Data-driven healthcare via constraint learning and analytics

by

Holly Mika Wiberg

B.S., Cornell University (2016)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . . . .
Sloan School of Management
April 22, 2022

Certified by . . . . . .
Dimitris J. Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Accepted by . . . . . .
Georgia Perakis
William F. Pounds Professor of Management Science
Co-director, Operations Research Center

Data-driven healthcare via constraint learning and analytics

by

Holly Mika Wiberg

Submitted to the Sloan School of Management
on April 22, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

The proliferation of digitally-available medical data has enabled a new paradigm of decision-making in medicine. Machine learning allows us to glean large-scale insights directly from data, systematizing the heuristic risk assessment process that physicians use on a local scale. Optimization similarly adds rigor to decision-making, providing a quantitative framework for optimizing decisions under certain constraints. The rise in data, coupled with methodological and computational advancements in these fields, presents both opportunities and challenges. In this thesis, we leverage machine learning and optimization to learn from data and drive better decisions in healthcare. We propose novel approaches motivated by current methodological gaps, and we use analytics to tackle clinically-driven problems. This thesis develops methods and applied models to bridge the gap between research and clinical practice, with interpretability and impact as guiding principles.

The first part of the thesis focuses on the development of new approaches for data-driven insights and decision-making. Chapter 2 introduces a constraint learning framework that embeds trained machine learning models directly into mixed-integer optimization formulations. We train machine learning models to approximate functional relationships between decisions and outcomes of interest and subsequently optimize decisions under these data-driven learned constraints and/or objectives. We also highlight an application of this framework in chemotherapy regimen design. In Chapter 3, we propose an interpretable clustering algorithm which learns a tree-based data partition in which each leaf comprises a distinct cluster. We recover high-quality clusters that can be explicitly described by their decision paths.

The second part of the thesis leverages machine learning and optimization to improve risk prediction and treatment decisions in various domains. We present three such applications. In Chapter 4, we study neutropenic events in chemotherapy patients. We propose a risk prediction model based on a patient's dynamic clinical trajectory over the course of multiple chemotherapy cycles. Chapter 5 demonstrates the use of analytics to address the COVID-19 pandemic. We curate a multi-center, international database of COVID-19 patients and their outcomes, which forms the basis for a COVID-19 mortality risk model for hospitalized patients. Finally, Chapter 6 examines the effectiveness of in-person vs. virtual care from a causal inference lens, considering the effect of visit modality on both operational and clinical outcomes. The resultant machine learning models inform an optimization formulation for allocating telehealth and in-person visits

for diabetic patients.

Thesis Supervisor: Dimitris J. Bertsimas
Title: Boeing Professor of Operations Research

Acknowledgments

I would first like to thank my advisor, Dimitris Bertsimas. I met Dimitris in my freshman year of college, and it truly did change the course of my life. I started my research career with him, and I am grateful for the journey we have been on together over the last nine years. He often says that our work can “change the world", and if there’s anyone who could make that happen, it would be Dimitris. He has boundless energy, positivity, and a deep commitment to impact, which have shaped my own research values. He is not only a trusted mentor, but also a friend. Our collaborations have involved many road trips (including a record 14 hours across two days), and Dimitris has filled these rides with stories from Greece, many rounds of 20 questions, and advice on how to find joy in work. I can only hope to match his enthusiasm and dedication to students in my own career.

I am also extremely lucky to have been guided and encouraged by a group of academic role models. First, I appreciate the support of my two thesis committee members, Cynthia Barnhart and Dick den Hertog. Cindy has advised me at many key junctures, including pursuing graduate school and ultimately an academic career. Her mentorship has inspired me in both research and university service. I have thoroughly enjoyed working with Dick over the past year as collaborators, and earlier taking robust optimization with him and Dimitris (a powerhouse duo!). His thoughtfulness has made me a better researcher, and his passion for OR is infectious. I also thank Georgia Perakis and Nikos Trichakis for their guidance on my General Exam committee, and Alexandre Jacquillat for his collaboration and advice. Laura Rose and Andrew Carvalho are the engine behind the ORC, and none of us would be graduating without you.

While I will finish my Operations Research education at MIT, I fell in love with OR at Cornell. I am grateful to Shane Henderson, David Shmoys, Jamol Pender, and Robert Bland, among others, for all that they have taught me about research, teaching, and life. I also appreciate the mentorship and friendship from Daniel Freund and Nanjing Jian, who are fantastic researchers and even better friends. I am especially lucky that Daniel landed in Cambridge after his PhD, and I owe him many coffees for his advice and encouragement in the past years.

My research has been made possible by phenomenal collaborators. In the past two years, I have had the pleasure of working with Donato Maragno, Ade Fajemisin, Ilker Birbil, and Dick

den Hertog from the University of Amsterdam. Donato and I have spent countless hours on Zoom, brainstorming formulations and debating F1, and I am excited for the day when our teams finally convene in person. Additionally, my work with ORC colleagues, including the SwissRe, COVID Analytics, and HAIM teams, has been both intellectually rewarding and quite fun. Finally, I have learned an incredible amount from my clinical collaborators, and I am forever indebted for the endless data requests that they answered. This list includes Peter Yu, Sue Barrett, Chris Bombara, and others at Hartford HealthCare, Peter Masiakos and Casey Luckhurst at Massachusetts General Hospital, and Georgios Antonios Margonis at Memorial Sloan Kettering Cancer Center. Barry Stein has been a champion of analytics at Hartford HealthCare, which has opened the door to many interesting projects and an opportunity to develop a new medical education program. Beyond research, he has been a fantastic supporter throughout graduate school and particularly in the last year as I planned my next steps.

I cannot imagine my PhD without the other half of the WOAH dream team, Agni Orfanoudaki. Agni has been my partner-in-crime through it all, with countless hours spent together in the ORC, whether coding something for research, drafting our textbook, editing slide decks, or grading final projects. Somehow, all of these things seem fun with Agni. We have also managed to make it out of the office for many adventures, from the Cape to Crete (which are decidedly even more fun). I look forward to many more collaborations and UK-US trips in the years to come.

I am so grateful for our ORC cohort. Ted, Jess, and Pato were the ultimate party planners and the best co-INFORMS officers I could ask for. Long days were made better with a coffee break with Yannis, Ryan, and Peter, or a walk home to Beacon Hill with Ted and Andy. Lea, Rebecca, and Xiaoyue have been the best support team. Beyond our stellar fifth year crew, the ORC has become a second home thanks to colleagues who are both brilliant and incredibly fun. Our time together has been a highlight of my life, and I look forward to the ORC brew club's national tour over the coming years.

My friends from outside the ORC have kept me smiling and remind me about the world outside research, from near and far. I wouldn't have made it here without Jessie, Megan, Christina, Kairys, Becks, Carolyn, and many others. I am so lucky to be surrounded by such amazing friends, who encourage me to dream big and not to take life so seriously. I also thank Sunny, who has taught me to stay positive, work hard, and take time to bring joy to those around you.

To George, who has been a phenomenal partner. We have tackled a lot over the last six years, from New York to Kentucky to Michigan to Massachusetts. He is a constant source of encouragement, both pushing me to take new risks and reminding me to go easy on myself. And as an added bonus, he is a great unpaid medical consultant. I can't wait for our next adventure.

My grandparents, Ken and Marge Wiberg and Richard and Toyoko Sperry, paved the way for my parents and for me. They worked tirelessly to build a life where their children could thrive, and they have done the same for their grandchildren. They have picked me up when I have fallen (literally and figuratively), encouraged me to be more adventurous (after all taking their own risks of moving across the country and world), and most of all remained fiercely dedicated to their families. I am so lucky to have started this PhD journey with them all by my side. I am especially grateful to my professor role models in my own family–Ken and Pat Wiberg. They have provided advice and edited statements, but most importantly shown me the joys of academia. Their love of learning, teaching, and discovery inspires me as I embark on a new journey as a professor.

Finally, I owe everything to my family, who has supported my every dream. My parents, Lynda and Bill, and my sister, Wendy, have been my anchors through this PhD and through my life. They have stuck with me through it all–always ready to pick up the phone (or drive to upstate NY) when a day has been hard, and ready to pop the champagne when there's something to celebrate. They show me unconditional love every day, and knowing I have them in my corner provides the greatest comfort I can imagine. My successes are yours.

This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 174530. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation. This work is additionally supported by Hartford HealthCare and SwissRe.

Contents

Chapter 1

Introduction

Machine learning (ML) and optimization have the potential to transform medicine. In the past decade, as medical data has become increasingly digitized, an opportunity has emerged for data-driven healthcare decisions. Patient care has traditionally been driven by formal medical training and personal experience; predictions about patient outcomes and treatment decisions evolve based on a physician's previous patients and those of their colleagues. ML introduces a new paradigm of evidence-based medicine, scaling the best practices generated from physician knowledge and experience across sites and diverse populations. Optimization similarly systematizes decision-making, providing a quantitative framework for optimizing decisions under various imposed constraints. Coupled with data, these fields can shape future medical care with more accurate risk predictions, better treatment personalization, earlier detection of disease, and ultimately preventative intervention [119, 136].

This opportunity is not limited to the clinical domain. Healthcare organizations are complex systems with numerous stakeholders and operational challenges. Efficient appointment scheduling, employee staffing, and patient flow require accurate predictions of patient needs to inform resource allocation. In the policy domain, policymakers similarly grapple with multi-objective, complex decisions. Analytics can be used to identify effective community-level interventions, develop clinical guidelines, and audit systems for disparate impact. Once again, a combination of predictive modeling and optimization presents an opportunity to improve decision-making.

Despite the power of ML and optimization to derive data-driven insights and prescriptions, and the growing availability of clinical data to inform models, there are challenges to realizing

this vision [64, 95, 162]. Electronic medical record (EMR) data are notoriously noisy and often consist of several sources and modalities, requiring extensive data curation. Furthermore, there are obstacles to applying analytics in healthcare that necessitate the development of new methods. For example, state-of-the-art methods often lack interpretability, which is critical to gaining credibility with clinical audiences. Additionally, in order to leverage optimization, models require explicit functions that tie decisions to their predicted outcomes, which are frequently unknown and must be learned from data.

In this thesis, we harness the power of ML and optimization to glean insights from data and improve both clinical and operational decision-making. In the chapters that follow, we develop methods and applied models to bridge the gap between research and clinical practice, with interpretability and impact as guiding principles.

1.1 The intersection of optimization and machine learning

The first part of the thesis focuses on the development of novel approaches for data-driven decision-making. We first propose a framework for mixed-integer optimization (MIO) with constraint learning. In this work, we leverage ML within a broader MIO formulation. We train ML models to approximate functional relationships between decisions and outcomes of interest and subsequently optimize decisions under these data-driven learned constraints and/or objectives.

We then take an optimization perspective to ML in the development of an interpretable clustering algorithm. We introduce ICOT, interpretable clustering via optimal trees, which learns a decision tree in which each leaf comprises a distinct cluster. We propose an MIO formulation for this tree partitioning problem, and implement a fast local search algorithm with scaling heuristics. We recover high-quality clusters that can be explicitly described by their decision paths. ICOT adds interpretability to unsupervised learning, which is particularly relevant in exploratory analysis such as subgroup identification.

Both of these chapters bring together optimization and ML to more effectively learn from data and use data to inform decisions. We address problems that have natural application in healthcare, but their relevance is quite broad to settings with complex decisions and those in which interpretability is crucial. A key goal of these works is the development of tools that are useful in

practice. The constraint learning framework is implemented as a Python package, `OptiCL`, which allows end-users to flexibly incorporate ML models in MIO problems. `ICOT` is also implemented in Julia and freely available to academic users.

1.1.1 Outline and main contributions

Mixed-integer optimization with constraint learning In Chapter 2, we propose a combined ML and optimization framework for learning constraints and objectives from data. MIO is a valuable tool for modeling and optimizing decisions, but the challenge lies in translating the real-world problem into a mathematical formulation. We often have no deterministic functions relating the decision variables to the outcomes of interest. Critically, however, we generally do have data. In healthcare, this consists of data about patients, care decisions, and their outcomes, whether from EMR, registries, or clinical trials. We can thus use ML to learn predictive models for various outcomes that we may want to constrain or optimize, and subsequently embed these models in an MIO formulation to generate the desired prescriptions. We exploit the MIO-representability of many classes of ML methods, including linear models, decision trees, ensembles, and multi-layer perceptron networks. Our learning process is therefore able to capture quite general relationships between the patient features, treatments, and outcomes. We are also able to flexibly include additional explicit constraints or objectives, such as a limit on the total cost of the treatment regimen or on the number of drugs included in the treatment. Finally, we characterize the feasible region using a trust region to avoid model extrapolation and obtain reasonable decisions based on previous observations. By combining ML with optimization, we are able to integrate both *data-driven* and *context-driven* constraints and objectives to generate prescriptive recommendations.

We further demonstrate the power of our proposed constraint learning framework through a case study on chemotherapy regimen design. Chemotherapy regimen selection involves complex decisions: plans generally include multiple drugs given at different doses, making it impossible to enumerate and explore all treatment options. This is a natural setting for constraint learning: we have outcomes of interest (survival and multiple toxicities) that have complex dependencies on patient characteristics and selected treatment regimens. We leverage a database of nearly 500 Phase II/III clinical trials for advanced gastric cancer for this task [28]. For a cohort with specified

features ($\boldsymbol{w}$), we seek to identify the optimal treatment regimen ($\boldsymbol{x}$) that maximizes overall survival under multiple toxicity constraints. Our framework is amenable to different model specifications, allowable toxicity risks, and alternative measures of survival. This provides an actionable tool for the medical community to quantify the clinical benefit of various treatment options and to evaluate tradeoffs between toxicity and survival.

We also provide a software package, `OptiCL`, that implements the end-to-end constraint learning pipeline. Given data ($\boldsymbol{x}, \boldsymbol{w}$) and a set of outcomes to constrain or optimize ($\boldsymbol{y}$), we train various ML models, perform automated model selection using a validation criterion, embed the selected models within a single MIO formulation, incorporate other known constraints, and return the optimal treatment prescription. `OptiCL` offers an easy interface for model development and experimentation, equipping users with a powerful toolkit and bridging the gap to practice.

This is joint work with Donato Maragno, Dimitris Bertsimas, Ilker Birbil, Dick den Hertog, and Ade Fajemisin, and is under review at *Operations Research* (Major Revision) [117].

Interpretable clustering Clustering is a popular tool for exploratory analysis, revealing underlying patterns or subgroups in data. Clustering can be used on its own to gain insight into a dataset or to partition a population for a downstream predictive task. For these aims, it is desirable, and sometimes necessary, to explicitly characterize subgroup membership. This is a challenge in the clustering literature: existing popular methods, such as K-means, provide no explanation of cluster membership. While there is a significant body of literature on interpretable supervised learning methods, this area is relatively unexplored in the unsupervised learning setting.

In Chapter 3, we introduce Interpretable Clustering via Optimal Trees (ICOT), an algorithm that partitions data into clusters through a single decision tree. This representation yields an explicit characterization of cluster membership, providing a more intuitive view of the resultant subgroups and their differentiating features. ICOT constructs clusters that are inherently interpretable, rather than considering cluster interpretation as a post-processing step. We formulate the tree construction as an MIO problem seeking to maximize a measure of cluster quality and implement it using a local search procedure that leverages the geometric structure of the clustering setting. ICOT performs comparably to state-of-the-art methods on benchmark clustering tasks, showing almost no loss in cluster quality while obtaining a significant gain in interpretability and practical utility. Scaling

heuristics allow ICOT to scale to large real-world datasets in tasks such as identifying bikesharing rider profiles and heart attack patterns.

We also implement `ICOT` within the InterpretableAI software [90]. Given a feature matrix X, the algorithm returns a decision tree that partitions observations into clusters and can be used to assign clusters to new observations. We include optional user-specified parameters used in tree construction (cluster quality metric, tree size limit, and the weight of categorical vs. numerical features) as well as optional scaling heuristics. The implementation is freely available to all academic users.

This is joint work with Dimitris Bertsimas and Agni Orfanoudaki, and appears in *Machine Learning* [31].

1.2 Analytics for clinical and operational decisions

While there is great interest in ML-driven healthcare [136], a gap remains between this research area and clinical practice. There are many hurdles involved in translating models into practical decision support tools, including technical hurdles (implementation barriers, inconsistent EMR structures), organizational hurdles (lack of clinical buy-in), and a disconnect between models and the clinical questions that they intend to address.

The second part of my thesis focuses on leveraging ML and optimization to improve risk prediction and treatment decisions in various domains. These works seek to bridge the gap between research and practice, balancing quantitative and clinically-oriented perspectives to derive data-driven insights. The following chapters include representative papers from a broad set of clinical application areas, including cancer [33, 26, 167], pediatric trauma [27], diabetes care, and COVID-19 [21, 22, 25]. While spanning diverse application areas, this body of work is unified by common themes: the problems were identified in collaboration with medical partners; the data curation and modeling were guided by clinical utility; and the models have led to the development of practical tools.

These goals have manifested in several ways. We worked closely with collaborators in *clinical problem definition* to ensure that we are answering a clinically relevant question. Clinical decisions are complex, involving many assessment points over time and many data inputs. By identifying

end-user stakeholders in the and engaging them from the outset, we define precise problems that make our models useful to their daily practice. Our *data curation process* is oriented towards the point-of-care, ensuring that the feature space accurately encodes the patient's condition at the time of interest. We develop data mappings and ontologies to synthesize disparate data sources, which remain as artifacts in their respective institutions to accelerate new research endeavors. Finally, we prioritize interpretability in model selection. Our development of *user-friendly interfaces* has proved critical in establishing credibility with clinicians and validating models.

1.2.1 Outline and main contributions

Prediction of neutropenic events in chemotherapy patients Severe and febrile neutropenia pose significant hazards to patients undergoing chemotherapy. A better assessment of neutropenic risk at the initiation of a chemotherapy cycle enables better care management and potential intervention. In Chapter 4, we train and validate a neutropenia prediction model using data from nearly 18,000 chemotherapy cycles from Hartford HealthCare. We obtain a sparse logistic regression risk prediction model that accurately predicts neutropenia onset within a 4 week window (test AUC = 0.87, 95% CI 0.83-0.91) and only requires 20 clinical features. These features reflect a patient's dynamic clinical state, allowing for repeated evaluation over subsequent chemotherapy cycles. Our resultant model is both sparse and interpretable, providing transparency for clinical validation and lowering barriers to future implementation and adoption. This work addresses a pressing clinical question and provides a tool to aid oncologists in assessing their patients and designing care plans.

This is joint worth with Dimitris Bertsimas and our clinical collaborators at Hartford HealthCare, Peter Yu, Pat Montanaro, Jeff Mather, Suzi Birz, and Michelle Schneider, and appears in *JCO Clinical Cancer Informatics* [167].

COVID-19 mortality risk assessment As the COVID-19 pandemic emerged in spring 2020, it raised many clinical, operational, and epidemiological questions. Researchers were eager to tackle these questions, yet the novelty of the disease and its rapid onset posed an immense challenge. In particular, efforts to understand risk factors and treatment effectiveness were hindered by a lack of large-scale clinical data. We set out to address several problems, starting with COVID-19 mortality prediction for hospitalized patients, which is the focus of Chapter 5. In this work, we develop

and validate a COVID-19 mortality risk calculator (CMR). The model derivation and validation are performed using a manually curated dataset of detailed clinical features and outcomes. We established several collaborations with international institutions, generating a database of 4,000 COVID-19 patients across 33 sites. The final CMR model demonstrates strong quantitative performance (test AUC 0.90, 95% CI, 0.87–0.94) and identifies risk factors consistent with the literature, such as hypoxemia and elevated renal function lab values. This tool offers an accurate and interpretable model for understanding COVID-19 severity upon hospital admission, with implications on patient management and triage. In particular, it allows users to calculate risk scores based on demographics, vitals, comorbidities, and (optionally) lab values. We also surface personalized insights into risk drivers, adding interpretability and allowing for clinical collaborators to engage with and validate the model findings. In addition to the public interface, CMR was subsequently implemented in one of our external validation sites, a hospital system serving Seville, Spain.

This is joint work with Dimitris Bertsimas, Galit Lukin, Luca Mingardi, Omid Nohadani, Agni Orfanoudaki, and Bartolomeo Stellato, as well as our clinical collaborators, and appears in *PLOS One* [25].

Optimizing virtual care for chronic disease patients In partnership with the Hartford HealthCare Medical Group, we investigate the effectiveness of virtual and in-person care for diabetic patients. This is a timely issue in the wake of the COVID-19 pandemic, which has significantly accelerated telehealth adoption. Policymakers and hospital systems must now determine how to best utilize telehealth in patient care beyond the pandemic. Chapter 6 tackles this question from a causal ML and optimization approach. We consider the visit modality, virtual vs. in-person, as a treatment and use causal inference methods to estimate individual treatment effects. These effects inform a scheduling model that optimizes a provider's virtual/in-person visit mix given their patient characteristics. We vary the problem to prioritize operational (no-show rate) and clinical (A1C control) outcomes, and consider various upper bounds on the overall virtual appointment rate. Overall, our findings suggest a benefit to increasing virtual care and can be extended to other chronic care settings.

Chapter 2

Mixed-integer Optimization with Constraint Learning

Abstract

We establish a broad methodological foundation for mixed-integer optimization with learned constraints. We propose an end-to-end pipeline for data-driven decision making in which constraints and objectives are directly learned from data using machine learning, and the trained models are embedded in an optimization formulation. We exploit the mixed-integer optimization representability of many machine learning methods, including linear models, decision trees, ensembles, and multi-layer perceptrons. The consideration of multiple methods allows us to capture various underlying relationships between decisions, contextual variables, and outcomes. We also characterize a decision trust region using the convex hull of the observations, to ensure credible recommendations and avoid extrapolation. We efficiently incorporate this representation using column generation and clustering. In combination with domain-driven constraints and objective terms, the embedded models and trust region define a mixed-integer optimization problem for prescription generation. We implement this framework as a Python package (`OptiCL`) for practitioners. We demonstrate the method in both chemotherapy optimization and World Food Programme planning. The case studies illustrate the benefit of the framework in generating high-quality prescriptions, the value added by the trust region, the incorporation of multiple machine learning methods, and the inclusion of multiple learned constraints.

2.1 Introduction

Mixed-integer optimization (MIO) is a powerful tool that allows us to optimize a given objective subject to various constraints. This general problem statement of optimizing under constraints is

nearly universal in decision-making settings. Some problems have readily quantifiable and explicit objectives and constraints, in which case MIO can be directly applied. The situation becomes more complicated, however, when the constraints and/or objectives are not explicitly known.

For example, suppose we deal with cancerous tumors and want to prescribe a treatment regimen with a limit on toxicity; we may have observational data on treatments and their toxicity outcomes, but we have no natural function that relates the treatment decision to its resultant toxicity. We may also encounter constraints that are not directly quantifiable. Consider a setting where we want to recommend a diet, defined by a combination of foods and quantities, that is sufficiently "palatable." Palatability cannot be written as a function of the food choices, but we may have qualitative data on how well people "like" various potential dietary prescriptions. In both of these examples, we cannot directly represent the outcomes of interest as functions of our decisions, but we have *data* that relates the outcomes and decisions. This raises a question: how can we consider data to learn these functions?

In this work, we tackle the challenge of data-driven decision making through a combined machine learning (ML) and MIO approach. ML allows us to learn functions that relate decisions to outcomes of interest directly through data. Importantly, many popular ML methods result in functions that are MIO-representable, meaning that they can be embedded into MIO formulations. This MIO-representable class includes both linear and nonlinear models, allowing us to capture a broad set of underlying relationships in the data. While the idea of learning functions directly from data is core to the field of ML, data is often underutilized in MIO settings due to the need for functional relationships between decision variables and outcomes. We seek to bridge this gap through *constraint learning*; we propose a general framework that allows us to learn constraints and objectives directly from data, using ML, and to optimize decisions accordingly, using MIO. Once the learned constraints have been incorporated into the larger MIO, we can solve the problem directly using off-the-shelf solvers.

The term *constraint learning*, used several times throughout this work, captures both constraints and objective functions. We are fundamentally learning functions to relate our decision variables to the outcome(s) of interest. The predicted values can then either be incorporated as constraints or objective terms; the model learning and embedding procedures remain largely the same. For this reason, we refer to them both under the same umbrella of *constraint learning*. We

describe this further in Section 2.2.2.

2.1.1 Literature review

Previous work has demonstrated the use of various ML methods in MIO problems and their utility in different application domains. The simplest of these methods is the regression function, as the approach is easy to understand and easy to implement. Given a regression function learned from data, the process of incorporating it into an MIO model is straightforward, and the final model does not require complex reformulations. As an example, Bertsimas et al. [28] use regression models and MIO to develop new chemotherapy regimens based on existing data from previous clinical trials. Kleijnen [98] provides further information on this subject.

More complex ML models have also been shown to be MIO-representable, although more effort is required to represent them than simple regression models. Neural networks which use the ReLU activation function can be represented using binary variables and big-M formulations [7, 76, 8, 48, 150, 163]. Where other activation functions are used [79, 110, 144], the MIO representation of neural networks is still possible, provided the solvers are capable of handling these functions.

With decision trees, each path in the tree from root to leaf node can be represented using one or more constraints [38, 164, 81]. The number of constraints required to represent decision trees is a function of the tree size, with larger trees requiring more linearizations and binary variables. The advantage here, however, is that decision trees are known to be highly interpretable, which is often a requirement of ML in critical application settings [155]. Random forests [36, 120] and other tree ensembles [52] have also been used in MIO in the same way as decision trees, with one set of constraints for each tree in the forest/ensemble along with one or more additional aggregate constraints.

Data for constraint learning can either contain information on continuous data, feasible and infeasible states (two-class data), or only one state (one-class data). The problem of learning functions from one-class data and embedding them into optimization models has been recently investigated with the use of decision trees [100], genetic programming [128], local search [151], evolutionary strategies [127], and a combination of clustering, principal component analysis and wrapping ellipsoids [129].

The above selected applications generally involve a single function to be learned and a fixed ML method for the model choice. Verwer et al. [164] use two model classes (decision trees and linear models) in a specific auction design application, but in this case the models were determined a priori. Some authors have presented a more general framework of embedding learned ML models in optimization problems [110, 20], but in practice these works are restricted to limited problem structures and learned model classes. Recently, Bergman et al. [20] introduced a software to embed neural networks and logistic and linear regression models as objective terms in an MIO formulation. These works can be viewed as special cases of our framework and cannot be directly applied in our case studies. We take a broader perspective, proposing a comprehensive end-to-end pipeline that encompasses the full ML and optimization components of a data-driven decision making problem.

Our work falls under the umbrella of prescriptive analytics. Bertsimas and Kallus [24] and Elmachtoub and Grigas [63] leverage ML model predictions as inputs into an optimization problem. Our approach is distinct from existing work in that we directly embed ML models rather than extracting predictions, allowing us to optimize our decisions over the model. In the broadest sense, our framework relates to work that jointly harnesses ML and MIO, an area that has garnered significant interest in recent years in both the optimization and machine learning communities [19].

2.1.2 Contributions

Our work unifies several research areas in a comprehensive manner. Our key contributions are as follows:

1. We develop an end-to-end framework that takes data and directly implements model training, model selection, integration into a larger MIO, and ultimately optimization. We make this available as an open-source software, `OptiCL` (Optimization with Constraint Learning) to provide a practitioner-friendly tool for making better data-driven decisions. The code is available at `https://github.com/hwiberg/OptiCL`.

2. We implement a model selection procedure that allows us to capture quite general functional relationships between contextual variables, treatments, and outcomes. We use cross-validation to select from a broad set of ML methods, assuming no single model's dominance,

and further allow for the combined use of different algorithms for different outcomes. Our framework supports models for both regression and classification functions, and handles constraint learning in cases with both one-class and two-class data. Additionally, we give mathematical representations of the ML functions to enable their use in MIO applications.

3. Due to the uncertainty associated with learning from data, we introduce a concept which we call a trust region. This allows us to restrict the solution of the optimization problem to be consistent with the domain of the predictive models. Defining this trust region in cases where there is a huge amount of data to learn from can be computationally intensive, so we also provide a column selection algorithm that significantly improves the computation time. We furthermore propose a clustering heuristic for a general MIO formulation that shows significant computational gains while obtaining near-optimality. These approaches allow us to reduce the computational burden of our approach while keeping the benefits of the trust region.

4. We demonstrate the power of our method in two real-world case studies, using data from the World Food Programme and chemotherapy clinical trials. We pose relevant questions in the respective areas and formalize them as constraint learning problems. We implement our framework and subsequently evaluate the quantitative performance and scalability of our methods in these settings.

2.2 Methodology

Suppose we have data $\mathscr{D} = \{(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{w}}_i, \bar{\boldsymbol{y}}_i)\}_{i=1}^{N}$, with observed treatment decisions $\bar{\boldsymbol{x}}_i$, contextual information $\bar{\boldsymbol{w}}_i$, and outcomes of interest $\bar{\boldsymbol{y}}_i$ for sample i. Following the guidelines proposed in [67], we present a framework that, given data $\mathscr{D}$, learns functions for the outcomes of interest ($\boldsymbol{y}$) that are to be constrained or optimized. These learned representations can then be used to generate predictions for a new observation with context $\boldsymbol{w}$. Figure 2-1 outlines the complete pipeline, which is detailed in the sections below.

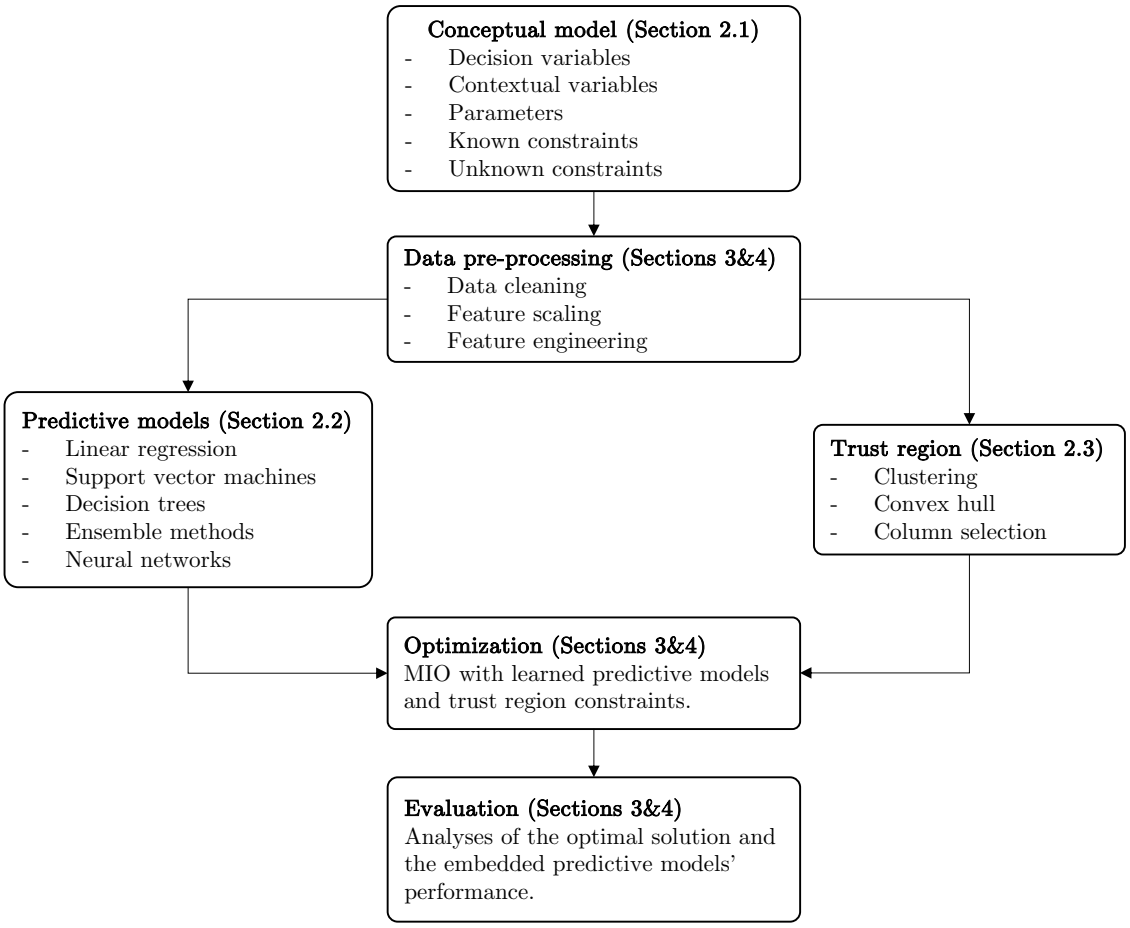


Figure 2-1: Constraint learning and optimization pipeline.

2.2.1 Conceptual model

Given the decision variable $\boldsymbol{x} \in \mathbb{R}^n$ and the fixed feature vector $\boldsymbol{w} \in \mathbb{R}^p$, we propose model M($\boldsymbol{w}$)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^k} f(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{y}) \tag{2.1}$$

$$\text{s.t. } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{y}) \leq \boldsymbol{0}, \tag{2.2}$$

$$\boldsymbol{y} = \hat{\boldsymbol{h}}_{\mathscr{D}(\boldsymbol{x},\boldsymbol{w})}, \tag{2.3}$$

$$\boldsymbol{x} \in \mathscr{X}(\boldsymbol{w}), \tag{2.4}$$

where $f(.,\boldsymbol{w},.) : \mathbb{R}^{n+k} \mapsto \mathbb{R}$, $\boldsymbol{g}(.,\boldsymbol{w},.) : \mathbb{R}^{n+k} \mapsto \mathbb{R}^m$, and $\hat{\boldsymbol{h}}_{\mathscr{D}(.,\boldsymbol{w})} : \mathbb{R}^n \mapsto \mathbb{R}^k$. Explicit forms of f and $\boldsymbol{g}$ are known but they may still depend on the predicted outcome $\boldsymbol{y}$. Here, $\hat{\boldsymbol{h}}_{\mathscr{D}(\boldsymbol{x},\boldsymbol{w})}$ represents the predictive models, one per outcome of interest, which are ML models trained on $\mathscr{D}$. Although our subsequent discussion mainly revolves around linear functions, we acknowledge the significant progress in nonlinear (convex) integer solvers. Our discussion can be easily extended to nonlinear models that can be tackled by those ever-improving solvers.

We note that the embedding of a single learned outcome may require multiple constraints and auxiliary variables; the embedding formulations are described in Section 2.2.2. For simplicity, we omit $\mathscr{D}$ in further notation of $\hat{\boldsymbol{h}}$ but note that all references to $\hat{\boldsymbol{h}}$ implicitly depend on the data used to train the model. Finally, the set $\mathscr{X}(\boldsymbol{w})$ defines the trust region, *i.e.*, the set of solutions for which we trust the embedded predictive models. In Section 2.2.3, we provide a detailed description of how the trust region $\mathscr{X}(\boldsymbol{w})$ is obtained from the observed data. We refer to the final MIO formulation with the embedded constraints and variables as EM($\boldsymbol{w}$).

Model M($\boldsymbol{w}$) is quite general and encompasses several important *constraint learning* classes:

1. **Regression.** When the trained model results from a regression problem, it can be constrained by a specified upper bound τ, *i.e.*, $g(y) = y - \tau \leq 0$, or lower bound τ, *i.e.*, $g(y) = -y + \tau \leq 0$. If $\boldsymbol{y}$ is a vector (*i.e.*, multi-output regression), we can likewise provide a threshold vector $\boldsymbol{\tau}$ for the constraints.

2. **Classification.** If the trained model is obtained with a binary classification algorithm, in which the data is labeled as "feasible" (1) or "infeasible" (0), then the prediction is generally a probability $y \in [0,1]$. We can enforce a lower bound on the feasibility probability, *i.e.*,

$y \geq \tau$. A natural choice of τ is 0.5, which can be interpreted as enforcing that the result is more likely feasible than not. This can also extend to the multi-class setting, say k classes, in which the output $\boldsymbol{y}$ is a k-dimensional unit vector, and we apply the constraint $y_i \geq \tau$ for whichever class i is desired. When multiple classes are considered to be feasible, we can add binary variables to ensure that a solution is feasible, only if it falls in one of these classes with sufficiently high probability.

3. **Objective function.** If the objective function has a term that is also learned by training an ML model, then we can introduce an auxiliary variable $t \in \mathbb{R}$, and add it to the objective function along with an epigraph constraint. Suppose for simplicity that the model involves a single learned objective function, $\hat{h}$, and no learned constraints. Then the general model becomes

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n, y\in\mathbb{R}, t\in\mathbb{R}} \quad & t \\ \text{s.t. } & \boldsymbol{g}(\boldsymbol{x},\boldsymbol{w}) \leq 0, \\ & y = \hat{h}(\boldsymbol{x},\boldsymbol{w}), \\ & y - t \leq 0, \\ & \boldsymbol{x} \in \mathscr{X}(\boldsymbol{w}). \end{aligned}$$

Although we have rewritten the problem to show the generality of our model, it is quite common in practice to use y in the objective and omit the auxiliary variable t.

We observe that constraints on learned outcomes can be applied in two ways depending on the model training approach. Suppose that we have a continuous scalar outcome y to learn and we want to impose an upper bound of $\tau \in \mathbb{R}$ (it may also be a lower bound without loss of generality). The first approach is called *function learning* and concerns all cases where we learn a regression function $\hat{h}(\boldsymbol{x},\boldsymbol{w})$ without considering the feasibility threshold (τ). The resultant model returns a predicted value $y \in \mathbb{R}$. The threshold is then applied as a constraint in the optimization model as $y \leq \tau$. Alternatively, we could use the feasibility threshold τ to binarize the outcome of each sample in $\mathscr{D}$ into feasible and infeasible, that is $\bar{y}_i := \mathbb{I}(\bar{y}_i \leq \tau),\ i = 1,\ldots,N$, where $\mathbb{I}$ stands for the indicator function. After this relabeling, we train a binary classification model $\hat{h}(\boldsymbol{x},\boldsymbol{w})$ that

returns a probability $y \in [0,1]$. This approach, called *indicator function learning*, does not require any further use of the feasibility threshold τ in the optimization model, since the predictive models directly encode feasibility.

The function learning approach is particularly useful when we are interested in varying the threshold τ as a model parameter. Additionally, if the fitting process is expensive and therefore difficult to perform multiple times, learning an indicator function for each potential τ might be infeasible. In contrast, the indicator function learning approach is necessary when the raw data contains binary labels rather than continuous outcomes, and thus we have no ability to select or vary τ.

2.2.2 MIO-representable predictive models

Our framework is enabled by the ability to embed learned predictive models into an MIO formulation with linear constraints. This is possible for many classes of ML models, ranging from linear models to ensembles, and from support vector machines to neural networks. In this section, we detail the embedding procedure. In all cases, the model has been *pre-trained*; we embed the trained model $\hat{h}(\boldsymbol{x}, \boldsymbol{w})$ into our larger MIO formulation to allow us to constrain or optimize the resultant predicted value. Consequently, the optimization model is not dependent on the complexity of the model training procedure, but solely the size of the final trained model. Without loss of generality, we assume that y is one-dimensional; *i.e.*, we are learning a single model, and this model returns a scalar, not a multi-output vector.

All of the methods below can be used to learn constraints that apply upper or lower bounds to y, or to learn y that we incorporate as part of the objective. We present the model embedding procedure for both cases when $\hat{h}(\boldsymbol{x}, \boldsymbol{w})$ is a continuous or a binary predictive model, where relevant. We assume that either regression or classification models can be used to learn feasibility constraints, as described in Section 2.2.1.

Linear Regression. Linear regression (LR) is a natural choice of predictive function given its inherent linearity and ease of embedding. A regression model can be trained to predict the outcome of interest, y, as a function of $\boldsymbol{x}$ and $\boldsymbol{w}$. The algorithm can optionally use regularization; the embedding only requires the final coefficient vectors $\boldsymbol{\beta}_x \in \mathbb{R}^n$ and $\boldsymbol{\beta}_w \in \mathbb{R}^p$ (and intercept term β_0)

to describe the model. The model can then be embedded as

$$y = \beta_0 + \boldsymbol{\beta}_x^\top \boldsymbol{x} + \boldsymbol{\beta}_x^\top \boldsymbol{w}.$$

Support Vector Machines. A support vector machine (SVM) uses a hyper-plane split to generate predictions, both for classification [51] and regression [58]. We consider the case of *linear* SVMs, since this allows us to obtain the prediction as a linear function of the decision variables $\boldsymbol{x}$. In linear support vector regression (SVR), which we use for function learning, we fit a linear function to the data. The setting is similar to linear regression, but the loss function only penalizes residuals greater than an ε threshold [58]. As with linear regression, the trained model returns a linear function with coefficients $\boldsymbol{\beta}_x$, $\boldsymbol{\beta}_w$, and β_0. The final prediction is

$$y = \beta_0 + \boldsymbol{\beta}_x^\top \boldsymbol{x} + \boldsymbol{\beta}_w^\top \boldsymbol{w}.$$

For the classification setting, linear support vector classification (SVC) identifies a hyper-plane that best separates positive and negative samples [51]. A trained SVC model similarly returns coefficients $\boldsymbol{\beta}_x$, $\boldsymbol{\beta}_w$, and β_0, where a sample's prediction is given by

$$y = \begin{cases} 1, & \text{if } \beta_0 + \boldsymbol{\beta}_x^\top \boldsymbol{x} + \boldsymbol{\beta}_w^\top \boldsymbol{w} \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

In SVC, the output variable y is binary rather than a probability. In this case, the constraint can be embedded as $y \geq 1$.

Decision Trees. Decision trees partition observations into distinct *leaves* through a series of *feature splits*. These algorithms are popular in predictive tasks due to their natural interpretability and ability to capture nonlinear interactions among variables. [40] first introduced Classification and Regression Trees (CART), which constructs trees through parallel splits in the feature space. Decision tree algorithms have subsequently been adapted and extended. [23] propose an alternative decision tree algorithm, Optimal Classification Trees (and Optimal Regression Trees), that improves

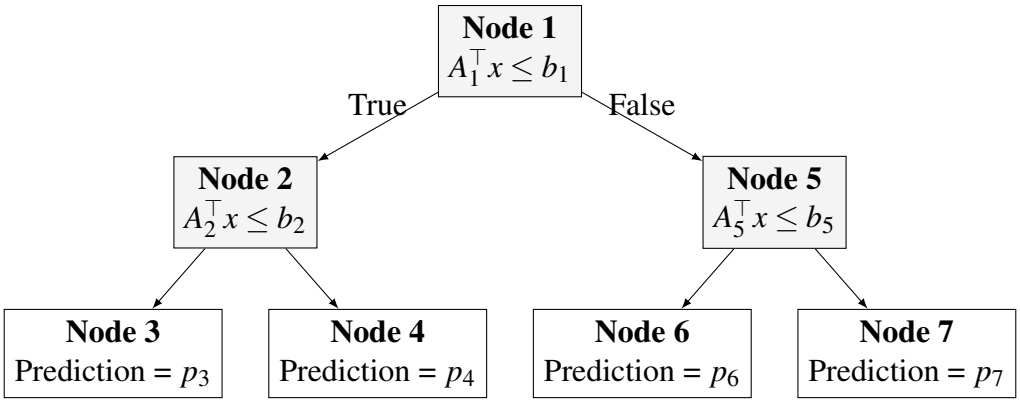


Figure 2-2: A decision tree of depth 2 with four terminal nodes (leaves).

on the basic decision tree formulation through an optimization framework that approximates globally optimal trees. Optimal trees also support multi-feature splits, referred to as *hyper-plane splits*, that allow for splits on a linear combination of features [34].

A generic decision tree of depth 2 is shown in Figure 2-2. A split at node i is described by an inequality $A_i^\top \boldsymbol{x} \leq b_i$. We assume that A can have multiple non-zero elements, in which we have the hyper-plane split setting; if there is only one non-zero element, this creates a parallel (single feature) split. Each terminal node j (*i.e.*, leaf) yields a prediction (p_j) for its observations. In the case of regression, the prediction is the average value of the training observations in the leaf, and in binary classification, the prediction is the proportion of leaf members with the feasible class. Each leaf can be described as a polyhedron, namely a set of linear constraints that must be satisfied by all leaf members. For example, for node 3, we define $\mathscr{P}_3 = \left\{x : A_1^\top x \leq b_1, A_2^\top x \leq b_2\right\}$.

Suppose that we wish to constrain the predicted value of this tree to be at most τ, a fixed constant. After obtaining the tree in Figure 2-2, we can identify which paths satisfy the desired bound ($p_i \leq \tau$). Suppose that p_3 and p_6 do satisfy the bound, but p_4 and p_7 do not. In this case, we can enforce that our solution belongs to $\mathscr{P}_3$ or $\mathscr{P}_6$. This same approach applies if we only have access to two-class data (feasible vs. infeasible); we can directly train a binary classification algorithm and enforce that the solution lies within one of the "feasible" prediction leaves (determined by a set probability threshold).

If the decision tree provides our only learned constraint, we can decompose the problem into multiple separate MIOs, one per feasible leaf. The conceptual model for the subproblem of leaf i

then becomes

$$\begin{aligned} \min_{\boldsymbol{x}} \; & f(\boldsymbol{x}, \boldsymbol{w}) \\ \text{s.t. } & \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{w}) \leq 0, \\ & (\boldsymbol{x}, \boldsymbol{w}) \in \mathscr{P}_i, \end{aligned}$$

where the learned constraints for leaf i's subproblem are implicitly represented by the polyhedron $\mathscr{P}_i$. These subproblems can be solved in parallel, and the minimum across all subproblems is obtained as the optimal solution. Furthermore, if all decision variables $\boldsymbol{x}$ are continuous, these subproblems are linear optimization problems (LOs), which can provide substantial computational gains. This is explored further in Appendix A.1.

In the more general setting where the decision tree forms one of many constraints, or we are interested in varying the τ limit within the model, we can directly embed the model into a larger MIO. We add binary variables representing each leaf, and set y to the predicted value of the assigned leaf. An observation can only be assigned to a leaf, if it obeys all of its constraints; the structure of the tree guarantees that exactly one path will be fully satisfied, and thus, the leaf assignment is uniquely determined. A solution belonging to $\mathscr{P}_3$ will inherit $y = p_3$. Then, y can be used in a constraint or objective. The full formulation for the embedded decision tree is included in Appendix A.1.

Ensemble Methods. Ensemble methods, such as random forests (RF) and gradient-boosting machines (GBM) consist of many decision trees that are aggregated to obtain a single prediction for a given observation. These models can thus be implemented by embedding many "sub-models" [39]. Suppose we have a forest with P trees. Each tree can be embedded as a single decision tree (see previous paragraph) with the constraints from Appendix A.1, which yields a predicted value y_i.

RF models typically generate predictions by taking the average of the predictions from the individual trees:

$$y = \frac{1}{P} \sum_{i=1}^{P} y_i.$$

This can then be used as a term in the objective, or constrained by an upper bound as $y \leq \tau$; this can

be done equivalently for a lower bound. In the classification setting, the prediction averages the probabilities returned by each model ($y_i \in [0,1]$), which can likewise be constrained or optimized.

Alternatively, we can further leverage the fact that unlike the other model classes, which return a single prediction, the RF model generates P predictions, one per tree. When constraining the prediction, we can optionally impose a violation limit, enforcing that the constraint must hold for most of the trees within the forest, but can be violated by a proportion $\alpha \in [0,1]$. This allows for a degree of robustness to individual model predictions by discarding a small number of potential outlier predictions:

$$\frac{1}{P}\sum_{i=1}^{P} \mathbb{I}(y_i \leq \tau) \geq 1-\alpha.$$

Note that $\alpha = 0$ enforces the bound for all trees within the forest, yielding the most conservative estimate, whereas $\alpha = 1$ removes the constraint entirely.

In the case of GBM, we have an ensemble of base-learners which are not necessarily decision trees. The model output is then computed as

$$y = \sum_{i=1}^{P} \beta_i y_i,$$

where y_i is the predicted value of the i-th regression model $\hat{h}_i(\boldsymbol{x}, \boldsymbol{w})$, β_i is the weight associated with the prediction. Although trees are typically used as base-learners, in theory we might use any of the MIO-representable predictive models discussed in this section.

Neural Networks. We implement multi-layer perceptrons (MLP) with a rectified linear unit (ReLU) activation function, which form an MIO-representable class of neural networks [76, 8]. These networks consist of an input layer, $L-2$ hidden layer(s), and an output layer. In a given hidden layer l of the network, with nodes N^l, the value of a node $i \in N^l$, denoted as v_i^l, is calculated using the weighted sum of the previous layer's node values, followed by the ReLU activation function, $\text{ReLU}(x) = \max\{0, x\}$. The value is given as

$$v_i^l = \max\left\{0, \beta_{i0}^l + \sum_{j \in N^{l-1}} \beta_{ij}^l v_j^{l-1}\right\},$$

where β_i^l is the coefficient vector for node i in layer l. This nonlinear transformation of the input space over multiple nodes (and layers) allows MLPs to capture complex functions that other algorithms cannot adequately encode, making them a powerful class of models.

Critically, the ReLU operator, $v = \max\{0, x\}$, can be encoded using linear constraints, as detailed in Appendix A.1. The constraints for an MLP network can be generated recursively starting from the input layer, with a set of ReLU constraints for each node in each internal layer, $l \in \{2, \ldots, L-1\}$. This allows us to embed a trained MLP with an arbitrary number of hidden layers and nodes into an MIO. In a regression setting, the output layer L consists of a single node that is a linear combination of the node values in layer $L-1$, so it can be encoded directly as

$$y = v^L = \beta_0^L + \sum_{j \in N^{L-1}} \beta_j^L v_j^{L-1}.$$

In the binary classification setting, the output layer requires one neuron with a sigmoid activation function, $S(x) = \frac{1}{1+e^{-x}}$. The value is given as

$$v^L = \frac{1}{1 + e^{-(\beta_0^L + \boldsymbol{\beta}^{L\top} \boldsymbol{v}^{L-1})}}$$

with $v^L \in (0,1)$. This function is nonlinear, and thus, cannot be directly embedded into our formulation. However, if τ is our desired probability lower bound, it will be satisfied when $\beta_0^L + \boldsymbol{\beta}^{L\top} \boldsymbol{v}^{L-1} \geq \ln\left(\frac{\tau}{1-\tau}\right)$. Therefore, the neural network's output, binarized with a threshold of τ, is given by

$$y = \begin{cases} 1, & \text{if } \beta_0^L + \boldsymbol{\beta}^{L\top} \boldsymbol{v}^{L-1} \geq \ln\left(\dfrac{\tau}{1-\tau}\right); \\ 0, & \text{otherwise.} \end{cases}$$

For example, at a threshold of $\tau = 0.5$, the predicted value is 1 when $\beta_0^L + \boldsymbol{\beta}^{L\top} \boldsymbol{v}^{L-1} \geq 0$. Here, τ can be chosen according to the minimum necessary probability to predict 1. As for the SVC case, y is binary and the constraint can be embedded as $y \geq 1$. We refer to Appendix A.1 for the case of neural networks trained for multi-class classification.

2.2.3 Convex hull as trust region

As the optimal solutions of optimization problems are often at the extremes of the feasible region, this can be problematic for the validity of the trained ML model. Generally speaking the accuracy of a predictive model deteriorates for points that are farther away from the data points in $\mathscr{D}$ [74]. To mitigate this problem, we use a so-called trust region that prevents the predictive model from extrapolating. According to [62], when data is enclosed by a boundary of convex shape, the region inside this boundary is known as an interpolation region. This interpolation region is also referred to as the convex hull, and by excluding solutions outside the convex hull, we prevent extrapolation. If $\mathscr{Z} = \{\bar{\boldsymbol{z}}_i\}_{i=1}^N$ is the set of observed input data with $\bar{\boldsymbol{z}}_i = (\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{w}}_i)$, we define the trust region as the convex hull of this set and denote it by CH($\mathscr{Z}$). Recall that CH($\mathscr{Z}$) is the smallest convex polytope that contains the set of points $\mathscr{Z}$. It is well-known that computing the convex hull is exponential in time and space with respect to the number of samples and their dimensionality [147]. However, since the convex hull is a polytope, explicit expressions for its facets are not necessary. More precisely, CH($\mathscr{Z}$) is represented as

$$\text{CH}(\mathscr{Z}) = \left\{ \boldsymbol{z} \,\middle|\, \sum_{i \in \mathscr{I}} \lambda_i \bar{\boldsymbol{z}}_i = \boldsymbol{z},\ \sum_{i \in \mathscr{I}} \lambda_i = 1,\ \boldsymbol{\lambda} \geq 0 \right\}, \tag{2.5}$$

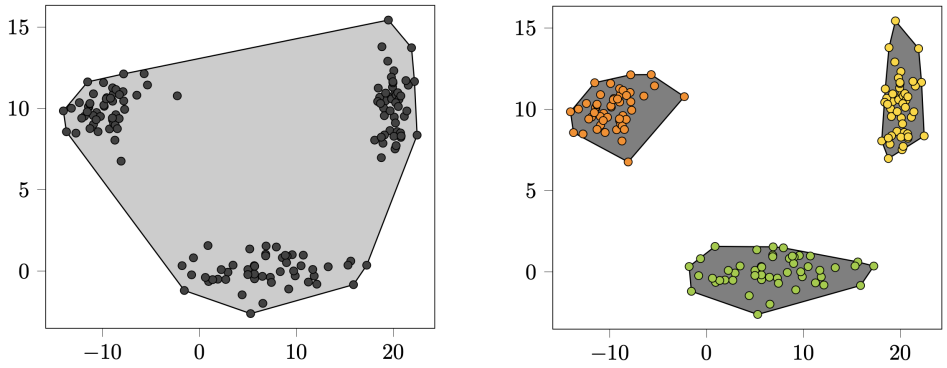
where $\boldsymbol{\lambda} \in \mathbb{R}^N$, and $\mathscr{I} = \{1, \ldots, N\}$ is the index set of samples in $\mathscr{Z}$.

In situations such as the one shown in Figure 2-3a, CH($\mathscr{Z}$) includes regions with few or no data points (low-density regions). Blindly using CH($\mathscr{Z}$) in this case can be problematic if the solutions are found in the low-density regions. We therefore advocate the use of a two-step approach. First, clustering is used to identify distinct high-density regions, and then the trust region is represented as the union of the convex hulls of the individual clusters (Figure 2-3b).

We can either solve EM($\boldsymbol{w}$) for each cluster, or embed the union of the $|\mathscr{K}|$ convex hulls into the MIO given by

$$\bigcup_{k \in \mathscr{K}} \text{CH}(\mathscr{Z}_k) = \left\{ \boldsymbol{z} \,\middle|\, \sum_{i \in \mathscr{I}_k} \lambda_i \bar{\boldsymbol{z}}_i = \boldsymbol{z},\ \sum_{i \in \mathscr{I}_k} \lambda_i = u_k\ \forall k \in \mathscr{K},\ \sum_{k \in \mathscr{K}} u_k = 1,\ \boldsymbol{\lambda} \geq 0,\ \boldsymbol{u} \in \{0,1\}^{|\mathscr{K}|} \right\}, \tag{2.6}$$

where $\mathscr{Z}_k \subseteq \mathscr{Z}$ refers to subset of samples in cluster $k \in \mathscr{K}$ with the index set $\mathscr{I}_k \subseteq \mathscr{I}$. The union



(a) CH($\mathscr{Z}$) with single region. (b) CH($\mathscr{Z}$) with clustered regions.

Figure 2-3: Use of the two-step approach to remove low-density regions.

of convex hulls requires the binary variables u_k to constrain a feasible solution to be exactly in one of the convex hulls. More precisely, $u_k = 1$ corresponds to the convex hull of the k-th cluster. As we show in Section 2.3, solving EM($\boldsymbol{w}$) for each cluster may be done in parallel, which has a positive impact on computation time. We note that both formulations (2.5) and (2.6) assume that $\bar{\boldsymbol{z}}$ is continuous. These formulations can be extended to datasets with binary, categorical and ordinal features. In the case of categorical features, extra constraints on the domain and one-hot encoding are required.

In addition to embedding the trust region for predictive models, this approach offers independent value in one-class constraint learning, which is an often studied problem in the literature [127, 129]. Here, data is composed of only feasible samples, so the predictive models discussed in Section 2.2.2 (which require both feasible and infeasible samples) are no longer suitable. A typical example occurs in real-world business processes like machine scheduling. Most of the schedules created by the machine shop supervisor are feasible, even if they may not be optimal. Thus, infeasible schedules are so infrequent that they may not be part of the dataset. We handle this one-class constraint learning task by employing the two-step approach, where we first cluster to identify separate high-density regions, and then use the union of convex hulls to represent the trust region.

Although the convex hull can be represented by linear constraints, the number of variables in EM($\boldsymbol{w}$) increases with the increase in the dataset size, which may make the optimization process prohibitive when the number of samples becomes too large. We therefore provide a column selec-

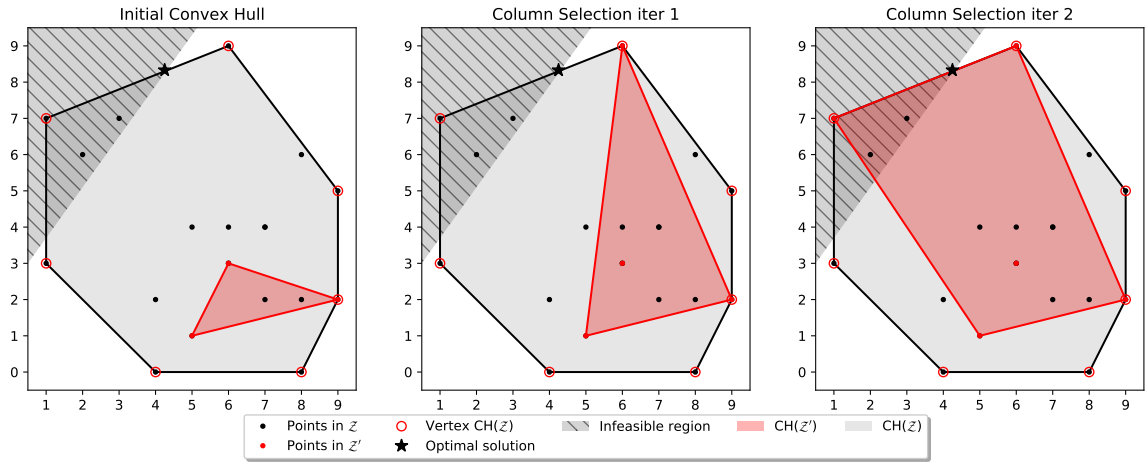


Figure 2-4: Visualization of the column selection algorithm. Known and learned constraints define the infeasible region. The column selection algorithm starts using only a subset of data points (red filled circles), $\mathscr{Z}' \subseteq \mathscr{Z}$ to define the trust region. In each iteration a vertex of CH($\mathscr{Z}$) is selected (red hollow circle) and included in $\mathscr{Z}'$ until the optimal solution (star) is within the feasible region, namely the convex hull of $\mathscr{Z}'$. Note that with column selection we do not need the complete dataset to obtain the optimal solution, but rather only a subset.

tion algorithm that selects a small subset of the samples. This algorithm can be used for EM($\boldsymbol{w}$) when all variables are continuous. Figure 2-4 visually demonstrates the procedure; we begin with an arbitrary sample of the full data, and use column selection to iteratively add samples $\bar{\boldsymbol{z}}_i$ until no improvement can be found. In Appendix A.2, we provide a full description of the approach, as well as a formal lemma which states that in each iteration of column selection, the selected sample from $\mathscr{Z}$ is also a vertex of CH($\mathscr{Z}$). In synthetic experiments, we observe that the algorithm scales well with the dataset size. The computation time required by solving the optimization problem with the algorithm is near-constant and minimally affected by the number of samples in the dataset. The experiments in Appendix A.2 show optimization with column selection to be significantly faster than a traditional approach, which makes it an ideal choice when dealing with massive datasets.

2.3 Case study: a palatable food basket for the World Food Programme

In this case study, we use a simplified version of the model proposed by [132], which seeks to optimize humanitarian food aid. Its extended version aims to provide the World Food Programme

(WFP) with a decision-making tool for long-term recovery operations, which simultaneously optimizes the food basket to be delivered, the sourcing plan, the delivery plan, and the transfer modality of a month-long food supply. The model proposed by [132] enforces that the food baskets address the nutrient gap and are palatable. To guarantee a certain level of palatability, the authors use a number of "unwritten rules" that have been defined in collaboration with nutrition experts. In this case study, we take a step further by inferring palatability constraints directly from data that reflects local people's opinions. We use the specific case of Syria for this example. The conceptual model presents an LO structure with only the food palatability constraint to be learned. Data on palatability is generated through a simulator, but the procedure would remain unchanged if data were collected in the field, for example through surveys. The structure of this problem, which is an LO and involves only one learned constraint, allows the following analyses: (1) the effect of the trust-region on the optimal solution, and (2) the effect of clustering on the computation time and the optimal objective value. Additionally, the use of simulated data provides us with a ground truth to use in evaluating the quality of the prescriptions.

2.3.1 Conceptual model

The optimization model is a combination of a capacitated, multi-commodity network flow model, and a diet model with constraints for nutrition levels and food basket palatability.

The sets used to define the constraints and the objective function are displayed in Table 2.1. We have three different sets of nodes, and the set of commodities contains all the foods available for procurement during the food aid operation.

Sets	
$\mathscr{N}_{\mathscr{S}}$	Set of source nodes
$\mathscr{N}_{\mathscr{T}}$	Set of transshipment nodes
$\mathscr{N}_{\mathscr{D}}$	Set of delivery nodes
$\mathscr{K}$	Set of commodities ($k \in \mathscr{K}$)
$\mathscr{L}$	Set of nutrients ($l \in \mathscr{L}$)

Table 2.1: Definition of the sets used in the WFP model.

The parameters used in the model are displayed in Table 2.2. The costs used in the objective function concern transportation (p^T) and procurement (p^P). The amount of food to deliver depends

on the demand (d) and the number of feeding days ($days$). The nutritional requirements ($nutreq$) and nutritional values ($nutrval$) are detailed in Appendix A.3. Here, the parameter γ is needed to convert the metric tons used in the supply chain constraints to the grams used in the nutritional constraints. The parameter t is used as a lower bound on the food basket palatability.

Parameters	
γ	Conversion rate from metric tons (mt) to grams (g)
d_i	Number of beneficiaries at delivery point $i \in \mathscr{N}_{\mathscr{D}}$
$days$	Number of feeding days
$nutreq_l$	Nutritional requirement for nutrient $l \in \mathscr{L}$ (grams/person/day)
$nutval_{kl}$	Nutritional value for nutrient $l \in \mathscr{L}$ per gram of commodity $k \in \mathscr{K}$
p^P_{ik}	Procurement cost (in \$ / mt) of commodity k from source $i \in \mathscr{N}_{\mathscr{S}}$
p^T_{ijk}	Transportation cost (in \$ / mt) of commodity k from node $i \in \mathscr{N}_{\mathscr{S}} \cup \mathscr{N}_{\mathscr{T}}$ to node $j \in \mathscr{N}_{\mathscr{T}} \cup \mathscr{N}_{\mathscr{D}}$
t	Palatability lower bound

Table 2.2: Definition of the parameters used in the WFP model.

The decision variables are shown in Table 2.3. The flow variables F_{ijk} are defined as the metric tons of a commodity k transported from node i to j. The variable x_k represents the average daily ration per beneficiary for commodity k. The variable y refers to the palatability of the food basket.

Variables	
F_{ijk}	Metric tons of commodity $k \in \mathscr{K}$ transported between node i and node j
x_k	Grams of commodity $k \in \mathscr{K}$ in the food basket
y	Food basket palatability

Table 2.3: Definition of the variables used in the WFP model.

The full model formulation is as follows:

$$\min_{\boldsymbol{x},y,\boldsymbol{F}} \sum_{i\in\mathscr{N}_{\mathscr{S}}} \sum_{j\in\mathscr{N}_{\mathscr{T}}\cup\mathscr{N}_{\mathscr{D}}} \sum_{k\in\mathscr{K}} p^P_{ik} F_{ijk} + \sum_{i\in\mathscr{N}_{\mathscr{S}}\cup\mathscr{N}_{\mathscr{T}}} \sum_{j\in\mathscr{N}_{\mathscr{T}}\cup\mathscr{N}_{\mathscr{D}}} \sum_{k\in\mathscr{K}} p^T_{ijk} F_{ijk} \tag{2.7a}$$

$$\text{s.t.} \sum_{j\in\mathscr{N}_{\mathscr{T}}} F_{ijk} = \sum_{j\in\mathscr{N}_{\mathscr{T}}} F_{jik}, \quad i\in\mathscr{N}_{\mathscr{T}},\, k\in\mathscr{K}, \tag{2.7b}$$

$$\sum_{j\in\mathscr{N}_{\mathscr{S}}\cup\mathscr{N}_{\mathscr{T}}} \gamma F_{jik} = d_i x_k days, \quad i\in\mathscr{N}_{\mathscr{D}},\, k\in\mathscr{K}, \tag{2.7c}$$

$$\sum_{k\in\mathscr{K}} Nutval_{kl} x_k \geq Nutreq_l, \quad l\in\mathscr{L}, \tag{2.7d}$$

$$x_{salt} = 5, \tag{2.7e}$$

$$x_{sugar} = 20, \tag{2.7f}$$

$$y \geq t, \tag{2.7g}$$

$$y = \hat{h}(\boldsymbol{x}), \tag{2.7h}$$

$$F_{ijk}, x_k \geq 0, \quad i,j\in\mathscr{N},\, k\in\mathscr{K}. \tag{2.7i}$$

The objective function consists of two components, procurement costs and transportation costs. Constraints (2.7b) are used to balance the network flow, namely to ensure that the inflow and the outflow of a commodity are equal for each transhipment node. Constraints (2.7c) state that flow into a delivery node has to be equal to its demand, which is defined by the number of beneficiaries times the daily ration for commodity k times the feeding days. Constraints (2.7d) guarantee an optimal solution that meets the nutrition requirements. Constraints (2.7e) and (2.7f) force the amount of salt and sugar to be 5 grams and 20 grams respectively. Constraint (2.7g) requires the food basket palatability (y), defined by means of a predictive model (2.7h), to be greater than a threshold (t). Lastly, non-negativity constraints (2.7i) are added for all commodity flows and commodity rations.

2.3.2 Dataset and predictive models

To evaluate the ability of our framework to learn and implement the palatability constraints, we use a simulator to generate diets with varying palatabilities. Each sample is defined by 25 features representing the amount (in grams) of all commodities that make up the food basket. We then

use a ground truth function to assign each food basket a palatability between 0 and 1, where 1 corresponds to a perfectly palatable basket, and 0 to an inedible basket. This function is based on suggestions provided by WFP experts. The data is then balanced to ensure that a wide variety of palatability scores are represented in the dataset. The final data used to learn the palatability constraint consists of 121,589 samples. Two examples of daily food baskets and their respective palatability scores are shown in Table 2.4. In this case study, we use a palatability lower bound (t) of 0.5 for our learned constraint.

The next step of the framework involves training and choosing the predictive model that best approximates the unknown constraint. The predictive models used to learn the palatability constraints are those discussed in Section 2.2, namely LR, SVM, CART, RF, GBM with decision trees as base-learners, and MLP with ReLU activation function.

Commodity	**Basket 1 Amount (g)**	**Basket 2 Amount (g)**
Dried skim milk	31.9	33.9
Chickpeas	–	75.7
Lentils	41	–
Maize meal	48.9	–
Meat	–	17.2
Oil	22	28.6
Salt	5	5
Sugar	20	20
Wheat	384.2	131.2
Wheat flour	–	261.3
Wheat soya blend	67.3	59.8
Palatability Score	0.436	0.741

Table 2.4: Two examples of daily food baskets.

2.3.3 Optimization results

The experiments are executed using `OptiCL` jointly with Gurobi v9.1 [78] as the optimization solver. Table 2.5 reports the performances of the predictive models evaluated both for the validation set and for the prescriptions after being embedded into the optimization model.The table also compares the performance of the optimization with and without the trust region. Runtimes are reported using an Intel i7-8665U 1.9 GHz CPU, 16 GB RAM (Windows 10 environment).

The column “Validation MSE" gives the Mean Squared Error (MSE) of each model obtained in cross-validation during model selection. While all scores in this column are desirably low, the MLP model significantly achieves the lowest error during this validation phase. The column “MSE" gives the MSE of the predictive models once embedded into the optimization problem to evaluate how well the predictions for the optimal solutions match their true palatabilities (computed using the simulator). It is found using 100 optimal solutions of the optimization model generated with different cost vectors. The MLP model exhibits the best performance (0.055) in this context, showing its ability to model the palatability constraint better than all other methods.

Model	**Validation MSE**	**MSE**	**MSE-TR**	**Time (SD)**	**Time-TR (SD)**
LR	0.046	0.256	0.042	0.003 (0.0008)	1.813 (0.204)
SVM	0.019	0.226	0.027	0.003 (0.0006)	1.786 (0.208)
CART	0.014	0.273	0.059	0.012 (0.0030)	7.495 (5.869)
RF	0.018	0.252	0.025	0.248 (0.1050)	30.128 (13.917)
GBM	0.006	0.250	0.017	0.513 (0.4562)	60.032 (41.685)
MLP	0.001	0.055	0.001	14.905 (41.764)	28.405 (23.339)

Table 2.5: Predictive models performances for the validation set (“Validation MSE"), and for the prescriptions after being embedded into the optimization model with (“MSE-TR") and without the trust region (“MSE"). The last two columns show the average computation time in seconds and its standard deviation (SD) required to solve the optimization model with (“Time-TR") and without the trust region (“Time").

Benefit of trust region. Table 2.5 shows that when the trust region is used (“MSE-TR"), the MSEs obtained by all models are now much closer to the results from the validation phase. This shows the benefit of using the trust region as discussed in Section 2.2.3 to prevent extrapolation. With the trust region included, the MLP model also exhibits the lowest MSE (0.001). The improved performance seen with the inclusion of the trust region does come at the expense of computation speed. The column “Time-TR" shows the average computation time in seconds and its standard deviation (SD) with trust region constraints included. In all cases, the computation time has clearly increased when compared against the computation time required without the trust region (column “Time"). This is however acceptable, as significantly more accurate results are obtained with the trust region.

Benefit of clustering. The large dataset used in this case study makes the use of the trust region expensive in terms of time required to solve the final optimization model. While the column selection algorithm described in Section 2.2.3 is ideal for significantly reducing the computation time, optimization models that require binary variables, either for embedding an ML model or to represent decision variables, cannot use the column selection algorithm. However, in this more general MIO case, it is possible to divide the dataset into clusters and solve in parallel an MIO for each cluster. By using parallelization, the total solution time can be expected to be equal to the longest time required to solve any single cluster's MIO. Contrary to column selection, the use of clusters can result in sub-optimal solutions; the trust region gets smaller with more clusters and prevents the model from finding solutions that are convex combinations of members of different clusters. However, as described in Section 2.2.3, optimal solutions that lie between clusters may in fact reside in low-density areas of the feature space that should not be included in the trust region. In this sense, the loss in optimality might actually coincide with more trustable solutions.

Figure 2-5 shows the effect of clusters in solving the model (2.7a-2.7i) with GBM as the predictive model used to learn the palatability constraint. K-means is used to partition the dataset into K clusters, and the reported values are averaged over 100 iterations. In the left graph, we report the maximum runtime distribution across clusters needed to solve the different MIOs in parallel. In the right graph, we have the distributions of optimality gap, *i.e.*, the relative difference between the optimal solution obtained with clusters compared to the solution obtained with no clustering. In this case study, the use of clusters significantly decreases the runtime (89.2% speed up with $K = 50$) while still obtaining near-optimal solutions (less then 0.25% average gap with $K = 50$). We observe that the trends are not necessarily monotonic in K. It is possible that a certain choice of K may lead to a suboptimal solution, whereas a larger value of K may preserve the optimal solution as the convex combination of points within a single cluster.

2.4 Case study: chemotherapy regimen design

In this case study, we extend the work of Bertsimas et al. [28] in the design of chemotherapy regimens for advanced gastric cancer. Late stage gastric cancer has a poor prognosis with limited treatment options [174]. This has motivated significant research interest and clinical trials [124].

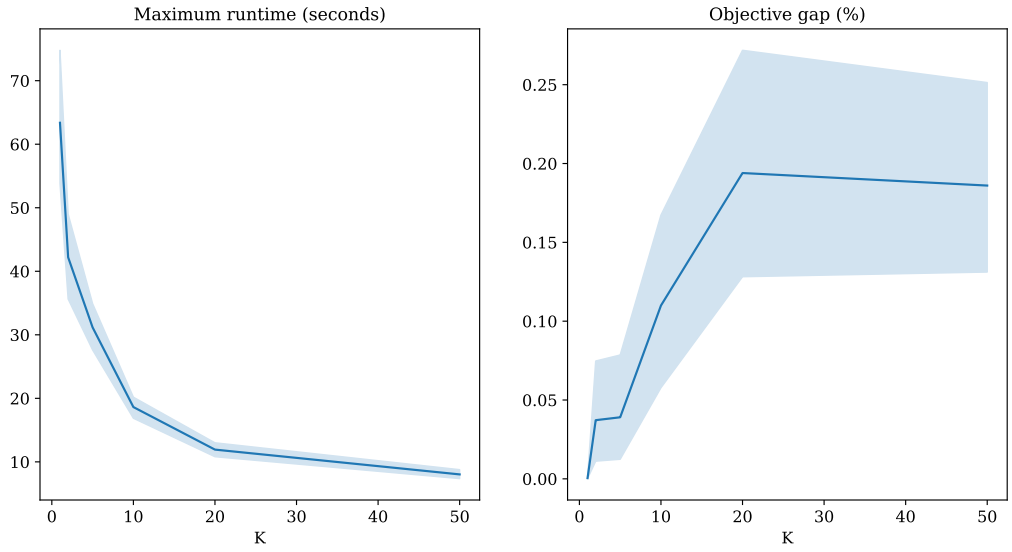


Figure 2-5: Effect of the number of clusters (K) on the computation time and the optimality gap across clusters, with bootstrapped 95% confidence intervals.

In Bertsimas et al. [28], the authors pose the question of algorithmically identifying promising chemotherapy regimens for new clinical trials based on existing trial results. They construct a database of clinical trial treatment arms which includes cohort and study characteristics, the prescribed chemotherapy regimen, and various outcomes. Given a new study cohort and study characteristics, they optimize a chemotherapy regimen to maximize the cohort's survival subject to a constraint on overall toxicity. The original work uses linear regression models to predict survival and toxicity, and it constrains a single toxicity measure. In this work we leverage a richer class of ML methods and more granular outcome measures. This offers benefits through higher performing predictive models and more clinically-relevant constraints.

Chemotherapy regimens are particularly challenging to optimize, since they involve multiple drugs given at potentially varying dosages, and they present risks for multiple adverse events that must be managed. This example highlights the generalizability of our framework to complex domains with multiple decisions and learned functions. The treatment variables in this problem consist of both binary and continuous elements, which are easily incorporated through our use of MIO. We have several learned constraints which must be simultaneously satisfied, and we also learn the objective function directly as a predictive model.

2.4.1 Conceptual model

The use of clinical trial data forces us to consider each cohort as an observation, rather than an individual, since only aggregate measures are available. Thus, our model optimizes a cohort's treatment. The contextual variables ($\boldsymbol{w}$) consist of various cohort and study summary variables. The inclusion of fixed, *i.e.*, non-optimization, features allows us to account for differences in baseline health status and risk across study cohorts. These features are included in the predictive models but then are fixed in the optimization model to reflect the group for whom we are generating a prescription. We assume that there are no unobserved confounding variables in this prescriptive setting.

The treatment variables ($\boldsymbol{x}$) encode a chemotherapy regimen. A regimen is defined by a set of drugs, each with an administration schedule of potentially varied dosages throughout a chemotherapy cycle. We characterize a regimen by drug indicators and each drug's average daily dose and maximum instantaneous dose in the cycle:

$$\begin{aligned}
\boldsymbol{x}_b^d &= \mathbb{I}(\text{drug } d \text{ is administered}),\\
\boldsymbol{x}_a^d &= \text{average daily dose of drug } d,\\
\boldsymbol{x}_i^d &= \text{maximum instantaneous dose of drug } d.
\end{aligned}$$

This allows us to differentiate between low-intensity, high-frequency and high-intensity, low-frequency dosing strategies. The outcomes of interest ($\boldsymbol{y}$) consist of overall survival, to be included as the objective (y_{OS}), and various toxicities, to be included as constraints ($y_i,\ i \in \mathscr{Y}_C$).

To determine the optimal chemotherapy regimen $\boldsymbol{x}$ for a new study cohort with characteristics

$\boldsymbol{w}$, we formulate the following MIO:

$$
\begin{aligned}
&\min_{\boldsymbol{x},\boldsymbol{y}} y_{OS} \\
&\text{s.t.} y_i \leq \tau_i, && i \in \mathscr{Y}_C, \\
&\quad y_i = \hat{h}_i(\boldsymbol{x}, \boldsymbol{w}), && i \in \mathscr{Y}_C, \\
&\quad y_{OS} = \hat{h}_{OS}(\boldsymbol{x}, \boldsymbol{w}), \\
&\quad \sum_d \boldsymbol{x}_b^d \leq 3, \\
&\quad \boldsymbol{x} \in \mathscr{X}(\boldsymbol{w}).
\end{aligned}
$$

In this case study, we learn the full objective. However, this model could easily incorporate deterministic components to optimize as additional weighted terms in the objective. We include one domain-driven constraint, enforcing a maximum regimen combination of three drugs.

The trust region, $\mathscr{X}(\boldsymbol{w})$, plays two crucial roles in the formulation. First, it ensures that the predictive models are applied within their valid bounds and not inappropriately extrapolated. It also naturally enforces a notion of “clinically reasonable" treatments. It prevents drugs from being prescribed at doses outside of previously observed bounds, and it requires that the drug combination must have been previously seen (although potentially in different doses). It is nontrivial to explicitly characterize what constitutes a realistic treatment, and the convex hull provides a data-driven solution that integrates directly into the model framework.

2.4.2 Dataset

Our data consists of 495 clinical trial arms from 1979-2012 [28]. We consider nine contextual variables, including the average patient age and breakdown of primary cancer site. We include several “dose-limiting toxicities" (DLTs) for our constraint set: Grade 3/4 constitutional toxicity, gastrointestinal toxicity, and infection, as well as Grade 4 blood toxicity. As the name suggests, these are chemotherapy side effects that are severe enough to affect the course of treatment. There are 28 unique drugs that appear in multiple arms of the training set, yielding 84 decision variables.

We apply a temporal split, training the predictive models on trial arms through 2008 and generating prescriptions for the trial arms in 2009-2012. The final training set consists of 320 obser-

vations, and the final testing set consists of 96 observations. The full feature set, inclusion criteria, and data processing details are included in Appendix A.4.

To define the trust region, we take the convex hull of the treatment variables ($\boldsymbol{x}$) on the training set. This aligns with the temporal split setting, in which we are generating prescriptions going forward based on an existing set of past treatment decisions. In general it is preferable to define the convex hull with respect to both $\boldsymbol{x}$ and $\boldsymbol{w}$ as discussed in Appendix A.2, but this does not apply well with a temporal split. Our data includes the study year as a feature to incorporate temporal effects, and so our test set observations will definitionally fall outside of the convex hull defined by the observed $(\boldsymbol{x}, \boldsymbol{w})$ in our training set.

2.4.3 Predictive models

Several ML models are trained for each outcome of interest using cross-validation for parameter tuning, and the best model is selected based on the validation criterion. We employ function learning for all toxicities, directly predicting the toxicity incidence and applying an upper bound threshold within the optimization model.

Based on the model selection procedure, overall DLT, gastrointestinal toxicity, and overall survival are predicted using GBM models. Blood toxicity and infection are predicted using linear models, and constitutional toxicity is predicted with a RF model. This demonstrates the advantage of learning with multiple model classes; no single method dominates in predictive performance. A complete comparison of the considered models is included in Appendix A.4.

2.4.4 Evaluation framework

We generate prescriptions using the optimization model outlined in Section 2.4.1, with the embedded model choices specified in Section 2.4.3. In order to evaluate the quality of our prescriptions, we must estimate the outcomes under various treatment alternatives. This evaluation task is notoriously challenging due to the lack of counterfactuals. In particular, we only know the true outcomes for observed cohort-treatment pairs and do not have information on potential unobserved combinations. We propose an evaluation scheme that leverages a “ground truth" ensemble (GT ensemble). We train several ML models using all data from the study. These models are not embedded in an

MIO model, so we are able to consider a broader set of methods in the ensemble. We then predict each outcome by averaging across all models in the ensemble. This approach allows us to capture the maximal knowledge scenario. Furthermore, such a “consensus" approach of combining ML models has been shown to improve predictive performance and is more robust to individual model error [21]. The full details of the ensemble models and their predictive performances are included in Appendix A.4.

2.4.5 Optimization results

We evaluate our model in multiple ways. We first consider the performance of our prescriptions against observed (given) treatments. We then explore the impact of learning multiple subconstraints rather than a single aggregate toxicity constraint. All optimization models have the following shared parameters: toxicity upper bound of 0.6 quantile (as observed in training data) and maximum violation of 25% for RF models. We report results for all test set observations with a feasible solution.

Table 2.6 reports the predicted outcomes under two constraint approaches: (1) constraining each toxicity separately (“All Constraints"), and (2) constraining a single aggregate toxicity measure (“DLT Only"). For each cohort in the test set, we generate predictions for all outcomes of interest under both prescription schemes and compute the relative change of our prescribed outcome from the given outcome predictions.

Benefit of prescriptive scheme. We begin by evaluating our proposed prescriptive scheme (“All Constraints") against the observed actual treatments. For example, under the GT ensemble scheme, 84.7% of cohorts satisfied the overall DLT constraint under the given treatment, compared to 94.1% under the proposed treatment. This yields an improvement of 11.10%. We obtain a significant improvement in survival (11.40%) while also improving toxicity limit satisfaction across all individual toxicities. Using the GT ensemble, we see toxicity satisfaction improvements between 1.3%-25.0%.

Benefit of multiple constraints. Table 2.6 also illustrates the value of enforcing constraints on each individual toxicity rather than as a single measure. When only constraining the aggregate

		All Constraints		DLT Only	
	Given (SD)	**Prescribed (SD)**	**% Change**	**Prescribed (SD)**	**% Change**
Any DLT	0.847 (0.362)	0.941 (0.237)	11.10%	0.906 (0.294)	6.90%
Blood	0.812 (0.393)	0.824 (0.383)	1.40%	0.706 (0.458)	-13.00%
Constitutional	0.953 (0.213)	1.000 (0.000)	4.90%	1.000 (0.000)	4.90%
Infection	0.882 (0.324)	0.894 (0.310)	1.30%	0.800 (0.402)	-9.30%
Gastrointestinal	0.800 (0.402)	1.000 (0.000)	25.00%	1.000 (0.000)	25.00%
Overall Survival	10.855 (1.939)	12.092 (1.470)	11.40%	12.468 (1.430)	14.90%

Table 2.6: Comparison of outcomes under given treatment regimen, regimen prescribed when only constraining the aggregate toxicity, and regimen prescribed under our full model.We report the mean and standard deviation (SD) of constraint satisfaction (binary indicator) and overall survival (months) across the test set. The relative change is reported against the given treatment.

toxicity measure (“DLT Only"), the resultant prescriptions actually have lower constraint satisfaction for blood toxicity and infection than the baseline given regimens. By constraining multiple measures, we are able to improve across all individual toxicities. The fully constrained model actually improves the overall DLT measure satisfaction, suggesting that the inclusion of these “sub-constraints" also makes the aggregate constraint more robust. This improvement does come at the expense of slightly lower survival between the “All" and “DLT Only" models (-0.38 months) but we note that incurring the individual toxicities that are violated in the “DLT Only" model would likely make the treatment unviable.

2.5 Discussion

Our experimental results illustrate the benefits of our constraint learning framework in data-driven decision making in two problem settings: food basket recommendations for the World Food Programme and chemotherapy regimens for advanced gastric cancer. The quantitative results show the improvement in predictive performance when incorporating the trust region and learning from multiple candidate model classes. We also see a benefit in incorporating multiple learned constraints over a single aggregate measure. Our framework scales to large problem sizes, enabled by efficient formulations and tailored approaches to specific problem structures. Our approach for efficiently learning the trust region also has broad applicability in one-class constraint learning.

We recognize several opportunities to further extend this framework. Our work naturally re-

lates to the causal inference literature and individual treatment effect estimation [11, 146]. These methods do not directly translate to our problem setting; existing work generally assumes highly structured treatment alternatives (*e.g.*, binary treatment vs. control) or a single continuous treatment (*e.g.*, dosing), whereas we allow more general decision structures. In future work, we are interested in incorporating ideas from causal inference to relax the assumption of unobserved confounders.

Additionally, our framework is dependent on the quality of the underlying predictive models. We constrain and optimize point predictions from our embedded models. This can be problematic in the case of model misspecification, a known shortcoming of "predict-then-optimize" methods [63]. We mitigate this concern in two ways. First, our model selection procedure allows us to obtain higher quality predictive models by capturing several possible functional relationships. Second, our inclusion of the violation limit in constrained ensemble models allows us to directly parametrize how conservative our predictions are and our robustness to the predictions of individual learners. This concept could be extended to a more general ensemble, in which we embed multiple separate models for an outcome of interest and enforce the constraint over some subset of these models. In future work, there is an opportunity to incorporate ideas from robust optimization to directly account for prediction uncertainty in the constraints.

In this work, we present a unified framework for optimization with learned constraints that leverages both ML and MIO for data-driven decision making. Our work flexibly learns problem constraints and objectives with supervised learning, and incorporates them into a larger optimization problem of interest. We also learn the trust region, providing more credible recommendations and improving predictive performance, and accomplish this efficiently using column generation and unsupervised learning. The generality of our method allows us to tackle quite complex decision settings, such as chemotherapy optimization, but also includes tailored approaches for more efficiently solving specific problem types. Finally, we implement this as a Python software package (`OptiCL`) to enable practitioner use. We envision that `OptiCL`'s methodology will be added to state-of-the-art optimization modeling software packages.

Chapter 3

Interpretable clustering: an optimization approach

Abstract

State-of-the-art clustering algorithms provide little insight into the rationale for cluster membership, limiting their interpretability. In complex real-world applications, the latter poses a barrier to machine learning adoption when experts are asked to provide detailed explanations of their algorithms' recommendations. We present a new unsupervised learning method that leverages Mixed Integer Optimization techniques to generate interpretable tree-based clustering models. Utilizing a flexible optimization-driven framework, our algorithm approximates the globally optimal solution leading to high quality partitions of the feature space. We propose a novel method which can optimize for various clustering internal validation metrics and naturally determines the optimal number of clusters. It successfully addresses the challenge of mixed numerical and categorical data and achieves comparable or superior performance to other clustering methods on both synthetic and real-world datasets while offering significantly higher interpretability.

3.1 Introduction

Clustering is the unsupervised classification of patterns, observations, data items, or feature vectors, into groups. The clustering problem has been addressed in many machine learning contexts where there is no clear outcome of interest, such as data mining, document retrieval, image segmentation, and pattern classification; this reflects its broad appeal and usefulness in exploratory data analysis [84]. In many such problems, there is little prior information available about the data, and the decision-maker must make as few assumptions about the data as possible. It is un-

der these restrictions that clustering methodology is particularly appropriate for the exploration of relationships between observations to make an assessment, perhaps preliminary, of their structure.

Unlike supervised classification, there are no class labels and thus no natural measure of accuracy. Instead, the goal is to group objects into clusters based only on their observable features, such that each cluster contains objects with similar properties and different clusters have distinct features. There have been numerous approaches to generating these clusters. Partitional methods such as K-means [114] provide a single partition of the data into a fixed number of clusters; these methods have been improved by new initialization methods in recent decades [9]. Hierarchical methods produce a nested series of partitions [149] based on a distance metric. Other more sophisticated methods include model-based clustering [84] and density-based clustering [65] which are better able to capture clusters of irregular shape or varied density.

The end product of a clustering algorithm is a partition of the dataset. In some cases, this final cluster assignment is sufficient for the machine learning purpose, such as when one wants to simply assess the separability of the data points into distinct clusters or use it as a preprocessing step in certain prediction tasks. However, in many other decision-making applications, there is a need to interpret the resulting clusters and characterize their distinctive features in a compact form [70]. For example, consider a medical setting in which we seek to group similar patients together to understand subgroups within a patient base. In this application, it is critical to understand how the resulting clusters differ, whether by demographics, diagnoses, or other factors.

While the importance of cluster interpretability is well-understood, there has been limited success in addressing the issue [57]. None of the clustering algorithms described above were constructed with a goal of interpretability in the original feature space. They therefore require a post-processing step to synthesize the cluster meanings. The notion of cluster representation was introduced by Duran and Odell [61] and was subsequently studied by Diday and Simon [55] and Stepp and Michalski [152]. The representation of a cluster of points by its centroid has been popular across various applications [135]. This works well when the clusters are compact or isotropic, but fails when the clusters are elongated or non-isotropic [91]. These clusters can be better characterized computing additional metrics, such as the variance in each dimension. However, this increases the number of summary statistics used for each cluster and creates a high burden in interpretation, especially when the number of features grows large. Another common approach is the

visualization of clusters on a two-dimensional graph using Principle Component Analysis (PCA) projections [93, 137]. However, in reducing the dimensionality of the feature space, PCA obscures the relationship between the clusters and the original variables.

Tree-based supervised learning methods, such as CART, [40] are a natural fit for problems that prioritize interpretability, since their feature splits and decision paths offer insight into the differentiating features between members in each leaf. Most recursive partitioning algorithms generate trees in a top-down, greedy manner, which means that each split is selected in isolation without considering its effect on subsequent splits in the tree. Bertsimas and Dunn [23], Bertsimas, D. and Dunn, J. [34] have proposed a new algorithm which leverages modern mixed-integer optimization (MIO) techniques to form the entire decision tree in a single step, allowing each split to be determined with full knowledge of all other splits. The Optimal Classification Trees (OCT) algorithm enables the construction of decision trees for classification and regression that have performance comparable with state-of-the-art methods such as random forests and gradient boosted trees without sacrificing the interpretability offered by a single tree.

A general hybrid approach can leverage such methods by first running a partitional or hierarchical clustering method and using the resulting assignments as class labels. The data can then be fit using a classification tree, in which each leaf is given a cluster label based on the most common assignment of observations in that leaf, and the decision paths leading to each cluster's leaves give insight into the differentiating features [91]. Hancock et al. [83] use decision trees to interpret and refine hierarchical clustering results for global sea surface temperatures. While these trees give an explicit delineation of cluster attributes, the methods involve a two-step process of first building the clusters and subsequently identifying their differentiating features. Thus, the main clustering mechanism utilizes a different architecture compared to the decision tree which might be hard to capture with univariate feature splits.

Several algorithms have been proposed to build interpretable clusters, where interpretability is a consideration during cluster creation rather than considered as a later analysis step. Chavent et al. [44] presented a method that constructs binary clustering trees characterized by a novel transformation of the feature space. Further efforts focused on alternative measures for feature selection in the transformation function as well as new algorithmic implementation schemes [16]. In both of these cases, the feature space transformation involved in these methods takes a toll

on interpretability. Other researchers have proposed methods to construct decision trees in the original feature space, which more closely matches our objective. Liu et al. [108] introduced the idea of translating a clustering problem to a supervised problem that is amenable to decision tree construction. A modified purity criterion is used to evaluate splits in a way that identifies dense regions as well as sparse regions. However, this method requires additional pre-processing through the introduction of synthetic data in order to create a binary classification setting. Blockeel et al. [37] also proposed a general top-down tree induction framework with applicability to clustering ("Predictive Clustering Trees") as well as other supervised learning tasks. Fraiman et al. [72] developed another clustering algorithm, "Clustering using unsupervised binary trees" (CUBT), which forms greedy splits to optimize a cluster heterogeneity measure. Though these algorithms make progress towards the goal of constructing clusters directly using trees, they both employ a greedy splitting approach and do not offer flexibility in the choice of cluster validation criterion.

The need for accurate and interpretable machine learning methods is undoubtedly present, being voiced even from regulatory organizations such as the European Union [75]. Even though tree-based methods have been introduced, no existing interpretable unsupervised learning algorithm can accurately partition the feature space both for numerical and categorical data.

3.1.1 Contributions

Motivated by the limitations of existing solutions to interpretable clustering, we develop a novel tree-based unsupervised learning method that leverages traditional optimization and machine learning techniques to obtain interpretable clusters with comparable or superior performance when compared to existing algorithms. Our contributions are as follows:

1. We provide an MIO formulation of the unsupervised learning problem that leads to the creation of globally optimal clustering trees, motivating our new algorithm *Interpretable Clustering via Optimal Trees* (ICOT). Our method builds upon the OCT algorithm and extends it to the unsupervised setting. In ICOT, interpretability is taken into consideration during cluster creation rather than considered as a later analysis step.

2. We provide an implementation of our method with an iterative coordinate-descent approach that scales to larger problems, well-approximating the globally optimal solution. We use

widely two established validation criteria, the Silhouette Metric [141] and the Dunn Index [59], as the algorithm's objective function. We propose additional techniques that leverage the geometric principles of cluster creation to improve the algorithm's efficiency. Furthermore, we introduce sampling heuristics that recover fast, high-quality solutions in our empirical experiments and provide a complexity analysis of the local search procedure for one iteration of the algorithm.

3. We develop our algorithm in a way such that tuning of the tree's complexity is redundant. This is enabled by the fact that our loss functions take into account both intra-cluster density as well as inter-cluster separation. The user can optionally tune the algorithm by selecting the maximum depth of the tree and the minimum number of observations in each cluster.

4. We propose a solution to the incorporation of both mixed numerical and categorical data. Our re-weighted distance measure prevents a single variable type from dominating the distance calculation and allows users to optionally tune the balance the two types of covariates.

5. We evaluate the performance of our method against various clustering approaches across synthetic datasets from the Fundamental Clustering Problems Suite (FCPS) [159] which offer different levels of variance and compactness. We demonstrate ICOT's superior performance against a two-step supervised learning method across both the Silhouette Metric and Dunn Index, offering a 27.8% and 352.7% score improvement respectively. We also compare ICOT against several state-of-the-art methods that represent various clustering approaches, namely partitional, hierarchical, model-based, and density-based clustering. We find that ICOT is competitive against these methods across multiple internal validation criteria.

6. We provide examples of how the algorithm can be used in real-world settings. We perform clustering on patients at risk of cardiovascular disease from the Framingham Heart Study (FHS) dataset [54, 69] to identify similar patient profiles and group economic profiles of European countries during the Cold War [99]. Through these experiments, we illustrate the effect of varying key parameters in the ICOT algorithm. We also compare ICOT to other state-of-the-art algorithms in the FHS experiment and to CUBT in the economic profile experiment. We discuss the interpretability of the methods as well as their performance on

the internal validation criteria.

7. Finally, we test the capability of the algorithm to scale to large problem instances using both the FCPS as well as real-world data from a Boston-based bike sharing program. We demonstrate that our suggested heuristic techniques do not significantly impact the quality of the recovered solutions. In addition, our experiments illustrate that ICOT can efficiently handle datasets of sizes up to hundreds of thousands of observations.

The structure of the paper is as follows. In Section 3.2, we formulate the problem of optimal tree creation within an MIO framework. Section 3.3 provides a comprehensive description of the algorithm implementation. In Sections 3.4 and 3.5, we conduct a range of experiments using synthetic and real-world datasets to evaluate the performance and interpretability of our method compared to other state-of-the-art algorithms. In Section 3.6, we investigate the effect of our scaling methods on runtime and solution quality. In Section 3.7, we discuss the key findings from our work and in Section 3.8 we include our concluding remarks.

3.2 MIO formulation

In this section, we present an MIO approach which allows us to construct globally optimal tree-based models in an unsupervised learning setting. In Section 3.2.1, we provide an overview of the MIO framework introduced by Bertsimas and Dunn [23], Bertsimas, D. and Dunn, J. [34]. Section 3.2.2 introduces the validation criteria that are used as objective functions in the optimization problem. In Section 3.2.3, we outline the complete ICOT formulation for one of the loss functions considered.

3.2.1 The OCT framework

The OCT algorithm formulates tree construction using MIO which allows us to define a single problem, as opposed to the traditional recursive, top-down methods that must consider each of the tree decisions in isolation. It allows us to consider the full impact of the decisions being made at the top of the tree, rather than simply making a series of locally optimal decisions, avoiding the need for pruning and impurity measures.

We are given the training data $(\mathbf{X}, \mathbf{Y})$, containing n observations $(\mathbf{x_i}, y_i)$, $i = 1, \ldots, n$, each with p features and a class label $y_i \in \{1, \ldots, K\}$ as an indicator of which of the K potential labels is assigned to point i. We assume without loss of generality that the values of each training vector are normalized such that $\mathbf{x_i} \in [0,1]^p$. A decision tree recursively partitions the feature space to identify a set of distinct, hierarchical regions that form a classification tree. The final tree $\mathscr{T}$ is comprised of nodes that can be categorized in:

- Branch Nodes: Nodes $t \in \mathscr{T}_{\mathscr{B}}$ apply a split with parameters $\mathbf{a}$ and b. For observation i, if the corresponding vector $\mathbf{x_i}$ satisfies the relation $\mathbf{a}^T\mathbf{x_i} < b$, the point will follow the left branch from the node. Otherwise it takes the right branch.

- Leaf Nodes: Nodes $t \in \mathscr{T}_{\mathscr{L}}$ assign a class to all the points that fall into them. Each leaf node is characterized by one class which is generally determined by the most frequently occurring class among the observations that belong to it.

First, we formally define the constraints that construct the decision tree. We use the notation $p(t)$ to refer to the parent node of node t, and $A(t)$ to denote the set of ancestors of node t. We define the split applied at node $t \in \mathscr{T}_{\mathscr{B}}$ with variables $\mathbf{a}_t \in \mathbb{R}^p$ and $b_t \in \mathbb{R}$. The vector $\mathbf{a}_t$ indicates which variable is chosen for the split, meaning that $a_{jt} = 1$ for the variable j used at node t. b_t gives the threshold for the split, which is between $[0,1]$ after normalization of the feature vector. If a branch node does not apply a split, then we model this by setting $\mathbf{a}_t = \mathbf{0}$ and $b_t = 0$. Together, these form the constraint $\mathbf{a}_t^T x < b_t$. The indicator variables d_t are set to 1 for branch nodes and 0 for leaf nodes. Using the above variables, we introduce the following constraints that allows us to model the tree structure (for a detailed analysis of the constraints, see Bertsimas and Dunn [23]):

$$\sum_{j=1}^{p} a_{jt} = d_t, \ \forall t \in \mathscr{T}_{\mathscr{B}}, \tag{3.1}$$

$$0 \leq b_t \leq d_t, \ \forall t \in \mathscr{T}_{\mathscr{B}}, \tag{3.2}$$

$$a_{jt} \in \{0,1\}, \ j = 1, \ldots, p, \ \forall t \in \mathscr{T}_{\mathscr{B}} \tag{3.3}$$

We next enforce the hierarchical structure of the tree. Branch nodes are allowed to apply a split

only if their parent nodes apply a split:

$$d_t \leq d_{p(t)},\ \forall t \in \mathscr{T}_{\mathscr{B}} \backslash \{1\} \tag{3.4}$$

Next we present the corresponding constraints that track the allocation of points to leaves. For this purpose, we introduce the indicator variables $z_{it} = \mathbb{1}\{x_i \text{ is in node } t\}$ and $l_t = \mathbb{1}\{\text{leaf } t \text{ contains any points}\}$. We let N_{min} be a constant that defines the minimum number of observations required in each leaf. We apply the following constraints:

$$z_{it} \leq l_t,\ \forall t \in \mathscr{T}_{\mathscr{L}}, \tag{3.5}$$

$$\sum_{i=1}^{n} z_{it} \geq N_{min} l_t,\ \forall t \in \mathscr{T}_{\mathscr{L}} \tag{3.6}$$

We also enforce each point to belong to exactly one leaf:

$$\sum_{t \in \mathscr{T}_{\mathscr{L}}} z_{it} = 1,\ i = 1, \ldots, n \tag{3.7}$$

Finally, we introduce constraints that force the assignments of observations to leaves to obey the structure of the tree given by the branch nodes. We want to apply a strict inequality for points going to the lower leaf. To accomplish this, we define the vector $\varepsilon \in \mathbb{R}^p$ as the smallest separation between two observations in each dimension p, and ε_{max} as the maximum over this vector.

$$a_m^\intercal x_i \geq b_t - (1 - z_{it}),\ i = 1, \ldots, n, \quad \forall t \in \mathscr{T}_{\mathscr{B}}, \quad \forall m \in A_R(t) \tag{3.8}$$

$$a_m^\intercal (x_i + \varepsilon) \leq b_t + (1 + \varepsilon_{max})(1 - z_{it}),\ i = 1, \ldots, n, \quad \forall t \in \mathscr{T}_{\mathscr{B}}, \quad \forall m \in A_L(t) \tag{3.9}$$

In the classification setting the objective function of MIP formulation is comprised of two components, prediction accuracy and tree complexity. The tradeoff between those two parameters is controlled by the complexity parameter α. Given the training data $(\mathbf{x_i}, y_i)$, $i = 1 \ldots n$, a general formulation of the objective function is the following:

$$\underset{T}{\text{minimize}} \quad R_{xy}(T) + \alpha|T|$$

where $R_{xy}(t)$ is a loss function assessed on training data and $|T|$ is the number of branch nodes in the tree T.

The above model can be used as an input for an MIO solver. Empirical results suggest that such a model leads to optimal solutions in minutes when the maximum depth of the tree is small (approximately 4). Effectively, the rate of finding solutions is directly dependent to the number of binary variables z_{it} and therefore a faster implementation was needed for more complex problems. For this reason, the authors introduced the idea of warm starts as the initial starting point of the method. Using a high-quality integer feasible solution as a warm start increases the speed of the algorithm and provides a strong initial upper bound on the final solution. In addition, heuristics, like local search, allow a further speed up as shown in Bertsimas and Dunn [23], Bertsimas, D. and Dunn, J. [34] that leads to a good approximation of the optimal solution.

3.2.2 Loss functions for cluster quality

Clustering validation, the evaluation of the quality of a clustering partition [118], has long been recognized as one of the vital issues essential to the success of a clustering application [109]. External clustering validation and internal clustering validation are the two main categories of clustering quality metrics. The main difference lies in whether or not external labels are used to assess the clusters; internal measures evaluate the goodness of a clustering structure without respect to ground-truth labels [102]. An example of external validation measure is entropy, which evaluates the "purity" of clusters based on the given class labels [171]. True class labels are not present in real-world datasets, and thus these cases necessitate the use of internal validation measures for cluster validation.

We will consider two internal validation measures as loss functions for our MIO formulation of our problem. The chosen loss functions consider the global assignment of observations to clusters. The score of a clustering assignment depends on both the compactness of the observations within a single cluster, as well as its separation from observations in other clusters. Compactness measures how closely related the objects in a cluster are. Separation measures how distinct a cluster is

from other clusters. Several internal validation metrics have been proposed to balance these two objectives [109]. Two common criteria, the Silhouette Metric and Dunn Index, are outlined below.

Silhouette Metric The Silhouette Metric introduced by Rousseeuw [141] compares the distance from an observation to other observations in its cluster relative to the distance from the observation to other observations in the second closest cluster. The Silhouette Metric for observation i is computed as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}, \tag{3.10}$$

where $a(i)$ is the average distance from observation i to the other points in its cluster, and $b(i)$ is the average distance from observation i to the points in the second closest cluster. In other words, $b(i) = \min_k b(i,k)$ where $b(i,k)$ is the average distance of i to points in cluster k, minimized over all clusters k other than the cluster that point i is assigned to. From this formula it follows that $-1 \leq s(i) \leq 1$.

When $s(i)$ is close to 1, one may infer that the i^{th} sample has been "well-clustered", i.e. it was assigned to an appropriate cluster. If observation i has score close to 0, it suggests that it could also be assigned to the nearest neighboring cluster with similar quality. If $s(i)$ is close to -1, one may argue that such a sample has been assigned to the wrong partition. These individual scores can be averaged to reflect the quality of the global assignment.

$$SM = \frac{1}{n} \sum_{i=1}^{n} s(i), \tag{3.11}$$

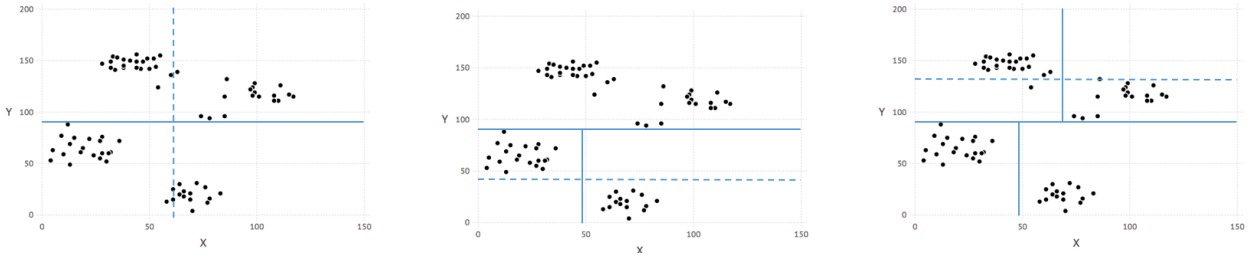
Dunn Index The Dunn Index [59] characterizes compactness as the maximum distance between observations in the same cluster, and separation as the minimum distance between two observations in different clusters. The metric is computed as the ratio of the minimum inter-cluster separation to the maximum intra-cluster distance.

$$DI = \frac{\min\limits_{1 \leq i < j \leq m} \delta(C_i, C_j)}{\max\limits_{1 \leq k \leq m} \Delta_k}, \tag{3.12}$$

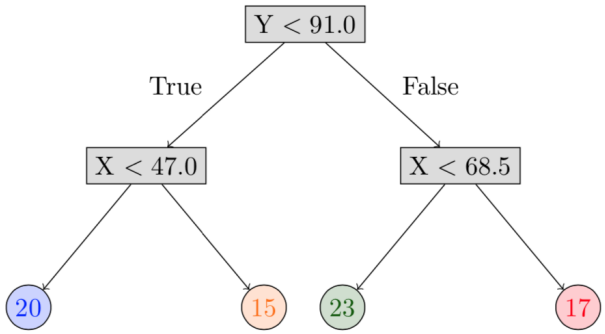
where we let the maximum distance of cluster C be denoted by Δ_C and the distance between clusters i and j be denoted by $\delta(C_i, C_j)$. If the dataset contains compact and well-separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. Thus, large values of the metric correspond to better partitions and signify that the distance between clusters is large relative to the distance between points within a cluster.

We provide an example to illustrate how an internal validation criterion can be used to geometrically partition the space through a decision tree. In Figure 3-1, we cluster observations from the Ruspini dataset [143] using the Silhouette Metric. In Figure 3-1a, the algorithm identifies the best candidate splits on both features, x_1 and x_2, at the root node, and then compares their resultant cluster scores, as measured by the Silhouette Metric. The x_2 split provides a better cluster assignment, so this split is chosen as denoted by the solid line. After the first data partition, splits are considered for each of the child nodes, which corresponds to further separating the lower and upper halves of the graph. Upon identification of candidate x_1 and x_2 splits on the left child node, the x_1 split is chosen based on the Silhouette Metric of the global cluster assignment, as shown in Figure 3-1b. The process is then completed for the right child node, and an x_1 split is also chosen here in Figure 3-1c. Now, each of the four leaves is evaluated, which corresponds to exploring splits in the four quadrants defined by the solid blue lines. There are no splits within any of these four leaves that improve the overall score of the clustering assignment, so the tree construction is complete. The final tree is shown in Figure 3-1d. The resultant tree provides a final partition which clearly elucidates the distinguishing features of each group. We note that this example demonstrates a *greedy* tree construction. In the ICOT algorithm, all splits would be subsequently reoptimized with respect to the overall tree. However, in this case the greedy tree is able to provide the optimal partition.

Note that both of our considered criteria require the definition of at least two clusters since they both involve a pairwise distance computation between clusters to measure separation. As a result, calculations for the null-case are not considered. The determination of the best internal validation criterion for a given dataset remains an open question in the field of unsupervised learning theory [109]. As stated in [82], the Dunn Index is more computationally expensive and more sensitive to noisy data compared to the Silhouette Metric. It is also less robust to outliers compared to the



(a) Root Node (b) Left Child Node (c) Right Child Node



(d) Final Tree

Figure 3-1: An example of a clustering tree built on the Ruspini dataset.

Silhouette Metric which averages an observation-based score for the global assignment. However, empirical results suggest that the Dunn Index has superior performance in returning intuitive partitions of the data when they are well-separated.

3.2.3 The ICOT formulation

The OCT framework needs to be modified to address an unsupervised learning task. We present changes in the original MIO formulation of OCT to be able to partition the data space into distinct clusters following the same structure and notation as in Section 3.2.1. We outline in detail the model for the Silhouette Metric loss function. The Dunn Index formulation follows closely and is thus omitted. There are two primary modifications in the ICOT formulation compared to the OCT:

1. The objective function is comprised solely by the chosen cluster quality criterion, such as the Silhouette Metric, and does not include any penalty for the tree complexity. The separation component of the validation criterion naturally controls the complexity of the tree and thus for the ICOT formulation the complexity parameter is rendered redundant.

2. Each leaf of the tree is equivalent to a cluster. Observations in different leaves are not allowed to belong to the same cluster.

The objective of the new formulation is to maximize the Silhouette Metric (SM) of the overall partition. The Silhouette Metric quantifies the difference in separation between a point and points in its cluster, versus the separation between that point and points in the second closest cluster.

Let d_{ij} be the distance (i.e. Euclidean) of observation i from observation j. We define K_t to be number of points assigned assigned to cluster t.

$$K_t = \sum_{i=1}^{n} z_{it},\ \forall t \in \mathscr{T}_{\mathscr{L}} \tag{3.13}$$

We define c_{it} to be the average distance of observation i from cluster t:

$$c_{it} = \frac{1}{K_t} \sum_{j=1}^{n} d_{ij} z_{jt},\ \forall i = 1,\ldots,n,\ t \in \mathscr{T}_{\mathscr{L}}. \tag{3.14}$$

We define r_i to be the average distance of observation i from all the points assigned in the same cluster:

$$r_i = \sum_{t \in \mathscr{T}_{\mathscr{L}}} c_{it} z_{it},\ \forall i = 1,\ldots,n. \tag{3.15}$$

We then let q_i denote the minimum average distance of observation i to the observations from the next closest cluster. We define auxiliary variables γ_{it} to enforce this constraint, such that γ_{it} an indicator of whether t is the second closest cluster for observation i.

$$q_i \geq \sum_{t \in \mathscr{T}_{\mathscr{L}}} \gamma_{it} c_{it},\ i = 1,\ldots,n. \tag{3.16}$$

$$\sum_{t \in \mathscr{T}_{\mathscr{L}}} \gamma_{it} = 1,\ i = 1,\ldots,n. \tag{3.17}$$

$$\gamma_{it} \leq M(1 - z_{it}),\ i = 1,\ldots,n,\ \forall t \in \mathscr{T}_{\mathscr{L}}. \tag{3.18}$$

Finally, to define the Silhouette Metric of observation i, we will need the maximum value between r_i and q_i which normalizes the metric.

$$m_i \geq r_i,\ i = 1,\ldots,n. \tag{3.19}$$

$$m_i \geq q_i,\ i = 1, \ldots, n. \tag{3.20}$$

The score for the Silhouette Metric for each observation is computed as $s(i)$ and the overall score for the clustering assignment is then the average overall all the Silhouette Metric scores from the training population:

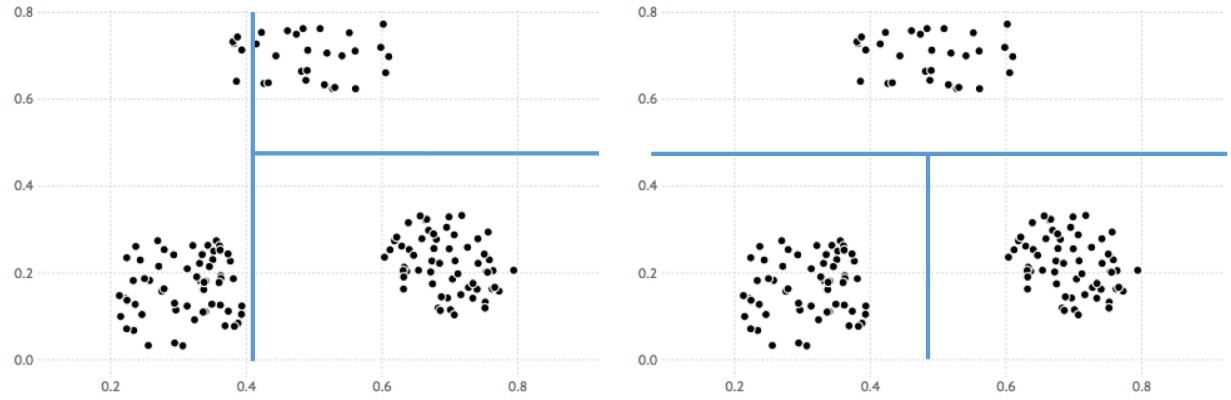
$$s_i = \frac{q_i - r_i}{m_i},\ i = 1, \ldots, n. \tag{3.21}$$

$$SM = \frac{1}{n} \sum_{i=1}^{n} s_i. \tag{3.22}$$

Putting all of this together gives the following MIO formulation for the ICOT model:

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & -\frac{1}{n}\sum_{i=1}^{n} s_i \\
\text{subject to} \quad & s_i = \frac{q_i - r_i}{m_i}, && i = 1,\ldots,n, \\
& m_i \geq q_i, && i = 1,\ldots,n, \\
& m_i \geq r_i, && i = 1,\ldots,n, \\
& q_i \geq \sum_{t \in \mathscr{T}_{\mathscr{L}}} \gamma_{it} c_{it}, && i = 1,\ldots,n, \\
& \sum_{t \in \mathscr{T}_{\mathscr{L}}} \gamma_{it} = 1, && i = 1,\ldots,n, \\
& \gamma_{it} \leq M(1 - z_{it}), && i = 1,\ldots,n,\ \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& r_i = \sum_{\forall t \in \mathscr{T}_{\mathscr{L}}} c_{it} z_{it}, && i = 1,\ldots,n, \\
& c_{it} = \frac{1}{K_t}\sum_{j=1}^{n} d_{ij} z_{jt}, && i = 1,\ldots,n, \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& K_t = \sum_{i=1}^{n} z_{it} && \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& \sum_{j=1}^{p} a_{jt} = d_t, && \forall t \in \mathscr{T}_{\mathscr{B}}, \\
& 0 \leq b_t \leq d_t, && \forall t \in \mathscr{T}_{\mathscr{B}}, \\
& d_t \leq d_{p(t)}, && \forall t \in \mathscr{T}_{\mathscr{B}} \backslash \{1\}, \\
& z_{it} \leq l_t, && \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& \sum_{i=1}^{n} z_{it} \geq N_{min} l_t, && \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& \sum_{t \in \mathscr{T}_{\mathscr{L}}} z_{it} = 1,\ i = 1,\ldots,n, \\
& a_m^{\mathsf{T}} x_i \geq b_t - (1 - z_{it}), && i = 1,\ldots,n,\ \forall t \in \mathscr{T}_{\mathscr{B}},\ m \in A_R(t), \\
& a_m^{\mathsf{T}}(x_i + \varepsilon) \leq b_t + (1 + \varepsilon_{max})(1 - z_{it}), && i = 1,\ldots,n,\ \forall t \in \mathscr{T}_{\mathscr{B}},;m \in A_L(t), \\
& a_{jt}, d_t \in \{0,1\}, && j = 1,\ldots,p,\ \forall t \in \mathscr{T}_{\mathscr{B}}, \\
& z_{it}, l_t \in \{0,1\}, && i = 1,\ldots,p,\ \forall t \in \mathscr{T}_{\mathscr{L}}, \\
& \gamma_{it} \in \{0,1\}, && i = 1,\ldots,n,\ \forall t \in \mathscr{T}_{\mathscr{L}}.
\end{aligned}
$$

Figure 3-2 illustrates the benefit of an optimization framework over greedy tree construction. The synthetic dataset seen in the figure has two dense lower regions and one less dense upper region. In a greedy approach, the first split separates the lower clusters and cuts through the upper cluster. While it is clearly better to split horizontally first (since it does not split a region), a greedy algorithm chooses the split without consideration of the possibility of future splits. Therefore, if the tree can only make one split, it is better to separate the lower clusters since they have such high density. ICOT's optimization approach considers the global tree structure, avoiding such pitfalls and identifying the true optimal partition. It starts by making a horizontal split and subsequently separates the high-density lower regions without cutting through the upper cluster. A globally optimal partition has Silhouette Metric score equal to 0.758 whereas the greedy tree yields only 0.688.



(a) Greedy Tree Approach (b) ICOT Approach

Figure 3-2: An illustration in a synthetic example of a local optimum that might be identified by a greedy unsupervised learning algorithm.

3.3 Algorithm overview

In this section, we outline the practical details of the algorithm implementation. Section 3.3.1 describes ICOT's coordinate-descent algorithm that approximates the globally optimal solution in an efficient and intuitive manner. Section 3.3.2 addresses the challenge of computing distance scores in the presence of mixed numerical and categorical variables and introduces a solution for appropriately handling distance in this setting. Finally, in Section 3.3.3 we propose heuristics in

our algorithm implementation which leverage the underlying structure of the data to more quickly traverse the search space and identify high-quality solutions.

3.3.1 Coordinate-descent implementation

The MIO formulation provides the optimization framework for our problem solving approach. In practice, the algorithm is implemented using a coordinate-descent procedure which allows it to scale to much higher dimensions than directly solving the optimization problem. The implementation provides a good approximation of the optimal solution while still abiding by the same core principles of the original formulation.

ICOT initializes a greedy tree and subsequently runs a local search procedure until the objective value, a cluster quality measure, converges. This process is repeated from many different starting greedy trees, generating many candidate clustering trees. The final tree is chosen as the one with the highest cluster quality score across all candidate trees. This single tree is returned as the output of the algorithm.

The initial greedy tree is constructed from a single root node. A split is made on a randomly chosen feature by scanning over all potential thresholds for splitting observations into the lower and upper leaves. At each candidate split, we compute the global score for the potential assignment. We choose the split threshold that gives the highest score and update the node to add the split if this score improves upon the global score of the current assignment. We perform the same search for each leaf that gets added to the tree, continuing until either the maximum tree depth is reached or no further improvement in our objective value is achieved through further splitting on a leaf.

Following the creation of the greedy tree, a local search procedure is performed to optimize the clustering assignment. Tree nodes are visited in a randomly chosen order, and various modifications are considered. A branch node has two options; it can be deleted, in which case it is replaced with either its lower or upper subtree, or a new split can be made at the node using a different feature and threshold. A leaf node can be further split into two leaves. At each considered node, the algorithm finds the best possible change and updates the tree structure only if it improves the objective from its current value. All nodes get added back to the list of nodes to search once an improvement has been found. The algorithm terminates when the objective value converges. The

algorithm is explained further in Algorithm 1.

Algorithm 1 ICOT Algorithm.

Input: Feature vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n$
Output: Cluster assignments $y^1, \ldots, y^n$

Initialize a greedy tree, with clusters $c_1, \ldots, c_K$ and loss l_0.
Indices to search: $S = \{1, \ldots, K\}$; Loss: $l = l_0$.
while S not empty **do**
    for all $k \in S$ **do**
        if C_k is leaf node **then**
            Find best possible new split with loss $\hat{l}$.
        else
            Find best possible node modification, either through a different split or split deletion, with loss $\hat{l}$.
        end if
        if $\hat{l} < l$ **then**
            Update tree and add all leaves to S. $l \leftarrow \hat{l}$.
        else
            Remove k from S.
        end if
    end for
end while

The user can specify to optimize either the Silhouette Metric or Dunn Index described in Section 3.2.2. These metrics penalize low separation, which naturally limits the depth of the tree. In traditional tree-based algorithms such as CART or OCT, the loss function improves with successive tree splits. Thus, these methods require a pruning step or additional parameter, such as a complexity penalty of maximum depth, to control the tree size. ICOT does not require the explicit control of tree size due to this natural balance between separation and compactness in the cluster quality metrics. This eliminates the need for setting an explicit K parameter, which is typically required in both partitional and hierarchical clustering methods. The tree continues to split until further splits no longer improve the quality of the overall assignment, and so the final number of leaves represents the optimal number of clusters.

The user can enforce further structure on the tree through setting the optional minimum bucket parameter, N_C. This controls the minimum number of observations that are required in each leaf and effectively in each cluster. Note that there is not a monotonic relationship between the magnitude of N_C and the number of leaves (clusters) generated by the algorithm. Smaller minimum

buckets may lead to smaller cluster counts due to the positive effect of isolated outlier clusters on the metrics; overfitting is difficult to quantify in an unsupervised learning setting because there is no ground truth to compare against, and thus the metrics do not naturally penalize single outliers. Thoughtful choice of the minimum bucket parameter allows ICOT to avoid creating clusters of single or small sets of outliers, which often lack meaning and generalizability in grouping tasks. Traditional methods, such as K-means, deal with outliers by increasing the K parameter and forcing the algorithm to provide with a higher number of clusters. N_C can significantly affect the clustering solution and should be cross-validated or experimented on in order to get accurate and intuitive results from ICOT. The maximum depth can be used to impose an upper bound on the number of clusters if desired, although this parameter does not address potential outlier issues.

The ICOT algorithm is implemented in Julia [35] and is available to academic researchers under a free academic license.[1]

3.3.2 Mixed-variable handling

Both the Silhouette Metric and Dunn Index assess the quality of a given cluster assignment using the pairwise distance matrix of the observations. Distance is quantified differently for numerical and categorical variables and thus must be adjusted appropriately in the presence of mixed variable types. In the case of continuous features, the data are first normalized to be in the $[0, 1]$ range. The pairwise numerical distance matrix d^N is computed using the Euclidean distance between each pair of normalized variables. In the case of categorical features, distance is defined based on whether the observations take on different values. For example, if one observation takes on category A and another observation takes on category B on a given feature, the distance on this feature will be 1. The distance is zero if the observations take on the same value. For each pair of observations, these indicators are summed over all categories to define the categorical feature distance matrix d^C.

When the feature space includes both numerical and categorical variables, special consideration must be given to avoid over-weighting the categorical variables. In particular, categorical variables are often one-hot encoded (i.e. converted to binary 0/1 columns) to allow them to be treated as numerical in machine learning methods. This adjustment is insufficient in our case as it will result in placing too high of an importance on the categorical distance.

[1] Please email icot@mit.edu to request an academic license for the ICOT package.

We handle this issue by taking a linear combination of the two separate distance matrices for numerical and categorical variables. We first compute separate distance matrices for the numerical and categorical features. We let S^N denote the set of indices for the numerical features, and S^C denote the categorical indices. The computations for d^N and d^C are explicitly defined in Equations 3.23 and 3.24.

$$d_{ij}^N = \sqrt{\sum_{k \in S^N} (x_k^i - x_k^j)^2} \tag{3.23}$$

$$d_{ij}^C = \sum_{k \in S^C} \mathbb{1}\{x_k^i \neq x_k^j\} \tag{3.24}$$

We then compute the final distance matrix by taking a linear combination of these two matrices, given in Equation 3.25.

$$d_{ij} = \alpha d_{ij}^N + (1 - \alpha) d_{ij}^C \tag{3.25}$$

By default, the two distances are weighted according to their proportion of all covariates, so $\alpha = \frac{|S^N|}{|S^N| + |S^C|}$. The user can also specify an alternative α parameter. At $\alpha = 1$, the distance matrix only accounts for numerical covariates, whereas $\alpha = 0$ only considers disagreements in categorical variables.

3.3.3 Scaling methods

Our coordinate-descent procedure is more computationally intensive than the original OCT algorithm due to unique characteristics of clustering. In particular, we must compute a global clustering quality score at each split threshold evaluation, unlike classification tasks in which the loss change for a potential split can be assessed locally at the node. This global score assessment involves higher computational effort per split evaluation and thus motivates the development of more efficient search procedures. We introduce two scaling methods to take advantage of the geometric intuition behind cluster creation as well as existing clustering methods. We furthermore propose a subsampling approach to allow the algorithm to scale to much larger problems.

Restricted geometric search space

ICOT leverages the geometric structure of the feature space by restricting the set of candidate splits to those with sufficient separation. An exhaustive search of candidate splits on a given numerical feature requires $n_k - 1$ threshold evaluations, where n_k is the number of observations in a given node. This is due to the fact that there are exactly $n_k - 1$ different possible partitions of the data on the given feature at node k (less if multiple observations have the same value on this feature).

To improve the efficiency of our algorithm, we only consider a subset of these thresholds. For any feature, we refer to a threshold's gap as the separation between the observations directly below and above it. Since the quality of a clustering assignment is directly tied to the distance separating distinct clusters, the cluster quality will be superior when considering thresholds with large gaps. We take advantage of this intuition by skipping over thresholds with small gaps.

We control the extent of search space restriction through the parameter T. When considering a numerical feature split at node k, all threshold gaps for observations in the node are sorted ($n_k - 1$ values). Only thresholds above the T^{th} percentile of gap size are considered. For example, if $T = .9$ and $n_k = 100$, only the thresholds with the 10 largest gaps are considered, reducing the number of computations per node by 90%.

Figure 3-3 provides an illustration of how the Restricted Geometric Search would be applied in a simple example. When $T = 0.7$, ICOT will investigate only the top 30% of the gaps between observations. Thus only the larger, bold, gaps would be potential splits for a branch node that considers the covariate corresponding to the horizontal axis.

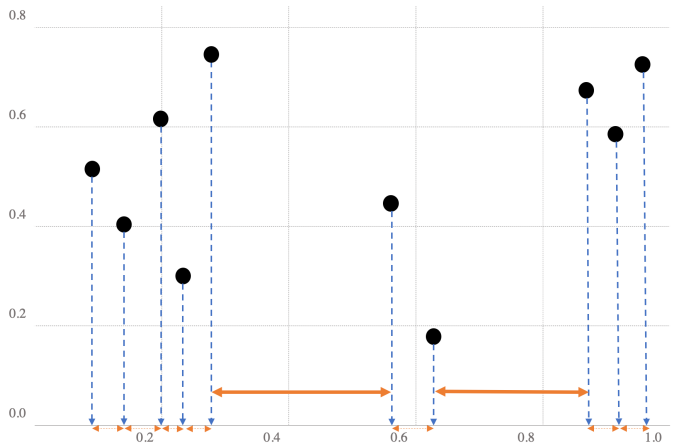


Figure 3-3: An example of the Restricted Geometric Search Function.

K-means warm start

We also employ warm starts to more efficiently identify high-quality clustering trees. We leverage the K-means algorithm to partition the data into clusters and use OCT to generate a tree that reasonably separates these clusters. This becomes the starting point of ICOT's coordinate descent algorithm. The algorithm first runs K-means on the original data across various K parameters and selects the assignment that optimizes our chosen cluster quality criterion. The resulting assignments are used as class labels for the construction of a supervised classification tree using OCT. ICOT's coordinate-descent procedure then begins from the resultant OCT tree rather than a greedy tree. Each leaf from the OCT tree becomes a separate cluster when initializing the ICOT algorithm, even though the predicted class labels may match between multiple leaves. Overall, the K-means warm start expedites tree initialization and improves the efficiency of the search procedure.

Bootstrapping

We introduce bootstrapping on the number of input observations, N. Our goal is to make the algorithm amenable to solve problems of larger sample size. This procedure involves subsampling a reduced population of size N_r and solving smaller problems N_{rep} times. This allows the algorithm to scale linearly with respect to the number of repetitions. It can be easily parallelized as it contains multiple independent sub-problems. Each iteration samples N_r observations without replacement and runs ICOT, returning a tree model which is then evaluated on a validation population. Upon completion of all N_{rep} iterations, the algorithm selects the best performing tree model on the validation criterion. Beyond improving the speed of the algorithm, bootstrapping provides a lot of flexibility to the user. The choice of N_r and N_{rep} may vary depending on the time constraints and the required quality of the final solution. We explore the latter in greater detail in Sections 3.6.2-3.6.2.

Complexity analysis

We provide a brief analysis of the worst-case complexity for each iteration of the coordinate-descent implementation of the algorithm. The argument is an extension of the complexity analysis for Optimal Classification Trees [60]. First, we consider the complexity of calculating our cluster

quality criteria.

An initial step for the computation of any score is the construction of a distance matrix that contains all the distances between each point $i, j \in [N]$, the training population. The matrix creation involves $\frac{n(n-1)}{2}$ calculations, which has complexity $\mathcal{O}(n^2)$.

Silhouette Metric (SM): For each observation i, we must compute the average distance between i and the members of each cluster. If we have T nodes, and each cluster contains at most n points, this has complexity $\mathcal{O}(nT)$. We need to find the distance to the next-closest cluster for which i is not a member. As we iterate through each of the clusters, we track the closest distance found so far and update if it improves. We note that the number of clusters is $\mathcal{O}(T)$ and is upper bounded by the total number of nodes. This computation is repeated for all n observations. Thus, the complexity of computing the Silhouette Metric is

$$cp_{SM} = \mathcal{O}(n(nT)) = \mathcal{O}(n^2T)$$

.

Dunn Index (DI): For each cluster, we must find the largest distance between any two points within the cluster and the smallest distance between a point in the cluster and outside of the cluster. This involves sorting at worst all pre-computed pairwise distances of which there are $\frac{n(n-1)}{2}$, giving complexity $\mathcal{O}(Tn^2\log(n))$. As we iterate through the sorted values, we track the highest intra-cluster and lowest inter-cluster distances and update if we find a value that improves either metric. In total, this yields complexity

$$cp_{DI} = \mathcal{O}(Tn^2\log(n)) = \mathcal{O}(Tn^2\log(n))$$

.

We now move on to the calculation of the algorithm's complexity in each iteration. Once an initial tree is constructed, each inner iteration of ICOT's local search consists of identifying the best potential split change at a given node. For each of the p features, there are at most $n-1$ potential split thresholds (if all observations are in this node). At each of these thresholds, we must (1) find

the assignment of all points to clusters (i.e. tree leaves), which has complexity $\mathcal{O}(nT)$, where T is the total number of nodes in the tree and (2) calculate the cluster quality criterion cp, either cp_{SM} or cp_{DI}. Thus, the inner iteration has complexity $\mathcal{O}(np(nT+cp))$. We must repeat this for each leaf, which adds a factor of T.

Ultimately, one iteration of ICOT when trained on the Silhouette Metric has worst-case complexity:

$$\mathcal{O}(npT(nT+n^2T)) = \mathcal{O}(n^2pT^2+n^3pT^2) = \mathcal{O}(n^2pT^2+n^3pT^2)$$

When optimizing the Dunn Index, ICOT's complexity is:

$$\mathcal{O}(npT(nT+n^2T\log(n))) = \mathcal{O}(n^2pT^2+n^3pT^2\log(n))$$

Both of these results demonstrate that each iteration of ICOT is highly sensitive to scaling with respect to n, with a higher cost when training on the Dunn Index (by a factor of $\log(n)$. Through the geometric search in Section 3.3.3, we are able to reduce the number of splits considered by a constant factor; with a threshold of 0.99, rather than considering np splits, we only consider $0.01*np$ splits. Additionally, the warm-starts explained in Section 3.3.3 provide higher quality starting solutions which reduces the number of iterations required to reach convergence and thus reduces runtime. This is demonstrated empirically in Section 3.6. Finally, the sub-sampling method introduced in Section 3.3.3 allows us to leverage ICOT for arbitrarily large problems; Section 3.6 also shows empirical evidence that the resultant trees still generalize well to the larger datasets despite only being trained on a subset.

3.4 Experiments based on synthetic datasets

In this section, we present results of ICOT across various synthetic datasets. We use these experiments to assess the quality of the algorithm's solution on both validation criteria. Section 3.4.2 compares ICOT to other popular clustering alternatives in terms of their ability to recover high-quality clustering assignments when training on both the Silhouette Metric and Dunn Index. We also examine the tradeoff between the two metric scores when training on one and evaluating on

the other.

3.4.1 Experimental setup

We evaluated ICOT on the Fundamental Clustering Problems Suite datasets (FCPS) [159], a standard set of synthetic datasets for unsupervised learning evaluation. These datasets have ground truth cluster labels, which allow for an objective comparison of cluster quality. Our experiments only consider nine of the 10 FCPS datasets, as the tenth contains no true clusters and thus does not offer insight into clustering algorithms.

The ICOT experiments use the “fully scaled" version of the algorithm, with a K-means warm start and a geometric threshold of 0.99. We left the minimum bucket size at its default value (1 observation) and restricted the maximum depth of the tree to depth 3. We left the α parameter at its default value. We ran 100 random restarts of the algorithm in each experiment.

We consider six alternative clustering algorithms which span a range of methodological approaches and interpretations. The following methods are compared:

1. Optimal Classification Trees Hybrid Method (`OCT`): A two-step K-means and OCT hybrid approach, in which K-means clusters serve as class labels for a supervised multi-class classification problem. Each observation is assigned a label based on the predicted class of its leaf. `OCT` is implemented using the InterpretableAI package in Julia [23, 34].

2. K-`means++`: We run K-means with a K-means++ initialization, which was introduced by Arthur and Vassilvitskii [9] and has been shown to improve upon a standard K-means implementation. K-`means++` has been incorporated in the `ClusterR` R package [123]. We run the method with 100 random restarts and a maximum of 100 clustering iterations.

3. Hierarchical Clustering (`Hclust`): Hierarchical clustering is the most popular agglomerative clustering method. It combines individual points into clusters using a linkage measure until all points end up in a single cluster, returning a single dendogram that exhaustively links all individual points [84]. While this is a tree-based method, it does not have binary splits and cannot be explicitly represented as a function of the features. `Hclust` is implemented in R using average linkage.

4. Gaussian Mixture Models (`GMM`): `GMM` assigns observations to clusters characterized by Gaussian distributions. The algorithm uses expectation-maximization (EM) to find the parameters for each of K Gaussian distributions, each representing a cluster [84]. This approach has a key advantage of accounting for cluster variance in assignment, which is a deficiency of traditional methods such as K-means. For each observation, this method returns a soft-assignment, which gives a probability of belonging to each cluster. To make this assignment amenable to our quantitative comparison which requires an explicit assignment, we assign observations to their most likely cluster. `GMM` is implemented in the `ClusterR` R package [123]. We run the method with 20 EM and K-means iterations and confirmed that the results stabilize by this point. We compute observation distances using Euclidean distance.

5. Density-based Spatial Clustering of Applications with Noise (`DBSCAN`): `DBSCAN` is a popular method that constructs clusters based on the highest density regions of a dataset [65]. `DBSCAN` does not return a complete assignment; outliers in low-density areas are left out of any clusters. While this exclusion approach makes the method robust to outliers, it complicates quantitative evaluation. To allow for a fair comparison on the internal validation metrics, we assign each outlier point to the most common cluster of its five nearest neighbors. If all neighbors are also unassigned, we assign the point to its own cluster. This method is implemented in the `DBSCAN` package in R [80], with additional post-processing to complete the outlier assignment.

6. Predictive Clustering Trees (`PCT`): Predictive clustering trees build recursive binary decision trees for clustering tasks [37]. The methodology is implemented in Java through the `Clus` package. We adopt the default “VarianceReduction" splitting heuristic.

We are unable to present synthetic comparisons to other recent work in interpretable clustering, such as CUBT, as there are no available implementations of the algorithms. We present results of ICOT against the CUBT experiments presented by Fraiman et al. [72] in Section 3.5.3.

We run all of the comparison methods on normalized data. ICOT normalizes the distance matrix within the algorithm, and we input a normalized dataset into the other comparison method functions. For each of the comparison methods, we tune key parameters to optimize the Silhouette Metric (or Dunn Index). In K`-means++`, `Hclust`, and `GMM`, we tune the number of clusters

$K \in [2, 10]$. `DBSCAN` does not have an explicit K parameter, but the ε parameter informs the neighborhood size when constructing clusters; larger ε values generally translate to larger clusters (and lower K). We tune $\varepsilon \in [0.1, 0.11, 0.12 \ldots, 1.0]$. Finally, `PCT` matches our methodology most closely and does not require an explicit cluster number (K) or density threshold (ε); for this algorithm, we simply tune the maximum depth from 1 to 3. In all cases, we select the parameter value that yields the best internal validation score on the metric of interest.

In the following experiments, all results are averaged over five experiments per algorithm and parameter combination. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

3.4.2 Solution quality

In these experiments, we look to assess various clustering methods in terms of their recovery of high-quality solutions, as measured by both the Silhouette Metric and the Dunn Index. We additionally investigate the performance of the "true" cluster labels on both of these criteria.

Tables 3.1 and 3.2 show the results of these methods along with the true FCPS labels, evaluated with both the Silhouette Metric and Dunn Index.

Data	(N,P)	ICOT	OCT	K-means++	Hclust	GMM	DBSCAN	PCT	Truth
Atom	(800,2)	0.503	0.433	0.611*	0.593	0.565	0.540	0.516	0.311
Chainlink	(1000,2)	0.396	0.28	0.479	0.496*	0.409	0.357	0.312	0.158
EngyTime	(4096,2)	0.573*	0.4	0.439	0.379	0.433	0.450	0.377	0.398
Hepta	(212,3)	0.453	0.332	0.702*	0.702*	0.608	0.702*	0.368	0.702*
Lsun	(400,2)	0.549	0.534	0.569*	0.554	0.537	0.439	0.564	0.439
Target	(770,2)	0.629*	0.409	0.593	0.619	0.578	0.533	0.516	0.295
Tetra	(400,3)	0.504*	0.266	0.504*	0.504*	0.504*	0.504*	0.307	0.504*
TwoDiamonds	(800,2)	0.486*	0.486*	0.486*	0.485	0.412	0.266	0.486*	0.486*
WingNut	(1070,2)	0.422	0.393	0.426*	0.418	0.407	0.384	0.422	0.384
Count Best/Tie		4	1	6	3	1	2	1	3
Average Score		0.502	0.393	0.534	0.528	0.495	0.464	0.430	0.409
Std. Dev Score		0.074	0.089	0.091	0.101	0.081	0.126	0.095	0.153

Table 3.1: Comparison of methods across the FCPS datasets, when trained and evaluated on the Silhouette Metric.
The asterisks indicate the best score across all algorithms for each criterion.

ICOT dominates the two-step supervised learning method in all cases for both metrics, offering an average Silhouette Metric improvement of 27.8% and Dunn Index improvement of 352.7%

Data	(N,P)	ICOT	OCT	K-means++	Hclust	GMM	DBSCAN	PCT	Truth
Atom	(800,2)	0.137	0.035	0.052	0.097	0.048	0.371*	0.064	0.371*
Chainlink	(1000,2)	0.028	0.013	0.038	0.037	0.016	0.265*	0.018	0.265*
EngyTime	(4096,2)	0.064*	0.002	0.005	0.014	0.004	0.029	0.002	0.000
Hepta	(212,3)	0.357	0.162	1.080*	1.080*	0.482	1.080*	0.293	1.080*
Lsun	(400,2)	0.077	0.027	0.056	0.071	0.117*	0.117*	0.026	0.117*
Target	(770,2)	0.550*	0.011	0.029	0.550*	0.113	0.117	0.013	0.253
Tetra	(400,3)	0.200*	0.044	0.200*	0.200*	0.200*	0.200*	0.046	0.200*
TwoDiamonds	(800,2)	0.044	0.022	0.031	0.049*	0.021	0.030	0.022	0.022
WingNut	(1070,2)	0.063*	0.020	0.026	0.036	0.016	0.063*	0.063*	0.063*
Count Best/Tie		4	0	2	4	2	6	1	6
Average Score		0.169	0.037	0.169	0.237	0.113	0.253	0.061	0.264
Std. Dev Score		0.176	0.048	0.347	0.358	0.153	0.330	0.090	0.330

Table 3.2: Comparison of methods across the FCPS datasets, when trained and evaluated on the Dunn Index.

The asterisks indicate the best score across all algorithms for each criterion.

over OCT. This demonstrates the advantage of building clusters directly through a tree-based approach rather than using a hybrid supervised learning method that applies a tree to cluster labels *a posteriori*.

ICOT matches or outperforms the best alternative clustering method in 4/9 cases with both the Silhouette Metric and with the Dunn Index. ICOT ties or beats K-means++ in 7/9 cases on the Dunn Index and 4/9 on the Silhouette Metric, attesting to its competitiveness against the most widely-used clustering technique. We also note that when measured against our most interpretable alternative, PCT, ICOT ties or wins in all cases on the Dunn Index and 7/9 on the Silhouette Metric.

When considering performance by the ranked wins/ties of each method, K-means++ is the best method for the Silhouette Metric and DBSCAN is the best method for the Dunn Index. No method dominates ICOT in the win/tie ranking; namely, there is no method that performs better on both the Silhouette Metric and Dunn Index. When looking at the average score across all nine datasets, Hclust is the only method to dominate ICOT on both training metrics. However, we note that Hclust also has a significantly higher standard deviation on both metrics, indicating a lack of consistency in solution recovery quality.

Our method is weakest when the underlying clusters are non-separable with parallel splits, since ICOT places hard constraints on an observation's cluster membership based on splits in feature values. In these cases, such as with the Hepta dataset, ICOT is unable to recover the true

structure. The flexibility offered by alternative methods is advantageous in these cases. Overall, our results demonstrate that despite the highly constrained setting that we impose on the solution structure, we are still able to perform competitively with far less constrained (and less interpretable) methods.

Cluster quality evaluation is highly dependent on the chosen metric; the ground truth assignment is only the “best" method in 3/9 cases with the Silhouette Metric and 6/9 cases with the Dunn Index. ICOT identifies strictly “better" clusters than the ground truth in 6/9 cases for the Silhouette Metric and 3/9 cases for the Dunn Index, as measured by their scores on the respective metrics. This phenomenon raises the broader question of how to assess cluster quality, as recovering known labels in synthetic data does not necessarily translate to meaningful cluster assignments.

Sensitivity to training criterion choice

Table 3.3 shows the ICOT scores on the FCPS datasets as measured by each validation criterion, broken down by training loss function. The values refer to the average score across all nine datasets. As expected, both metrics have their best performance when they are used as the training criterion to optimize for ICOT. The choice to train on the Silhouette Metric results in a 12.4% loss in Dunn Index score as compared to when training on the Dunn Index. Similarly, training originally on the Dunn Index results in a loss of 15.8% in the Silhouette Metric. This quantifies the sensitivity to the choice of training criterion. Both metrics incur a cost in terms of performance loss on other internal validation criteria, with a slightly lower loss on the Dunn Index.

Training Criterion	**Silhouette Metric**	**Dunn Index**
Silhouette Metric	0.475	0.149
Dunn Index	0.416	0.177

Table 3.3: Comparison of internal validation scores by choice of training criterion in the ICOT algorithm.

3.5 Experiments based on real-world datasets

In this section, we present results for two real-world examples. We address two important questions often encountered in practice and demonstrate the value of clustering in their analysis; interpretability and performance on internal validation criteria. We illustrate models produced by ICOT, `OCT`, K`-means++`, `Hclust`, `GMM`, `DBSCAN`, `PCT`, and the CUBT algorithm. We also consider the impact of tuning key user-defined parameters on the ICOT model. Section 3.5.2 outlines a patient similarity case study utilizing data from the well-known Framingham Heart Study (FHS). In these models we consider results across several minimum bucket sizes which offer different levels of granularity in the final output. We also experiment with various α parameters, allowing us to control the weight of numerical vs. categorical features in the distance matrix. Section 3.5.3 focuses on grouping economic profiles of European countries during the Cold War using only tree-based unsupervised learning techniques.

3.5.1 Experimental setup

We adopted a similar experimental setup to the one described in Section 3.4.1 for the synthetic experiments. In particular, the ICOT experiments use the “fully scaled" version of the algorithm, with a K-means warm start and a geometric threshold of 0.99. We ran 100 random restarts of the algorithm in each experiment. The α and minimum bucket parameters are varied as part of the experiments. We ran all of the experiments on normalized data, which is particularly relevant in this setting where features vary greatly in magnitude.

We consider the same six alternative clustering algorithms: `OCT`, K`-means++`, `Hclust`, `GMM`, `DBSCAN`, and `PCT`. The latter four methods cannot integrate both categorical and numerical features, so we updated the feature space to one-hot encode the categorical variables as binary features. We used the same fixed algorithm parameters for all methods as outlined in Section 3.4.1. We tuned the K parameter over the range of 2 to 10 clusters for all methods other than `DBSCAN`. We tuned $\varepsilon \in [1,5]$ for `DBSCAN`. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

3.5.2 Patient similarity for the Framingham Heart Study

Patient similarity is the concept of identifying groups of individuals with comparable health profiles from their electronic medical records, often with the goal of assessing treatment receptivity and outcomes. The goal is to cluster patients in compact groups without any particular outcome of interest and to study the health progression for those individuals over time. Clustering methods have been particularly popular in this application as they do not require an independent covariate in model creation.

We provide an illustration of our method using data from the Offspring Cohort from the FHS, a large-scale longitudinal clinical study. It started in 1948 with the goal of observing a large population of health adults over time to better understand cardiovascular disease risk factors. Over 80 variables were collected for 5,209 people over the course of more than 40 years. The FHS is arguably one of the most influential longitudinal studies in the field of cardiovascular and cerebrovascular research. This data has now been used in more than 2,400 studies and is considered one of the top 10 cardiology advances of the twentieth century alongside the electrocardiogram and open-heart surgery [54].

Our dataset consists of 1,200 observations from distinct participants of the Offspring Cohort and 11 covariates (age, gender, presence of diabetes, levels of HDL, BMI status, Blood Pressure (BP) status, blood glucose levels, hematocrit levels, history of myocardial infarction, history of stroke, and current smoking habits) [54, 69]. We explore how the ICOT model is impacted as we vary the α parameter and the minimum bucket parameter, N_C (Sections 3.5.2,3.5.2). Subsequently, we compare the results of ICOT with other clustering methods in terms of interpretability and quantitative performance on the validation criteria (Sections 3.5.2-3.5.2).

The effect of the α parameter

In this set of experiments, we focus on the impact of the α parameter on the creation of the ICOT model. The FHS dataset contains mixed numerical and categorical attributes and thus the determination of this parameter clearly affects the feature selection process during tree construction as well as the final number of clusters. We fix the minimum bucket parameter, $N_C = 50$, requiring at least 50 patients in each cluster to ensure that groups are not skewed by outliers in the data.

Figure 3-4 shows the model output when $\alpha = 0.3$. The number of observations in each group is indicated by the numbers in the leaves. When the distance matrix places 70% weight on categorical features, the algorithm partitions the feature space based only on those. As a result, only BP status and gender appear as splits in the tree. ICOT identifies eight groups of patients: (1) 100 women with Elevated BP; (2) 175 men with Elevated BP; (3) 96 women with Hypertensive Status I; (4) 163 women with Hypertensive Status II; (5) 163 men with Hypertensive Status I; (6) 172 men with Hypertensive Status II; (7) 135 women with normal BP; (8) 196 men with normal BP.

When $\alpha = 0.6$ the output model contains variables from both types of data, balancing better the numerical and categorical feature space. Due to the distance metric re-weighting, the new model is now able to incorporate both numerical and categorical features, yielding intuitive groups of participants by cardiovascular risk. Figure 3-5 illustrates the final tree with five split nodes and six clusters. Given these parameters, ICOT distinguishes between female and male participants in the presence or absence of diabetes. Moreover, it highlights the importance of smoking solely for the diabetic subgroup.

Finally, when $\alpha = 0.9$, ICOT only distinguishes the FHS population based on numeric features such as smoking and diabetes. These results highlight the importance of the algorithm tuning process when leveraging data with mixed features. In the absence of a ground truth, the decision maker is called to select the most appropriate model depending on the application or a potential downstream predictive task. The ability to directly parametrize the distance matrix provides the user with higher flexibility and clarity during the model development process. We discuss the implications of categorical features in the quantitative performance evaluation in Section 3.5.2.

The effect of the minimum bucket parameter

In these experiments, we set $\alpha = 0.6$ to balance the distance between numerical and categorical features and we vary the minimum number of observations required to form a distinct cluster. Figures 3-5, 3-7, and 3-8 show the models produced by the algorithm for different values of the minimum bucket, N_C, when training on the Silhouette Metric. Note that varying this constraint directly affects the end model, changing the structure of the final tree. Even though our empirical results may suggest that there is a monotonic relation between the size of the minimum bucket and the number of clusters identified, this assumption is not necessarily a general rule.

Comparing between Figures 3-5 and 3-7, we see that the output is stable given the minimum bucket restrictions. Both models share the same features in the splits. In the latter model, splits that already had at least 100 members in both leaves (the leftmost two clusters) remained intact and new ones were created in order to closely match the tree with $N_C = 50$. When we increase the minimum sample size to 200 participants, the resulting model only separates the population by gender.

Notice that across all the experiments presented, three variables appear to bear the highest importance in the clustering task: smoking habits, diabetic status, and gender. The results appeared to be stable in the feature selection process, confirming the intuition behind the effect of both the minimum bucket and α. ICOT's interpretable structure allowed us to specify the key differentiating characteristics between the participants and contextualize them in the medical setting.

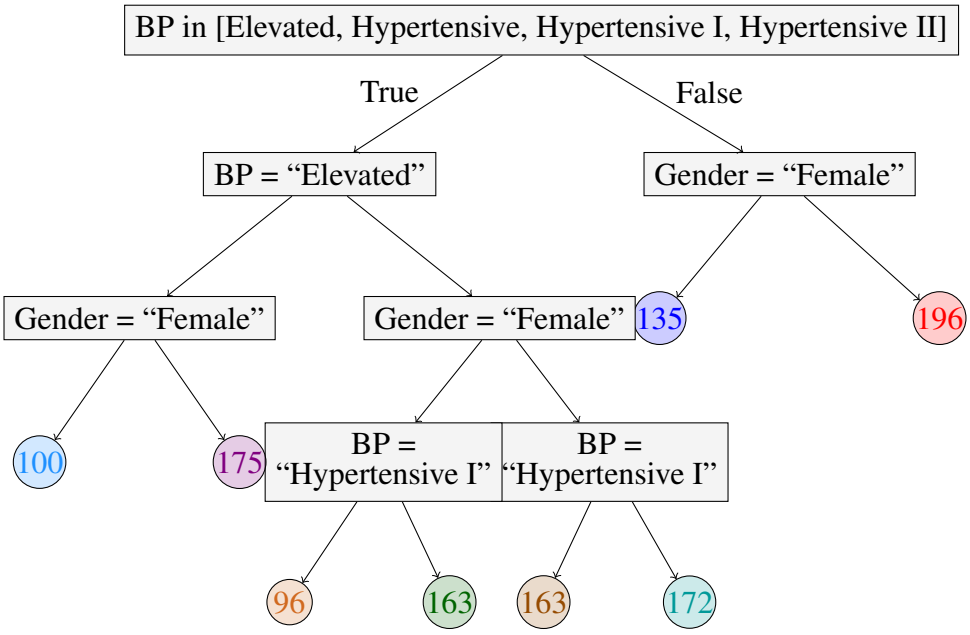


Figure 3-4: ICOT tree for minimum bucket = 50 and $\alpha = 0.3$.

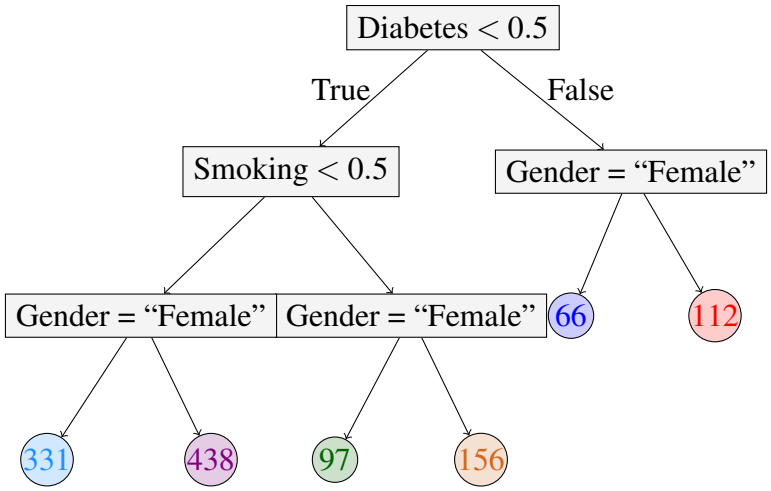


Figure 3-5: ICOT tree for minimum bucket = 50 and $\alpha = 0.6$.

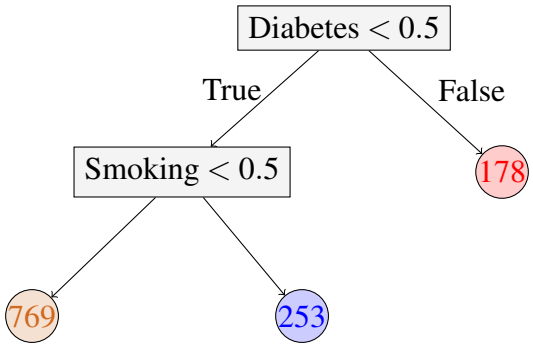


Figure 3-6: ICOT tree for minimum bucket = 50 and $\alpha = 0.9$.

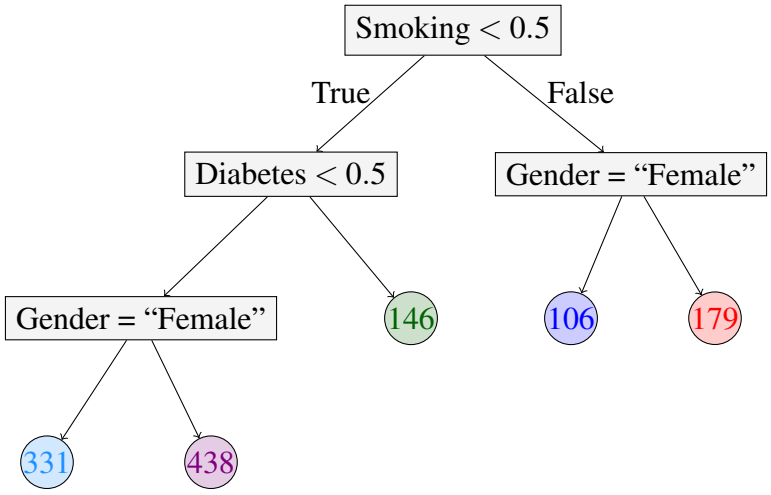


Figure 3-7: ICOT tree for minimum bucket = 100 and $\alpha = 0.6$.

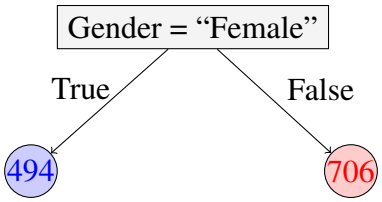


Figure 3-8: ICOT tree for minimum bucket $= 200$ and $\alpha = 0.6$.

Results on interpretability

In this section, we compare the interpretability of partitions from different clustering algorithms. For tree based approaches, such as the two step OCT method and `PCT`, we present the final model. For the rest of the algorithms, we outline the centroids of each cluster. Since these methods also do not allow us to directly control the minimum number of observations per cluster, we present the results of each algorithm for the number of clusters that maximizes the Silhouette Metric. We present detailed results for K`-means++`.

Figures 3-4-3-8 demonstrate different ICOT models when we vary the algorithm's hyperparameters. Note that the trees provide meaningful categorizations that clinicians frequently use and think about in stratifying patient risk. Elevated BP measurements, gender, smoking are all commonly used categories that determine future health trajectories, such as the risk of cardiovascular events or potential interventions for managing chronic diseases (i.e., blood pressure). The role of these variables has been widely recognized in medical literature [94, 169, 125, 66].

Variable Names	Cluster 1		Cluster 2		Cluster 3	
	Mean	**Std. Deviation**	**Mean**	**Std. Deviation**	**Mean**	**Std. Deviation**
Gender: female	0.367	0.485	0.376	0.485	0.487	0.5
Gender: male	0.633	0.485	0.624	0.485	0.513	0.5
Diabetes	0.922	0.269	0.054	0.227	0.142	0.35
Smoking	0.2	0.402	0.249	0.433	0.226	0.419
Age	64	7.114	61.102	9.976	65.335	9.156
HDL	39.497	12.679	46.681	14.592	46.547	14.663
Blood Glucose Levels	198.901	39.916	98.792	10.908	103.898	15.428
Myocardial Infarction	0.333	0.519	0.337	0.632	0.239	0.518
Hematocrit Levels	44.929	3.163	43.942	3.866	43.409	3.634
Blood Pressure Status: Elevated	0.211	0.41	0.358	0.48	0	0
Blood Pressure Status: Hypertensive Crisis	0.044	0.207	0	0	0.066	0.249
Blood Pressure Status: Hypertensive Status 1	0.256	0.439	0.239	0.427	0.165	0.372
Blood Pressure Status: Hypertensive Status 2	0.356	0.481	0	0	0.769	0.422
Blood Pressure Status: Normal	0.133	0.342	0.404	0.491	0	0
BMI Category: Normal	0.1	0.302	0.263	0.44	0.246	0.431
BMI Category: Obese	0.489	0.503	0.296	0.457	0.305	0.461
BMI Category: Overweight	0.411	0.495	0.44	0.497	0.447	0.498
BMI Category: Underweight	0	0	0.001	0.037	0.003	0.05
Number of Observations	90		716		395	

Table 3.4: The centroid mean, standard deviation values, and number of observations for all identified clusters from the K-means++ algorithm on the one-hot encoded dataset.

Table 3.4 shows the covariate values of the cluster centroids created by the K-means++ algorithm. Notice that there is no clear distinction of features that characterize each cluster. For the categorical ones, the centroid value depends on the relative frequency of the classes in the particular covariate and not only on its predominance in the cluster. For example, the fact that the Smoking value for Centroid 1 is equal to 0.2 does not provide deep insights in the smoking habits of the participants in that group. There is a similar proportion of smokers in this cluster compared to Clusters 2 and 3. It is difficult to provide intuitive labels for the groups with clinical implications by only studying Table 3.4. Furthermore, analyzing the centroid means and standard deviations to gain intuition into the distinctive attributes and spread of each cluster becomes increasingly harder as the number of features increases. Relative ranking of the centroid values could be used in the FHS case, where $p = 18$ (after one-hot encoding) and the number of clusters is small. In a high dimensional dataset,

delving into such a table would be practically impossible.

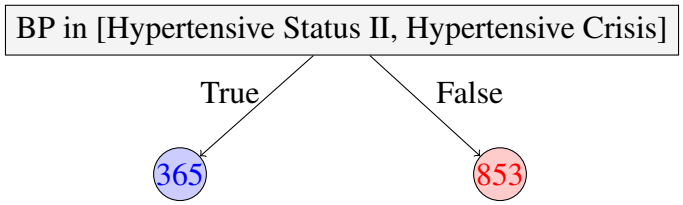


Figure 3-9: Two-step OCT tree, optimized with respect to the Silhouette Metric.

Figure 3-9 shows the result of the hybrid OCT tree. The model contains just one split, resulting in two clusters providing limited insights regarding the data. In this setting, changing the minimum bucket did not affect the final solution. Figure 3-10 shows the final PCT tree. This method proposes a deeper tree involving four features: Gender, Diabetes status, BMI, and Systolic Blood Pressure. It suggests that diabetes status is a differentiator only in obese patients (BMI above 30). It also suggests that the relevant Systolic Blood Pressure threshold is higher for “less healthy" patients, namely those who are diabetic or have higher BMI.

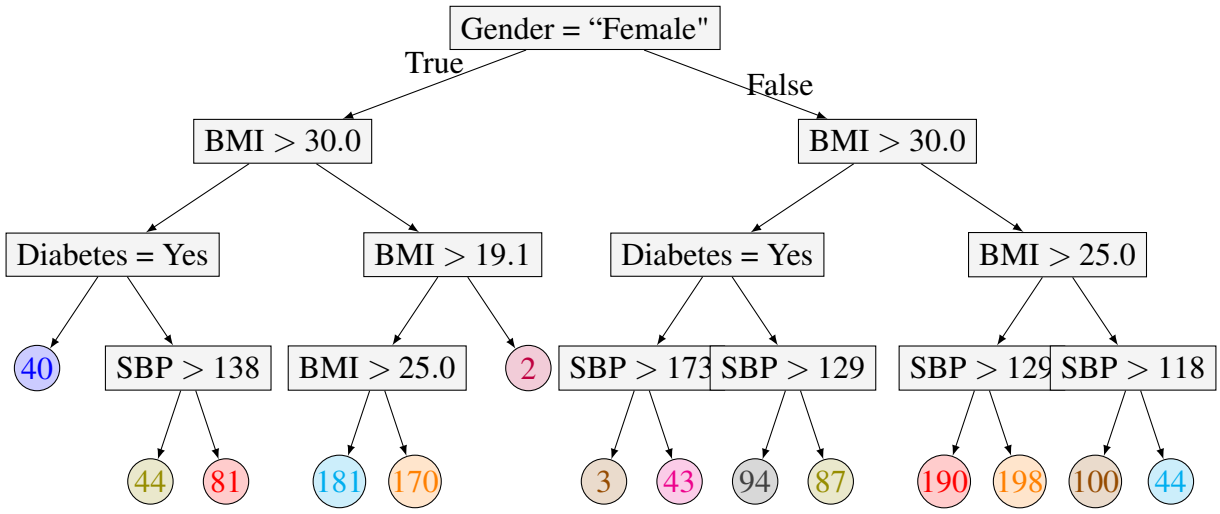


Figure 3-10: PCT Tree for FHS patients.

Results on quantitative performance

Although interpretability is our primary objective in cluster development, we also want to ensure that our resultant groupings are reasonable from the perspective of the internal validation criteria which provide a quantitative evaluation. Table 3.5 shows the metric scores obtained for both the

Silhouette Metric and the Dunn Index. For each method, we use the Silhouette Metric to cross-validate and find the optimal number of clusters. We then report the score on both metrics for the entire population.

ICOT dominates all competing algorithms in the Dunn Index (0.509) and has the second to best performance in the Silhouette Metric (0.296) after `DBSCAN` (0.511). In particular, we note that it has an advantage over `PCT` in both metrics, consistent with our findings in the synthetic experiments. Overall these results suggest that ICOT's advantage in interpretability does not come at the expense of identifying well-separated and compact clusters. The gains over OCT also attest to the value of ICOT's ability to train directly on the cluster quality criterion over simply applying a two-step method where K-means clusters are used as class labels for a supervised problem.

Metric	**ICOT**	**OCT**	**K-means++**	**Hclust**	**GMM**	**DBSCAN**	**PCT**
Silhouette Metric	0.296	0.131	0.264	0.270	0.224	0.511	0.249
Dunn Index	0.561	0.256	0.150	0.469	0.503	0.448	0.503

Table 3.5: The validation criteria results for ICOT, `K-means++`, `Hclust`, `GMM`, `DBSCAN`, `PCT` and the two-step hybrid OCT method when trained on each metric.

3.5.3 Economic profiles of European countries

In this section we consider European countries by their employment statistics during the Cold War to develop groupings of similar economic profiles. We present this example to offer a comparison to the CUBT algorithm [72] as this is the primary real-world experiment offered in their work.

Our dataset [99] provides the breakdown of where citizens were employed in 1979 across major industry sectors: agriculture (Agr), mining (Min), manufacturing (Man), power supplies services (PS), construction (Con), service industries (SI), finance (Fin), social and personal services (SPS), and transportation and communication (TC). Thus our feature space includes nine covariates ($p = 9$) observed for 26 distinct European countries ($n = 26$).

Results on interpretability: ICOT

We trained a clustering tree using the Silhouette Metric, the default α parameter, and a minimum bucket size of 3 to prevent individual outlier countries from dominating the tree in a single split. The final tree is shown in Figure 3-11, and the resulting groupings are shown in Table 3.6.

ICOT's chosen partition is highly intuitive given the economic and political climate of the Cold War. With the exception of Yugoslavia, all Eastern Bloc countries are placed in Cluster 1 due to their particularly low percentage of workers in the financial sector. This split reflects the broader political setting for those countries that were under a Communist regime. Greece, Turkey and Yugoslavia are grouped together due to their notably high agricultural sector employment. They are also located in the same geographical region and thus their economy similarity is justified. The rest of the countries form Cluster 2, which is composed of all the Western European countries.

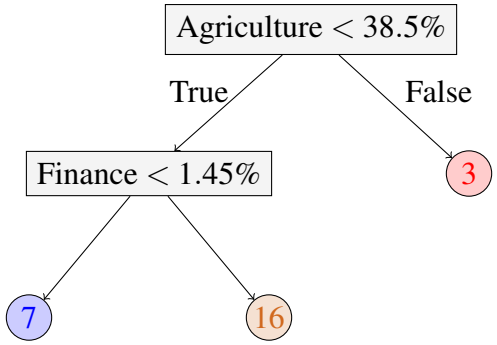


Figure 3-11: Visualization of the ICOT tree for the European Jobs dataset.

Cluster 1	**Cluster 2**		**Cluster 3**
Bulgaria	Austria	Belgium	Greece
Czechoslovakia	Denmark	Finland	Turkey
E. Germany	France	Ireland	Yugoslavia
Hungary	Italy	Luxembourg	
Poland	Netherlands	Norway	
Romania	Portugal	Spain	
USSR	Sweden	Switzerland	
	United Kingdom	W. Germany	

Table 3.6: European country clusters from the ICOT algorithm.

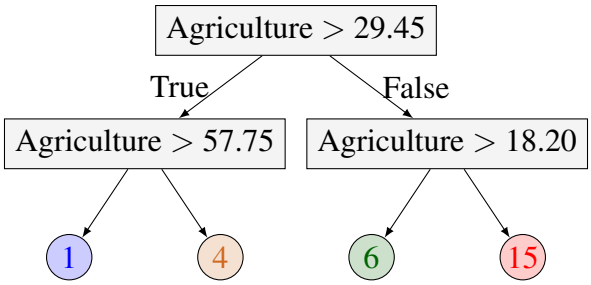


Figure 3-12: CUBT tree with four clusters.

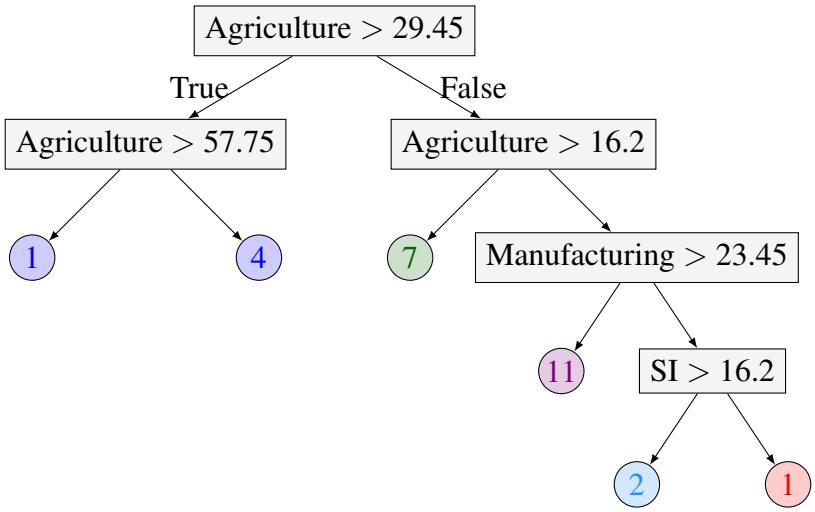


Figure 3-13: CUBT tree with five clusters.

Results on interpretability: CUBT

Fraiman et al. [72] provide two alternative clustering partitions using their proposed CUBT algorithm, one with four clusters and the other with five clusters. The resultant tree for $K = 4$ is shown in Figure 3.7 with the groupings listed in Table 3.7. The corresponding results for $K = 5$ and Table 3.8, respectively. Due to inconsistencies between the trees and country groups listed in the paper [72], we report results based on the tree models presented. It is possible to select a minimum bucket size in the CUBT algorithm, but the authors chose to omit it in these experiments, resulting in isolated clusters with single outlier countries. While this provides insight on its own, we chose to enforce a sufficiently large leaf size to make our results more generalizable and insightful for the full set of European countries.

The tree with four clusters splits only on agriculture sector employment through a series of recursive splits, providing less insight into the differentiating characteristics of the countries. The

tree with five clusters splits on high agriculture employment first to separate out the first two clusters, but then further differentiates the low agriculture countries on both manufacturing and service industry employment. The bulk of the countries fall into the third cluster, which is characterized by a manufacturing-heavy workforce. Note that CUBT allows for cluster re-joining in the algorithm, which results in multiple leaves being assigned to the same cluster (indicated by a single color). Overall, while the CUBT algorithm provides high interpretability as with ICOT, a qualitative analysis of the resulting clusters suggests that there is a slight loss in meaningful cluster separation.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	
Turkey	Greece	Bulgaria	Austria	Belgium
	Poland	Hungary	Czechoslovakia	Denmark
	Romania	Ireland	E. Germany	Finland
	Yugoslavia	Portugal	France	Italy
		Spain	Luxembourg	Netherlands
		USSR	Norway	Sweden
			Switzerland	United Kingdom
			W. Germany	

Table 3.7: European country clusters from the CUBT algorithm, with $K = 4$.

Cluster 1	**Cluster 2**	**Cluster 3**		**Cluster 4**	**Cluster 5**
Greece	Bulgaria	Austria	Belgium	Netherlands	Denmark
Poland	Czechoslovakia	E. Germany	Finland	Norway	
Romania	Hungary	France	Italy		
Turkey	Ireland	Luxembourg	Sweden		
Yugoslavia	Portugal	Switzerland	United Kingdom		
	Spain	W. Germany			
	USSR				

Table 3.8: European country clusters from the CUBT algorithm, with $K = 5$.

Results on the validation criteria

The quantitative performance of these models on our two key internal validation criteria are shown in Table 3.9. ICOT obtains significantly better clusters as quantified by both the Dunn Index and Silhouette Metric. We note that ICOT has an advantage in the Silhouette Metric due to the fact that it was trained to optimize this criterion, whereas the CUBT results were trained via a different

method. However, the Dunn Index provides a neutral evaluation criterion and shows a preference towards ICOT's results as well.

Metric	**ICOT**	**CUBT** ($K = 4$)	**CUBT** ($K = 5$)
Silhouette Metric	0.344	0.140	0.044
Dunn Index	0.346	0.262	0.259

Table 3.9: Comparison of ICOT (trained on the Silhouette Metric) and the CUBT algorithm on the internal validation criterion.

3.6 Scaling experiments

In this section, we present results regarding the effect of scaling techniques on ICOT with respect to both the quality of the final solutions as well as the degree to which the algorithm is able to scale. In Section 3.6.1, we discuss the impact of algorithm heuristics, such as the K-means warm start and the geometric threshold, using the FCPS suite. We use real-world data from Hubway for testing the scalability and quantitative performance of bootstrapping in Section 3.6.2.

3.6.1 Scaling via algorithm heuristics

In this section, we evaluate the impact of implementing the scaling methods described in Section 3.3.3. We first consider how the heuristics affect solution recovery in Section 3.6.1. Section 3.6.1 then examines the runtime reductions that we obtain as we vary the scaling parameters.

Experimental setup

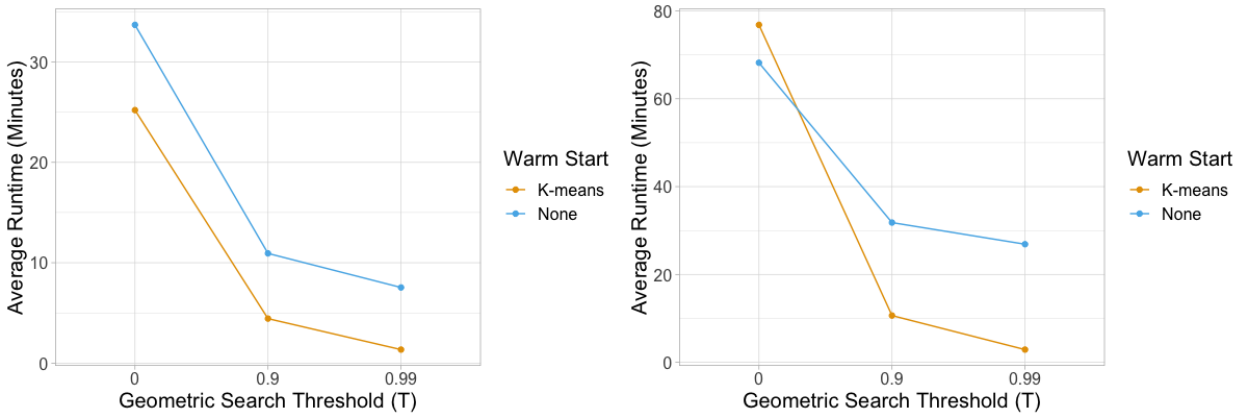
We evaluated the impact of our scaling methods on algorithm speed through a comparison of the average runtime across eight datasets in the FCPS suite with various parameters. The ninth dataset (EngyTime) was omitted as the experiment size was intractable on the unscaled method. We ran experiments over restricted geometric search thresholds of $T = 0$ (scan all thresholds), $T = 0.9$ and $T = 0.99$. We also repeated the experiments with and without the K-means warm start. The parameter pair ($T = 0$, no warm start) represents the original "baseline" method, and the

pair ($T = 0.99$, K-means warm start) represents the fully scaled method. We ran each dataset and parameter combination across five seeds and present the averaged results.

All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

Scaling runtimes

The runtimes for the Silhouette Metric and Dunn Index are shown in Figure 3-14. The geometric search alone reduces the runtime by 77.6% (60.6%) at the $T = 0.99$ threshold for the Silhouette Metric (Dunn Index). When combining the geometric search ($T = .99$) with the K-means warm start, our fully scaled method offers a 96.0% (95.7%) reduction in algorithm runtime for Silhouette (Dunn). We observe that the baseline method actually has a slight runtime advantage over the K-means warm start when there is no restriction on the search space ($T = 0$). The apparent shorter runtime with the baseline method at $T = 0$ can be explained by the possibility of getting caught in a locally optimal solution with a naive start, which can lead the algorithm to terminate faster.



(a) Silhouette Metric Runtimes

(b) Dunn Index Runtimes

Figure 3-14: Average runtimes across FCPS datasets with varied scaling parameters for the geometric search threshold (T) and choice to use a warm start.

Due to the speedups from these two scaling techniques, ICOT is able to scale to handle datasets with a number of observations (N) in the thousands and the number of covariates (p) in the hundreds. The scaled algorithm solves within several hours for problems of this magnitude.

High quality solution recovery

The scores of the baseline model and our fully scaled version are shown in Table 3.10. The scaled method yields an average loss of -0.28% over the baseline when trained on the Silhouette Metric, and gives an average improvement of 0.64% with the Dunn Index. Of the eight datasets considered using the Silhouette Metric (Dunn Index), three (five) have identical cluster recovery in both the original and fully scaled experiments; three (two) have a slight loss when using scaling heuristics, and two (one) actually improve with the scaling methods. These results suggest that the scaled ICOT algorithm still yields high quality results.

	Silhouette Metric			Dunn Index		
Dataset	**Baseline**	**Fully Scaled**	**% Change**	**Baseline**	**Fully Scaled**	**% Change**
Atom	0.521	0.503	-3.45%	0.137	0.137	0.00%
Chainlink	0.391	0.396	1.28%	0.032	0.028	-12.62%
Hepta	0.455	0.453	-0.44%	0.357	0.357	0.00%
Lsun	0.567	0.549	-3.17%	0.117	0.077	-34.10%
Target	0.629	0.629	0.00%	0.362	0.550	51.93%
Tetra	0.504	0.504	0.00%	0.200	0.200	0.00%
TwoDiamonds	0.486	0.486	0.00%	0.044	0.044	0.00%
WingNut	0.406	0.422	3.94%	0.063	0.063	0.00%
Average Score	0.495	0.493	-0.23%	0.164	0.182	0.65%

Table 3.10: Comparison of cluster quality scores with the original vs. fully scaled ICOT versions.

The differences in the score between the baseline and scaled versions are largely attributable to the warm start rather than the choice of geometric threshold. The score improves in the scaled version when the baseline algorithm was caught in a local optimum, but the K-means warm start enabled it to avoid this. This score improvement offered by the K-means warm starts further supports the use of this heuristic beyond runtime improvements.

3.6.2 Scaling via bootstrapping

In Section 3.6.2, we introduce the Hubway dataset, a real-world collection of user ride data from a Boston-based bike sharing program. Section 3.6.2 outlines the experimental setup, providing details on the parameters of the method. Sections 3.6.2 and 3.6.2 explore the effect of the bootstrapping methodology on the quality of the final solution and the algorithm runtime respectively.

The Hubway dataset

In this setting, our goal is to identify similar groups of registered users of the Hubway bike-sharing program [29]. This Boston-based company allows citizens to rent bicycles from any of their 140 stations and ride to any other station in the city. The platform has emerged as a popular form of transportation for daily commuters and leisure riders alike. Our dataset includes 194,301 observations from Hubway trips taken from June 2012 through September 2012. The dataset contains nine mixed numerical and categorical attributes, including the duration of the trip, the age and the gender of the rider, the time period of the ride and whether it took place during the week or the weekend.

This experiment illustrates an application of clustering for market segmentation. This is a strategy that divides a broad target market into smaller groups of similar customers. It can then be used to tailor marketing strategies to individual groups through means such as promotions or differentiated pricing. Unsupervised learning is often employed for this task since it naturally identifies similar groups within a given dataset.

Experimental setup

In these experiments, we aim to quantify the benefit of using bootstrapping as a wrapper function over the ICOT algorithm. We explore the effect of three key parameters that might affect both the quality and runtime of the solutions.

1. Sample Size (N): The number of observations included in the training set. Since the Hubway dataset contains 194,301 data points, we sub-sample randomly without replacement to create a sample of size N. We follow the same process to create a different testing set that is used for the evaluation of the validation criterion. We restrict N to numbers that can be efficiently solved by ICOT, $N \in [2500, 5000, 10000]$, to allow us to compare to the algorithm's solutions on the full input data.

2. Size of reduced data (N_r): The number of observations included in each iteration of the bootstrap algorithm. Each sub-sample is randomly created from the training set without replacement, but the iteration samples are constructed independently. Thus, different iterations can contain the same observation. We let $N_{\text{rep}} \in [250, 500]$.

3. Number of repetitions (N_{rep}): The number of iterations of the bootstrapping method. We test the quality and runtime of the final model by letting $N_{\text{rep}} \in [25, 50, 75, 100, 200, 500, 1000]$.

All results presented for ICOT use a version of the algorithm that includes the K-means warm start and a geometric threshold of 0.99. The minimum bucket size is set to one and the maximum depth of the tree to depth four. We assigned to the α parameter its default value. Similarly to the FCPS experiments, we ran 100 random restarts of the algorithm in each round. Results summarize the outcomes of five randomized repetitions of each experiment.

In the following experiments, all results are averaged over 50 experiments per algorithm and parameter combination. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 30GB of NUMA enabled memory were used per CPU.

Scaling performance

The purpose of introducing bootstrapping into the ICOT framework is to extend its application to problems of larger size that the fully scaled version was not able to efficiently manage. Bootstrapping provides a lot of flexibility to the user and thus can be easily adapted to the speed requirements of a specific case study. In this section, our aim is to demonstrate how choices regarding the parameters affect the overall running time and compare the outcomes with and without bootstrapping.

Figure 3-15 provides an overview of the results when the algorithm was trained on the Silhouette Metric. We report the $log(\text{time})$ to render the y-scale more comprehensible to the reader, especially for higher instances of N. The average runtime scales linearly with respect to N_{rep} and exponentially to N_r. As we include additional repetitions, the method sequentially runs more iterations of the same "reduced" experiment. However, as we increase the N_r, the runtime scales at the same rate as the original ICOT method. When $N_{\text{rep}} > 500$, bootstrapping starts improving on the original algorithm only for instances of $N > 2500$. Nevertheless, in cases of larger sample size ($N = 10,000$), bootstrapping can achieve the same solution quality ($N_r = 250, N_{\text{rep}} = 500$) in 27.65 minutes instead of 554.693. When $N = 5,000$, the discrepancy is not as high but still considerable, 13.095 and 96.529 minutes respectively.

These results indicate the value of adding bootstrapping into the ICOT framework, as it solves in reasonable time problems of much larger size that otherwise would have been out of the algorithm's scope.

High quality solution recovery

The bootstrapping approach constructs trees on a sub-group of the overall population and thus does not access the full input data. We sought to ensure that the speed-up in runtime would not come at a high toll with respect to solution quality. Thus, we performed a direct comparison of the two methods over the validation criteria for different ranges of the parameters described above. Figure 3-16 provides a results summary for the Silhouette Metric. The shaded region around ICOT indicates the standard deviation of the metric. Similarly, the error bars illustrate the same measure for each combination of the tuning parameters. As expected, larger sample sizes are positively correlated with the validation score. The graphs show that increasing the number of repetitions can significantly improve the quality of the solution. We notice that for $N_{\text{rep}} > 500$, bootstrapping can achieve equivalent performance to ICOT, with minor losses in some cases. The effect of the N_r parameter is less evident, though, as the results indicate minor discrepancies between $N_r = 250$ and $N_r = 500$. In conclusion, these experiments provide evidence that bootstrapping does not result in a high toll on the quality of suggested feature partitions.

3.7 Discussion

ICOT builds trees that provide explicit separations of the data on the original feature set, creating interpretable models with real-world applicability to a wide range of settings. From healthcare to revenue management to macroeconomics, our algorithm can significantly benefit practitioners that may find value in unsupervised learning techniques in their work.

Our empirical results on the FCPS dataset offer insight into ICOT's performance against existing methods, including traditional approaches such as K-means, density-based, and hierarchical algorithms. We also report results with respect to other interpretable methods, including the Predictive Clustering Trees framework and the hybrid two-step supervised approach. Overall, our proposed method is superior to the majority of the algorithms for both validation criteria. Specifically, in Section 3.4, we show that when assessing clusters with the Silhouette Metric, ICOT is the second best method after K-`means++` while on the Dunn Index ICOT is only outperformed by `DBSCAN`. Essentially, our experiments demonstrate that our newly proposed framework is able to achieve comparable performance to the state-of-the-art clustering algorithms while enabling the

explicit characterization of cluster membership. We thus accept a slight decrease in the validation criteria for the gain in interpretability, which is critical in many settings.

We also observe significant improvements in ICOT over other interpretable approaches. The relatively poor performance of the two-step OCT approach validates the utility of a method that simultaneously builds clusters and identifies a tree-based structure rather than simply employing existing tree-based methods on clustered data *a posteriori*. Additionally, ICOT offers a considerable advantage over PCT and CUBT, suggesting that our algorithmic approach improves upon on existing interpretable clustering work and offers a novel contribution to the space.

Most clustering methods, including ICOT, identified data partitions with higher cluster quality scores than the true FCPS data labels, highlighting the subjectivity of what constitutes good clusters. We leave the choice of cluster quality metric to the user, since both criterion have their respective merits and perform well in different data contexts. In general, the Dunn Index excels on well-separated datasets but is not robust to outliers. In contrast, the Silhouette Metric is often better at accounting for mixed densities and identifying meaningful separation in less structured data settings.

The additional scaling experiments on the FCPS dataset demonstrate substantial runtime reductions offered by both the restricted geometric search space and K-means warm start. Overall these empirical results suggest that the scaling methods are successful at significantly decreasing runtime while maintaining high-quality cluster identification. The geometric search heuristic is particularly useful for problems with a high number of observations as it lowers the computational load per node evaluation by a factor of T. We note that despite the efficiency gains offered by our scaling methods, our current implementation of ICOT does not scale beyond 1000s of observations and 100s of covariates. However, using the Hubway dataset we were able to demonstrate that the ICOT algorithm coupled with bootstrapping is able to scale to even hundreds of thousands of observations at a reasonable time without a considerable toll on the solution quality. This functionality broadens the method's applicability to even high-dimensional settings; for example, bootstrapping might be particularly useful when clustering a large company's customer transaction records (n in the millions). This is a case where we would recommend the subsampling approach. A similar technique could be applied for cases where the number of features is very high (p in the 10000s), such as when using genomic profiles for patients. Additionally, variables could be

preprocessed to restrict to the most significant subset, either using traditional statistical tests or the variable importance ranking provided in the K-means algorithm output.

Therefore, we believe that ICOT is the best performing alternative for interpretable clustering although computationally more intensive. PCTs are more efficient but in many cases lead to lower quality solutions. Our method has an edge over K-means++ and DBSCAN due to the transparency it offers, although these alternatives sometimes show a slight edge on the Silhouette Metric and the Dunn Index. ICOT is most appropriate in applications where the user values both interpretation of the cluster labels and high performance on clustering metrics, and the efficiency of the algorithm is not a bottleneck. These conditions are generally true in the exploratory analysis contexts where clustering is most often applied.

Our work's handling of numerical and categorical features offers a contribution beyond the realm of clustering. The issue of mixed-type attributes is considered among specialists as one of the most important challenges in machine learning [133, 175]. The overwhelming majority of state-of-the-art clustering algorithms are restricted to numerical objects, like vectors or metric objects, which does not correspond to datasets usually found in practice. This problem extends more broadly to algorithms that rely on distance computations, such as k-Nearest Neighbors. In contrast, our solution gives a comprehensive answer to this problem by introducing a novel distance metric for the algorithm.

We note that the algorithm's single-variable splits are unable to represent all possible cluster shapes and could potentially cut through clusters. This structure allows us to maintain the direct interpretation of a tree leaf representing a single cluster. In many applications, a simple interpretation of the tree partition is highly valued, which was a key motivation behind this method's development. In order to capture more complex structures, one could consider the possibility of "rejoining" leaves, namely allowing multiple leaves to be considered as a single cluster. Rejoining can occur between two adjacent leaves coming from a single parent node through the local search's consideration of split deletions. However, we do not consider the possibility of joining other leaves. While ICOT does not natively support this, it could easily be incorporated as a post-processing step. After obtaining the final ICOT tree, one can consider the effect of merging different node combinations on the chosen metric.

We finally observe that despite the tree structure of our algorithm output, our model does not

obey a hierarchical structure. Namely, truncating the tree to a lower depth does not necessarily represent the optimal clustering solution at this depth. Our coordinate-descent algorithm allows for nodes to be re-optimized with knowledge of deeper nodes. In contrast, a hierarchical interpretation only holds in cases where the tree grows greedily since the shallow truncated tree cannot be affected by deeper levels.

The application of ICOT to real-world datasets reveals the significant benefit on both interpretability and performance in the unsupervised learning field. The combination of the OCT mechanism, the employment of established internal validation criteria as well as the systematic handling of mixed numerical and categorical attributes allow ICOT to provide complete partitions of the feature space with actionable insights to practitioners. Moreover, the flexibility of the method to user specific constraints with respect to the minimum bucket size, the maximum depth of the tree and the α parameter render the algorithm particularly amenable to a wide range of applications from various fields.

3.8 Conclusion

In this paper, we have introduced a new methodology of cluster construction that addresses the issue of cluster interpretability. We propose a novel unsupervised learning tree-based algorithm that yields high-quality solutions via an optimization approach. Through computational experiments with benchmark and real-world datasets, we show that ICOT offers significant gains in interpretability over state-of-the-art clustering methods while achieving comparable or even better performance as measured by well-established internal validation criteria. This makes ICOT an ideal tool for exploratory data analysis as it reveals natural separations of the data with intuitive reasoning.

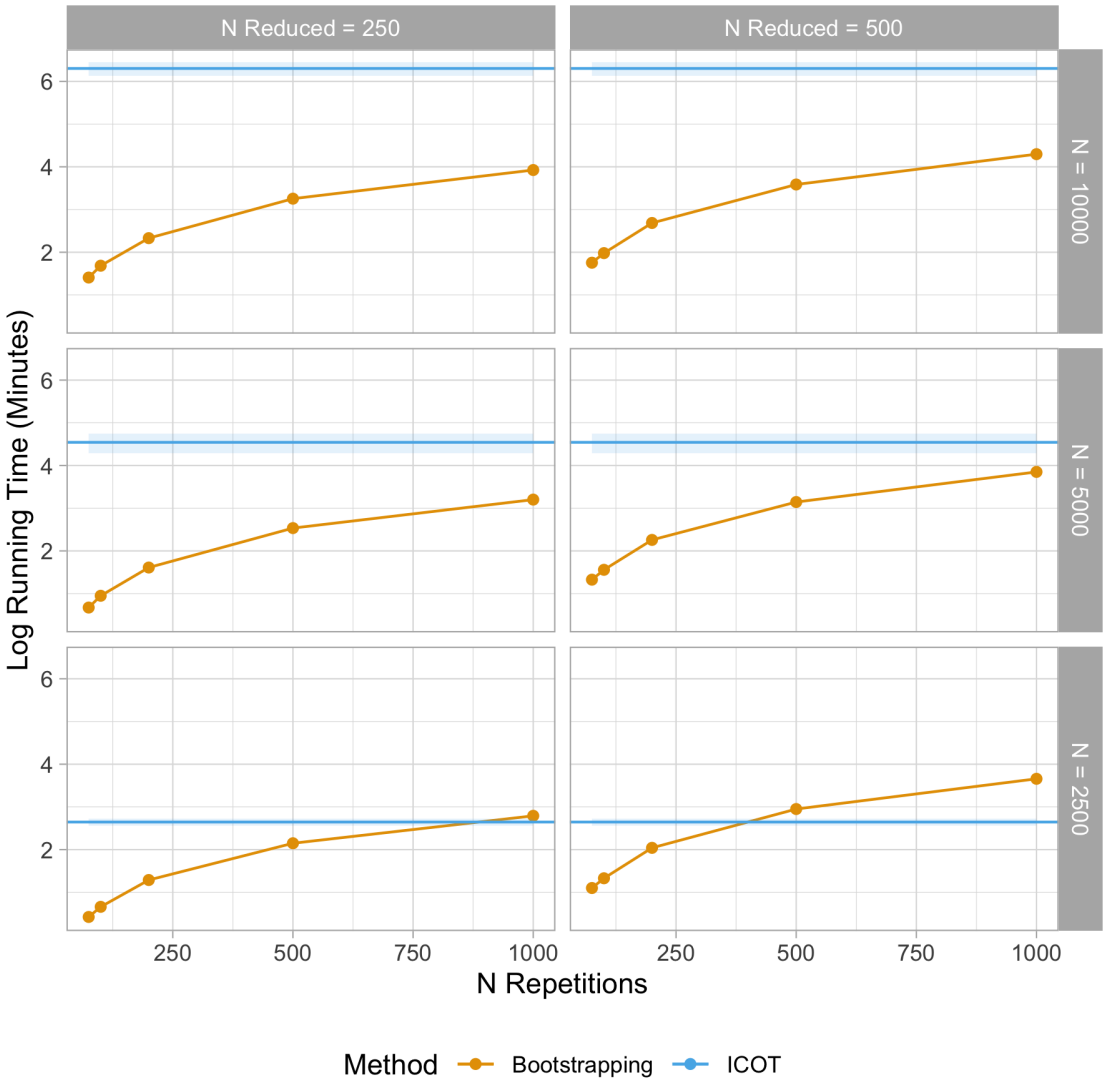


Figure 3-15: Results regarding the impact of bootstrapping on the runtime (Log of Minutes) as the number of repetitions (N_{rep}), sub-sample size (N_r), and sample size (N) change. Both methods were trained on the Silhouette Metric. The error bars express the standard deviation of the metric.

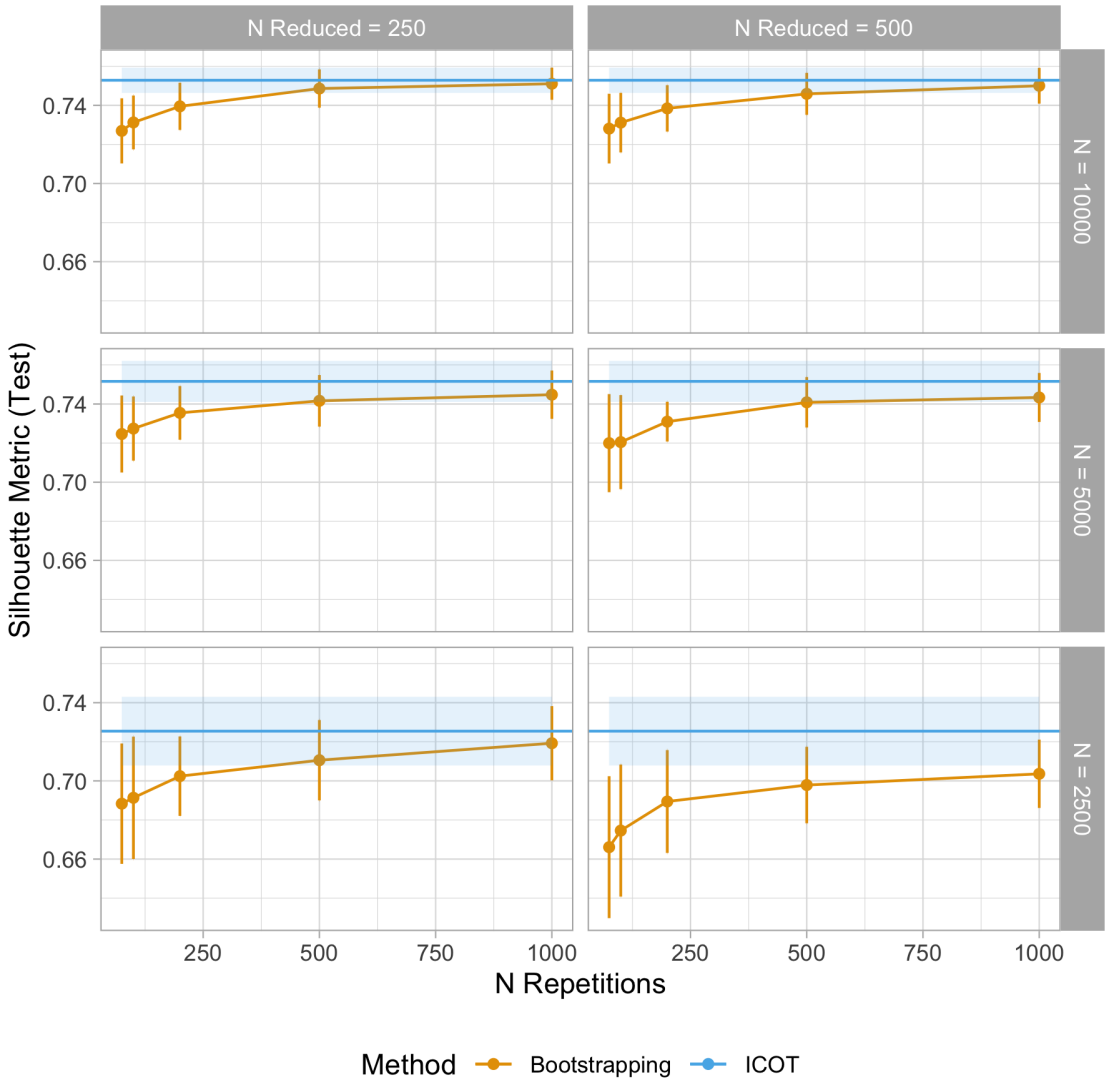


Figure 3-16: Results regarding the impact of bootstrapping on the Silhouette Metric as the number of repetitions (N_{rep}), sub-sample size (N_r), and sample size (N) change. The error bars express the standard deviation of the metric.

Chapter 4

Prediction of neutropenic events in chemotherapy patients: a machine learning approach

Abstract

Severe and febrile neutropenia present serious hazards to cancer patients undergoing chemotherapy. We seek to develop a machine learning-based neutropenia prediction model that can be used to assess risk at the initiation of a chemotherapy cycle. We leverage rich electronic medical records data from a large healthcare system and apply machine learning methods to predict severe and febrile neutropenic events. We outline the data curation process and challenges posed by electronic medical records data. We explore a range of algorithms with an emphasis on model interpretability and ease-of-use in a clinical setting. Our final proposed model demonstrates an out-of-sample AUC of 0.865 (95% CI 0.830-0.891) in the prediction of neutropenic events based on only 20 clinical features. The model validates known risk factors and offers insight into potential novel clinical indicators and treatment characteristics that elevate risk. It relies on factors that are directly extractable from electronic medical records, providing a tool can be easily integrated into existing workflows. A cost-based analysis provides insight into optimal risk thresholds and offers a framework for tailoring algorithms to individual hospital needs. A better understanding of neutropenic risk on an individual level enables a more informed approach to patient monitoring and treatment decisions.

4.1 Introduction

Severe and febrile neutropenia (SN/FN) pose severe risks to cancer patients receiving chemotherapy. ASCO guidelines recommend the use of granulocyte colony stimulating-factor (G-CSF) prophylaxis when the risk of SN/FN exceeds 20%[148]. A model of SN/FN risk for chemotherapy patients over the course of multiple cycles of chemotherapy and adjustable for economic considerations can provide more personalized insights into patient risk throughout their treatment. In this work, we propose a highly accurate and interpretable machine learning (ML) model for predicting neutropenic events using electronic medical record (EMR) data from a large healthcare system.

Neutropenia risk models have been introduced previously by several independent research groups. Most existing models rely on logistic regression [113, 89, 45, 96, 130] or ML methods that lack interpretability [50, 87]. Our proposed model differs from existing works in several ways: we assess the risk of SN/FN at the initiation of any chemotherapy cycle, not only the patient's first cycle; we consider a broad set of cancers and drugs, rather than focusing on targeted populations or treatment regimens; and we restrict our dataset to discrete EMR fields, allowing for direct integration into oncology workflows without manual data manipulation.

In this work, we present an end-to-end analytical pipeline, from data extraction to model implementation considerations. We develop a neutropenia risk score using Optimal Feature Selection (OFS), a novel approach that trains sparse additive models with strong performance and high usability. The final model provides clinical insight into neutropenic risk, both validating risk factors from existing models and identifying additional clinical indicators.

4.2 Materials and methods

4.2.1 Study population

This retrospective, observational study was carried out at Hartford HealthCare (HHC), an integrated healthcare system composed of 7 acute care hospitals in Connecticut with over 6,000 analytic cases per annum. The study consists of antineoplastic chemotherapy encounters between May 2016 and October 2019. Leukemia was excluded because often the goal of chemotherapy is to produce profound and prolonged neutropenia through bone marrow suppression; all other can-

cers were included. Each cycle start date is considered as a separate observation, which implies that a single patient can appear multiple times within the dataset. The full criteria used in querying the database are included in Appendix B.1.

4.2.2 Data curation

Data Extraction All data used in this study were curated from HHC's EMR (Epic Systems, WI). Only data directly extractable from the EMR were included. This approach makes it feasible to directly run the model off of the EMR and improves the transferability of our model to other institutions. With these same advantages in mind, unstructured data, such as free-text notes on patient condition, detailed information about regimen adjustments, and imaging results, were not selected for the dataset.

Clinical Features The outcome of interest was the occurrence of either SN or FN within 4 weeks (28 days) of a chemotherapy encounter [154]. We defined SN as Absolute Neutrophil Count (ANC) below 500 cells/uL and FN as ANC < 1000 cells/uL, accompanied by a fever (> 101 F) [154]. We curated discrete EMR data elements reflecting the patient's demographics, medical features, and cancer treatment information. By incorporating both static and temporal components, we captured how these factors change over time. Table 4.1 lists the data elements and sources. In total, each encounter was represented by 107 distinct clinical features.

Category	Source	Sample Features
Demographics	Patient data	Age, gender
Comorbidities	Problem list	Cerebrovascular disease, diabetes, hypertension
Other procedures	Procedure charges	Indicator of concurrent radiation
Treatment Information	Cancer patient history	Cancer site, treatment intent (e.g., curative, maintenance)
Drugs administered	Medication charges	Chemotherapy drugs (individual and combinations), indicator of G-CSF administration
Vitals Measurements	Flowsheets	BMI, Pulse, Systolic Blood Pressure
Lab Measurements	Lab order results	Complete Blood Count results

Table 4.1: Overview of clinical features and sources.

While the data we selected from the EMR are more highly structured than free-text notes, there is still significant variability in data capture that hinders the creation of a unified dataset. For

example, a single clinical entity such as temperature might be captured in different ways based on hospital department and equipment, or lab results may be a mix of numeric and text data (10, 15, <5, >100). We standardized our data elements through close collaboration with clinical experts, creating data mappings and common vocabularies in the process. These can serve as artifacts for future projects to facilitate meaningful information extraction.

Data Imputation Missing data present a challenge in developing a comprehensive feature space, particularly for vital and lab features. Many measurements are not recorded at all visits, resulting in missing data for some encounters. Any features that were missing in more than 40% of encounters were fully excluded. For the remaining features, imputation was performed using MedImpute [30], a novel method that estimates missing values based on the known values of similar observations. The imputation balances the known values from proximal encounters of the same patient with data from other patients; this is formalized through an optimization algorithm.

4.2.3 Machine learning methods

The prediction of whether an encounter will be followed by SN/FN is a binary classification problem. Table 4.2 outlines the methods considered, which offer various levels of interpretability and complexity. The models were trained using a common training set of 80% of the data. 5-fold cross-validation was employed on the training set to tune the relevant internal parameters for each method. The final models were then evaluated on the remaining out-of-sample data (20%). The training and testing data were split by patient, meaning that no patient can have encounters in both the training and testing set, to prevent bias in model evaluation. We additionally report results on a temporal split of the data in Appendix B.2.3. In both cases, the training and testing sets were imputed independently.

4.2.4 Model evaluation

When evaluating candidate models, we considered two primary criteria: quantitative performance and model interpretability.

Method	Reference	Description
Logistic regression (LR)	[85]	Fits an additive model. Regularization is employed to limit model complexity and control over-fitting. This serves as a benchmark traditional statistical method.
Optimal Feature Selection (OFS)	[32]	Fits an additive model using a sparse set of features. For example, OFS_{20}, with a maximum sparsity of 20, would fit a model with at most 20 features having non-zero coefficients. In this study, three maximum sparsity parameters, 20, 30, and 50, are considered.
Classification and Regression Trees (CART)	[40]	Partitions data using a single decision tree. The tree is comprised of binary feature splits, and each leaf yields a predicted risk probability.
Optimal Classification Trees (OCT)	[23, 34]	Constructs a single decision tree, as with CART. In contrast to traditional greedy tree-based models such as CART, OCT uses an optimization framework when fitting the tree which generally demonstrates superior performance.
Random Forests (RF)	[39]	Fits many decision trees each using a subset of features and data, forming an ensemble of models. Final predictions aggregate the "votes" of the individual trees.
Gradient Boosted Machines (XGB)	[73, 47]	Another ensemble approach which trains many decision trees but employs a weighting scheme to better account for errors in individual learners.

Table 4.2: Overview of ML methods used for binary classification.

Quantitative Performance A model must provide accurate predictions to be useful in a clinical setting. Binary classification models output a probability of a positive response; in this setting, a "positive response" is defined as the occurrence of a neutropenic event. We used out-of-sample Area Under the ROC Curve (AUC) as the primary performance metric. Given the low incidence of SN/FN, we also considered the average precision for each model. Average precision provides a threshold-independent measure of the precision-recall curve, like AUC for the ROC curve, that is particularly useful when the outcomes are highly imbalanced. We report these metrics for our final models along with bootstrapped 95% confidence intervals.

While the model returns a probability of SN/FN, in practice a probability threshold is often used to label high-risk patients. For a fixed threshold τ, all encounters with probabilities greater than τ would be categorized as having a high risk of an SN/FN event. This is useful in making the outcomes of the predictive tool actionable; for example, the threshold could be used to determine when to surface EMR alerts. For a fixed threshold, we can assess the number of false negatives

(FN_τ) and false positives (FP_τ) incurred by the model. Lower thresholds predict more neutropenic events: this increases sensitivity at the expense of specificity. The opposite is true as the threshold increases.

The desirable threshold is user-determined and driven by the relative costs of mistaken positive and negative events. To determine an optimal risk cutoff threshold, we must quantify the cost of a false positive (C_{FP}) and false negative (C_{FN}). From a financial perspective, the negative consequence of a false positive is unnecessary intervention and for a false negative, hospitalization. For this estimate, we assume that the intervention for high-risk patients would be G-CSF administration. For a given threshold τ, the cost incurred is then given as:

$$C_\tau = C_{FP} * FP_\tau + C_{FN} * FN_\tau$$

The optimal threshold minimizes this cost.

Interpretability The methods considered vary in their inherent interpretability. Linear models and single decision trees explicitly tie clinical inputs to the resultant predictions. Ensemble methods, such as Random Forests (RF) and gradient boosted machines (XGB), aggregate many individual models which limits their interpretability. Models that lack clinical interpretation make it difficult to justify predictions and assess their validity, hindering clinician trust [156].

An additional aspect of interpretability is model sparsity, namely the number of features used to generate a prediction. For example, a decision tree with six splits would use a maximum of 6 features to output a prediction. While the input feature space contains 107 unique features, it is desirable for the final model to rely on a subset of these features for clinical interpretability.

4.3 Results

4.3.1 Study population

The final population consists of 17,513 encounters across 2,806 patients. 449 (2.6%) of the encounters had a neutropenic event within 28 days of the encounter. 421 encounters had SN and 77 had FN; outcomes that met the criteria for both SN and FN count as a single neutropenic event. The

most common cancers observed in the data are breast, lung, colon, rectal/anal cancer and multiple myeloma, which comprise more than 60% of the encounters.

4.3.2 Model performance

Table 4.3 reports the test AUC and average precision for the various models. The methods have out-of-sample AUCs ranging from 0.789–0.869 and average precisions ranging from 0.080–0.148. Given the significant class imbalance in our data (2.6% positives), the baseline precision is 0.026. Therefore, the best performing models offer a roughly five-fold increase from the baseline precision. The additive models, Optimal Feature Selection (OFS) and logistic regression (LR), demonstrate strong performance. OFS with 20 features (OFS_{20}) is able to achieve the second-best AUC (0.865, 95% CI 0.830-0.891) and the highest average precision (0.148, 95% CI 0.117-0.188) with fewer features than other linear models. RF performs comparably but at the price of lower interpretability. Overall, the strength of the linear models suggests that nonlinear feature interactions are not highly significant in this prediction problem.

Of the models considered, OFS_{20} offers the most insight given its balance of both quantitative performance and model interpretability. This is therefore our proposed final model.

Model	**AUC**	**Avg. Precision**
OFS_{20} (20 Features)	0.865 (0.830-0.891)	0.148 (0.117-0.188)
OFS_{30} (30 Features)	0.866 (0.836-0.894)	0.136 (0.107-0.173)
OFS_{50} (50 Features)	0.854 (0.824-0.893)	0.131 (0.104-0.170)
LR	0.858 (0.818-0.893)	0.146 (0.117-0.195)
OCT	0.805 (0.773-0.841)	0.112 (0.089-0.154)
CART	0.789 (0.749-0.828)	0.104 (0.078-0.138)
RF	0.869 (0.842-0.889)	0.145 (0.117-0.193)
XGB	0.819 (0.799-0.848)	0.080 (0.064-0.110)
No Skill	0.5	0.026

Table 4.3: AUC and average precision (with 95% confidence intervals) reported on the test set.

4.3.3 Threshold-based analysis

An optimal decision threshold can be found after estimating the costs and probabilities associated with false positives and false negatives. To illustrate threshold selection, we consider an example

for non-small cell lung cancer (NSCLC) patients. Based on analysis by Li et al. [103], we estimate the cost of G-CSF administration as $2580 and the cost of a neutropenia-related hospitalization for an NSCLC patient at $21822.50, $5075 per day with an average length of 4.3 days. Figure 4-1 shows the total expected cost incurred across all thresholds $\tau \in [0,1]$ for the OFS_{20} model on the test set. The cost-minimizing threshold is $\tau = 0.16$. At this threshold, the model obtains out-of-sample specificity of 95.7% and sensitivity of 42.9%.

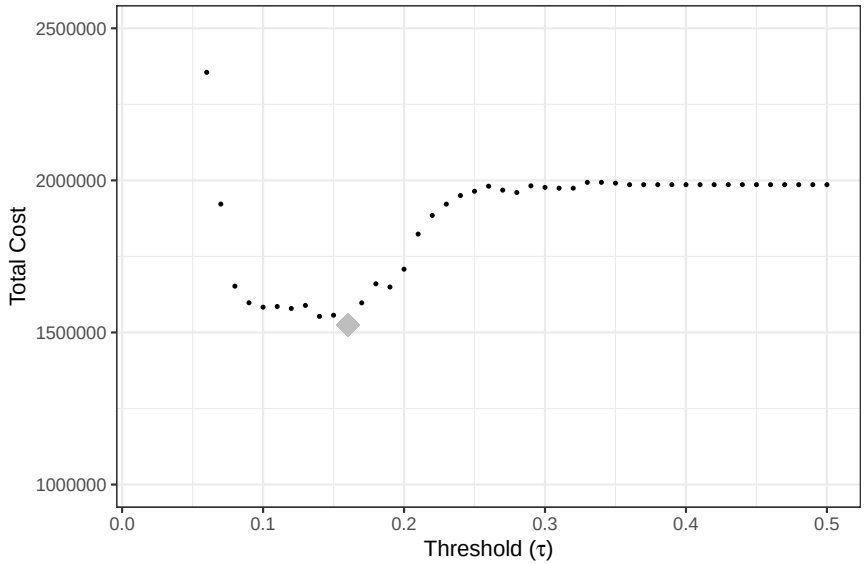


Figure 4-1: The cost incurred by false positives and false negatives as a function of the risk cutoff threshold (τ), assuming C_{FP}=$2580 and C_{FN}=$21822.50. The optimal threshold that minimizes the total cost is displayed in dark gray.

To obtain a more general characterization of the tradeoffs between false positives and false negatives, we compute the optimal threshold as a function of the ratio between C_{FN} and C_{FP}. The optimal threshold is determined purely by the cost ratio. The example above demonstrates the computation for a cost ratio of 8.5 ($C_{FN}/C_{FP} = \$21822.50/\$2580$). Figure 4-2 shows the optimal thresholds for varied cost ratios using the OFS_{20} model on the test set. As the ratio increases, the optimal threshold decreases; model sensitivity (identification of true positives) becomes more valuable, and so the model flags more patients as high risk. The optimal threshold begins to decrease above a ratio of 5. This implies that when the cost of hospitalization is more than five times as expensive as G-CSF intervention, it is economically advantageous to lower the decision threshold, which allows the model to recover true positives despite the risk of over-treatment of

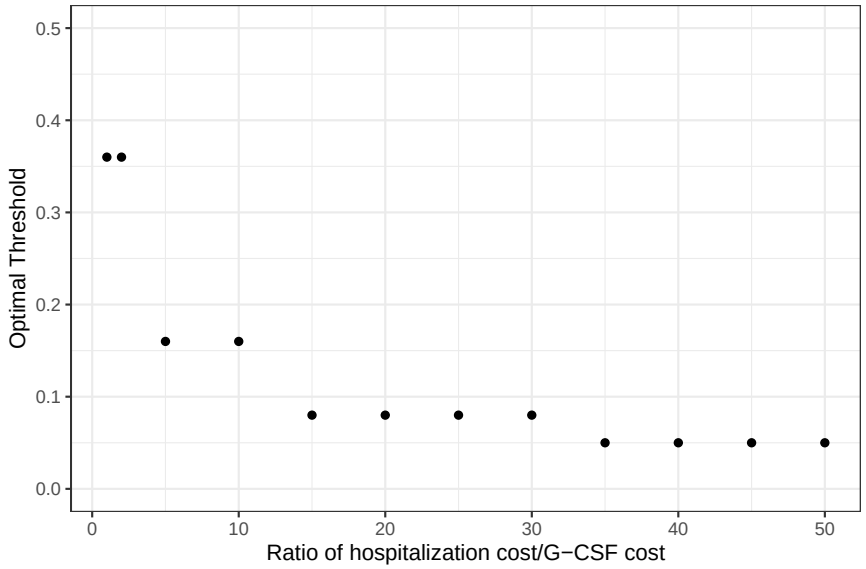


Figure 4-2: Optimal cutoff threshold as the ratio of hospitalization cost to G-CSF cost (C_{FN}/C_{FP}) varies.

false positives. The optimal threshold stabilizes at 16% for ratios between 5-10 and lowers to 8% for higher cost ratios between 15-30.

4.3.4 Model interpretation

The OFS_{20} coefficients are shown in Table 4.4. Positive coefficients indicate an increase in risk as the value increases (e.g., risk increases with the number of drugs given in recent weeks), while negative coefficients represent an inverse relationship (risk decreases as the cumulative infusion count increases). Individual drugs have varied risk impacts. Since a patient can receive multiple drugs in a single encounter, the net contribution of the drugs is determined by the sum of these coefficients.

4.4 Discussion

In this work, we have developed a practical tool for assessing SN/FN risk in patients upon initiation of a chemotherapy cycle. We leveraged discrete EMR data and used state-of-the-art algorithms to synthesize clinical features into a single risk score. The resultant model enables a personalized approach to patient care based on an individual's cancer characteristics, vitals, lab values, and

Category	Feature	Coefficient
	Intercept	-2.460
Treatment	Antineoplastic Drug Count (previous 3 weeks)	0.139
	Cumulative Number of Chemo Cycles	-0.021
	Filgrastim (G-CSF) Administered? (1 = Yes)	-0.142
Labs/Vitals	Hematocrit (%)	-0.03
	Platelet Count (Thou/uL)	-0.001
	Pulse	0.004
Comorbidities	Relative change in Weight (Lbs) from previous cycle	-3.065
	Diseases of the genitourinary system? (1 = Yes)	0.147
Drugs	Atezolizumab? (1 = Yes)	-0.696
	Carboplatin? (1 = Yes)	0.369
	Cisplatin? (1 = Yes)	0.374
	Cyclophosphamide? (1 = Yes)	0.787
	Dacarbazine? (1 = Yes)	0.469
	Docetaxel? (1 = Yes)	0.485
	Doxorubicin? (1 = Yes)	0.839
	Etoposide? (1 = Yes)	1.353
	Pertuzumab? (1 = Yes)	-0.321
	Trastuzumab? (1 = Yes)	-0.217
	Vinblastine? (1 = Yes)	0.469
	Vinorelbine? (1 = Yes)	0.791

Table 4.4: OFS coefficients with sparsity of 20 features.

course of treatment.

We compared various ML algorithms but ultimately selected the OFS_{20} model due to its high interpretability and competitive quantitative performance. Our final model has an out-of-sample AUC of 0.865. This model outperforms the proposed neutropenia risk model by Lyman et al. [113] (out-of-sample AUC of 0.81), while also using a smaller feature set that is directly extractable from the EMR. Cho et al. [50] report an AUC of 0.908 in their proposed ML model for FN prediction, although this model addresses a narrower clinical question of neutropenic risk for Korean breast cancer patients.

The OFS_{20} model coefficients provide insights into the risk contributions of individual features. Risk increases when more drugs are involved in the regimen, which could indicate more aggressive treatments. G-CSF administration leads to lower risk, consistent with its use as an intervention to

mitigate SN/FN risk, as other models have found [96, 113]. Genitourinary comorbidities, which includes renal diseases, increase risk; this aligns with findings of increased risk associated with kidney dysfunction [96, 113]. Higher blood counts (hematocrit and platelets) are associated with lower risk, as is a lower pulse. Finally, our model is the first to incorporate temporal elements. Risk decreases as patients are further along in treatment, i.e. as they have more previous cycles. We also see that the change in clinical features over time, particularly a decrease in weight, implies higher risk. We note that as with any retrospective study using observational data, we cannot establish causation of the observed risk factors or rule out the significance of unobserved factors. Nevertheless, the proposed final model provides highly accurate characterizations of patient risk based on the included features.

It is also informative to observe the features that were not selected in the model. While the feature space included cancer site and drug combination, rather than just individual drugs, neither of these features were selected in the final model. Certain clinical elements identified in other models, such as age [89, 96, 113], also do not appear in our model. This suggests that these factors are less significant in determining a patient's risk, or that their risk contribution can be explained through other observed clinical characteristics. We note that the commitment to the exclusive use of structured EMR data requires the omission of other potentially relevant data elements, such as relative dose intensity or qualitative assessments of patient health, in exchange for portability and reproducible results.

Our proposed approach to determining an optimal cutoff threshold for flagging high risk patients can be adapted to inform reasonable site-specific cutoffs for new populations and cost estimates. ASCO and NCCN guidelines recommend primary use of G-CSF at a threshold risk level of 20%. NCCN guidelines recommend consideration of G-CSF depending upon patient risk assessment for intermediate risk levels between 10-20% [18, 148]. Previously published models are specific to a chemotherapy regimen, cancer type or first cycle of treatment and thus lack generalizability. Our model finds that the cost analysis supports a threshold risk level of 8-16%. Our framework allows for tuning the performance specifications of the predictive model relative to the economic costs of treatment inherent to a healthcare delivery system and can be used to guide payer reimbursement policy.

Our threshold modeling provides a framework for determining an appropriate risk threshold

that can be extended to incorporate other factors. We did not attempt to model the positive economic benefits of true positives and negatives. An economic model of clinical decision support should ideally minimize unnecessary costs while also maximizing healthcare benefits. Additionally, while our analysis defines cost as financial costs incurred by either hospitalization or unnecessary intervention, there are also non-financial health economic costs that cannot be measured by this model. We remain cognizant of the burden of false positives which could lead to alarm fatigue [145], while also recognizing that false negatives associated hospitalization carry a quality of life cost which in economic terms are disutilities of care. The analysis can be modified to capture additional financial and quality-of-life costs.

A central goal of this paper was to create a frictionless point-of-care tool for assessing neutropenic risk while patients are undergoing treatment. This motivated the creation of a feature space using only discrete data elements. While individual health systems have distinct ways of recording patient data, all of the features included in the model should be available as structured data within the EMR. After establishing a mapping of a hospital's data elements to our feature space, the model can be integrated into a new EMR system to provide real-time insights in clinical encounters. Our selection of a model that relies on a relatively small subset of clinical features reduces the burden of creating such a data mapping; only 20 features need to be extracted from the EMR to calculate the risk score. These considerations lower the barrier to model validation and adoption at external sites. The ultimate test of any risk prediction model is its performance on external populations, and we hope to continue this work through prospective validation both within HHC and at external sites.

4.5 Conclusion

This work presents the development of a neutropenia risk prediction tool, from data curation to practical implementation considerations. This tool offers the potential to improve patient care, providing personalized insights for chemotherapy patients that enable more informed treatment planning and care decisions.

Chapter 5

COVID-19 mortality risk assessment: an international multi-center study

Abstract

Timely identification of COVID-19 patients at high risk of mortality can significantly improve patient management and resource allocation within hospitals. This study seeks to develop and validate a data-driven personalized mortality risk calculator for hospitalized COVID-19 patients. De-identified data was obtained for 3,927 COVID-19 positive patients from six independent centers, comprising 33 different hospitals. Demographic, clinical, and laboratory variables were collected at hospital admission. The COVID-19 Mortality Risk (CMR) tool was developed using the XGBoost algorithm to predict mortality. Its discrimination performance was subsequently evaluated on three validation cohorts. The derivation cohort of 3,062 patients has an observed mortality rate of 26.84%. Increased age, decreased oxygen saturation ($\leq$ 93%), elevated levels of C-reactive protein ($\geq$ 130 mg/L), blood urea nitrogen ($\geq$ 18 mg/dL), and blood creatinine ($\geq$ 1.2 mg/dL) were identified as primary risk factors, validating clinical findings. The model obtains out-of-sample AUCs of 0.90 (95% CI, 0.87-0.94) on the derivation cohort. In the validation cohorts, the model obtains AUCs of 0.92 (95% CI, 0.88-0.95) on Seville patients, 0.87 (95% CI, 0.84-0.91) on Hellenic COVID-19 Study Group patients, and 0.81 (95% CI, 0.76-0.85) on Hartford Hospital patients. The CMR tool is available as an online application at covidanalytics.io/mortality_calculator and is currently in clinical use. The CMR model leverages machine learning to generate accurate mortality predictions using commonly available clinical features. This is the first risk score trained and validated on a cohort of COVID-19 patients from Europe and the United States.

The ongoing coronavirus disease pandemic (COVID-19) caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has led to an alarming number of casualties across the world [56]. As the pandemic progresses globally, much remains unknown about the disease dynamics and risk factors. A better understanding of the clinical determinants of disease severity can

improve patient management throughout the healthcare system. This task is challenging due to the rapid spread of the disease and the lack of detailed patient data.

Leveraging machine learning (ML) methods enables the rapid discovery of insights across large populations of heterogeneous patients. An algorithmic approach provides an objective evaluation and can often capture nonlinear interactions that are not obvious from pure observation of the population. Researchers have recognized the potential of these data-driven approaches across various facets of the effort to combat COVID-19 [5].

In this work, we present the COVID-19 Mortality Risk (CMR) tool, a novel ML model for predicting mortality in hospitalized COVID-19 patients. It enables physicians to better triage patient care in a resource-constrained system through a personalized mortality risk score. The CMR model synthesizes various clinical data elements from multiple European and US centers, including demographics, lab test results, symptoms, and comorbidities. We use the XGBoost algorithm [47], a leading ML method, to predict mortality probabilities. This score is able to capture nonlinearities in risk factors, resulting in strong predictive performance with an out-of-sample area under the receiver operating characteristic curve (AUC) of 0.90 (95% CI, 0.87-0.94). It also validates commonly accepted risk factors, such as age and oxygen saturation, while discerning novel insights.

The CMR tool leverages an international cohort from three hospital systems in Italy, Spain, and the United States. The model is subsequently validated on hospitalized patients in a consortium of six hospitals from Greece, Spain, and the United States. Each region presents a diverse set of patient profiles and mortality rates for the model. By considering severely ill populations from different countries and healthcare systems, the final dataset captures a wide array of features.

In recent months, ML scores have been proposed to predict COVID-19 mortality [134, 105] as well as disease severity [173]. Existing literature largely focuses on Chinese hospitals due to the disease's emergence in Wuhan [105, 173]. However, it is instrumental to understand the clinical characteristics for more recent and diverse cases, considering that the virus strain may have mutated since surfacing in Wuhan [153]. Pourhomayoun and Shakibi [134] proposed a model based on a large international dataset, yet this model lacks comprehensive patient data and is thus limited in its ability to derive personalized insights. In this work, we study patients in Europe and the US, offering a new lens into the clinical characteristics of this disease.

5.1 Methods

5.1.1 Study population

The study comprises 33 different hospitals, spanning across three countries in southern Europe as well as the US. The collaborating institutions were split into derivation and validation cohorts, as summarized in Table 5.1. The derivation cohort includes the healthcare systems of ASST Cremona (Northern Italy), HM Hospitals (Spain), and Hartford HealthCare affiliate hospitals (United States). The broad geographic spread of data sources offers a comprehensive sample of some of the most severely impacted regions in the world. To further validate the results, we partnered with Hospital Universitario Virgen del Rocío (Spain), the Hellenic COVID-19 Study Group (Hellenic CSG), a consortium of Greek hospitals, and Hartford HealthCare's main hospital (CT, USA). The study population consists of adult patients who were admitted to the hospital with confirmed SARS-CoV-2 infection by polymerase chain reaction testing of nasopharyngeal samples. The time horizon of admissions is displayed in Table 5.1.

All independent organizations and the Massachusetts Institute of Technology institutional review boards approved this protocol as minimal-risk research using data collected for standard clinical practice and waived the requirement for informed consent. The survey was anonymous and confidentiality of information was assured.

5.1.2 Clinical features

Data is collected using the electronic medical record (EMR) databases and COVID-19 specific registries of the collaborating hospitals. We compile 22 features, including patient demographic information, comorbidities, vitals upon admission, and laboratory test results. The full set of features is outlined in Table 5.2. The outcome of interest, mortality during the hospital admission, is derived from discharge records. Only the first recorded laboratory test results are considered, typically within 24 hours of admission. Comorbidities are identified using the International Classification of Diseases, 9th and 10th revision, codes of hospital discharges and are aggregated into four categories using the Clinical Classifications Software [2]. Missing values are imputed using k-nearest neighbors imputation [158](Appendix C.2). We exclude risk factors that are not consis-

Organization	Region	Study Dates	Hospital Count	Description
				Derivation Cohort
ASST Cremona	Lombardy (Italy)	02/01 - 05/08	3	Azienda Socio-Sanitaria Territoriale di Cremona (ASST Cremona) includes the Ospedale di Cremona, Ospedale Oglio Po and other minor public hospitals in the Province of Cremona. Cremona is one of the most hit italian provinces in Lombardy in the Italian COVID-19 crisis with a total of 4,422 positive cases to date. Ospedale di Cremona has around 750 beds. During the COVID-19 crisis all elective activities and surgeries were suspended and most of the hospital was converted to treat COVID-19.
HM Hospitals	Madrid, Galicia, León, Cataluña (Spain)	02/01 - 04/20	17	HM Hospitals, a leading Hospital Group in Spain with 15 general hospitals and 21 clinical centers that cover the regions of Madrid, Galicia, and León. The group has served more than 2,300 COVID-19 patients over the last two months. Its total capacity includes more than 1,468 beds and 101 operating rooms.
Hartford HealthCare (Affiliates)	Connecticut (USA)	03/18 - 05/14	5	Hartford HealthCare is a major hospital network serving patients throughout Connecticut. In addition to its primary hospital in Hartford, it operates five acute care hospitals: Backus Hospital, Charlotte Hungerford, the Hospital of Central Connecticut, MidState Medical Center, and Windham Hospital. These sites have a total of 1,087 beds and nearly 2,500 physicians on staff.
				Validation Cohort
Hospital Universitario Virgen del Rocío	Seville, Andalusia (Spain)	03/11 - 05/05	1	The Hospital serves a basic population of 557,576 users, between the districts of Seville, Aljarafe and Seville South in the region of Andalusia. It has a provision of 1,279 beds installed, with a staff of 8,409 professionals. During the COVID-19 crisis, elective activities and surgeries were suspended, and most of the hospital was converted to care, COVID19 patients, it has attended approximately 320 COVID-19 positive cases discharges (exits included) until May 5th.
Hellenic COVID-19 Study Consortium	Attika, Thraki, Thessaly, Peloponnese (Greece)	03/01 - 05/15	6	This is a collection of the referral center of Greece for the management of COVID-19 patients. It includes the Sotiria Thoracic Diseases Hospital of Athens, Evangelismos Hospital, the University Hospitals of Alexandroupolis and Patra, the Attikon General Hospital and the General University Hospital of Larissa. All organizations are independent public (NHS and academic) institutions.
Hartford HealthCare (Main Hospital)	Connecticut (USA)	03/18 - 05/14	1	Hartford Hospital is an 867-bed acute care teaching hospital located in Hartford, Connecticut. Hartford Hospital was established in 1854 and is the central campus of the broader Hartford HealthCare organization. It employs 1,200 physicians and dentists and has a caseload of over 100,000 annual emergency room visits.

Table 5.1: Overview of participating institutions in the derivation and validation cohorts.

tently recorded in the derivation cohort, thereby omitting features whose values are more than 40% missing.

Feature	All (N=3,062)	Survivor (N=2,302)	Non-Survivor (N=760)	P-Value [1]
Age	68.0 (57.0-79.0)	64.0 (54.0-74.0)	80.0 (73.0-85.0)	<1.0E-04
Female*	1207.0 (39.42%)	958.0 (41.62%)	249.0 (32.76%)	<1.0E-04
Heart Rate (bpm)	90.0 (80.0-102.0)	91.0 (80.0-102.0)	88.0 (79.0-100.25)	7.7E-03
Oxygen Saturation (%)	94.0 (91.0-96.0)	94.4 (92.0-96.0)	90.55 (85.5-94.0)	<1.0E-04
Temperature (°F)	98.6 (97.7-99.86)	98.42 (97.59-99.86)	98.79 (97.7-100.08)	9.3E-04
Alanine Aminotransferase (U/L)	27.0 (17.0-43.0)	27.0 (17.22-44.53)	26.0 (16.12-41.0)	7.9E-02
Aspartate Aminotransferase (U/L)	35.9 (25.3-54.5)	34.0 (24.55-50.2)	44.0 (30.0-68.0)	<1.0E-04
Blood Glucose (mg/dL)	118.35 (105.0-142.0)	115.0 (103.4-134.0)	134.0 (113.0-170.55)	<1.0E-04
Blood Urea Nitrogen (mg/dL)	17.0 (12.62-25.56)	15.0 (11.61-20.96)	29.0 (20.0-46.0)	<1.0E-04
C-Reactive Protein (mg/L)	73.37 (28.88-146.43)	58.62 (22.74-117.83)	137.93 (69.81-214.13)	<1.0E-04
Creatinine (mg/dL)	0.95 (0.77-1.22)	0.9 (0.74-1.08)	1.25 (0.95-1.75)	<1.0E-04
Hemoglobin (g/dL)	13.9 (12.6-14.9)	13.9 (12.8-15.0)	13.4 (11.9-14.6)	<1.0E-04
Mean Corpuscular Volume (fL)	88.0 (85.0-91.2)	87.7 (84.9-90.7)	89.4 (85.93-92.9)	<1.0E-04
Platelets (103/μL)	202.0 (157.0-259.75)	205.0 (160.0-263.0)	187.0 (146.5-248.5)	2.0E-04
Potassium (mmol/L)	4.05 (3.7-4.4)	4.0 (3.7-4.4)	4.1 (3.7-4.6)	<1.0E-04
Prothrombin Time (INR)	1.11 (1.02-1.25)	1.11 (1.02-1.23)	1.13 (1.02-1.31)	<1.0E-04
Sodium (mmol/L)	137.1 (135.0-140.0)	137.0 (135.0-139.5)	138.0 (135.0-141.0)	<1.0E-04
White Blood Cell Count (103/μL)	6.73 (5.13-9.09)	6.51 (5.05-8.59)	7.92 (5.57-11.0)	<1.0E-04
Cardiac dysrhythmias*	201.0 (6.56%)	128.0 (5.56%)	73.0 (9.61%)	5.8E-04
Chronic kidney disease*	72.0 (2.35%)	40.0 (1.74%)	32.0 (4.21%)	1.5E-03
Heart disease*	125.0 (4.08%)	80.0 (3.48%)	45.0 (5.92%)	9.2E-03
Diabetes*	384.0 (12.54%)	263.0 (11.42%)	121.0 (15.92%)	2.5E-03

[1] P-value reports significance of a two-sided T-test between the survivor and non-survivor populations.

Table 5.2: Summary statistics of all patient characteristics for the total sample, the survivor, and non-survivor cohorts. Median (IQR) is reported for continuous variables, and count (proportion) is reported for binary variables.

5.1.3 Modeling approach

We train a binary classification model in which the outcome is patient mortality: 1, if the patient was deceased, or 0, if discharged. Specifically, we use the XGBoost algorithm [47] for the training process, described further in Appendix C.2. For comparison, we also present the predictive performance of other ML methods in Appendix C.3. The derivation population is randomly divided into training (85%) and testing (15%) sets, ensuring that mortality prevalence was consistent between the two. We tune seven model parameters by maximizing the K-fold cross-validation AUC using the Optuna optimization framework [4]; more details are provided in Appendix C.2. This technique provides a more accurate parameter search compared to grid search by efficiently pruning suboptimal parameter combinations and continuously refining the search space. We apply

SHapley Additive exPlanations (SHAP) to generate importance plots for transparency of the model predictions and risk drivers [112, 111]; more details are provided in Appendix C.3. All statistical analysis is conducted using version 3.7 of the Python programming language.

5.1.4 Performance evaluation

All predictive models are evaluated based on their ability to discriminate between outcomes for each population. We report results for the training and testing sets of the derivation cohort, as well as for each independent institution in the validation cohort, with the corresponding confidence intervals (CI). The AUC, accuracy, specificity, precision, and negative predictive value are computed for all patient subpopulations across different thresholds. Receiver operating characteristic (ROC) curves were created for each of the cohorts.

5.2 Results

5.2.1 Patient characteristics

The CMR model is created using a derivation population of 3,062 patients, of which 1,441 are from ASST Cremona, 1,390 from HM Hospitals, and 231 from Hartford Affiliates. The validation population consists of 865 patients: 219 patients from Seville, 323 from the Hellenic CSG, and 323 from Hartford Hospital. The clinical characteristics of the derivation population are outlined in Table 5.2. The average observed mortality rate in this population is 26.84%. In comparison to survivors, non-survivors tend to be older (median age 80 vs. 64) and more commonly men (67.2% vs. 58.4% of cohort). Moreover, the prevalence of comorbidities such as cardiac dysrhythmias, chronic kidney disease, and diabetes is higher in the non-survivor population (9.61%, 4.21% and 15.92% versus 5.56%, 1.74%, and 11.42%, respectively). The clinical characteristics for each participating study site are reported in Appendix C.1.

5.2.2 Performance metrics

The final mortality model exhibits an out-of-sample AUC of 0.90 (95% CI, 0.87-0.94) on the derivation testing set; see Table 5.3. The AUC for the Seville cohort is slightly higher at 0.92 (95%

CI, 0.88-0.95). For the other two validation centers, there is a decrease in AUC. In the Hellenic CSG cohort, the model performs 0.87 (95% CI, 0.84-0.91) and in the Hartford Hospital population 0.81 (95% CI, 0.76-0.85). The corresponding ROC curves are included in Appendix Figure C-1.

Cohort	N	AUC	Threshold	Accuracy	Specificity	Precision	Negative predictive value
Training Set	2755	94.7 (93.87,95.54)	38.44 (36.62,40.25)	89.62 (88.48,90.76)	92.76 (91.79,93.73)	78.51 (76.98,80.04)	93.39 (92.46,94.32)
Testing Set	307	90.19 (86.86,93.52)	28.3 (23.26,33.34)	85.02 (81.02,89.01)	86.58 (82.77,90.39)	66.3 (61.02,71.59)	93.02 (90.17,95.87)
Hellenic CSG	323	87.45 (83.83,91.06)	20.23 (15.85,24.61)	74.92 (70.2,79.65)	74.23 (69.46,79.0)	25.74 (20.97,30.51)	97.3 (95.53,99.07)
Seville	219	91.62 (87.95,95.29)	33.21 (26.98,39.45)	86.76 (82.27,91.25)	87.43 (83.04,91.82)	48.94 (42.32,55.56)	97.09 (94.87,99.32)
Hartford	323	80.66 (76.36,84.97)	29.74 (24.75,34.72)	61.3 (55.99,66.61)	58.12 (52.74,63.5)	24.18 (19.51,28.85)	94.71 (92.26,97.15)

Table 5.3: AUC performance (%) and threshold-based metrics for training, testing, and validation population.

A different threshold is selected for each cohort to enforce a minimum sensitivity of 80%. Given the implications of these predictions, we report conservative risk estimates in order to ensure that all critically ill patients are accounted for. This comes at the expense of specificity, i.e., it increases the number of patients whom we may incorrectly flag as high risk of mortality. For the fixed sensitivity requirement, we achieve a classification accuracy of 0.85 (95% CI, 0.81-0.89) in the testing set with specificity of 0.87 (95% CI, 0.83-0.90); see Table 5.3. The model generalizes better in the Seville cohort with an accuracy of 0.87 (95% CI, 0.82-0.91) and specificity of 0.87 (95% CI, 0.83-0.92). The necessary threshold for a sensitivity of 80% is lower for the Hellenic CSG compared to the other populations. This is due to the low baseline incidence of mortality in this sample when compared to the derivation and other validation cohorts. The model achieves lower performance in this set of patients, with an accuracy of 0.75 (95% CI, 0.7-0.8) and specificity of 0.74 (95% CI, 0.69-0.79). For Hartford Hospital, the accuracy of CMR is 0.61 (95% CI, 0.56-0.67) with a specificity of 0.58 (95% CI, 0.53-0.64).

5.2.3 Model results

Through the SHAP framework, we identified the most important drivers of mortality risk and the interplay between individual features. For a particular patient, SHAP values indicate the feature

contributions towards the risk. The patient risk normalized between 0 and 1 is the sum of the SHAP values of all the features. Figure 5-1a displays the risk contributions of the 10 most important features. For example, higher values of age (red) yield higher SHAP values, suggesting that older patients are at higher risk. In contrast, the SHAP value increases with lower values (blue) of Oxygen Saturation, suggesting an inverse relationship with this feature.

When BUN is below 20 mg/dL, the mortality risk decreases, particularly for ages below 55 years. On the other hand, BUN values greater than 25 mg/dL for older patients increase the risk (Figure 5-1b). A C-reactive protein (CRP) between 50 and 130 mg/L does not affect the risk, independent of age. As CRP goes below 50 mg/L, the mortality risk decreases. For a CRP above 160 mg/L, the elevated risk does not change and is higher for older patients (Figure 5-1c). An oxygen saturation below 93% increases the mortality risk rapidly and this trend is accelerated by growing age (Figure 5-1d). A blood creatinine level greater than 1.2 mg/dL increases the risk moderately, specifically for older patients. Levels above 3 mg/dL rapidly escalate the mortality risk (Figure 5-1e). Figure 5-1f illustrates that while a blood glucose less than 130 mg/dL lowers the risk, it can increase the risk for levels above 180 mg/dL, in particular for older patients. An aspartate aminotransferase (AST) level above 65 U/L increases the risk, while a level below 25 U/L sharply decreases the risk, independent of age (Figure 5-1g). A platelet count in 103/μL affects the risks in 4 distinct ranges: (i) below 50 the risk is elevated, (ii) between 50 and 180 the risk is marginally increased (more for older patients), (iii) between 180 and 330 the risk is slightly decreased, and (iv) above 330 the risk is sizably decreased (Figure 5-1h). Figure 5-1i shows that a mean corpuscular volume (MCV) between 90 and 94 fL increases the risk moderately, while other values have only small effects. Lastly, an increased risk is observed when white blood cell (WBC) count is above 10 in 103/μL, in particular for older patients (Figure 5-1j).

5.3 Discussion

The CMR calculator predicts mortality with high accuracy using clinical measurements collected early within a patient's hospital admission. An early risk assessment of patient mortality allows physicians to triage patients and prioritize resources in a highly congested system. It uses commonly available laboratory results and does not require imaging results or advanced testing. The

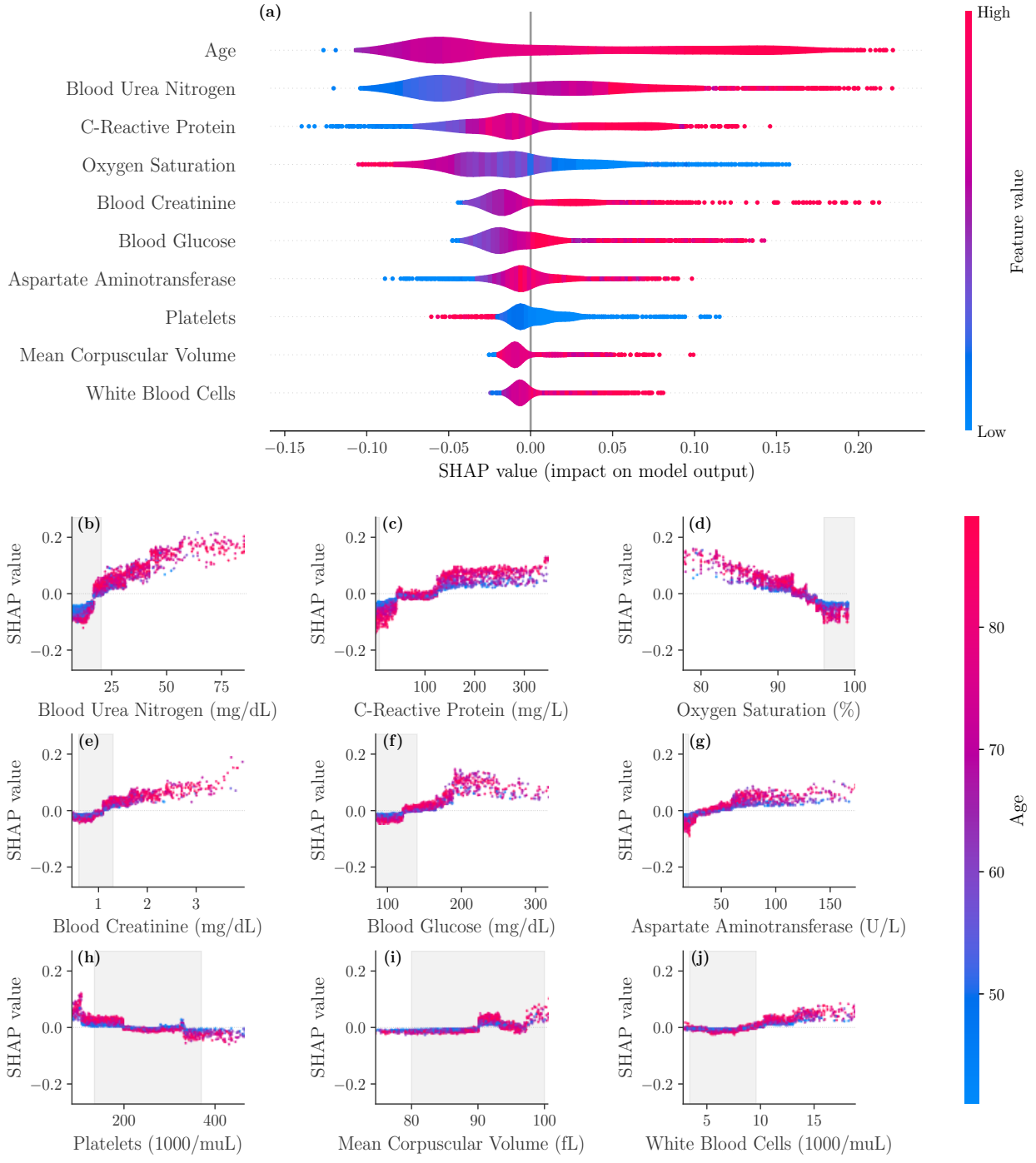


Figure 5-1: SHAP importance plots for final model.

SHAP importance plots: (a) for order of the top 10 features in the model; (b-j) for impact of each feature's values on risk and the interaction with age. Gray shaded area highlight the reference range.

presented tool can be particularly useful in lower acuity facilities or remote hospitals with constrained diagnostic capabilities.

Age is the most important determinant of mortality in the model: older patients have higher mortality risk, which has been observed in retrospective patient analysis [176] and subsequently reflected in public health guidance [43]. Predicted mortality also increases for patients with low oxygen saturation, corroborating findings that link hypoxemia to mortality [172], as well as the observed prevalence of shortness of breath in severe patients [166]. This measurement additionally serves as signal of respiratory distress, and respiratory failure has been found clinically as one of the major mortality causes of COVID-19 [142]. This can also appear in cases of silent hypoxia where shortness of breath is not observed [168].

Our study finds that elevated BUN, CRP, creatinine, glucose, AST, and platelet counts are highly significant laboratory features. Several of these biomarkers have been identified in other retrospective analyses of mortality outcomes of COVID-19 [142, 46]. Prior work has also uncovered the critical role of these biomarkers in identifying severe cases of patients with community acquired pneumonia [101]. The PSI, CURB-65, and SCAP scores are also based on similar risk factors such as glucose levels ≥ 250 mg/dL and BUN > 19 mg/dL [106, 68]. Moreover, CRP levels have been recognized to characterize severity for H1N1 patients [3].

CRP is a widely available inflammatory marker which has been independently observed as a biomarker of COVID-19 severity [173, 165]. Our findings show that CRP values outside the reference ranges do not necessarily increase the risk of mortality. In fact, CRP has a negative effect on mortality until approximately 50 mg/L, it has a negligible effect between approximately 50mg/L and 130 mg/L, and it significantly increases the mortality risk above 130mg/L. Elevated BUN and creatinine levels are both indicative of impaired kidney function, which has been associated with poor prognosis [49]. The individual feature plots indicate a clear transition from low to high risk when BUN exceeds approximately 18 mg/dL and creatinine exceeds approximately 1.2 mg/dL. These values are slightly lower than reference ranges for these values, providing data-driven validation of the ranges [161] targeted for COVID-19. The increase in mortality risk for patients with elevated glucose levels is consistent with the reports in other studies of diabetes as a risk factor [176, 53]. Elevated AST levels have been observed due to liver dysfunction in severe COVID-19 cases [77]. Finally, low platelets are associated with increased risk, which match

findings of thrombocytopenia in critical COVID-19 patients [107].

We recognize that the derivation populations may differ from other populations based both on hospital conditions and inherent demographic differences. An external validation using Seville, Greece, and US populations allows us to assess the broader clinical utility of our findings. The CMR model performs well on these patients, with the strongest performance observed in Seville. Seville consists of a South European population similar to the majority of the derivation cohort. However, it did not face the same capacity challenges as ASST Cremona and HM Hospitals during the study period. Greece had a significantly lower disease spread, resulting in a lower mortality rate compared to the derivation population. Nevertheless, the model yields comparable results in this cohort to the other European hospitals. Hartford has the weakest validation performance, which may suggest inherent differences between Europe and the US in disease dynamics, treatment protocols, or underlying population susceptibility. This attests to the need for training and validation on a diverse set of populations. We observe that the thresholds needed for obtaining 80% sensitivity differ across the external validation cohorts. When applying the CMR tool to a new hospital, the threshold should be calibrated to the severity of this population. A sample of historical patients at the hospital can be used to validate the model. Using the risk predictions and true outcomes of this sample, various risk thresholds can be evaluated for sensitivity and specificity. Clinicians can determine the relevant threshold for their hospital's needs. For example, highly constrained systems may employ a higher threshold (lower sensitivity) due to capacity limits, whereas other centers may use this tool as an initial screening tool where sensitivity is required to be very high.

Risk models are most useful when they are readily available for healthcare clinicians. For this reason, a dynamic online application has been created as the interface of the CMR model for use by clinical providers. Figure 5-2 provides a visualization of the application that is available at covidanalytics.io/mortality_calculator. After entering a patient's clinical features, the model returns a predicted mortality risk. It additionally produces a SHAP plot to elucidate the major factors contributing to an individual patient's risk score. Features in blue decrease risk from the population baseline, whereas features in red increase risk. The contribution is proportional to the width of the feature's bar. In the example, we see that the patient's age and oxygen saturation levels increase his risk assessment, but his temperature and glucose lower his risk. The CMR tool is currently undergoing prospective validation at two of the collaborating institutions in the study:

the application is in use in the emergency room of ASST Cremona to prioritize hospitalizations on higher risk patients, and the model also interoperates with the EMR of the Virgen del Rocío University Hospital in Seville, Spain.

5.3.1 Limitations

Limited hospital capacity can impose potential biases in the training population. Only severe patients were able to be treated, particularly in Europe, and some hospitals were forced to turn away patients deemed too critically ill during the peak of the virus. Thus, hospital admissions data may exclude patients on both ends of the acuity spectrum. Additionally, the scarcity of hospital resources may have led patients to receive insufficient care, increasing mortality risk due to lack of treatment. While this warrants further investigation, initial validation results suggest that the CMR tool generalizes well to less congested systems in Greece and the United States. The differences related to Hartford Hospital might also be related to the timing of the virus. The virus affected Europe before the US. This provided an opportunity to learn from the experience in Europe, which may have resulted in different or more effective treatment decisions as well as governmental policies in the US. This is an opportunity for further study through validation on additional US cohorts.

Our clinical features are limited by the data that was commonly available across all sites in the derivation population. We expect that a more comprehensive set of clinical features such as D-Dimer and IL-6 levels, Body Mass Index, radiographic diagnosis, symptoms, and time elapsed between the disease and treatment onset will yield more accurate results. A broader set of comorbidities, including hypertension, cancer, chronic obstructive pulmonary disease, and others could be included when available. Recent reports on racial disparities and socio-economic determinants of COVID-19 severity [1, 97] could be addressed through the incorporation of additional demographic data and external data sources.

Additionally, there is significant variability in treatment protocols across countries and individual organizations. In future work, we hope to expand the set of captured clinical features and incorporate treatments to disentangle some of the observed heterogeneity in outcomes and clinical characteristics.

5.4 Conclusion

This international study provides a mortality risk calculator of high accuracy for hospitalized patients with confirmed COVID-19. The CMR model validates several reported risk factors and offers insights through a user-friendly interface. Validation on external data shows strong generalization to unseen populations in both Europe and the United States and offers promise for adoption by clinicians as a support tool.

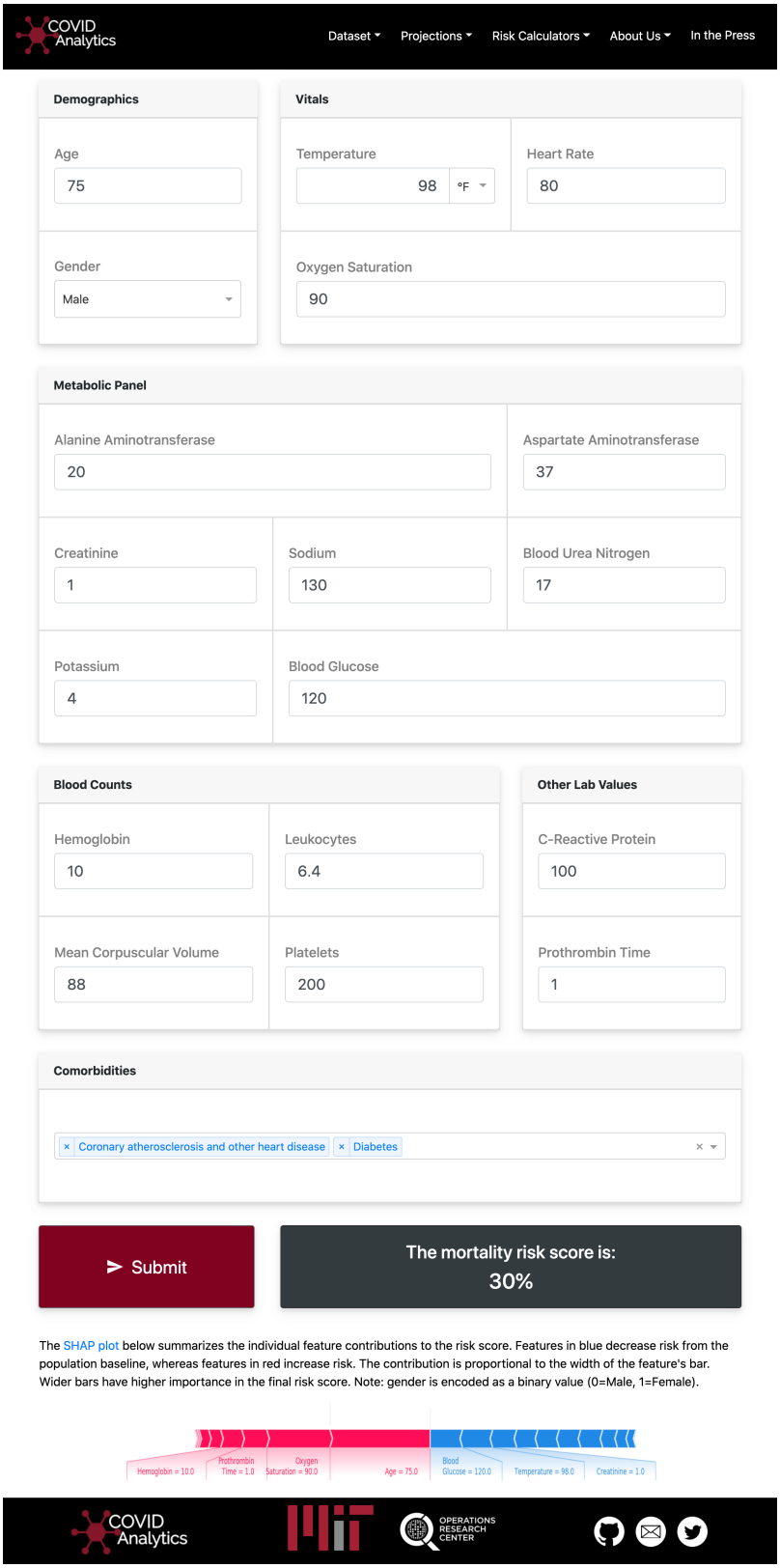


Figure 5-2: Visualization of the Calculator interface. Using the SHAP package, personalized interpretations of the predicted score are provided to the user.

Chapter 6

Optimizing virtual care for chronic disease patients: a case study in diabetes

Abstract

Telemedicine has ushered in a new era of healthcare delivery. The incorporation of virtual care can benefit patients and providers, enabling more frequent touchpoints and lowering geographic barriers to access. However, these benefits must be weighed against potential competing effects, such as lower patient engagement or the introduction of technological barriers. The COVID-19 pandemic accelerated the adoption of virtual care. In its aftermath, healthcare providers and policymakers must devise a long-term strategy for how to leverage telehealth in combination with traditional care modalities. In this work, we propose an integrated machine learning and optimization framework for scheduling virtual and in-person visits. We take a causal inference perspective to visit modality decisions, considering this as a treatment with potential effects on clinical and operational outcomes. We obtain treatment effect estimates through machine learning models, which are then used as inputs into an optimization model for patient scheduling. We study the endocrinology practice of a large medical center, with A1C control and no-shows as the two outcomes of interest. We find that increasing telehealth utilization is beneficial with respect to both maximizing A1C control and minimizing no-shows, with greater increases in telehealth recommended as the no-show objective is prioritized. Overall, our machine learning models provide personalized insights into virtual care effectiveness for diabetic patients, and the optimization model yields a tactical tool for jointly harnessing both visit types.

6.1 Introduction

Virtual care offers an opportunity to improve patient engagement, access, and provider efficiency within a healthcare system. The COVID-19 pandemic led to widespread adoption of virtual

care [88, 157]. As the pandemic recedes, the healthcare industry must determine the long-term vision for telehealth's role in patient care. While virtual care has garnered interest in recent years, it had not been used widely due to technology gaps and insurance reimbursement policies. The high utilization of virtual visits in the past two years provides, for the first time, large-scale data on telehealth visits and outcomes. This enables an investigation of the effectiveness and tradeoffs involved in virtual care across a variety of specialties.

In this work, we consider virtual care in the endocrinology division of a large medical group. Diabetic patients are often treated over long periods of time, involving a combination of clinical interventions and monitoring or follow-up visits. Many patients also receive other forms of care, such as diabetes education or behavioral health. This chronic care setting is well-suited for expanding virtual care, given the frequency of visits and incorporation of non-clinical interventions. Both frequent touch points [121] and auxiliary services [126] have been shown to be effective intervention strategies. Additionally, technological advancements have enabled remote monitoring of patient A1C values, a key indicator of diabetic status. Continuous glucose monitors provide real-time A1C measurements for patients and are able to transmit reports to providers. This gives providers visibility into their patients' progress without requiring point-of-care testing [42, 116].

There are various considerations involved in virtual vs. in-person care [115]. From an operational perspective, visit types differ in allowable schedule density, overhead for setting up appointments, and potential no-show rates. There are also patient access implications: virtual care generally lowers barriers to access (e.g. by geography). However, there is also a potential competing effect of technology barriers that can increase disparities for under-served populations [115, 140]. Finally, different policies of in-person and virtual care can affect clinical outcomes through their impact on patient engagement, care continuity, and appointment quality. A recent study of diabetes management in COVID-19 found no adverse clinical effect of shifting to telehealth [13], although there is no conclusive guidance on how to best leverage both modalities. Ashrafzadeh and Hamdy [10] provide a comprehensive review of the telehealth utilization and effectiveness in diabetic care.

The potential benefits–and competing drawbacks–of telehealth motivate further study of the effectiveness of virtual vs. in-person care modalities. We pose this as a treatment effect estimation problem, where we represent the visit type as a treatment and estimate its impact on outcomes based on the patient and appointment characteristics. While causal frameworks have been applied

to study the impact of various COVID-19 interventions [104], this is, to our knowledge, the first study that takes a causal approach to understanding telehealth vs. in-person visit outcomes and optimal appointment assignment.

The combined use of telehealth and in-person visits has received limited attention in the operations research community. While schedule optimization has been long-studied, existing work generally assumes that all visits occur in-person. In recent years, telehealth schedule optimization has gained interest. Ayca Erdogan et al. [14] propose a two-stage stochastic linear programming model for scheduling remote patient visits with no-show and provider idle time considerations. Ji et al. [92] consider a two-stage robust optimization approach for telehealth scheduling that incorporates uncertainty in provider behavior. These works operate in a pure telehealth setting and thus fundamentally differ from the present work, in which we seek to inform the choice between encounter modalities. Our work is also distinct in how outcome estimates (e.g., no-shows) are obtained. We take a data-driven, nonparametric approach, using machine learning (ML) models in a causal inference framework to estimate outcomes under both encounter types.

In this work, we compare care effectiveness for virtual and in-person visits, as measured by both clinical and operational metrics, and propose an scheduling model that optimizes the use of both visit types. Our contributions are as follows:

- We conduct a retrospective analysis of virtual and in-person endocrinology care in a large regional medical group. Virtual care was introduced in March 2020 as a result of the COVID-19 pandemic and has continued to be utilized. This provides us with data to compare treatment modalities and obtain insights regarding clinical outcomes, system utilization, and department operations.

- We propose a causal approach to estimating the effect of visit modality on the outcomes of interest. We model the visit modality, virtual vs. in-person, as a binary treatment decision. We train causal ML models for each outcome and implement a rigorous model selection and evaluation pipeline.

- We use the treatment effect estimates as inputs to a scheduling model, linking the ML models to decisions through a prescriptive optimization model. Our framework can flexibly incorporate multiple clinical and operational outcomes that depend on treatment modality, and

these outcomes can appear as objective or constraint terms. The scheduling model additionally captures various system constraints, integrating a traditional optimization model with learned treatment effects to inform decision-making.

This work allows departments to quantify the efficiency and clinical impact of various care delivery models that jointly leverage virtual and in-person care. We provide a data-driven approach to tackle questions such as: how effective is virtual care, and what is the ideal balance between in-person and virtual visits?

6.2 Data overview

We study the patients of two high-volume endocrinology providers in Hartford HealthCare's Medical Group (HHCMG). We consider patients with established endocrinology care, defined by the completion of at least two endocrinology visits recorded in HHCMG's EMR, which was implemented in 2017. We further restrict to patients with a history of diabetes, excluding patients who have never been active on HHCMG's diabetes registry. We restrict to encounters for which the patient has both a previous and future A1C reading, as this is the primary clinical metric of interest. We consider encounters that occur between April 2020 through December 2021 with a status of completed, no-show, or same-day cancellation. We exclude all other cancellations, since COVID-19 posed a major disruption to pre-existing scheduled visits. We identify completed visits by matching encounters to claims based on the patient identifier and service date. We perform a temporal split of the data. Encounters from April - December 2020 comprise our training data; encounters from January - March 2021 are used as our validation data for model selection; and encounters from April - December 2021 are used to test the models.

Encounter Features Our feature space consists of static patient features, time-varying features that reflect a patient's current health state, and appointment schedule characteristics. We represent patients by their demographics (age, sex, race, and ethnicity), registration status for HHCMG's patient portal ("MyChart"), and whether they have Medicaid or Medicare as their primary insurance payor. The MyChart variable acts as a proxy for technology use and access, and the Medicaid indicator as a socioeconomic status proxy. For clinical features, we include indicators of common

encounter diagnoses (diabetes without complication, diabetes with complication, thyroid disorders, other endocrine disorders, and osteoporosis), cumulative completed appointment counts for each specialty in HHCMG (e.g., internal medicine, endocrinology, nephrology), and the patient's most recent recorded A1C value. We also record a provider identifier, the appointment day of week, and appointment time. Finally, each visit is scheduled as either "in-person" or "virtual"; this is our treatment variable. Throughout this work, "virtual" care and telehealth are used interchangeably, referring to encounters that are conducted remotely by video call or telephone. "In-person" and office visits, refer to traditional appointments where a patient travels to the provider's office.

Outcomes We consider two outcomes of interest.

- Visit Completion: We predict whether a scheduled appointment is completed ($y = 1$) vs. a no-show or same-day cancellation ($y = 0$). Visit completion provides a metric of operational efficiency for the provider. It also indirectly reflects accessibility and patient engagement.

- A1C Control: We predict whether the patient's next A1C value is controlled. We define A1C control using an upper bound of 7.0%, as recommended by the American Diabetes Association [6]. Formally, $y = 1$ if the patient's next A1C value is $\leq 7.0\%$; $y = 0$ otherwise. This outcome quantifies clinical effectiveness of the visit modality.

6.3 Methods

Our modeling pipeline has two key components. First, we use causal ML to derive insights from data over the past year, allowing us to quantitatively define the impact of care type on the above outcomes. These models help answer questions about how telehealth vs. in-person care affects the outcomes of interest, and how other patient attributes (like baseline health, demographics, insurance type, etc.) contribute as well. Next, we then apply an optimization framework that uses the resultant ML models to optimize care according to HHCMG's priorities and constraints.

6.3.1 Problem notation

The encounter features available at encounter i, including both static and time-varying features, are represented as X_i. The visit modality is considered a treatment; this is the decision that we have control over in optimizing care. The treatment received at visit i is encoded as:

$$Z_i = \begin{cases} 1 & \text{if visit } i \text{ is virtual} \\ 0 & \text{if visit } i \text{ is in-person} \end{cases}$$

For visit i, Y_i^a indicates whether the patient's next A1C value is controlled, i.e., at most 7%. Y_i^c indicates whether the visit is completed.

6.3.2 Causal models

We train models to estimate the treatment effect of a virtual vs. in-person visit at encounter i, given information available prior to the appointment. We use a doubly robust (DR) estimator to estimate outcomes under each treatment alternative [15, 139]. This method combines both direct estimation and inverse-propensity weighting. The direct estimation component predicts outcomes as a function of features X and treatment Z, namely $f_Y(X,Z)$. As an alternative approach, inverse-propensity weighting trains a model to predict the probability that an observation was assigned each treatment and then estimates potential outcomes by reweighting each observation X's outcome under treatment Z by the probability that they received this treatment [86, 138].

A DR estimator combines the two approaches. Suppose we have an observation X_i with treatment z_i and observed outcome y_i. For each treatment t, the direct estimator provides us with an outcome estimates $f_Y(X,t)$, and the treatment classifier $f_Z(X)$ yields a probability $\mathbb{P}(z_i = t)$. The estimate $y_{i,t}$ is then computed as:

$$\hat{y}_{i,t} = f_Y(X,t) + \mathbb{1}[z_i = t] * \frac{y_i - f_Y(X,t)}{\mathbb{P}(z_i = t)} \tag{6.1}$$

Informally, $\hat{y}_{i,t}$ is a corrected version of the direct estimator, where the actual observed outcome, adjusted by treatment propensity, is incorporated for predicting the outcome under the observed treatment. When predicting $\hat{y}_{i,t}$ for unobserved treatments, i.e., $t \neq z_i$, the direct estimator is used.

This treatment effect estimator is consistent if either the outcome model or propensity model are correctly specified [139]. This robustness to to potential misspecification of either modeling component is particularly relevant when learning treatment effects from observational data, as in this virtual care setting.

The DR estimator is obtained through the outcome and treatment prediction models, which are trained using standard supervised learning methods given. We train a classification model for treatment propensity, where the binary treatment assignment is a function of X, i.e., $f_Z(X)$. We train a regression model for the outcome prediction, where Y is a function of X and T, i.e., $f_Y(X,T)$. As in `OptiCL` [117], we employ a model selection pipeline with five-fold cross-validation to train and select ML models for $f_Y(X,T)$ and $f_Z(X)$, considering linear models, decision trees, and ensemble methods. The models for the treatment and outcome tasks are not restricted to lie in the same class. This nonparametric approach allows us to capture potential nonlinearities in the predictive tasks.

Once the estimated rewards have been obtained, we train a causal model to predict individual treatment effects, where the outcome targets for the causal model are estimated by the DR estimator. We denote the estimated effect of a virtual (vs. in-person) appointment on visit completion for patient k as $\tilde{y}_k^c$ and the estimated effect on A1C control as $\tilde{y}_k^a$. The effect is defined as:

$$\tilde{y}_k = \hat{y}_{k,1} - \hat{y}_{k,0}.$$

We train multiple such models and select between them by scoring their performance on the validation set. We consider causal forests [12] as well as a regularized linear regression model [71]. The modeling pipeline is implemented in Python v3.7.4 using `scikit-learn` [131] and `econml` [17]. We repeat the effect estimation for both outcomes of interest, A1C control and visit completion.

6.3.3 Mixed-integer optimization model

We next apply the causal models developed in Section 6.3.2 to optimize the weekly schedule for a single endocrinologist. We frame the following problem: given an existing schedule, which visits should be “flipped" from in-person to virtual, or vice versa, to improve the outcomes of interest? This can be viewed as a modified scheduling problem, where we incorporate *learned*

effects for A1C ccontrol ($\tilde{y}_k^a$) and visit completion ($\tilde{y}_k^c$). Note that in this setting, we consider a binary treatment, so each patient decision has two potential outcomes. This stands in contrast to the more general `OptiCL` setting, in which decisions are multi-dimensional. In this case, we are able to pre-compute the estimated treatment effects and do not need to directly embed the trained models. This significantly reduces the problem dimensionality and allows us to incorporate causality into our estimates, but it also reduces the richness of the decisions that we can optimize.

A schedule is defined by slots, which are day-time pairs (i, j) where the day is in set $\mathscr{S}_{day}$ and appointment time is in set $\mathscr{S}_{time}$. We also have a set of patients to schedule, given by set $\mathscr{P}$. Our decision variables are defined as:

$$z_{ijk}^v = \begin{cases} 1 & \text{if slot } (i,j) \text{ is assigned to patient } k \text{ for a virtual visit} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ijk}^o = \begin{cases} 1 & \text{if slot } (i,j) \text{ is assigned to patient } k \text{ for an in-person visit} \\ 0 & \text{otherwise} \end{cases}$$

for all $i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time}, k \in \mathscr{P}$.

Given the schedule assignment, we can compute the average effect of the visit modality on each outcome of interest. For A1C control, the estimated treatment effect for a patient scheduled in slot (i, j) is given as:

$$\tilde{y}_k^a (z_{ijk}^v - z_{ijk}^o).$$

This follows from the fact that $\tilde{y}_k^a$ gives the estimated effect of a virtual visit over an in-person visit. Suppose that $\tilde{y}_k^a = 0.1$. Then the chosen visit modality will have an estimated effect of $+0.1$, improving the likelihood A1C control, if the patient is scheduled for a virtual visit ($z_{ijk}^v = 1$), or an estimated effect of -0.1, reducing the likelihood of A1C control, if the visit is in-person. By averaging over all patients and slots, we obtain an average treatment effect of the assigned schedule.

We begin with an existing schedule, $(\bar{\mathbf{Z}}^{\mathbf{v}}, \bar{\mathbf{Z}}^o)$, with indicators for the current allocation of visits and slots to patients, i.e. $\bar{z}_{ijk}^v = \mathbb{I}\{\text{slot } (i,j) \text{ is } \textit{currently} \text{ assigned to patient } k \text{ for a virtual visit}\}$, and

likewise for $\bar{z}^o_{ijk}$. This schedule provides a baseline for the model and informs several key parameters defined in Table 6.1. In the absence of an existing schedule, or in the presence of additional context, these parameters can be set in other ways. For example, we assume that a provider's availability corresponds exactly to their current schedule, but a provider could submit their availability through a preference form to determine their open slots. Similarly, we propose lower bounds on the average effect of visit completion and A1C control to ensure that the proposed solution does *no worse* than the existing schedule. This reflects the schedule re-optimization setting, in which we assume there is a baseline that we are amending, rather than constructing a schedule from scratch. These could be replaced by department-level benchmarks, or the corresponding constraints in the optimization model could be removed entirely.

Parameter	**Definition**	**Calculation**
D	Total visit demand for week	$\sum_{i,j,k}(\bar{z}^v_{ijk} + \bar{z}^o_{ijk})$
A_{ij}	Indicator of provider availability in slot (i, j)	$\sum_k(\bar{z}^v_{ijk} + \bar{z}^o_{ijk})$
L_c	Lower bound on visit completion effect	$\sum_{i,j,k}\left[\tilde{y}^c_k(\bar{z}^v_{ijk} - \bar{z}^o_{ijk})\right]$
L_a	Lower bound on A1C control effect	$\sum_{i,j,k}\left[\tilde{y}^a_k(\bar{z}^v_{ijk} - \bar{z}^o_{ijk})\right]$

$\sum_{i,j,k}$ implies a sum over $i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time}, k \in \mathscr{P}$.

Table 6.1: Input data for scheduling model.

We further enforce some patients' visits to remain fixed to their original modality. These patients comprise the sets $\mathscr{P}_v$ and $\mathscr{P}_o$, which are defined through the following rules:

- Any new patient k must be seen in person ($k \in \mathscr{P}_o$).
- Any non-diabetic patient k on the schedule must remain fixed to their original modality. If $\bar{z}^v_{ijk} = 1$ for some (i, j), then $k \in \mathscr{P}_v$; otherwise, $k \in \mathscr{P}_o$.
- If a patient k does not have data available for causal estimates, their visit modality must remain fixed. For example, if a patient does not have a previous A1C value available, we are unable to estimate the treatment effects of in-person vs. virtual care and thus keep their original appointment.

Each of these sets $\mathscr{P}_v, \mathscr{P}_o$ are subsets of the patient base $\mathscr{P}$. Furthermore, $\mathscr{P}_v \cap \mathscr{P}_o = \emptyset$. These sets could be modified based on other factors, such as patient preference or transportation ability.

Finally, there are user-specified parameters that inform the model. V_{max} denotes the maximum allowable proportion of visits that can be held virtually. α controls the relative weight of A1C control and visit completion in the objective; the objective only optimizes for A1C control if $\alpha = 1$, and only for visit completion if $\alpha = 0$.

Given the model inputs and selected parameters, we propose the following optimization model:

$$\max_{z^v, z^o} \frac{\alpha}{D} \sum_{i,j,k} \left[\tilde{y}^a_k (z^v_{ijk} - z^o_{ijk}) \right] + \frac{1-\alpha}{D} \sum_{i,j,k} \left[\tilde{y}^c_k (z^v_{ijk} - z^o_{ijk}) \right] \tag{6.2a}$$

$$\text{s.t.} \sum_{i,j,k} \left[\tilde{y}^a_k (z^v_{ijk} - z^o_{ijk}) \right] \geq L_a \tag{6.2b}$$

$$\sum_{i,j,k} \left[\tilde{y}^c_k (z^v_{ijk} - z^o_{ijk}) \right] \geq L_C \tag{6.2c}$$

$$\sum_{i,j,k} z^v_{ijk} \leq D * V_{max} \tag{6.2d}$$

$$z^o_{ijk} = 1, \qquad k \in \mathscr{P}_o, i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time} \tag{6.2e}$$

$$z^v_{ijk} = 1, \qquad k \in \mathscr{P}_v, i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time} \tag{6.2f}$$

$$\sum_{i,j} \left[z^v_{ijk} + z^o_{ijk} \right] = 1 \qquad k \in \mathscr{P} \tag{6.2g}$$

$$\sum_{k \in \mathscr{P}} \left[z^v_{ijk} + z^o_{ijk} \right] \leq A_{ij} \qquad i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time} \tag{6.2h}$$

$$z^v_{i,j,k} \in \{0,1\} \qquad k \in \mathscr{P}, i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time} \tag{6.2i}$$

$$z^o_{i,j,k} \in \{0,1\} \qquad k \in \mathscr{P}, i \in \mathscr{S}_{day}, j \in \mathscr{S}_{time}. \tag{6.2j}$$

Constraints 6.2b and 6.2c enforce lower bounds on the average effects, as described in Table 6.1. Constraint 6.2d imposes a maximum allowable proportion of virtual visits, which might be guided by department-level or payer mandates. This is applied at the weekly level, although it could easily be enforced daily.

The remaining constraints ensure schedule feasibility. Constraints 6.2e and 6.2f fix the visit modality for any patients in $\mathscr{P}_o$ or $\mathscr{P}_v$. Constraint 6.2g ensures that the demand is met: every patient must be scheduled. Constraint 6.2h imposes that a visit can only be scheduled in slot (i, j) if the provider is available.

This formulation optimizes a provider's schedule for a fixed set of patients. In its general form,

the appointment day, time, and modality are all determine by the model. In our case study, we apply a final additional constraint that all patients must be seen in their original slot. In other words, the visit modality is the only decision lever.

6.4 Results

In this section, we present the results from our case study on HHCMG's endocrinology group. We first examine the data scope and key features. We then quantitatively evaluate the models used to generate treatment effect estimates, namely the DR estimator component models and final causal model. Finally, we generate proposed schedules for several weeks of data according to the estimated treatment effects. We perform a sensitivity analysis of the proposed schedules to the V_{max} and α parameters. We also compare the proposals to the true observed schedules in terms of virtual care utilization and outcomes.

6.4.1 Data scope

There are 2546 encounters that meet the inclusion criteria, across 721 unique patients, of which 30.79% were virtual. A descriptive summary of all encounters, split by visit modality, is included in Table 6.2. Significance tests are performed for each feature, using a Wilcoxon rank-sum test for continuous features and Chi-square test for binary features. There are significant differences in patient demographics for virtual and in-person visits. Patients being seen through virtual visits tend to be younger ($p < 0.001$) and have different racial ($p < 0.001$) and ethnic ($p = 0.046$) backgrounds than patients seen in person. These patients also generally have fewer cumulative endocrinology appointments ($p < 0.001$) and more cumulative internal medicine appointments ($p = 0.033$), indicating more consistent established primary care and potentially lower complexity or more recent endocrinology needs. Appointments earlier in the week ($p < 0.001$) and earlier in the day ($p < 0.001$) are also more likely to be virtual. For the outcomes of interest, virtual visits have lower A1C control rates than in-person visits (54.85% vs. 57.32%, $p = 0.263$) but higher visit completion (95.41% vs. 92.68%, $p = 0.013$).

Our temporal split yields the following partition: the training data (April - December 2020) consists of 1239 observations, with 49% virtual; the validation data (January - March 2021) con-

Feature	All ($N = 2,546$)	Office ($N = 1,762$)	Virtual ($N = 784$)	P-Value
Sex = Female	1560 (61.27%)	1065 (60.44%)	495 (63.14%)	0.213
Age	63.00 (53.00-71.00)	64.00 (54.00-71.00)	61.00 (50.00-69.00)	<0.001
Race				<0.001
White or Caucasian	2046 (80.36%)	1456 (82.63%)	590 (75.26%)	
Asian	144 (5.66%)	106 (6.02%)	38 (4.85%)	
Black or African American	127 (4.99%)	65 (3.69%)	62 (7.91%)	
American Indian or Alaska Native	3 (0.12%)	1 (0.06%)	2 (0.26%)	
Other	152 (5.97%)	85 (4.82%)	67 (8.55%)	
Unknown	94 (3.69%)	62 (3.52%)	32 (4.08%)	
Ethnicity				0.046
Not Hispanic or Latino	2315 (90.93%)	1616 (91.71%)	699 (89.16%)	
Hispanic or Latino	130 (5.11%)	80 (4.54%)	50 (6.38%)	
Unknown	101 (3.97%)	66 (3.75%)	35 (4.46%)	
MyChart Status				0.326
Activated	2194 (86.17%)	1510 (85.70%)	684 (87.24%)	
Pending Activation	209 (8.21%)	146 (8.29%)	63 (8.04%)	
Inactivated	93 (3.65%)	68 (3.86%)	25 (3.19%)	
Patient Declined	50 (1.96%)	38 (2.16%)	12 (1.53%)	
Primary Insurance				<0.001
Medicare	555 (21.80%)	402 (22.81%)	153 (19.52%)	
Medicaid	225 (8.84%)	129 (7.32%)	96 (12.24%)	
Appointment Diagnosis				0.027
Diabetes mellitus with complications	1242 (48.78%)	859 (48.75%)	383 (48.85%)	
Diabetes mellitus without complication	757 (29.73%)	530 (30.08%)	227 (28.95%)	
Thyroid Disorders	143 (5.62%)	105 (5.96%)	38 (4.85%)	
Other endocrine disorders	88 (3.46%)	51 (2.89%)	37 (4.72%)	
Osteoporosis	15 (0.59%)	9 (0.51%)	6 (0.77%)	
Last A1C	7.00 (6.30-8.00)	7.00 (6.23-7.90)	7.10 (6.30-8.00)	0.128
Cumulative endocrinology appointments	7.00 (4.00-11.00)	8.00 (4.00-12.00)	5.00 (3.00-8.00)	<0.001
Cumulative internal medicine appointments	2.00 (0.00-11.00)	1.00 (0.00-11.00)	3.00 (0.00-12.00)	0.033
Appointment day of week	2.00 (1.00-4.00)	2.00 (2.00-4.00)	2.00 (1.00-3.00)	<0.001
Appointment time (Hour)	13.00 (10.00-14.00)	13.00 (10.00-15.00)	11.00 (9.00-14.00)	<0.001
A1C control	1440 (56.56%)	1010 (57.32%)	430 (54.85%)	0.263
Visit complete	2381 (93.52%)	1633 (92.68%)	748 (95.41%)	0.013

Table 6.2: Descriptive summary of encounter characteristics, split by visit modality.

tains 485 visits, with 25% virtual; the testing data (April - December 2021) contains 825 visits, with 7% virtual.

6.4.2 Treatment effect estimation

To assess the performance of the ML models, we first consider the predictive performance of the component models in the DR estimator, trained with the training set. The reward estimation involves both a treatment propensity and outcome prediction model. Given that the treatment decision and outcomes of interest (A1C control and visit completion) are binary, we evaluate the

quality of each of these predictive models using area under the ROC curve (AUC).

Table 6.3 shows the results of the model selection procedure for the three predictive models, derived from the training data ("Training Model"): treatment (virtual = 1, vs. in-person = 0), A1C control (0/1), and visit completion (0/1). The outcome models have strong predictive performance (test AUCs 0.875-0.959), while the treatment model has moderate results (test AUC 0.693). The selection of tree-based ensemble methods, random forests (RF) and gradient boosting machines (GBM), suggests nonlinear dependence on the input variables. With the predictive models fixed, a causal forest (CF) model was selected for both treatment effect estimation problems, outperforming the regularized linear model on the validation set by 0.5% in the A1C control model and 13.7% in the visit completion model.

Task	**Model**	**Train AUC**	**Test AUC**
Treatment	GBM	0.774	0.693
Outcome - A1C Control	GBM	0.979	0.875
Outcome - Visit Completion	RF	0.928	0.959

Table 6.3: Selected predictive models for treatment and outcome classification tasks.

We repeat this procedure to derive models from our testing data ("Testing Model"), which are used in model evaluation. The test set models allow us to approximate the rewards that are observed in the test set, independent of the training set, which provides an objective evaluation of the optimization model's prescriptions on out-of-sample treatment outcomes. The models exhibit strong quantitative performance, with AUCs of 0.944 for treatment (GBM model), 0.956 for A1C control (GBM model), and 0.980 for visit completion (RF model).

6.4.3 Optimization results

We next turn to evaluation of the schedule optimization model. The causal models from Section 6.3.2 are used to estimate the treatment effect parameters, $\tilde{y}_k^a$ and $\tilde{y}_k^c$, for each patient visit k. We generate optimized schedules obtained over a range of objective weights, $\alpha \in \{0, .1, .2, \ldots, 1.0\}$ and maximum virtual visit proportions, $V_{max} \in \{0.2, 0.3, 0.4, 0.5\}$. We repeat this procedure for five weeks of appointments in our test window (April 2021 onwards). We compare the average treatment effect under alternative schedules, including the baseline policy. In all cases, we report

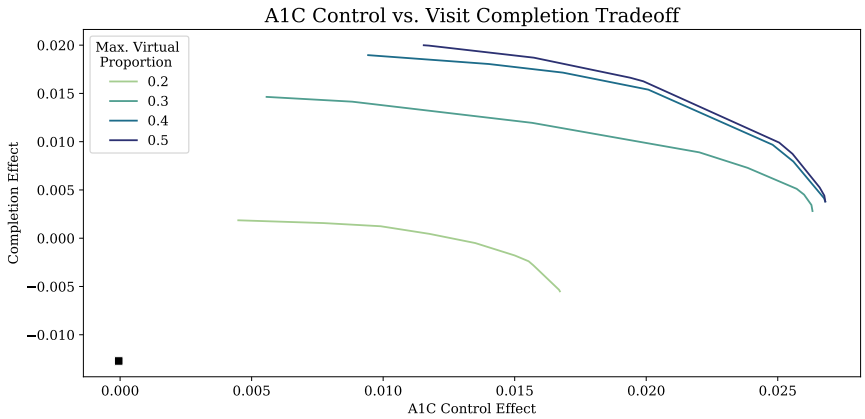


Figure 6-1: Tradeoffs as α and V_{max} vary. The black square represents the baseline effects obtained from the implemented schedules.

the average effect estimates for encounters that are not fixed, meaning that they are candidates for optimizing the visit modality.

Effect of V_{max} and α parameters. Figure 6-1 shows the tradeoffs in the two objective terms, A1C control and visit completion, as V_{max} and α vary. Both objective terms improve as V_{max} increases, as this relaxes the upper bound on virtual visits and thus broadens the feasible region. The average optimized effects significantly outperform the average baseline effects for 0.0 A1C control and -0.01 visit completion across all parameter combinations. This is necessarily true, as it is enforced by constraints 6.2b and 6.2c in the model formulation. For a fixed V_{max}, there is a tradeoff between the two objective terms which favors A1C control as α increases.

We further explore a single week's solution to gain insight into the structure of the optimal schedules as priorities shift between A1C control and visit completion. We fix $V_{max} = 0.5$, which renders the virtual proportion limit non-binding, and compare the optimized schedules for $\alpha \in [0.0, 0.5, 1.0]$. The resultant schedules are displayed in Figure 6-2, and the average effects on A1C control and visit completion are shown in Table 6.4. At $\alpha = 0$, the model switches all flexible appointments to virtual, leaving only the forced office visits to occur in person (overall virtual rate of 44.4%). As α increases, some flexible visits flip to in-person, reflecting that in-person visits offer a benefit when A1C control gets prioritized. However, even at the highest value of $\alpha = 1.0$, there is still a 29.6% virtual visit rate, exceeding the 18.5% baseline rate.

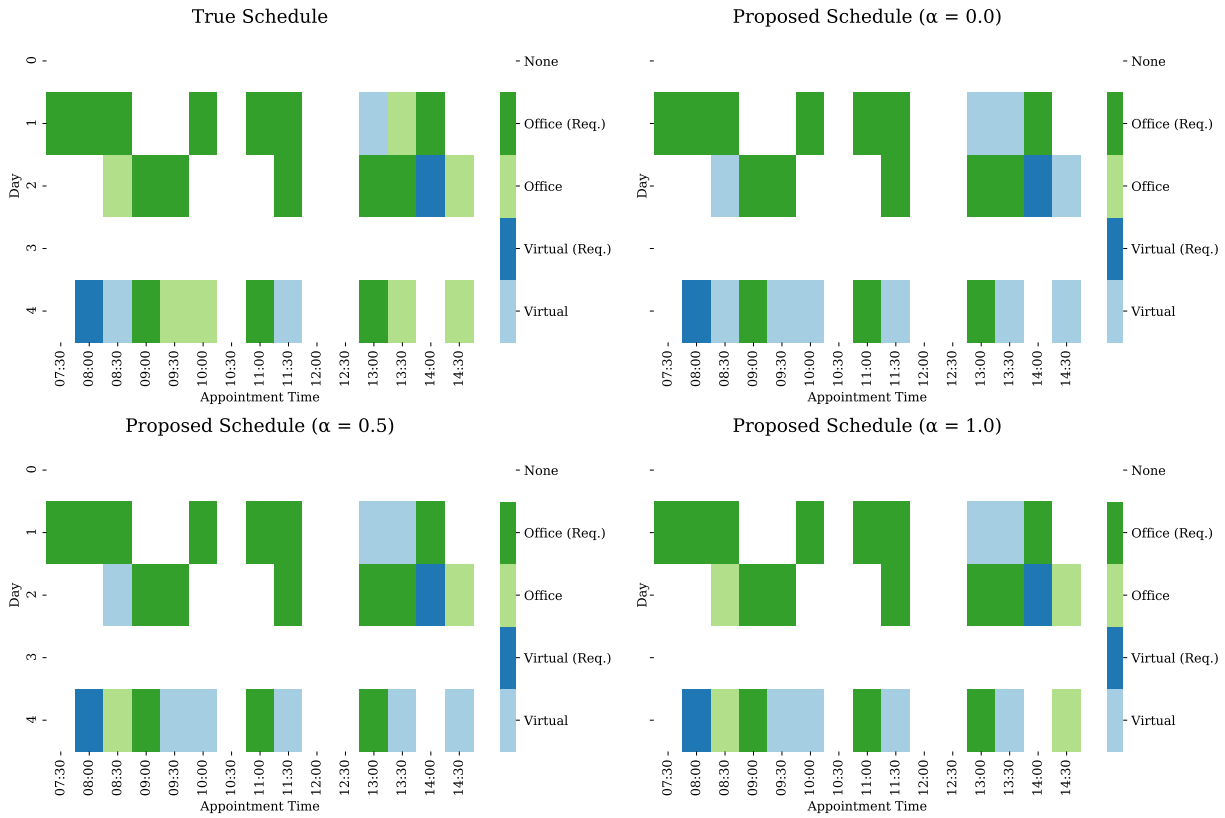


Figure 6-2: Visualization of a single week's schedule.

Comparison to baseline current schedule. Finally, with $V_{max} = 0.5$ and $\alpha = 0.5$ fixed, we compare the performance of our prescription model across all five test weeks against the true schedule. We perform this evaluation using the original trained effect estimation model ("Training Model") as well as the models derived from the test set ("Testing Model"). The results are displayed in Table 6.5. Across the test weeks, the true modality assignment has a slight negative effect in both the training and testing models, while the optimized schedule has a positive effect in all cases. The benefit estimated from the independent testing model is similar to the training model's estimate for visit completion (+0.025, vs. +0.029), and actually higher than the training model's estimate for A1C control (+0.097, vs. +0.020). This suggests that the training model's effect estimates are consistent with the trends seen in the test set, providing a robustness check.

Alpha	Virtual Proportion	A1C Control Effect	Visit Completion Effect
0	0.444	0.009	0.023
0.5	0.370	0.016	0.018
1	0.296	0.028	0.004
Baseline	0.185	0.001	-0.007

Table 6.4: Treatment effect for various optimized schedules.

	A1C Control Effect		Visit Completion Effect	
	Training Model	*Testing Model*	*Training Model*	*Testing Model*
True Schedule	0.000	-0.036	-0.013	-0.012
Optimized Schedule	0.020	0.061	0.016	0.013
Benefit (Absolute)	0.020	0.097	0.029	0.025

Table 6.5: Comparison of average outcome effects across five test weeks. The true schedule indicates the visit modalities implemented in practice, and the optimized schedule reflects the recommended modalities with $V_{max} = 0.5$ and $\alpha = 0.5$.

6.5 Discussion

The optimization results suggest that increasing virtual care utilization for diabetic patients can improve both operational (visit completion probability) and clinical (A1C control) outcomes. For a given candidate schedule, the optimal allocation of visit modalities depends on how these two objectives are weighted. In general, the proportion of virtual visits increases as visit completion is prioritized. However, the recommended virtual visit rate increases above the baseline level even when the objective purely maximizes A1C control. This suggests that incorporating virtual care to some degree is beneficial for both outcomes. While virtual care was heavily utilized at the beginning of the pandemic, it decreased significantly over the course of 2021. The results motivate further study of virtual visits as a long-term care strategy. We focus on diabetic patients being seen in endocrinology practices but note that the framework can be applied to other specialties.

As with `OptiCL` [117], the integration of ML and optimization allows us to both *learn* the relationship between decisions and outcomes and to *optimize* these decisions. The critical difference in this work is the use of causal models rather than standard supervised learning models to estimate the decision-outcome relationship. We obtain outcome estimates using a DR estimator, and further train a CF model to predict treatment effects. Causal models are a natural fit for

learning decision-outcome relationships in optimization settings, as they are tailored to estimating the effect of a certain treatment, rather than focused on predictive accuracy. This aligns with the downstream task, which is ultimately to prescribe decisions. Furthermore, the DR estimator provides safeguards against potential misspecification of either the propensity or outcome model. The descriptive analysis in Table 6.2 reveals differences in treatment propensity across several characteristics, including demographics and clinical status, attesting to the value of propensity-based reweighting.

However, the use of causal models also restricts the richness of the decisions that can be optimized. The causal approach is enabled by the presence of a binary decision ($Z \in \{0,1\}$, representing visit modality). Causal effect estimation compares outcomes between different treatment alternatives, and existing methods generally rely on discrete treatment alternatives (e.g., two drug options) or a single continuous treatment (e.g., drug dose). For example, the CF model loss is based on the treatment effect, namely the difference between treatments, which fundamentally assumes a binary treatment decision. Additionally, the DR estimator that we employ in this work requires direct estimation outcome models and a propensity model. The constraint learning approach is a form direct estimation, where Y is predicted using a joint feature space of features and treatments that naturally handles multi-dimensional decision settings. However, the treatment propensity model requires a discrete set of treatment alternatives and thus cannot be directly applied to an arbitrary multi-dimensional decision setting with combination treatments.

In future work, we look to bridge the gap between causal approaches and multi-dimensional decision settings, allowing us to leverage treatment effect estimation in the more general constraint learning framework. This would allow us to consider a much broader space of decisions for care optimization. At the weekly level, provider, day, and time assignment could be treated as additional decision variables. To make such a model practical, it would require additional data on patient and provider availability outside of the existing appointments. Additionally, a multi-dimensional representation of decisions would allow us to more holistically optimize a patient's care plan, rather than a single visit. For example, a care plan "decision" could be represented by visit frequency of each modality, referrals to auxiliary services (e.g., nutrition, diabetes education), and provider team composition (e.g., primary endocrinologist, medical assistants).

An MIO approach also allows for the flexible incorporation of additional model constraints.

The treatment effect terms that appear in the objective and constraints could also be modified to enforce group-level fairness. Currently, we maximize the *average* effect across all flexible appointments and enforce that this average is no worse than the baseline schedule, but this can result in worsening at the *individual and group level*. A modified objective could maximize the minimum average effect across groups (i.e., the average effect of the group that benefits least from the proposed schedule), or likewise constrain the minimum group-level average effect. Given the connection between visit modality and care access, a group-level perspective would add an important safeguard against disparate impact.

Finally, as with all retrospective studies involving observational data, we are limited by the data collected. We make a standard assumption of no unobserved confounders, which relies on a sufficiently rich feature space. While the existing feature space captures a variety of demographic, clinical, and access metrics, there are other factors that could be relevant as features or additional outcomes to optimize. Going forward, we plan to collect additional clinical features (e.g., other lab results, comorbidities), patient-reported data (e.g., satisfaction, social determinants of health), and provider features (e.g., provider preferences, availability). These will populate a prospective database for ongoing monitoring of virtual health utilization and care effectiveness.

6.6 Conclusion

In this work, we establish a prescriptive framework for virtual care utilization in endocrinology. We consider this problem from a causal inference lens, modeling the visit modality as a treatment lever with impact on both operational and clinical outcomes. We integrate causal ML methods with MIO to bridge the gap between data and decision-making. The resultant model offers a timely policy analysis tool, with potential implications for both local healthcare systems and national guidelines.

Chapter 7

Conclusions

This thesis tackles data-driven decision-making in healthcare. The chapters address both domain-specific clinical questions, such as assessing chemotherapy patients for neutropenia risk, and more general challenges, such as learning constraints in mixed-integer optimization (MIO) formulations. We recognize the enormous opportunity presented by digitized health data, as well as the gaps that hinder this opportunity from being fully realized. In this thesis, we take steps towards closing these gaps.

Chapters 2 and 3 present new approaches to more effective learning from data. Chapter 2 provides a data-to-decisions pipeline that integrates machine learning-driven constraints and objectives into MIO formulations. This framework allows us to address complex problems with multi-dimensional decisions, such as identifying combination chemotherapy regimens for advanced gastric cancer or designing palatable food baskets for humanitarian aid efforts. In Chapter 3, we propose an interpretable clustering algorithm for more informative exploratory data analysis or subgroup identification. The work in both of these chapters have been made accessible through software tools, `OptiCL` and `ICOT`, for broader use by researchers and practitioners.

Chapters 4, 5, and 6 put analytics into practice, tackling three clinically-driven problems. These problems span oncology, COVID-19, and chronic disease care, developing both predictive and prescriptive models to improve risk assessment and inform patient care. As with the methodological contributions, a central goal of these works has been the development of useful tools. The works discussed in this thesis are oriented towards impact. The COVID-19 mortality model has been deployed in practice and the neutropenia prediction model is undergoing a follow-up study with

Hartford HealthCare. The virtual health project is an area of active work with our clinical collaborator, and the model is being used to guide discussions with hospital leaders to inform future care strategies.

Future Directions Healthcare analytics is a rich space, and the questions that this thesis has answered have prompted new questions and future areas to explore.

Our constraint learning framework introduces a new lens for decision making, which has broad applicability to treatment personalization as well as hospital operations and care delivery. It is particularly relevant in the setting of multi-dimensional decisions in complex environments, which describes many healthcare problems. There is a natural tie between constraint learning and causal inference, as decision optimization relies on distilling the effect of such decisions on the outcomes of interest. While causal inference methods address treatment effect estimation, such methods significantly restrict the richness of decision space that can be considered. The integration of these two fields is of great interest as a future direction. The virtual health effort forms a strong basis for this work, given its use of causal models within a larger optimization framework.

There is also an opportunity to delve further into the failure points that hinder adoption of clinical decision support systems. This includes further development of interpretable methods, which become increasingly important–and challenging–in the presence of multi-modal data. Another barrier is the tension between clinical knowledge and machine learning (ML) models: while much can be learned from data, there is also much to learn from practitioners who see patients every day. The balance between domain knowledge and algorithms is typically struck in an ad hoc manner. Predictive and prescriptive tools would benefit from a unified approach to jointly harnessing these two valuable information sources. Finally, perhaps the most significant obstacle to clinical decision support tools is their lack of validation on external populations. The burden of data curation and standardization prevents models from being tested on other sites, and without evidence of generalizability, these models rightfully remain unused. There is need for further work in auditing ML models, particularly with a lens towards disparate impacts on underrepresented populations. This includes characterizing the “trust limits" of a model and quantifying the representativeness of a training population.

Appendix A

Appendix for Chapter 2

A.1 Methodology

Embedding a decision tree

Consider the leaves in Figure 2-2. We can then encode the leaf assignment of an observation $\boldsymbol{x}$ through the following constraints:

$$A_1^\top \boldsymbol{x} - M(1-l_3) \leq b_1, \tag{A.1a}$$

$$A_2^\top \boldsymbol{x} - M(1-l_3) \leq b_2, \tag{A.1b}$$

$$A_1^\top \boldsymbol{x} - M(1-l_4) \leq b_1, \tag{A.1c}$$

$$-A_2^\top \boldsymbol{x} - M(1-l_4) \leq -b_2 - \varepsilon, \tag{A.1d}$$

$$-A_1^\top \boldsymbol{x} - M(1-l_6) \leq -b_1 - \varepsilon, \tag{A.1e}$$

$$A_5^\top \boldsymbol{x} - M(1-l_6) \leq b_5, \tag{A.1f}$$

$$-A_1^\top \boldsymbol{x} - M(1-l_7) \leq -b_1 - \varepsilon, \tag{A.1g}$$

$$-A_5^\top \boldsymbol{x} - M(1-l_7) \leq -b_5 - \varepsilon, \tag{A.1h}$$

$$l_3 + l_4 + l_6 + l_7 = 1, \tag{A.1i}$$

$$y - (p_3 l_3 + p_4 l_4 + p_6 l_6 + p_7 l_7) = 0, \tag{A.1j}$$

where l_3, l_4, l_6, l_7 are binary variables associated with the corresponding leaves.

An observation will be assigned to the leftmost leaf (node 3) if $A_1^\top \boldsymbol{x} \leq b_1$ and $A_2^\top \boldsymbol{x} \leq b_2$. An observation would be assigned to node 4 if $A_1^\top \boldsymbol{x} \leq b_1$ and $A_2^\top \boldsymbol{x} > b_2$, or equivalently, $-A_2^\top \boldsymbol{x} < -b_2$. Furthermore, we can remove the strict inequalities using a sufficiently small ε parameter, so that $-A_2^\top \boldsymbol{x} \leq -b_2 - \varepsilon$. For a given $\boldsymbol{x}$, if $A_1^\top \boldsymbol{x} \leq b_1$, Constraints (A.1e) and (A.1h) will force l_6 and l_7 to zero, respectively. If $A_2^\top \boldsymbol{x} \leq b_2$, constraint (A.1d) will force l_4 to 0. The assignment constraint (A.1i) will then force $l_3 = 1$, assigning the observation to leaf 3 as desired. Finally, constraint (A.1j) sets y to the prediction of the assigned leaf (p_3). We can then constrain the value of y using our desired upper bound of τ (or lower bound, without loss of generality).

More generally, consider a decision tree $\hat{h}(\boldsymbol{x}, \boldsymbol{w})$ with a set of leaf nodes $\mathscr{L}$ each described by a binary variable l_i and a prediction score p_i. Splits take the form $(A_x)^\top \boldsymbol{x} + (A_w)^\top \boldsymbol{w} \leq b$, where A_x gives the coefficients for the optimization variables $\boldsymbol{x}$ and A_w gives the coefficients for the non-optimization (fixed) variables $\boldsymbol{w}$. Let $\mathscr{S}^l$ be the set of nodes that define the splits that observations in leaf i must obey. Without loss of generality, we can write these all as $(\bar{A}_x)_j^\top \boldsymbol{x} + (\bar{A}_w)_j^\top \boldsymbol{w} - M(1 - l_i) \leq \bar{b}_j$, where $\bar{A}$ is A if leaf i follows the left split of j and $-A$ otherwise. Similarly, $\bar{b}$ equals b if the leaf falls to the left split, and $-b - \varepsilon$ otherwise, as established above. This decision tree can then be embedded through the following constraints:

$$(\bar{A}_x)_j^\top \boldsymbol{x} + (\bar{A}_w)_j^\top \boldsymbol{w} - M(1 - l_i) \leq \bar{b}_j,\ i \in \mathscr{L}, j \in \mathscr{S}^l, \tag{A.2a}$$

$$\sum_{i \in \mathscr{L}} l_i = 1, \tag{A.2b}$$

$$y - \sum_{i \in \mathscr{L}} p_i l_i = 0. \tag{A.2c}$$

Here, M can be selected for each split by considering the maximum difference between $(\bar{A}_x)_j^\top \boldsymbol{x} + (\bar{A}_w)_j^\top \boldsymbol{w}$ and b_j. A prescription solution $\boldsymbol{x}$ for a patient with features $\boldsymbol{w}$ must obey the constraints determined by its split path, *i.e.* only the splits that lead to its assigned leaf i. If $l_i = 0$ for some leaf i, the corresponding split constraints need not be considered. If $l_i = 1$, constraint (A.2a) will enforce that the solution obeys all split constraints leading to leaf i. If $l_i = 0$, no constraints related to leaf i should be applied. When $l_i = 0$, constraint (A.2a) will be nonbinding at node j if $M \geq (\bar{A}_x)_j^\top \boldsymbol{x} + (\bar{A}_w)_j^\top \boldsymbol{w} - \bar{b}_j$. Thus we can find the minimum necessary value of M by maximizing these expressions over all possible values of $\boldsymbol{x}$ (for the patient's fixed $\boldsymbol{w}$). For a given patient with

features $\boldsymbol{w}$ for whom we wish to optimize treatment, EM($\boldsymbol{w}$) is the solution of

$$\max_{\boldsymbol{x}} (\bar{A}_x)_j^\top \boldsymbol{x} + (\bar{A}_w)_j^\top \boldsymbol{w} - \bar{b}_j \tag{A.3a}$$

$$\text{s.t. } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{w}) \leq 0, \tag{A.3b}$$

$$\boldsymbol{x} \in \mathscr{X}(\boldsymbol{w}). \tag{A.3c}$$

Note that the non-learned constraints on $\boldsymbol{x}$, namely constraint (A.3b), and the trust region constraint (A.3c) allow us to reduce the search space when determining M.

MIO vs. LO formulation for decision trees

In Section 2.2, we proposed two ways of embedding a decision tree as a constraint. The first uses an LO to represent each feasible leaf node in the tree, while the second directly uses the entire MIO representation of the tree as a constraint. To compare the performance of these two approaches, we learn the palatability constraint using a decision tree (CART) grown to various depths (from a maximum depth of 3 to 20) and solve the optimization model with both approaches. Figure A-1 shows that as the maximum allowable tree depth is increased, the number of LOs to be solved also increases. This is because there are more feasible leaves which need to be represented using LOs. Once the tree has reached the optimal depth (selected via cross-validation), increasing the maximum allowable depth of the tree does not cause the tree to grow any further. At this point, the number of LOs to be solved remains constant. When comparing the solution times (averaged over 10 runs), the right graph in Figure A-1 shows that the MIO approach is relatively consistent in terms of solution time regardless of the tree depth. With the LO approach however, as the depth of the tree grows, the number of LOs to be solved also grows. While the solution time of a single LO is very low, solving multiple LOs sequentially might be heavily time consuming. A way to speed up the process is to solve the LOs in parallel. A tree of depth 3 requires only one LO to be solved, which takes 1.8 seconds in this problem setting. By parallelizing the solution of the LOs, the total solution time can be expected to take only as long as it takes for the slowest LO to be solved.

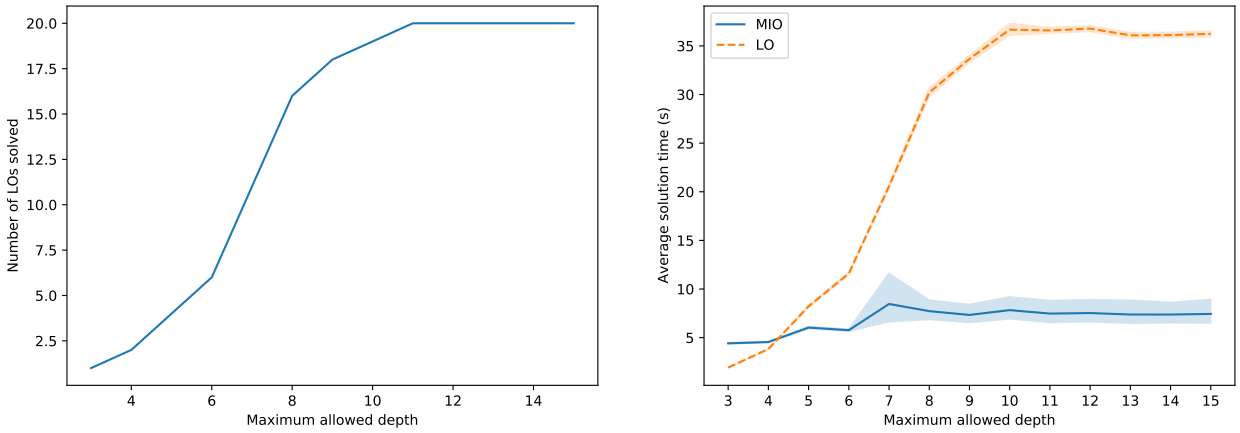


Figure A-1: Comparison of MIO and multiple LO approach to tree representation, as a function of allowable tree depth.

MIO representation of the ReLU activation function

We can represent the ReLU operator, $v = \max\{0, x\}$ the following way:

$$v \geq x, \tag{A.4a}$$

$$v \leq x - M_L(1-z), \tag{A.4b}$$

$$v \leq M_U z, \tag{A.4c}$$

$$v \geq 0, \tag{A.4d}$$

$$z \in \{0,1\}, \tag{A.4e}$$

where $M_L < 0$ is a lower bound on all possible values of x, and $M_U > 0$ is an upper bound. While this embedding relies on a big-M formulation, it can be improved in multiple ways. The model can be tightened by careful selection of M_L and M_U. Furthermore, [8] recently proposed an additional iterative cut generation procedure to improve the strength of the basic big-M formulation.

Embedding a multi-layer perceptron for multi-class classification

In multi-class classification, the outputs are traditionally obtained by applying a *softmax* activation function, $S(\boldsymbol{x})_i = e^{x_i} / \left(\sum_{k=1}^{K} e^{x_k}\right)$, to the final layer. This function ensures that the outputs sum to one and can thus be interpreted as probabilities. In particular, suppose we have a K-class classification problem. Each node in the final layer has an associated weight vector $\boldsymbol{\beta}_i$, which maps the

nodes of layer $L-1$ to the output layer by $\boldsymbol{\beta}_i^\top \boldsymbol{v}^{L-1}$. The softmax function rescales these values, so that class i will be assigned probability

$$v_i^L = \frac{e^{\boldsymbol{\beta}_i^\top \boldsymbol{v}^{L-1}}}{\sum_{k=1}^K e^{\boldsymbol{\beta}_k^\top \boldsymbol{v}^{L-1}}}.$$

We cannot apply the softmax function directly in an MIO framework with linear constraints. Instead, we use an *argmax* function to directly return an indicator of the highest probability class, similar to the approach with SVC and binary classification MLP. In other words, the output $\boldsymbol{y}$ is the identity vector with $y_i = 1$ for the most likely class. Class i has the highest probability if and only if

$$\beta_{i0}^L + \boldsymbol{\beta}_i^{L\top} \boldsymbol{v}^{L-1} \geq \beta_{k0}^L + \boldsymbol{\beta}_k^{L\top} \boldsymbol{v}^{L-1}, \quad k = 1, \ldots, K.$$

We can constrain this with a big-M constraint as follows:

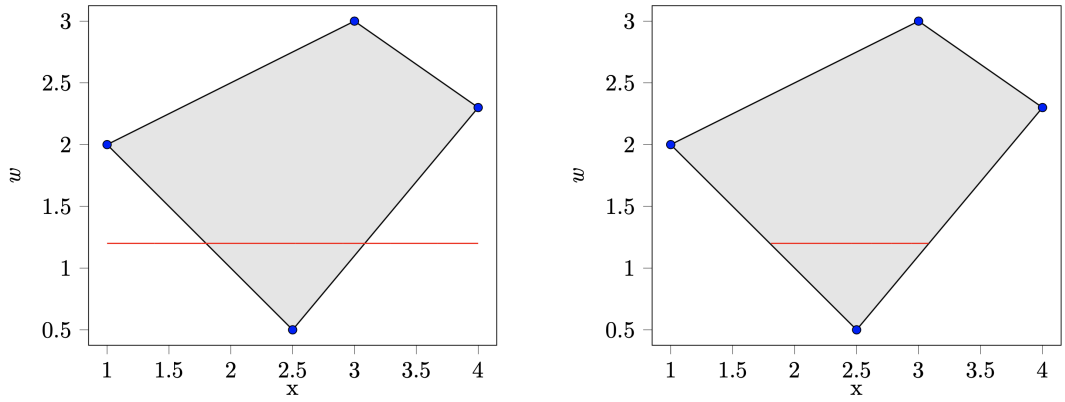
$$\beta_{i0}^L + \boldsymbol{\beta}_i^{L\top} \boldsymbol{v}^{L-1} \geq \beta_{k0}^L + \boldsymbol{\beta}_k^{L\top} \boldsymbol{v}^{L-1} - M(1-y_i), \quad k = 1, \ldots, K, \tag{A.5a}$$

$$\sum_{k=1}^K y_k = 1. \tag{A.5b}$$

Constraint (A.5a) forces $y_i = 0$, if the constraint is not satisfied for some $k \in \{1, \ldots, K\}$. Constraint (A.5b) ensures that $y_i = 1$ for the highest likelihood class. We can then constrain the prediction to fall in our desired class i by enforcing $y_i = 1$.

A.2 Trust region

As we explain in Section 2.2.3, the trust region prevents the predictive models from extrapolating. It is defined as the convex hull of the set $\mathscr{X} = \{(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{w}}_i)\}_{i=1}^N$, with $\bar{\boldsymbol{x}}_i \in \mathbb{R}^n$ observed treatment decisions, and $\bar{\boldsymbol{w}}_i \in \mathbb{R}^p$ contextual information. In Section A.2, we explain the importance of using both $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{w}}$ in the formulation of the convex hull. When the number of samples (N) is too large, the optimization model trust region constraints may become computationally expensive. In this case, we propose a column selection algorithm which is detailed in Section A.2.



(a) Solutions lie within trust region.

(b) Solutions may lie outside the trust region.

Figure A-2: Effect of $\bar{\boldsymbol{w}}$ on the trust region.

Defining the convex hull

We characterize the feasible decision space using the convex hull of our observed data. In general, we recommend defining the feasible region with respect to both $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{w}}$. This ensures that our prescriptions are reasonable with respect to the contextual variables as well. Note that for different values of $\boldsymbol{w}$, the convex hull in the $\boldsymbol{x}$ space may be different. In Figure A-2, the shaded region represents the convex hull of $\mathscr{Z}$ formed by the dataset (blue dots), and the red line represents the set of trusted solutions when $\boldsymbol{w}$ is fixed to a certain value. In Figure A-2a, we see that the set of trusted solutions (red line) lies within CH($\mathscr{Z}$) when we include $\bar{\boldsymbol{w}}$. If we leave out $\bar{\boldsymbol{w}}$ in the definition of the trust region, then we end up with the undesired situation shown in Figure A-2b, where the solution may lie outside of CH($\mathscr{Z}$). We observe that in some cases we must define the convex hull with a subset of variables. This is true in cases where the convex hull constraint leads to excessive data thinning, in which case it may be necessary to define the convex hull on treatment variables only.

Column selection

Let $\mathrm{P}_{\mathscr{J}}$ be a convex and continuously differentiable model consisting of an objective function and constraints that may be known a priori as well as learned from data. Like in Section 2.2.3, we denote the index set of samples by $\mathscr{I}$. As part of the constraints, the trust region is defined on the entire set $\mathscr{Z}$. We start with the matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times(n+p)}$, where each row corresponds to a given data

point in $\mathscr{Z}$. Then, model $\mathrm{P}_{\mathscr{I}}$ is given as

$$\min_{\boldsymbol{\lambda}} f(\boldsymbol{Z}^\top \boldsymbol{\lambda}) \tag{A.6a}$$

$$\text{s.t. } g_j(\boldsymbol{Z}^\top \boldsymbol{\lambda}) \leq 0, \quad j = 1, \ldots, m, \qquad \perp \boldsymbol{\mu}, \tag{A.6b}$$

$$\sum_{i \in \mathscr{I}} \lambda_i = 1, \qquad \perp \rho, \tag{A.6c}$$

$$\lambda_i \geq 0, \quad i \in \mathscr{I}, \qquad \perp \boldsymbol{\upsilon}, \tag{A.6d}$$

where the decision variable $\boldsymbol{x}$ is replaced by $\boldsymbol{Z}^\top \boldsymbol{\lambda}$. Constraints (A.6b) include both *known* and *learned* constraints, while constraints (A.6c) and (A.6d) are used for the trust region. The dual variables associated with with constraints (A.6b), (A.6c), and (A.6d) are $\boldsymbol{\mu} \in \mathbb{R}^m, \rho \in \mathbb{R}$, and $\boldsymbol{\upsilon} \in \mathbb{R}^N$, respectively. Note that for readability, we omit the contextual variables ($\boldsymbol{w}$) without loss of generality.

When we deal with huge datasets, solving $\mathrm{P}_{\mathscr{I}}$ may be computationally expensive. Therefore, we propose an iterative column selection algorithm (Algorithm 2) that can be used to speed up the optimization while still obtaining a global optima.

Algorithm 2 Column Selection

Require: $\mathscr{I}$ ▷ Index set of columns of $\boldsymbol{Z}^\top$
Ensure: $\boldsymbol{\lambda}^*$ ▷ Optimal solution
$\mathscr{I}' \leftarrow \mathscr{I}^0$ ▷ Initial column pool
while TRUE **do**
  $\boldsymbol{\lambda}^*, (\boldsymbol{\mu}^*, \rho^*, \boldsymbol{\upsilon}^*) \leftarrow \mathrm{P}_{\mathscr{I}'}$
  $\bar{\mathscr{I}} \leftarrow$ WOLFEDUAL($\boldsymbol{\lambda}^*$, $(\boldsymbol{\mu}^*, \rho^*, \boldsymbol{\upsilon}^*)$, $\mathscr{I}', \mathscr{I}$) ▷ Column(s) selection
  if $\bar{\mathscr{I}} \neq \emptyset$ **then**
    $\mathscr{I}' \leftarrow \mathscr{I}' \cup \bar{\mathscr{I}}$
  else
    Break
  end if
end while

The algorithm starts by initializing $\mathscr{I}' \subseteq \mathscr{I}$ with an arbitrarily small subset of samples $\mathscr{I}^0$ and iteratively solves the restricted master problem $\mathrm{P}_{\mathscr{I}'}$ and the WOLFEDUAL function. By solving $\mathrm{P}_{\mathscr{I}'}$, we get the primal and dual optimal solutions $\boldsymbol{\lambda}^*$ and $(\boldsymbol{\mu}^*, \rho^*, \boldsymbol{\upsilon}^*)$, respectively. The primal and dual optimal solutions, together with $\mathscr{I}$ and $\mathscr{I}'$, are given as input to WOLFEDUAL which

returns a set of samples $\bar{\mathscr{I}} \subseteq \mathscr{I} \setminus \mathscr{I}'$ with negative reduced cost. If $\bar{\mathscr{I}}$ is not empty it is added to $\mathscr{I}'$ and a new iteration starts, otherwise the algorithm stops, and $\boldsymbol{\lambda}^*$ (with the corresponding $\boldsymbol{x}^*$) is returned as the global optima of $\mathrm{P}_{\mathscr{I}}$. A visual interpretation of Algorithm 2 is shown in Figure 2-4.

In function WOLFEDUAL, samples $\bar{\mathscr{I}}$ are selected using the Karush–Kuhn–Tucker (KKT) stationary condition which corresponds to the equality constraint in the Wolfe dual formulation of $\mathrm{P}_{\mathscr{I}}$ [170]. The KKT stationary condition of $\mathrm{P}_{\mathscr{I}'}$ is

$$\nabla_{\boldsymbol{\lambda}} f(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) + \sum_{i=1}^{m} \mu_i^* \nabla_{\boldsymbol{\lambda}} g_i(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) - \boldsymbol{e}\rho^* - \boldsymbol{\upsilon}^* = \boldsymbol{0}, \tag{A.7}$$

where $\tilde{\boldsymbol{Z}}$ is the matrix constructed with samples in $\mathscr{I}'$, and $\boldsymbol{e}$ is an N'-dimensional vector of ones with $N' = |\mathscr{I}'|$. Equation (A.7) can be rewritten as

$$\tilde{\boldsymbol{Z}} \nabla_{\boldsymbol{x}} f(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) + \sum_{i=1}^{m} \mu_i^* \tilde{\boldsymbol{Z}} \nabla_{\boldsymbol{x}} g_i(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) - \boldsymbol{e}\rho^* - \boldsymbol{\upsilon}^* = \boldsymbol{0}. \tag{A.8}$$

Equation (A.8) is used to evaluate the reduced cost related to each sample $\bar{\boldsymbol{z}} \in \mathscr{Z}$ which is not in matrix $\tilde{\boldsymbol{Z}}$. Consider a new sample $\bar{\boldsymbol{z}}$ in (A.8), with its associated $\lambda_{N'+1}$ set equal to zero. $(\lambda_1^*, \ldots, \lambda_{N'}^*, \lambda_{N'+1})$ is still a feasible solution of the restricted master problem $\mathrm{P}_{\mathscr{I}'}$, since it does not affect the value of $\boldsymbol{x}$. As a consequence, $\boldsymbol{\mu}$ and ρ will not change their value, nor will f and $\boldsymbol{g}$. The only unknown variable is $\upsilon_{N'+1}$, namely the reduced cost of $\bar{\boldsymbol{z}}$. However, we can write it as

$$\begin{pmatrix} \boldsymbol{\upsilon}^* \\ \upsilon_{N'+1} \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{Z}} \\ \bar{\boldsymbol{z}}^\top \end{pmatrix} \nabla_{\boldsymbol{x}} f(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) + \sum_{i=1}^{t} \mu_i^* \begin{pmatrix} \tilde{\boldsymbol{Z}} \\ \bar{\boldsymbol{z}}^\top \end{pmatrix} \nabla_{\boldsymbol{x}} g_i(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) - \boldsymbol{e}\rho^*. \tag{A.9}$$

If $\upsilon_{N'+1}$ is negative it means that we may improve the incumbent solution of $\mathrm{P}_{\mathscr{I}'}$ by including the sample $\bar{\boldsymbol{z}}$ in $\tilde{\boldsymbol{Z}}$.

Lemma A.2.1. *After solving the convex and continuously differentiable problem $\mathrm{P}_{\mathscr{I}'}$, the sample in $\mathscr{I} \setminus \mathscr{I}'$ with the most negative reduced cost is a vertex of the convex hull CH($\mathscr{Z}$).*

Proof. From equation (A.9) we have

$$\upsilon_{N'+1} = \bar{\boldsymbol{z}}^\top \nabla_{\boldsymbol{x}} f(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) + \bar{\boldsymbol{z}}^\top \nabla_{\boldsymbol{x}} \boldsymbol{g}(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) \boldsymbol{\mu}^* - \rho^*. \tag{A.10}$$

The problem of finding $\bar{z}$, such that its reduced cost is the most negative one, can be written as a linear program where equation (A.10) is being minimized, and a solution must lie within CH($\mathscr{Z}$). That is,

$$\begin{aligned} \min_{\boldsymbol{z},\boldsymbol{\lambda}} \; & \boldsymbol{z}^\top \nabla_{\boldsymbol{x}} f(\tilde{\boldsymbol{Z}}^\top \boldsymbol{\lambda}^*) + \boldsymbol{z}^\top \nabla_{\boldsymbol{x}} \boldsymbol{g}(\tilde{\boldsymbol{Z}}) \boldsymbol{\mu}^* - \rho^* \\ \text{s.t. } & \boldsymbol{Z}^\intercal \boldsymbol{\lambda} = \boldsymbol{z}, \\ & \sum_{j \in \mathscr{I}} \lambda_j = 1, \\ & \lambda_j \geq 0, \quad j \in \mathscr{I}, \end{aligned} \tag{A.11}$$

where $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ are the decision variables, and $\boldsymbol{\mu}^*$, $\boldsymbol{\lambda}^*$, ρ^* are fixed parameters. Since the objective function is linear with respect to $\boldsymbol{z}$, the optimal solution of (A.11) will necessarily be a vertex of CH($\mathscr{Z}$). □

To illustrate the benefits of column selection, consider the following convex optimization problem that we shall refer to as P_{exp}:

$$\min_{\boldsymbol{x}} \; \boldsymbol{c}^\top \boldsymbol{x} \tag{A.12a}$$

$$\text{s.t. } log(\sum_{i=1}^{n} e^{x_i}) \leq t, \tag{A.12b}$$

$$\boldsymbol{A}\boldsymbol{x} \leq b, \tag{A.12c}$$

$$\sum_{i=1}^{N} \lambda_i \bar{\boldsymbol{z}}_i = \boldsymbol{x}, \tag{A.12d}$$

$$\sum_{j=1}^{N} \lambda_j = 1, \tag{A.12e}$$

$$\lambda_j \geq 0, \quad j = 1 \ldots N. \tag{A.12f}$$

Without a loss of generality, we assume that the constraint (A.12b) is known a priori, and constraints (A.12c) are the linear embeddings of learned constraints with $\boldsymbol{A} \in \mathbb{R}^{k \times n}$ and $\boldsymbol{b} \in \mathbb{R}^k$. Constraints (A.12d-A.12f) define the trust region based on N datapoints. Figure A-3 shows the computation time required to solve P_{exp} with different values of n, k, and N. The "No Column Selection" approach consists of solving P_{exp} using the entire dataset. The "Column Selection" approach makes use of Algorithm 2 to solve the problem, starting with $|\mathscr{I}^0| = 100$, and selecting

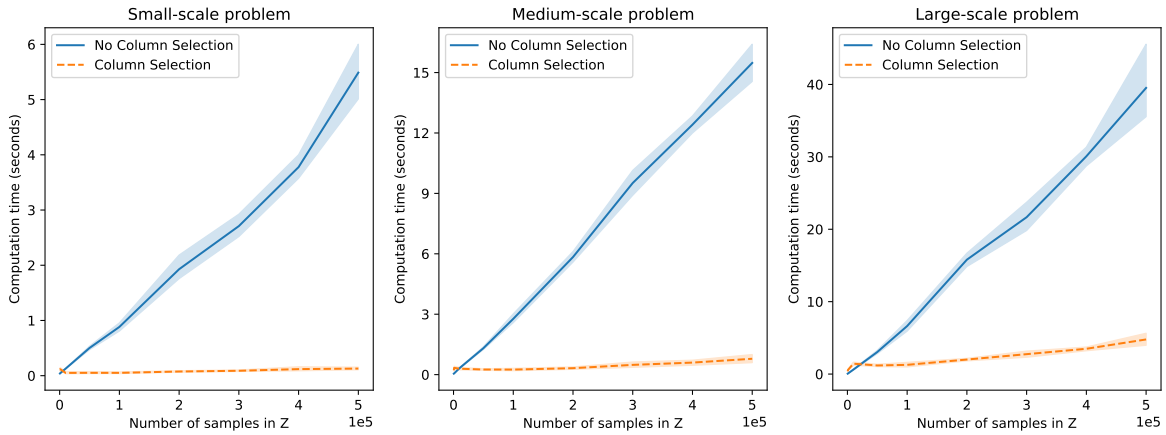


Figure A-3: Effect of column selection on computation time. Solution times are reported for three different sizes of problem P_{exp}. Small-scale: $n = 5,\ k = 10$. Medium-scale: $n = 10,\ k = 50$. Large-scale: $n = 20,\ k = 100$. The number of samples goes from 500 to 5×10^5. In each iteration, the sample with most negative reduced cost is selected. The same problem is solved using [122] with conic reformulation for 10 different instances where c, A, and b are randomly generated.

only one sample at each iteration, *i.e.*, the one with the most negative reduced cost. It can be seen that in all cases, the use of column selection results in significantly improved computation times. This allows us to more quickly define the trust region for problems with large amounts of data.

A.3 WFP case study

Table A.1 and Table A.2 show the nutritional value of each food and our assumed nutrient requirements, respectively. The values adopted are based on the World Health Organization (WHO) guidelines [160].

Effect of ensemble violation limit

Figure A-4 reports the effect of the ensemble violation limit (α) on the objective (Total Cost) and constraint (Palatability) in the WFP case study. As expected, we see a tradeoff between the total cost of the recommended WFP food baskets and the achieved palatability. Higher violation limits (larger α) obtain lower costs at the expense of lower palatability. Lower α values result in more conservative solutions with higher cost but better palatability. This parametrization of individual model violation tolerance allows us to directly quantify this tradeoff and can provide a useful tool

Food	Eng(kcal)	Prot(g)	Fat(g)	Cal(mg)	Iron(mg)	VitA(ug)	ThB1(mg)	RibB2(mg)	NicB3(mg)	Fol(ug)	VitC(mg)	Iod(ug)
Beans	335	20	1.2	143	8.2	0	0.5	0.22	2.1	180	0	0
Bulgur	350	11	1.5	23	7.8	0	0.3	0.1	5.5	38	0	0
Cheese	355	22.5	28	630	0.2	120	0.03	0.45	0.2	0	0	0
Fish	305	22	24	330	2.7	0	0.4	0.3	6.5	16	0	0
Meat	220	21	15	14	4.1	0	0.2	0.23	3.2	2	0	0
Corn-soya blend	380	18	6	513	18.5	500	0.65	0.5	6.8	0	40	0
Dates	245	2	0.5	32	1.2	0	0.09	0.1	2.2	13	0	0
Dried skim milk	360	36	1	1257	1	1,500	0.42	1.55	1	50	0	0
Milk	360	36	1	912	0.5	280	0.28	1.21	0.6	37	0	0
Salt	0	0	0	0	0	0	0	0	0	0	0	1000000
Lentils	340	20	0.6	51	9	0	0.5	0.25	2.6	0	0	0
Maize	350	10	4	13	4.9	0	0.32	0.12	1.7	0	0	0
Maize meal	360	9	3.5	10	2.5	0	0.3	0.1	1.8	0	0	0
Chickpeas	335	22	1.4	130	5.2	0	0.6	0.19	3	100	0	0
Rice	360	7	0.5	7	1.2	0	0.2	0.08	2.6	11	0	0
Sorghum/millet	335	11	3	26	4.5	0	0.34	0.15	3.3	0	0	0
Soya-fortified bulgur wheat	350	17	1.5	54	4.7	0	0.25	0.13	4.2	74	0	0
Soya-fortified maize meal	390	13	1.5	178	4.8	228	0.7	0.3	3.1	0	0	0
Soya-fortified sorghum grits	360	360	1	40	2	0	0.2	0.1	1.7	50	0	0
Soya-fortified wheat flour	360	16	1.3	211	4.8	265	0.66	0.36	4.6	0	0	0
Sugar	400	0	0	0	0	0	0	0	0	0	0	0
Oil	885	0	100	0	0	0	0	0	0	0	0	0
Wheat	330	12.3	1.5	36	4	0	0.3	0.07	5	51	0	0
Wheat flour	350	11.5	1.5	29	3.7	0	0.28	0.14	4.5	0	0	0
Wheat-soya blend	370	20	6	750	20.8	498	1.5	0.6	9.1	0	40	0

Table A.1: Nutritional contents per gram for different foods.
Eng = Energy, Prot = Protein, Cal = Calcium, VitA = Vitamin A, ThB1 = ThiamineB1, RibB2 = RiboflavinB2, NicB3 = NicacinB3, Fol = Folate, VitC = Vitamin C, Iod = Iodine

Type	Eng(kcal)	Prot(g)	Fat(g)	Cal(mg)	Iron(mg)	VitA(ug)	ThB1(mg)	RibB2(mg)	NicB3(mg)	Fol(ug)	VitC(mg)	Iod(ug)
Avg person day	2100	52.5	89.25	1100	22	500	0.9	1.4	12	160	0	150

Table A.2: Nutrient requirements used in optimization model.
Eng = Energy, Prot = Protein, Cal = Calcium, VitA = Vitamin A, ThB1 = ThiamineB1, RibB2 = RiboflavinB2, NicB3 = NicacinB3, Fol = Folate, VitC = Vitamin C, Iod = Iodine

in assessing solution alternatives.

A.4 Chemotherapy regimen design

Data Processing

The data for this case study includes three components, study cohort characteristics ($\boldsymbol{w}$), treatment variables ($\boldsymbol{x}$), and outcomes ($\boldsymbol{y}$). The raw data was obtained from Bertsimas et al. [28], in which the authors manually curated data from 495 clinical trial arms for advanced gastric cancer. Our feature space was processed as follows:

Cohort Characteristics. We included several cohort characteristics to adjust for the study context: fraction of male patients, median age, primary site breakdown (Stomach vs. GEJ), fraction of patients receiving prior palliative chemotherapy, and mean ECOG score. We also included vari-

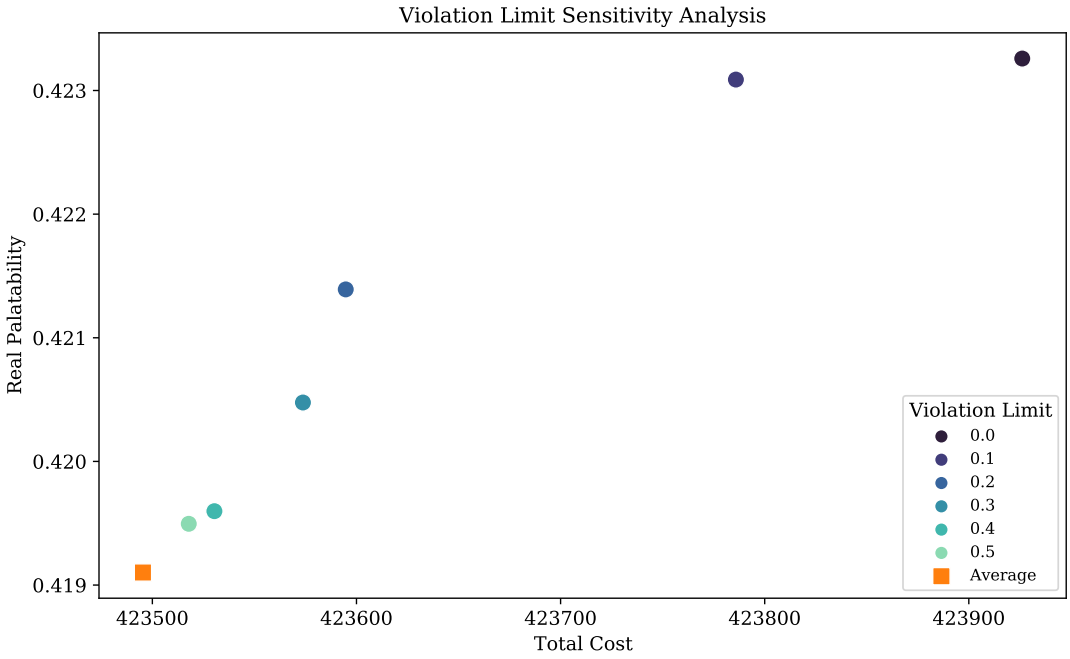


Figure A-4: Effect of violation limit on objective (total cost) and constraint (palatability). The average is reported over 100 problem instances.

ables for the study context: the study year, country, and number of patients. Missing data was imputed using multiple imputation based on the other contextual variables; 20% of observations had one missing feature and 6% had multiple missing features.

Treatment Variables. Chemotherapy regimens involve multiple drugs being delivered at potentially varied frequencies over the course of a chemotherapy cycle. As a result, multiple dimensions of the dosage must be encoded to reflect the treatment strategy. As in Bertsimas et al. [28], we include three variables to represent each drug: an indicator (1 if the drug is used in the regimen), instantaneous dose, and average dose.

Outcomes. We use Overall Survival (OS) as our survival metric, as reported in the clinical trials. Any observations with unreported OS are excluded. We consider several “dose-limiting toxicities" (DLTs): Grade 3/4 constitutional, gastrointestinal, infection, and neurological toxicities, as well as Grade 4 blood toxicities. The toxicities reported in the original clinical trials are aggregated according to the CTCAE toxicity classes [41]. We also include a variable for the occurrence of any

of the four individual toxicities (t_i for each toxicity $i \in T$, called DLT proportion; we treat these toxicity groups as independent and thus define the DLT proportion as

$$DLT = 1 - \prod_{i \in T} (1 - t_i).$$

We define Grade 4 blood toxicity as the maximum of five individual blood toxicities (related to neutrophils, leukocytes, lymphocytes, thrombocytes, anemia). Observations missing all of these toxicities were excluded; entries with partial missingness were imputed using multiple imputation based on other blood toxicity columns. Similarly, observations with no reported Grade 3/4 toxicities were excluded; those with partial missingness were imputed using multiple imputation based on the other toxicity columns. This exclusion criteria resulted in a final set of 461 (of 495) treatment arms.

We split the data into training/testing sets temporally. The training set consists of all clinical trials through 2008, and the testing set consists of all 2009-2012 trials. We exclude trials from the testing set if they use new drugs not seen in the training data (since we cannot evaluate these given treatments). We also identify sparse treatments (defined as being only seen once in the training set) and remove all observations that include these treatments. The final training set consists of 320 observations, and the final testing set consists of 96 observations.

Predictive Models

Table A.3 shows the out-of-sample performance of all considered methods in the model selection pipeline. We note that model choice is based on the 5-fold validation performance, so it does not necessarily correspond to the highest test set performance.

Prescription Evaluation

Table A.4 shows the performance of the models that comprise the ground truth ensemble used in the evaluation framework. These models trained on the full data. We see that the ensemble models, particularly RF and GBM, have the highest performance. These models are trained on more data and include more complex parameter options (*e.g.*, deeper trees, larger forests) since they are not

Outcome	Linear	SVM	CART	ORT-H	RF	GBM
Any DLT	0.268	-0.094	-0.016	0.000	0.152	0.202
Blood	0.196	-1.102	0.012	0.026	0.153	0.105
Constitutional	0.106	0.144	0.157	-0.179	0.194	0.136
Infection	0.082	-0.511	-0.222	0.000	0.070	0.035
Gastrointestinal	0.141	-0.196	-0.023	-0.067	0.066	0.083
Overall Survival	0.448	0.385	0.474	0.505	0.496	0.450

Table A.3: Comparison of out-of-sample R^2 all considered models for learned outcomes in chemotherapy regimen selection problem.

required to be embedded in the MIO and are rather used directly to generate predictions. For this reason, the GT ensemble could also be generalized to consider even broader method classes that are not directly MIO-representable, such as neural networks with alternative activation functions, providing an additional degree of robustness.

outcome	Linear	SVM	CART	RF	GBM	XGB
Any DLT	0.301	0.330	0.250	0.573	0.670	0.323
Blood	0.287	0.351	0.211	0.701	0.813	0.446
Constitutional	0.139	0.224	0.246	0.602	0.682	0.285
Infection	0.217	0.303	0.139	0.514	0.588	0.247
Gastrointestinal	0.201	0.328	0.238	0.563	0.733	0.475
Overall Survival	0.528	0.469	0.421	0.815	0.827	0.756

Table A.4: Performance (R^2) of individual models in ground truth ensemble for model evaluation.

Appendix B

Appendix for Chapter 4

B.1 Data processing

The Hartford Hospital Institutional Review Board (Assurance number: FWA000000601) approved the study and certified that it met the criteria for a waiver of the requirement to obtain informed consent. Algorithm training and testing were completed on servers at MIT. For these purposes, HHC served as the honest broker, de-identifying the data prior to transmission through a recoded patient identifier that was only accessible by HHC. Data were transferred to MIT through a secure file transfer protocol and remained stored on a secure virtual machine for the duration of the project.

B.1.1 Encounter inclusion criteria

Our eligible patient population was comprised of people with chemotherapy infusion encounters that were recorded in Epic between May 2016 and mid-October 2019. Clinical data were not available from prior to HHC's transition to Epic in May 2016. As a consequence, a patient's historical record begins with the first Epic encounter. A chemotherapy encounter is defined as:

- Encounter Type: "Infusion"
- Appointment Status: "Completed"
- CPT Charge: charge description contains: "CHEMOTHERAPY - IV", "CHEMOTHERAPY - INJECTED", or "IV THERAPY"

- Beacon Regimen: Patient on an active Beacon chemotherapy regimen at the time of encounter.

- Meds Administered: Patient has at least one of our identified chemotherapy drugs administered in the encounter.

After identifying the set of all chemotherapy encounters, we further restrict to the start date of each chemotherapy cycle. The day 1 encounter of each cycle is included as an observation, and subsequent visits within the cycle are not included. As a result, a patient can appear multiple times in the data. We exclude patients with Leukemia, as identified through the cancer site listed in the patient's Beacon regimen, given that it is a known elevated risk group. With this filtering criteria, we have 2,806 unique patients with 17,513 chemotherapy infusion encounters.

B.1.2 Chemotherapy drugs

Our goal was to assess neutropenia risk in the weeks following a chemotherapy encounter. In order to identify these encounters, we needed a list of all relevant chemotherapy drugs. Our database included drug class information, but many drugs were missing class mapping information. We instead compiled the drugs from four sources:

- Beacon protocols: The HHC Care Connect team has assembled descriptions for all Beacon protocols, including the names of the individual drugs administered in the protocol. We parsed the free text descriptions of these protocols to pull any potential chemotherapy drugs.

- Wikipedia: List of antineoplastic agents.

- Manual curation: Literature review of individual regimens that appear in our data and their relevant drugs.

- Antineoplastic drugs in medications table: All drugs matching therapeutic class (theraClass) of "Antineoplastic Agents."

By using multiple sources, we were able to cross-check the drug lists and decrease the reliance on any single source. We manually investigated all drugs that appeared in only 1 of 4 sources and

removed any that were erroneously included. In the end, our drug set contained 187 candidate drugs.

B.1.3 Clinical features

All clinical features were obtained directly from structured fields in the EMR. Although Hartford HealthCare Cancer Registry contains curated patient data collected for reporting to the CDC National Program of Cancer Registries, NCI SEER, and Commission on Cancer NCDB registries, the data abstraction into these registries is retrospective with a greater than six-month delay and not suitable for use for point of care calculations.

Demographics We include static patient features (race, ethnicity, sex), as well as the patient's age (calculated based on the encounter date).

Cancer site Cancer site is identified using the cancer regimen table. Each active regimen has an associated with a diagnosis identifier and name. These identifiers map to ICD10 codes which can then be aggregated into cancer types using the CCS mapping. We modified the mapping given by CCS slightly based on clinical chart review, merging together certain rare cancers and drawing distinctions between clinically distinct diseases. Cancer site has 26 distinct values in our dataset. To make it amenable to ML algorithms, it is then "one-hot" encoded, meaning that it is converted from a single column with many potential values to many binary columns indicating which value the feature takes.

Labs/Vitals Common lab test results and vitals readings are included in the model to incorporate a time-varying dimension to patient status. Our lab and vitals features were curated with guidance from the HHC teams. Clinical experts helped us manually group together clinically identical fields (e.g. combining multiple field names that all indicate Glucose readings) to further complete our feature space. For example, a single clinical concept such as temperature might be captured as different entities based on hospital department and equipment. When constructing a dataset for large-scale analysis, these various entities must be harmonized to reflect the clinically meaningful field.

For each vital/lab, we pull in the patient's most recent reading for that feature ("most recent": recorded on or before the first chemo drug is administered for the appointment). We apply different lookback periods based on the type of reading and its medical window of relevance:

- Labs - Blood Count: 2 days
- Labs - Other Chemistry: 7 days
- Vitals: 30 days

After pulling all relevant labs and vitals by encounter, we eliminated features that had more than 50% missingness. Our final feature space includes 12 lab features and 12 vitals features. We also compute the relative change in values for all labs and vitals between encounters. These derived variables are calculated after data imputation due to the potential missingness of individual labs/vitals values in certain encounters. We report the percentage increase or decrease of the feature to incorporate a notion of trend into the feature space.

Chemotherapy regimen information We want to capture the individual drugs administered since this gives us the most unfiltered look at the treatment setting and captures potential modification or deviance from a prescribed plan. Drug information is pulled directly from the Medications Administered table, restricting to medication entries with an action of either "Given" or "New Bag." We encode the drugs as separate columns (e.g. Doxorubicin = "Yes", "No") since a patient may have multiple drugs administered. We also create a single column for the name of the patient's drug combination (e.g. "Cyclophosphamide-Doxorubicin"). The inclusion of regimens in addition to individual drug indicators was recommended by pharmaceutical collaborators, since often individual drugs have low toxicity but can be risky when administered in certain combinations. Regimens (in addition to individual drugs) also appear in risk models throughout the literature.

To limit the dimensionality to drugs and regimens with meaningful rates of occurrence, we only include drugs and regimens that occur in at least 100 encounters across at least 25 patients. We confirmed that no patient has multiple active regimens. As with cancer site, drug combination is a categorical feature and thus is one-hot encoded in the final feature space.

Other treatment information We have added an indicator of whether a patient receives G-CSF injections within 3 days following her chemotherapy appointment. The time window was selected to reflect the clinical setting based on insight from the pharmacy team. Generally, a G-CSF injection is given in association with chemotherapy, within 3 days of the infusion. We have separated the indicator into two types of G-CSF drugs: filgrastim and pegfilgrastim. Pegfilgrastim is longer acting (longer half-life) and thus has different prescription patterns and potentially different impact on FN risk.

We have also added a radiation treatment history feature, which indicates whether a patient is receiving radiation therapy concurrently with chemotherapy. This is queried from the diagnosis and procedure charge table, identifying all encounters with a CPT charge including "RADIATION THERAPY" within three days of the infusion encounter. This [-3,3] day window captures the days in which radiation is generally administered when given concurrently with chemotherapy. We had previously considered the inclusion of a radiation history indicator but found this is unreliable. Therapy history is obtained from procedure charges; since this table is only populated from the Epic launch onwards, it only captures patients with a history of radiation from roughly mid-2017 onwards.

Other medical conditions We have also added indicators of whether a patient has other ongoing medical conditions. We aggregate these conditions using the CCS Classification System [2]. A patient's active problems are found using active problems on the problem list. We consider a problem active if the noted date is prior to the encounter, and the resolved date is either after the encounter or undefined. Most comorbidities are grouped to CCS Level 1, although circulatory diseases were kept at a higher granularity (CCS Level 2) due to their higher prevalence and relevance in this problem setting.

B.1.4 Incorporating temporal effects

For this study, we defined an observation as a single chemotherapy encounter beginning a treatment cycle. Each chemotherapy encounter for each patient became a unique entry in the dataset, and thus a patient can appear as multiple observations in the dataset. This enables the desired prediction task, namely the prediction of neutropenia within four weeks of any chemotherapy cycle initiation,

not only the first. It also increases the sample size.

In order to capture the patient's changing health condition across visits, temporal features are encoded for vital and lab values. Both the raw value and the relative change since the beginning of the last cycle appear as features. For example, a patient with a BMI of 23.0 at their first cycle and 23.5 at her next cycle would have a +2.2% change in BMI encoded in the second cycle. This allows for the identification of important trends; weight loss, rather than actual weight, may be significant as a predictor. We also record cumulative treatment history, such as the total number of cycles that a patient has had through the current encounter.

B.2 Machine learning models

B.2.1 Model selection

A model selection procedure is also used to determine the feature space for each method. For each method, we train models using four variants of the clinical features and three alternate encodings of the temporal features. The clinical feature space variations intend to see the effect of excluding highly granular categorical features, either drug combination or cancer site, which could potentially lead to overfitting. The features indicating value changes over time are considered either as continuous (raw percentage change) or discretized (indicator of loss or increase beyond a threshold) at 10% or 25%. Within each algorithm, the variant that yields the best validation AUC is selected as the final model.

Figure B-1 shows the Receiver Operating Characteristic (ROC) Curve and precision-recall curve across all considered methods. The baseline ROC and precision-recall curves are denoted by the dotted lines.

B.2.2 Model interpretation

Given the lack of natural interpretability in ensemble methods, we interpret the Random Forests (RF) model using SHapley Additive eXplanations (SHAP), an algorithm that estimates the risk impact of individual features through a game theoretic framework [111, 112]. The SHAP feature importance plot for the RF model is shown in Figure B-2. A variable's importance is measured

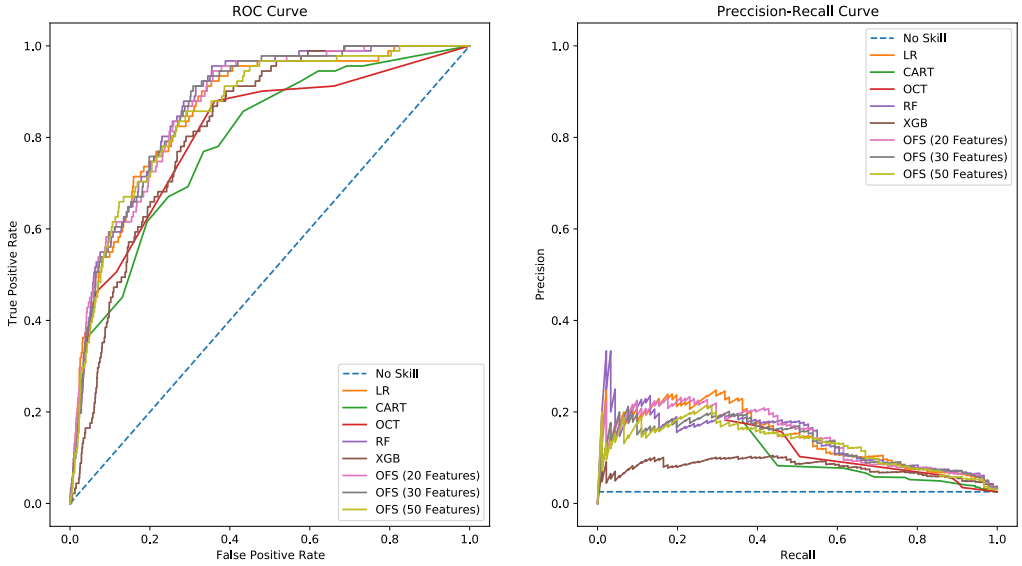


Figure B-1: Receiver Operating Characteristic (ROC) Curve and Precision-Recall Curve evaluated on the test set.

as its mean absolute SHAP value, which quantifies the magnitude (but not directionality) of the feature's impact on resultant risk scores. For further investigation, SHAP dependence plots could be used to visualize the directionality of these relationships.

The 20 most important features are displayed with bars proportional to their importance, starting with Doxorubicin as the most significant predictor. The significant variables in this model are consistent with the main OFS_{20} model. Similar treatment factors appear important in both the OFS_{20} and RF models: the number of drugs administered and cumulative treatment history both appear, although RF also includes an indicator of curative treatment intent. Of the important drugs in the RF model, doxorubicin, cyclophosphamide, etoposide, and carboplatin overlap with the OFS model. The RF model also identifies the combination of doxorubicin and cyclophosphamide as an important predictor, although the OFS model did not identify the interaction beyond their individual risk contributions as significant.

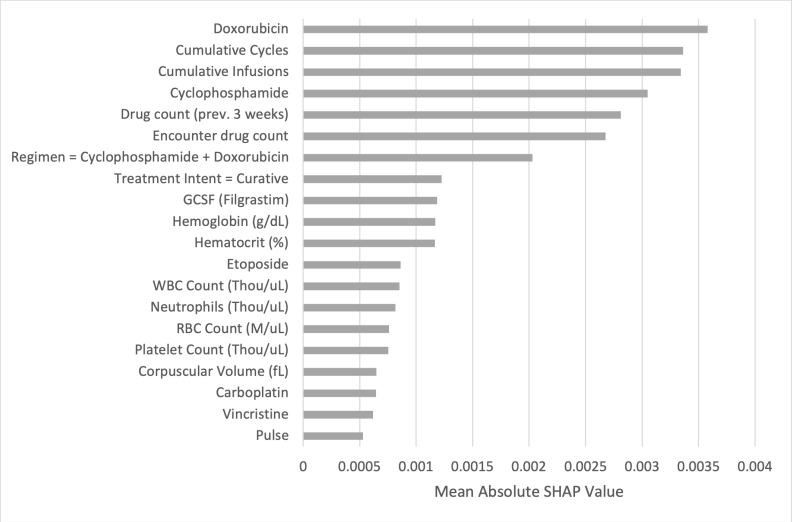


Figure B-2: SHAP feature importance plot for RF model.

B.2.3 Temporal split results

While the main model is trained and evaluated using a patient-wise split, where patients are assigned to either the training or testing set, we also consider a temporal split of the data. A patient-wise split provides insight into how well the model might perform at another institution, where the model is applied to an entirely new set of patients. The temporal split provides insight into how the model might perform prospectively at HHC, when trained on historical data and then applied to patients going forward. In a temporal split, patients can appear in both the training and testing sets since their encounters may span both time windows. This analysis provides an alternative performance evaluation that allows us to assess the robustness of the ML methods.

For our temporal split, we train the models on all encounters prior to 2019, and test the models on all encounters from January 2019 onwards. We follow the same model training, tuning, and selection procedures outlined for the patient-wise split. The 2019 test set AUC and average precision results are reported in Table B.1 across all ML methods, along with bootstrapped 95% confidence intervals. The models have similar out-of-sample performance compared to the patient-wise split (Table 4.3), with the best performing methods demonstrating AUCs of 0.857 (RF) and average precision of 0.173 (XGB). The OFS_{20} model has a slight loss in out-of-sample AUC (-0.03) but offers nearly identical average precision (-0.003). These results suggest that the model performance is robust to the split type (either patient-wise or temporal) and indicate strong generalizability to unseen

data.

Model	AUC	Avg. Precision
OFS_{20} (20 Features)	0.837 (0.818-0.857)	0.145 (0.108-0.201)
OFS_{30} (30 Features)	0.821 (0.801-0.846)	0.153 (0.121-0.207)
OFS_{50} (50 Features)	0.822 (0.799-0.852)	0.126 (0.104-0.178)
LR	0.842 (0.819-0.869)	0.139 (0.108-0.191)
OCT	0.815 (0.782-0.849)	0.094 (0.076-0.120)
CART	0.788 (0.755-0.812)	0.085 (0.066-0.110)
RF	0.857 (0.833-0.882)	0.150 (0.119-0.190)
XGB	0.851 (0.830-0.870)	0.173 (0.134-0.227)
No Skill	0.5	0.023

Table B.1: AUC and average precision (with 95% confidence intervals) reported on the test set, using a temporal split.

Appendix C

Appendix for Chapter 5

C.1 Population characteristics

The clinical features of each derivation and validation cohort are described in Tables C.1 and C.2. We observe differing rates of comorbidities in the populations: our identification of comorbidities was limited by how they were captured in an admission's diagnosis list. Chronic conditions that did not appear in the diagnosis list are considered to not be present in the patient, which could lead to under-reporting of comorbidities. While this is a limitation, these features are not significant in the model on their own and thus do not greatly affect the model predictions. As we expand our models to incorporate richer medical history and treatment information, we will revisit this topic.

C.2 Method details

C.2.1 Missing data imputation

Missing values were encountered in the majority of the included risk factors since the electronic health records of many patients were not complete. Employing imputation techniques instead of complete case analysis allows the inclusion of a wider set of features which otherwise would have been omitted by the model. The k-Nearest Neighbors algorithm [158] is a machine learning technique that can be applied to both supervised and unsupervised learning problems. In the missing data imputation setting, given a missing value for a patient, the algorithm searches for k

	Cremona (N = 1441)		HM Hospitals (N = 1390)		Hartford Affiliates (N = 231)	
	Median (IQR)	Missing %	Median (IQR)	Missing %	Median (IQR)	Missing %
Age	70.0 (58.0-80.0)	1.60%	67.0 (56.0-78.0)	1.60%	68.0 (55.5-79.0)	1.70%
Female *	558.0 (38.7%)	0.00%	537.0 (38.6%)	0.00%	112.0 (48.5%)	0.00%
Heart Rate (bpm)	89.0 (79.1-100.0)	11.00%	90.0 (80.0-102.0)	8.80%	99.0 (88.0-110.0)	4.80%
Oxygen Saturation (%)	93.9 (90.2-96.0)	36.80%	94.0 (92.0-96.0)	10.60%	93.0 (89.0-95.0)	4.80%
Temperature (°C)	37.2 (36.6-37.9)	3.30%	36.6 (36.2-37.2)	9.50%	37.8 (37.0-38.6)	1.30%
ALT (U/L)	26.0 (17.0-43.0)	4.90%	27.8 (17.2-44.7)	19.60%	25.0 (16.0-41.0)	15.20%
AST (U/L)	37.0 (26.0-56.0)	10.10%	34.7 (25.0-52.9)	18.70%	34.0 (24.0-51.8)	17.70%
Blood Glucose (mg/dL)	119.0 (106.0-144.0)	6.20%	116.0 (104.0-137.5)	10.60%	122.0 (102.0-159.5)	5.20%
BUN (mg/dL)	18.0 (14.0-29.0)	7.80%	15.5 (12.0-22.3)	11.10%	19.0 (12.0-31.0)	6.10%
CRP (mg/L)	76.3 (28.5-158.6)	3.70%	70.9 (29.0-132.5)	6.80%	77.3 (30.4-124.0)	46.80%
Creatinine (mg/dL)	1.0 (0.8-1.3)	4.10%	0.9 (0.7-1.1)	5.70%	1.0 (0.8-1.4)	6.50%
Hemoglobin (U/g)	13.5 (12.4-14.7)	22.30%	14.2 (13.1-15.2)	1.90%	12.6 (11.2-13.9)	3.90%
MCV (μm^3)	87.3 (84.6-90.5)	23.60%	88.2 (85.5-91.4)	1.30%	90.0 (86.0-94.0)	6.50%
Platelets ($10^3/\mu L$)	198.0 (154.0-261.5)	22.90%	204.5 (159.0-259.2)	2.40%	204.0 (162.5-250.0)	6.90%
Potassium (mEq/L)	3.9 (3.6-4.3)	9.60%	4.2 (3.9-4.6)	6.50%	4.0 (3.7-4.4)	6.50%
Prothrombin Time (INR)	1.0 (1.0-1.1)	22.30%	1.2 (1.1-1.3)	29.60%	1.1 (1.1-1.4)	67.50%
Sodium (mEq/L)	138.0 (136.0-140.0)	4.20%	136.6 (134.6-139.0)	8.30%	136.0 (134.0-140.0)	4.30%
WBC (/μL)	6900 (5300-9400)	22.80%	6600 (5100-8900)	2.90%	6500 (4800-8700)	3.50%
Cardiac dysrhythmias *	60.0 (4.2%)	0.00%	140.0 (10.1%)	0.00%	1.0 (0.4%)	0.00%
Chronic kidney disease *	16.0 (1.1%)	0.00%	49.0 (3.5%)	0.00%	7.0 (3.0%)	0.00%
Heart disease *	48.0 (3.3%)	0.00%	77.0 (5.5%)	0.00%	0.0 (0.0%)	0.00%
Diabetes *	138.0 (9.6%)	0.00%	207.0 (14.9%)	0.00%	39.0 (16.9%)	0.00%
Mortality *	472.0 (32.8%)	0.00%	239.0 (17.2%)	0.00%	49.0 (21.2%)	0.00%

Table C.1: Descriptive summary of derivation population broken down by study site.
* Count (proportion) is reported for binary variables.

observations in the population that are nearest in feature space, where $k = 5$ in our analysis. The observation is then imputed to the average of the values of its neighbors belong. Though the k-NN algorithm is a simple technique, it often has powerful empirical performance. Its simplicity is also an advantage in terms of interpretability – one can assess the imputed value of a certain point by looking at its neighbors and in which features they are most similar. The training set was imputed independently of the testing set to avoid any bias in the resulting data.

C.2.2 The XGBoost algorithm

The XGBoost algorithm is one of the most popular ensemble methods for binary classification in the machine learning field [47]. It is based on a large number of trees that are built in an iterative fashion. Later trees are constructed based on the errors that existed in earlier trees, giving the model more power to handle “harder" cases. This error correction ability often gives XGBoost a performance edge over other linear or tree-based methods. There is multitude of hyperparameters that need to be tuned for this algorithm. Three of them are particularly important: number of

	Hellenic CSG (N = 323)		Seville (N = 219)		Hartford Hospital (N = 323)	
	Median (IQR)	Missing %	Median (IQR)	Missing %	Median (IQR)	Missing %
Age	59.0 (47.0-72.0)	0.31%	64.0 (54.0-78.5)	0.00%	73.0 (57.0-84.0)	0.00%
Female *	125.0 (38.7%)	0.00%	91.0 (41.55%)	0.00%	176.0 (54.49%)	0.00%
Heart Rate (bpm)	88.0 (80.0-98.0)	4.95%	88.0 (77.0-100.0)	37.44%	98.0 (86.0-112.75)	0.31%
Oxygen Saturation (%)	95.0 (92.0-97.0)	16.72%	95.0 (92.0-97.0)	8.22%	93.0 (90.0-95.0)	0.31%
Temperature (°C)	38.0 (37.2-38.5)	5.57%	38.5 (38.0-38.9)	42.01%	37.8 (37-38.4)	0.31%
ALT (U/L)	27.0 (18.0-40.0)	1.86%	24.0 (16.5-39.5)	10.96%	24.0 (16.0-39.0)	12.69%
AST (U/L)	29.0 (22.0-41.0)	0.62%	28.0 (21.0-39.75)	11.42%	38.0 (29.0-58.0)	11.76%
Blood Glucose (mg/dL)	106.0 (95.0-124.0)	1.86%	111.5 (95.0-129.0)	21.46%	127.0 (107.0-165.5)	4.02%
BUN (mg/dL)	24.0 (14.56-33.8)	1.55%	16.82 (12.15-24.53)	21.46%	20.0 (13.0-33.0)	4.64%
CRP (mg/L)	53.7 (13.0-130.7)	1.86%	66.9 (23.45-138.45)	7.31%	67.85 (33.9-129.38)	35.60%
Creatinine (mg/dL)	0.9 (0.7-1.1)	1.55%	0.9 (0.76-1.15)	0.00%	1.0 (0.8-1.5)	4.33%
Hemoglobin (U/g)	13.3 (12.2-14.5)	3.10%	13.4 (11.8-14.88)	2.28%	12.2 (10.8-13.7)	3.41%
MCV (μm^3)	86.9 (83.9-89.9)	2.48%	91.1 (88.25-94.18)	2.28%	90.0 (86.0-95.0)	3.72%
Platelets ($10^3/\mu L$)	193.0 (156.0-245.0)	0.93%	204.5 (163.75-261.75)	2.28%	183.5 (140.0-241.5)	4.02%
Potassium (mEq/L)	4.1 (3.9-4.4)	1.24%	3.9 (3.6-4.3)	0.91%	4.1 (3.8-4.5)	4.95%
Prothrombin Time (INR)	1.03 (0.96-1.11)	4.95%	1.08 (1.01-1.2)	73.52%	1.2 (1.1-1.4)	53.87%
Sodium (mEq/L)	138.0 (135.0-140.0)	1.55%	139.0 (136.0-141.0)	0.91%	138.0 (135.0-140.0)	4.64%
WBC (/μL)	5710 (4380-7430)	1.55%	7180 (5200-10050)	2.28%	6800 (5000-9500)	3.72%
Cardiac dysrhythmias *	45.0 (13.98%)	0.31%	nan (nan%)	100.00%	0.0 (0.0%)	0.00%
Chronic kidney disease *	16.0 (4.97%)	0.31%	21.0 (9.95%)	3.65%	10.0 (3.1%)	0.00%
Heart disease *	60.0 (18.63%)	0.31%	55.0 (25.82%)	2.74%	0.0 (0.0%)	0.00%
Diabetes	42.0 (13.04%)	0.31%	32.0 (15.02%)	2.74%	61.0 (18.89%)	0.00%
Mortality *	32.0 (9.91%)	0.00%	28.0 (12.79%)	0.00%	46.0 (14.24%)	0.00%

Table C.2: Descriptive summary of validation population broken down by study site.
* Count (proportion) is reported for binary variables.

trees, depth of trees and learning rate. In this study, we tune the parameters: learning rate, γ, λ, α, minimum child weight, maximum tree depth, number of estimators. The learning rate, also called shrinkage factor or η, controls the weighting factor for corrections by new trees added in the model: it takes values between 0 and 1, with values closer to 1 having more corrections for each tree and higher risk of overfitting on the training data. Gamma (γ) is a regularization parameter controlling the minimum loss reduction required to make a further partition on a leaf node of a tree: it takes positive values, with larger ones defining a more conservative model. Lambda (λ) is the L2 regularization parameter on the feature weights: it takes positive values, with the larger ones encouraging smaller weights, thus making the model more conservative. Alpha (α) is the L1 regularization parameter on the feature weights: it takes positive values, with the larger ones driving to 0 the weights, defining a more conservative model. Minimum child weight is the minimum Hessian weight required to create a new node, with a role similar to that of γ, i.e. regularization at the splitting step: it takes positive values, with higher values making the model more conservative. The maximum depth of a tree controls the maximum number of nodes that can exist between the root node and the farthest leaf in the tree: it is a positive integer, and large values

usually lead to overfitting on the training data. The number of estimators determines the number of trees to fit in the model: it is a positive integer, and large values usually lead to overfitting on the training data. All remaining parameters are set to their default values.

C.2.3 SHAP methodology

SHapley Additive exPlanations (SHAP) are useful tools to interpret model predictions and risk drivers [112, 111]. The SHAP methodology explains a patient risk prediction (normalized between 0 and 1) by computing the contribution of each feature. This is obtained by approximating the nonlinear XGBoost prediction model as a linear model around the patient prediction. The coefficients of the linear approximation are estimated by introducing every feature one at a time and comparing the model output variations. We use the SHAP Python package [111], featuring an efficient algorithm to compute the SHAP values and the plot generation functions, to interpret the outcomes of XGBoost model in Figure 5-1.

C.3 Model comparison

We compared three different machine learning methods in the development of our model. In all cases, we formulate a binary classification problem to predict mortality (1) or discharge (0) as the endpoint of a patient's hospitalization. Predictive models are trained using XGBoost, Logistic Regression, and Classification And Regression Trees (CART); all methods are implemented in Scikit-learn [131]. Logistic Regression assumes an additive relationship between risk factors, whereas CART and XGBoost are able to capture non-linearities and feature interactions. While CART forms a single decision tree, XGBoost is an ensemble method: it constructs a set of decision trees which are then combined to yield a single prediction for a given patient.

We leverage the hyperparameter optimization framework Optuna [4] as follows. We first identify the corresponding parameter spaces for the Scikit-Learn implementations of XGBoost, Logistic Regression and CART [131]. Second, we define the objective function as the 300-folds cross validation area under the curve (AUC). Finally, we employ a pipeline to maximize the objective over 500 maximum iterations on multiple cores.

Table C.3 reports the AUC and various threshold-based metrics for the three algorithms. For

each method, we select the threshold that yields a sensitivity of at least 80% to reflect the priority of correctly identifying mortality. Of the three methods, XGBoost is able to capture the most sophisticated interactions between features and subsequently demonstrates the strongest performance. Logistic Regression reports a strong test set AUC but indicates a loss in specificity and precision for the chosen thresholds. CART has the highest negative predictive value but is outperformed by both other models on all other metrics.

Method	AUC	Threshold	Accuracy	Specificity	Precision	NPV
XGBoost	90.19 (86.86,93.52)	28.3 (23.26,33.34)	85.02 (81.02,89.01)	86.58 (82.77,90.39)	66.3 (61.02,71.59)	93.02 (90.17,95.87)
Logistic Regression	88.45 (84.87,92.02)	21.99 (17.36,26.62)	80.46 (76.02,84.89)	80.52 (76.09,84.95)	57.55 (52.02,63.08)	92.54 (89.6,95.48)
CART	85.85 (81.95,89.75)	23.4 (18.67,28.14)	79.8 (75.31,84.3)	77.49 (72.82,82.16)	55.93 (50.38,61.49)	94.71 (92.2,97.21)

Table C.3: AUC performance and threshold-based metrics for different machine learning methods, evaluated on the test set from the derivation cohort.

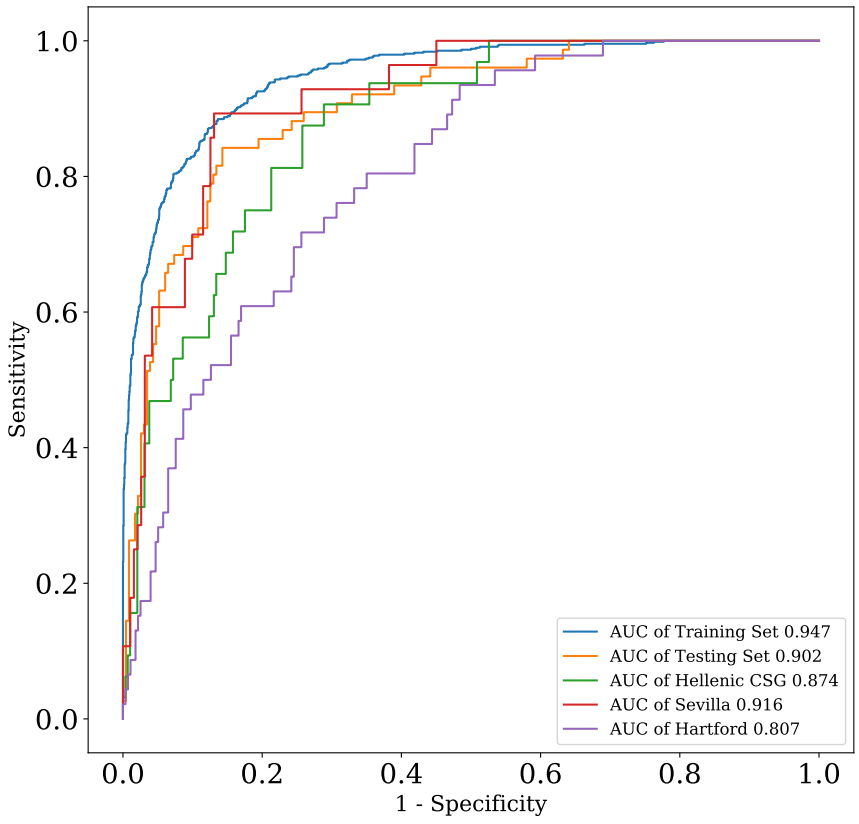


Figure C-1: Receiver operator curves (ROC) evaluating the model's performance on the testing set for patient subgroups.

Organization	Sample Size	Study Dates	Description
Sotiria Thoracic Diseases Hospital of Athens	83	03/12 – 05/07	The Sotiria Thoracic Diseases Hospital of Athens is a tertiary care hospital and the reference centre for respiratory medicine in Greece with a capacity of 710 inpatient beds, of which 400 are dedicated to pulmonary care and ICU. There is also a large sector of internal medicine and infectious disease. The clinic cares for around 150.000 in- and outpatients yearly and is on emergency rota almost daily admitting patients from the large Athens area (5.000.000 inhabitants). The Hospital was the referral centre for COVID infection in Greece and stopped all other operations and admissions during the pandemic. There were 278 admissions, most patients have been discharged and there were 28 deaths.
Evangelismos Hospital	82	03/10 – 05/04	It is the largest tertiary hospital in Greece. It is a referral center for patients with Covid-19 for ICUs and a secondary center for patients in need of hospitalization. The data comes from the Covid - 19 Patient Care Unit set up in a former 90-bed surgery wards. The staff of the clinic was provided by the internal medicine and pulmonary clinics of the hospital as well as colleagues of other hospitals who were seconded to Evangelismos including specialized internists of the hospital.
University Hospital of Alexandroupolis	50	03/14 – 05/10	It is the COVID-19 Reference Hospital for the Region of Eastern Macedonia - Thrace, an area with a large heterogeneity of population. It includes a total of 572 beds, of which 40 (Special Infections Unit and COVID-19 Clinic) exclusively for patients with SARSCoV-2, as well as the ICU (16 beds).
University Hospital of Patra	49	03/03 – 30/04	It is a modern tertiary hospital, with about 800 beds and $>$30 specialized clinics, and serves $>$ 1,500 patients a day. During the COVID-19 epidemic from February 2020, it became a reference center for Southern and Western Greece, serving a multitude of both externally confirmed cases and cases that required hospitalization in common wards as well as in ICUs.
Attikon GH	40	03/01 – 05/15	It is a 650-bed tertiary hospital in Western Attica. During the COVID-19 pandemic, the hospital was designated as a Covid-19 referral hospital. Confirmed cases were admitted in the Infectious Diseases Unit with a capacity of 8 isolated single-patient rooms or in dedicated hospital wards of 60 beds in total. Twenty ICU beds were also dedicated to Covid-19 in specific ICU areas with negative pressure.
General University Hospital of Larissa	34	03/13 – 05/14	The General University Hospital of Larissa is the referral center of the 5th Health Region of Central Greece for the management of COVID-19 patients, covering more than 1,000,000 population. Since March 2020, COVID-19 patients are managed in its Infectious Disease Unit. Patients were treated according to the therapeutic algorithms proposed by the Greek Committee of Public Health of the Ministry of Health, using hydroxychloroquine and azithomycin as the first-line main antiviral agents.

S4 Table. Overview of participating institutions in the The Hellenic COVID-19 Study Group.

Bibliography

[1] Abrams, E. M. and Szefler, S. J. (2020). COVID-19 and the impact of social determinants of health. *The Lancet Respiratory Medicine*.

[2] Agency for Healthcare Research and Quality (2017). HCUP CCS.

[3] Ahn, S., Kim, W. Y., Kim, S.-H., Hong, S., Lim, C.-M., Koh, Y., Lim, K. S., and Kim, W. (2011). Role of procalcitonin and C-reactive protein in differentiation of mixed bacterial infection from 2009 H1N1 viral pneumonia. *Influenza and Other Respiratory Viruses*, 5(6):398–403.

[4] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631.

[5] Alimadadi, A., Aryal, S., Manandhar, I., Munroe, P. B., Joe, B., and Cheng, X. (2020). Artificial intelligence and machine learning to fight COVID-19. *Physiological Genomics*, 52(4):200–202.

[6] American Diabetes Association (2018). 6. Glycemic Targets: Standards of Medical Care in Diabetes—2018. *Diabetes Care*, 41(Supplement_1):S55–S64.

[7] Amos, B., Xu, L., and Kolter, J. Z. (2016). Input Convex Neural Networks. *arXiv preprint 1609.07152*.

[8] Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. (2020). Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183(1-2):3–39.

[9] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

[10] Ashrafzadeh, S. and Hamdy, O. (2019). Patient-Driven Diabetes Care of the Future in the Technology Era. *Cell Metabolism*, 29(3):564–575.

[11] Athey, S. and Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360.

[12] Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *The Annals of Statistics*, 47(2):1–33.

[13] Aubert, C. E., Henderson, J. B., Kerr, E. A., Holleman, R., Klamerus, M. L., and Hofer, T. P. (2022). Type 2 Diabetes Management, Control and Outcomes During the COVID-19 Pandemic in Older US Veterans: an Observational Study. *Journal of General Internal Medicine*, pages 870–877.

[14] Ayca Erdogan, S., Krupski, T. L., and Lobo, J. M. (2018). Optimization of telemedicine appointments in rural areas. *Service Science*, 10(3):261–276.

[15] Bang, H. and Robins, J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973.

[16] Basak, J. and Krishnapuram, R. (2005). Interpretable hierarchical clustering by constructing an unsupervised decision tree. *Knowledge and Data Engineering, IEEE Transactions on*, 17:121– 132.

[17] Battocchi, K., Dillon, E., Hei, M., Lewis, G., Oka, P., Oprescu, M., and Syrgkanis, V. (2019). EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML. Version 0.x.

[18] Becker, P. S., Griffiths, E. A., Alwan, L. M., Bachiashvili, K., Brown, A., Cool, R., Curtin, P., Dinner, S., Gojo, I., Hicks, A., Kallam, A., Kidwai, W. Z., Kloth, D. D., Kraut, E. H., Landsburg, D., Lyman, G. H., Miller, R., Mukherjee, S., Patel, S., Perez, L. E., Poust, A., Rampal, R., Rosovsky, R., Roy, V., Rugo, H. S., Shayani, S., Vasu, S., Wadleigh, M., Westbrook, K., Westervelt, P., Burns, J., Keller, J., and Pluchino, L. A. (2020). NCCN Guidelines Insights: Hematopoietic Growth Factors, Version 1.2020. *Journal of the National Comprehensive Cancer Network*, 18(1):12–22.

[19] Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421.

[20] Bergman, D., Huang, T., Brooks, P., Lodi, A., and Raghunathan, A. U. (2019). JANOS: An integrated predictive and prescriptive modeling framework.

[21] Bertsimas, D., Borenstein, A., Mingardi, L., Nohadani, O., Orfanoudaki, A., Stellato, B., Wiberg, H., Sarin, P., Varelmann, D. J., Estrada, V., Macaya, C., and Gil, I. J. (2021a). Personalized prescription of ACEI/ARBs for hypertensive COVID-19 patients. *Health Care Management Science*, 24(2):339–355.

[22] Bertsimas, D., Boussioux, L., Cory-Wright, R., Delarue, A., Digalakis, V., Jacquillat, A., Kitane, D. L., Lukin, G., Li, M., Mingardi, L., Nohadani, O., Orfanoudaki, A., Papalexopoulos, T., Paskov, I., Pauphilet, J., Lami, O. S., Stellato, B., Bouardi, H. T., Carballo, K. V., Wiberg, H., and Zeng, C. (2021b). From predictions to prescriptions: A data-driven response to COVID-19. *Health Care Management Science*, 24(2):253–272.

[23] Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.

[24] Bertsimas, D. and Kallus, N. (2020). From Predictive to Prescriptive Analytics. *Management Science*, 66(3):1025–1044.

[25] Bertsimas, D., Lukin, G., Mingardi, L., Nohadani, O., Orfanoudaki, A., Stellato, B., Wiberg, H., Gonzalez-Garcia, S., Parra-Calderón, C. L., Robinson, K., Schneider, M., Stein, B., Estirado, A., a Beccara, L., Canino, R., Dal Bello, M., Pezzetti, F., and Pan, A. (2020). COVID-19 mortality risk assessment: An international multi-center study. *PLOS ONE*, 15(12):e0243262.

[26] Bertsimas, D., Margonis, G. A., Huang, Y., Andreatos, N., Wiberg, H., Ma, Y., Mcintyre, C., Pulvirenti, A., Wagner, D., van Dam, J., Gavazzi, F., Buettner, S., Imai, K., Stasinos, G., He, J., Seeliger, H., Kreis, M., Weiss, M. J., Cameron, J. L., Wei, A. C., Kornprat, P., Baba, H., Koerkamp, B. G., Zerbi, A., D'Angelica, M., and Wolfgang, C. L. (2021c). Towards an Optimized Staging System for Pancreatic Ductal Adenocarcinoma: A Clinically Interpretable, Artificial Intelligence-Based Model. *JCO Clinical Cancer Informatics (To Appear)*.

[27] Bertsimas, D., Masiakos, P. T., Mylonas, K. S., and Wiberg, H. (2019). Prediction of cervical spine injury in young pediatric patients: an optimal trees artificial intelligence approach. *Journal of Pediatric Surgery*, 54(11):2353–2357.

[28] Bertsimas, D., O'Hair, A., Relyea, S., and Silberholz, J. (2016a). An analytics approach to designing combination chemotherapy regimens for cancer. *Management Science*, 62(5):1511–1531.

[29] Bertsimas, D., O'Hair, A. K., and Pulleybank, W. R. (2016b). *The Analytics Edge*. Dynamic Ideas LLC.

[30] Bertsimas, D., Orfanoudaki, A., and Pawlowski, C. (2021d). Imputation of clinical covariates in time series. *Machine Learning*, 110(1):185–248.

[31] Bertsimas, D., Orfanoudaki, A., and Wiberg, H. (2021e). Interpretable clustering: an optimization approach. *Machine Learning*, 110(1):89–138.

[32] Bertsimas, D., Pauphilet, J., and Van Parys, B. (2017). Sparse classification and phase transitions: A discrete optimization perspective. *arXiv preprint arXiv:1710.01352*.

[33] Bertsimas, D. and Wiberg, H. (2020). Machine Learning in Oncology: Methods, Applications, and Challenges. *JCO Clinical Cancer Informatics*, (4):885–894.

[34] Bertsimas, D. and Dunn, J. (2018). *Machine Learning under a Modern Optimization Lens*. Dynamic Ideas, Belmont.

[35] Bezanson, J., Karpinski, S., Shah, V. B., and Edelman, A. (2012). Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*.

[36] Biggs, M., Hariss, R., and Perakis, G. (2017). Optimizing Objective Functions Determined from Random Forests. *SSRN Electronic Journal*, pages 1–46.

[37] Blockeel, H., De Raedt, L., and Ramon, J. (2000). Top-down induction of clustering trees. *arXiv preprint cs/0011032*.

[38] Bonfietti, A., Lombardi, M., and Milano, M. (2015). Embedding decision trees and random forests in constraint programming. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9075:74–90.

[39] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[40] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

[41] Cancer Therapy Evaluation Program (2006). Common terminology criteria for adverse events v3.0.

[42] Carlson, A. L., Martens, T. W., Johnson, L., and Criego, A. B. (2021). Continuous glucose monitoring integration for remote diabetes management: Virtual diabetes care with case studies. *Diabetes Technology and Therapeutics*, 23(S3):S56–S65.

[43] Center for Disease Control and Prevention (2020). Groups at Higher Risk for Severe Illness.

[44] Chavent, M., Guinot, C., Lechevallier, Y., and Tenenhaus, M. (1999). Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine. *Revue de Statistique Appliquée*, 47(4):87–99.

[45] Chen, K., Zhang, X., Deng, H., Zhu, L., Su, F., Jia, W., and Deng, X. (2014). Clinical predictive models for chemotherapy-induced febrile neutropenia in breast cancer patients: A validation study. *PLoS ONE*, 9(6).

[46] Chen, R., Liang, W., Jiang, M., Guan, W., Zhan, C., Wang, T., Tang, C., Sang, L., Liu, J., Ni, Z., Hu, Y., Liu, L., Shan, H., Lei, C., Peng, Y., Wei, L., Liu, Y., Hu, Y., Peng, P., Wang, J., Liu, J., Chen, Z., Li, G., Zheng, Z., Qiu, S., Luo, J., Ye, C., Zhu, S., Liu, X., Cheng, L., Ye, F., Zheng, J., Zhang, N., Li, Y., He, J., Li, S., and Zhong, N. (2020a). Risk Factors of Fatal Outcome in Hospitalized Subjects With Coronavirus Disease 2019 From a Nationwide Analysis in China. *Chest*, 158(1):97–105.

[47] Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-Augu, pages 785–794.

[48] Chen, Y., Shi, Y., and Zhang, B. (2020b). Input Convex Neural Networks for Optimal Voltage Regulation. *arXiv preprint 2002.08684*, pages 1–20.

[49] Cheng, Y., Luo, R., Wang, K., Zhang, M., Wang, Z., Dong, L., Li, J., Yao, Y., Ge, S., and Xu, G. (2020). Kidney disease is associated with in-hospital death of patients with COVID-19.

[50] Cho, B. J., Kim, K. M., Bilegsaikhan, S. E., and Suh, Y. J. (2020). Machine learning improves the prediction of febrile neutropenia in Korean inpatients undergoing chemotherapy for breast cancer. *Scientific Reports*, 10(1):1–8.

[51] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

[52] Cremer, J. L., Konstantelos, I., Tindemans, S. H., and Strbac, G. (2019). Data-driven power system operation: Exploring the balance between cost and risk. *IEEE Transactions on Power Systems*, 34(1):791–801.

[53] Cummings, M. J., Baldwin, M. R., Abrams, D., Jacobson, S. D., Meyer, B. J., Balough, E. M., Aaron, J. G., Claassen, J., Rabbani, L. E., Hastie, J., Hochman, B. R., Salazar-Schicchi, J., Yip, N. H., Brodie, D., and O'Donnell, M. R. (2020). Epidemiology, clinical course, and outcomes of critically ill adults with COVID-19 in New York City: a prospective cohort study. *medRxiv*, 6736(20):2020.04.15.20067157.

[54] Daniel Levy, S. B. (2006). *A change of heart : unraveling the mysteries of cardiovascular disease*. New York : Vintage.

[55] Diday, E. and Simon, J. C. (1976). *Clustering Analysis*, pages 47–94. Springer Berlin Heidelberg, Berlin, Heidelberg.

[56] Dong, E., Du, H., and Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet infectious diseases*, 20(5):533–534.

[57] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning.(2017).

[58] Drucker, H., Surges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 1:155–161.

[59] Dunn, J. C. (1974). Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, 4(1):95–104.

[60] Dunn, J. W. (2018). *Optimal trees for prediction and prescription*. PhD thesis, Massachusetts Institute of Technology.

[61] Duran, B. and Odell, P. (1974). *Cluster Analysis*. 100. Springer-Verlag Berlin Heidelberg, 1 edition.

[62] Ebert, T., Belz, J., and Nelles, O. (2014). Interpolation and extrapolation: Comparison of definitions and survey of algorithms for convex and concave hulls. In *2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 310–314.

[63] Elmachtoub, A. N. and Grigas, P. (2021). Smart "Predict, then Optimize". *Management Science*, pages 1–46.

[64] Emanuel, E. J. and Wachter, R. M. (2019). Artificial Intelligence in Health Care: Will the Value Match the Hype?

[65] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

[66] Everhart, J. and Wright, D. (1995). Diabetes mellitus as a risk factor for pancreatic cancer: a meta-analysis. *Jama*, 273(20):1605–1609.

[67] Fajemisin, A., Maragno, D., and den Hertog, D. (2021). Optimization with constraint learning: A framework and survey.

[68] Falcone, M., Corrao, S., Venditti, M., Serra, P., and Licata, G. (2011). Performance of PSI, CURB-65, and SCAP scores in predicting the outcome of patients with community-acquired and healthcare-associated pneumonia. *Internal and Emergency Medicine*, 6(5):431–436.

[69] Feinleib, M., Kannel, W., Garrison, R., McNamara, P., and Castelli, W. (1975). The framingham offspring study. design and preliminary data. *Preventive Medicine*, 4(4):518–525.

[70] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769.

[71] Foster, D. J. and Syrgkanis, V. (2019). Orthogonal Statistical Learning. *arXiv*, pages 1–86.

[72] Fraiman, R., Ghattas, B., and Svarc, M. (2013). Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2):125–145.

[73] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

[74] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

[75] Goodman, B. and Flaxman, S. (2016). European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*.

[76] Grimstad, B. and Andersson, H. (2019). ReLU networks as surrogate models in mixed-integer linear programs. *Computers and Chemical Engineering*, 131:106580.

[77] Guan, W.-j., Ni, Z.-y., Hu, Y., Liang, W.-h., Ou, C.-q., He, J.-x., Liu, L., Shan, H., Lei, C.-l., Hui, D. S. C., and Others (2020). Clinical characteristics of coronavirus disease 2019 in China. *New England Journal of Medicine*, 382(18):1708–1720.

[78] Gurobi Optimization, LLC (2021). Gurobi Optimizer Reference Manual.

[79] Gutierrez-Martinez, V. J., Cañizares, C. A., Fuerte-Esquivel, C. R., Pizano-Martinez, A., and Gu, X. (2011). Neural-network security-boundary constrained optimal power flow. *IEEE Transactions on Power Systems*, 26(1):63–72.

[80] Hahsler, M., Piekenbrock, M., and Doran, D. (2019). dbscan: Fast density-based clustering with r. *Journal of Statistical Software, Articles*, 91(1):1–30.

[81] Halilbasic, L., Thams, F., Venzke, A., Chatzivasileiadis, S., and Pinson, P. (2018). Data-driven security-constrained AC-OPF for operations and markets. *20th Power Systems Computation Conference, PSCC 2018*.

[82] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145.

[83] Hancock, T. P., Coomans, D. H., and Everingham, Y. L. (2003). Supervised Hierarchical Clustering Using CART. In *Proceedings of MODSIM 2003 International Congress on Modelling and Simulation*, pages 1880–1885, Townsville, QLD, Australia.

[84] Hastie, T., Tibshirani, R., and Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer.

[85] Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York.

[86] Hirano, K., Imbens, G. W., and Ridder, G. (2003). Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189.

[87] Holborow, A., Coupe, B., Davies, M., and Zhou, S. (2019). Machine learning methods in predicting chemotherapy-induced neutropenia in oncology patients using clinical data. *Clinical Medicine*, 19(Suppl 3):89–90.

[88] Hollander, J. E. and Carr, B. G. (2020). Virtually Perfect? Telemedicine for Covid-19. *New England Journal of Medicine*, 382(18):1679–1681.

[89] Hosmer, W., Malin, J., and Wong, M. (2011). Development and validation of a prediction model for the risk of developing febrile neutropenia in the first cycle of chemotherapy among elderly patients with breast, lung, colorectal, and prostate cancer. *Supportive Care in Cancer*, 19(3):333–341.

[90] Interpretable AI, L. (2021). Interpretable ai documentation.

[91] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.

[92] Ji, M., Li, J., Wang, S., and Peng, C. (2021). Two-Stage Robust Telemedicine Assignment Problem with Uncertain Service Duration and No-Show Behaviours. *SSRN Electronic Journal*, pages 1–37.

[93] Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer.

[94] Kannel, W. B. (1996). Blood pressure as a cardiovascular risk factor: prevention and treatment. *Jama*, 275(20):1571–1576.

[95] Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., and King, D. (2019). Key challenges for delivering clinical impact with artificial intelligence.

[96] Kim, H. S., Lee, S. Y., Kim, J. W., Choi, Y. J., Park, I. H., Lee, K. S., Seo, J. H., Shin, S. W., Kim, Y. H., Kim, J. S., and Park, K. H. (2016). Incidence and Predictors of Febrile Neutropenia among Early-Stage Breast Cancer Patients Receiving Anthracycline-Based Chemotherapy in Korea. *Oncology*, 91(5):274–282.

[97] Kirby, T. (2020). Evidence mounts on the disproportionate effect of COVID-19 on ethnic minorities. *The Lancet Respiratory Medicine*.

[98] Kleijnen, J. P. (2015). Design and analysis of simulation experiments. In *International Workshop on Simulation*, pages 3–22. Springer.

[99] Krim, H. and Hamza, A. B. (2015). *Geometric methods in signal and image analysis*. Cambridge University Press.

[100] Kudła, P. and Pawlak, T. P. (2018). One-class synthesis of constraints for Mixed-Integer Linear Programming with C4.5 decision trees. *Applied Soft Computing Journal*, 68:1–12.

[101] Lacoma, A., Bas, A., Tudela, P., Gimanez, M., Madol, J. M., Parez, M., Ausina, V., Dominguez, J., and Prat-Aymerich, C. (2014). Correlation of inflammatory and cardiovascular biomarkers with pneumonia severity scores. *Enfermedades Infecciosas y Microbiologiaa Clininica*, 32(3):140–146.

[102] Larose, D. T. and Larose, C. D. (2014). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.

[103] Li, E., Mezzio, D. J., Campbell, D., Campbell, K., and Lyman, G. H. (2021a). Primary Prophylaxis With Biosimilar Filgrastim for Patients at Intermediate Risk for Febrile Neutropenia: A Cost-Effectiveness Analysis. *JCO Oncology Practice*, page OP.20.01047.

[104] Li, Z., Xu, T., Zhang, K., Deng, H. W., Boerwinkle, E., and Xiong, M. (2021b). Causal Analysis of Health Interventions and Environments for Influencing the Spread of COVID-19 in the United States of America. *Frontiers in Applied Mathematics and Statistics*, 6(January):1–13.

[105] Liang, W., Liang, H., Ou, L., Chen, B., Chen, A., Li, C., Li, Y., Guan, W., Sang, L., Lu, J., Xu, Y., Chen, G., Guo, H., Guo, J., Chen, Z., Zhao, Y., Li, S., Zhang, N., Zhong, N., He, J., and for the China Medical Treatment Expert Group for COVID-19 (2020). Development and Validation of a Clinical Risk Score to Predict the Occurrence of Critical Illness in Hospitalized Patients With COVID-19. *JAMA Internal Medicine*.

[106] Lim, W. S. (2003). Defining community acquired pneumonia severity on presentation to hospital: an international derivation and validation study. *Thorax*, 58(5):377–382.

[107] Lippi, G., Plebani, M., and Henry, B. M. (2020). Thrombocytopenia is associated with severe coronavirus disease 2019 (COVID-19) infections: A meta-analysis. *Clinica Chimica Acta*, 506(March):145–148.

[108] Liu, B., Xia, Y., and Yu, P. S. (2000). Clustering through decision tree construction. In *Proceedings of the ninth international conference on Information and knowledge management - CIKM '00*, pages 20–29, McLean, VA.

[109] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE.

[110] Lombardi, M., Milano, M., and Bartolini, A. (2017). Empirical decision model learning. *Artificial Intelligence*, 244:343–367.

[111] Lundberg, S., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):2522–5839.

[112] Lundberg, S. M. and Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 4766–4775.

[113] Lyman, G. H., Kuderer, N. M., Crawford, J., Wolff, D. A., Culakova, E., Poniewierski, M. S., and Dale, D. C. (2011). Predicting individual risk of neutropenic complications in patients receiving cancer chemotherapy. *Cancer*, 117(9):1917–1927.

[114] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

[115] Mahtta, D., Daher, M., Lee, M. T., Sayani, S., Shishehbor, M., and Virani, S. S. (2021). Promise and Perils of Telehealth in the Current Era. *Current Cardiology Reports*, 23(9):1–6.

[116] Majithia, A. R., Kusiak, C. M., Lee, A. A., Colangelo, F. R., Romanelli, R. J., Robertson, S., Miller, D. P., Erani, D. M., Layne, J. E., Dixon, R. F., and Zisser, H. (2020). Glycemic outcomes in adults with Type 2 diabetes participating in a continuous glucose monitor-driven virtual diabetes clinic: Prospective trial. *Journal of Medical Internet Research*, 22(8):1–9.

[117] Maragno, D., Wiberg, H., Bertsimas, D., Birbil, S. I., Hertog, D. d., and Fajemisin, A. (2021). Mixed-integer optimization with constraint learning. *arXiv preprint arXiv:2111.04469*.

[118] Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654.

[119] Miller, D. D. and Brown, E. W. (2018). Artificial Intelligence in Medical Practice: The Question to the Answer? *The American Journal of Medicine*, 131:129–133.

[120] Mišić, V. V. (2020). Optimization of tree ensembles. *Operations Research*, 68(5):1605–1624.

[121] Morrison, F., Shubina, M., and Turchin, A. (2011). Encounter frequency and serum glucose level, blood pressure, and cholesterol level control in patients with diabetes mellitus. *Archives of Internal Medicine*, 171(17):1542–1550.

[122] MOSEK (2019). *MOSEK Optimizer API for Python 9.3.7*.

[123] Mouselimis, L. (2019). *ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering*. R package version 1.2.0.

[124] National Cancer Institute (2021). Treatment clinical trials for gastric (stomach) cancer.

[125] Offner, P. J., Moore, E. E., and Biffl, W. L. (1999). Male gender is a risk factor for major infections after surgery. *Archives of Surgery*, 134(9):935–940.

[126] Pastors, J. G., Franz, M. J., Warshaw, H., Daly, A., and Arnold, M. S. (2003). How effective is medical nutrition therapy in diabetes care?(commentary). *Journal of the American Dietetic Association*, 103(7):827–832.

[127] Pawlak, T. P. (2019). Synthesis of mathematical programming models with one-class evolutionary strategies. *Swarm and Evolutionary Computation*, 44:335–348.

[128] Pawlak, T. P. and Krawiec, K. (2019). Synthesis of constraints for mathematical programming with one-class genetic programming. *IEEE Transactions on Evolutionary Computation*, 23(1):117–129.

[129] Pawlak, T. P. and Litwiniuk, B. (2021). Ellipsoidal one-class constraint acquisition for quadratically constrained programming. *European Journal of Operational Research*, 293(1):36–49.

[130] Pawloski, P. A., Thomas, A. J., Kane, S., Vazquez-Benitez, G., Shapiro, G. R., and Lyman, G. H. (2017). Predicting neutropenia risk in patients with cancer using electronic data. *Journal of the American Medical Informatics Association*, 24(e1):e129–e135.

[131] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

[132] Peters, K., Silva, S., Gonçalves, R., Kavelj, M., Fleuren, H., den Hertog, D., Ergun, O., and Freeman, M. (2021). The nutritious supply chain: Optimizing humanitarian food assistance. *INFORMS Journal on Optimization*, 3(2):200–226.

[133] Piatetsky-Shapiro, G., Djeraba, C., Getoor, L., Grossman, R., Feldman, R., and Zaki, M. (2006). What are the grand challenges for data mining?: Kdd-2006 panel report. *ACM SIGKDD Explorations Newsletter*, 8(2):70–77.

[134] Pourhomayoun, M. and Shakibi, M. (2020). Predicting Mortality Risk in Patients with COVID-19 Using Artificial Intelligence to Help Medical Decision-Making. *medRxiv*, page 2020.03.30.20047308.

[135] Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919 – 938.

[136] Rajkomar, A., Dean, J., and Kohane, I. (2019). Machine learning in medicine.

[137] Rao, C. R. (1964). The use and interpretation of principal component analysis in applied research. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):329–358.

[138] Robins, J. M., Hernan, M. A., and Brumback, B. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560.

[139] Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994). Estimation of Regression Coefficients When Some Regressors are not Always Observed. *Journal of the American Statistical Association*, 89(427):846–866.

[140] Rodriguez, J. A., Saadi, A., Schwamm, L. H., Bates, D. W., and Samal, L. (2021). Disparities in telehealth use among california patients with limited english proficiency. *Health Affairs*, 40(3):487–495.

[141] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

[142] Ruan, Q., Yang, K., Wang, W., Jiang, L., and Song, J. (2020). Clinical predictors of mortality due to COVID-19 based on an analysis of data of 150 patients from Wuhan, China.

[143] Ruspini, E. H. (1970). Numerical methods for fuzzy clustering. *Information Sciences*, 2(3):319–350.

[144] Schweidtmann, A. M. and Mitsos, A. (2019). Deterministic Global Optimization with Artificial Neural Networks Embedded. *Journal of Optimization Theory and Applications*, 180(3):925–948.

[145] Sendelbach, S. and Funk, M. (2013). Alarm fatigue: A patient safety concern. *AACN Advanced Critical Care*, 24(4):378–386.

[146] Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR.

[147] Skiena, S. S. (2008). *The Algorithm Design Manual*. Springer Publishing Company, Incorporated, 2nd edition.

[148] Smith, T. J., Bohlke, K., Lyman, G. H., Carson, K. R., Crawford, J., Cross, S. J., Goldberg, J. M., Khatcheressian, J. L., Leighl, N. B., Perkins, C. L., and Others (2015). Recommendations for the use of WBC growth factors: American Society of Clinical Oncology clinical practice guideline update. *Journal of Clinical Oncology*, 33(28):3199–3212.

[149] Sneath, P., Sneath, P., Sokal, R., and Sokal, U. (1973). *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. A Series of books in biology. W. H. Freeman.

[150] Spyros, C. (2020). From Decision Trees and Neural Networks to MILP: Power System Optimization Considering Dynamic Stability Constraints. In *2020 European Control Conference (ECC)*, pages 594–594. IEEE.

[151] Sroka, D. and Pawlak, T. P. (2018). One-class constraint acquisition with local search. *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, pages 363–370.

[152] Stepp, R. E. and Michalski, R. S. (1986). Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43–69.

[153] Tang, X., Wu, C., Li, X., Song, Y., Yao, X., Wu, X., Duan, Y., Zhang, H., Wang, Y., Qian, Z., Cui, J., and Lu, J. (2020). On the origin and continuing evolution of SARS-CoV-2. *National Science Review*, pages 1–24.

[154] Taplitz, R. A., Kennedy, E. B., Bow, E. J., Crews, J., Gleason, C., Hawley, D. K., Langston, A. A., Nastoupil, L. J., Rajotte, M., Rolston, K., Strasfeld, L., and Flowers, C. R. (2018). Outpatient management of fever and neutropenia in adults treated for malignancy: American Society of Clinical Oncology and Infectious Diseases Society of America Clinical practice guideline update. *Journal of Clinical Oncology*, 36(14):1443–1453.

[155] Thams, F., Halilbaši, L., Pinson, P., Chatzivasileiadis, S., and Eriksson, R. (2017). Data-Driven Security-Constrained OPF. In *Proc. 10th Bulk Power Syst. Dyn. Control Symp.*, pages 1–10.

[156] The Editorial Board (2018). Towards trustable machine learning. *Nature Biomedical Engineering*, 2(10):709–710.

[157] Totten, A. M., McDonagh, M. S., and Wagner, J. H. (2020). The evidence base for telehealth: Reassurance in the face of rapid expansion during the COVID-19 pandemic. *Agency for Healthcare Research and Quality*, pages 1–9.

[158] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525.

[159] Ultsch, A. (2005). Fundamental clustering problems suite (fcps). Technical report, University of Marburg.

[160] UNHCR, UNICEF, WFP, and WHO (2002). Food and nutrition needs in emergencies.

[161] University of California San Francisco (2019). Basic metabolic panel.

[162] Vayena, E., Blasimme, A., and Cohen, I. G. (2018). Machine learning in medicine: Addressing ethical challenges. *PLoS Medicine*, 15(11).

[163] Venzke, A., Viola, D. T., Mermet-Guyennet, J., Misyris, G. S., and Chatzivasileiadis, S. (2020). Neural Networks for Encoding Dynamic Security-Constrained Optimal Power Flow to Mixed-Integer Linear Programs. *arXiv preprint 2003.07939*.

[164] Verwer, S., Zhang, Y., and Ye, Q. C. (2017). Auction optimization using regression trees and linear models as integer programs. *Artificial Intelligence*, 244:368–395.

[165] Wang, G., Wu, C., Zhang, Q., Wu, F., Yu, B., Lv, J., Li, Y., Li, T., Zhang, S., Wu, C., Wu, G., and Zhong, Y. (2020a). C-Reactive Protein Level May Predict the Risk of COVID-19 Aggravation. *Open Forum Infectious Diseases*, 7(5).

[166] Wang, Y., Wang, Y., Chen, Y., and Qin, Q. (2020b). Unique epidemiological and clinical features of the emerging 2019 novel coronavirus pneumonia (COVID-19) implicate special control measures. *Journal of Medical Virology*, 92(6):568–576.

[167] Wiberg, H., Yu, P., Montanaro, P., Mather, J., Birz, S., Schneider, M., and Bertsimas, D. (2021). Prediction of neutropenic events in chemotherapy patients: A machine learning approach. *JCO Clinical Cancer Informatics*, 5:904–911.

[168] Wilkerson, R. G., Adler, J. D., Shah, N. G., and Brown, R. (2020). Silent hypoxia: A harbinger of clinical deterioration in patients with COVID-19. *The American Journal of Emergency Medicine*, pages undefined–undefined.

[169] Wolf, P. A., D'Agostino, R. B., Kannel, W. B., Bonita, R., and Belanger, A. J. (1988). Cigarette smoking as a risk factor for stroke: the framingham study. *Jama*, 259(7):1025–1029.

[170] Wolfe, P. (1961). A duality theorem for non-linear programming. *Quarterly of Applied Mathematics*, 19(3):239–244.

[171] Wu, J., Xiong, H., and Chen, J. (2009). Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–886. ACM.

[172] Xie, J., Covassin, N., Fan, Z., Singh, P., Gao, W., Li, G., Kara, T., and Somers, V. K. (2020). Association Between Hypoxemia and Mortality in Patients With COVID-19. *Mayo Clinic Proceedings*.

[173] Yan, L., Zhang, H.-T., Goncalves, J., Xiao, Y., Wang, M., Guo, Y., Sun, C., Tang, X., Jing, L., Zhang, M., Huang, X., Xiao, Y., Cao, H., Chen, Y., Ren, T., Wang, F., Xiao, Y., Huang, S., Tan, X., Huang, N., Jiao, B., Cheng, C., Zhang, Y., Luo, A., Mombaerts, L., Jin, J., Cao, Z., Li, S., Xu, H., and Yuan, Y. (2020). An interpretable mortality prediction model for COVID-19 patients. *Nature Machine Intelligence*, pages 1–6.

[174] Yang, D., Hendifar, A., Lenz, C., Togawa, K., Lenz, F., Lurje, G., Pohl, A., Winder, T., Ning, Y., Groshen, S., and Lenz, H.-J. (2011). Survival of metastatic gastric cancer: Significance of age, sex and race/ethnicity. *Journal of Gastrointestinal Oncology*, 2(2):77–84.

[175] Yang, Q. and Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604.

[176] Zhou, F., Yu, T., Du, R., Fan, G., Liu, Y., Liu, Z., Xiang, J., Wang, Y., Song, B., Gu, X., Guan, L., Wei, Y., Li, H., Wu, X., Xu, J., Tu, S., Zhang, Y., Chen, H., and Cao, B. (2020). Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study. *The Lancet*, 395.