polishing_stop_gap	0.0	Max optimality gap required to stop using polishing heuristic
Initialization Procedure	initialization_flag	True	Use initialization procedure
	init_display_progress	False	Show progress of initialization procedure
	init_display_cplex_progress	False	Show progress of CPLEX during intialization procedure
	init_max_runtime	300.0	Max time to run CPA in initialization procedure
	init_max_iterations	10000	Max # of cuts needed to stop CPA
	init_max_tolerance	0.0001	Tolerance of solution to stop CPA
	init_max_runtime_per_iteration	300.0	Max time per iteration of CPA
	init_max_cplex_time_per_iteration	10.0	Max time per iteration to solve surrogate problem in CPA
	init_use_rounding	True	use rounding in initialization procedurue
	init_rounding_max_runtime	30.0	Max runtime for Rd in initialization procedure
	init_rounding_max_solutions	5	Max solutions to round using rounding
	init_use_sequential_rounding	True	Use SeqRd in initialization procedure
	init_sequential_rounding_max_runtime	10.0	Max runtime for SeqRd in initialization procedure
	init_sequential_rounding_max_solutions	5	Max solutions to round using SeqRd
	init_polishing_after	True	Polish after rounding
	init_polishing_max_runtime	30.0	Max runtime for polishing
	init_polishing_max_solutions	5	Max solutions to polish
CPLEX Solver Parameters	cplex_randomseed	0	Random seed
	cplex_mipemphasis	0	Cplex MIP strategy

*All parameters defined in Ustun and Rudin [208]