Analytics and Decision Making in Sustainable Operations

by

Leann Thayaparan

B.S.E., Princeton University (2016)
MBAn., Massachusetts Institute of Technology (2019)

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 2024

Authored by: Leann Thayaparan
Sloan School of Management
May 1, 2024

Certified by: Georgia Perakis
John C Head III Dean (Interim), MIT Sloan School of Management
Professor, Operations Management, Operations Research & Statistics
Thesis Supervisor

Accepted by: Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Analytics and Decision Making in Sustainable Operations

by

Leann Thayaparan

Submitted to the Sloan School of Management
on May 1, 2024 in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

ABSTRACT

Sustainable operations has transformed in the past decade, as interest from consumers, companies and regulators has increased. There has been a growing excitement and necessity to leverage the large-scale data collected to improve the modelling and decision making around sustainable operations. In this thesis, we introduce new methodologies to support data-driven sustainable operations, and in specific deal with topics around electric vehicles and the COVID-19 pandemic.

In Chapter 2, we consider the problem of electric vehicles (EVs) as distributed storage for the electric grid. While electric vehicles can act as batteries supporting both the home and the electric grid consumption, uncertainty around car usage must be accounted for before they can be used in practice. We introduce a driver behavior-focused dynamic optimization for the charging and discharging of electric vehicles. We characterize policies that are interpretable to drivers to address distrust of automatized discharging of car batteries and prove analytically the regimes under which such policies are optimal. Finally, we work closely with an American EV manufacturer to show the dollar and carbon footprint that can be expected to be saved from discharging based on driving behavior. We do this by clustering drivers based on their driving behavior to derive probability distributions of when and how much drivers use their car to feed into the dynamic optimization.

We further develop the challenge of data-driven decision making in sustainability through Chapter 3. Rather than learning probability distributions as in Chapter 2, we introduce a deterministic approach in which a tree-ensemble model, specifically a random forest, forecasts how much drivers use their EV. This gives rise to a challenge from the predict-then-optimize literature around the tractability of optimizations in which an objective function is determined by a tree ensemble model. In this chapter we introduce an Upper Bounding Method for Optimizing over Tree Ensemble Models, UMOTEM. We demonstrate the scalability of UMOTEM, showing it grows linearly with regard to both the number of trees in the ensemble as well as those trees' depth. This is a strong improvement over comparable formulations which grow exponentially. We also bound the optimality gap introduced through the approximation, characterizing it using features of the random forest such as leaf separation and in-sample error. We computationally compare our approximation to similar methods, demonstrating that the algorithm captures over 90% of optimality in 2% of the runtime for publicly available datasets. Finally, we demonstrate the use of UMOTEM through two case studies. First, we take the same case as Chapter 2, and show how UMOTEM can be

leveraged to optimize the charging and discharging of EVs. Second, we work closely with Oracle Retail to apply UMOTEM to promotion scheduling in order to determine an optimally markdown strategy for a fashion retailer.

In the final chapter of this thesis, we address data-driven decision making in one of the other major operational challenges to affect the globe, the COVID-19 pandemic. We develop a SIR-based model that can account for multiple waves. This model is agnostic to what drives the new waves (new variants, behavior changes, government policies, etc.) but takes a data-driven approach to identify when infection rates change. We prove analytical guarantees on how fast new waves can be detected. When modelling COVID-19, we show a new wave can be expected to be flagged within half a week. We also show strong computational results on COVID-19, demonstrating improvement over top COVID-19 forecasting models used by the CDC.

Thesis supervisor: Georgia Perakis
Title: John C Head III Dean (Interim), MIT Sloan School of Management
Professor, Operations Management, Operations Research & Statistics

Acknowledgments

"If I have seen further, it is by standing upon the shoulders of giants." I owe my utmost gratitude to all those who have guided me through my time at MIT.

First and foremost, I would like to thank my advisor Georgia Perakis. It is an understatement to say that without her guidance, I would not be here. I am unbelievably fortunate to have had the opportunity to be Georgia's student. Getting a PhD is infamous for being a long and arduous process, but I would consider my time as Georgia's student as some of my most delightful years yet. Georgia is unparalleled in the dedication she gives to her students. Under her care, I have had fun learning how to be a researcher, a mentor, an academic and an advocate. At every turn, Georgia has supported me with all she has, encouraging me to go farther and achieve more than I would have thought possible. I will be forever grateful to Georgia for everything she has done for me. It is a debt that can never be repaid.

I would like to thank my thesis committee, Stephen Graves, Retsef Levi and Karen Zheng. They have been supportive of me throughout my time as a PhD student. I would like to specially thank them for this past year as they have spent a significant amount of time helping me shape not just my thesis but how I present myself to our research community. They have also been instrumental in encouraging me in my pursuit of an academic career.

I am also very grateful to the ORC faculty and staff who have helped me through my time as a PhD student. I would like to thank Patrick Jaillet and Georgia, for their roles as the ORC co-directors. During my time in the center they have fostered an environment that focused on collaboration and community above all else. I would like to thank Vivek Farias, Dimitris Bertsimas, Nikos Trichakis, Andy Sun, Negin Golrezaei, Swati Gupta, Daniel Freund and Alexandre Jacquillat for investing their time and consideration to help, support and encourage me in my research and career. I am also grateful to Laura Rose and Andrew Carvalho for all they have done to help me navigate the PhD and keep the ORC moving.

I have had the pleasure to collaborate with some amazing students at the ORC. I would like to extend a special thank to Tamar Cohen-Hillel, one of my very first collaborators at the ORC. Tamar took me under her wing when I was still in my Masters, encouraged me to pursue a PhD, and taught me how to keep my stamina up during the marathon that was this PhD. I would also like to thank Divya Singhvi, who I started working with during the COVID-19 pandemic. Divya has been incredibly generous with me, both in terms of time and support. From teaching me how to write a theorem in a paper to allowing me to practice all my major presentations with him, his guidance has been invaluable. My gratitude to Omar Skali Lami, whose organization and patience worked wonders in helping "Team 2" navigate the pandemic. My thanks to Yannis Spantidakis, who always knows how to keep meetings fun and exciting. I would like to thank Asterios Tsiourvas, Lennart Baardman,

Rares Cristian, Amine Bennouna, and David Nze-Ndong, all of whom have been incredible to collaborate with over the years. I would like to thank the Masters and Undergraduate students with whom I have had the pleasure of working with over the years. Stacy Wang, Ian Tongs, Bibhabasu Das, Theo Dawson, Chloe Pariente, Shane Weisberg and Divya Shyamal, have all been crucial in pushing the work of this thesis forward.

Without my collaborators in industry, this thesis would have neither the data nor the opportunity for impact. I will forever be grateful to Derek Hazard, whom I met during my Masters capstone and has been a massive advocate for my sustainability work. Derek championed the work that became Chapter 1, from getting our research approved within his company, to leading the project through its early years. Even now as he has gone on to greater endeavors, he has continued to support my research and academic career. I would like to thank Robert Welborn, my other Masters capstone mentor, who also supported the start of our research collaboration. My thanks to Maya Bam and Mahsa Moussavi, who has patiently and consistently worked with us over the years, helping us navigate everything from data, to industry experts, to system changes. I would like to thank our collaborators at Oracle Retail, Kiran Panchamgam, Setareh Borjian and Rebecca Schubertruegmer, who were crucial in the development of UMOTEM.

I am so grateful for all the friends I have made during my time at the ORC. To my fellow Georgia students, Elisabeth, Angela, Emily, Manuel, Alkiviadis, William and Xuening, it has been so good being in this with you. I will treasure all our dinners together, and look forward with anticipation to our next one! It was my honor to serve as an INFORMS officer with my fellow champions of community building, Léonard, Dan and Cynthia. To Léonard, you shine with an optimism, kindness and energy that lights up everyone who has the honor to be around you. To Dan, I have never met someone with such unflagging persistence or willingness to serve others. To Cynthia, I will forever be inspired by your ability to find beauty in every moment. Working with the three of you helped cut through the isolation of COVID and I'm so proud of what we accomplished together. I'm grateful to have entered the ORC already with a community of fellow AOE alumns to support me. A special thank you to Ted Papalexopulous, who from the first time we shared an office years before either of us entered the ORC, has been an incredible friend and confidante for every step I have made since. To Julia Yan, who has been a source of wisdom since I was in undergrad, thank you for your generosity in time and advice. To Rainy, whose determination, work-ethic and dedication is only rivaled by your kindness. I would not have survived my early years at the ORC if it wasn't for my friends who joined the ORC alongside me, Moise, Shuvo, Victor, Kevin, Evan, Kim, Sabrina and Ken. I would like to thank Agni, Arthur, Emma, Matthew, Holly, Peter, Ryan, Lea, Patricio, Andy, Michael, Ghali, Qinyi, Raluca, Yu and Alex, for the conversations, advice and memories over the years. I would especially like to thank the INFORMS officers, REFS and other students who have so thanklessly served the ORC during my time. The ORC would not run without your contributions keeping it going.

To the many friends who have supported me outside the ORC, I'm so grateful for the help you've given over these past years. Rachael, who has been there for me from the beginning. Alice, who taught me that we could actually find our family. Kristin, whose faith and kindness is matched only by her willingness to serve her family, friends and community. Riley, whose mad genius, creativity and thoughtfulness makes him a great friend and terrifying DM. Colin, whose eagerness to learn about the world around him inspires me to do the same.

Leea, whose strength of conviction motivates me to question the privileges I take for granted and to do better for the world. Lena, my longest roommate, I will forever cherish our dinners and pots of tea together. Angeline, who during our days in College Ave taught me how to be a functioning adult. Kayla, who thanklessly advocates for those who couldn't help themselves. Anna, your willingness to listen to anything at truly any hour, inspires me to be a better friend to others. Joyce, who has surprised me at every turn but most impressed me by her integrity. Amelia and Miloš, who has shown me that distance is not a barrier to closeness. Emma, who has taught me to gamely embrace trying new things. Bob, Haley, Megan and Matt, who started as my coworkers but have long since become my friends, advisors and inspiration.

Last, but possibly most influentially, I would like to thank my large and extremely supportive family. To my grandparents, Appamma, Appappa, Ammamma, Phata, Grandma and Grandpa, thank you for building the foundation upon which all the rest of this rests. You taught me the value, and more importantly, the joy in learning. Thank you to my Uncles, Aunts, and many cousins who have cheered me on throughout this process.

Thank you to my (baby) brother and companion in arms, Aadrien. You are my closest friend, my biggest champion, my sternest teacher and my greatest advocate. Thank you for constantly being curious about and supportive of my research and my academic pursuits. You always encourage me to be the best version of myself and pick me up when I'm not.

Finally, to my parents, to whom this thesis is dedicated. Amma and Appa, you were the first to encourage me to go to grad school. I cannot begin to list the ways this thesis is owed to your support. I would not be the person I am today without you. If I do any good with my work, it is because you taught me the importance of leaving a place better than how I found it. My passion for sustainability comes from your teachings. Thank you for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The United Nation identified 17 goals as part of its 2030 Agenda for Sustainable Development [1]. These goals are intended to "recognize that ending poverty and other deprivations must go hand-in-hand with strategies that improve health and education, reduce inequality, and spur economic growth – all while tackling climate change and working to preserve our oceans and forests". Among several other initiatives, these goals include good health and well-being, affordable and clean energy, climate action and decent work and economic growth. It is exactly this framework that we aim to make actionable through developing data-driven decision making tools that enable sustainable operations. We focus on two specific areas within this thesis. First we will discuss affordable and clean energy by exploring the potential of the vehicle-to-grid market. Second we will deal with the operational challenges around good health and well-being during the global crisis of the COVID-19 pandemic. Drawing insights from close collaborations with industry partners such as an American EV Manufacturer, Oracle and the Center for Disease Control, we show a tangible impact on sustainable decision making in a variety of contexts.

Regarding clean energy and climate action, we are at a point of unprecedented investment in green technology. In 2022, global investments in decarbonizing energy surpassed $1 trillion, nearly doubling our investment in 2019 and for the first time reaching equivalence with our investment in supporting fossil fuel supply [2]. In the United States, the Inflation Reduction Act committed over $369 billion towards energy security and reducing climate impact [3]. This interest has only been matched from the industry perspective: "92 percent of executives say that new businesses built in the next five years will address sustainability to some extent—and 42 percent expect to put sustainability at the center of their new businesses' value proposition" [4]. A significant role will be played by renewable energy generators and the electric grid during these investments, as clean energy is the base upon which a lot of green technology rests. However, with these unprecedented investments rise new and exciting operations management challenges. Specifically, as we move towards more renewable resources, the ability to produce electricity in time with demand diminishes. Instead, rises a need for energy storage or the ability to produce electricity when renewables allow and store it for when demand needs it later. Electric Vehicles (EVs) have been discussed as a

way of providing a distributed energy storage resource to the electric grid, allowing for a new market to emerge from vehicle discharging. Amidst this excitement about a potential new market, there has been interest in understanding exactly how much capacity EVs have to offer the grid.

In the realm of health and well-being, one of the greatest challenges in recent years has been the COVID-19 pandemic. COVID-19 led to widespread illness, hospitalizations, and deaths globally. It strained healthcare systems, causing shortages of medical supplies and hospital resources. The pandemic exacerbated mental health issues due to social isolation, anxiety about the virus, grief from loss of loved ones, and economic stressors such as job loss. Beyond the phsyical and mental health impact, it had significant economic consequences as well. Global supply chains faced disruptions due to factory closures, transportation restrictions, and increased demand for certain goods such as personal protective equipment (PPE) and medical supplies. Small businesses, restaurants, and other establishments faced closures or significant revenue losses due to lockdowns and reduced consumer spending. It became important to be able to understand and predict new waves of COVID-19 so that individuals, businesses and policy-makers could respond accordingly. From helping epidemiologists and public health officials understand the transmission dynamics of the virus, to providing decision-makers with valuable insights into the potential impact of different policy interventions, to forecasting the trajectory of the pandemic and anticipating future waves of infections, operational modelling played a significant role in the response to COVID-19.

In parallel with this interest in sustainable operations, both regarding clean energy and healthcare, there has been growing excitement around using large data to drive decision making. Organizations have become increasingly reliant on data from their operations and customers to determine forecasts and decisions. When planning, businesses often engage in a two step process, first forecasting what will happen in the future and then second, using these forecasts to optimize decisions. However, the quality of the decisions eventually made, and their real world outcomes, are wholly dependent on the caliber of the predictive models built. Even if an optimization method is able to find the true optimal point when the objective function is determined by a model, the usefulness of this solution will be negligible if the model's representation of the real world is poor. It is for this reason that there has been a large push in research to understand how cutting edge methods can be used as inputs when optimizing decisions. This thesis focuses on data-driven modeling to uncover patterns and behaviors essential for informed decision-making. Beyond modeling, the thesis explores the integration of these models into optimization frameworks to determine optimal strategies. Analytical proofs validate the robustness and generalizability of the proposed models, ensuring their applicability across diverse contexts. By bridging the gap between technical methodologies and real-world business needs, this thesis offers an approach to tackling the operational challenges of sustainability-focused businesses in today's dynamic marketplace.

1.2 Contributions

The main contribution of this thesis is developing methods for making data-driven decisions in sustainable operations. This is achieved through a focus on methodological work which

pairs new algorithms with analytical theory to show generalizability and close collaboration with partners in industry to quantify impact.

In terms of methodology, this thesis introduces three novel algorithms for incorporating the wealth of available data into optimal decision making via machine learning models. All three models address settings where there is access to high-volume, granular data but there is uncertainty in how to filter or transform the data to inform decisions in a tractable way. In Chapter 2, we discuss a clustering approach to subset the drivers into relevant groups and learn probability distributions on consumer behavior. This allows us to reduce information on a specific driver to relevant clusters, and we validate the number of clusters through testing. In Chapter 3, we deal with tree ensemble models, a category of machine learning models, and introduce an approximation algorithm that can tractably solve objective functions determined by these tree ensembles. In Chapter 4, we model how the granular data on new COVID-19 cases and deaths can be used to identify when a new wave is starting. In doing so we create an algorithmic solution to not just flagging new waves but also to determining when COVID-19 infection data is too outdated to be considered when training models for the current wave of the infection.

In addition to introducing these methods, we are able to guarantee properties on these models that make them widely applicable to other settings. Specifically, in Chapter 2, we guarantee that policies produced by the model will be first interpretable, specifically following a two-threshold policy and second, tractable to solve, showing that the optimization solution can be found in closed form. In Chapter 3, we bound the expected optimality gap of the approximation algorithm based on features of the random forest to show under what setting the algorithm will perform well. Specifically we demonstrate that as separation between predicted values and in-sample accuracy increases, the expected optimality gap of the approximation will decrease. Finally in Chapter 4, we prove how long it should take to identify new waves of COVID-19, and for the parameters of the 2020 pandemic, show that it should take less than a week.

Finally we work closely with industry collaborators to quantify the impact of the work. This verifies that the practical relevance of the work. We do not just identify an application where the model's performance is strong but collaborate closely with industry partners to make sure that the outputs of the algorithm give tangible insights and recommendations to decision makers to improve their process. In Chapters 2 and 3, we work closely with an American EV manufacturer to assess how much capacity EVs have to potentially power a home or give back to the grid. This has been used to evaluate the potential of the vehicle discharging market and determine investments. Through this we show that EV drivers can save between \$41-\$95 in 12 days. In Chapter 3, we also discuss a case study with Oracle Retail where the model is used to determine promotion markdowns for a fashion retailer. We show an optimal discount strategy could improve store revenue by 12-13% in a 21 week horizon. In Chapter 4, we compare the results of the multiwave model against top models submitted to the CDC. We demonstrate that the multiwave model, with an out-of-sample error of 7-8%, significantly improves upon the original epidemiology model, which had an out-of-sample error of 27-31%. We also outperform top CDC forecasting models, especially as a new wave begins.

The organization of the thesis is as follows. In Chapter 2, we develop an optimization and guarantees for how to optimize the charging/discharging of EVs in the distributed electricity

storage market of vehicle-to-grid. In Chapter 3, we propose an algorithm for tractably optimizing when objectives are determined by tree ensemble models and discuss examples in electric vehicles and retail promotions. In Chapter 4, we propose a multiwave epidemiology model for flagging and modelling new epidemic waves. The chapters of this thesis are self-contained and can be read in any order.

Chapter 2

The Role of Driver Behavior and Interpretability in the Vehicle-to-Grid Market

2.1 Introduction

In the coming decades, two major changes are expected to affect the electric grid network. First, we expect a growing investment in decarbonizing the grid. In 2022, global investments in decarbonizing energy surpassed $1 trillion, nearly doubling the investment in 2019. For the first time these investments were equivalent to the investment supporting fossil fuel supply. This growth is only expected to accelerate, as demand for wind and solar increases and governments enact regulation to ease the transition. The second change is the automotive industry's transition from Internal Combustion Engines to Electric Vehicles (EVs). The demand for EVs has shot up exponentially, with year over year growth going from 4% between 2020 and 2021, to 14% between 2021 and 2022 to a forecasted 35% between 2022 and 2023 [5]. This demand has only been matched by automotive companies, as manufacturers promise to transition their fleets to fully EV. Both Ford and Stellantis forecast that their American fleet will be at least 40% EVs by 2030 [6], [7]. General Motors has promised that they will sell only zero emission vehicles by 2035 [8]. While promising, both of these changes pose risks to the electric grid network. Specifically, as the electric grid network becomes less dependent on carbon-based sources, and instead replaces general generation with renewables, the ability to match supply of electricity with demand decreases.

Instead what rises in importance is storage, that is, the ability to create energy when renewables allow and store it for when demand requires it later. The supply and demand mismatch is very costly for the electric grid, as carbon-based generators must compensate the fluctuating demand with generation in order to match the new variability of the net demand (demand without solar or wind production) curve. In addition, the cost of flexible generation is reflected in the price of energy, making increased renewable portfolios expensive for consumers [9]. Other concerns about greater reliance on renewables include oversupply risk, where renewable generators which were planned to have lower generation unexpectedly produce at higher capacity resulting in price volatility and reliability costs. Experts also

expect challenges around frequency management if the grid is dependent on renewables as the primary source of electricity generation. In frequency management, utility companies manage the load on the grid to hold a consistent Hertz in the electricity through granular control of the supply. As dependency on renewables increases, this kind of minute control decreases leading to concerns about frequency reliability. However, if the grid supported large-scale storage, these problems could be directly addressed as during the times the grid is oversupplied, the energy could be directed towards storage and when it is under-supplied, the energy can be pulled from storage. Furthermore, by pulling from storage the grid could maintain its granular control of the grid's frequency.

However, supporting storage at the scale the grid would need is a costly proposition. Therefore, in this work we consider EVs as a form of distributed storage for the electric grid, when EVs are stationary and plugged in the grid. Selling energy from an EV's battery to the grid, commonly called vehicle-to-grid (V2G), or giving energy from the vehicle to the home to reduce demand referred to as vehicle-to-home (V2H) is a proposed market that has yet to appear in the US at scale. Nevertheless, this is of significant interest to both utility companies and regulators. For example, in Spring 2023, California considered requiring all electric vehicles sold in the state to be required to have V2G capabilities built in [10]. Despite this interest, there are still outstanding questions on how large the V2G market is expected to be. For example, a significant concern is if enough EVs will be attached to the grid during the times when the grid needs storage or if these EVs will have enough energy in their battery to support the grid. Furthermore, drivers purchase EVs to use them as personal vehicles first and foremost. Two significant questions in this field are *how can the car be discharged to support the grid given uncertainty around how the driver will use their vehicle?* And *how can we generate charging and discharging policies that are easy to communicate and trustworthy to drivers?* In this work we address these questions by proposing and analyzing a dynamic programming (DP) model which controls the charging and discharging of electric vehicles. We take a driver-focused perspective, optimizing when the car should charge/discharge to serve driver interests regarding their profit, carbon footprint, driving ability and range anxiety. This is because, as a privately owned vehicle, the driver's needs must be satisfied for them to participate in the market. We then aggregate independent individual-level optimizations to size the total potential of the market. We learn using actual deidentified data from our industry collaborator (a large EV manufacturer) probability distributions for driver usage, and use these as transition probabilities in the DP. Through this work, we propose policies that are easy to communicate to drivers and have structures that can be considered reliable. Through our partnership with a large American EV manufacturer, we use their deidentified driver data to quantify what these policies are for drivers and demonstrate dollar and carbon benefits to the drivers. More generally, this close collaboration aims to help our industry collaborator to assess how much capacity EVs have to potentially power a home or give back to the grid.

2.1.1 Contributions

In what follows, we highlight the contributions of this work in more detail:

- **We introduce a driver-focused dynamic optimization model that incorpo-**

rates the EV charging and discharging behavior: We take a driver-centric view in this optimization and formulate a DP that allows for both the charging and discharging of EVs while accounting for uncertainty around driver behavior. We include as part of our objective and constraints concerns around having enough energy in the EV for emergency driving and departing with less than necessary charge. We explicitly value how much the driver uses their car.

- **We characterize policies that can be understandable to drivers**: We address driver distrust of discharging by guaranteeing that policies are easy to communicate. We quantify this by defining policies which have a monotonic relationship between the recommended quantity to charge/discharge and the charge level of the car. As a result, drivers can see exactly at what charge level and time their car's policy will switch from charging to discharging.

- **We analytically prove the monotonicity of the policies that the DP gives rise to**: We guarantee parameter regimes under which the DP policies are monotonic and can be found in closed form. As part of this work, we show that the required parameters directly reflect the optimization's focus on supporting the drivers' car usage. We validate our theorem's assumptions through actual deidentified data from our industry collaborator and confirm that these assumptions indeed hold in practice.

- **We quantify the impact in terms of dollar and carbon benefit**: We work closely with a large American EV manufacturer to cluster their drivers into subgroups from which driver behavior probability distributions can be learned to use in the DP. As part of this we show that eight clusters of driver behavior are cohesive and provide clear patterns for car usage. We then show what is the dollar and carbon impact of discharging on each of these clusters, with on average $95 dollars and the carbon equivalent of 91.1 gallons of gasoline saved in 12 days.

2.2 Literature Review

In this section, we provide a brief review of the related literature for V2G, energy storage and load management for the grid, and EV charging. While existing literature has analyzed the problem from the utility's perspective of load management or storage, as well as from a fleet owner of EVs, to the best of our knowledge this is the first work that has taken an individual driver's perspective on how much drivers can offer the grid outside of their daily driving needs. This is also the first work to accomplish this using deidentified individual EV driver data from a collaboration with a large American EV manufacturer.

Vehicle-to-Grid:

V2G is still an emerging market, with only a couple of pilots occurring around the United States as of 2023. As such the literature around the capacity, levers, marketing and impact of V2G is still emerging. However, forecasting the potential of the V2G market has already begun both in literature and in practice. From a broad forecast of V2G capabilities, [11] describe the stages of V2G capabilities as both the market share and technical sophistication of V2G management increase. More concretely, [12] model the economics of V2G from the

perspective of an aggregator of a fleet of EVs, where contract parameters, such as how long the car will be attached to the grid, or the car's usage when it is away is fully known, demonstrating an estimated profit to drivers of $132 to $177 annually (translated to USD from the euro). [13] also optimizes V2G charging and discharging, assuming drivers provide knowledge of departure times when joining the charging system. [14] propose a heuristic to the optimization, where they assume the car will be used from 8 AM to 5 PM, and as a result, no charging is allowed during this window. Their model assumes that the car needs at the start of the day, 8 AM, to be fully charged, resulting in an estimated profit to the driver of $140 to $250 annually. [15] formulate a two-stage stochastic integer program to optimize EV fleet planning to support carsharing and V2G with the remaining capacity. Similarly, [16] considers the business case where the main purpose of the fleet is car rental and remaining capacity is offered to the grid in V2G. By comparison, we tackle this V2G management problem without assuming deterministic knowledge of driver usage or departure time. Unlike cases where an EV fleet owner controls how much of its inventory is used for carsharing or rental, we analyze the case where the aggregator does not control driver behavior and hence usage.

EV Charging:

Closely related to the field of V2G is EV battery charging and management, as timing charging is half of the V2G operations. [17] consider the chemical nature of lithium-ion batteries, the current standard for EV batteries, and model optimal management for safety and battery lifetime while also giving insights into the challenges of estimating a battery's state of charge. [18] models scheduling vehicle charging from the perspective of the electric charging service provider under stochastic EV arrivals to the system. [19] optimizes a charging policy for system operator of an on-demand transportation fleet while taking into account demand, fleet size and spatial considerations. [20] propose a smart charging model in which utility companies announce charging cost/time pair options in order to help with load management and show both cost and emissions savings that result from such business models. [21] tackle how to charge an EV fleet to make EV carsharing profitable. [22] use spatial pricing of electricity to manage the load EVs place on the electric grid. [23] model how an aggregator can negotiate on behalf of individual drivers for cheaper charging regimes.

Stationary Storage:

The optimization regime for charging and discharging under V2G we propose in this chapter, is similar to the stationary storage (SS) model, in which stationary batteries are used either by individual consumers or larger utility firms to support electric needs. Unlike V2G in which battery storage is an ancillary benefit to the vehicle, which is primarily a means for transportation, SS units are purchased for the purpose of storing energy. [24] consider off-grid use cases, namely found when consumers are remote, and understand appropriate investment in renewables and storage, with back up carbon-based generation. [25] create a storage control policy and analyze how utility companies should invest in storage capabilities alongside renewable energy (wind and solar) and flexible generators (natural gas). [26] propose a model for storage owners to bid in the day-ahead energy markets while maintaining capacity for the intraday market. [27] propose a dynamic programming model for coordinating energy storage when there is uncertainty about renewable supple, energy prices and energy demand. The main challenge we have to account for compared to these works is uncertainty about when the car will be attached and how much energy to hold in

reserve for the vehicles normal driving. SS units naturally are available all the time, while EVs as storage have significantly more uncertainty.

As stated at the beginning of this section, we bring actual deidentified driver data from an EV manufacturer to be able to model individual EV driver behavior. In doing so we can take a driver centric approach to the challenge of V2G when EVs are privately owned (as compared to the case when EVs are controlled by a single fleet owner). As a result we are able to propose a model formulation and optimal policies that can be easily communicated to drivers and directly incorporporate their driving patterns to serve their car usage first and foremost and then within remaining capacity, serve the grid.

2.3 Model Formulation

Individual drivers cannot just offer energy to the grid through V2G just when it is convenient for them. There needs to be some central coordinator (decision maker) for when cars discharge helping control the market and negotiating with utility companies for the prices the cars discharge at. This role can be played by the utility companies themselves, EV manufacturers or another market operator, who is offering vehicle-to-grid profit opportunities to the drivers and controls if and when an EV discharges energy. As a result, in this section, we will introduce a dynamic programming formulation from the point of view of the central decision maker which will incorporate the charging and discharging driver behavior of an EV's battery. Our formulation takes into account two stakeholders, the driver and the central decision maker, making decisions during T time periods. The driver owns the EV and wishes to first use their EV as a vehicle, but within their remaining capacity, use their EV to earn or save money by V2X (here V2X refers to the combined market of V2G and V2H). The driver will not change their driving behavior to earn or save more money and does not want any V2X decisions to significantly impact their driving behavior (that is, driving needs). The central decision maker, such as the car manufacturer or a utility company, acts on the driver's behalf by having aligned incentives (namely the central decision maker also aims for the EV to be used first as vehicle and then under remaining capacity allow the driver to earn or save money). They control the charging and discharging of the vehicle when the vehicle is attached to the grid. Nevertheless, it is important to note that the central decision maker does not have information on exactly what driving decisions the driver will make. As a result, we do not consider a bilevel optimization formulation because the central decision maker and the driver's incentives are aligned. At each time period, $t = 1, \ldots, T$, the following sequence of events occurs:

- If the car is attached to the grid:
  - At the start of the time period, the central decision maker decides whether to charge or discharge the car's battery during the time period.
  - At the end of the time period, the driver chooses whether to remain at the grid or depart from the grid.
- If the car is away from the grid:

– At the start of the time period, the driver decides how much to discharge their car during that time period.

– At the end of the time period, the driver chooses whether to remain away from the grid or return to it.

The central decision maker, while acting on behalf of and for the benefit of the driver, does not have direct information on the driver's choices during these T time periods. This means that when the central decision maker develops an optimal charging or discharging policy, they do not know with certainty if the driver will leave at the end of the time period, or how much the driver will use the car when away. However, they do have access to data with which to estimate the probability of these decisions occurring.

The main input to this optimization model is data on when drivers depart and reattach to the grid and how much of their battery they use when they are away from the grid. This is then used to calculate transition probabilities for what drivers do at the end of each time period and how much they discharge when away from the grid. We take as input pricing data on how much it costs to charge the EV, the price energy can be sold back to the grid and the price the home uses to pay for electricity. We also bring in data on the CO2 produced to generate electricity at each time step. The main parameters of the optimization are $C_{\min}$, the minimum charge level the car must be at before we can discharge, λ_R a penalty on how much the driver cares about being carbon neutral, $\lambda_{\min,t}$ the penalty for being below $C_{\min}$ when attached to the grid, $\lambda_{\max,t}$ the penalty for being below fully charged when attached to the grid, $\lambda_{\text{time},t}$ the temporal discount drivers have of future time steps, and β a linear utility on driving. Finally, the output of our proposed optimization model is a policy for the central decision maker for how much to charge or discharge to home or the grid for each time step and charge level. The policy implicitly affects driver behavior, as choosing to charge/discharge the vehicle will affect the charge level of the car, in turn influencing how much the driver uses the vehicle.

The states of our proposed dynamic programming model are the charge level, c, the time t, and whether or not the car is attached to the grid, j. We assume discrete charge level buckets for tractability. For simplicity of notation, we assume c is integer defined between 0 and the maximum battery value $C_{\max}$. Nevertheless, this can be generalized to different sized buckets. In this setting j is a binary variable, in which if $j = 0$ then the car is not attached to the grid and if $j = 1$ the car is attached to the grid. Thus the optimization formulation needs to use estimates of $p_{c,t,j}$, namely, of the probability the driver attaches to the grid at charge c and time t, given that it started the time period t either attached to the grid $j = 1$ or away from the grid $j = 0$. The optimization also needs to use estimates $q_{\gamma,c,t}$, which represent the probability that the driver uses γ of its battery given the car starts away from the grid at time t and at charge level c. Both $p_{c,t,j}$ and $q_{\gamma,c,t}$ are estimated empirically from data.

Given these estimations that are inputs to the formulation, in what follows we will describe the dynamic optimization formulation in detail. The main decision variable is x, that represents the amount to either charge or discharge in terms of the car battery. We define x^{charge} as the amount of energy purchased to charge the battery, x^{V2H}, as the amount of energy given to run the home and x^{V2G} as the amount of energy sold back to the grid. For ease of notation, we will define x^{charge} as negative, as the driver needs to pay for the energy.

Conversely, x^{V2H} and x^{V2G} will be positive variables, as the driver earns money with each of these quantities.

We define the price for making these decisions as w. w_t^{charge} is the price paid at time t to charge one unit of x. Similarly, w_t^{V2H} is the amount saved from running the home off of the car's battery at time t while w_t^{V2G} is the amount earned from selling a unit of x to the grid at time t. Therefore, the amount of money spent, saved or earned at time t can be described as $x^{\text{charge}} w_t^{\text{charge}} + x^{\text{V2H}} w_t^{\text{V2H}} + x^{\text{V2G}} w_t^{\text{V2G}}$. It is worth noting, that these prices implicitly incorporate the price of charging/discharging as well as the cost or fee of the action.

We also consider the possible carbon benefit of deciding to charge or discharge at time t. We define r_t as the amount of carbon used to produce a unit of energy x at time t and the parameter λ_R as the drivers sensitivity to being carbon neutral. A larger λ_R indicates a more sensitive driver, meaning they would rather charge when carbon production is low and discharge when carbon production is high. We account for the carbon benefit as part of the price of charging or discharging so that the monetary and carbon benefit of the central decision maker's choices can be described as $-x^{\text{charge}} \left(w_t^{\text{charge}} + \lambda_R r_t \right) + x^{\text{V2H}} \left(w_t^{\text{V2H}} + \lambda_R r_t \right) + x^{\text{V2G}} \left(w_t^{\text{V2G}} + \lambda_R r_t \right)$. For ease of notation we abbreviate the equation by defining the set of option categories $A := \{\text{charge, V2G, V2H}\}$ and writing the profit and carbon benefit as:

$$\sum_{a \in A} x^a \left(w_t^a + \lambda_R r_t \right). \tag{2.1}$$

The optimization's decision must account for factors other than just pure monetary and carbon benefit. We define $C_{\max}$ as the maximum value for the battery and $C_{\min}$ as the minimum necessary charge needed in the battery before the optimization can discharge the battery. We define the parameter $\lambda_{\max,t}$ as the driver's aversion to being less than fully charged at time t and $\lambda_{\min,t}$ as the driver's aversion to being less than minimum charged at time t. The idea behind $\lambda_{\max,t}$ is that some drivers will have a strong preference to being higher charged, while with $\lambda_{\min,t}$, some drivers there is a strong preference to just being reasonably charged. We assume a linear utility to being at higher charge levels and mathematically represent this by penalizing being at charge level c at time t as:

$$-\lambda_{\max,t} \left(C_{\max} - c \right) - \lambda_{\min,t} \left(C_{\min} - c \right)^+. \tag{2.2}$$

Finally the optimization must take into account the value of possible next states. Let $V\left(s_{c,t,j}\right)$ represent the value of being at charge level c, time t, and attached to the grid or not, j. Let $E\left[V_j\left(s_{c,t+1}\right)\right]$ represent the expected value of the next states, given that the car is ending time period t with charge c and attached to the grid j, but there is uncertainty if at the next step they will remain either attached to the grid or away. At time t and charge level c, the optimization decides how much to charge or discharge the car, $\sum_{a \in A} x^a$. This means that at the end of the time period, when the driver is deciding whether to depart or remain attached, they are doing so at charge level $c - \sum_{a \in A} x^a$ and time t. The expected value of the next possible states, $E\left[V_1\left(s_{c-\sum_{a \in A} x^a, t+1}\right)\right]$, is the probability the car remains attached to the grid at this new charge level $\left(p_{c-\sum_{a \in A} x^a, t, 1}\right)$ times the value of having remained plus the probability the car departs $\left(1 - p_{c-\sum_{a \in A} x^a, t, 1}\right)$ times the value starting the next time state away from the grid, or mathematically:

$$E\left[V_1\left(s_{c-\sum_{a\in A}x^a,t+1}\right)\right] = p_{c-\sum_{a\in A}x^a,t,1}V\left(s_{c-\sum_{a\in A}x^a,t+1,1}\right) + \left(1-p_{c-\sum_{a\in A}x^a,t,1}\right)V\left(s_{c-\sum_{a\in A}x^a,t+1,0}\right). \quad (2.3)$$

Temporal discounting also must be accounted for. Drivers are more concerned with the current time step, both in terms of dollar profit and whether the car is charged enough, than potential future earnings. So we add a discount to the future time step by multiplying the expectation by $\frac{1}{\lambda_{\text{time},t}}$.

Therefore, we can define the optimization's objective as the sum of these different pieces. The value of making the optimal decision at time t and charge c when attached to the grid, is by definition $V\left(s_{c,t,1}\right)$ which gives rise to:

$$V\left(s_{c,t,1}\right) = \max_x \sum_{a\in A} x^a\left(w_t^a + \lambda_R r_t\right) - \lambda_{\max,t}\left(C_{\max}-c\right) - \lambda_{\min,t}\left(C_{\min}-c\right)^+ + \frac{1}{\lambda_{\text{time},t}}E\left[V_1\left(s_{c-\sum_{a\in A}x^a,t+1}\right)\right]. \quad (2.4)$$

Naturally, the optimization is constrained in several directions. First, offered capacity to the home or the grid must be less than the amount of energy in the car's battery:

$$x^{\text{V2H}} + x^{\text{V2G}} \le c.$$

The car cannot both charge and discharge during the same time step:

$$\begin{aligned} x^{\text{charge}} &\le C_{\max}\left(1-r\right), \\ x^{\text{V2H}} + x^{\text{V2G}} &\le C_{\max} r, \\ r &\in \{0,1\}. \end{aligned}$$

If the decision is to discharge, then the charge level at the next stage cannot be below $C_{\min}$. This prevents the optimization from driving the car battery so low that the driver cannot use the car in the case of an emergency:

$$c - \sum_{a\in A} x^a \ge C_{\min} r$$

The car's battery in the next stage cannot exceed maximum charge:

$$c - \sum_{a\in A} x^a \le C_{\max}.$$

Any quantity offered the grid must be positive, and conversely we will define the quantity of energy for charging as negative:

$$\begin{aligned} x^{\text{V2H}}, x^{\text{V2G}} &\ge 0, \\ x^{\text{charge}} &\le 0. \end{aligned}$$

The car's charging or discharging cannot exceed the maximum rate for the port. We define the maximum charging rate as $X_{\max}^{\text{charge}}$ and the maximum discharging rate as $X_{\max}^{\text{V2X}}$

$$\begin{aligned} x^{\text{charge}} &\ge -X_{\max}^{\text{charge}}, \\ x^{\text{V2H}} + x^{\text{V2G}} &\le X_{\max}^{\text{V2X}}. \end{aligned}$$

We now present these constraints together with the objective function to define our proposed dynamic optimization formulation in full. In doing so, we combine a few constraints

to simplify the presentation of the formulation:

$$\begin{aligned}
&\max_{x} \sum_{a\in A} x^a \left(w_t^a + \lambda_R r_t\right) - \lambda_{\max,t}\left(C_{\max} - c\right) - \lambda_{\min,t}\left(C_{\min} - c\right)^+ \\
&\quad + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^a, t+1}\right)\right] \\
\text{s.t.}\quad & x^{\text{V2H}} + x^{\text{V2G}} \le c, \\
& x^{\text{charge}} \ge -X_{\max}^{\text{charge}}\left(1 - r\right), \\
& x^{\text{V2H}} + x^{\text{V2G}} \le X_{\max}^{\text{V2X}} r, \\
& C_{\min} r \le c - \sum_{a\in A} x^a \\
& c - \sum_{a\in A} x^a \le C_{\max}, \\
& x^{\text{V2H}}, x^{\text{V2G}} \ge 0, \\
& x^{\text{charge}} \le 0, \\
& x^{\text{V2H}}, x^{\text{V2G}}, x^{\text{charge}} \in Z, \\
& r \in \{0,1\}.
\end{aligned} \tag{2.5}$$

For the sake of guaranteeing structural properties of the solution, we relax the constraint $C_{\min} r \le c - \sum_{a\in A} x^a$ into the objective with constant λ_c:

$$\begin{aligned}
V\left(s_{c,t,1}\right) = \max_{x} & \sum_{a\in A} x^a \left(w_t^a + \lambda_R r_t\right) - \lambda_{\max,t}\left(C_{\max} - c\right) - \lambda_{\min,t}\left(C_{\min} - c\right)^+ \\
& + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^a, t+1}\right)\right] + \lambda_c \left(c - \sum_{a\in A} x^a\right) \\
\text{s.t.}\quad & x^{\text{V2H}} + x^{\text{V2G}} \le c, \\
& x^{\text{charge}} \ge -X_{\max}^{\text{charge}}\left(1 - r\right), \\
& x^{\text{V2H}} + x^{\text{V2G}} \le X_{\max}^{\text{V2X}} r, \\
& c - \sum_{a\in A} x^a \le C_{\max}, \\
& x^{\text{V2H}}, x^{\text{V2G}} \ge 0, \\
& x^{\text{charge}} \le 0, \\
& x^{\text{V2H}}, x^{\text{V2G}}, x^{\text{charge}} \in Z, \\
& r \in \{0,1\}.
\end{aligned} \tag{2.6}$$

In Section 2.4 we will show how to select λ_c so that $C_{\min} r \le c - \sum_{a\in A} x^a$ is guaranteed to hold.

Formulation (2.6) defines the Bellman equation for if the car starts the time step attached to the grid. However, when the car starts the time step away from the grid, we consider a different Bellman equation. When the car is away from the grid, there is no decision that

the optimization makes with regard to charging or discharging. Instead the driver uses an unknown amount, γ, of its battery with probability $q_{\gamma,c,t}$. Let the maximum the car's battery can discharge be $\gamma_{\max}$. The driver will end the time step with charge $c-\gamma$, at which point they have some probability of returning to the grid ($p_{c-\gamma,t,0}$) or remaining away. We once again account for temporal discounting using λ_{time}. The value in terms of next states of having discharged gamma is:

$$\frac{1}{\lambda_{\text{time},t}} E\left[V_0\left(s_{c-\gamma,t+1}\right)\right] = \frac{1}{\lambda_{\text{time},t}} \left(p_{c-\gamma,t,0} V\left(s_{c-\gamma,t+1,1}\right) + \left(1-p_{c-\gamma,t,0}\right) V\left(s_{c-\gamma,t+1,0}\right)\right).$$

It is also important to account for the value of having used γ of the car's battery. Otherwise the optimization would endeavor to minimize driving with the car in a way that does not serve the driver. We incorporate the linear utility a driver receives from driving as: $\beta_t \gamma$. This results in the following value function for being away from the grid at charge level c and time t:

$$V\left(s_{c,t,0}\right) = \sum_{\gamma=0}^{\gamma_{\max}} q_{\gamma,c,t} \left(\frac{1}{\lambda_{\text{time},t}} E\left[V_0\left(s_{c-\gamma,t+1}\right)\right] + \beta_t \gamma\right) \tag{2.7}$$

For the rest of the chapter we will assume a linear utility for driving, but we note that any increasing function $b_t(\gamma)$ could be used.

We define the value of the final time step, T, as 0 for all charge levels of the car. These values, along with the value functions (2.6) and (2.7) define the DP. In the next section we analyze the properties of the DP to define regimes under which the solution to the DP is tractable and implementable.

2.4 Structural Properties of the DP Policies

We start this section by discussing the selection of the optimization's parameters $\lambda_{\max,t}$ and β_t. The selection of these parameters can guarantee that the expected value of the states, $E\left[V_j\left(s_{c,t}\right)\right]$, is non-decreasing and concave in the charge level c across all time steps t. We then prove that if the value function is non-decreasing and concave, then all optimal solutions to the optimization will be monotonic in terms of the charge level, a necessary property for the optimization to be implemented in service of drivers. Finally, we will leverage the concave nature of the objective function along with the monotonic property of the solution to provide a closed form solution to the optimization formulation. This will allow us to establish interesting insights.

The need for $E\left[V_j\left(s_{c,t}\right)\right]$ to be non-decreasing and concave is intuitive. A non-decreasing function indicates that it should always be valuable to the driver to charge up to a higher charge state, and this should be weighted against the price of charging to determine if charging is the optimal decision. Furthermore, there is diminishing returns on the higher charge levels. Namely, it should be more important to charge at a low state (such as charging from 40% to 55%) than it is to charge at the higher levels (such as charging from 85% to 100%). This indicates that the value of the next states should also be concave.

There are several reasons why naive parameter selection might result in these properties being violated by the value function. For example, if β_t is not selected correctly, then a

car left unused at 40% for several hours because its charge level is too low to be used is considered more valuable to the optimization than a car that was charged up to 55%, used by its driver and then returned back to the grid at 10%. In this chapter, we control $\lambda_{\max,t}$, the penalty for being less than fully charged when attached to the grid, and β_t, the value of having used the car's battery when driving, in order to appropriately weight the importance of driving behavior.

In order to guarantee monotonic policies and concave expected value functions, we need some structure on the driver behavior decisions of EVs. We start by defining concave stochastic dominant distributions, a necessary property for driver usage. The property of concave stochastic dominance holds across a set of random variables Γ_c if first order stochastic dominance [28] holds for the distributions for every c and $c+1$ and if the cumulative distributions are concave in c (i.e., $P(\Gamma_{c+1,t} \leq \gamma) - P(\Gamma_{c,t} \leq \gamma) \leq P(\Gamma_{c,t} \leq \gamma) - P(\Gamma_{c-1,t} \leq \gamma) \quad \forall\gamma$). Concave stochastic dominant distributions have some "ideal" distribution that the set approaches as the index c increases. In the case of the distribution of driving usage, this can be thought of as if there is a driving behavior the driver wishes to achieve with a trip. If they have the charge level to achieve it, they will and if not they will get as close as possible.

We now introduce three assumptions on the distributions learned on driver behavior:

1. The probability of being at the grid is lower at higher charge ($p_{c,t,j} \geq p_{c+1,t,j}$).

2. The probability of being at the grid has a second derivative that is smaller than the value function in terms of magnitude:

$$\frac{-2p_{c,t,j} + p_{c+1,t,j} + p_{c-1,t,j}}{p_{c,t,j}} \leq \frac{2V(s_{c,t,1}) - V(s_{c+1,t,1}) - V(s_{c-1,t,1})}{V(s_{c,t,1})}$$

and

$$\begin{aligned}\frac{2p_{c,t,j} - p_{c+1,t,j} - p_{c-1,t,j}}{1 - p_{c,t,j}} \leq & \frac{2V(s_{c,t+1,0}) - V(s_{c+1,t+1,0}) - V(s_{c-1,t+1,0})}{V(s_{c,t+1,0})} \\ & - \frac{2(V(s_{c+1,t+1,0}) - V(s_{c,t+1,0}))(p_{c,t,j} - p_{c+1,t,j})}{(1 - p_{c,t,j})V(s_{c,t+1,0})} \\ & - \frac{2(V(s_{c,t+1,0}) - V(s_{c-1,t+1,0}))(p_{c-1,t,j} - p_{c,t,j})}{(1 - p_{c,t,j})V(s_{c,t+1,0})}.\end{aligned}$$

3. A driver's car usage, $q_{\gamma,c,t}$ is concave stochastic dominant across charge level c

These assumptions on the behavior of drivers are reasonable (as we discuss below) and hold for the data from our industry collaborator. The first assumption says that as the charge in the car increases, the probability of departure also increases. This assumption makes sense, as the charge in the car affects how much usage a driver can get out of it, so the driver is likely to wait for a higher charge before departing.

The second assumption says that the probabilities have a small second derivative over the charge level. In practice we use linear regression to learn the probability of being attached or away from the grid. As a result, the probabilities have a second derivative of 0, ensuring that Assumption 2 holds in the model.

We illustrate how Assumptions 1 and 2 hold in our industry collaborator's data using Figure 2.3 in Section 2.5.

The third and final assumption is that the usage when the cars are away is concavely stochastic dominant. When calculating this distribution in practice, there is a challenge of data sparsity when calculating how drivers' usage varies by charge level. For example, while cars attached to the grid at 4AM appear at a variety of charge levels in the data, there are already so few cars in use at 4AM that further splitting by charge level results in significant missing values. Therefore, when we learn these distributions for EV usage in Section 2.5, we learn distributions without separating data by charge level. This means that $q_{\gamma,c,t}$ learned in Section 2.5 is the same for all c, for a given driver. While less than ideal in terms of data quality, it does guarantee that Assumption 3 will hold.

Using these assumptions, Theorem 2.4 establishes regimes of $\lambda_{\max,t}$ and β_t, that allow us to characterize the optimal solution and establish structural properties. Finally, we propose an algorithm to solve the optimization formulation.

Let $V\left(s_{c,t,j}\right)$ be the expected value of starting time step t with charge c where the driver is either attached to the grid, $j=1$, or away from the grid, $j=0$. We define $p_{c,t,j}$ as the probability the driver attaches to the grid at charge c and time t, given that they started the time period t either attached to the grid $j=1$ or away from the grid $j=0$. We further define $q_{\gamma,c,t}$ as the probability that the driver uses γ of its battery given the car starts away from the grid at time t and charge c. If Assumptions 1-3 hold, then:

1. For a given time t, we can guarantee that the expected value for all states, $E\left[V_j\left(s_{c,t}\right)\right] = p_{c,t,j}V\left(s_{c,t+1,1}\right) + \left(1-p_{c,t,j}\right)V\left(s_{c,t+1,0}\right)$ is non-decreasing and concave in c, for the following choice of parameters:

$$\lambda_{\max,t} \geq \frac{p_{c,t,1} - p_{c+1,t,1}}{p_{c+1,t,1}} V^*_{c,t}, \quad \forall c \in [0,\ldots,C_{\max}]$$

where

$$V^*_{c,t} = X^{\text{V2X}}_{\max}\left(\max\left(w^{\text{V2G}}_t, w^{\text{V2H}}_t\right) + \lambda_R r_t - \lambda_c\right) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{\min\left(c+X^{\text{charge}}_{\max},100\right),t+1}\right)\right].$$

and

$$\beta_t \geq \frac{1}{\lambda_{\text{time},t}}\left(E\left[V_0\left(s_{c+1,t+1}\right)\right] - E\left[V_0\left(s_{c,t+1}\right)\right]\right) \quad \forall c \in [0,\ldots,C_{\max}-1]$$

2. The DP can be solved in closed form and the optimal solution of the DP is monotonic in terms of the charge level c.

The values for both $\lambda_{\max,t}$ and β_t help intuitively translate the driver utility to a monetary value. In the case of $\lambda_{\max,t}$, there are two components to its bound. First there is $\frac{p_{c,t,1}-p_{c+1,t,1}}{p_{c,t,1}}$, this represents the driver's sensitivity to the charge level. If the probability of the driver leaving varies greatly by charge level for a particular time, then this term will be high. However, if the driver is insensitive to the charge level, such as possibly when they are leaving for work, then this term will be small. The sensitivity to the charge level is then multiplied by $V^*_{c,t} = X^{\text{V2X}}_{\max}\left(\max\left(w^{\text{V2G}}_t, w^{\text{V2H}}_t\right) + \lambda_R r_t - \lambda_c\right) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{\min\left(c+X^{\text{charge}}_{\max},100\right),t+1}\right)\right]$, which upper bounds the maximum possible dollar value to be earned at time t, thus normalizing the sensitivity to the dollar value. Therefore, $\lambda_{\max,t}$ allows us to translate the driver's sensitivity

to being at higher charge to a dollar value that can be compared to the rest of the objective function's terms.

β_t similarly translates driver behavior to a value that the optimization formulation can use. $\beta_t \geq \frac{1}{\lambda_{\text{time},t}} \left(E\left[V_0\left(s_{c+1,t+1}\right)\right] - E\left[V_0\left(s_{c,t+1}\right)\right]\right),\ \forall c \in [0, \ldots, C_{\max} - 1]$, which means that the linear utility the driver gets from using their car is greater than the value they lose in the next steps from discharging the battery. This has to be the case or else the driver would never use the car, and the optimization in turn would try to minimize the amount of driving the driver engages in. Appropriately setting β_t means that the optimization values the car being used as a car more highly than it being used as a battery.

Before we present the proof of Theorem 2.4 in detail, we first present some intuition behind the key ideas of the proof. To prove the first part of Theorem 1, we use Step 1 in order to break $E\left[V_j\left(s_{c,t}\right)\right]$ into its two components, $p_{c,t,j} V\left(s_{c,t+1,1}\right)$ and $\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$, for which we show that both of these terms are non-decreasing and concave separately. In doing so, we guarantee that their sum is also non-decreasing and concave. As part of Step 1 we also present the optimal solution in closed form. Then in Step 2 of the proof we use this solution and show that it is monotonic in terms of charge level c. Next we present the proof in detail.

Proof of Theorem 2.4: We assume that $V\left(s_{c,t+1,j}\right)$ is non-negative for all c. This holds without loss of generality as we can add $\min_c V\left(s_{c,t+1,j}\right)$ to all values of the function. We start by proving that the $E\left[V_j\left(s_{c,t}\right)\right]$ is non-decreasing and concave. This is proved in Step 2.4.

Under the assumptions of Theorem 2.4, we construct $\lambda_{\max,t}$ and β_t so that $E\left[V_j\left(s_{c,t}\right)\right]$ is non-decreasing and concave. The proof of this step follows from the fact that $E\left[V_j\left(s_{c,t}\right)\right]$ can be written as $p_{c,t,j} V\left(s_{c,t+1,1}\right) + \left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$. Namely, it equals the probability the car remains attached to the grid times the value of remaining attached to the grid in the next time step plus the probability of being away from the grid times the value of being away from the grid at the next step. In the final stage T, we have that $V\left(s_{c,T,j}\right) = 0$. This makes $E\left[V_j\left(s_{c,T}\right)\right] = 0$, regardless of the transition probabilities. Thus in the final stage, $E\left[V_j\left(s_{c,t}\right)\right]$ is non-decreasing and concave in c. The proof will follow by induction. We will start by assuming that $E\left[V_j\left(s_{c,t+2}\right)\right]$ is non-decreasing and concave and prove this is also true for time step $t+1$. We establish the result of Step 1 in four parts as follows:

- Step 1.1: $p_{c,t,j} V\left(s_{c,t+1,1}\right)$ is non-decreasing
- Step 1.2: $\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$ is non-decreasing
- Step 1.3: $p_{c,t,j} V\left(s_{c,t+1,1}\right)$ is concave
- Step 1.4: $\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$ concave

Step 1.1 $p_{c,t,j} V\left(s_{c,t+1,1}\right)$ is non-decreasing

We start by establishing that $V\left(s_{c,t+1,1}\right)$ is non-decreasing in c. $V\left(s_{c,t+1,1}\right)$ by definition is the objective value in Formulation (2.6). The charge level c plays a role in a few places in the formulation. The first place is in the objective function, $-\lambda_{\max,t}\left(C_{\max} - c\right) - \lambda_{\min,t}\left(C_{\min} - c\right)^+ + \lambda_c\left(c - \sum_{a \in A} x^a\right)$. As these are penalties for being at lower charge, these naturally will result in higher objective payoffs for higher values of c. We also observe that

even if $\lambda_{\max,t} = \lambda_{\min,t} = \lambda_c = 0$, $V(s_{c,t+1,1})$ would still be non-decreasing in c. There are two constraints which have c: $x^{\text{V2H}} + x^{\text{V2G}} \leq cs$ and $c - \sum_{a \in A} x^a \leq C_{\max}$. In $x^{\text{V2H}} + x^{\text{V2G}} \leq cs$, the feasible region is larger for the problem with charge level $c+1$ relative to the problem with charge level c. The final use of c is in constraint $c - \sum_{a \in A} x^a \leq C_{\max}$, which represents the battery's maximum capacity. Any solution for a problem at charge level c in which this constraint is active, means that the optimization is paying to charge up to maximum capacity. In such a case, charge level $c+1$ can achieve the same solution, but will pay less to reach maximum capacity because it will be starting at a higher charge. Therefore, this implies that $V(s_{c+1,t+1,1})$ is an upper bound on $V(s_{c,t+1,1})$, even if $\lambda_{\max,t} = \lambda_{\min,t} = \lambda = 0$.

For $p_{c,t,j} V(s_{c,t+1,1})$ to be non-decreasing in c means that

$$p_{c+1,t,j} V(s_{c+1,t+1,1}) - p_{c,t,j} V(s_{c,t+1,1}) \geq 0$$

. Let $x^*_{c,t,1}$ be the optimal solution to Formulation (2.6) that achieves value $V(s_{c,t,1})$.

$$
\begin{aligned}
& p_{c+1,t,j} V(s_{c+1,t+1,1}) - p_{c,t,j} V(s_{c,t+1,1}) \\
& = p_{c+1,t,j} \left(\sum_{a \in A} x^{*a}_{c+1,t,1} (w^a_t + \lambda_R r_t) - \lambda_{\max,t} (C_{\max} - c - 1) - \lambda_{\min,t} (C_{\min} - c - 1) \right. \\
& \left. + \frac{1}{\lambda_{\text{time},t}} E \left[V_1 \left(s_{c+1-\sum_{a \in A} x^{*a}_{c+1,t,1}, t+1} \right) \right] + \lambda_c \left(c + 1 - \sum_{a \in A} x^{*a}_{c+1,t,1} \right) \right) \\
& - p_{c,t,j} \left(\sum_{a \in A} x^{*a}_{c,t,1} (w^a_t + \lambda_R r_t) - \lambda_{\max,t} (C_{\max} - c) - \lambda_{\min,t} (C_{\min} - c) \right. \\
& \left. + \frac{1}{\lambda_{\text{time},t}} E \left[V_1 \left(s_{c-\sum_{a \in A} x^{*a}_{c,t+1,1}, t+1} \right) \right] + \lambda_c \left(c - \sum_{a \in A} x^{*a}_{c,t,1} \right) \right) \\
& \geq p_{c+1,t,j} \left(\sum_{a \in A} x^{*a}_{c+1,t,1} (w^a_t + \lambda_R r_t - \lambda_c) - \lambda_{\max,t} (C_{\max} - c - 1) - \lambda_{\min,t} (C_{\min} - c - 1) \right. \\
& \left. + \frac{1}{\lambda_{\text{time},t}} E \left[V_1 \left(s_{c+1-\sum_{a \in A} x^{*a}_{c+1,t,1}, t+1} \right) \right] + \lambda_c (c+1) \right) \\
& - p_{c,t,j} \left(\sum_{a \in A} x^{*a}_{c,t,1} (w^a_t + \lambda_R r_t - \lambda_c) - \lambda_{\max,t} (C_{\max} - c) - \lambda_{\min,t} (C_{\min} - c) \right. \\
& \left. + \frac{1}{\lambda_{\text{time},t}} E \left[V_1 \left(s_{c-\sum_{a \in A} x^{*a}_{c,t+1,1}, t+1} \right) \right] + \lambda_c (c) \right) \\
& \geq p_{c+1,t,j} \left(\sum_{a \in A} x^{*a}_{c,t,1} (w^a_t + \lambda_R r_t - \lambda_c) - \lambda_{\max,t} (C_{\max} - c - 1) - \lambda_{\min,t} (C_{\min} - c - 1)^+ \right. \\
& \left. + \frac{1}{\lambda_{\text{time},t}} E \left[V_1 \left(s_{c-\sum_{a \in A} x^{*a}_{c,t,1}, t+1} \right) \right] + \lambda_c (c+1) \right) \\
& - p_{c,t,j} \left(\sum_{a \in A} x^{*a}_{c,t,1} (w^a_t + \lambda_R r_t - \lambda_c) - \lambda_{\max,t} (C_{\max} - c) - \lambda_{\min,t} (C_{\min} - c)^+ \right.
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right] + \lambda_c(c)\Bigg) \\
&= (p_{c+1,t,j} - p_{c,t,j})\left(\sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) - \lambda_{\min,t}\left(C_{\min} - c\right)^+\right. \\
&\left.+\lambda_c(c) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right]\right) + p_{c+1,t,j}\left(\lambda_{\max,t} + \lambda_{\min,t} + \lambda_c\right) \\
&\geq (p_{c+1,t,j} - p_{c,t,j})\left(\sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) - \lambda_{\min,t}\left(C_{\min} - c\right)^+\right. \\
&\left.+\lambda_c(c) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right]\right) + p_{c+1,t,j}\left(\frac{p_{c,t,1} - p_{c+1,t,1}}{p_{c+1,t,1}} V^*_{c,t}\right) \\
&= (p_{c,t,j} - p_{c+1,t,j})\left(V^*_{c,t} - \left(\sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right)\right.\right. \\
&\left.\left.-\lambda_{\min,t}\left(C_{\min} - c\right)^+ + \lambda_c(c) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right]\right)\right) \\
&= (p_{c,t,j} - p_{c+1,t,j})\left(X^{\text{V2X}}_{\max}\left(\max\left(w^{\text{V2G}}_t, w^{\text{V2H}}_t\right) + \lambda_R r_t - \lambda_c\right)\right. \\
&+\frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{\min\left(c+X^{\text{charge}}_{\max},100\right),t+1}\right)\right]\left(\sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right)\right. \\
&\left.\left.-\lambda_{\min,t}\left(C_{\min} - c\right)^+ + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right]\right)\right) \\
&\geq 0
\end{aligned}
$$

The first equality follows from the definition of $V\left(s_{c,t+1,1}\right)$. The first inequality comes from the observation that because the $V\left(s_{c+1,t+1,1}\right)$ is an upper bound for $V\left(s_{c,t+1,1}\right)$ (regardless of the λ-penalties), then

$$
\begin{aligned}
&\sum_{a\in A} x^{*a}_{c+1,t,1}\left(w^a_t + \lambda_R r_t\right) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c+1-\sum_{a\in A} x^{*a}_{c+1,t,1},t+1}\right)\right] + \lambda_c\left(c + 1 - \sum_{a\in A} x^{*a}_{c+1,t,1}\right) \\
&\geq \sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t\right) + \frac{1}{\lambda_{\text{time},t}} E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t,1},t+1}\right)\right] + \lambda_c\left(c + 1 - \sum_{a\in A} x^{*a}_{c,t,1}\right).
\end{aligned}
$$

The second inequality comes from Theorem 2.4 as $\lambda_{\max,t} \geq \frac{p_{c,t,1} - p_{c+1,t,1}}{p_{c,t,1}} V^*_{c,t}$ and the definition of $\lambda_{\min,t} \geq 0$ and $\lambda_c \geq 0$. The final inequality comes from the observation that, $X^{\text{V2X}}_{\max}\left(\max\left(w^{\text{V2G}}_t, w^{\text{V2H}}_t\right) + \lambda_R r_t - \lambda_c\right)$ is an upper bound on $\sum_{a\in A} x^{*a}_{c,t,1}\left(w^a_t + \lambda_R r_t - \lambda_c\right)$, $E\left[V_1\left(s_{\min\left(c+X^{\text{charge}}_{\max},100\right),t+1}\right)\right]$ is an upper bound on $E\left[V_1\left(s_{c-\sum_{a\in A} x^{*a}_{c,t+1,1},t+1}\right)\right]$ and that $p_{c,t,j}$ is decreasing in c.

Step 1.2 $\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$ **is non-decreasing**

We observe that both $\left(1 - p_{c,t,j}\right)$ and $V\left(s_{c,t+1,0}\right)$ are non-negative and by Assumption 1,

$(1-p_{c,t,j})$ is non-decreasing. Therefore, all that is needed for $(1-p_{c,t,j})\,V\left(s_{c,t+1,0}\right)$ to be non-decreasing, is that $V\left(s_{c,t+1,0}\right)$ is non-decreasing.

Let us define function $g_t(\gamma) = \frac{1}{\lambda_{\text{time},t}} E\left[V_0\left(s_{c-\gamma,t+1}\right)\right] + \beta_t\gamma$. Then we can write the value function as, $V\left(s_{c,t+1,0}\right) = \sum_{\gamma=0}^{\gamma_{\max}} q_{\gamma,c,t+1}\left(\frac{1}{\lambda_{\text{time},t}} E\left[V_0\left(_{c-\gamma,t+2}\right)\right] + \beta_{t+1}\gamma\right) = E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right]$. For clarification we note that $E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right]$ is an expectation over the discharge usage while the car is away, while $E\left[V_0\left(s_{c-\gamma,t+2}\right)\right]$ is the expectation over whether the car will be at the grid or away at the next time step.

From Theorem 2.4,

$$\beta_{t+1} \geq \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_0\left(s_{c+1,t+2}\right)\right] - E\left[V_0\left(s_{c,t+2}\right)\right]\right), \quad \forall c \in [0,\ldots,C_{\max}-1]$$

, which makes $g_{t+1}(\gamma)$ an increasing function. Thus $V\left(s_{c,t+1,0}\right)$ is the expectation of an increasing function on a random variable, $\Gamma_{c,t+1}$, the car's discharge during the time period. From [28], we have if random variable X_{c+1} first order stochastic dominates X_c, then for any increasing function $f(x)$, $E\left[f(X_{c+1})\right] \geq E\left[f(X_c)\right]$. From Assumption 3, we have that $\Gamma_{c+1,t+1}$ dominates $\Gamma_{c,t+1}$, therefore we have that $E_\gamma\left[g_{t+1}(\Gamma_{c+1,t+1})\right] \geq E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right]$, which by definition means that $V\left(s_{c+1,t+1,0}\right) \geq V\left(s_{c,t+1,0}\right)$.

Step 1.3 $p_{c,t,j}V\left(s_{c,t+1,1}\right)$ **is concave**

We start by proving that $V\left(s_{c,t+1,1}\right)$ is a concave function. We do this by characterizing the optimal solution to Formulation (2.6), and then by showing the objective function of these solutions is concave in state c. We start by defining the optimal solution, then proving its optimality and then demonstrating concavity of the objective function.

Optimal Solution: We observe that we will never engage in both V2G and V2H during the same time period. Let us assume first that $w_{t+1}^{\text{V2G}} \geq w_{t+1}^{\text{V2H}}$. A solution that has both $x^{\text{V2G}} > 0$ and $x^{\text{V2H}} > 0$ will always be less than or equal to a policy that only engages in x^{V2G}. This is because x^{V2G} and x^{V2H} are subject to the same constraints, so any allocation to x^{V2H} can instead be assigned to x^{V2G} for an improved solution in terms of the objective. The reverse holds true for if $w_{t+1}^{\text{V2H}} \geq w_{t+1}^{\text{V2G}}$. Therefore, for simplicity we will refer to x_{t+1}^{V2X} as the allocation to either V2G or V2H in time period $t+1$ and $w_{t+1}^{\text{V2X}} = \max\left(w_{t+1}^{\text{V2HG}}, w_{t+1}^{\text{V2H}}\right)$.

First calculate $c^{*\text{charge}}$ and $c^{*\text{V2X}}$. These represent the threshold above or below which it is optimal to either charge or discharge. $c^{*\text{charge}}$ is defined the charge level such that:

$$w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c > \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{charge}}+1,t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right]\right)$$

and

$$w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c \leq \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{charge}}-1,t+2}\right)\right]\right).$$

Similarly, we calculate $c^{*\text{V2X}}$ such that:

$$w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} - \lambda_c \geq \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{V2X}}+1,t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{V2X}},t+2}\right)\right]\right)$$

and

$$w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} - \lambda_c < \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{V2X}},t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{V2X}}-1,t+2}\right)\right]\right).$$

We observe that as long as $\lambda_c \geq w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} - \frac{1}{\lambda_{\text{time},t}} \left(E\left[V_1\left(s_{C_{\min},t+2}\right)\right] - E\left[V_1\left(s_{C_{\min}-1,t+2}\right)\right]\right)$, it will always be optimal for $c^{*\text{V2X}} > C_{\min}$, which will guarantee that the constraint $C_{\min} r \leq c - \sum_{a\in A} x^a$ holds.

Due to the fact that $E\left[V_1\left(s_{c,t+2}\right)\right]$ is concave, namely its intervals, $E\left[V_1\left(s_{c+1,t+2}\right)\right] - E\left[V_1\left(s_{c,t+2}\right)\right]$, is non-increasing, there will only be one charge level that satisfies each of these conditions.

If $c^{*\text{charge}} \leq c^{*\text{V2X}}$, which will occur if $w_{t+1}^{\text{charge}} \geq \max\left(w_{t+1}^{\text{V2HG}}, w_{t+1}^{\text{V2H}}\right)$, then the optimal decision is:

$$\begin{aligned}
&x_{c,t+1,1}^{*charge} = -\min\left(c^{*\text{charge}} - c, X_{\max}^{\text{charge}}\right), \quad x_{c,t+1,1}^{*V2X} = 0, && \text{if } c < c^{*\text{charge}} \\
&x_{c,t+1,1}^{*charge} = 0, \quad x_{c,t+1,1}^{*V2X} = 0, && \text{if } c^{*\text{charge}} \leq c \leq c^{*\text{V2X}} \\
&x_{c,t+1,1}^{*charge} = 0, \quad x_{c,t+1,1}^{*V2X} = \min\left(c - c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right), && \text{if } c^{*\text{V2X}} < c
\end{aligned} \tag{2.8}$$

If $c^{*\text{charge}} > c^{*\text{V2X}}$, then we first calculate, $c^{*\text{switch}}$ as the minimum value of c between $c^{*\text{V2X}}$ and $c^{*\text{charge}}$ such that:

$$-\min\left(c^{*\text{charge}} - c, X_{\max}^{\text{charge}}\right)\left(w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c+\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right),t+2}\right)\right] \tag{2.9}$$

$$\leq \min\left(c - c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right)\left(w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} - \lambda_c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c-\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right),t+2}\right)\right] \tag{2.10}$$

Equation (2.9) represents the optimal payoff for charging at c while equation (2.10) is the optimal payoff for discharging. $c^{*\text{switch}}$ is the first time the policy for discharging has a payoff higher than the policy for charging.

Then we define the optimal decision as:

$$\begin{aligned}
&x_{c,t+1,1}^{*charge} = -\min\left(c^{*\text{charge}} - c, X_{\max}^{\text{charge}}\right), \quad x_{c,t+1,1}^{*V2X} = 0, && \text{if } c < c^{*\text{switch}} \\
&x_{c,t+1,1}^{*charge} = 0, \quad x_{c,t+1,1}^{*V2X} = \min\left(c - c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right), && \text{if } c^{*\text{switch}} \geq c
\end{aligned} \tag{2.11}$$

Proof of Optimality: Please see Appendix A.1.

Concavity of the Optimal Objective Value in terms of the Charge Level: Given that we have the optimal solution of the optimization, we can then derive the objective value of the solution, $V\left(s_{c,t+1,1}\right)$ and then its first derivative. We start by calculating $V\left(s_{c,t+1,1}\right)$ based on c by directly plugging the solution into the objective value.

We first consider the case of $c^{*\text{charge}} \leq c^{*\text{V2X}}$, for which we calculate the value of the objective function, $V\left(s_{c,t+1,1}\right)$. For the sake of completeness, we assume that $C_{\min} = 0$, as the most conservative value the constant can take:

$$\begin{aligned}
&-X_{\max}^{\text{charge}}\left(w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c+X_{\max}^{\text{charge}},t+2}\right)\right] + \lambda_c c, && \text{if } c < c^{*\text{charge}} - X_{\max}^{\text{charge}} \\
&-\left(c^{*\text{charge}} - c\right)\left(w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right] + \lambda_c c, && \text{if } c^{*\text{charge}} - X_{\max}^{\text{charge}} \leq c < c^{*\text{charge}} \\
&-\lambda_{\max,t}\left(C_{\max} - c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c,t+2}\right)\right] + \lambda_c c, && \text{if } c^{*\text{charge}} \leq c \leq c^{*\text{V2X}} \\
&\left(c - c^{*\text{V2X}}\right)\left(w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c^{*\text{V2X}},t+2}\right)\right] + \lambda_c\left(c\right), && \text{if } c^{*\text{V2X}} < c \leq c^{*\text{V2X}} + X_{\max}^{\text{V2X}} \\
&X_{\max}^{\text{V2X}}\left(w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} - \lambda_c\right) - \lambda_{\max,t}\left(C_{\max} - c\right) + \frac{1}{\lambda_{\text{time},t+1}} E\left[V_1\left(s_{c-X_{\max}^{\text{V2X}},t+2}\right)\right] + \lambda_c\left(c\right), && \text{if } c^{*\text{V2X}} + X_{\max}^{\text{V2X}} < c
\end{aligned}$$

We can observe the following relationship $V\left(s_{c,t+1,1}\right) - V\left(s_{c-1,t+1,1}\right)$ as c increases:

$$
\begin{aligned}
&\lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c+X_{\max}^{\text{charge}},t+2}\right)\right] - E\left[V_1\left(s_{c-1+X_{\max}^{\text{charge}},t+2}\right)\right]\right) \quad \text{if } c \leq c^{*\text{charge}} - X_{\max}^{\text{charge}} \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{charge}}-1,t+2}\right)\right]\right(\\
&\geq \lambda_{\max,t} + w_{t+1}^{\text{charge}} + \lambda_R r_{t+1} \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{charge}}+1,t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right]\right) \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c,t+2}\right)\right] - E\left[V_1\left(s_{c-1,t+2}\right)\right]\right) \qquad \text{if } c^{*\text{charge}} < c \leq c^{*\text{V2X}} \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{V2X}},t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{V2X}}-1,t+2}\right)\right]\right) \\
&\geq \lambda_{\max,t} + w_{t+1}^{\text{V2X}} + \lambda_R r_{t+1} \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{V2X}}+1,t+2}\right)\right] - E\left[V_1\left(s_{c^{*\text{V2X}},t+2}\right)\right]\right) \\
&\geq \lambda_{\max,t} + \lambda_c + \frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c-X_{\max}^{\text{V2X}},t+2}\right)\right] - E\left[V_1\left(s_{c-X_{\max}^{\text{V2X}},t+2}\right)\right]\right) \qquad \text{if } c^{*\text{V2X}} + X_{\max}^{\text{V2X}} < c
\end{aligned}
$$

Because $V\left(s_{c,t+1,1}\right) - V\left(s_{c-1,t+1,1}\right)$ is non-increasing, we have concavity for $V\left(s_{c,t+1,1}\right)$ if $c^{*\text{charge}} \leq c^{*\text{V2X}}$. The extension to the case of $c^{*\text{charge}} > c^{*\text{V2X}}$ is straightforward as the solutions follow the same structure as the $c^{*\text{charge}} \leq c^{*\text{V2X}}$ case.

We now prove that $p_{c,t,j}V\left(s_{c,t+1,1}\right)$ is concave. Once again we do this by showing that $2p_{c,t,j}V\left(s_{c,t+1,1}\right) - p_{c-1,t,j}V\left(s_{c-1,t+1,1}\right) - p_{c+1,t,j}V\left(s_{c+1,t+1,1}\right) \geq 0$.

$$
\begin{aligned}
&2p_{c,t,j}V\left(s_{c,t+1,1}\right) - p_{c-1,t,j}V\left(s_{c-1,t+1,1}\right) - p_{c+1,t,j}V\left(s_{c+1,t+1,1}\right) \\
&= p_{c,t,j}\left(2V\left(s_{c,t+1,1}\right) - V\left(s_{c-1,t+1,1}\right) - V\left(s_{c+1,t+1,1}\right)\right) + p_{c,t,j}V\left(s_{c-1,t+1,1}\right) + p_{c,t,j}V\left(s_{c+1,t+1,1}\right) \\
&- p_{c-1,t,j}V\left(s_{c-1,t+1,1}\right) - p_{c+1,t,j}V\left(s_{c+1,t+1,1}\right) \\
&\geq V\left(s_{c,t+1,1}\right)\left(-2p_{c,t,j} + p_{c+1,t,j} + p_{c-1,t,j}\right)\right) + p_{c,t,j}V\left(s_{c-1,t+1,1}\right) + p_{c,t,j}V\left(s_{c+1,t+1,1}\right) \\
&- p_{c-1,t,j}V\left(s_{c-1,t+1,1}\right) - p_{c+1,t,j}V\left(s_{c+1,t+1,1}\right) \\
&= V\left(s_{c,t+1,1}\right)\left(-2p_{c,t,j} + p_{c+1,t,j} + p_{c-1,t,j}\right)\right) + V\left(s_{c-1,t+1,1}\right)\left(p_{c,t,j} - p_{c-1,t,j}\right) \\
&+ V\left(s_{c+1,t+1,1}\right)\left(p_{c,t,j} - p_{c+1,t,j}\right) \\
&= \left(V\left(s_{c,t+1,1}\right) - V\left(s_{c-1,t+1,1}\right)\right)\left(p_{c-1,t,j} - p_{c,t,j}\right) + \left(V\left(s_{c+1,t+1,1}\right) - V\left(s_{c,t+1,1}\right)\right)\left(p_{c,t,j} - p_{c+1,t,j}\right) \\
&\geq 0
\end{aligned}
$$

The first inequality comes from Assumption 1 on the relationship between the second derivatives of $p_{c,t,j}$ and $V\left(s_{c,t+1,1}\right)$. The second inequality comes from the fact that we assume that $p_{c,t,j}$ is non-increasing and prove that $V\left(s_{c,t+1,1}\right)$ is non-decreasing.

Step 1.4 $\left(1 - p_{c,t,j}\right)V\left(s_{c,t+1,0}\right)$ **is concave**

We start by proving that $V\left(s_{c,t+1,0}\right)$ is concave. To do this we first establish that if a set of random variables X_c are concave stochastic dominant, which means that X_{c+1} first-order stochastic dominates X_c for all c in the set and $P\left(X_{c+1} > x\right) - P\left(X_c > x\right) \leq P\left(X_c > x\right) - P\left(X_{c-1} > x\right)$ for all c and x, then any increasing function, $f(x)$ applied to X_c will have the property $E\left[f\left(X_{c+1}\right)\right] - E\left[f\left(X_c\right)\right] \leq E\left[f\left(X_c\right)\right] - E\left[f\left(X_{c-1}\right)\right]$. This follows very close to how

[28] established that if X first order stochastic dominates Y, then for an increasing function f it holds that $E[f(X)] \geq E[f(Y)]$.

We can rewrite $P\left(X_{c+1} > x\right) - P\left(X_c > x\right) \leq P\left(X_c > x\right) - P\left(X_{c-1} > x\right)$ as:

$$E\left[1_U\left(X_{c+1}\right)\right] - E\left[1_U\left(X_c\right)\right] \leq E\left[1_U\left(X_c\right)\right] - E\left[1_U\left(X_{c-1}\right)\right] \qquad \text{for all } U \subseteq (-\infty, \infty)$$

where 1 is the indicator function and U represents an upper set.

If this holds, then it follows that:

$$\begin{aligned}
&\left(E\left[\sum_{i=1}^{m} a_i 1_{U_i}\left(X_{c+1}\right)\right] - b\right) - \left(E\left[\sum_{i=1}^{m} a_i 1_{U_i}\left(X_c\right)\right] - b\right) \\
&\leq \left(E\left[\sum_{i=1}^{m} a_i 1_{U_i}\left(X_c\right)\right] - b\right) - \left(E\left[\sum_{i=1}^{m} a_i 1_{U_i}\left(X_{c-1}\right)\right] - b\right)
\end{aligned}$$

for all $a_i \geq 0, i = 1, 2, \ldots, m$, $b \in (-\infty, \infty)$ and $m \geq 0$. For any increasing function, $f(x)$ we can define a_i, U_i and b such that as m goes to infinity, we have that $E\left[\sum_{i=1}^{m} a_i 1_{U_i}\left(X\right)\right]$ converges to $E\left[f(X)\right]$. Therefore we have that

$$E\left[f\left(X_{c+1}\right)\right] - E\left[f\left(X_c\right)\right] \leq E\left[f\left(X_c\right)\right] - E\left[f\left(X_{c-1}\right)\right].$$

In 1.2, when we establish that $V\left(s_{c,t+1,0}\right)$ is increasing, we establish

$$V\left(s_{c,t+1,0}\right) = E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right],$$

where $g_{t+1}(\gamma)$ is an increasing function. Since $g_{t+1}(\gamma)$ is an increasing function and that the driver's car usage, $\Gamma_{c,t+1}$ is concave stochastic dominant as defined above, we have that

$$E_\gamma\left[g_{t+1}(\Gamma_{c+1,t+1})\right] - E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right] \leq E_\gamma\left[g_{t+1}(\Gamma_{c,t+1})\right] - E_\gamma\left[g_{t+1}(\Gamma_{c-1,t+1})\right]$$

which by definition makes $V\left(s_{c,t+1,0}\right)$ concave in c.

We now establish that $\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right)$ is concave.

$$\begin{aligned}
&2\left(1 - p_{c,t,j}\right) V\left(s_{c,t+1,0}\right) - \left(1 - p_{c-1,t,j}\right) V\left(s_{c-1,t+1,0}\right) - \left(1 - p_{c+1,t,j}\right) V\left(s_{c+1,t+1,0}\right) \\
&= V\left(s_{c,t+1,0}\right)\left(2\left(1 - p_{c,t,j}\right) - \left(1 - p_{c-1,t,j}\right) - \left(1 - p_{c+1,t,j}\right)\right) + \left(1 - p_{c-1,t,j}\right) V\left(s_{c,t+1,0}\right) \\
&+ \left(1 - p_{c+1,t,j}\right) V\left(s_{c,t+1,0}\right) - \left(1 - p_{c-1,t,j}\right) V\left(s_{c-1,t+1,0}\right) - \left(1 - p_{c+1,t,j}\right) V\left(s_{c+1,t+1,0}\right) \\
&= V\left(s_{c,t+1,0}\right)\left(-2p_{c,t,j} + p_{c-1,t,j} + p_{c+1,t,j}\right) + \left(1 - p_{c-1,t,j}\right)\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c-1,t+1,0}\right)\right) \\
&+ \left(1 - p_{c+1,t,j}\right)\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c+1,t+1,0}\right)\right) \\
&\geq \left(1 - p_{c,t,j}\right)\left(-2V\left(s_{c,t+1,0}\right) + V\left(s_{c+1,t+1,0}\right) + V\left(s_{c-1,t+1,0}\right)\right) \\
&+ 2\left(V\left(s_{c+1,t+1,0}\right) - V\left(s_{c,t+1,0}\right)\right)\left(p_{c,t,j} - p_{c+1,t,j}\right) + 2\left(V\left(s_{c,t+1,0}\right)\right. \\
&\left.-V\left(s_{c-1,t+1,0}\right)\right)\left(p_{c-1,t,j} - p_{c,t,j}\right) + \left(1 - p_{c-1,t,j}\right)\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c-1,t+1,0}\right)\right) \\
&+ \left(1 - p_{c+1,t,j}\right)\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c+1,t+1,0}\right)\right) \\
&= -p_{c,t,j}\left(-2V\left(s_{c,t+1,0}\right) + V\left(s_{c+1,t+1,0}\right) + V\left(s_{c-1,t+1,0}\right)\right) \\
&+ 2\left(V\left(s_{c+1,t+1,0}\right) - V\left(s_{c,t+1,0}\right)\right)\left(p_{c,t,j} - p_{c+1,t,j}\right) + 2\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c-1,t+1,0}\right)\right)\left(p_{c-1,t,j}\right. \\
&\left.-p_{c,t,j}\right) - p_{c-1,t,j}\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c-1,t+1,0}\right)\right) - p_{c+1,t,j}\left(V\left(s_{c,t+1,0}\right) - V\left(s_{c+1,t+1,0}\right)\right) \\
&= 0
\end{aligned}$$

With this we have that $p_{c,t,j}V\left(s_{c,t+1,1}\right)$ and $\left(1-p_{c,t,j}\right)V\left(s_{c,t+1,0}\right)$ are both non-decreasing and concave. The sum of two non-decreasing and concave functions, maintain these two properties. Therefore, we have established that $E\left[V_j\left(s_{c,t}\right)\right]$ is non-decreasing and concave.

We can solve the DP in closed form. The optimal solution to the DP is monotonic in terms of the charge level c.

We prove this inductively. The solution to the DP, given that in the next time step, we assume by induction that the solution is non-decreasing and concave implies the following: If $c^{*\text{charge}} \leq c^{*\text{V2X}}$, then the optimal decision is:

$$\begin{array}{lll} x_{c,t+1,1}^{*charge} = -\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right), & x_{c,t+1,1}^{*V2X}=0, & \text{if } c < c^{*\text{charge}} \\ x_{c,t+1,1}^{*charge}=0, & x_{c,t+1,1}^{*V2X}=0, & \text{if } c^{*\text{charge}} \leq c \leq c^{*\text{V2X}} \\ x_{c,t+1,1}^{*charge}=0, & x_{c,t+1,1}^{*V2X}=\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right), & \text{if } c^{*\text{V2X}} < c \end{array}$$

If $c^{*\text{charge}} > c^{*\text{V2X}}$, then we define the optimal decision as:

$$\begin{array}{lll} x_{c,t+1,1}^{*charge} = -\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right), & x_{c,t+1,1}^{*V2X}=0, & \text{if } c < c^{*\text{switch}} \\ x_{c,t+1,1}^{*charge}=0, & x_{c,t+1,1}^{*V2X}=\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right), & \text{if } c^{*\text{switch}} \geq c \end{array}$$

We prove the optimality of this solution in Part 1.3 of Step 2.4. We observe that as c increases in these solutions, $x_{c,t+1,1}^{*charge}$ is strictly non-increasing and $x_{c,t+1,1}^{*V2X}$ is strictly non-decreasing, guaranteeing monotonic behavior.

We observe that the structure of the solutions the DP gives rise to make them easy to communicate to drivers and also as a result, easier to be trusted. These policies have a two-threshold structure to them. If the driver is below a minimum level (determined by either $c^{*\text{charge}}$ or $c^{*\text{switch}}$) then the car is charged up to the threshold. If the driver is a above a sufficient level (determined by $c^{*\text{switch}}$ or $c^{*\text{V2X}}$), then the car is discharged down to the threshold. These thresholds can be explained to customers that are interested in understanding when and why their car is being charged or discharged. They are easily trusted not only due to their simplicity but also because their value can be loosely justified in terms of the price of charging/discharging and the driver's own behavior. Furthermore, this structure of policy opens the markets to opportunities for interaction where drivers could be provided calculated thresholds and given flexibility to readjust them based on their own preferences. These changes can then be reincorporated into the system for future interactions. We have a closed form relationship between the thresholds, $c^{*\text{charge}}$, $c^{*\text{V2X}}$ or $c^{*\text{switch}}$, and the parameters of the optimization. If the driver prefers a different threshold, we can recalculate $\lambda_{\max}$ and λ_c to make their preferred threshold optimal, or if the required value for $\lambda_{\max}$ is outside the regime where policies are monotonic, we know we need to relearn probabilities around the driver's behavior.

In using such a structure we aim to provide drivers with clearly explained policies that they can trust. Often times we think that simple, interpretable policies that can be easily understood by consumers require the optimality of the solution to be compromised. However in this case we find that by ensuring that the value function of the next states is concave, a necessary property for the optimization to serve driver behavior, we are also able to achieve a simple policy that drivers can buy into. By making sure that the optimization is tuned to drivers' sensitivity around charge level at departure and driving utility, we are able to achieve simple optimal solutions.

Finally, computing the solution in closed form makes sense once we consider the fact that for a given decision (either to charge or discharge) the objective function becomes concave. It is simple to calculate the optimal decision based on the feasible region and the peak of the concave objective function, allowing us to solve the DP in a scalable manner.

2.5 Computational Results

In this section we discuss how the results in this chapter apply in practice through our close collaboration with a large American EV manufacturer. We first discuss the data we use on driver behavior, centered in a major American city during January and February 2021, the resulting policies of the DP and the implications for larger profit and carbon benefit.

In order to learn driver data we start from deidentified telemetry data gathered from the EV manufacturer's electric vehicles. The telemetry data records the approximate location of the car along with what mode it is in (driving, parked, charging, etc) and battery charge level. We use this to first calculate each vehicle's most common parking location, which we consider their main charging location and the place where we have opportunity to engage with charging/discharging optimization. We limit the dataset down to vehicles that made trips in January and February 2021. For this dataset, for each driver and every hour in the dataset we calculate the distance the driver drove during that hour and the number of minutes they spent at their most common parking location. After long discussions with experts at the EV manufacturer, we consider the time a driver spends at their most common parking location as their on-grid time, regardless of if the driver is actually charging during this time. In doing so, we aim to capture total time that they could be charging/discharging under a V2X pilot program rather than their current charging behavior.

The dataset has 48 columns, 24 representing the driver's average distance driven for each hour of the day and 24 for the driver's average on grid minutes for each hour. As a result we use a time step of an hour for all modelling. This dataset is then clustered into eight groups with KNN. We did sensitivity testing around the number of clusters to use, but eight consistently had the strongest performance in terms of silhouette score, as shown in Figure 2.1.

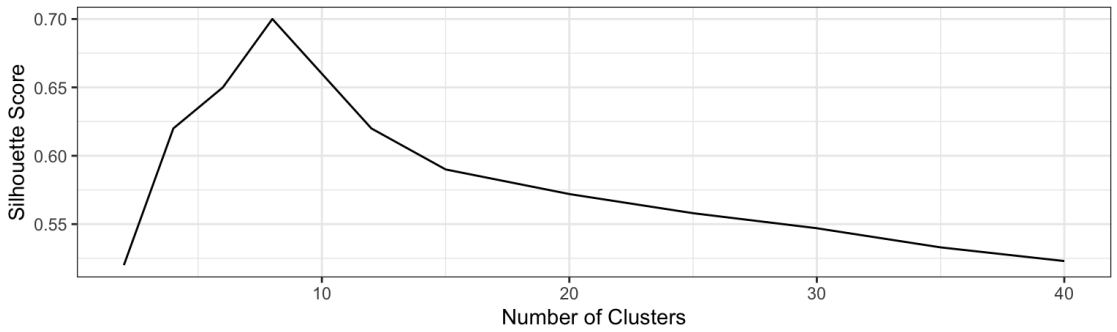


Figure 2.1: Silhouette score testing for number of clusters

These eight clusters are summarized in Figure 2.2, in which we plot for each cluster its average distance driven and minutes spent on the grid across the hours in a day. We also

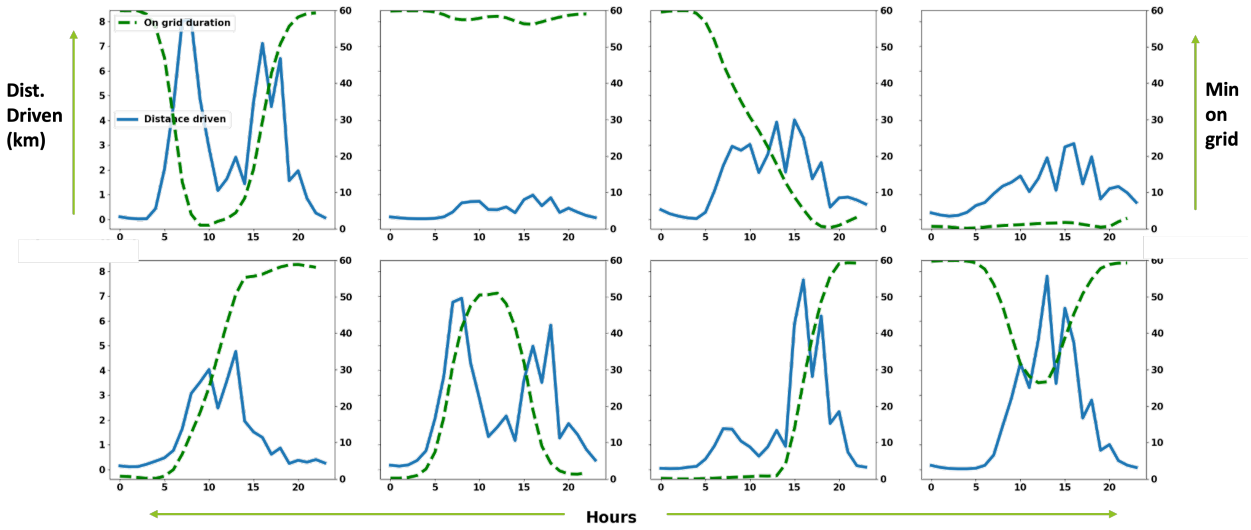


Figure 2.2: Driver clusters: Average distance driven and minutes spent on grid over the day

characterize each cluster as follows, working left to right, top to bottom:

- **Cluster 1, Commuters:** This represents 3% of the data. The drivers in this cluster park at the beginning of the day and at the end, while when they are away seem to drive mostly in the morning and in the afternoon, with a lull during normal working hours
- **Cluster 2, Parkers:** This represents 25% of the data. The drivers in this cluster have very little driving and high on-grid minutes implying they on-average use their car sparingly, such as might be seen with work from home drivers
- **Cluster 3, Morning Parkers:** This represents 16% of the data. The drivers in this cluster have normal during day driving, but tend to park their car at their base in the morning, while having inconsistent evening parking
- **Cluster 4, Away Parkers:** This represents 15% of the data. The drivers in this cluster tend to park their car inconsistently in the same location, such as might be seen with street parkers
- **Cluster 5, Afternoon Parkers:** This represents 5% of the data. The drivers in this cluster have normal during day driving, but tend to park their car at their base in the evening while having inconsistent morning parking
- **Cluster 6, Nightime or at-Work Commuters:**. This represents 9% of the data. The drivers in this cluster exhibit similar driving behavior to commuters with peaks in the morning and afternoon, but tend to park most consistently during working hours, as might be seen with nightime workers or street parkers who commute to work
- **Cluster 7, Evening Parkers:** This represents 7% of the data. The drivers in this cluster show similar behavior to the afternoon parkers (Cluster 5) but with a stronger driving usage peak in the evening and consistent parking only after 8PM

- **Cluster 8, Light Users:**. This represents 10% of the data. The drivers in this cluster exhibit normal day driving and have high on-grid minutes both in the morning and evening.

We use these clusters to derive transition probabilities for our proposed DP formulation. The distribution of kilometers driven are translated directly to charge dispensed to learn the distribution of usage when the car is away, namely to calculate $q_{\gamma,c,t}$, the probability that the driver uses γ of its battery given the car starts away from the grid at time t and charge c. The on grid minutes are used to calculate the probability the car leaves/returns to the grid. If the on grid minutes equals 60 for a given hour, then the driver is considered to have spent the hour on the grid. If it is less than 60, then the car is considered to have spent the hour away from the grid. This is a conservative estimate, meant to help create a lower bound on the potential capacity EVs have. We then learn the conditional probability for a given cluster using a linear regression, for $p_{c,t,j}$, the probability the driver remains or returns to the grid at charge c and time t, given that it started the time period t either attached to the grid $j = 1$ or away from the grid $j = 0$. We illustrate $p_{c,t,1}$ the probability driver remains attached to the grid, given they are currently attached to the grid, for each of the Clusters 1-8 (left to right, top to bottom) in Figure 2.3. We see that the probabilities uphold Assumptions 1 and 2 of Theorem 2.4. First there is a strict hierarchy, in the probabilities, as the charge level of the car increases, their probability of remaining on grid decreases. Second, for any given cluster and time step, the probability of remaining on grid has a linear spread, showing no second derivative. We also see that the probabilities logically reflect the driving patterns we see from Figure 2.2. For example, Cluster 1, Commuters, have decreasing probabilities of remaining on grid leading into 10AM, just as their on grid minutes are going down and driving distances are going up. For Cluster 2, Parkers, their probability of remaining on grid is higher than any other cluster, never dropping below 80%.

We get data on charging of EV pricing from PG&E, utility provider in California, on its time of use pricing. This assumes the EV pays the same amount to charge as the driver pays to run any appliance in their home. This time of use price is also used for the price of giving energy to the home, because this is what the driver is saving by running the home off of the car rather than buying energy from PG&E. We get data on V2G, for selling energy back to the grid based on CAISO wholesale energy price. This is the price that generators are paid to provide energy to the grid. In using this price, we assume that the driver is paid at the same rate as a generator, which is a lower bound on the actual price drivers can expect. It is worth noting that the wholesale price is very volatile. This volatility stems from the unpredictability of energy given at a specific time and can be leveraged to generate a profit. For example, the spike seen on February 17, 2021 at 3PM is not repeated every day, but most days reflect some level of price spikes depending on energy needs. Finally, we get the tons of CO2 produced per kWh in California from the EPA, which provides this data publicly. For the sake of this example, we show this pricing for February 17, 2021 in Figure 2.4.

We set the parameters of the DP as follows. First we consider a single day period, where the last hour of the day, 11PM to midnight is the final period of the DP. For the final period, we set the terminal payoffs as zero. We discretize the charge level of the battery into percentages with 2% intervales, making 0-2%, 2-4%, ..., 98-100% discrete charge level

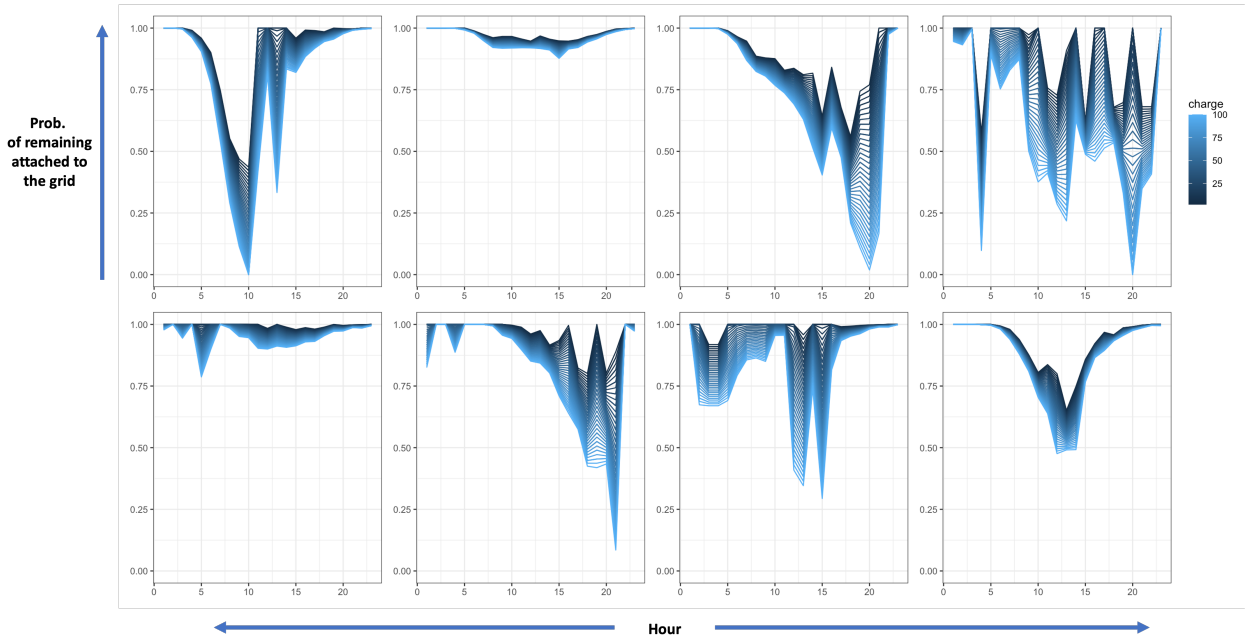


Figure 2.3: Driver clusters: Probability of remaining on grid in next hour given currently on grid

buckets. The maximum charge/discharge allowed for a given hour is 16%, which we learned as reasonable from the data. We set $C_{\min}$ as 40%. We assume the driver is purely a profit maximizer, making $\lambda_R = 0$. We penalize being below $C_{\min}$ by setting $\lambda_{\min} = 15$. Finally we scale the temporal constant with $\lambda_{\max}$, $\lambda_{\text{time},t} = 2.5\lambda_{\max}$, which we settled on after testing a few different parameters.

We first summarize the policy results across all clusters. We show the results in terms of the dollar value and carbon benefit earned in expectation by each cluster on February 17, 2021. In order to avoid accounting for the end of horizon sell-off, we only consider the dollar value and carbon benefit earned during 1AM-7PM. Table 2.1 shows the dollar value of the policy in expectation, while Table 2.2 shows the tons of CO2 saved/consumed from the policy. As expected, Cluster 2 has the highest monetary benefit for engaging with V2X, as it has the most on grid minutes, especially in the morning when the V2X prices spiked above charging. We also see that there is always a benefit to starting the day attached to the grid, rather than away, because then the optimization has the opportunity to use its policy. The biggest discrepancy between away and attached happens for Cluster 4, which makes sense because if the EV leaves the grid, it is unlikely to return. To contextualize the carbon benefit of tons of C02 consumed/saved, it is worth noting that, 0.01 tons of C02 is equivalent to 25.6 miles driven by a gasoline-powered passenger vehicle or 1.1 gallons of gas consumed.

For a more intuitive analysis of how these results are achieved, we plot the exact policy decisions from the DP for when the EV is attached to the grid for a few of the clusters, specifically Clusters 2, 4 and 6 to show a sampling of the results. The heatmaps are plotted in Figure 2.5, with the x-axis showing the hour of the day, and the y-axis showing the charge bucket with the top row being the 98-100% and the bottom row being 0-2%. A positive

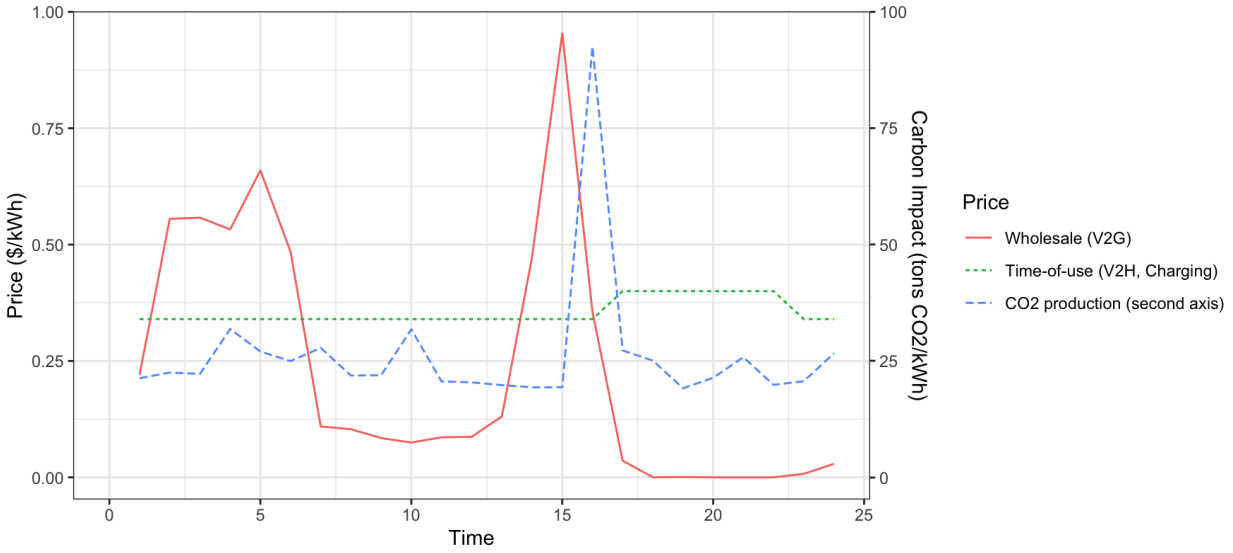


Figure 2.4: Pricing and CO2 production for February 17, 2021

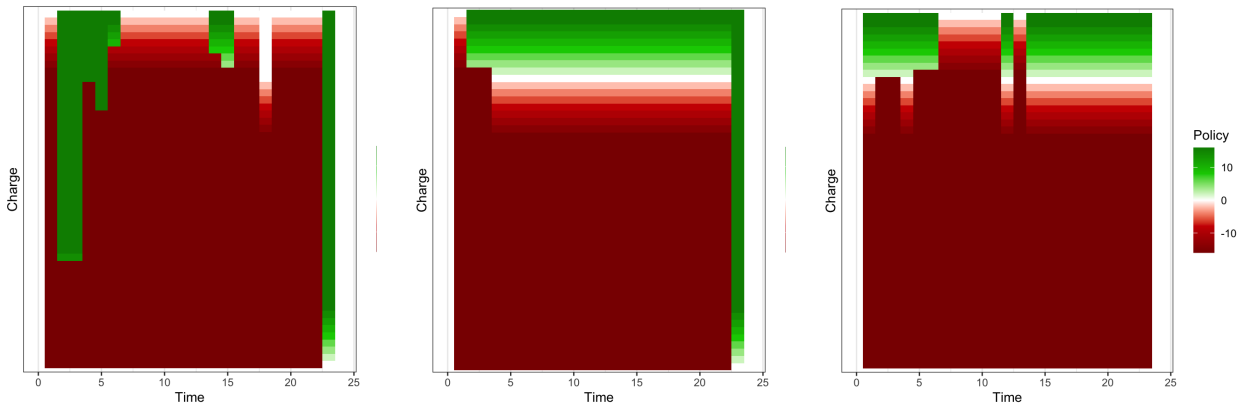
Starting Charge Level	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5		Cluster 6		Cluster 7		Cluster 8	
	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached
8-10%	-30.15	-28.79	-22.55	-22.40	-28.12	-28.02	-3.89	-23.43	-29.97	-21.04	-25.78	-16.55	-13.86	-29.15	-29.99	-30.01
18-20%	-26.72	-24.70	-19.23	-13.89	-25.07	-22.22	-3.57	-21.65	-27.58	-13.63	-24.07	-13.44	-13.14	-26.28	-26.70	-24.31
28-30%	-22.33	-19.77	-14.74	-10.49	-20.67	-17.33	-3.12	-18.20	-24.43	-10.59	-21.45	-8.98	-12.23	-23.40	-21.97	-20.91
38-40%	-18.85	-14.54	-8.94	-8.76	-16.89	-12.70	-2.63	-13.79	-21.02	-7.43	-18.42	-3.86	-11.32	-19.98	-17.86	-17.51
48-50%	-13.12	-11.32	-5.15	-3.06	-11.32	-9.30	-2.16	-8.80	-17.46	-4.20	-15.35	-0.96	-10.47	-16.29	-14.37	-11.41
58-60%	-9.85	-6.68	-2.42	3.15	-7.38	-4.29	-1.69	-5.26	-13.82	-0.98	-12.30	1.81	-9.58	-12.89	-11.00	-7.62
68-70%	-5.60	-0.85	4.29	6.55	-3.24	0.99	-1.23	-1.86	-10.08	2.36	-9.26	4.66	-8.49	-8.63	-5.43	-3.82
78-80%	-0.45	1.94	7.85	9.93	1.61	4.39	-0.76	3.89	-6.15	5.74	-6.19	7.53	-7.06	-3.07	-1.40	-0.42
88-90%	3.06	5.59	11.58	16.43	5.87	7.79	-0.28	7.11	-2.17	9.68	-3.10	10.63	-5.30	0.79	2.17	2.98
98-100%	7.67	8.99	16.61	19.83	9.62	11.19	0.17	10.51	1.87	13.08	0.01	14.03	-3.18	4.19	5.56	6.38

Table 2.1: Expected monetary benefit/cost to optimal policy for 1AM-7PM on February 17, 2021

policy value means that the EV is discharging the battery to earn money, with a maximum of 16% discharge. A policy value of 0 means that the policy is to do nothing. A negative policy value means that the EV is spending money to charge the battery, with a maximum of 16%. We note that policies are very different for each cluster. Cluster 2, is the parking drivers who have low probabilities of leaving the grid. Therefore, when V2G pricing spikes above the price to charge, we engage deeply in discharging the battery in order to earn money on selling energy back to the grid before charging back up. There is then an end of horizon sell-off in the final period. Cluster 4, is the Away Parkers, who have low probabilities of being on grid or remaining on grid. For these drivers, the policy recommends charging up to an appropriate charge level because the drivers have a high probability of leaving at any given hour. For Cluster 6, the policy is very similar to Cluster 4, however during 7AM to 1PM, when the car has high probability of being attached and not leaving, we focus on charging as much as possible to prepare for departure. In this way we see how the DP both serves driver behavior based on the distribution of the cluster, but also aims to help the driver earn through V2X when the opportunity is available. We also observe that all these polices are very interpretable. If drivers are interested, there can be communicated a clear charge level for each hour of the day for which the optimization will aim to charge/discharge

Starting Charge Level	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5		Cluster 6		Cluster 7		Cluster 8	
	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached	Away	Attached
8-10%	0.0070	0.0110	0.0111	0.0104	0.0071	0.0066	-0.0006	0.0006	-0.0010	0.0568	0.0728	0.1660	-0.0015	0.0017	0.0305	0.0303
18-20%	0.0026	0.0322	0.0112	0.0536	0.0073	0.0453	-0.0006	-0.0001	0.0009	0.0942	0.0746	0.1649	-0.0015	0.0013	0.0305	0.0700
28-30%	0.0177	0.0476	0.0306	0.0536	0.0241	0.0389	-0.0005	-0.0001	0.0020	0.0930	0.0774	0.1761	-0.0013	0.0010	0.0541	0.0700
38-40%	0.0300	0.0393	0.0426	0.0464	0.0236	0.0497	-0.0004	0.0178	0.0029	0.0923	0.0808	0.1949	-0.0010	0.0116	0.0667	0.0700
48-50%	0.0518	0.0792	0.0480	0.0860	0.0412	0.0497	-0.0003	0.0467	0.0039	0.0918	0.0835	0.1917	-0.0007	0.0266	0.0688	0.1146
58-60%	0.0545	0.0780	0.0447	0.0896	0.0457	0.0691	-0.0001	0.0322	0.0048	0.0912	0.0854	0.1877	-0.0005	0.0266	0.0692	0.1210
68-70%	0.0739	0.0748	0.0744	0.0896	0.0651	0.0853	0.0001	0.0375	0.0060	0.0910	0.0868	0.1846	-0.0005	0.0356	0.1061	0.1055
78-80%	0.0814	0.1089	0.0806	0.1321	0.0689	0.0853	0.0003	0.0756	0.0078	0.0909	0.0879	0.1817	-0.0004	0.0581	0.1041	0.1055
88-90%	0.0865	0.1190	0.0835	0.1406	0.0769	0.0853	0.0008	0.0797	0.0098	0.0951	0.0889	0.1799	-0.0004	0.0643	0.1057	0.1055
98-100%	0.1048	0.1190	0.1115	0.1406	0.0807	0.0853	0.0020	0.0797	0.0121	0.0951	0.0898	0.1799	-0.0004	0.0643	0.1039	0.1055

Table 2.2: Expected carbon benefit/cost to optimal policy for 1AM-7PM on February 17, 2021



(a) Cluster 2 (b) Cluster 4 (c) Cluster 6

Figure 2.5: DP policy for charging and discharging the EV on February 17, 2021

to, and this charge level can be directly tied to the price for charging/discharging or the the driver's own expected behavior. This allows for clear opportunity to earn driver trust as an implemented version of the program could allow drivers to manually change the target charge/discharge rate if they wished to be more or less conservative.

We aggregate potential over the course of a test set of February 16-28, 2021. During this 12-day period, we create policies for each day, such as is shown for February 17, 2021 above and then link the policies by holding the charge constant if the car is attached (neither charging nor discharging) between 10PM and midnight, to prevent falling prey to end of horizon sell off. Thus, if the driver ends the day attached to the grid, they begin the subsequent day at the charge level we left off at 10PM the night before. We summarize the results of the test set in Table 2.3. Both dollar and carbon benefit are highest for clusters that (1) have high on grid minutes, (2) end the day attached to the grid and (3) have low EV usage, while the highest dollar and carbon costs are found in clusters that represent the opposite behavior. We see that even the pure profit maximizer still has a positive carbon benefit, regardless of cluster. This is because price and carbon production are highly correlated in the current electric market, and thus someone trying minimize the cost they pay will implicitly also be minimizing the carbon they consume. When we take a weighted average, based on the size of the test set, the savings are $95 per vehicle from engaging in the vehicle to grid market and a carbon benefit of 0.81 tons of CO2. This is a carbon benefit equivalent to 2,076 miles driven by the average gas powered car, 91.1 gallons

of gasoline consumed, 0.102 of a homes annual energy usage or 13.4 tree samplings grown for 10 years, all achieved in just 12 days.

Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5		Cluster 6		Cluster 7		Cluster 8	
Monetary	Carbon	Monetary	Carbon	Monetary	Carbon	Monetary	Carbon	Monetary	Carbon	Monetary	Carbon	Monetary	Carbon	Monetary	Carbon
131.97	1.34	208.86	1.08	-17.81	0.33	-8.98	0.01	182.46	1.00	-95.72	0.37	124.36	1.57	169.64	1.40

Table 2.3: Expected monetary and carbon benefit/cost to optimal policy for February 16-28, 2021

2.6 Conclusions

In conclusion this chapter addresses both individual EV driver behavior and how this can connect to the grid by proposing a vehicle-to-grid and a vehicle-to-home dynamic optimization model which controls how much EVs charge and discharge. In this chapter, we have taken a driver-centric focus in our proposed optimization model by introducing an objective and constraints that value drivers using their vehicle and incorporate driver behavior. We establish analytical results showing that our policies are easy to communicate and reliable. In doing so we understand the close relationship driver behavior has with how we define the parameters of the optimization. Specifically, we discuss how driver's sensitivity to departing at lower charge levels and driving utility must be directly incorporated into the optimization in order for the policy to serve the EV's ideal driving. While usually optimality must be sacrificed for simplicity, in this setting we show that optimal policies are easy to communicate. When we compute these policies in practice using real driver data, we see that drivers have a lot to gain from participating in discharging markets. On average, there is potential to save \$95 from discharging. We also see that even pure profit maximizers, engaging in discharging markets have a net carbon benefit. The V2G and V2H market is still emerging, and in the coming years we will learn exactly how drivers react to battery discharging and if the proposed dollar and carbon impact are enough for drivers to engage in such programs. However work such as this is necessary to build driver trust and build formulations which put the driver needs at the forefront of the market.

Chapter 3

UMOTEM: Upper Bounding Method for Optimizing over Tree Ensemble Models

3.1 Introduction

When using predictive models to make prescriptive recommendations, better predictive models will result in better outcomes in the real world. For this reason, decision makers are interested in using their most accurate forecasts when optimizing decisions. When the objective of an optimization model is determined by a model's forecast, it is typically more tractable to optimize it when the model itself is linear or parametric. The optimization problem gets increasingly difficult when the optimization formulation uses more complex machine learning models. Forecasts from machine learning models might be more accurate but they are often highly nonlinear and difficult to tractably optimize over, especially when the optimization involves constraints. It is for this reason that recent research (such as [29]–[33]) has focused on different methods for tractably incorporating trained machine learning models into optimization formulations.

Specifically in this chapter, we will focus on tree ensemble methods, such as Random Forests and XGBoost, and how their outputs can be used as inputs for the objective function of an optimization model. Tree ensemble models are a categorization of machine learning models which aggregate single decision trees, such as CART [34], into a single prediction. Tree ensembles usually fall into one of two categorizations: bagging or boosting. In the case of bagging, the weak learner decision trees are generated using different sampled subsets of the data and their predictions are then ensembled together for the final prediction. For boosting, decision trees are generated sequentially so that each new tree models the error of the previous level. These models are extremely popular for their strong predictive power and robustness against over-fitting. This makes these models ideal for forecasting, but their aggregate and black box nature means that using their predictions to make decisions can be difficult. Because tree ensembles aggregate decision trees, their predictive function is a complex piece-wise constant function. A small change in one of the features can result in either no change in the aggregate model's prediction, if the end leaf nodes stay the same,

or a significant change, if the leaf nodes switch. This creates a non-convex function that is difficult to optimize over. Most optimization formulations for this type of problem require an exponentially growing number of binary variables with respect to tree depth, which causes tractability issues when optimizing over objectives that come from tree ensembles even for a reasonable scale.

In this work we propose UMOTEM, an Upper Bounding Method for Optimizing over Tree Ensemble Methods. UMOTEM leverages knowledge known when the underlying trees of the ensemble model are being built to iteratively optimize down the layers of the trees. This results in an approximation of the true optimization in which only a single branch of each tree is traced. This means that UMOTEM scales well, with the number of necessary binary variables growing only linearly with regard to both depth and number of trees in the ensemble. The formulation is generalize-able, with the flexibility to account for various additional business constraints and parallel or hyperplane trees. UMOTEM can also be used to jointly predict and optimize. Rather than training the entire ensemble forest and then optimizing over it, UMOTEM can be combined with the predictive phase of decision making so that only the branch of each tree that has the potential to be optimal is built out. This would not change the optimality of the algorithm but would improve the runtime, as the entire forest would not need to be trained, but rather only the branch of each tree UMOTEM considers optimal. We bound the optimality gap of the UMOTEM approximation in expectation and show that there is a relationship between the quality of the prediction model and the quality of the prescription. The smaller the tree ensemble model's in-sample error, the smaller the expected optimality gap of UMOTEM's solution. This further emphasizes how better predictive models result in better real world outcomes.

3.1.1 Contributions

Our main contribution through this work is the introduction of UMOTEM, an upper bounding method for optimizing over tree ensemble models in a tractable and scalable manner. In more detail:

- **We introduce UMOTEM, an Upper Bounding Method for Optimizing over Tree Ensemble Methods**: UMOTEM is an algorithm that at each step considers a small scale Mixed Integer Optimization (MIO) which finds the near optimal feature vector that maximizes (or minimizes) a payoff determined by the prediction of a tree ensemble model. Unlike many of its peers in the literature, UMOTEM's binary variables grow only linearly with regards to both the tree depth ($d_{\max}$) and the number of trees (T), resulting in a very tractable formulation. The small scale MIO from UMOTEM has $\mathcal{O}(T)$ binary variable and is repeated $\mathcal{O}(d_{\max})$ times, an improvement over its peers which have $\mathcal{O}(2^{d_{\max}}T)$ binary variables.

- **We show how UMOTEM can also be used to jointly predict and optimize**: UMOTEM can be combined with the prediction stage of decision making to improve runtime. Most methods require the entire tree ensemble model to be trained before optimization. However, UMOTEM due to its structure, allows us to combine it with the training phase of the tree ensemble so that only the branch which UMOTEM "suspects" to be optimal gets built. This will not affect the optimality of the solution

found by UMOTEM. As a result, the UMOTEM method allows tree ensembles to be jointly trained and optimized over in a scalable manner even for large datasets.

- **We bound the optimality gap introduced by UMOTEM in expectation**: Because UMOTEM is an approximation method, it can result in an optimality gap. However, we bound this value in expectation and show in the case of well-learned (as defined in Section 3.4) tree ensembles, UMOTEM is expected to do very well. Specifically, we show that as the in-sample error decreases relative to the separation between leaf payoff, UMOTEM's optimality gap also decreases. As a result, more accurate predictive models result in better real world outcomes.

- **We show strong computational results compared to similar methods**: We compare UMOTEM's performance to true optimality on several different publicly available machine learning datasets, covering a wide range of applications. These datasets come from the University of California Irvine machine learning repository and publicly available R packages. We show that UMOTEM is able to consistently reach near optimality (over 90%, and in many cases over 98%) while still scaling linearly. For example, UMOTEM can capture 98% of optimality in 2% of the runtime for some models (see the Concrete dataset in Section 3.5.1).

- **We demonstrate the use of UMOTEM in a case study where other methods do not scale**: Through our collaboration with an American EV manufacturer and the Oracle Retail group, we tackle two case studies highlighting real business cases where the scale of the problems are such that exact optimization methods do not scale to these settings when a random forest is used to predict demand. We apply UMOTEM in these settings to show strong improvements over business practices and other policies. Specifically for the case of the EV manufacturer looking to estimate the size of the discharging market, we show that compared to the baseline policy, drivers can save $41.1 and the CO2 equivalent of 667 miles driven by a gas powered car in two weeks from engaging in the discharging market. For the retail case study, we optimize discounts for fashion items over a 21 week time horizon to maximize revenue and show an improvement of 12-13% in terms of revenue.

3.2 Literature Review

In recent years, there has been a significant increase in research on how predictive tools can be incorporated into prescriptive tasks. Researchers are interested in how machine learning models can be used as an input function in the objective when optimizing decisions. In this work, we focus on how to optimize over tree ensemble models, and therefore touch upon a few different areas of research. Specifically we discuss research on tree ensemble models themselves, optimization methods over various predictive tasks, and finally research that discusses optimization of types of tree ensembles.

Tree ensemble models combine the best of tree modelling structure with the robustness and accuracy of ensembling. Popular in practice because of their high performance, there has been a lot of development in different types of tree ensemble models. At their core, tree

ensemble models use many "weak learner" decision trees that are then aggregated together to form a single prediction. Among the most popular are Random Forest [35] and XGBoost [36]. Random Forests build their weak learners by sampling the input data which builds the trees. XGBoost is a variant on the gradient boosting strategy, in which trees are built sequentially with each additional one targeting the residuals of the previous. These two models are just the tip of the iceberg. Other tree ensemble models include AdaBoost [37] and LightGBM [38]. For a more thorough survey of tree ensemble models, we direct interest readers to [39]. We build on this work by showing how these types of models can be used as inputs to prescriptive tasks. The only requirement for our work is that the tree ensemble model must have the weak learners be decision trees and the method of aggregating the weak learners should take on some form of a weighted sum. This weighted sum is the case for the most popular methods of tree ensemble models, including but not limited to Random Forest, gradient boosting machines (such as XGBoost and LightGBM), and AdaBoost.

Additionally, there has also been focus in the literature on how predictive tasks, especially machine learning models, can be used as inputs into prescriptive tasks. This stems from a natural need, as decision makers are interested in using their high-quality forecasting models to improve decision making. [29] optimize chemotherapy regimes for cancer using a ridge regression as input to predict outcomes of clinical trials. [30] incorporate a log-linear regression of demand into their optimization of promotion vehicles. [40] and [41] optimize revenue with a ranking-based choice model. Neural networks, as one of the cutting edge forms of machine learning, have been the focus of a lot of research. [42]–[47] have all worked to incorporate them into the objective of optimization formulations. [48] in particular use the eventual prescriptive decision of the optimization as a factor in the evaluation of a model's statistical validity.

Specifically regarding tree ensemble methods, such as the aforementioned Random Forest or XGBoost, being incorporated into optimization, a few different methods have been explored. [31] formulate an optimization where the objective function coefficients are determined by a Random Forest. They do this by precalculating the coefficients through the Random Forest in advance for every permutation of the variables. In doing so, [31] avoid many of the computational challenges that come from directly encoding the Random Forest into the optimization formulation. However while this works well for the pricing problem considered in [31], this approach is less efficient when there are many variables that are being optimized, for which the decision maker would need to forecast for all possible combinations of their realizations. [32] and [33] both discuss methods for incorporating the tree structure of the model directly into the optimization formulation. Both formulate a version of a mixed-integer optimization that represents the splits of the trees in the constraints of the model, but this requires an exponential number of binary variables as the trees grow in depth. For scalability purposes, these papers approximate either the depth of the tree ensemble or the breadth (that is, in the latter case, they sample only a handful number of trees). UMOTEM is an approximation of these MIOs with better tractability properties. We discuss these two methods in greater detail in Section 3.3.1 and compare how UMOTEM performs regarding both runtime growth and computational performance throughout the chapter. There are other recent works in this area, for example, [49] extend the work of [32] to the case of product assortment optimization and show that a stronger formulation can be achieved through aggregating constraints over products. [50] draw the connection between

multilinear optimization and optimization over tree ensembles. [50] use this to create tighter formulations and show when the problem can be reduced to polynomial-sized formulations (rather than exponential). [51] accomplish two main tasks: first, they show how to optimize when the objective function is the sum between a function on the output of a boosted tree model and a convex penalty term and second, they help reduce the scale of the optimization by introducing a branch-and-bound methodology. While they solve a more complex problem with the convex penalty term, they do not provide guarantees on scaling or analytically bound the expected optimality gap. Our work also touches upon a wide range of end-to-end literature where the tree ensemble model is used as part of a larger process. [52] introduce a framework for training machine learning models with the subsequent optimization task in mind. More specific to decision trees, [53], adjust the training process for ensemble forests, specifically split selection, to account for the downstream optimization and improve the optimality of the end-result. Unlike these works, this work keeps the prediction quality the same, but shows how when tree ensembles are being used for optimization, UMOTEM can be employed to save on training time by only building out the branch that is expected to be optimal.

3.3 UMOTEM

We start by describing two different formulations of the MIO which optimize an objective function determined by an tree ensemble model. One formulation is proposed by [32], and another by [33]. These formulations are largely equivalent in terms of optimality. In the rest of the chapter, we will refer to the MIO described by [32] or [33] as the true MIO or exact MIO because both formulations achieve full optimality while UMOTEM is an approximation of this MIO. We expand upon our discussion in Section 3.2 on their limitations in terms of scalability, which we address through UMOTEM. Both papers also propose approximations for their exact MIOs, which we address analytically (end of Section 3.4) and computationally (Section 3.5.1). One key advantage of UMOTEM over other approximation methods is that UMOTEM does not require the practitioner to select an approximation level, which if misselected, has the potential to give rise to suboptimal results. In this section, we first describe the single-tree version of UMOTEM before finally introducing the full ensemble forest version of UMOTEM. We assume in this work that the feasible region UMOTEM is optimizing over is bounded. We also assume that the tree ensemble uses a weighted aggregation method and the underlying trees of the ensembles use parallel splits.

The main challenge for directly incorporating ensemble forests into an optimization formulation, and hence the need for approximations like UMOTEM, comes from the binary nature of the trees themselves. Decision trees, such as those found in models like CART, Random Forest, or XGBoost, ask a series of questions with a yes/no outcome in order to make a final prediction. At each interior node, i, of tree, t, there is a linear condition of the form: $(a_i^t)^\top x \geq b_i^t$. If x satisfies this linear condition, then the model moves to the left child of the tree, and if x fails the condition, the model moves to the right child. This iterative process continues until a leaf or terminal node is reached, at which point the model produces a prediction for feature vector x. Each leaf node, j, in tree, t, has a payoff S_j^t associated with it. This payoff is usually either the average of the dependent variable, y, of all training

data points which end up in node j, or some function on this average. In order to help with occasionally dense notation, we introduce a concrete example in Figure 3.1. Here we show a three tree Random Forest, of max depth two, that uses synthetic retail data to predict sales. There are two features used to create this forest: full Price and Discount. The Price variable here is the price before the discount is applied. In this case, if x is a feature vector with Discount $= 0.3$ and Price $= 19$, this means that this item had a full price of \$19 and a discount offer of 30% off. Following the splits of tree 1, x would thus end up in leaf node n_5^1.

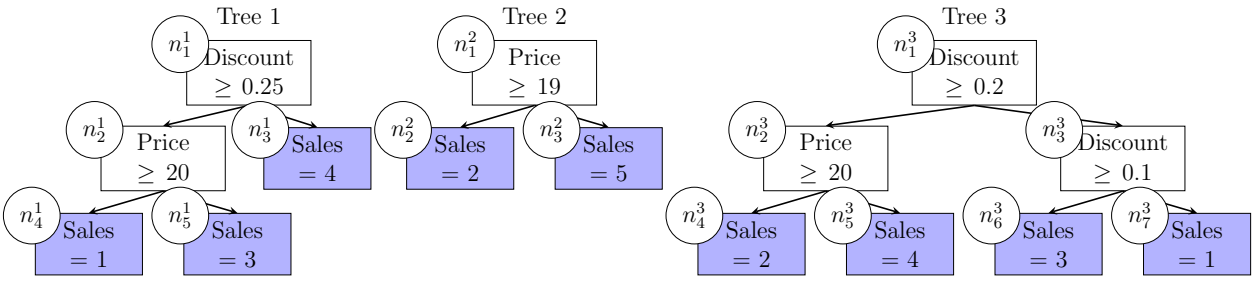


Figure 3.1: Sample Random Forest 1

3.3.1 The MIO Formulations for True Optimality

We define the following elements of the tree ensemble model. The ensemble consists of T trees, where each tree has weight w_t in the final prediction. Let m be the minimum leaf size of the trees, which is specified when the ensemble forest is trained. N^t defines the set of interior nodes in tree t, and L^t defines the set of leaf nodes in tree t. Each node $i \in N^t$ has a split criterion $(a_i^t)^\top x \geq b_i^t$, a parent node p_i^t, a left child node l_i^t, and a right child node r_i^t. From the example in Figure 3.1, we would have $T = 3$, and because it is a random forest, $w_t = \frac{1}{T} = \frac{1}{3}$. For Tree 1, $N^1 = \{n_1^1, n_2^1\}$ and $L^1 = \{n_3^1, n_4^1, n_5^1\}$. If we consider node n_2^1, its parent node is $p_2^1 = n_1^1$, its left child is $l_2^1 = n_4^1$, and its right child is $r_2^1 = n_5^1$.

We start by describing the formulation introduced by [32]. Here, one assumes that all the splits of the tree are parallel. This means that the vector a_i^t is all zeros except for a single one. A single feature is considered for each split. Then one can rewrite $(a_i^t)^\top x \geq b_i^t$ as $x_{p(i)} \geq b_i^t$, where $p(i)$ specifies which feature is being considered in interior node i. For example, when looking at the sample Random Forest in Figure 3.1, $p(n_1^1) =$ Discount.

We can then group all interior nodes across the trees by which feature is being considered in the split. We can define $\mathcal{P}$ as the set of all features which are split on within the tree ensemble. For each feature $p \in \mathcal{P}$, we can define $\mathcal{B}_p$ to be the ordered set of all split points in the ensemble. Let $K_p = |\mathcal{B}_p|$ be the number of unique split points across the ensemble that use feature p. $\mathcal{B}_p$ is characterized such that for any $b_{p,k} \in \mathcal{B}_p$, there exists some node i such that its split criterion is $x_p \geq b_{p,k}$. Because the set is ordered, we have that $b_{p,1} < b_{p,2} < \cdots < b_{p,K_p}$. $b_{p,k}$ will be the k^{th} smallest split point for feature p. We define the function $C(i) = \{j\}$, so that for node i it identifies the index j so that the split criterion of node i can be written as $x_{p(i)} \geq b_{p(i),C(i)} = b_{p(i),j}$. If we consider the sample forest, we would have that $\mathcal{P} = \{\text{Price}, \text{Discount}\}$, $\mathcal{B}_{\text{Price}} = \{19, 20\}$, $\mathcal{B}_{\text{Discount}} = \{0.1, 0.2, 0.25\}$, $K_{\text{Price}} = 2$,

and $K_{\text{Discount}} = 3$. If we consider node n_1^1, we have that $C(n_1^1) = 3$. Therefore, we can represent the split criterion of node n_1^1 as $x_{p(n_1^1)} \geq b_{p(n_1^1),C(n_1^1)}$, or $x_{\text{Discount}} \geq b_{\text{Discount},3} = 0.25$.

We also define for every interior node i in the ensemble, $\mathbf{left}(i)$ and $\mathbf{right}(i)$. $\mathbf{left}(i)$ is the set of all leaf nodes which are descendants of the left child of node i, while $\mathbf{right}(i)$ is the set of all leaf nodes which are descendants of the right child of node i. If x ends up in a leaf node $j \in \mathbf{left}(i)$, then it is known that the criterion of node i is satisfied by x, while if x ends up in a leaf node $j \in \mathbf{right}(i)$, then it is known that the criterion of node i is failed by x. As an example, for the sample random forest, we have that $\mathbf{left}(n_1^1) = \{n_4^1, n_5^1\}$ and $\mathbf{right}(n_1^1) = \{n_3^1\}$.

With this notation we can begin reviewing a formulation for the exact MIO for this problem. This formulation starts by defining binary variables v to determine which leaf node a feature vector, x ends up in. $v_{t,j}$ are only defined for the leaf nodes j in trees t. $v_{t,j} = 1$ if and only if solution x resides in leaf j of tree t. [32] also defines binary variable z to specify the optimal range of the solution x, by describing which splits of the ensemble are satisfied or have failed. Binary variable $z_{p,k} = 1$ if for feature p, the value of x is greater than or equal to the k^{th} split point in $\mathcal{B}_p$, and 0 otherwise. For example, if $z_{\text{Price},1} = 0$, $z_{\text{Price},2} = 0$ then we know that $x_{\text{Price}} < 19$. If $z_{\text{Price},1} = 1$, $z_{\text{Price},2} = 0$ then $19 \leq x_{\text{Price}} < 20$. And finally, if $z_{\text{Price},1} = 1$, $z_{\text{Price},2} = 1$ then $20 \leq x_{\text{Price}}$. This is enough to specify which leaves the features vector x will end in and what the prediction of the tree ensemble model will be.

First we start with the objective function. The objective of the MIO is to find a feature vector x which maximizes the payoff of the leaf it ends up in: $\max_{z,v} \sum_{t=1}^{T} \sum_{j \in L^t} w_t v_{t,j} S_j^t$.

For each tree we have to end in one and only one leaf: $\sum_{j \in L^t} v_{t,j} = 1$.

To enforce split constraints we leverage $\mathbf{left}(i)$ and $\mathbf{right}(i)$. If we fail the split constraint at node i, then we know that we cannot end up in any of the leaves which descend from the left child (ie $\sum_{j \in \mathbf{left}(i)} v_{t,j} = 0$). We also know that if we fail the split constraint of node i, then $z_{p(i),C(i)} = 0$. Similarly, if we satisfy the split constraint, then we cannot end up in any of the leaves which descend from the right child (ie $\sum_{j \in \mathbf{right}(i)} v_{t,j} = 0$), and $z_{p(i),C(i)} = 1$. In order to enforce this relationship, we introduce the following constraints: $\sum_{j \in \mathbf{left}(i)} v_{t,j} \leq z_{p(i),C(i)}$ and $\sum_{j \in \mathbf{right}(i)} v_{t,j} \leq 1 - z_{p(i),C(i)}$.

If we satisfy the j^{th} constraint of feature p, then we know that $x \geq b_{p,j} \geq b_{p,j-1} \geq \cdots \geq b_{p,1}$. Thus if $z_{p,j+1} = 1$ then it must be true that $z_{p,j} = 1$: $z_{p,j} \geq z_{p,j+1}$.

Finally, we need z to be binary and $v \geq 0$. Putting this all together gives rise to the following MIO formulation (please see [32] for more details and for the case where features are categorical):

$$
\begin{aligned}
\max_{z,v} \quad & \sum_{t=1}^{T} \sum_{j \in L^t} w_t v_{t,j} S_j^t & & \\
\text{s.t.} \quad & \sum_{j \in L^t} v_{t,j} = 1, & & \forall t \in \{1, \ldots, T\} \\
& \sum_{j \in \textbf{left}(i)} v_{t,j} \leq z_{p(i),C(i)}, & & \forall t \in \{1, \ldots, T\}, i \in N^t \\
& \sum_{j \in \textbf{right}(i)} v_{t,j} \leq 1 - z_{p(i),C(i)}, & & \forall t \in \{1, \ldots, T\}, i \in N^t \\
& z_{p,j} \geq z_{p,j+1}, & & \forall p \in \mathcal{P}, j \in \{1, \ldots, K_p - 1\} \\
& z_{p,j} \in \{0, 1\}, & & \forall p \in \mathcal{P}, j \in \{1, \ldots, K_p\} \\
& v_{t,i} \geq 0 & & \forall t \in \{1, \ldots, T\}, i \in L_t
\end{aligned}
\tag{3.1}
$$

For illustrative purposes, we include Formulation (B.1) in the Appendix. Formulation (B.1) is a concrete realization of Formulation (3.1) for the forest shown in Figure 3.1.

An alternative way of formulating the same problem is proposed by [33]. While Formulation (3.1) used binary variables z to determine which leaf x ended in, this formulation uses binary variables q. In order to represent whether x satisfies the split criterion of node i in tree t, [33] define the binary variable $q_{i,j}^t$. $q_{i,j}^t = 1$ if and only if in tree t feature vector x lies in a descendent of node j and if node j is a direct child node (either left or right) of node i. We can use q to start building the objective function and constraints of the true MIO.

We start again with the objective funtion. In this formulation, for a given tree, the payoff can be calculated as $\sum_{j \in L^t} q_{p_j,j}^t S_j^t$. We then sum this over all trees, weighted by w_t, to get the objective function: $\max_{x,q} \sum_{t=1}^{T} \sum_{j \in L^t} w_t q_{p_j,j}^t S_j^t$.

One needs to enforce that the variables q accurately represent which leaf x ends up in. In order to do this, the formulation needs to enforce the split criterion of all the interior nodes. For interior node i, if $(a_i^t)^\top x \geq b_i^t$ then $q_{i,l_i} = 1$. Similarly if $(a_i^t)^\top x \leq b_i^t$ then $q_{i,r_i} = 1$. The paper accomplishes this by introducing big-M constraints: $(a_i^t)^\top x + M(1 - q_{i,l_i}^t) \geq b_i^t$ and $(a_i^t)^\top x - M(1 - q_{i,r_i}^t) \leq b_i^t$.

This needs to be paired with genealogy or flow constraints. Namely, if x exists in a leaf descendent of node i (ie $q_{p_i,i}^t = 1$), then it must exist in a descendent of one of i's children ($q_{i,l_i}^t = 1$ or $q_{i,r_i}^t = 1$). If x does not exist in a leaf descendent of node i ($q_{p_i,i}^t = 0$), then it cannot exist in a descendent of either of its children ($q_{i,l_i}^t = q_{i,r_i}^t = 0$). Therefore, one needs to also include the following constraint: $q_{p_i,i}^t = q_{i,l_i}^t + q_{i,r_i}^t$.

In addition, x must end up in one of the leaf nodes of tree t: $\sum_{j \in L^t} q_{p_j,j}^t = 1$.

Finally, one may wish to enforce any other relevant business constraints: $Ax = f$.

When one puts all these pieces together, Formulation (B.2) as shown in the Appendix, rises. This formulation was originally introduced by [33]. For illustrative purposes, we also include Formulation (B.3) in the Appendix. Formulation (B.3) is a specific example of Formulation (B.2) for the forest shown in Figure 3.1.

The main limitation of both Formulations (3.1) and (B.2) are the binary variables $z_{p,j}$ in Formulation (3.1) and $q_{i,j}$ in Formulation (B.2). Each new interior node can require a new set of binary variables. As the depth of the tree increases, the number of needed

binary variables increases exponentially. Even for just the illustrative sample forest we show in Figure 3.1 with merely three trees and max depth two, Formulation (B.2) requires 12 binary variables, and Formulation (3.1) requires 5. Tractability is particularly challenging when applying this formulation to Random Forest models because one of the premises of the model is that its trees are not pruned (or shortened), so Random Forest trees can become rather deep. For example, even for a dataset that is less than 200 data points, the average tree depth in a forest can be over 10. With a forest of 500 trees, this could easily result in over 500,000 binary variables. Overall, it can be very difficult to run these formulations for most reasonably sized ensemble forests.

3.3.2 Upper Bound Function for a Single Tree

It is with this in mind that we begin to introduce UMOTEM. Rather than optimizing all levels of the tree at once, we introduce a method that iteratively optimizes each level of the trees in an ensemble. The key idea behind UMOTEM is to only use the information available up to depth d to decide whether the optimal leaf is a descendent of the left or right child of a node at depth d. We only explore the branch that we suspect to have the optimal leaf, resulting in us only tracing a single branch through a tree. This linearizes the exponential growth of variables in the optimization formulation. In this section, we start with a single tree without additional constraints for ease of exposition, and then expand to the full tree ensemble model with additional constraints in Section 3.3.3.

Let us first consider node i in a decision tree. The goal is to determine, from all the leaves which are a descendant of node i, whether the leaf with the highest payoff descends from the left branch or right branch. Any leaf to come from this node i will be limited by the data first passed to node i by the higher levels of the tree and filtered out by the interior nodes that are descendants of node i. We can use this data passed to node i and the split criterion of node i, to calculate an upper bound on the payoff of the best leaf. We will refer to this upper bound as the UB function for short. We define $u_{i,L}$ and $u_{i,R}$ as the value of the UB function for the left and right children of node i. We calculate the UB function of a node as the average of the top m payoffs to fall to either child node, where m is the minimum leaf size of the tree. By calculating the UB function, we approximate whether the left or right branch of node i has the highest leaf value.

Mathematically we calculate this by first defining the set $\mathcal{D}$ to be the set of all training data points which fall into node i. For example, in the sample forest in Figure 3.1, the data which is passed to node n_2^1 is all the data used to train tree 1 that satisfies the split criterion of node n_1^1. We let $y_k : k \in \mathcal{D}$ represent the dependent variable of data point k in the set $\mathcal{D}$. We denote the order statistic to be $y_{(k)}$, such that $y_{(1)} \geq y_{(2)} \geq y_{(3)} \geq \dots$. This means that $y_{(1)}$ is the highest dependent value passed through node i, $y_{(2)}$ is the second highest and so on. Given this, we can define the UB function as:

$$u = \sum_{k=1}^{m} \frac{1}{m} y_{(k):k\in\mathcal{D}}. \tag{3.2}$$

It is safe to consider this as an upper bound of the possible leaf values because any leaf descendent of node i will be limited to the data points in $\mathcal{D}$. u captures essentially what

happens if the top payoffs all end up in a leaf and nothing else, marking it as the best possible payoff value a leaf of node i could have. This calculation of the UB function requires no knowledge of the tree beyond information on the node itself and can be pre-calculated when the tree is being built.

Again, we refer to the sample forest introduced in Figure 3.1 as a concrete example, and focus on Tree 1. Let us assume Tree 1 was built on a dataset of ten data points, shown in Table 3.1, and with a minimum leaf size of 3. This means that all leaves of the tree must have at least 3 data points in them. Knowledge of the entire tree would make it clear that the optimal leaf is n_3^1. However, at depth 0, the algorithm is considering only information available to us at node n_1^1 to determine if the optimal leaf is a descendent of n_2^1 or n_3^1. All ten data points fall into the root node, n_1^1, and we can split the data based on the split criterion for n_1^1, namely Discount ≥ 0.25. Based on this we know that data points 1-6 will pass to node n_2^1 and data points 7-10 will pass to node n_3^1. For Tree 1, where $m = 3$, we would have that $u_{1,L}^1 = \frac{2+2+7}{3} = 3.6$ and $u_{1,R}^1 = \frac{4+4+5}{3} = 4.3$. Based on this model, one would assume the leaf with the highest payoff is a descendent of the right branch of node n_1^1.

While for simplicity of exposition we will continue using this version of the upper bound method, the exact formulation for u can be adjusted based on the decision maker's preferences. For example, one of the criticisms of most tree-based regressions is that they produce point estimates, which can be easily skewed by outliers. If the decision maker is interested in being robust against such error, they can remove outliers of the set before calculating the average. In such a case $u_{1,L}^1$ would be $\frac{2+2+1}{3} = 1.6$. Another example of the flexibility would be the case where the decision maker is interested in hedging against the worst case scenario. In this case they are trying to maximize the payoff of the worst possible leaf that could come from the branch. In such a scenario u would be the average of the smallest payoffs in the branch, or $\sum_{k=|\mathcal{D}|-m+1}^{|\mathcal{D}|} \frac{1}{m} y_{(k):k\in\mathcal{D}}$ and thus $u_{1,L}^1 = \frac{0+0+1}{3} = 0.3$ and $u_{1,R}^1 = \frac{4+4+3}{3} = 3.6$. In this way the upper bound method leverages the learning of the tree ensemble methods, namely how to separate out the feature space into areas of different payoff but incorporates more flexibility when optimizing over the forests to allow for more objectives.

Table 3.1: Synthetic training data for Tree 1

Data ID	**Price**	**Discount**	**Sales**	**Leaf Node**
1	21	0.3	0	n_4^1
2	24	0.5	2	n_4^1
3	22	0.3	1	n_4^1
4	19	0.4	0	n_5^1
5	15	0.3	2	n_5^1
6	18	0.5	7	n_5^1
7	20	0	4	n_3^1
8	17	0.1	4	n_3^1
9	22	0.2	5	n_3^1
10	16	0	3	n_3^1

This is enough to approximate the single tree optimization when there are no additional constraints. The optimization would select whether to move to the left or right child of node

i based on the following optimization, pulled from Formulation (3.1):

$$\begin{aligned} \max_{z,v} \quad & (v_L u_{i,L} + v_R u_{i,R}) \\ \text{s.t.} \quad & v_L + v_R = 1 \\ & v_L \leq z_{p(i),C(i)} \\ & v_R \leq 1 - z_{p(i),C(i)} \\ & z_{p(i),C(i)} \in 0, 1 \\ & v_L, v_R \geq 0 \end{aligned} \tag{3.3}$$

It is worth noting that unlike Formulation (3.1), which has binary variables and constraints for every split point in the feature space, Formulation (3.3) only has one binary variable, representing whether to go left or right.

For a single tree, the node pointer n_i would start at the root node. Then the binary variable $z_{p(i),C(i)}$, and by extension v_L and v_R would indicate whether the pointer i should move to the left or right child node. If $v_L = 1$ the n_i would point to the left child, otherwise n_i would point to the right child and Formulation (3.3) would be run again. The process would repeat until n_i pointed towards a leaf node, which would indicate an optimal solution. This process would trace only a single branch of the tree.

This algorithm has a linear growth in terms of binary variables. Even though the number of nodes in a tree of depth d is worst case $2^d - 1$, meaning that Formulations such (3.1) and (B.2) would scale similarly, UMOTEM would in the worst case use only d binary variables (one binary variable for each iteration of the algorithm).

While for the sake of exposition, we stick to the version of UMOTEM in which we greedily iterate one level of the tree at the time, this is not a requirement of the algorithm. It is possible to consider more levels of the tree at once. For example, one can consider a two-level iteration in which the algorithm considers whether to go left or right on the first two levels of the tree and pick which of it's (at most) four "grandchildren" nodes to move to, and then iterate from there, considering the next two levels. The trade-off to consider is that the more levels of the tree optimized at once, the more likely UMOTEM is to find optimality. However, as the number of levels considered in one iteration increases, the tractability decreases. This can be considered on a trade off curve. At one extreme is one-depth UMOTEM, as outlined in this section of the chapter. It is the most tractable but with the highest risk of suboptimality. At the other extreme is the exact MIO which considers all levels of the tree at once. It guarantees optimality, but has the highest tractability risk. In between these points lie the different depths of iteration, creating an efficient frontier of the best optimality/time that can be achieved. We expand upon this idea in Section 3.5.1, where we plot this curve for one of the datasets. We now introduce the multi-tree or ensemble version of UMOTEM.

3.3.3 UB Function for Tree Ensembles

The expansion to ensemble models is fairly straight-forward at this stage. Rather than optimizing a single level of a single tree and repeating until a leaf node is reached, UMOTEM optimizes one level of all trees at the same time and iterates until it reaches a leaf node in all

trees. The key for doing this is maintaining feasibility. If at depth d, the algorithm chooses to satisfy a split criterion $(a_i^t)^\top x \geq b_i^t$, then when the algorithm moves to depth $d+1$ and beyond, it needs to know that all future versions of decision variable x still satisfies the split constraint $(a_i^t)^\top x \geq b_i^t$.

For example, consider if in the sample Random Forest in Figure 3.1, at depth 0, the optimization decides that it should satisfy interior nodes n_1^1 and n_1^3 and fail to satisfy node n_1^2. At the next level it would be evaluating nodes n_2^1, n_3^2 and n_2^3. n_3^2 is a leaf node so that is not a concern, but if it is considering whether to go left or right on nodes n_2^1 and n_2^3, then the algorithm needs to have stored the decision of previous nodes. Given it does not satisfy the constraint of n_1^2, Price ≥ 19, it must by definition then also not satisfy the constraint of node n_2^1, Price ≥ 20.

One way of storing these requirements would be at each iteration of the algorithm adding constraints for whether each split node is satisfied or failed. However, the number of constraints this would result in would be quite burdensome as the algorithm gets deeper in the tree. Instead we use box constraints to store the split criterion the algorithm has chosen to satisfy or fail. Box constraints would store the maximum and minimum value of the feasible region for each feature. The benefit of box constraints are that they are simple and easy to store. Rather than adding two new constraints for every tree at every iteration, tractability is maintained because there will only be two constraints for each feature at all iterations. This simplification is possible because in the case of parallel split trees, these maximum and minimum constraints are enough to summarize the split constraints for each feature. For example, in the case of the example from the previous paragraph, where the optimization chooses to not satisfy the split constraints of both nodes n_1^2 and n_2^1, then rather than adding both the constraints Price ≤ 19 and Price ≤ 20 to the optimization problem, the box constraints would just store Price ≤ 19. The box constraints effectively summarize the requirements on the split node constraints and are easy to learn. The process of learning the box constraints is explained in Step 2 of the UMOTEM formulation.

While in this chapter we focus on parallel split trees, it is worth noting that UMOTEM can be easily extended to the case of hyper-plane trees. Hyper-plane trees are trees where split constraints are built off of multiple features. In this case the algorithm can either add all the split constraints at each iteration or approximate the polytope-shaped feasible region introduced by hyper-plane splits with box constraints. The benefit of adding all the split constraints is that it will be an exact representation of the feasible region at each iteration of the algorithm. Box constraints will only be an approximation of this feasible region, but their benefit is guaranteed tractability. However, for the rest of this work we will focus on the case of parallel splits, as these trees are the most common weak learners for tree ensemble models.

UMOTEM can be summarised in three steps. We initialize $x_{\min}$ and $x_{\max}$, the upper and lower bounds for the box constraints, as the initial bounds of the feature set. We also initialize the node pointer i to the root node of every tree t in the ensemble. We define $\hat{\mathcal{B}}_p$ as the set of splits in the nodes i across all trees, along with $x_{\min,p}$ and $x_{\max,p}$. Like before we define $\hat{K}_p = |\hat{\mathcal{B}}_p|$. While the original $\mathcal{B}_p$ would grow with the number of splits in the forest, $\hat{K}_p \leq T+2$ (one split for each node in each tree, plus $x_{\min}$ and $x_{\max}$), is only defined for a single level of the forest and thus is significantly smaller. We initialize the set of trees to

optimize over, $\mathcal{T}$, as all the trees in the forest.

1. Optimize for One Level For nodes n_i^t that we are considering, we evaluate based on the UB function whether we want to move to the left or right child. The objective is to maximize the payoff, as estimated through the UB function. Therefore the objective function becomes: $\max_{z,v} \sum_{t\in\mathcal{T}} w_t \left(v_{t,L} u^t_{i_t,L} + v_{t,R} u^t_{i_t,R}\right)$.

As was the case for Formulation (3.1), v represents whether we move to the left or right child of node n_i and z defines the optimal feature vector which will maximize the payoff.

Binary variable v needs to accurately represent whether we move to left or right child of node i in tree t based on the value of z: $v_{t,L} \leq z_{p(i_t),C(i_t)}$ and $v_{t,R} \leq 1 - z_{p(i_t),C(i_t)}$.

We need to pick the left or right branch of node i of tree t, therefore, we enforce: $v_{t,L} + v_{t,R} = 1$.

As before, if we satisfy the j^{th} constraint of feature p, then we know that $x \geq b_{p,j} \geq b_{p,j-1} \geq \cdots \geq b_{p,1}$. Thus if $z_{p,j+1} = 1$ then it must be true that $z_{p,j} = 1$: $z_{p,j} \geq z_{p,j+1}$.

We need to enforce feasibility. To do this, we find the $j_p^{\max}$ and $j_p^{\min}$ such that $b_{p,j_p^{\max}} = x_{\max,p}$ and $b_{p,j_p^{\min}} = x_{\min,p}$. We then enforce that $z_{p,j_p^{\max}} = 0$ (we have to be less than $x_{\max}$) and $z_{p,j_p^{\min}} = 1$ (we have to be greater than $x_{\min}$).

Finally we need z to be binary and v to be non-negative: $z_{p,j} \in \{0,1\}$ and $v_{t,L}, v_{t,R} \geq 0$.

When we put all these constraints together we get, where index i_t refers to the node being pointed at by n_i^t:

$$
\begin{aligned}
\max_{z,v} \quad & \sum_{t\in\mathcal{T}} w_t \left(v_{t,L} u^t_{i_t,L} + v_{t,R} u^t_{i_t,R}\right) & & \\
\text{s.t.} \quad & v_{t,L} \leq z_{p(i_t),C(i_t)}, & & \forall t \in \mathcal{T} \\
& v_{t,R} \leq 1 - z_{p(i_t),C(i_t)}, & & \forall t \in \mathcal{T} \\
& v_{t,L} + v_{t,R} = 1, & & \forall t \in \mathcal{T} \\
& z_{p,j} \geq z_{p,j+1}, & & \forall p \in \mathcal{P}, j \in \{1,\ldots,\hat{K}_p - 1\} \\
& z_{p,j_p^{\max}} = 0, & & \forall p \in \mathcal{P} \\
& z_{p,j_p^{\min}} = 1, & & \forall p \in \mathcal{P} \\
& z_{p,j} \in \{0,1\}, & & \forall p \in \mathcal{P}, j \in \{1,\ldots,\hat{K}_p\} \\
& v_{t,L}, v_{t,R} \geq 0, & & \forall t \in \mathcal{T}.
\end{aligned}
\tag{3.4}
$$

The solution to Formulation (3.4) suggests whether it is more optimal to move to node n_i^t's left or right child. It also provides a sample feature representation z which guarantees the selections of left or right children are feasible. It is worth noting that while we use the formulation from [32], we can replace the formulation with almost any of the other proposed exact versions of the tree ensemble MIO using only the constraints and variables for a single depth. All that is required is the ability to replace the leaf payoffs with the calculated upper bounds, to add the box feasibility constraints and only include the constraints which pertain to the nodes in $\mathcal{T}$.

2. Update Optimal Feasible Region Before we move down in depth, we need to first update the box constraints which limit our feasible region, based on whether the algorithm has decided to move left or right in each tree. Specifically, for each feature $p \in \mathcal{P}$, Formulation (3.4) provides z, which tells us if the optimal solution fails or satisfies each of the split

constraints of the node. For feature p, we find the $j_p^{*\min}$ such that it is largest j such that $z_{p,j} = 1$. Then we define a new $x_{\min,p} = b_{p,j_p^{*\min}}$. Similarly, we find $j_p^{*\max}$ such that it is smallest j such that $z_{p,j} = 0$ and define $x_{\max,p} = b_{p,j_p^{*\max}}$.

3. Update Node Pointers The final step of the algorithm is to update the pointer of n_i^t. As with the single tree example, if $v_{t,L} = 1$ then the pointer would move to the left child and if $v_{t,R} = 1$ then the pointer would move to the right child. If the new n_i^t points to a leaf node, then its tree t is removed from the set of trees to optimize over, $\mathcal{T}$. When the pointers all point to leaf nodes, or in other words when the set $\mathcal{T}$ is empty, the algorithm terminates.

We write this process more formally in Algorithm 1.

Algorithm 1 UMOTEM

$\mathcal{T} \leftarrow \{1 \dots T\}$ ▷ Start with all trees to be optimized
$n_i^t \leftarrow n_1^t \quad \forall t \in \mathcal{T}$ ▷ Start at the root node
Initialize $x_{\min}$ and $x_{\max}$
while $|\mathcal{T}| > 0$ **do** ▷ End when there are no more trees to optimize
  Optimize for One Level:
  Initialize $\hat{\mathcal{B}}_p$ ▷ Create the ordered set of n_i^t splits, $x_{\min}$ and $x_{\max}$
  Run Formulation (3.4)
  Update Optimal Feasible Region:
  for $p \in \mathcal{P}$ **do**
    $j_p^{*\max} = \min_{j:z_{p,j}=0} j$ ▷ Find the smallest split the optimization fails
    $j_p^{*\min} = \max_{j:z_{p,j}=1} j$ ▷ Find the smallest split the optimization satisfies
    $x_{\max,p} \leftarrow b_{p,j_p^{*\max}}$
    $x_{\min,p} \leftarrow b_{p,j_p^{*\min}}$
  end for
  Update Node Pointers:
  for $t \in \mathcal{T}$ **do**
    if $v_{t,L} = 1$ **then**
      $n_i^t \leftarrow n_{l_i}^t$ ▷ Move to the left child of node i
    else
      $n_i^t \leftarrow n_{r_i}^t$ ▷ Move to the right child of node i
    end if
    if $n_i^t \in \mathcal{L}^t$ **then** ▷ If in a leaf node, remove t from optimization set
      $\mathcal{T} \leftarrow \mathcal{T} \setminus \{t\}$
    end if
  end for
end while

The key benefit of UMOTEM as described by Algorithm 1 is that the number of binary variables grows linearly in terms of both the number of trees and tree depth. This stems from the fact that the algorithm traces only a single branch of each tree when optimizing. The first optimization in an iteration of UMOTEM, Formulation (3.4), consists of at most $T + 2$ binary variables ($z_{p,j}$). The upper bound on the number of iterations of the algorithm is the maximum depth of the trees in the forest. If we define the depth of the deepest tree as $d_{\max}$, then it follows that UMOTEM will involve at most $(T + 2)d_{\max}$ binary variables.

This is better than other approximation methods proposed for this problem because it does not require exponentially-growing binary variables.

It is worth noting that because UMOTEM explores only a single branch of each tree and only makes its decision based on information available to the node it is at, it is possible to combine the predict and optimize steps of the decision making process. Rather than training an ensemble tree model and then optimizing it using UMOTEM, it is possible to build the trees and optimize the branches at the same time. This algorithm would involve, for each level of each tree, the following steps:

1. Determine if the node is a leaf node. If not, continue on to step 2.
2. Select the split criterion for the node
3. Use UMOTEM to determine whether to build out the left or right child node and move to this node
4. Repeat steps 1-3 until a leaf node is reached

The resulting forest would be an ensemble of "trees" in which only a single branch is fully fleshed out. This branch for each tree would be the branch where decision makers would expect the optimal solution to lie. The result of this process would yield the same solution as the solution found by Algorithm 1. However because this process would involve only building out a single branch for each tree rather than the full forest, this can lead to improvements in the time it takes to build and optimize tree ensemble models. Please see Section 3.5.1 for computational results showing the benefit of jointly predicting and optimizing. This is important because runtime is often the main restriction when building tree ensemble models like random forests, as these models can be take a long time to train. For models that are only built to then optimize over, combining the training and optimizing process in such a manner would allow for tractability gains over large datasets and tree ensembles. In the next section we discuss the quality of UMOTEM's optimality gap in expectation.

3.4 Bounds on the Quality of Approximation for Random Forests

In this section we bound the expected loss from using UMOTEM. This loss is calculated in terms of the value of the objective function when compared to the true MIO formulation specifically for random forests. We do this by analyzing the expected difference between the solution found by the exact MIO and the optimal solution at each iteration of UMOTEM's algorithm. We define this bound in terms of two features of the random forest: the quality of the in-sample fit and how disparate the payoffs of the top leaves in the forest are. This makes intuitive sense because if the forest is very well learned (ie small in-sample error), it will be easier for UMOTEM to find the optimal solution. Similarly, if the difference between the payoffs of the top leaves is large, especially compared to the in-sample error, then UMOTEM will be more likely to find the optimal solution. The proof follows by first showing that as the number of trees goes to infinity, both the UB function and the random forest prediction

function converge almost surely to expected functions over the "average" tree. We then show that these expected functions differ from each other by a finite, quantifiable amount. Finally we use this result to calculate the expectation by integrating over the CDF of the difference between functions.

1. Optimality Gap Definition To accomplish this we start by defining three functions: $h(x)$, $u^d(x)$, and $h^{*d}(x)$. $h(x)$ is the random forest prediction function for a feature vector x. For any data point x_k with realized payoff y_k in the training set, we can also write $y_k = h(x_k) + \epsilon_k$, where ϵ_k is the error term. $u^d(x)$ is what the UB function estimates as the upper-bound of the prediction for feature vector x at depth d of the random forest. We index by the depth because at each iteration of the algorithm, as we move deeper into the tree, this estimate will change. Finally $h^{*d}(x)$ represents the highest prediction that can be achieved by feature vector x at depth d. At depth d, we do not yet know exactly which leaves feature vector x will end in, but we can define $h^{*d}(x)$ as the best possible payoff among the leaves that are children of the node that contains x at depth d. We define each of these functions for the forest as a whole, but we also define the functions for individual trees, t: $h_t(x)$, $u_t^d(x)$ and $h_t^{*d}(x)$.

To illustrate the difference between these functions, we again reference the sample forest in Figure 3.1. We assume for the rest of this section that we want to optimize the sales with respect to the discount rate for an item whose price is set to \$19. At depth one, or during the first iteration of UMOTEM, $u_t^d(x)$ and $h_t^{*d}(x)$ would be approximated as Figure 3.2.

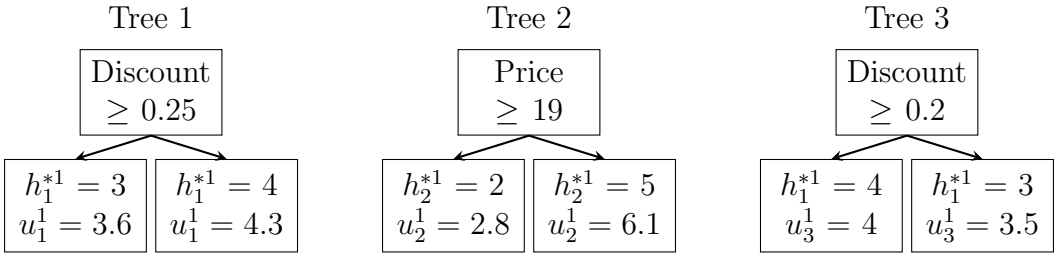


Figure 3.2: Depth 1 Approximation of Sample Random Forest 1

For this section we assume that the data lies in a bounded set and the random forest is well-learned. Here we use the term well-learned to imply that the expectation of the in-sample error is zero. This means that for any data point x_k with payoff y_k, it can be described as its prediction in the random forest plus a bounded noise term: $y_k = h(x_k) + \epsilon_k$. Because the data is bounded, the noise term is also bounded between $[-\delta, \delta]$, where δ is the magnitude of the largest in-sample error. As the random forest is well learned, we also assume that $E[\epsilon_k] = 0$.

We define x_u^{*d} as the optimal solution when optimizing with $u^d(x)$ as the objective function. Similarly, we define x_h^{*d} as the optimal solution when optimizing with $h^{*d}(x)$ as the objective function. For a dataset with feature matrix $\mathbf{X}$ and outcome vector Y, and a random forest with N trees, we aim to bound: $E_{\mathbf{X},Y}\left[h^{*d}(x_u^{*d}) - h^{*d}(x_h^{*d}) | u^d(*)\right]$. Namely, we optimize the difference in terms of the best payoff between the solution found from optimizing along $u^d(x)$ and $h^{*d}(x)$, given knowledge of the entire function $u^d(*)$. This represents the bound as it would be calculated in practice because we will have full knowledge of the

UB function, $u^d(*)$, and seek to understand what our optimality gap uncertainty is. It is worth noting that if $u^d(x) = h^{*d}(x)$, then the payoff found by UMOTEM would be the same as the true MIO, because at each iteration the algorithm would pick the true optimal child branch to go to. We estimate $E_{\mathbf{X},Y}\left[h^{*d}(x_u^{*d}) - h^{*d}(x_h^{*d})\right]$ rather than $E_{\mathbf{X},Y}\left[h(x_u^{*d}) - h(x_h^{*d})\right]$ because we wish to evaluate the quality of the decision made at depth d. For example, if we consider the sample forest shown in Figure 3.1, in Tree 3, it matters whether the Discount variable is greater or less than 0.2 because this decides whether we go to the left or right child of the root node. It does not matter, however whether or not the optimization selects the Discount variable to be between 0.2 and 0.1 or less than 0.1. Whether the Discount variable is greater or less than 0.1 will be considered at the next iteration of the algorithm if the right branch of the tree is explored.

2. Random Forest Characteristics In order to bound the optimality gap, we consider two features of the random forest.

Leaf Separation: First we consider how disparate the highest leaves of the forest are. This matters because if the payoffs of the top branches are very different from each other, especially relative to the in-sample error, the lower the optimality gap is expected to be. This is because the stronger the separation between branches, the more likely it is that UMOTEM explores the correct branches of the forest. For example, compare the depth one approximation shown in Figure 3.2, with the one shown in Figure 3.3. Even though the differences between u^1 and h^{*1} are greater, the scale of the differences between the branches means that UMOTEM is likely to end up at the optimal solution. This is because even though u^d is not the same as h^{*d}, the separation between the leaves means that u^d will still accurately indicate whether the algorithm should explore the left or right branch of the tree. We quantify this by defining $x_u^{k^*d}$ as a point in the k^{th} highest payoff region of $u^d(x)$. For example, in Sample Random Forest 1 from Figure 3.2, when optimizing along the Discount variable, $x_u^{1^*1}$ would be between 0.2 and 0.25, $x_u^{2^*1}$ would be less than 0.2, and $x_u^{3^*1}$ would be greater than 0.25. We can then evaluate how disparate the top payoffs of UB estimate are by calculating $u^d(x_u^{1^*d}) - u^d(x_u^{2^*d})$. For Sample Random Forest 1 from Figure 3.2 at depth 1, this would be 0.17, while for Sample Random Forest 2 from Figure 3.3, it would be 16.3.

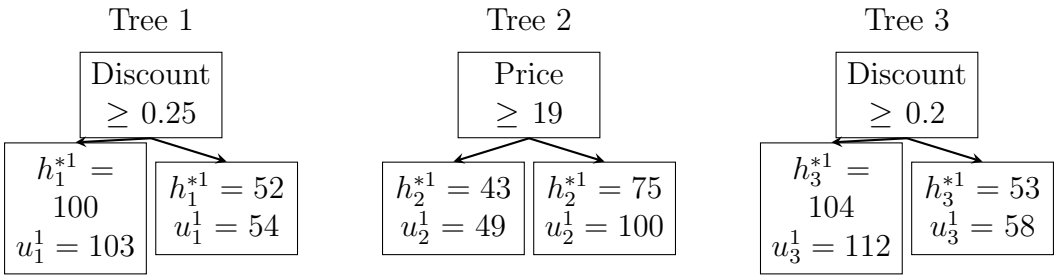


Figure 3.3: Depth 1 Approximation of Sample Random Forest 2

In-Sample Error: The second feature of the random forest that is relevant to the optimality gap is the size of the in-sample error. Two aspects are relevant, first the largest possible magnitude of the error term ϵ, which is δ, but also the expected size of the largest errors. Specifically, for the iteration of the algorithm at depth d, what matters is the size of the largest errors in the branches being considered to create the UB estimate. When

creating the estimate of $u_t^d(x)$, we average the payoffs of the m highest data points in tree t of the branch below depth d containing x. The larger these errors, the farther $u_t^d(x)$ is expected to be from $h_t^{*d}(x)$. It is worth noting that if there is no in-sample error ($\delta = 0$), then $u^d(x) = h^{*d}(x)$ and there will be no optimality gap. We begin to formally state this relationship by first defining $\mathcal{D}_t^d(x)$ as the set of training data points which fall into descendent leaves of the node containing x at depth d. We define $\epsilon_{(k)}$ as the order statistic on the noise term, or the k^{th} largest ϵ. Then we can say that our confidence in the UB estimate of $u^d(x)$ is highly correlated with the size of $\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$. More concretely, because the UB estimate consists of the m largest data points in a region, where m is the minimum leaf size of the random forest, our largest expected error on $u_t^d(x)$ can be approximated by $\frac{1}{m}\sum_{k=1}^{m}\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$. The larger this term, the more likely the UB function is to select the less optimal branch to explore, and thus, the higher the expected optimality gap.

3. Bounds on Sub-Optimality Gap Given the leaf separation and in-sample error of the random forest, $u_t^d(x_u^{1^*d})$, $u_t^d(x_u^{2^*d})$, δ, and $\frac{1}{m}\sum_{k=1}^{m}\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$, we can now state the bound on our optimality gap. As the intuition above confirms, in the optimality gap, we have that as $u_t^d(x_u^{1^*d}) - u_t^d(x_u^{2^*d})$ grows, the bound on the optimality gap in expectation gets smaller. Similarly, as δ and $\frac{1}{m}\sum_{k=1}^{m}\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$ get smaller, the bound on the optimality gap in expectation also shrinks. We state this relationship formally in Theorem 3.4:

Let $u_t^d(x)$ and $h_t^{*d}(x)$ be the upper bound estimate and highest prediction value respectively of feature vector x for tree t at depth d. We define the optimal solution when optimizing with objective function $u^d(x)$ as x_u^{*d}. Similarly, we define x_h^{*d} as the optimal solution when optimizing with objective function $h^{*d}(x)$. Let E_Θ indicate the expectation over the random sample which generates the trees in the random forest and let $E_{\mathbf{X},Y}$ indicate the expectation over the $\mathbf{X}, Y$ feature and payoff space. Let m be the minimum leaf size in the forest. We define $\epsilon_{t,(k)}^d$ as $\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$ or the k^{th} largest noise term of error ϵ in the set of points, $\mathcal{D}_t^d(x)$, which fall into children leaves of the node containing x at depth d of tree t. ϵ is bounded in $[-\delta, \delta]$. Finally, we define $x_u^{k^*d}$ as a point in the k^{th} most optimal region of $u^d(x)$ so that $x_u^{1^*d}$ has the highest value of $u^d(x)$ and $x_u^{2^*d}$ has the second highest. Then it follows that:

1. As the number of trees goes to infinity, the optimality gap, $E_{\mathbf{X},Y}\left[h^{*d}(x_u^{*d}) - h^{*d}(x_h^{*d})|u^d(*)\right]$, is bounded from below almost surely by:

$$-\int_{-\delta}^{0}\exp\left(-\frac{2\left(\overbrace{E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right]}^{\text{The separation in estimated UB values}} - \overbrace{E\left[\frac{1}{m}\sum_{k=1}^{m}E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right]}^{\text{Expected size of the largest errors}} - b\right)^2}{\underbrace{\delta^2}_{\text{Maximum in-sample error}}}\right)db$$

2. For a finite forest of size T, the optimality gap, $E_{\mathbf{X},Y}\left[h^{*d}(x_u^{*d}) - h^{*d}(x_h^{*d})|u^d(*)\right]$, is bounded from below by:

$$-\int_{-\delta}^{0}\exp\left(-\frac{2\left(\frac{1}{T}\sum_{t=1}^{T}\left(u_t^d(x_u^{1^*d})\right)-\frac{1}{T}\sum_{t=1}^{T}\left(u_t^d(x_u^{2^*d})\right)-E\left[\frac{1}{mT}\sum_{k=1}^{m}\sum_{t=1}^{T}\left(\epsilon_{t,(k)}^d\right)\right]-b\right)^2}{\delta^2}\right)db.$$

Theorem 3.4 provides a worst case bound on the loss we experience by using the UB approximation function rather than the true objective function at each iteration of UMOTEM. This bound is dependent on the two key characteristics of the random forest that we have discussed. First, the larger the separation between the estimated UB values of different regions, the more confident we are that there will be a low optimality gap. This is intuitive because if the top estimates of UB values are very different, especially relative to the maximum error of the points, δ, then it is likely we have selected the optimal region to explore. Second, the better the wellness of fit of the random forest, the more confident we are in the solution found by UMOTEM. This is represented both by the expected size of the largest errors in the nominator and the maximum possible error represented in the denominator. The smaller either of these values, the smaller the expected optimality loss.

Proof of Theorem 3.4:

We start by showing that as the number of trees in the forest goes to infinity, the prediction of the random forest and the UB function converge almost surely to an expected function representing the average tree. This is proved in Step 3.4.

Let $h_t^{*d}(x)$ be the highest prediction value feature vector x can achieve from depth d in tree t. Let $u_t^d(x)$ be the upper bound estimate of the y value from tree t on feature vector x at depth d. Let E_Θ indicate expectation over the random sample which generates the trees in the random forest. Then as the number of trees in the forest, T, goes to infinity, $\frac{1}{T}\sum_{t\in T}h_t^{*d}(x)$ goes almost surely to $E_\Theta\left[h_\Theta^{*d}(x)\right]$ and $\frac{1}{T}\sum_{t\in T}u_t^d(x)$ goes almost surely to $E_\Theta\left[u_\Theta^d(x)\right]$.

The proof follows directly from [35] where instead of evaluating the convergence of the random forest prediction $h(x)$, we evaluate the highest prediction at depth d, $h^{*d}(x)$, and the UB estimate $u^d(x)$. However, for the sake of completeness, we provide the proof in the Appendix.

Step 3.4 shows convergence to an expectation; however there is also a relationship between $h^{*d}(x)$ and $u^d(x)$, and by extension $E_\Theta\left[h_\Theta^{*d}(x)\right]$ and $E_\Theta\left[u_\Theta^d(x)\right]$. By definition, we have that $h^{*d}(x)\leq u^d(x)\quad\forall x$, because the UB function $u^d(x)$ uses the top m data points from the region which x falls into to make the estimate, and there are at least m data points defining $h^{*d}(x)$ from the same region. As its lowest value, $u^d(x)$ will just use the same points that define $h^{*d}(x)$. We can also bound $u^d(x)$ from above when we consider that the highest value it can possibly take is $h^{*d}(x)+f(\epsilon)$ where $f(\epsilon)$ is some function on the in-sample noise term. This is because the data points which create $u^d(x)$ can only deviate from their leaf's payoff by ϵ. We formalize these upper and lower bounds on $E_\Theta\left[u_\Theta^d(x)\right]$ in Step 3.4.

Let $u_t^d(x)$ and $h_t^{*d}(x)$ be the upper bound estimate and highest prediction value of feature vector x respectively for tree t at depth d. Let the subscript Θ indicate expectation over the random sample which generates the trees in the random forest. Let m be the minimum leaf size in the forest and we define $\epsilon_{t,(k)}^d$ as $\epsilon_{(k):k\in\mathcal{D}_t^d(x)}$, or that is the k^{th} largest noise term, ϵ, in the set of points, $\mathcal{D}_t^d(x)$, which fall into children leaves of the node containing x at depth d of tree t. Then we have:

$$E_\Theta\left[h^{*d}_\Theta(x)\right] \leq E_\Theta\left[u^d_\Theta(x)\right] \leq E_\Theta\left[h^{*d}_\Theta(x)\right] + \frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]. \tag{3.5}$$

From our initial assumption, every data point in the training set can be described as its prediction from the random forest plus some noise: $y_k = h(x_k) + \epsilon_k$, where $E[\epsilon_k] = 0$ and $\epsilon_k \in [-\delta, \delta]$. Because the upper bound at depth d does not know the specific leaf for which x falls into, it takes the top m points from the union of leaves which stem from the node at depth d which contains x. Let $\mathcal{D}^d_t(x)$ be the set of all points considered to build tree t which fall into a child leaf of the node at depth d which contains x.

Therefore the upper bound function can be described as:

$$\begin{aligned} u^d_t(x) &= \frac{1}{m}\sum_{k=1}^{m} y_{(k):k\in\mathcal{D}^d_t(x)} \\ &= \frac{1}{m}\sum_{k=1}^{m} h\left(x_{(k):k\in\mathcal{D}^d_t(x)}\right) + \epsilon_{(k):k\in\mathcal{D}^d_t(x)}. \end{aligned}$$

The subscript (k) denotes the order statistic, indicating that we are averaging over the m largest payoffs in set $\mathcal{D}^d_t(x)$. We can upper and lower bound this amount.

The lower bound for this function is straightforward. By definition, it cannot be less than payoff of the highest leaf which stems from the node containing x at depth d. Therefore we have: $u^d_t(x) \geq h^{*d}_t(x)$. When we take the limit of both sides as the number of trees goes to infinity, we also have: $E_\Theta\left[u^d_\Theta(x)\right] \geq E_\Theta\left[h^{*d}_\Theta(x)\right]$.

We can take an upper bound as follows:

$$\begin{aligned} u^d_t(x) &= \frac{1}{m}\sum_{k=1}^{m} h\left(x_{(k):k\in\mathcal{D}^d_t(x)}\right) + \epsilon_{(k):k\in\mathcal{D}^d_t(x)} \\ &\leq \frac{1}{m}\sum_{k=1}^{m} h^{*d}(x) + \epsilon_{(k):k\in\mathcal{D}^d_t(x)} \\ &= h^{*d}(x) + \frac{1}{m}\sum_{k=1}^{m} \epsilon_{(k):k\in\mathcal{D}^d_t(x)}. \end{aligned}$$

The first inequality comes from the fact that the highest random forest prediction that can be achieved for points in $\mathcal{D}^d_t(x)$ is by definition $h^{*d}(x)$. As the number of trees goes to infinity, we have:

$$\begin{aligned} E_\Theta\left[u^d_\Theta(x)\right] &\leq \lim_{T\to\infty}\left(h^*(x) + \frac{1}{m}\sum_{k=1}^{m}\frac{1}{T}\sum_{t=1}^{T}\epsilon_{(k):k\in\mathcal{D}^d_t(x)}\right) \\ &= E_\Theta\left[h^*_\Theta(x)\right] + \frac{1}{m}\sum_{k=1}^{m}\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\left(\epsilon_{(k):k\in\mathcal{D}^d_t(x)}\right). \end{aligned}$$

We have that $\frac{1}{T}\sum_{t=1}^{T}\left(\epsilon_{(k):k\in\mathcal{D}^d_t(x)}\right)$ converges almost surely to $E_\Theta\left[\epsilon_{(k):k\in\mathcal{D}^d_\Theta(x)}\right]$. This follows directly from the proof of Step 3.4, but rather than taking j as the average across the

hyper-rectangle, it is the k^{th} largest noise term. This gives us: $E_\Theta\left[u^d_\Theta(x)\right] \leq E_\Theta\left[h^*_\Theta(x)\right] + \frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon_{(k):k\in\mathcal{D}^d_\Theta(x)}\right]$.

For ease of notation we will define $E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]$ as $E_\Theta\left[\epsilon_{(k):k\in\mathcal{D}^d_\Theta(x)}\right]$, giving us the equation.

Step 3.4 demonstrates that the difference between $E_\Theta\left[h^{*d}_\Theta(x)\right]$ and $E_\Theta\left[u^d_\Theta(x)\right]$ is at most $\frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]$. This makes sense because the upper bound is limited by the in-sample error. The smaller the ϵ, the closer the upper bound will be to the true value of the highest leaf. Using this relationship, we can bound the optimality gap $E_{\mathbf{X},Y}\left[h^{*d}(x^{*d}_u) - h^{*d}(x^{*d}_h)|u^d(*)\right]$ as the number of trees goes to infinity, or $E_{\mathbf{X},Y}\left[E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right] - E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right]|u^d(*)\right]$.

$$
\begin{aligned}
&E_{\mathbf{X},Y}\left[E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right] - E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right]|u^d(*)\right] \\
&= -\int_{-\infty}^0 P\left(E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right] - E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right] \leq b|u^d(*)\right) db \\
&\geq -\int_{-\infty}^0 P\left(E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - \frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right] - E_\Theta\left[u^d_\Theta(x^{*d}_h)\right] \leq b|u^d(*)\right) db \\
&\geq -\int_{-\delta}^0 P\left(E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - \frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right] - E_\Theta\left[u^d_\Theta(x^{*d}_h)\right] \leq b|u^d(*)\right) db
\end{aligned}
$$

We obtain the first equality from the fact that this expectation is strictly non-positive. This is because by definition the optimal solution from the upper bound on the random forest prediction cannot be greater than the true optimal solution. Given this, we have that for any non-positive random variable X with CDF $F(x)$, the following holds: $EX = -\int_{-\infty}^0 F(x)dx$.

We obtain the first inequality from taking the worst case scenario (ie the case where the probability will be the highest), which is when $E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right]$ is as small as possible and $E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right]$ is as large as possible. We use Step 3.4 to obtain that $E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right] \geq E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - \frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]$, where $E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]$ is $E_\Theta\left[\epsilon_{(k):k\in\mathcal{D}^d_\Theta(x^{*d}_u)}\right]$. Similarly, we use Step 3.4 to obtain $E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right] \leq E_\Theta\left[u^d_\Theta(x^{*d}_h)\right]$.

We obtain the second inequality from noting that the minimum value that $E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - \frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right] - E_\Theta\left[u^d_\Theta(x^{*d}_h)\right]$ can take is $-\delta$. This is because by definition $E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - E_\Theta\left[u^d_\Theta(x^{*d}_h)\right] \geq 0$ and $-\frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right] \geq -\delta$. Therefore if $b < -\delta$, this probability becomes 0. We then need to only integrate the probability from $-\delta$ to 0.

Given that we have knowledge of the entire UB function and the only source of randomness is $\frac{1}{m}\sum_{k=1}^m E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]$, we pull all the constants of the equation to the right side of the inequality. We now want to find a lower bound on $E_\Theta\left[u^d_\Theta(x^{*d}_u)\right] - E_\Theta\left[u^d_\Theta(x^{*d}_h)\right] - b$. We start by sorting the possible optimal solutions of the function $u^d(x)$. Let $x^{k^*d}_u$ be a point in the k^{th} most optimal region of $u^d(x)$. We have that $x^{1^*d}_u$ and x^{*d}_u are synonymous. If x^{*d}_h also pointed to the same region as $x^{1^*d}_u$, then there would be no optimality gap because the UB estimate of the best branch to explore and the best random forest leaf would be in the same region. Instead we have the highest probability for an optimality gap when x^{*d}_h falls into the region characterized by $x^{2^*d}_u$. Therefore, the inequality can be rewritten when we substitute

$x_u^{1^*d}$ as x_u^{*d} and bound x_h^{*d} with $x_u^{2^*d}$. We use $u^d(*)$ to represent knowledge of the entire UB function.

$$
\begin{aligned}
&= -\int_{-\delta}^{0} P\left(E_\Theta\left[u_\Theta^d(x_u^{*d})\right] - \frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right] - E_\Theta\left[u_\Theta^d(x_h^{*d})\right] \le b | u^d(*)\right) db \\
&= -\int_{-\delta}^{0} P\left(\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right] \ge E_\Theta\left[u_\Theta^d(x_u^{*d})\right] - E_\Theta\left[u_\Theta^d(x_h^{*d})\right] - b | u^d(*)\right) db \\
&\ge -\int_{-\delta}^{0} P\left(\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right] \ge E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right) db
\end{aligned}
$$

We then apply the monotonic function $f(x) = \exp(\tau x)$, where τ is a positive constant to make both sides of the inequality non-negative. Because both sides are of the probability inequality are non-negative, we can apply Markov's inequality to obtain the last inequality:

$$
\begin{aligned}
&= -\int_{-\delta}^{0} P\left(\exp\left(\tau\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right) \ge \exp\left(\tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right)\right)\right) db \\
&\ge -\int_{-\delta}^{0} \frac{E\left[\exp\left(\tau\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right)\right]}{\exp\left(\tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right)\right)} db.
\end{aligned}
$$

The random variable $\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]$ is bounded between 0 and δ because each of the ϵ terms are at most δ and we know the sum as a whole is strictly non-negative. Therefore, we can use Hoeffding's lemma to bound the nominator of the fraction. We then bring the denominator of the fraction up into the exponent:

$$
\begin{aligned}
&-\int_{-\delta}^{0} \frac{E\left[\exp\left(\tau\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right)\right]}{\exp\left(\tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right)\right)} db \\
&\ge -\int_{-\delta}^{0} \frac{\exp\left(\tau E\left[\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right] + \frac{\tau^2\delta^2}{8}\right)}{\exp\left(\tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right)\right)} db \\
&= -\int_{-\delta}^{0} \exp\left(\frac{\tau^2\delta^2}{8} - \tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - E\left[\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right] - b\right)\right) db.
\end{aligned}
$$

Therefore, we have that, for all positive values of τ, we have:

$$
\begin{aligned}
&E_{\mathbf{X},Y}\left[E_\Theta\left[h_\Theta^{*d}(x_u^{*d})\right] - E_\Theta\left[h_\Theta^{*d}(x_h^{*d})\right] | u^d(*)\right] \\
&\ge -\int_{-\delta}^{0} \exp\left(\frac{\tau^2\delta^2}{8} - \tau\left(E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - E\left[\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right]\right] - b\right)\right) db.
\end{aligned}
$$

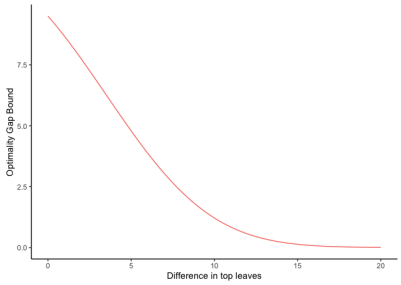
We find the tightest bound possible by minimizing the RHS over τ to obtain:

$$E_{\mathbf{X},Y}\left[E_\Theta\left[h^{*d}_\Theta(x^{*d}_u)\right] - E_\Theta\left[h^{*d}_\Theta(x^{*d}_h)\right]|u^d(*)\right]$$
$$\geq -\int_{-\delta}^{0}\exp\left(-\frac{2\left(E_\Theta\left[u^d_\Theta(x^{1^*d}_u)\right] - E_\Theta\left[u^d_\Theta(x^{2^*d}_u)\right] - E\left[\frac{1}{m}\sum_{k=1}^{m}E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]\right] - b\right)^2}{\delta^2}\right)db.$$

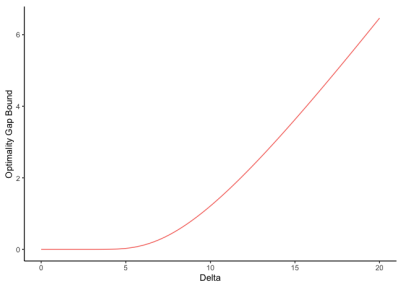
The second part of the theorem follows from the same proof, but without first taking the limit as the number of trees goes to infinity. The rest of the proof follows similarly.

We can also plot the bound in terms of each of its components to see the quality of the bound for different settings. In Figures 3.4a, 3.4b, and 3.4c we show how the optimality gap changes for different values of the difference in top leaves (ie $E_\Theta\left[u^d_\Theta(x^{1^*d}_u)\right] - E_\Theta\left[u^d_\Theta(x^{2^*d}_u)\right]$), the maximum in-sample error (ie δ), and the expectation of the top m errors (ie $E\left[\frac{1}{m}\sum_{k=1}^{m}E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]\right]$) respectively. Unless the value is changing over the x-axis, we assume that in this forest, the difference between the top leaves is 10, the maximum error is 10, and the expectation of the top m errors is 3.5. We see that as the difference in the top leaves increases or as the maximum error decreases, the optimality gap drops off sharply. The drop off is not as steep for the expectation of the top m errors, but as the expectation of the errors decreases, so does the optimality bound.

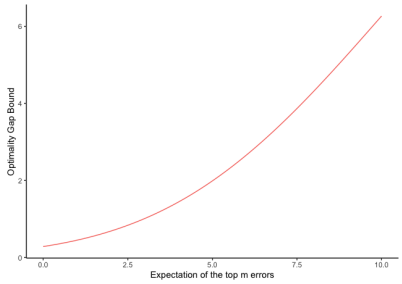
Figure 3.4: Optimality Gap over different features of the random forest



(a) Optimality Gap over Differences in top leaves ($E_\Theta\left[u^d_\Theta(x^{1^*d}_u)\right] - E_\Theta\left[u^d_\Theta(x^{2^*d}_u)\right]$)



(b) Optimality Gap over Maximum in-sample error (δ)



(c) Optimality Gap over Expectation of the top m errors ($E\left[\frac{1}{m}\sum_{k=1}^{m}E_\Theta\left[\epsilon^d_{\Theta,(k)}\right]\right]$)

The structure on Theorem 3.4 also allows us to gain insights into what specific situations UMOTEM is guaranteed to reach optimality. The first scenario, where UMOTEM is guaranteed to have no optimality gap, is the ideal situation where a tree is perfectly learned. This is because, when a tree is perfectly learned, the UB function will directly reflect the true predictive function of the random forest. Therefore, the UB function will always indicate correctly whether the optimal leaf lies in the left or right child branch, resulting in UMOTEM always resulting in optimality. This is proved in Corollary 3.4:

When the trees in an ensemble are perfectly learned, meaning there is no in-sample error, then UMOTEM finds an optimal solution.

Proof of Corollary 3.4: In the case where there is no in-sample error for the trees making up the ensemble, then δ, the size of the largest error will be zero. This means that for any

depth d and point i, $0 = -\delta \leq \epsilon_i^d \leq \delta = 0$, meaning that all $\epsilon = 0$. In this case it follows Step 2 of the Proof of Theorem 3.4, that $E_\Theta\left[h_\Theta^{*d}(x)\right] \leq E_\Theta\left[u_\Theta^d(x)\right] \leq E_\Theta\left[h_\Theta^{*d}(x)\right]$, meaning that $E_\Theta\left[h_\Theta^{*d}(x)\right] = E_\Theta\left[u_\Theta^d(x)\right]$. Because both UMOTEM and the true MIO are subject to the same constraints, and they are now optimizing along the same objective function, we will have that x_u^{*d} will be in the same child leaf as x_h^{*d}, meaning there will be no optimality gap.

The second scenario that is of interest is when the size of the in-sample error is less than the separation between the top regions of the upper bound function. In this case, again we will have that the optimality gap is equal to zero. This is because if the error is less than the separation, then while the UB function will most likely not directly equal the random forest prediction function, the probability that the UB function will indicate a leaf different than what is optimal is still zero. This is discussed in Corollary 3.4.

Let $u_t^d(x)$ and $h_t^{*d}(x)$ be the upper bound estimate and highest prediction value respectively of feature vector x for tree t at depth d. We define $x_u^{k^*d}$ as a point in the k^{th} most optimal region of $u^d(x)$, so that $x_u^{1^*d}$ has the highest value of $u^d(x)$ and $x_u^{2^*d}$ has the second highest. Finally let ϵ_i be the in-sample error of data point i. ϵ lies in $[-\delta, \delta]$. Then UMOTEM will not have an optimality gap if: $\delta \leq u_t^d\left(x_u^{1^*d}\right) - u_t^d\left(x_u^{2^*d}\right)$.

Proof of Corollary 3.4: We have that if $\delta \leq u_t^d\left(x_u^{1^*d}\right) - u_t^d\left(x_u^{2^*d}\right)$ then, as we consider the case when the number of trees goes to infinity, we will also have that $\delta \leq E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right]$.

During the proof of Theorem 3.4, we show that the optimality gap,

$$E_{\mathbf{X},Y}\left[E_\Theta\left[h_\Theta^{*d}(x_u^{*d})\right] - E_\Theta\left[h_\Theta^{*d}(x_h^{*d})\right] | u^d(*)\right]$$

, is bounded by:

$$\begin{aligned}
&E_{\mathbf{X},Y}\left[E_\Theta\left[h_\Theta^{*d}(x_u^{*d})\right] - E_\Theta\left[h_\Theta^{*d}(x_h^{*d})\right] | u^d(*)\right] \\
&\geq -\int_{-\infty}^{0} P\left(\frac{1}{m}\sum_{k=1}^{m} E_\Theta\left[\epsilon_{\Theta,(k)}^d\right] \geq E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right] - b\right) db.
\end{aligned}$$

However, we have that $\epsilon_{\Theta,(k)}^d \leq \delta \leq E_\Theta\left[u_\Theta^d(x_u^{1^*d})\right] - E_\Theta\left[u_\Theta^d(x_u^{2^*d})\right]$, for all ϵ, which means the probability in the equation above comes out to be zero. This means that:

$$E_{\mathbf{X},Y}\left[E_\Theta\left[h_\Theta^{*d}(x_u^{*d})\right] - E_\Theta\left[h_\Theta^{*d}(x_h^{*d})\right] | u^d(*)\right] = 0,$$

resulting in no loss from using the UB function, or in other words, no optimality gap.

We see in these cases where either the error is less than the separation between the top regions of the upper bound function, such as when the trees are perfectly learned, the bound is tight. From this we not only bound the optimality gap in expectation of our approximation, but we also gain some insight into which situations UMOTEM is expected to perform well, or even obtain perfect optimality. In particular, we see that there is a close tie between the quality of the predictive model and the quality of the prescriptive decision. The better trained the ensemble model, or in other words, the smaller the in-sample error, the more confident we are that UMOTEM has found optimality. This is different from the bounds provided by the approximations in [32] and [33]. Both approximations have analytical

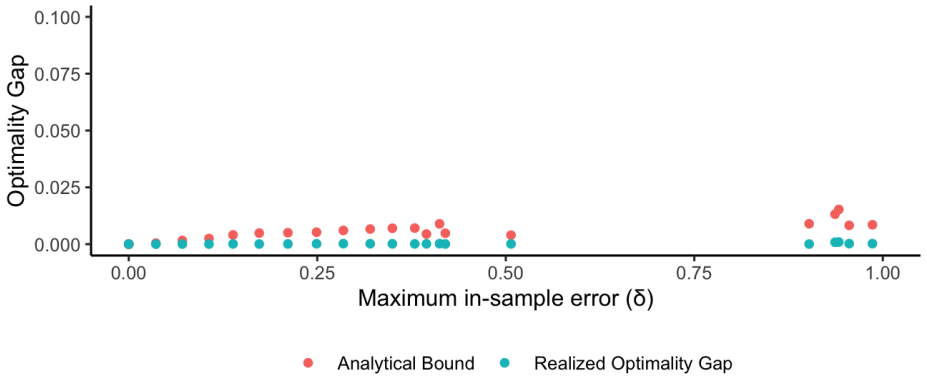


Figure 3.5: Analytical and realized optimality gap

guarantees, however they decay as leaf separation increases and have no dependency on how well the model fits the data. Therefore, when leaf payoffs are clustered close together or when the in-sample error is high, then either the depth approximation from [32] or the sampling approximation from [33] would have a better analytical guarantee. However, when stakes are higher and there is significant leaf separation, or when the model better fits the data, then UMOTEM will exhibit the better optimality gap.

We show this more visually using synthetic data. Here we generate data from an underlying tree structure across five features, with leaf payoffs ranging between -250 and 250. Noise is generated from a normal distribution with mean zero and added to the labels. As the variance of the noise is increased, the maximum in-sample error (δ), the analytical bound derived from Theorem 3.4, and the realized optimality gap of UMOTEM are all tracked. The results are plotted in Figure 3.5. Each generated version of the data has two corresponding points on the graph. These points are aligned vertically because they share the same maximum in-sample error (δ). First, in red, is the analytical bound for the data and the model built on it. This is calculated using Theorem 3.4 and the specific features of the data (such as maximum in-sample error). The second point, in blue, is the realized optimality gap, again as a fraction of optimality, found when UMOTEM was run on the dataset. We see that when the maximum in-sample error (δ) is small, the bound is tight. In the next section we will discuss how UMOTEM performs in practice with computational results.

3.5 Results from the Data

3.5.1 General Machine Learning Datasets

In this section we describe our computational results which show how UMOTEM performs in practice. In order to test the algorithm we build several sets of random forests to optimize over. All models and formulations are built in the Julia programming language. The random forests are created using the package `DecisionTree`. Trees in the random forest are built with a minimum node size of 5, and considering one third of the available features and all the data points. Optimization formulations are written using `JuMP` (Julia for Mathematical

Programming) and solved using Gurobi.

In this section we specifically compare two approaches to optimizing over a random forest. First we implement Formulations (3.1) and (B.2), introduced by [32] and [33] as a baseline for what true optimality and time to solve would be. We then compare this to UMOTEM and the Predict and Optimize formulation both in terms of optimality found and runtime. We test the algorithms on three publicly available datasets, from the University of California Irvine machine learning repository and R packages, detailed below:

- `Solubility`: [54], [55] Predict a compound's solubility.
  Features: 228, Data Points: 951, Average Tree Depth: 17
- `Concrete`: [56] Predict the compressive strength of concrete formulations.
  Features: 9, Data Points: 1,030, Average Tree Depth: 15
- `Wine`: [57] Predict the quality of Portuguese red wines.
  Features: 12, Data Points: 4,898, Average Tree Depth: 18

For each of these datasets, we solve the unconstrained optimization to find a feature vector x which maximizes the prediction of the random forest. We vary the number of trees from 10 to 500 to show how each formulation grows with the size of the forest. For each dataset and forest size we evaluate:

- S^*_{MIO}: the value of the optimal objective function found by Formulation (3.1) and (B.2). This is the benchmark for true optimality
- S^*_{UMOTEM}: the value of the optimal objective function found by UMOTEM
- Opt. Gap: The optimality gap as defined by $100\% * \frac{S^*_{\text{MIO}} - S^*_{\text{UMOTEM}}}{S^*_{\text{MIO}}}$, the percentage difference between the solution found by UMOTEM and true optimality
- Time (Mišić): the time (in seconds) to solve the MIO, Formulation (3.1), using the proposed Benders Decomposition from [32]
- Time (Biggs et al.): the time (in seconds) to solve the MIO, Formulation (B.2), using the proposed Benders Decomposition from [33]
- Time (UMOTEM): the time (in seconds) to solve UMOTEM
- Time (Pred. and Opt.): the time (in seconds) to solve Predict and Optimize formulation

In cases where the formulation was too large for the system's memory to compute, we label both the objective function and time as *NA*.

Table 3.2 summarizes how UMOTEM and Formulations (3.1) and (B.2) compare both in optimality and time to solve. For the full set of results, please see Tables B.1, B.2 and B.3 in the Appendix. We see that across the board, UMOTEM and Predict and Optimize reach near optimality in nearly negligible amounts of time compared to the true MIO. We see the linearity of the number of binary variables in UMOTEM's formulation reflected in

Dataset	# trees	S^*_{MIO}	S^*_{UMOTEM}	Opt. Gap	Time (Mišić)	Time (Biggs et al.)	Time (UMOTEM)	Time (Pred. and Opt.)
Solubility	100	1.13	1.12	0.8%	28.35	NA	3.89	2.68
	200	1.13	1.11	1.2%	50.98	NA	8.85	6.70
	300	1.13	1.12	0.7%	100.98	NA	12.11	8.40
	400	1.13	1.11	0.9%	136.00	NA	14.56	9.02
	500	1.12	1.11	1.0%	273.53	NA	20.07	12.95
Concrete	100	78.25	76.83	1.8%	14.31	454.15	0.42	0.22
	200	78.21	76.78	1.8%	28.35	NA	0.92	0.39
	300	78.20	76.72	1.9%	44.40	NA	1.30	0.49
	400	78.26	76.72	2.0%	51.48	NA	1.47	0.67
	500	78.26	76.83	1.8%	83.58	NA	1.89	0.83
Wine	100	7.37	6.69	9.2%	149.70	NA	0.85	0.31
	200	7.37	6.71	9.0%	271.79	NA	1.62	0.77
	300	7.37	6.78	8.1%	503.57	NA	2.44	0.93
	400	7.36	6.75	8.2%	989.18	NA	2.98	1.16
	500	7.36	6.81	7.5%	1216.24	NA	4.83	1.54

Table 3.2: Objective function value and times

its runtime compared to the exact MIO formulation. While Formulations (3.1) and (B.2) have rapid growth in time to run, UMOTEM runs in fractions of seconds.

Both [32] and [33] propose approximations to their MIOs. Specifically [32] suggest to only consider node split constraints up to some depth d, while [33] suggest sampling a subset of the ensemble for the optimization. We compare to a 30% level approximation in Table 3.3. For the depth approximation in [32], an approximation of 30% means the optimization considers only splits which are of depth $d_{\max} * 0.3$ or less. For the sampling approximation in [33], an approximation of 30% means the optimization samples 30% of the forest. We catalogue the following:

- $S^*_{\text{DA 30\%}}$: the value of the solution found by the depth approximation of 30% from [32]
- $S^*_{\text{SA 30\%}}$: the value of the solution found by the sampling approximation of 30% from [33]
- Opt $\text{Gap}_{\text{DA 30\%}}$: The optimality gap as defined by $100\% * \frac{S^*_{\text{MIO}} - S^*_{\text{DA 30\%}}}{S^*_{\text{MIO}}}$, the percentage difference between the solution found by the depth approx and true optimality
- Opt $\text{Gap}_{\text{SA 30\%}}$: The optimality gap as defined by $100\% * \frac{S^*_{\text{MIO}} - S^*_{\text{SA 30\%}}}{S^*_{\text{MIO}}}$, the percentage difference between the solution found by the sampling approx and true optimality
- $\text{Time}_{\text{DA 30\%}}$: the time (in seconds) to solve the depth approximation at 30%
- $\text{Time}_{\text{SA 30\%}}$: the time (in seconds) to solve the sampling approximation at 30%

We show more in-depth comparisons against these approximations for all datasets in Tables B.4-B.15 in the Appendix. What we find is that while the approximations have the potential to provide high-quality tractable solutions, they have risk of producing sub-optimal solutions if the approximation level is mis-selected. When we compare to UMOTEM, we

see that UMOTEM is often able run faster while providing a solution of equal quality if not better than either approximation. While an approximation of a higher level (such as 50% or 80%) would also produce higher results, the higher the approximation the closer it will scale to the original MIO, risking tractability challenges. The challenge for these approximations is selecting the level of approximation. It is also worth noting that we implemented both the original proposed approximation and the approximation with Bender's Decomposition. For both cases, the original approximations do not scale to forests of reasonable size and it is only with Bender's Decomposition are we able to get the results we show in Tables 3.2 and 3.3.

Dataset	# trees	$S^*_{\text{DA 30\%}}$	$S^*_{\text{SA 30\%}}$	Opt $\text{Gap}_{\text{DA 30\%}}$	Opt $\text{Gap}_{\text{SA 30\%}}$	$\text{Time}_{\text{DA 30\%}}$	$\text{Time}_{\text{SA 30\%}}$
Solubility	100	-4.13	0.47	467.2%	58.6%	12.77	2367.50
	200	-4.05	NA	459.7%	-	14.11	NA
	300	-3.96	NA	451.7%	-	16.92	NA
	400	-0.75	NA	166.6%	-	27.24	NA
	500	-0.64	NA	156.9%	-	26.53	NA
Concrete	100	53.97	77.27	31.0%	1.3%	7.58	35.36
	200	48.52	78.21	38.0%	0.0%	9.57	121.37
	300	42.27	78.12	45.9%	0.1%	11.58	186.10
	400	50.50	78.00	35.5%	0.3%	13.19	589.37
	500	54.68	78.13	30.1%	0.2%	12.85	2090.88
Wine	100	5.54	7.19	24.7%	2.3%	9.88	2940.74
	200	5.84	NA	20.8%	-	12.99	NA
	300	6.39	NA	13.3%	-	11.99	NA
	400	6.39	NA	13.2%	-	16.03	NA
	500	6.05	NA	17.8%	-	17.10	NA

Table 3.3: Objective function value and times of approximations

We show the comparison against all levels of the approximation on the Solubility dataset in Figures 3.6 and 3.7. In Figure 3.6 we see that depth approximations of 50% or higher perform comparably to UMOTEM and the true MIO in finding optimality, however once approximations hit this level of depth their runtimes start increasing rapidly. In Figure 3.7, we see similarly that as the sampling approximation level increases, so does the quality of the solution found. However, very few sampling approximations scale up to a reasonable sized forest, and those that do have a weaker performance than UMOTEM.

Figure 3.6: Solubility objective and runtimes of UMOTEM and the depth approx. by [32]

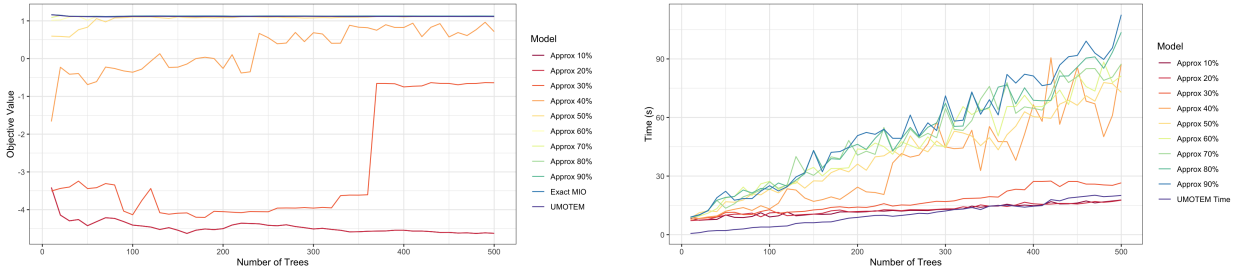
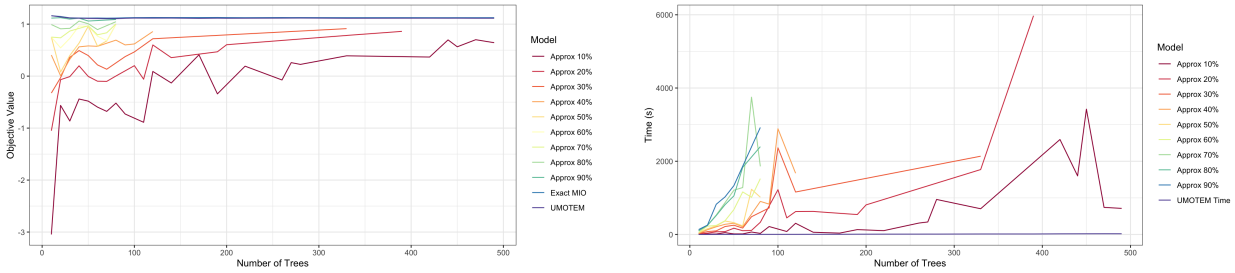
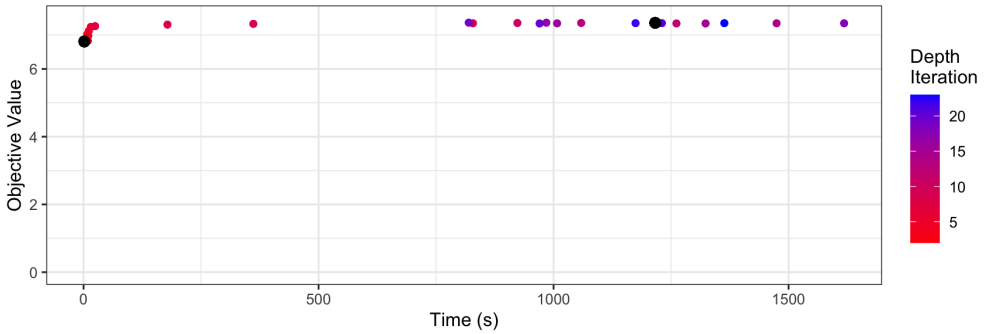


Figure 3.7: Solubility objective and runtimes of UMOTEM and the sampling approx. by [33]



Finally, the structure of UMOTEM allows to extend the method and consider optimizing over multiple depths at each iteration. As described at the end of Section 3.3.2, UMOTEM could consider instead of one level rather two, three or more levels at each iteration. The tradeoff created in this case is between optimality and runtime. As more depths are considered at each level of UMOTEM, the chances of optimality increases even further but at the price of tractability which deteriorates. We plot this trade off in Figure 3.8 for the wine dataset at 500 trees. For a zoomed version of the plot, please check Figure B.1 in the Appendix. The wine dataset was selected because, as one of the more difficult datasets for UMOTEM to optimize over, the trade-off with depth becomes clear. We have plotted in black both UMOTEM at one depth (in the left side of the graph), which has the lowest optimality achieved (at 7.5% optimality gap) but also fastest runtime, and the exact MIO as found by Formulation (3.1) with Benders Decomposition (in the right corner of the graph), which achieves optimality but with a longer runtime. In between, we plot instances of UMOTEM that consider more depths at each iteration, showing the efficient frontier of time/optimality for this forest. We observe that for depths less than 6, increasing the depth has an impact on the optimality achieved without significantly affecting runtime. Nevertheless, after that (e.g. after depth of 6), the objective function value found continues to improve until it reaches optimality, but runtime is impacted. It is also worth noting that after depth 10, there is significant randomness in how long the algorithm takes, even though all reach approximate optimality, which is why depths of UMOTEM 12, 14, 15, 18, 21 and 23 take longer than the exact MIO.

Figure 3.8: Wine, 500 trees: objective and runtimes of UMOTEM for different depth iterations



3.5.2 EV Case Study

In this section we return to the application from Chapter 2. Similar to Chapter 2, we work directly with a large American EV manufacturer to create a deterministic forecast of driver usage using a random forest and under this model, optimize the charging and discharging of EVs.

The data we use to build the forecast of drivers is the same as Section 2.5. Specifically, we pull deindentified telemetry data, which records the approximate location of the car, its mode (if it is driving, parked, charging, etc.), and battery charge level. We randomly select 1,852 EV vehicles that made trips in January and February 2021 within a major American city. Each row in the dataset represents one vehicle and one hour. Features for the dataset include data and time features (day of week, date, hour), location (at county level), vehicle information (make and model) and charge level. We also engineer lag features, such as if the car was parked in the previous time step. We then split the data into training and testing such that training is January 1, 2021 to February 15, 2021 and testing is February 16 to February 28, 2021. The training set is then used to create a model that forecasts the distance in kilometers the car drives during the hour. Using this set of features, we train several models, including two linear models, one neural network and two tree ensembles models, and evaluate out-of-sample MAE. MAE is a measure of prediction error, such that the lower the value the more accurate the model. MAE can be calculated as:

$$MAE = \frac{1}{P}\sum_{t=1}^{P}|A_t - P_t|$$

Where here P is the number of data points being evaluated, A_t is the actual value of the data point t amd P_t is the model's prediction of data point t. We show the accuracy of the models in Table 3.4

Model	Out-of-sample MAE
Linear Regression	2.443
Ridge Regression	1.820
Multilayer Perceptron Neural Network	1.126
XGBoost	0.983
Random Forest	0.789

Table 3.4: Model out-of-sample MAE for distance driven (km)

The tree ensembles perform the strongest of the set, with the Random Forest having the highest out-of-sample accuracy. We assume that if the vehicle's forecasted driving is within an epsilon of zero, then the car is parked, and more specifically parked in a location where it could charge or discharge back to a residential building or the grid. This assumption is based on discussions with industry experts from our collaborators who expect that by the time the vehicle-to-grid market emerges, charging locations will be widely abundant. Furthermore, industry experts are interested in sizing the potential of the market when, even if the car is parked in a public lot, there is the option of discharging to the local buildings (such as a driver's office space, mall, grocery store, etc.).

We optimize the charging and discharging of the EV in the testset using the random forest forecast of the vehicle's driving. In addition to the tree constraints required by the structure of the Random Forest, there are several business constraints that must be accounted for to create a reasonable charging/discharging policy. These constraints are very similar to those in Formulation (2.5), however they are adapted to account for the nature of the Random Forest prediction.

Due to the deterministic, point-prediction forecast of the model, we no longer need a dynamic programming formulation. Instead we consider all timesteps of the testset at once. Let G be the number of time-steps in the horizon, which we are optimizing over. Let variable o_g represent if the car departs from the grid at time g, how far from full the car is. Given this, we can create the objective function to be the sum across all time steps, of the monetary and carbon benefit of charging/discharging, the penalty for departing less than full, the linear benefit for using the vehicle and finally a penalty on the final time step to prevent end of horizon sell off. All these pieces come directly from the logic discussed in Chapter 2. The objective can be represented as follows:

$$\max_{x,q,o} \quad \sum_{g=1}^{G} \Bigg(\beta \sum_{t=1}^{T} \sum_{j \in L^t} \underbrace{w_t q^t_{g,p_j,j} S^t_j}_{\text{predicted distance driven}} + \underbrace{\sum_{a \in A} x_{a,g} \left(p^a_g + \lambda_R r_g \right)}_{\text{monetary and carbon benefit}} \Bigg) \quad !$$

$$- \underbrace{\lambda_{\max} o_g}_{\text{penalty for departing less than full}} \Bigg) - \underbrace{\lambda_{\max} x_{\text{charge level},G}}_{\text{final timestep penalty}}$$

Where $a \in \{\text{charge, V2G, V2H}\}$ represents the different decisions we can make, $x_{g,a}$ represents the decision to charge/discharge at time g, p^a_g is the price at which we charge/discharge and r_g is the carbon impact of this decision.

This objective is subject to several constraints. Let ψ_g represent if the car is attached to the grid at time g. We first need to ensure that if the prediction from the model is greater than zero then ψ_g is 0 and if the prediction from the model is 0 then, ψ_g is 1:

$$\sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j \leq C_{\max} \left(1 - \psi_g\right)$$

$$C_{\max} \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j \geq 1 - \psi_g$$

If the car is not attached to the grid (ie $\psi_g = 0$), then the car cannot charge or discharge. This means for all $a \in \{\text{charge, V2G, V2H}\}$:

$$x_{a,g} \leq C_{\max} \psi_g$$

In order to penalize when the vehicle leaves the grid, we add variable ϕ_g. $\phi_g = 1$ if the car was attached at time $g-1$ and is not attached at time g. By the nature of the penalties in the objective, the optimization will try and force ϕ_g to be zero whenever possible. Therefore

we only need to guarantee that if $\psi_{g-1} = 1$ and $\psi_g = 0$, then $\phi_g = 1$. We enforce this through constraint:

$$\phi_g \geq \psi_{g-1} - \psi_g$$

We then add constraints so that $o_g = 0$ unless the car departs at time g, in which case it equals $C_{\max} - x_{\text{charge},g}$. Similar to ϕ, the optimization will try to make o as small as possible:

$$\begin{aligned}
o_g &\leq C_{\max} - x_{\text{charge},g} + C_{\max} * (1 - \phi_g) \\
o_g &\geq C_{\max} - x_{\text{charge},g} - C_{\max} * (1 - \phi_g) \\
o_g &\geq 0
\end{aligned}$$

We introduce variable $x_{\text{charge diff},g}$ to represent the amount the car's battery changes at time g. If the car is attached to the grid, then this amount is how much the car charges/discharges:

$$\begin{aligned}
x_{\text{charge diff},g} &\leq \sum_{a \in A} x_{a,g} + C_{\max} (1 - \psi_g) \\
x_{\text{charge diff},g} &\geq \sum_{a \in A} x_{a,g} - C_{\max} (1 - \psi_g)
\end{aligned}$$

If the car is away from the grid, then this is the amount the car discharges from driving. We make this value be the minimum of the amount the model predicts and the amount of the charge in the battery. It is worth noting that the latter case is rare, but included to hedge against noisy predictions:

$$\begin{aligned}
x_{\text{charge diff},g} &\leq \sum_{a \in A} \hat{S}_g + C_{\max} \psi_g \\
x_{\text{charge diff},g} &\geq \sum_{a \in A} \hat{S}_g - C_{\max} \psi_g \\
x_{\text{charge level},g} - \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j &\leq C_{\max} (1 - \eta_g) \\
-x_{\text{charge level},g} + \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j &\leq C_{\max} \eta_g \\
\hat{S}_g &\leq x_{\text{charge level},g} + C_{\max} (1 - \eta_g) \\
\hat{S}_g &\geq x_{\text{charge level},g} - C_{\max} (1 - \eta_g) \\
\hat{S}_g &\leq \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j + C_{\max} \eta_g \\
\hat{S}_g &\geq \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{g,p_j,j} S^t_j - C_{\max} \eta_g
\end{aligned}$$

We ensure that charge level in the next time step, is the charge level in this time step minus however much has been discharged:

$$x_{\text{charge level},g+1} = x_{\text{charge level},g} - x_{\text{charge diff},g}$$

Finally we ensure all the other properties of a reasonable policy discussed in Section 2.3. Specifically we have that we cannot discharge more energy than what is in the car's battery, we cannot charge and discharge at the same time, we cannot discharge below some minimum charge level $C_{\min}$, we cannot charge above the battery's max level $C_{\max}$ and that there is a maximum charging/discharging rate $X^a_{\max}$:

$$\begin{aligned}
x_{\text{V2H},g} + x_{\text{V2G},g} &\leq x_{\text{charge level},g}, \\
x_{\text{charge},g} &\geq -X^{\text{charge}}_{\max}\left(1 - r_g\right), \\
x_{\text{V2H},g} + x_{\text{V2G},g} &\leq X_{\text{V2Xmax}} r_g, \\
C_{\min} r_g &\leq x_{\text{charge level},g} - \sum_{a \in A} x_{a,g} \\
x_{\text{charge level},g} - \sum_{a \in A} x_{a,g} &\leq C_{\max}, \\
x_{\text{V2H},g}, x_{\text{V2G},g} &\geq 0, \\
x_{\text{charge},g} &\leq 0, \\
r_g &\in \{0, 1\}.
\end{aligned}$$

These constraints can be added to Formulation (B.2) to create the exact MIO, which we then approximate using UMOTEM. Unfortunately, Formulation (B.2) does not scale to the size and business constraints of this scenario. In addition to the significant business constraints which are necessary to make a reasonable policy, the optimization is also run overall time steps in the horizon. This means that to run this optimization for every hour in just a day, it would require 24 forests to be embedded into the optimization. This means that we do not have the exact MIO to compare to the optimality of solution found to. Instead we compare to two other policies. First we compare to what is happening in the original policy. This would be if the discharging market doesn't exist, and cars only charged up when the car attached to the grid. We also compare against a policy proposed by [14]. This policy does not model what drivers are going to do, but instead assumes that all cars depart from the grid at 8AM, return to the grid at 5PM and use 34.1% of its battery when the car is away. This is a reasonable benchmark because it shows us what the market size estimates would be with conservative estimates of driver behavior rather than granular forecasting.

In Table 3.5 we compare the three policies using two metrics. We first show what the expected dollar savings would be when averaged across all 1,852 drivers in the 12 day test set. We then compare the policies regarding to the CO2 consumed/saved from using each policy.

We see that under the original policy drivers would have spent \$23.9 charging their car over the 12 day test set. While [14] gives improvement over this baseline, using UMOTEM savings can be as high as \$17.2, a \$41.1 improvement over baseline. In terms of CO2 benefit, we see that under the original policy, drivers would have consumed 0.15 tons of CO2, but

Policy	Monetary Benefit ($)	Carbon Benefit (tons of CO2)
Original	-23.90	-0.15
Policy from [14]	-0.80	-0.02
UMOTEM	17.20	0.11

Table 3.5: Model out-of-sample MAE for distance driven (km)

with UMOTEM the market is estimated to save 0.11 tons of CO2. This is 0.26 tons of CO2 saved over baseline in just two weeks, which translates to 667 miles driven by a gas powered car, 3.3% of a home's annual energy consumption or 4.3 trees grown for 10 years.

3.5.3 Retail Case Study

While in Section 3.5.1, we applied UMOTEM on general machine learning datasets, in this section we discuss an application in fashion promotions. We work with Oracle Retail to test our optimization process to improve a retailer's existing promotion schedule. We start by describing the underlying fashion data and the predictive Random Forest, before then applying UMOTEM to optimize promotions.

We worked directly with Oracle Retail to optimize promotions for a large American fashion retailer which caters to young men and women. The retailer has 550+ stores across the United States and Puerto Rico, with total revenue over $1.5 billion a year. The transaction data we use was collected over 106 weeks from four stores in Utah for sales around women's tops. Each transaction is associated with a store location, item ID, date of purchase, regular price of the item, and the price that was paid. The item identification also has a product hierarchy associated with it that provides style and color. To prevent biases from sparsity, we aggregate data to the store level across customers, week level across days and product class across different styles and color. In addition we supplement this data with demographic and market data. Specifically for each store we add census data on the age, race and gender of the population in the area. For each week, we add the SP Retail Select Industry Index, which represents companies in the retail segment of industry and the stock price of the fashion retailer itself. This was intended to represent market trends regarding general retail spending and attitudes towards the fashion company. To this data we also engineered several lag and seasonal trend variables to represent how price around the item had changed over time.

For this dataset we use the first 85 weeks as training, on which we build a random forest model to predict sales. The random forest has 200 trees, with a max depth of 20 and minimum leaf size of 10 data points. The random forest was built using the package `DecisionTree` in Julia. We evaluate the random forest model using WMAPE, a weighted measure of prediction error. The weight of each prediction error depends on the volume of demand, and the lower the score the better. WMAPE is calculated as follows: $WMAPE = \frac{1}{\sum_{k=1}^{K} A_k} \sum_{k=1}^{K} A_k \left| \frac{A_k - P_k}{A_k} \right|$, where for a dataset of K points, vector A represents the actual demand and P the predicted demand. Predictions are aggregated up to product class to calculate WMAPE. We compare the random forest both to the baseline and customer trends model presented in [58]. The baseline model had an out-of-sample WMAPE of 0.4322 and

the customer trends model had a WMAPE of 0.3636. By comparison, the random forest for the same data had a WMAPE of 0.3465, a 19.8% improvement over baseline and a 4.7% improvement over the customer trends model.

We then optimize the promotion schedule, for a single item and store over the weeks of the promotion horizon, using UMOTEM and the trained random forest to maximize revenue over the time horizon, and compare the new policy to the fashion retailer's current policy. We are also subject to a few business constraints, which require the optimization formulation to have the flexibility to include non-parallel (or hyperplane) constraints. Therefore we base the formulation of UMOTEM for this section on Formulation (B.2). The promotions offered must be part of a set of permissible discounts, referred to as the discount ladder, $\mathcal{R}$. There is also a limit on the maximum number of discounts that can be offered over the horizon.

In order to formulate this, let there be G time-steps in the schedule's horizon. Here the promotion horizon is the out-of-sample test set of 21 weeks. The objective, which is to maximize revenue generated by an item, can be represented as follows:

$$\max_{x,q} \quad \sum_{g=1}^{G} \sum_{t=1}^{T} \sum_{j \in L^t} \underbrace{w_t q^t_{g,p_j,j} S^t_j}_{\text{predicted sales}} \underbrace{x_{g,\text{Regular Price}} x_{g,\text{Discount}}}_{\text{price paid after discount}}$$

where x_g is the feature vector of the item at time g and the function S is the random forest's demand prediction for leaf j. The binary variable q is now also indexed by time step g to represent whether feature vector x_g ends up in leaf j. While the item's regular price is not a decision variable and can be replaced by a constant, the item's discount is the main decision variable, so we linearize by introducing binary variable $\mu^t_{g,r,j}$ to represent if discount $r \in R$ is offered at time g and if x_s lands in leaf j of tree t. $\mu^t_{s,r,j} = 1$ if both of these things are true and 0 otherwise. Then the objective function becomes where p is a constant to represent the item's regular price: $\max_{x,\mu,q} \quad p \sum_{g=1}^{G} \sum_{r \in \mathcal{R}} r \sum_{t=1}^{T} w_t \sum_{j \in L^t} S^t_j \mu^t_{g,r,j}$.

All the normal constraints of the tree from Formulation (B.2) are also enforced, but we have the following additional constraints. First the discount of the item must be from the discount ladder. We introduce binary variable $\alpha_{g,r}$ which will be 1 if at time g, discount r is offered, $x_{g,\text{Discount}} = \sum_{r \in \mathcal{R}} r \alpha_{g,r}$ and at each time step, one discount can be offered $\sum_{r \in \mathcal{R}} \alpha_{g,r} = 1$.

Next, if q indicates that x_g does not fall into leaf j, then so must μ: $\sum_{r \in \mathcal{R}} \mu^t_{g,r,j} \leq q^t_{g,p_j,j}$.

If α indicates that x_g does not have discount r, then so must μ: $\mu^t_{g,r,j} \leq \alpha_{g,r}$.

For each time step g, x_g falls into one of the leaves of tree t and have a discount from the discount ladder (one of the options being no discount): $\sum_{r \in \mathcal{R}} \sum_{j \in L^t} \mu^t_{g,r,j} = 1$. Finally to restrict that there are at most $R_{\max}$ discounts offered over the time horizon, we have: $\sum_{g=1}^{G} \sum_{r \in \mathcal{R}} \alpha_{g,r} \leq R_{\max}$.

These constraints and the objective can be added in to Formulation (B.2) to form the exact MIO that we are optimizing or into UMOTEM for the approximation. Similar to the previous section, the exact MIO does not scale to this size, creating the need for approximations like UMOTEM. This is because rather than optimizing over just one ensemble tree model, the formulation requires optimizing over G, here 21, ensemble tree models at the same time so that cross time period effects can be taken into account (for example effects of discount changes on lag variables or constraints on the maximum promotions offered over

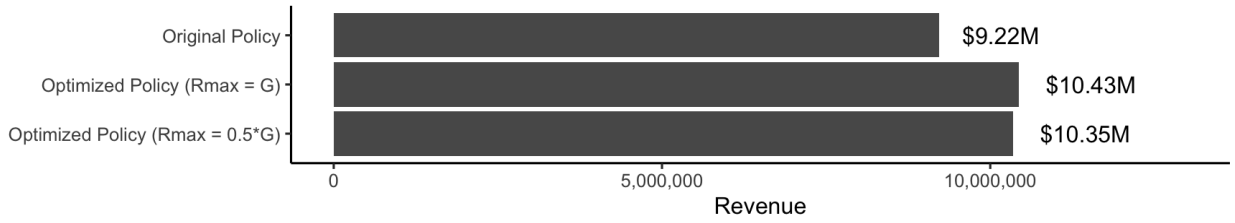
the time horizon). In an environment where optimizing over a single ensemble tree model is a challenge, optimizing over multiple at the same time becomes a significant issue. We only have UMOTEM's optimal promotion schedule to compare with the retailer's promotion strategy. To see the impact of UMOTEM's optimized promotion policy, we compare three different policies: the retailer's original policy, the optimized promotions from UMOTEM where there isn't a restriction on the maximum number of promotions (the maximum number of promotions equals the time horizon) and the optimized promotions from UMOTEM where promotions can only be offered half of the time (the maximum number of promotions equals half of the time horizon). For each of these policies, we evaluate the prediction of the random forest for the expected revenue. The results are shown in Figure 3.9 and can be summarised as follows:

- The predicted revenue from the random forest for the retailer's policy is 9.22M. This is often done through large discounts that drive up sales but can undercut revenue generated. Under the retailers original policy, the average customer only paid 65% of regular price on their fashion items. All other policies will be compared to this one.

- The predicted revenue under the optimized policy when there isn't a restriction on the number of promotions is 10.43M. This exhibits a 13.2% improvement over the retailer's policy. This policy consists of less discounts offered, so as to not undermine the overall revenue. We see that using UMOTEM, even though it is an approximation of the exact MIO, still shows improvement in terms of revenue performance. In this setting, where the exact MIO is not tractable on the data and scale of the problem, UMOTEM can direct the retailer to more optimal policies.

- The predicted revenue under the optimized policy when the maximum number of promotions for an item is restricted to half the time horizon is 10.35M. This exhibits a 12.2% improvement over the baseline. We see there isn't a strong impact on the optimal revenue from a stronger restriction on the number of promotions. This is because the policy found by UMOTEM both in this scenario and in the one where there is no maximum restriction, both use promotions sparingly for maximum impact. This results in a policy where strong sales are still generated but the discounts do not undercut the revenue generated.

Figure 3.9: Revenues for different promotion policies



3.6 Conclusions

Through this work we have proposed a new approximation for optimizing along tree ensemble models. While vast amounts of data availability have allowed for dramatic improvement and sophistication in forecasting models, the exact way they are incorporated into optimization methods is an area of interest. In the case of tree ensemble models, there have been several formulations which maximize an objective function dependent on their output. However most of these proposed works face tractability concerns, because they include exponentially growing binary variables. There are also proposed approximations of these formulations but very few address the exponential growth in binary variables, or if they do, they still fail to provide theoretical guarantees on the quality of the approximation. By comparison, in this work we have done both. We introduce UMOTEM, an Upper Bounding Method for Optimizing over Tree Ensemble Methods. We demonstrate that UMOTEM has linearly-growing binary variables. We also show analytically the expected optimality gap and describe under what scenarios the approximation method is expected to do well, namely when the in-sample error is small or the separation between payoffs is large. Finally, we show UMOTEM performs well in practice on several publicly available machine learning datasets, an EV case study and a retail case study.

Chapter 4

COVID-19: A Multiwave SIR Based Model for Learning Waves

4.1 Introduction

COVID-19 is an infectious disease caused by acute respiratory syndrome (SARS-CoV-2) [59]. Although the disease was first identified in 2019 in Wuhan, Central China, since then, it has quickly spread globally, resulting in an international health crisis, also often referred to as the 2019–20 coronavirus pandemic ([60], [61]). A major contributor to the rapid spread of the disease in 2020 was the combination of limited testing resources, an un-quantified proportion of asymptomatic cases ([62]), and the lack of a vaccine or specific antiviral treatment. On January 21, 2020 the United States identified its first confirmed case of COVID-19 ([63]) and since then, the disease has spread rapidly through the country resulting in major economic and social change. This has included but is not limited to, nearly 442,000 dead in just the United States as of February 1, 2021, massive unemployment, reaching as high as 14.7% in April 2020 ([64]), a decrease in GDP of 31.7% for the second quarter of 2020 ([65]), and massive societal change as nearly 42% of the United States' labor shifted to work from home ([66]).

A significant driver of this economic and social change has been the lack of widespread pharmaceutical interventions that would slow the spread of the disease. Even as vaccines are being administered across the country, most states have prioritized healthcare professionals, high risk civilians and essential workers. As such, government, companies and leaders have had to turn to other prevention measures to slow the disease among the larger population. Methods for intervention have included social distancing, work from home mandates, limits on the size of public gatherings and mask requirements. These interventions proved effective at first, curbing the first wave of COVID-19 cases in April 2020. However when many of these measures were rolled back and people grew tired of quarantine practices in late spring and early summer of 2020, cases surged in a second, significantly larger wave. This pattern was repeated again in the fall and winter of 2020, as the return of students to school combined with the many end of year holidays created a massive spike in cases at the end of the year. This spike continued to drive higher with each successive winter holiday, Thanksgiving, Christmas and New Years, until January 8th, when the wave reached its height ([67]). This

has been particularly challenging because the traditional epidemiology SIR model, upon which most of the modelling efforts have been based, only captures single waves.

For COVID-19 and for many other epidemics, modelling a single wave will not be able to capture the nature of the disease. The problem is further complicated by the many different possible drivers of waves (e.g. behavioral changes, medical practices, disease evolution). In practice it will often be impractical to try and capture all the different drivers that could affect changes on the epidemic through data. For example, if it seems like a new wave is starting, is this wave driven by behavior changes, such as people becoming more social? Have people's willingness to comply with mask mandates or social distancing changed? Is it due to changes in temperature or environment that's making the disease more infectious? Is it due to a new variant of the disease emerging? Some of these features can be incorporated in the model (e.g. mobility, temperature) but others are hard to quantify on a timely basis (e.g. new variants, compliance with restrictions). Furthermore, to incorporate all features that could possibly affect the disease will create a model that is first, difficult to learn parameters for, and second, hard to generalize as the model will only be applicable to the epidemic it is built for.

In this chapter, originally published in 2021, we propose a highly generalizable extension on the Susceptible-Infected-Recovered or SIR-based model (SIR and its many additional compartment extensions, such as the SIRD, SEIR and SIRS) which allows the traditional epidemiology models to accurately identify and capture multiple waves. The model's ability to detect new waves isn't dependent on expert advice to know key drivers of infections or data availability of these drivers. We show that in cases where changes in the nature of the epidemic have resulted in multiple waves, our model is able to appropriately identify and characterize these multiple waves. We then describe an algorithm that is able to learn the multiwave SIR model and prove that the algorithm will identify a change in infection rate and characterize how quickly the identification will occur. Finally, we show experimentally how the multiwave model allows for higher predictive accuracy and conclude with how this work can improve operations and supply chain management during a pandemic.

4.2 Contributions

Our main contribution through this work is introducing a multiwave SIR model and establishing guarantees on the rate of learning. Furthermore, we demonstrate significant improvements in forecasting of the COVID-19 pandemic for the different states in the US. We summarise our contributions as follows:

- **We introduce the multiwave SIR-based Models:** We develop a SIR-based model that is able to account for multiple waves regardless of the waves' drivers. The model is driven by the reality of multiple waves in cases experienced by many states across the United States and the world and investigates an area of SIR epidemiological modelling that has been largely under-explored.
- **We propose an algorithm to learn the model from data:** We introduce a dynamic method for learning the waves of an epidemic that requires no priors on the number of waves in the data or the infection/recovery rate of each wave, that would

have been hard to know in advance. The algorithm combines traditional probabilistic methods for learning SIR models with a martingale framework for detecting change points in data streams. The proposed algorithm is fast, scalable and works with live data streams.

- **We give theoretical guarantees on how fast we can detect new waves:** We leverage the martingale framework to create bounds on how fast we expect to detect a new wave of an epidemic. The bound is dependent on the parameters of the model, the population size and difference in infection rates between subsequent waves. Specifically for the parameters we use when modelling COVID-19, we show that we expect to flag a new wave within half of a week.

- **We show strong computational results on COVID-19 data:** We show how the multiwave model is able to create significant improvement in terms of forecasted infection accuracy. We describe how this multiwave SIR model can be adapted into a multiwave SEIRD (Susceptible-Exposed-Infected-Recovered-Dead) model and outperform the original SEIRD in in-sample fit and out-of-sample accuracy. We demonstrate that this performance is comparable with top COVID-19 forecasting models used by the CDC. The SEIRD model is used because of its wider popularity. Furthermore, it is more expressive because of larger number of compartments in the model.

4.3 Literature Review

Since the onset of COVID-19 pandemic, there has been a renewed interest in the modeling and analysis of epidemiological models in the operations community. Since in the current chapter we consider both predictive and prescriptive problems based on epidemiological models, we briefly discuss relevant studies from both aspects in what follows.

The SIR epidemic model forms the basis the multiwave modeling used in the current chapter. It is a compartmental epidemiological model that was first used by [68]. Compartmental models consider mathematical modeling of infectious diseases in which the overall population is divided into different compartments, for example, Susceptible, Infectious and Recovered (SIR). Differential equations are then used to model the transition of the population from one compartment to another. Since its introduction, the SIR compartmental model has been extended and analyzed in various forms. One such extension known as the SEIRD model considers further compartmentalization of the population into two extra compartments of the exposed (E) and the deceased (D) population. Various other extensions have been considered and we refer the interested readers to [69] for a detailed discussion.

Compartmental epidemiological models crucially rely on important model parameters such as infection rate, transmission rate and others. Very few studies focus on the problem of estimating these parameters from data. [70] consider the problem of identification and estimation of the SIR model parameters and develop algorithms to estimate model parameters using a least-squares methodology. Similarly, [71] proposes a robust parameter estimation technique that uses goodness of fit on out-of-sample data to estimate the SIR model parameters. Others like [72] use media coverage to augment the SIR model and then show how to estimate different model parameters. More recently [73] analyze the limits of SIR model

parameter estimation and in-turn prove that no unbiased estimator can provide accurate estimates of the model parameter until observing enough of the pandemic. The current work is complimentary to these studies since it focuses on understanding how to model and detect multiple waves of a pandemic.

The chapter is also related to change point detection in the context of epidemiological models. Change point detection is a widely studied topic in varied fields including statistics, finance and most recently in operations and demand forecasting. For a brief overview of the various applications and techniques used for change point detection, we refer the interested readers to the survey paper by [74]. Our approach most closely resembles that of probabilistic methods where the central assumption is that the sequence of observations may be divided into non-overlapping state partitions, and the data from each partition is sampled i.i.d from an underlying distribution (adams2007bayesian). [75] consider a general time series process with a change point that needs to be detected and recover mini-max and Bayesian formulations and guarantees for how fast the wave can be detected. While similar in spirit, the authors' model differs from the model of the current chapter in multiple ways. First, they consider a Bayesian framework where the change point itself has a prior distribution. Second, and more importantly, the authors assume that the observations are *i.i.d.* and the distribution parameters shifts after the change point. Contrasting this to the current chapter, the observations in our case are cases (deaths) which are *not i.i.d.* because of the underlying compartmental model that governs the dynamics of the system. As such the analysis depends on carefully modeling a martingale process from the observed cases (deaths) that is then analyzed using large deviations theory. Since the current chapter focuses on an epidemiological application, we discuss the relevant papers in this application domain in more detail. [76] consider the problem of estimating not just the start, but also the end of a pandemic using change point detection methods. Since they still focus on a single wave, our work differs considerably from theirs. More recently, [77] focus on detecting multiple waves of a pandemic using a Bayesian Poisson segmented regression model. While our work and theirs both focus on being able to flag change points in the pandemic and then training SIR parameters that only vary at the change point, the method of detection differs greatly. Specifically, [77] detect waves all at once, resulting in a process that requires knowledge of subsequent waves in order to characterize and model a specific wave. Our method models waves in sequential fashion that allows for wave detection in real-time. Similarly, [78] show how to estimate change points in a pandemic in the presence of noisy observations. The chapter uses a log-likelihood method along with modeling of a background parameter to estimate change points in the epidemic. We set ourselves apart from these works, first by proposing a real-time method of detecting when a wave is occurring by tracking when the predictive power of the SIR model suddenly deteriorates and proving theoretically we will be able to detect a new wave within approximately half of a week.

There have been a considerable effort from the research community to analyze the disease evolution of the COVID-19 pandemic. See for example, [79]–[85], amongst various others. These studies aim at developing both analytical as well as data-driven tools to understand the pandemic. [86] provides a brief overview of machine learning based studies for prediction. Nevertheless, none of these studies focus on developing analytical tools for analyzing multiple waves of the pandemic, the focus of the current work. Recent works have also focused on modeling multiple waves of the pandemic using compartmental models. [87] consider a

multiwave SIR model that extends the SIR model proposed by the same authors. The model captures the effect of lock-downs by incorporating a time changing parameter in the underlying model that causes the number of infections to go down once a lock-down is imposed. More generally, [88] consider a Bayesian Hierarchical Model to infer the effect of different interventions on the number of infections over time. Similarly [89] also focus on known changes in model parameters related to known interventions that drive changes in the pandemic case counts. Finally, [90] use wavelet theory and machine learning to directly fit wavelet models to estimate cases. All the aforementioned studies assume that the time of the change in parameters is *known* and is not learnt from the data. We focus on the case when the parameters of the underlying model shift, but not necessarily due to known interventions. Hence, our estimation problem becomes considerably harder. Furthermore, we also provide analytical guarantees on the learning rate of the change point, providing analytical justification to the modeling framework.

Researchers have also focused on empirically quantfying the effect of different interventions on the spread of the pandemic. For example, [91] quantify the impact of mobility restriction in Wuhan on the containment and delay in the spread of the virus, [92] focus on the effect of the shelter in place order in California, and [93] discuss the role of diffusion on an individual's ability to self isolate and find that income and internet access are correlated with an individuals ability to self isolate. Similarly, [94] provide evidence on the drivers of the early US pandemic using a compartmental model and find that the main drivers are population density and static characteristics of the state. Others, such as [88] consider a renewal process to model the number of infected individuals and consider six different levels of intervention which directly impact the reproduction number in the SIR models. While the above cited studies focus on measuring the effect of specific interventions on the spread of the pandemic, the current work focuses on the estimation of general shifts that are agnostic to what could be driving that change, regardless of whether it is changes in government intervention, mobility or new variant.

Finally, there has also been a considerable upsurge in operations management related prescriptive studies related to the COVID-19 pandemic. For example, [95] focus on optimal allocation of vaccines with limited supply. In contrast, [96] focuses on developing optimal targeted lockdown policies to control the spread of disease in a population. [97] creates a decision support tool for both governments and individuals to aid in various decision making problems during the pandemic. Finally, [98] study the impact of COVID-19 on supply chain credit risk. Similarly, [99] study the impact of the pandemic of gig economy and find substantial differential impact of the pandemic across different groups. Furthermore, [100] focus on finding the impact of government interventions on the spread of the pandemic. The current chapter instead focuses on the problem of optimally allocating testing capacity over different time periods to maximize detection of the virus amongst different population groups: symptomatic, asymptomatic and contact-tracing.

4.4 A Multiwave SIR-based Model

4.4.1 Intuition and Model Preliminaries:

We first describe the original SIR model upon which we are building the proposed multiwave model. We then describe the 2-wave SIR model and finally, discuss the multiwave SIR model. Compartmental epidemiology models, such as the original SIR model, are based on the idea that a given population experiencing an epidemic can be grouped into sub-groups or compartments that can be then used to characterize the spread of the disease. The original SIR model [68] has three such compartments: susceptible, infected and recovered. People move from being susceptible (S) to the epidemic, to becoming infected (I) and finally recovering (R), hence the name SIR. The order of the letters in the name often indicate the flow of the population through the compartments. Because the entire population is split into these compartments the size of the total population can be found by summing the size of each individual compartment. The SIR is considered the simplest of the compartmental epidemiology models, but most of the extensions are based on this foundation. A few examples of popular extensions include accounting for populations that deceased rather than recovered (SIRD), instances where people lose immunity and move from recovered to susceptible again (SIRS) and a compartment for people exposed to the disease but not yet infectious (SEIR). Movement between compartments in the SIR model is determined by the size of each compartment and two parameters, the infection rate β and recovery rate γ. The SIR model is governed by 3 equations, where at time t, S_t is the size of the susceptible population, I_t is the size of the infected population, R_t is the size of the recovered population and N is the total population ($N = S_t + I_t + R_t$):

$$S_{t+1} = S_t - \frac{\beta S_t I_t}{N}, \quad I_{t+1} = I_t + \frac{\beta S_t I_t}{N} - \gamma I_t, \quad R_{t+1} = \gamma I_t \tag{4.1}$$

By the nature of these equations, the original SIR model assumes that once the infected population has started to decline, it will not increase again. N, β and γ are constants and because the differential of the susceptible population is non-positive the susceptible population will never get bigger. Therefore, if the infected population has started to decline, indicating that $\frac{\beta SI}{N} - \gamma I < 0$ then as the size of the susceptible population continues to fall, $\frac{\beta S}{N}$ will only continue to get smaller ensuring that the differential of the infected population remains negative. This suggests that the traditional SIR model will always predict that once a single wave has occurred, the disease will die out. Unfortunately, a single wave has not been the case in most regions around the United States and in many other countries.

In the 2-wave SIR model we start with the original SIR model. However, in this new model, there is a time T_1, when wave 1 ends and wave 2 starts, where the infection and recovery rates change from β_1 and γ_1 to β_2 and γ_2. As we discuss in Section 3, our algorithm learns when this time T_1 occurs. The number of individuals in each compartment remains the same, but at time $T_1 + 1$ the system is characterized by new parameters, β_2 and γ_2. Thus

the 2-wave model can be characterized by the following 3 equations:

$$
\begin{aligned}
S_{t+1} &= \begin{cases} S_t - \frac{\beta_1 S_t I_t}{N} & t \leq T_1 \\ S_t - \frac{\beta_2 S_t I_t}{N} & T_1 < t \end{cases} \\
I_{t+1} &= \begin{cases} I_t + \frac{\beta_1 S_t I_t}{N} - \gamma_1 I_t & t \leq T_1 \\ I_t + \frac{\beta_2 S_t I_t}{N} - \gamma_2 I_t & T_1 < t \end{cases} \\
R_{t+1} &= \begin{cases} R_t + \gamma_1 I_t & t \leq T_1 \\ R_t + \gamma_2 I_t & T_1 < t \end{cases}
\end{aligned} \tag{4.2}
$$

Intuitively, T_1 can be thought of as the time of the onset of the second wave. At time T_1, some behavior about the population has changed resulting in a different rate of infection or recovery and we re-calibrate the model accordingly. For example, in states where government restrictions were lifted resulting in a second wave, β_1 would represent the infection rate when government measures were in place, T_1 would correspond with the lifting of restriction and β_2 would be the unconstrained rate of disease spread.

While we showed previously that the original SIR model is unable to model multiple waves, this model will be flexible enough to capture two waves. If time T_1 falls after the first peak, then even when $\frac{\beta_1 SI}{N} - \gamma_1 I < 0$ as it would be after the first peak, as long as neither the susceptible nor infected populations are zero, either β_2 and γ_2 can make the differential positive again. In this case the model will predict two waves for the infection.

When we consider a single change point T_1 then the model will consider, at most, two waves. However we can extend the model to multiple waves by considering multiple change points. If we wish to model $\bar{n}$ waves, we can consider change points, $T_1, T_2, ..., T_{\bar{n}-1}$, and corresponding parameters $\beta_1, \gamma_1, \beta_2, \gamma_2, ..., \beta_{\bar{n}}, \gamma_{\bar{n}}$. We will discuss in Section 4.4.3 how to determine the number of waves dynamically as the disease evolves.

There are two parts to learning the parameters of the multiwave SIR model. One, we need to detect the change points, $T_1, T_2, ...$ in order to segment the epidemic into waves. Two, for each wave, we need to learn the parameters for the SIR model. We propose a dynamic approach to learning both the change points and the infection/recovery rates. We assume in our training data we have for daily information on, $i(t)$, the number of new cases, and $r(t)$, the number of recovered cases on each day t. Given this data, we calculate the starting condition of the wave, either from the initial conditions of the epidemic or the end conditions of the previously modeled wave. Using the initial conditions and the data up to day t, we learn the "best" parameters for the wave happening. Then, based on these parameters, we evaluate whether a change point has occurred, indicating a new wave has started. If a new wave has not happened then we move on to the next day, updating our assessment of the parameters. If a new wave has started we record the population compartments on day t as the initial conditions of the new wave.

4.4.2 SIR Parameters

For a given wave n, the parameters are only dependent on the initial conditions of the wave (number of susceptible, infected and recovered at the start of the wave) and data recorded

during the wave. We don't need to know any information about the subsequent waves to learn these parameters. At the start of an epidemic, we can learn the first wave parameters directly from the data on new infection/recoveries. For each wave thereafter, training occurs sequentially. We use the learned parameters to help detect new waves and if a new wave is flagged, we use the end conditions of the wave we are in as the initial conditions of the new wave so that we can learn the new wave's parameters.

For the single wave model, it has been shown that learning the infection rate and recovery rate can occur quickly. The number of new infections/recovered can be modelled as a Poisson distribution and can be solved for directly using initial conditions and data (li2018parameter). Consider wave n that starts at time T_{n-1} and ends at time T_n and assume the $n-1$ previous waves have been modelled. Then we would have already modelled the initial conditions, $S_{T_{n-1}}$ and $I_{T_{n-1}}$ and previous literature (li2018parameter, hong2020estimation) shows that we can estimate the paramters of wave n as:

$$\hat{\beta}_n^t = \frac{\sum_{t=T_{n-1}}^{T_n} i(t)}{\sum_{t=T_{n-1}}^{T_n} \frac{S_{t-1}I_{t-1}}{N}}, \hat{\gamma}_n^t \quad = \frac{\sum_{t=T_{n-1}}^{T_n} r(t)}{\sum_{t=T_{n-1}}^{T_n} I_{t-1}} \tag{4.3}$$

where

$$S_{t-1} = S_{T_{n-1}} - \sum_{u=T_{n-1}}^{t} i(u), I_{t-1} \quad = I_{T_{n-1}} + \sum_{u=T_{n-1}}^{t} i(t) - \sum_{u=T_{n-1}}^{t} r(t) \tag{4.4}$$

Nevertheless, the estimation process does not always produce the most accurate results in practice, so the estimates for $\hat{\beta}_n^t$ and $\hat{\gamma}_n^t$ need to be further optimized. As a result, we use the above as a warm start in a maximum likelihood estimation procedure. Note that we do not need any information on future waves to estimate the parameters of the current wave.

4.4.3 Change points

Before discussing change points, it is worth first explicitly defining what a "wave" is. For the sake of our model we will define a wave as a section of time when the dynamics of the epidemic are relatively stable such that constant parameters in the SIR model are able to describe the spread of the disease. This means that the errors of a well tuned SIR model should just be noise, without a trend to them. A new wave is said to have happened when the dynamics of the underlying epidemic have changed such that the trained parameters are no longer able to describe the epidemic. The key to wave detection then becomes identifying when the errors of the model are no longer just noise but rather represent a shift in the underlying infection rate. To accomplish this, we apply a martingale framework in time-varying data streams (ho2005martingale) to the infection/recovery rate of the SIR model. In particular, the approach follows three key steps:

1. First we define an error score per time period (Z_t) based on weighted error in predictions to estimate how strange or unusual a data point is.

2. Using these error scores over a time scale, we define a p-value (p_t) that undergoes a distributional shift, whenever there is a change point in the underlying time series.

3. Finally, using the p-values, we define a martingale sequence and use a threshold test to check if the martingale has shifted away significantly. This threshold test is used to flag a new wave in the time series.

We note that while similar in spirit, the approach differs considerably from existing change point detection methods. Whereas most change point papers assume access to a stream of observations that are sampled *i.i.d* (see for example, [75], [101]–[103]), the observations in the current chapter are COVID-19 cases (deaths) which are *not i.i.d.* because of the underlying compartmental model that governs the dynamics of the system. As such the analysis depends on carefully modeling a martingale process from the observed cases (deaths) that is then analyzed using large deviations theory.

Error scores: First, we consider how the model looks when a new wave has not started. Let us assume again that waves $1 \dots n-1$ have been modeled, and we are considering wave n which begins on day T_{n-1} and ends on day T_n. On each day $t \in [T_{n-1}, T_n]$, we calculate our best estimate of wave n's infection rate, $\hat{\beta}_n^t$ and predict our estimate of the number of new infections, $\hat{i_t} = \frac{\hat{\beta}_n^t S_t I_t}{N}$. We can use the weighted error on this prediction, $Z_t = \frac{|\hat{i_t} - i_t|}{\sqrt{S_t I_t}}$, to understand how strange or unusual a new data point is. During the same wave, these errors should be identically distributed. Intuitively, this is because as long as we have a good estimate of the infection rate, the data points are all part of the same wave and after we weight to account for changing susceptible/infection levels, we have no prior on any days being more or less prone to error. We make the following technical assumption on the estimation error of the individual wave parameters, to quantify the effect of estimation error on time to detect waves.

Let $\hat{\beta}_n^t$ denote the estimate of the infection rate of the underlying wave n at time step t. Then, $\exists \delta \geq 0$ such that

$$|\hat{\beta}_n^t - \beta_n| = \frac{\delta}{\sqrt{S_t I_t}},$$

where S_t and I_t denote the susceptible and infected population at time t. The assumption above ensures that the error in the estimation of the wave parameters are bounded as a function of the compartmental model parameters, S_t and I_t. δ is the estimation error parameter that is large when the estimation of the underlying infection rate is not accurate and small otherwise. Note that we only assume the existence of such δ but not its knowledge for the proposed change point detection method to work.

We show next that the error scores constructed above follow an important property: they are *exchangeable* within a wave. That is, they are i.i.d. Note that the same does not hold for the observed sequence of cases (deaths).

The weighted sequence of errors within the same wave,

$$\left\{ Z_t = \frac{|\hat{i_t} - i_t|}{\sqrt{S_t I_t}} : T_{n-1} \leq t \leq T_n \right\}$$

$\forall n$ is exchangeable, i.e. the joint distribution $p(Z_{T_{n-1}}, ..., Z_{T_n})$ does not change under any permutation of the indexes. Proof: See Appendix C.1.

We call this sequence of errors the strangeness scores of the data points.

Exchangeability of the strangeness scores no longer holds true when a new wave has begun. Let us consider the days T_{n-1} to T_{n+1} where a new wave begins at time T_n. Then the error between T_{n-1} and T_n will be exchangeable, but this won't be true after T_n. We expect the error to go up because the infection rate has changed and we will be using the infection rate from wave n to model what's happening in wave $n+1$. Thus we can test whether a new wave has started by checking exchangeability, which consists largely of two steps (vovk2003testing). First we calculate a p-value measuring the proportion of previous instances that had errors equal to or bigger than the current data point.

Calculating p-values and defining a martingale sequence: Let p_t be defined as the following:

$$p_t = \frac{\sum_{u=T_{n-1}}^{t} 1_{Z_u \geq Z_t}}{t - T_{n-1}}. \tag{4.5}$$

As long as the underlying error remains exchangeable, then p_t is distributed uniformly in $[0, 1]$ (vovk2003testing). However when the next wave hits, we expect p_t to drop closer to 0 for a while, on account of larger estimation error.

The key then becomes determining when p_t is no longer uniformly distributed. We do this by creating a martingale M_t, using a sensitivity parameter s

$$M_t = \prod_{u=T_{n-1}}^{t} \frac{s}{1 - \exp(-s)} \exp(-sp_u) \tag{4.6}$$

where the martingale is initialized at some value, M_0. It is worth noting that $M_t = M_{t-1} \frac{s}{1-\exp(-s)} \exp(-sp_t)$. First this shows that it is not necessary to store the entire history of p_t. Second, it helps demonstrate that as long as p_t is uniformly distributed, M_t is a martingale because,

$$E[M_t | M_{t-1}] = M_{t-1} E\left[\frac{s}{1 - \exp(-s)} \exp(-sp_t)\right] = M_{t-1}\,.$$

As long as p_t is uniformly distributed, we expect M_t to remain around M_0. However, when p_t drops, M_t will rapidly grow signaling a wave has occurred. In general the only selection requirement for s is that it must be positive, but the larger s is, the more sensitive M_t is to changes in infection rate. In our experiments we have found $s = 2.1$ is a reasonable value for this parameter allowing the model to be sensitive to wave changes without overflagging.

We use a threshold test in order to determine when M_t has deviated significantly far away from M_0 (ho2005martingale, vovk2003testing). Note that to ensure that the martingale is not too sensitive with respect to initial estimates of the parameters, whenever M_t drops below 1, we reset the strangeness scores and martingale. We set a threshold, $\lambda > M_0$, and next define the concept of λ-detectable waves. [λ-Detectable Waves] Let M_t be the martingale sequence defined as before. Then, a wave is λ-detectable if there exists some time t within the length of the wave, such that $M_t > \lambda$.

Notice that if $M_t > \lambda$, then we have detected a new wave and we reset the martingale and strangeness scores, and begin learning the parameters of the new wave. The challenge is to determine values for λ that are large enough that we are not overly sensitive to white noise

but small enough so that we can quickly identify new waves, without missing anything. In the following sections we will characterize the expected time that it takes for this approach of using martingales to identify new waves based on the relationship between s, λ, M_0 and the other attributes of the epidemic.

4.5 Finite Sample Guarantees: how fast do we detect a new wave?

In this section we discuss the expected time it takes to flag a λ-detectable wave in the model. We start by first demonstrating that when a new wave hits, the martingale, M_t, becomes a submartingale and characterize the rate of growth of the submartingale. We then use the growth rate along with optional stopping theorem to prove bounds on the expected detection time. For the rest of this section, we will focus on the case when a new wave results in increasing daily new cases. This assumption is only made for analytical tractability and we note that the multiwave model works for all new waves. Furthermore, it is in this situation when the number of cases are going up that rapid wave detection is most relevant.

Let us say that wave $n+1$ has just kicked off but the model has not flagged it yet. This means that the martingale would be still considering all the data from wave n, which went from time T_{n-1} to T_n, along with all the data from the new wave.

When the $n+1$ wave hits at T_n, M_t becomes a submartingale. This is because when a new wave occurs, p_t is no longer uniform. The model will still be making predictions based on wave n, but the new infections will be driven by the parameters of wave $n+1$. This will result in higher strangeness scores, Z_t, which are the weighted difference between the predicted and actual new infections, causing p_t to skew closer to 0. When this happens, $E\left[\frac{s}{1-\exp(-s)}\exp(-sp_t)\right] > 1$, causing M_t to become a submartingale. M_t will be expected to grow until it hits λ, flagging a new wave. Theorem 4.5 introduces a lower bound on the growth rate of the submartingale which we use to estimate how long it will take M_t to hit the threshold.

Consider a λ-detectable wave that starts at T_n, and assume submartingale, $M_t < \lambda$ for $t > T_n$, so that a wave $n+1$ has occurred but a new wave has not been flagged yet. Let s be the martingale's sensitivity parameter, β_n represent the infection parameter of wave n and S_t, I_t, and N be the susceptible, infected and total populations respectively. Then the submartingale's expected growth on day t can be bounded as:

$$E[M_t|M_{t-1}] \geq$$
$$M_{t-1}\frac{s\exp\left(-s\left(\sqrt{\frac{2\pi}{S_tI_t}}\exp\left(-\frac{1}{4}\frac{(|\beta_n-\beta_{n+1}|-\frac{\delta}{\sqrt{S_tI_t}})^2}{\max(\beta_n,\beta_{n+1})}\frac{S_tI_t}{N}\right)+0.5\right)\right)}{1-\exp(-s)}\left(1+s0.5\frac{(T_n-T_{n-1})-(t-T_n)}{T_n-T_{n-1}}\right)$$

Proof: See Appendix C.1. Theorem 4.5 shows that the growth rate is largest for days right after a change point has happened. It is here, when t is small and the martingale is growing the fastest, that we are most likely to detect the new wave. The farther t gets, the smaller the growth rate of the martingale becomes.

Given that the wave is just starting, Theorem 4.5 shows us that beyond the model parameters, the speed at which M_t grows is largely dependent on (1) the proportional difference between the infection rates of wave n and $n+1$, and (2) the size of susceptible/infected populations. This makes sense intuitively. We would expect that if the infection rates between waves changes dramatically, the model should quickly identify a new wave has occurred. Furthermore, if there is a large susceptible/infected population relative to the total population, changes in infection rate will be amplified making new waves easier to flag.

To prove Theorem 4.5 we first use the martingale formulation to analyze the structure of the growth rate of the martingale in terms of other problem parameters in Proposition 4.5. Then we characterize the relationship of the strangeness scores in the new wave (Lemma 4.5) and between the new wave and old wave (Lemma 4.5) to prove Theorem 4.5.

First, we recall that if we let t be sometime after the wave $n+1$ has started and recall that $Z_u = \frac{|\hat{i_u} - i_u|}{\sqrt{S_u I_u}}$. Let u_1 be a time in wave n, then the $P(Z_{u_1} > Z_t)$ is the probability that the weighted error in the wave n is greater than the weighted error in wave $n+1$. Similarly if u_2 is a time in wave $n+1$ such that $u_2 < t$, $P(Z_{u_2} > Z_t)$ is the probability that the weighted error earlier in wave $n+1$ is greater than the weighted error later in the wave at time t.

Let t be a day in wave $n+1$ so that $t > T_n$. Let q_n represent the probability that the strangeness score, Z, on a day in wave n is higher than the strangeness score on day t and let q_{n+1} be the probability that the strangeness score on a day before t in wave $n+1$ is higher than the strangeness score on day t, we have:

$$E[M_t|M_{t-1}] \geq M_{t-1} \frac{s \exp\left(-s(q_n + q_{n+1})\right)}{1 - \exp(-s)} \left(1 + s q_{n+1} \frac{(T_n - T_{n-1}) - (t - T_n)}{T_n - T_{n-1}}\right)$$

Proof: See Appendix C.1.

Notice that to analyze the growth rate of the martingale, we now need to upper bound probabilities q_n and q_{n+1}. We focus on upper bounding these two quantities next. We start by stating Lemma 4.5 that upper bounds q_{n+1}.

Let $q_{n+1} = P(Z_{u_2} \geq Z_t)$, for $T_n < u_2 < t$ or the probability that the strangeness score on a day before t in wave $n+1$ is higher than the strangeness score on day t. Then $q_{n+1} \leq 0.5$
Proof: See Appendix C.1.

Lemma 4.5 provides a coarse upper bound on q_{n+1}. In Lemma 4.5 we provide a stronger bound on q_n by leveraging the fact that the errors will increase when the new wave hits because we will be using an infection rate from wave n to predict for wave $n+1$.

Let $q_n = P(Z_{u_1} \geq Z_t)$, for $u_1 \in [T_{n-1}, T_n]$ and $t > T_n$ or the probability that the strangeness score on a day in wave n is higher than the strangeness score on day t, which is in wave $n+1$. Then

$$q_n \leq \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4} \frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{S_t I_t}})^2}{\max(\beta_n, \beta_{n+1})} \frac{S_t I_t}{N}\right)$$

Proof: See Appendix C.1.

The proof of Lemma 4.5 follows by integrating over all possible weighted error scores in predicting cases on day t and using Chernoff bound to upper bound the probability estimates.

Notice that the upper-bound on q_n in Lemma 4.5, and by extension the lower-bound on the martinagale growth rate, grows with both the proportional difference in infection rate

between the waves and the size of the population. It is worth noting, as demonstrated by this bound, that it will be hard to identify changes in infection rate when either the susceptible population, S_t, is low, meaning everyone has either had COVID-19 or been vaccinated, or when the infected population, I_t, is low, which usually happens when an epidemic is just starting or about to end.

In Theorem 4.5, we use this growth rate from Theorem 4.5 to create an upper bound on the expected time it will take the submartingale to hit the threshold.

For any wave n, let c be a constant such that $\frac{S_t I_t}{N} \geq c > 0$ for all $T_n \leq t \leq T_{n+1}$. Then, if the length of the wave is such that,

$$T_n - T_{n-1} > \frac{\lambda - M_0 + \frac{s^2 \exp\left(-s\left(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c\right) + 0.5\right)\right)}{4(1-\exp(-s))}}{\frac{s \exp\left(-s\left(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c\right) + 0.5\right)\right)}{1-\exp(-s)} \left(1 + \frac{s}{4}\right) - 1}$$

then we can define T as the time the model successfully detects the new wave and we have that

$$E[T - T_n] \leq \frac{m_1 - \sqrt{m_1^2 - 4m_2(T_n - T_{n-1})(\lambda - M_0)}}{2m_2},$$

where

$$m_1 = \left(\frac{s \exp\left(-s\left(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c\right) + 0.5\right)\right)}{1 - \exp(-s)} \left(1 + \frac{s}{2} - \frac{s}{4(T_n - T_{n-1})}\right) - 1 \right) (T_n - T_{n-1})$$

$$m_2 = \frac{s^2 \exp\left(-s(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c\right) + 0.5)\right)}{4(1 - \exp(-s))}$$

Proof: See Appendix C.1.

Theorem 4.5 provides an upper bound on our expected time to flag a new wave. As we would expect, the upper bound is first driven by the submartingale's sensitivity parameter and the threshold at which we flag a new wave. Second, as with the growth rate, the time till wave detection is driven by the length of waves, the proportional difference in infection rates and the size of the population. As all of these get bigger, the sooner the model expects to detect a new wave.

To prove Theorem 4.5, we first create a new variable H_t based on M_t which we show to be a submartingale, and use optional stopping theorem on H_t to find the upper bound on the expected time till detection. We won't be able to solve for the expected time till detection directly, but we can use an iterative method to solve for tighter and tighter bounds on the time till detection until we get Theorem 4.5.

Let us define a new submartingale (see Lemma C.1 in Appendix C.1 for a formal proof) for $t \geq T_n$, where as before q_n represent the probability that the strangeness score, Z, on a day in wave n is higher than the strangeness score on day t and q_{n+1} is the probability that the strangeness score on a day before t in wave $n+1$ is higher than the strangeness score on

day t:

$$H_t = M_t - \sum_{u=T_n+1}^{t} \left(\frac{s \exp\left(-s(q_n + q_{n+1})\right)}{1 - \exp(-s)} \left(1 + s q_{n+1} \frac{(T_n - T_{n-1}) - (u - T_n)}{T_n - T_{n-1}} \right) - 1 \right).$$

Let T be the stopping time, the first time that $M_t \geq \lambda$, or the time when the new wave is flagged by the model and our estimate of T_n. We recall that by definition, a wave is λ-detectable if M_t hits λ within the length of a wave (where the length of the n^{th} wave is $T_{n+1} - T_n$). Therefore we have that $T - T_n \leq T_n - T_{n-1}$. From this we can say that $E[T - T_n] \leq m(T_n - T_{n-1})$ where $m \in [0, 1]$. Then the bound is finding the smallest value for m which can hold.

We do this by considering optional stopping theorem, given that H_t is a submartingale. We next check to see if T, the time we flag a new wave, and H_t satisfy the requirements of optional stopping theorem, namely that T is finite almost surely and $H_{\min(t,T)}$ is bounded (williams1991probability). We start by showing that T is finite almost surely.

For all $t \geq 0$, there exists some T_λ and $\epsilon > 0$ such that: $P(T \leq t + T_\lambda) \geq \epsilon$. Proof: See Appendix C.1.

Given Lemma 4.5, we have that $P(T < \infty) = 1$ (williams1991probability). Next we check if $H_{\min(t,T)}$ is bounded, such that there exists a b such that $|H_{\min(t,T)}| \leq b$. This is straightforward to check because $|H_{\min(t,T)}| \leq |M_{\min(t,T)}| \leq \lambda$. So we can set $b = \lambda$ and we satisfy the second requirement of optional stopping theorem. Applying optional stopping theorem allows us to prove Theorem 4.5.

For our modelling purposes, we have been using $s = 2.1, M_0 = 203, \lambda = 205$. Given that $q_{n+1} = 0.5$ and when the population is large enough, as is almost always the case when modelling state or county-level COVID-19 populations, q_n becomes negligible. As long as the length of wave (ie $T_n - T_{n-1}$) is greater than 8 days then we have that iterative process for m holds. Practically for COVID-19 modelling, this is rarely an issue. However when the model is applied to epidemics with quick, short waves, users need be certain to set the threshold λ close enough to M_0 that this is not an issue.

In Table 4.1 we characterize what the bound looks like in terms of wave length. Choosing the parametric values described in Table 4.1, we find that the expected time to detect a new wave in half a week or less.

Table 4.1: Upper bound on expected time till detection, given that $s = 2.1, M_0 = 203, \lambda = 205$ and c is significantly large

Wave Length	$E[T - T_n]$ **bound** Upper bound on the expected time to detect a new wave in days
10	4.04
15	3.40
20	3.20
25	3.10

Other epidemiological models: In this work we relax the assumption that parameters of the compartmental epidemiology model are constant, however we inherit the remaining

assumptions of the SIR model. For most of these assumptions there are known ways of relaxing them, largely by increasing the number of compartments in the model. The multiwave variation is easily extendable to these variations of compartmental modelling, for example, the computational section of the chapter uses a SEIRD model. The analytical results can also be extended easily. In particular, we argue that under the SEIRD model, the only thing that changes in the proof is the way we define the weighted error of the model. We define $Z_t = \frac{|\hat{e}_t - e_t|}{\sqrt{S_t I_t}}$ instead of $Z_t = \frac{|\hat{i}_t - i_t|}{\sqrt{S_t I_t}}$ where e_t is the number of new people in the exposed state, and $\hat{e}_t$ the estimated number of new people in the exposed state. The Recovered (R) and Dead (D) states do not affect the results at all, as what happens to people after leaving the (I) state is irrelevant in the martingale. The rest of the proof and the subsequent results remain unchanged. It is worth noting that the time till detection and ease of learning will vary with the number of compartments. Below we list the main assumptions imposed by the traditional SIR model, how they can be relaxed and accounted for in multiwave model:

- **All members of the population are equally susceptible:** Due to different demographic effects on the likelihood of contracting COVID-19, plus differences in behaviors which can reduce an individual's risk (social distancing, quarantining, mask-wearing, etc.), not everyone is equally susceptible. This is less of an issue when modelling at state level because differences in individual behavior can be averaged across the the population. This creates a relatively close approximation for high-level predictions. However if this method were to be applied to smaller populations or where differences are more stark, differences in susceptibility can be accounted for by splitting the susceptible population further into sub-groups based on their levels of susceptibility.

- **All members of the population have equal rates of recovery:** Similar to the discussion above we've seen this does not hold true across demographics but can be accounted for by including multiple infected compartments that differentiate different rates of recovery.

- **Complete immunity is inferred by recovering from the disease:** There has been evidence that this does not hold, as there have been reported cases of people contracting COVID-19 multiple times, especially as new variants of the disease crop up. The SIRS model is designed to account for this by allowing the recovered population to move back to susceptible at some rate. As long as this recovery rate can be learned or modelled, such that the size of the susceptible population is still known, this does not affect the analytical results we present in this chapter.

- **The population is closed (no births, deaths - other than that caused by the disease - or migration):** Almost all regional populations are not closed. Even in cases where a country is able to lock borders, births and deaths will change the population. When working with state populations, this is less of an issue because the scale of births/deaths/migration is very small compared to the state population. For example, births would only increase the susceptible population by approximately 1.1% and deaths would decrease the population by 0.8% in the United States, thus not really affecting the COVID-19 forecasts. However for small populations where births, deaths or migrations make a difference, rates of incoming and outgoing populations can be

accounted for. As long as the model is able to track these changes and can calculate the number of susceptible/infected people, then it will not affect the multiwave model.

- **All infections/recoveries are detectable:** The SIR model assumes that all movement between compartments (from Susceptible to Infected or from Infected to Recovered) is detectable. This means that there are no asymptomatic cases or testing gaps. Modelling with the SIR when data is incomplete is one of the major challenges of the COVID-19 pandemic. Asymptomatic cases and insufficient testing can be accounted for by adding an asymptomatic compartment and modelling assumptions around the proportion of asymptomatic cases. It would be difficult to detect waves affecting only the asymptomatic carriers, but the martingale wave detection approach can be applied to the symptomatic cases in the same manner as described in the SIR model.

4.6 Results from the Data

In this section we describe our forecasting efforts using the multiwave SEIRD model for different states in the United States. Specifically we discuss the data collected, the prediction process and finally the model's performance in comparison with the original SEIRD model. We note that the SEIRD model is used because of its wider popularity. Furthermore, it is more expressive because of larger number of compartments in the model.

We used data from the John Hopkins Coronavirus Resource Center (jhu) on daily COVID-19 cumulative cases and deaths. The JHU COVID-19 dataset is one the primary sources for COVID-19 case data in the United States but it does not provide reliable information on active or recovered populations. From the JHU cumulative data, we created an active cases feature using a 14-day rolling sum of the cumulative cases (ie active COVID-19 cases were people who tested positive within the past 14 days) and a recovered cases feature by assuming that everyone who tested positive for COVID-19 but isn't an active case or deceased, has recovered. Specifically we show two sets of results for this chapter. The first set is on models trained on state-level data between April 12, 2020 and September 1, 2020 and then compared out-of-sample results on September 2, 2020 through October 1, 2020. The second set is similar, state-level models but trained up to January 15, 2021 and evaluated out-of-sample on results between January 16, 2021 and February 15, 2021.

Given a region, either state or county, and it's active, deceased and recovered populations we trained an SEIRD model and multiwave SEIRD model. We chose to implement an SEIRD model rather than SIR for two reasons. First, because the incubation period for COVID-19 plays a significant role in it's spread, it was important to include an "exposed" population. Second, unfortunately part of COVID-19's severity is its fatality rate and COIVD-19 deaths are closely tracked. However, this meant we could use the reliability of the deceased population data to train the model more accurately. In practice, the multiwave SEIRD model is very similar to the multiwave SIR model. Waves are detected in the exact same way, however when training populations and parameters in the wave we use an SEIRD model rather than SIR.

For our results we use the original SEIRD model as a benchmark for our multiwave SEIRD. The SEIRD model is the most commonly used model for understanding and pre-

dicting new cases of COVID-19. The original SEIRD model is equivalent to a single-wave multiwave SEIRD model. This means that in states where we do not detect multiple waves, the two models will be exactly the same. However in cases where multiple waves have occurred, the SEIRD provides a comparison for if the multiwave SEIRD is able to better fit the data.

We describe the out-of-sample prediction performance of the SEIRD and multiwave SEIRD models using mean absolute percentage error (MAPE). MAPE is a measure of prediction error such that the lower the value, the more accurate the model. It can be calculated as:

$$MAPE = \frac{1}{P}\sum_{t=1}^{P}\frac{|A_t - P_t|}{A_t}$$

Where here P is the number of data points being evaluated, A_t is the actual value of the data point t amd P_t is the model's prediction of data point t. We measure the MAPEs of the cumulative infected cases predictions over the course of a month (September 2, 2020 - October 1, 2020 or January 16, 2021 - February 15, 2021, depending on the training set).

Table 4.2: MAPEs comparing SEIRD and multiwave SEIRD model performance aggregated to region

	Trained till Sept 01, 2020		Trained till Jan 15, 2021	
Region	**SEIRD**	**Multiwave SEIRD**	**SEIRD**	**Multiwave SEIRD**
Northeast	7.6%	3.9%	24.4%	11.6%
Southeast	30.9%	7.0%	13.9%	10.8%
Midwest	12.8%	6.3%	73.5%	5.8%
West	33.4%	18.1%	17.3%	5.6%
Southwest	99.1%	5.6%	17.6%	3.5%
USA	27.0%	8.4%	31.4%	7.4%

In Table 4.2 we compare the MAPEs of the SEIRD and multiwave SEIRD models, grouped by regions of the United States. Both models were trained at state-level, but predictions are aggregated to region level to show national trends. For full state-level MAPEs see Table 4.3.

These results align with the context of what was happening in the United States over the summer and winter of 2020. For the summer of 2020, much of the country had experienced a summer second-wave of COVID-19, which is why on average the multiwave SEIRD model trained up September 1, 2020 brings down the MAPEs by 18.6%. However the experience of second-waves was not the same across the country. The Southeast, Southwest and West were hit the hardest by the summer second-wave, particularly in states like California, Arizona, Texas and Florida. Therefore it makes sense that the most modelling improvement is seen in these regions. By comparison, the Northeast and the Midwest had very little change over the summer, which is why there is slight but not dramatic improvement in these regions.

The trend over the winter of 2020 is slightly different. Almost all states across the United States experienced a wave. Between the return to school, Thanksgiving, Christmas and New Years, there was a dramatic wave that hit in the last quarter of 2020, in most states

far exceeding the initial wave. We see therefore that the multiwave SEIRD demonstrates approximately similar improvement across regions.

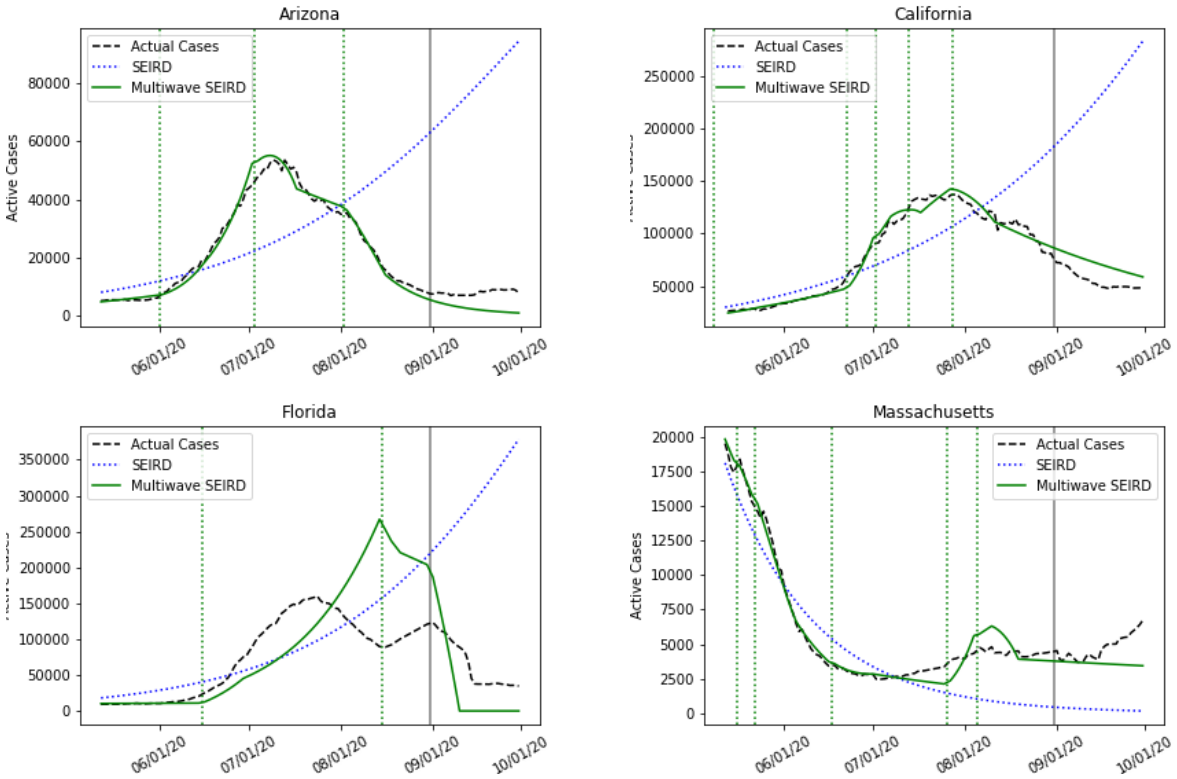


Figure 4.1: SEIRD and multiwave SEIRD in-sample and out-of-sample prediction in Florida, California, Texas and Massachusetts when trained up to Sept 01, 2020

We further illustrate the results of the multiwave SEIRD model by examining the in-sample fit and out-of-sample error in specific states which we know to have experienced multiple waves. For example, consider the states of Arizona, California, Florida and Massachusetts. In Figure 4.1 we see the comparative fit on active cases of the SEIRD and multiwave SEIRD models trained up to September 1, 2020, with the vertical line separating in-sample and out-of-sample predictions. We use vertical green dotted lines to show when the multiwave SEIRD model flagged a new wave. In all of the states, the SEIRD model struggles to match the in-sample fit, because the structure of the original SEIRD model forces it to average between waves. The curve of a single wave cannot fit the multiwave nature of the data, so the SEIRD model alternates between underestimating and overestimating in-sample and this results in very inaccurate out-of-sample values. However, the multiwave SEIRD model is able to more closely match the in-sample data, which results in far more accurate out-of-sample predictions.

This pattern repeats itself in most of the states where multiple waves occur and over time. For example, Figures 4.2 shows how the multiwave SEIRD handles the wave that happened at the end of 2020, with results from the models trained up until January 15, 2021. Across states the SEIRD is unable to capture the multiple waves, once again alternating between underestimating and overestimating to average the waves into something that does

Table 4.3: MAPEs comparing SEIRD and multiwave SEIRD model performance at state level

	Trained till Sept 01, 2020		Trained till Jan 15, 2021	
Region	**SEIRD**	**Multiwave SEIRD**	**SEIRD**	**Multiwave SEIRD**
Alabama	42.6%	6.8%	12.8%	4.7%
Alaska	56.7%	14.5%	40.3%	9.8%
Arizona	66.8%	4.0%	35.1%	2.6%
Arkansas	30.8%	2.6%	4.3%	4.4%
California	41.1%	3.2%	15.6%	1.8%
Colorado	3.6%	13.8%	8.4%	11.7%
Connecticut	8.3%	7.2%	29.4%	4.7%
Delaware	8.7%	5.8%	22.3%	26.4%
District of Columbia	8.5%	1.3%	25.1%	4.6%
Florida	70.7%	3.7%	21.9%	7.5%
Georgia	30.9%	0.9%	17.2%	5.2%
Hawaii	15.4%	104.4%	6.7%	6.1%
Idaho	58.3%	3.2%	18.0%	3.2%
Illinois	19.1%	2.9%	6.4%	1.2%
Indiana	9.5%	1.7%	6.3%	5.2%
Iowa	11.8%	11.2%	21.3%	5.7%
Kansas	2.6%	1.1%	14.2%	2.1%
Kentucky	2.8%	7.0%	7.4%	5.1%
Louisiana	37.1%	18.4%	15.2%	12.5%
Maine	1.2%	3.1%	18.4%	20.3%
Maryland	4.6%	5.9%	15.9%	10.0%
Massachusetts	10.9%	1.1%	35.3%	29.0%
Michigan	14.2%	0.6%	8.3%	21.9%
Minnesota	5.2%	4.7%	33.8%	16.5%
Mississippi	31.9%	6.3%	10.0%	8.5%
Missouri	6.9%	13.3%	21.1%	3.6%
Montana	47.7%	19.2%	30.5%	3.0%
Nebraska	13.1%	6.8%	16.3%	2.3%
Nevada	53.5%	7.3%	4.9%	1.5%
New Hampshire	4.6%	5.9%	20.2%	34.8%
New Jersey	9.3%	1.1%	30.5%	1.0%
New Mexico	243.8%	2.8%	11.4%	5.7%
New York	7.4%	0.6%	38.6%	1.9%
North Carolina	29.2%	11.3%	15.7%	1.5%
North Dakota	23.7%	16.9%	686.5%	2.0%
Ohio	5.9%	6.0%	4.5%	4.1%
Oklahoma	29.5%	10.6%	6.9%	3.5%
Oregon	36.5%	2.4%	14.7%	2.4%
Pennsylvania	10.3%	3.6%	15.2%	1.2%
Rhode Island	15.7%	10.3%	22.7%	1.3%
South Carolina	52.3%	7.1%	27.6%	9.3%
South Dakota	34.0%	2.7%	30.7%	1.6%
Tennessee	34.7%	0.7%	4.7%	2.2%
Texas	56.1%	4.9%	16.8%	2.0%
Utah	23.7%	9.7%	12.8%	3.3%
Vermont	2.0%	0.6%	19.3%	3.9%
Virginia	2.2%	0.7%	19.7%	12.0%
Washington	26.6%	0.8%	4.3%	14.9%
West Virginia	5.2%	18.9%	9.6%	22.8%
Wisconsin	7.8%	7.4%	32.8%	3.7%
Wyoming	4.3%	21.0%	33.9%	3.4%
USA	27.0%	8.4%	31.4%	7.4%

not describe the actual nature of the pandemic. By comparison, the multiwave SEIRD is again able to more closely match actual cases in-sample. Out of sample, even though the multiwave SEIRD tends towards over-prediction by the end of the test set because it hasn't learned that wave from the winter holidays is dying off by mid-February, it is still far more accurate then the SEIRD.

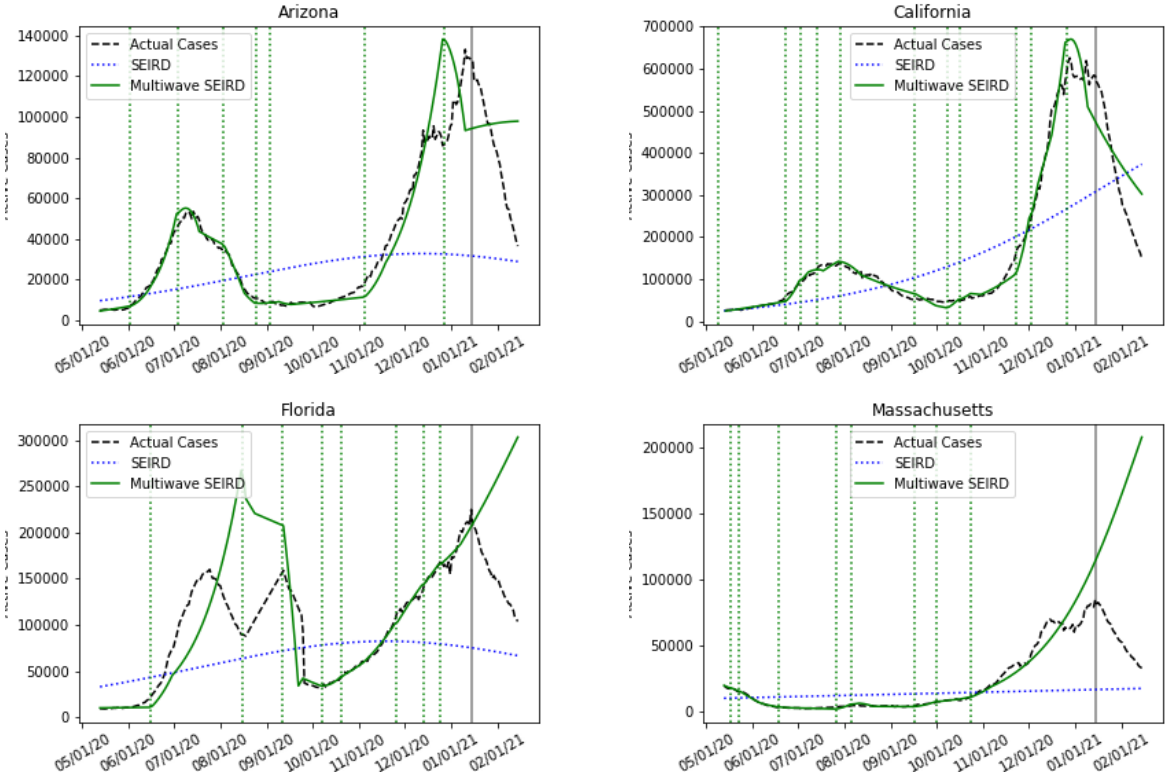


Figure 4.2: SEIRD and multiwave SEIRD in-sample and out-of-sample prediction in Florida, California, Texas and Massachusetts when trained up to January 15, 2021

It is this structured flexibility of the multiwave SEIRD model that allows it to vastly improve the SEIRD model. By accounting for the multiple waves being experienced in the United States, correctly detecting when they occur and allowing the model to change its parameters accordingly, the multiwave SEIRD model is able to fit the rise and fall of cases in-sample for accurate out-of-sample predictions. Similar graphs for all predicted regions can be found in Appendix C.2.1 and Appendix C.2.2.

It is worth noting that the model detects many waves. For example, in California in the January 15 model, the model flags 12 waves. This is a function of the selected lambda, in this case 205. In this work we present results from selecting lambda based on theoretical results (detection time under a week). For practitioners who are looking to maximize accuracy, we recommend using a validation set to tune the hyper-parameter lambda. The higher lambda is, the less sensitive the model will be to changes in infection rate and thus the less waves it is likely to detect. For example in California, if lambda was increased to 225 or 250, there would have been 8 detected waves and if lambda went up to 275, there would have been

5 detected waves. Table C.2 in Appendix C.3, shows the relationship between lambda, the number of detected waves and Out-of-Sample MAPEs for all modelled regions.

Another way to understand how the multiwave model forecasts COVID-19 cases is to analyze the relationship between the detected infection rates for each wave, $\hat{\beta}_n^t$ and the mobility during that time. The multiwave model has the ability to detect shifts in infection rate. This change in infection rate could be due to different government interventions (e.g. mask mandates, work from home orders, restaurant/bar closings, gathering size limits), changes in behavior from the population (e.g. compliance with government interventions, resuming or limiting long-distance travel), or epidemiology-based differences (e.g. vaccinations, new variants, transmission rate's dependence on temperature). Intuitively, the infection rates for the detected waves, $\hat{\beta}_n^t$ would correlate with several of these drivers.

In order to analyze this relationship we compiled data on population mobility, government intervention, weather and holidays. We utilized Google mobility data which tracked relative changes in mobility for a state in different types of locations (https://www.gstatic.com/covid19/mobility/Global_Mobility_Report.csv). Categories for these mobility scores were retail and recreation, grocery and pharmacy, parks, transit stations, workplaces and residential. This was combined with state-level social distancing policies gathered from COVID19StatePolicy Github repository which tracks government mandates that might affect social distancing and COVID-19 transmission (https://github.com/COVID19StatePolicy). Examples of tracked policies include emergency declarations, gathering restrictions, school/-workplace/restaurant/bar closures and public mask mandates. We also used historical weather data from National Climatic Data Center (NCDC) of National Oceanic and Atmospheric Administration (NOAA) to get regional temperatures (ftp.ncdc.noaa.gov) and manually flagged relavent holidays that could affect COVID-19 spread (e.g. Memorial day, Fourth of July, Thanksgiving, Christmas and New Years).

For each state, we created a regularized linear regression using lasso to predict $\hat{\beta}_n^t$, the infection rate found by the multiwave model, using the features we compiled (mobility, government measures, temperature and holiday). The regression needed to be highly regularized because there is strong correlation between all of these features. We see the results of the regression for all states in Table C.1 in Appendix C.3 and summarized in Table 4.4.

Table 4.4: Summary across States of In-Sample and Out-of-Sample R-squared for Lasso Predicting Infection Rates

	In-Sample R-squared	**Out-of-Sample R-squared**
First Quartile	0.61	0.49
Average	0.68	0.59
Third Quartile	0.78	0.70

With an average out-of-sample R-squared of 0.59, we find that many of the predictors (mobility, government intervention, etc.) can be used in predicting the prevailing infection rate estimated by the multiwave model. This shows another evidence of the correctness of the epidemiological model learnt by the multiwave model.

4.6.1 Benchmark with CDC Models

The multiwave SEIRD model is not only strong when considered against the original SEIRD model. We can also analyze these computational results in comparison with the wealth of other COVID-19 forecasting that has been undertaken in 2020. The Center for Disease Control (CDC) has acted as one of the major players in collecting, evaluating and disseminating COVID-19 forecasts for cases and deaths. Models that are submitted to the CDC are considered some of the top models in the country. With over 50 models submitted, stemming from academia and industry, these models have a variety of structures and learning processes. Those that follow the epidemiology compartmental structure often significantly change the basic SIR or SEIRD model to account for many sub-groups (with some citing over 10 compartments) or make the model's parameters a function of data, such as mobility and government interventions, to help improve accuracy. While this creates strong computational results, it means that these models often struggle to generalize or provide analytical guarantees. By comparison, the multiwave SEIRD model, sticks to the 5 compartments described in the name and does not incorporate additional features beyond daily COVID-19 cases and deaths. This simplicity is maintained intentionally to allow for strong and interpretable analytical results in addition to computational results. Despite this fact, we find that the multiwave SEIRD model performs strongly compared to the models uploaded to the CDC. It is because of this strong computational performance that the multiwave SEIRD model is part of the ensemble predictive model, MIT-CASSANDRA, one of the submitted models to the CDC and a component of the CDC's ensemble forecast.

In order to compare with CDC models, we simulated the process for submissions to the CDC. Teams must submit models on Monday, and thus only have data up to the Monday of each week, and give predictions for the number of new weekly cases for the next four Fridays. The CDC uses a variety of forecasts, but the primary one is weekly new cases for the United States. Because the SEIRD model is trained at state level and makes daily predictions, we aggregated predictions across the states and days of each week to get incident cases at the national level. Here we show two sample comparisons, the first for models trained up till October 19, 2020 with predictions for October 24, 2020, October 31, 2020, November 7, 2020 and November 14, 2020. The second sample is for models trained up till January 18, 2021 with predictions for January 23, 2021, January 30, 2021, February 6, 2021 and February 13, 2021. We see the results of this comparison in Table 4.5 and Table 4.6.

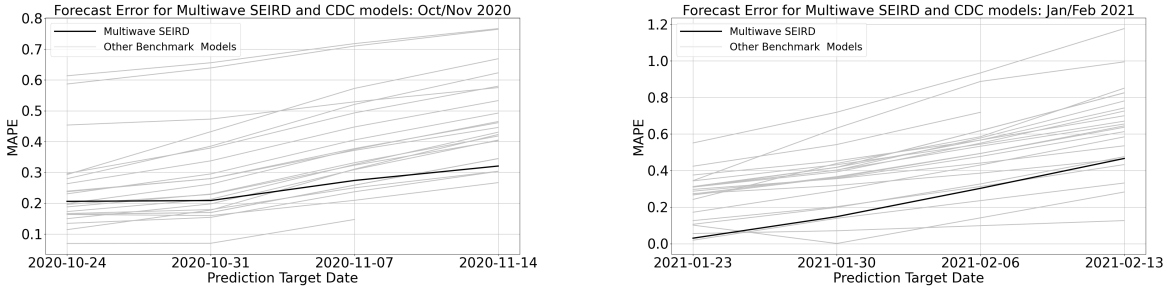
We show these same results visually to highlight the progress of the multiwave SEIRD model's performance over time. In Figure 4.3 we plot the MAPEs of the CDC models and the multiwave SEIRD over the four week predictions. We see in Figure 4.3a the results for predictions made for October/November 2020, the multiwave SEIRD starts out in the middle of the pack in terms of accuracy, but by the end of the forecast is one of the top performing models. This is because the end of October marked the start of the massive wave that hit the United States over the course of the 2020 fall/winter. Due to its wave detection process and epidemiology structure, the multiwave SEIRD model is able to flag a wave is starting and model the exponential growth in cases. This can be seen clearly when we plot the prediction of the October 19, 2020 model against the true cases and CDC's ensemble prediction in Figure 4.4a. The ensemble prediction is the CDC's best estimate of what is going to happen, after considering all of the submitted CDC models, and represents

	10/24/2020		10/31/2020		11/7/2020		11/14/2020	
	MAPE	**Rank**	**MAPE**	**Rank**	**MAPE**	**Rank**	**MAPE**	**Rank**
CU-scenario_high	0.163	6	0.160	3	0.209	2	0.267	1
Karlen-pypm	0.135	4	0.154	2	0.237	3	0.303	2
CU-nochange	0.166	9	0.179	7	0.250	4	0.304	3
Multiwave SEIRD	**0.206**	**14**	**0.209**	**9**	**0.274**	**7**	**0.321**	**4**
CU-scenario_mid	0.164	7	0.170	4	0.259	5	0.345	5
CU-select	0.164	8	0.170	5	0.259	6	0.345	6
JHU_CSSE-DECOM	0.196	12	0.228	11	0.323	10	0.403	7
LNQ-ens1	0.150	5	0.198	8	0.309	8	0.405	8
UMich-RidgeTfReg	0.188	11	0.230	12	0.332	12	0.418	9
RobertWalraven-ESG	0.114	3	0.178	6	0.311	9	0.423	10
CU-scenario_low	0.173	10	0.213	10	0.326	11	0.431	11
JHUAPL-Bucky	0.239	17	0.280	14	0.372	13	0.446	12
JCB-PRM	0.237	16	0.280	15	0.377	15	0.461	13
COVIDhub-baseline	0.199	13	0.262	13	0.375	14	0.465	14
COVIDhub-ensemble	0.230	15	0.295	16	0.405	16	0.493	15
⋮								
TTU-squider	0.614	24	0.656	23	0.718	23	0.767	22

Table 4.5: Multiwave SEIRD and CDC models trained up till October 19, 2020 forecast error and model rank for national weekly incident cases of COVID-19

its official forecast. We see that the CDC ensemble model predicts a flat curve for new cases, having not picked up yet on the scale of the new wave, while the multiwave SEIRD has begun to model the exponential growth.

The predictions made on January 18, 2021 are a little more interesting. Here we see that all the models are struggling to forecast what is going to happen, with MAPES for all models unusually high. The United States was at a tipping point and it was possible the cases would continue to grow or start dying off. This is clearly shown when we look at the state-level predictions in Figure 4.2. Some states have already begun to pivot down and cases have started dying off, but other states look like they were going to grow. For the



(a) Predictions made on Oct 19, 2020

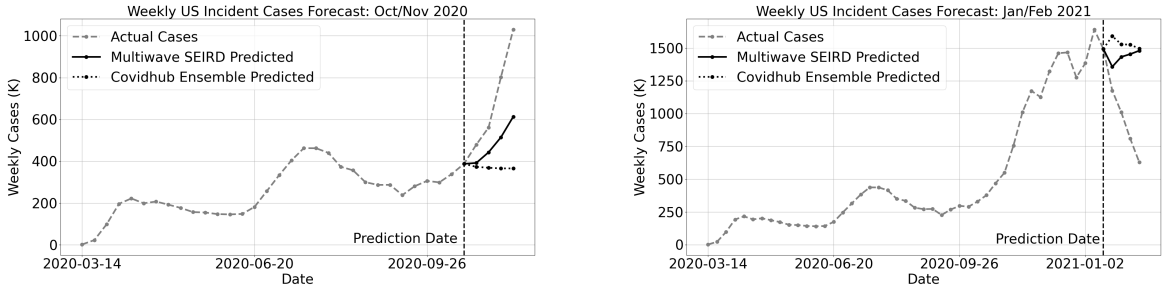
(b) Predictions made on Jan 18, 2021

Figure 4.3: Multiwave SEIRD and CDC models forecast error for national weekly incident cases of COVID-19

	1/23/2021		1/30/2021		2/6/2021		2/13/2021	
Model	**MAPE**	**Rank**	**MAPE**	**Rank**	**MAPE**	**Rank**	**MAPE**	**Rank**
UCLA-SuEIR	0.019	1	0.139	3	0.235	3	0.332	3
Multiwave SEIRD	**0.030**	**2**	**0.148**	**4**	**0.303**	**4**	**0.466**	**6**
TTU-squider	0.055	3	0.071	2	0.098	1	0.126	1
JHU_IDD-CovidSP	0.101	4	0.001	1	0.141	2	0.283	2
LANL-GrowthRate	0.106	5	0.198	5	0.327	6	0.476	7
LNQ-ens1	0.126	6	0.202	6	0.313	5	0.433	4
JHU_CSSE-DECOM	0.172	7	0.291	7	0.430	8	0.584	9
Geneva-DetGrowth	0.229	8	-	-	-	-	-	-
Google_Harvard-CPF	0.242	9	0.439	20	0.585	19	0.851	21
COVIDhub-baseline	0.266	10	0.363	11	0.493	11	0.644	12
Karlen-pypm	0.272	11	0.367	12	0.494	12	0.637	11
RobertWalraven-ESG	0.274	12	0.318	8	0.386	7	0.465	5
COVIDhub-ensemble	0.287	13	0.358	10	0.477	10	0.613	10
JHUAPL-Bucky	0.297	14	0.355	9	0.441	9	0.536	8
CU-nochange	0.309	15	0.393	13	0.550	15	0.726	16
⋮								
UCSB-ACTS	0.551	25	0.719	24	0.934	24	1.177	23

Table 4.6: Multiwave SEIRD and CDC models trained up till January 18, 2021 forecast error and model rank for national weekly incident cases of COVID-19

multiwave SEIRD model, this means that some states have forecasts that show exponential decay, while others continue to show exponential growth. We can see this tension in the national forecasts made by the multiwave SEIRD in Figure 4.4b, as the forecasts show some drop, but then growth again. However, because the SEIRD model is able to flag that the wave was dying off in enough states, it starts off closer to the truth in the first week of predictions, and this translates to strong performance across the rest of the weeks, resulting in the multiwave SEIRD model being among the most accurate models during this time, as shown in Figure 4.3b.



(a) Predictions made on Oct 19, 2020

(b) Predictions made on Jan 18, 2021

Figure 4.4: Multiwave SEIRD and CDC ensemble forecasts for national incident cases of COVID-19

4.7 Conclusions

The COVID-19 pandemic has shown us that disease transmission is not determined just by how contagious a virus is. Population behavior and mitigation strategies play a major role in how quickly a disease spreads. Because of this it is not enough to just model a virus's contagion and assume that population behavior remains constant. The multiwave SIR model creates a basis for understanding, modelling and predicting when epidemics are driven by human nature as much as a virus's transmission rate.

The multiwave SIR model accounts for the multiple waves of COVID-19 seen in the United States. The model provides a framework for detecting when a new wave is occurring, and allows the epidemiology model the flexibility to adjust for new waves. The detection process is theoretically proven to be rigorous, namely quick to pick up on new waves. Experimentally, the multiwave SIR model shows significant improvement over it's single wave counterpart. In regions that have experienced multiple waves, the multiwave SIR model is not forced to average between them like the original SIR model, but rather can match each wave for its infection/transmission/recovery rates. This results in significantly better in-sample fit and out-of-sample accuracy.

Forecasting the COVID-19 evolution, and specifically the occurrences of new waves, is critical in managing operations and supply chains through the pandemic. Identifying and forecasting COVID-19 waves is crucial in answering three major supply-chain related questions for manufacturers, retail companies, restaurants, and healthcare organizations: (i) *What products and resources should these organizations supply?*, (ii) *How should these organizations structure their supply chain in times of pandemic?*, (iii) *What medium should these organizations use to sell or distribute their products?*. Or alternatively, (i) *What to supply?*, (ii) *How to supply?*, and (iii) *How to distribute?*

1. *What to supply?*: the first point is about shifting priorities and changing customer behavior. New waves appearing trigger a panic-buying behavior. As a result, there is a demand surge for many essential supplies such as groceries or hygiene products at common point-of-sale locations ([104]). Being able to correctly predict new waves in particular locations allows organizations such as retailers to anticipate this demand surge and provide enough supply to avoid letting the hoarding behavior put the most vulnerable people at risk. This is also true in a healthcare setting, when hospitals in different locations experience a demand surge in resources like ventilators when a new wave of COVID-19 hits ([97]), requiring a rapid shift in the way these resources are allocated.

2. *How to supply?*: additionally, the supply chain during a pandemic should satisfy two main criteria: safety and flexibility. For the former, recent research such as [105] and [106] shows how the COVID-19 spread impacts the supply chain through the safety measures, and how to make these safety measures evolve accordingly. For the latter, as governments tighten or loosen the COVID-19 restrictions according to new waves appearing, forecasting models with high predictive power such as the multiwave SIR allows specific businesses such as restaurants to anticipate the re-opening and closure of their activities. Thanks to that, these businesses can plan the dependant supply

chains subsequently, e.g., by prioritizing diversification or limiting large inventories of perishables when a new wave is approaching. Finally, there have been a lot of literature on resource allocation during pandemics based on these dynamics, whether it is for general resources ([107]) or vaccine for examples ([108]), where similar predictions are used in optimization-based models.

3. *How to distribute?*: last but not least, as online retail jumped to $26.7 trillion in 2020, up 4% from the previous year ([109]), fuelled by COVID-19, retailers, for example in clothing or home furniture, need to be able to accommodate different distribution channels. The supply chain should shift to online retail during lock-downs and when new waves appear, and allow the safe re-opening of brick-and-mortar stores when restrictions are loosened and demand for in-person shopping increases.

This allows us to conclude that the multiwave SIR model, combined with the right optimization framework, can be used to drive meaningful real-world impact.

4.8 Acknowledgements

The authors would like to thank MIT's Quest for Intelligence for their generous support both in terms of funding but also in terms of research that allowed this chapter to become possible. A special thanks to Akarsh Garg and Shane Weisberg for their help on the computational parts of this work. The authors would also like to thank the review team for their insightful comments that helped us improve both the content and exposition of this work.

Chapter 5

Conclusions

Through this thesis, our objective is to translate goals around sustainable operations into practical decisions by creating data-driven decision-making tools. This thesis delves into the operational management challenges faced by organizations striving to operate competitively within markets while facing sustainable challenges such as increasing renewable energy and epidemic waves.

First we tackle the issue of affordable and clean energy and climate action by refining estimates on the potential of the vehicle-to-grid market. If this market proves fruitful it will help encourage adoption of EVs as vehicles become a source of passive income and allow for greater reliance on sustainable sources of energy as we increase the supply of energy storage to the grid. We do this by introducing a dynamic optimization with a focus on driver-behavior. This model leverages real deidentified driver data to take an individual driver's perspective on how much they can discharge out of their usual driving behavior. We guarantee for this setting that resulting policies serve driver behavior and are interpretable to it's users. Specifically we prove that policies will follow a two-threshold structure that can be easily communicated to drivers. Finally we work closely with an American EV manufacturer to quantify the impact of the market in terms of both monetary savings as well as CO2 savings.

Second, we tackle the problem of optimizing when objectives are determined by a tree ensemble model. This is relevant for decision makers that are looking to leverage their granular, accurate and deterministic tree models to make more optimal decisions. For this we propose the general approximation method UMOTEM, which is able to solve optimizations determined by tree ensemble models in a tractable manner. Specifically we show that the number of binary variables will grow linearly rather than exponentially. We are also able to bound the optimality gap of the approximation. We show that the model is able to perform strongly on a variety of publicly available datasets. We also bring the issue back to the sustainable goals of clean energy and climate action, by discussing an EV case study where we show the power of UMOTEM in estimating the potential size of the vehicle discharging market. We also contribute to business growth by showing how UMOTEM can be used in a retail case study to plan markdown promotions for a fashion retailer.

Finally, we contribute to the wealth of operations research around the COVID-19 pandemic. Specifically we introduce a way of modelling the multiple waves of the disease while remaining agnostic to the specific drivers of the new wave. This is particularly important

as we saw during the COVID-19 pandemic, decision makers would often have to respond to changes in the disease without fully understanding what was causing the change. The multiwave SIR model proposed pairs the traditional epidemiology model with a martingale framework to flag new waves. We prove that the model will be able to flag new waves of COVID-19 in less than a week. We also compare the model to other cutting-edge methods and show that it among the top performers of models submitted to the CDC.

This thesis aims to contribute to the literature on data-driven decision making in sustainable operations. Through this work and our close collaboration with industry partners, we aid decision makers through fact based modelling to improve sustainable practices in both literature and practice.

Appendix A

Proofs and Figures of Chapter 2

A.1 Proofs

Proof of Optimality: We prove this case by case. Constants in the objective, resulting in a linear slope regarding charge level, will not affect the concavity of the solution, so for simplicity of notation we will ignore the $\lambda_{\max,t+1}\left(C_{\max}-c\right)-\lambda_{\min,t+1}\left(C_{\min}-c\right)^{+}+\lambda_c c$ terms in the objective in this section.

Case 1: $c < c^{*charge} \leq c^{*V2X}$*:* We first observe that in this scenario the policy of discharging is dominated by doing nothing. Discharging down to $\hat{c} < c$ would result in an objective of $(c-\hat{c})\left(w_{t+1}^{\mathrm{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{\hat{c},t+2}\right)\right]$. We show that this is less than the value of doing nothing: $\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]$:

$$
\begin{aligned}
&\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]-(c-\hat{c})\left(w_{t+1}^{\mathrm{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{\hat{c},t+2}\right)\right]\\
&\geq(c-\hat{c})\frac{1}{\lambda_{\mathrm{time},t+1}}\left(E\left[V_1\left(s_{c,t+2}\right)\right]-E\left[V_1\left(s_{c-1,t+2}\right)\right]\right)-(c-\hat{c})\left(w_{t+1}^{\mathrm{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)\\
&=(c-\hat{c})\left(\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]-\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c-1,t+2}\right)\right]-w_{t+1}^{\mathrm{V2X}}-\lambda_R r_{t+1}+\lambda_c\right)\\
&\geq(c-\hat{c})\left(\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c^{*\mathrm{V2X}},t+2}\right)\right]-\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c^{*\mathrm{V2X}}-1,t+2}\right)\right]-w_{t+1}^{\mathrm{V2X}}-\lambda_R r_{t+1}+\lambda_c\right)\\
&\geq 0
\end{aligned}
$$

The first and second inequalities comes from the concavity of $E\left[V_1\left(s_{\hat{c},t+2}\right)\right]$ while the last comes from the definition of $c^{*\mathrm{V2X}}$.

Given that discharging is sub-optimal the only other policy remaining is charging or doing nothing. In such a scenario the objective of the optimization becomes concave for the decision variable $x_{c,t+1,1}^{*charge}$:

$$
x_{c,t+1,1}^{*charge}\left(w_{t+1}^{\mathrm{charge}}+\lambda_R r_{t+1}-\lambda_c\right)-\lambda_{\max,t+1}\left(C_{\max}-c\right)-\lambda_{\min,t+1}\left(C_{\min}-c\right)^{+}+\frac{1}{\lambda_{\mathrm{time},t+1}}E\left[V_1\left(s_{c-x_{c,t+1,1}^{*charge},t+2}\right)\right]+\lambda_c c.
$$

As a concave function we find its maximum when the first derivative is zero. This is at $c^{*\mathrm{charge}}$ by its definition. Therefore the optimal policy is to get as close to $c^{*\mathrm{charge}}$ as possible:

$-\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right)$. We have to take the minimum because we are constrained to charging at most $X_{\max}^{\text{charge}}$.

Case 2: $c^{*charge} \leq c \leq c^{*V2X}$: We continue to observe that it does not make sense to discharge the battery. The proof of this follows the same as the previous Case 1.

We next observe that in this scenario it also does not make sense to charge. Charging up to $\hat{c} > c$ would result in an objective of $-(\hat{c}-c)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\hat{c},t+2}\right)\right]$. We show that this is less than the value of doing nothing: $\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]$:

$$
\begin{aligned}
&\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]+(\hat{c}-c)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)-\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\hat{c},t+2}\right)\right]\\
&\geq(\hat{c}-c)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)-(\hat{c}-c)\frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c+1,t+2}\right)\right]-E\left[V_1\left(s_{c,t+2}\right)\right]\right)\\
&=(\hat{c}-c)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c-\frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c+1,t+2}\right)\right]-\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c,t+2}\right)\right]\right)\right)\\
&\geq(\hat{c}-c)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c-\frac{1}{\lambda_{\text{time},t+1}}\left(E\left[V_1\left(s_{c^{*\text{charge}}+1,t+2}\right)\right]-\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c^{*\text{charge}},t+2}\right)\right]\right)\right)\\
&\geq 0
\end{aligned}
$$

As before, the first and second inequalities comes from the concavity of $E\left[V_1\left(s_{\hat{c},t+2}\right)\right]$ while the last comes from the definition of $c^{*\text{V2X}}$.

Case 3: $c^{*charge} \leq c^{*V2X} < c$: Like the previous scenario, we observe that it does not make sense to charge. The proof of this follows directly as the previous Case 2.

Given that charging is sub-optimal the only other policy remaining is V2X or doing nothing. In such a scenario the objective of the optimization becomes concave for the decision variable $x_{c,t+1,1}^{*V2X}$: $\text{x}_{c,t+1,1}^{*V2X}\left(w_t^{\text{V2X}}+\lambda_R r_t-\lambda_c\right)-\lambda_{\max,t}\left(C_{\max}-c\right)-\lambda_{\min,t}\left(C_{\min}-c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c-x_{c,t+1,1}^{*V2X},t+2}\right)\right]+\lambda_c c$. As a concave function we find its maximum when the first derivative is zero. This is at $c^{*\text{V2X}}$ by its definition. Therefore the optimal policy is to get as close to $c^{*\text{V2X}}$ as possible: $-\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right)$. We have to take the minimum because we are constrained to charging at most $X_{\max}^{\text{V2X}}$.

We now consider the scenario where $c^{*\text{charge}} > c^{*\text{V2X}}$.

Case 4: $c < c^{*V2X} \leq c^{*switch} \leq c^{*charge}$: This case is exactly the same as Case 1. Doing nothing dominates discharging, and charging dominates doing nothing. The proof for this scenario follows exactly as Case 1.

Case 5: $c^{*V2X} \leq c < c^{*switch} \leq c^{*charge}$: This challenge with this scenario is that there are states where both charging and discharging dominate over doing nothing. Thus we define $c^{*\text{switch}}$ as the first time discharging as a policy dominates over charging. We note that the optimal charging policy at c is $\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right)$. This is because the objective function for charging is concave, and the function's argmax, $c^{*\text{charge}}$ is higher than c. Using this optimal charging policy would result in an objective of

$$
-\left(\min\left(c^{*\text{charge}}-c, X_{\max}^{\text{charge}}\right)\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{charge}},c+X_{\max}^{\text{charge}}\right),t+2}\right)\right]+\lambda_c c.
$$

Similarly the optimal discharging policy is $\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right)$ as the argmax for discharging, $c^{*\text{V2X}}$ is less than c. The optimal discharging policy would result in an objective of $\left(\min\left(c-c^{*\text{V2X}}, X_{\max}^{\text{V2X}}\right)\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\max\left(c^{*\text{V2X}},c-X_{\max}^{\text{V2X}}\right),t+2}\right)\right]+\lambda_c(c)$.

Therefore we define $c^{*\text{switch}}$ as the minimum value between $c^{*\text{V2X}}$ and $c^{*\text{charge}}$ such that:

$$
\begin{aligned}
&-\min\left(c^{*\text{charge}}-c^{*\text{switch}},X_{\max}^{\text{charge}}\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{charge}},c^{*\text{switch}}+X_{\max}^{\text{charge}}\right),t+2}\right)\right]\\
&\leq \min\left(c^{*\text{switch}}-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{V2X}},c^{*\text{switch}}-X_{\max}^{\text{V2X}}\right),t+2}\right)\right].
\end{aligned}
$$

If $c^{*\text{V2X}} \leq c < c^{*\text{switch}}$, then by definition, we have that

$$
\begin{aligned}
&-\min\left(c^{*\text{charge}}-c,X_{\max}^{\text{charge}}\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c+\min\left(c^{*\text{charge}}-c,X_{\max}^{\text{charge}}\right),t+2}\right)\right]\\
&>\min\left(c-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{c-\min(c-c^{*\text{V2X}},X_{\max}^{\text{V2X}}),t+2}\right)\right].
\end{aligned}
$$

and so the optimal policy is to charge.

Case 6: $c^{*V2X} \leq c^{*switch} \leq c < c^{*charge}$*:* Here we show that after the policy switches from charging to discharging, the policy will not switch back. Like in Case 5, we note that the optimal charging policy at c is $\min\left(c^{*\text{charge}}-c,X_{\max}^{\text{charge}}\right)$ and has an objective value of $-\left(\min\left(c^{*\text{charge}}-c,X_{\max}^{\text{charge}}\right)\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{charge}},c+X_{\max}^{\text{charge}}\right),t+2}\right)\right]$. Similarly the optimal discharging policy is $\min\left(c-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)$ and has an objective value of $\left(\min\left(c-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\max(c^{*\text{V2X}},c-X_{\max}^{\text{V2X}}),t+2}\right)\right]$. Here we show that the discharging policy dominates the charging policy:

$$
\begin{aligned}
&\left(\min\left(c-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\max(c^{*\text{V2X}},c-X_{\max}^{\text{V2X}}),t+2}\right)\right]\\
&-\left(-\left(\min\left(c^{*\text{charge}}-c,X_{\max}^{\text{charge}}\right)\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{charge}},c+X_{\max}^{\text{charge}}\right),t+2}\right)\right]\right)\\
&\geq \min\left(c^{*\text{switch}}-c^{*\text{V2X}},X_{\max}^{\text{V2X}}\right)\left(w_{t+1}^{\text{V2X}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{V2X}},c^{*\text{switch}}-X_{\max}^{\text{V2X}}\right),t+2}\right)\right]\\
&-\left(-\min\left(c^{*\text{charge}}-c^{*\text{switch}},X_{\max}^{\text{charge}}\right)\left(w_{t+1}^{\text{charge}}+\lambda_R r_{t+1}-\lambda_c\right)+\frac{1}{\lambda_{\text{time},t+1}}E\left[V_1\left(s_{\min\left(c^{*\text{charge}},c^{*\text{switch}}+X_{\max}^{\text{charge}}\right),t+2}\right)\right]\right)\\
&\geq 0
\end{aligned}
$$

The first inequality comes from the concavity of both the charging policy and discharging policy. In the case of charging, moving from c to $c^{*\text{switch}}$ moves the value farther away from the argmax of the function, $c^{*\text{charge}}$, reducing the value of the objective. Conversely in the case of discharging, moving from c to $c^{*\text{switch}}$ moves the value to the argmax of the function, $c^{*\text{V2X}}$, increasing the value of the objective. The second inequality comes from the definition of $c^{*\text{switch}}$.

Because discharging dominates we therefore follow the optimal discharging policy in Case 6.

Case 7: $c^{*V2X} \leq c^{*switch} \leq c^{*charge} \leq c$*:* This case is indistinguishable from Case 3, and therefore follows the same policy and proof.

We finally note that the resulting policies from Cases 4-7 can combine and simplify down. This is because the policy in Cases 4 and 5 are the same and the policy in Cases 6 and 7 are the same. This results in the policy described in Equation 2.11.

Appendix B

Proofs and Figures of Chapter 3

B.1 Formulations

Formulation (3.1) for the Sample Forest in Figure 3.1
For this example we assume that we are maximizing expected sales, so the payoff S_j^t is just the sales of leaf j without any additional business constraints.

$$
\begin{aligned}
\max_{z,y} \quad & \frac{1}{3}\left(3y_{1,3} + 1y_{1,4} + 2y_{1,5} + 2y_{2,2} + 5y_{2,3} + 2y_{3,4} + 4y_{3,5} + 3y_{3,6} + 1y_{3,7}\right) \\
\text{s.t.} \quad & y_{1,3} + y_{1,4} + y_{1,5} = 1 \\
& y_{2,2} + y_{2,3} = 1 \\
& y_{3,4} + y_{3,5} + y_{3,6} + y_{3,7} = 1 \\
& y_{1,4} + y_{1,5} \leq z_{\text{Discount},3} \\
& y_{1,3} \leq 1 - z_{\text{Discount},3} \\
& y_{1,4} \leq z_{\text{Price},1} \\
& y_{1,5} \leq 1 - z_{\text{Price},1} \\
& y_{2,2} \leq z_{\text{Price},2} \\
& y_{2,3} \leq 1 - z_{\text{Price},2} \\
& y_{3,4} + y_{3,5} \leq z_{\text{Discount},2} \\
& y_{3,6} + y3,7 \leq 1 - z_{\text{Discount},2} \\
& y_{3,4} \leq z_{\text{Price},1} \\
& y_{3,5} \leq 1 - z_{\text{Price},1} \\
& y_{3,6} \leq z_{\text{Discount},1} \\
& y_{3,7} \leq 1 - z_{\text{Discount},1} \\
& z_{\text{Price},1} \geq z_{\text{Price},2} \\
& z_{\text{Discount},1} \geq z_{\text{Discount},2} \\
& z_{\text{Discount},2} \geq z_{\text{Discount},3} \\
& z_{\text{Price},1}, z_{\text{Price},2}, z_{\text{Discount},1}, z_{\text{Discount},2}, z_{\text{Discount},3} \in \{0,1\} \\
& y_{1,3}, y_{1,4}, y_{1,5}, y_{2,2}, y_{2,3}, y_{3,4}, y_{3,5}, y_{3,6}, y_{3,7} \geq 0
\end{aligned}
\tag{B.1}
$$

Formulation for the exact MIO as proposed by [33]

$$
\begin{aligned}
\max_{x,q} \quad & \sum_{t=1}^{T} \sum_{j \in L^t} w_t q^t_{p_j,j} S^t_j \\
\text{s.t.} \quad & (a^t_i)^\top x + M(1 - q^t_{i,l_i}) \geq b^t_i, && \forall t \in \{1,\dots,T\}, i \in N^t \\
& (a^t_i)^\top x - M(1 - q^t_{i,r_i}) \leq b^t_i, && \forall t \in \{1,\dots,T\}, i \in N^t \\
& q^t_{p_i,i} = q^t_{i,l_i} + q^t_{i,r_i}, && \forall t \in \{1,\dots,T\}, i \in N^t \\
& \sum_{j \in L^t} q^t_{p_j,j} = 1, && \forall t \in \{1,\dots,T\} \\
& Ax \leq f \\
& q^t_{p_i,i}, q^t_{i,l_i}, q^t_{i,r_i} \in 0,1 && \forall t \in \{1,\dots,T\}, i \in N^t
\end{aligned}
\tag{B.2}
$$

Formulation (B.2) for the Sample Forest in Figure 3.1

Again, we assume that we are maximizing expected sales, so the payoff S_j^t is just the sales of leaf j without any additional business constraints.

$$
\begin{aligned}
\max_{x,q} \quad & \frac{1}{3}\left(3q_{1,3}^1 + 1q_{2,4}^1 + 2q_{2,5}^1 + 2q_{1,2}^2 + 5q_{1,3}^2 + 2q_{2,4}^3 + 4q_{2,5}^3 + 3q_{3,6}^3 + 1q_{3,7}^3\right) \\
\text{s.t.} \quad & x_{\text{discount}} + M(1 - q_{1,2}^1) \geq 0.25 \\
& x_{\text{discount}} - M(1 - q_{1,3}^1) \leq 0.25 \\
& x_{\text{price}} + M(1 - q_{2,4}^1) \geq 20 \\
& x_{\text{price}} - M(1 - q_{2,5}^1) \leq 20 \\
& x_{\text{price}} + M(1 - q_{1,2}^2) \geq 19 \\
& x_{\text{price}} - M(1 - q_{1,3}^2) \leq 19 \\
& x_{\text{discount}} + M(1 - q_{1,2}^3) \geq 0.2 \\
& x_{\text{discount}} - M(1 - q_{1,3}^3) \leq 0.2 \\
& x_{\text{price}} + M(1 - q_{2,4}^3) \geq 21 \\
& x_{\text{price}} - M(1 - q_{2,5}^3) \leq 21 \\
& x_{\text{discount}} + M(1 - q_{3,6}^3) \geq 0.1 \\
& x_{\text{discount}} - M(1 - q_{3,7}^3) \leq 0.1 \\
& 1 = q_{1,2}^1 + q_{1,3}^1 \\
& q_{1,2}^1 = q_{2,4}^1 + q_{2,5}^1 \\
& 1 = q_{1,2}^2 + q_{1,3}^2 \\
& 1 = q_{1,2}^3 + q_{1,3}^3 \\
& q_{1,2}^3 = q_{2,4}^3 + q_{2,5}^3 \\
& q_{1,3}^3 = q_{3,6}^3 + q_{3,7}^3 \\
& q_{1,3}^1 + q_{2,4}^1 + q_{2,5}^1 = 1, \\
& q_{2,4}^3 + q_{2,5}^3 + q_{3,6}^3 + q_{3,7}^3 = 1, \\
& q_{1,2}^1, q_{1,3}^1, q_{2,4}^1, q_{2,5}^1, q_{1,2}^2, q_{1,3}^2, q_{1,2}^3, q_{1,3}^3, q_{2,4}^3, q_{2,5}^3, q_{3,6}^3, q_{3,7}^3 \in 0, 1
\end{aligned}
\tag{B.3}
$$

B.2 Zoomed in Graph

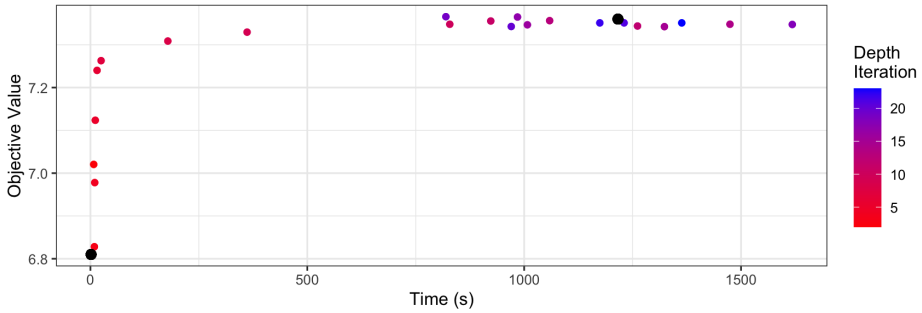


Figure B.1: Wine, 500 trees: objective and runtimes of UMOTEM for different depth iterations

B.3 Proofs

Proof of Step 3.4

For a given tree t in the random forest, the set of x such that $h_t^{*d}(x) = j$ is a set of hyper-rectangles. This is because a "leaf" of the tree is described as a hyper-rectangle region of the feature space, and within a leaf a constant prediction value is assigned. This holds for all the trees in the random forest. For all the trees in the forest we can define a finite number of unions of hyper-rectangles, $S_1, \dots S_K$ such that if $x \in S_k$, then there is a tree t such that $h_t^{*d}(x) = j$. We can definite a function $\phi(i) = k$ if $\{x : h_t^{*d}(x) = j\} = S_k$

We define N_k as the number of times that $\phi(i) = k$ in the first N trees. Then:

$$\frac{1}{N}\sum_{t=1}^{N} 1_{(h_t(x)=j)} = \frac{1}{N}\sum_{k=1}^{K} N_k 1_{(x\in S_k)}$$

In doing this, we move from averaging over the trees, to averaging over unions of hyper-rectangles. For $\frac{N_k}{N}$ we have:

$$\frac{N_k}{N} = \frac{1}{N}\sum_{n=1}^{N} 1_{(\phi(i)=k)}$$

Which by law of large numbers, converges almost surely to $P_\Theta\left(\phi(i) = k\right)$

We have then that the best leaf predictive function of the random forest converges as the number of trees grows to:

$$\begin{aligned}
h^{*d}(x) &= \frac{1}{N}\sum_{t}^{N} h_t^{*d}(x) \\
&= \frac{1}{N}\sum_{t}^{N} \int_{-\infty}^{\infty} j 1_{(h_t^{*d}(x)=j)} dj \\
&= \int_{-\infty}^{\infty} j \frac{1}{N}\sum_{k=1}^{K} N_k 1_{(x\in S_k)} dj \\
&\to \int_{-\infty}^{\infty} j \sum_{k=1}^{K} P_\Theta(\phi(\Theta) = k) 1_{(x\in S_k)} dj \\
&= \int_{-\infty}^{\infty} j P_\Theta(h_\Theta^{*d}(x) = j) dj \\
&= E_\Theta\left[h_\Theta^{*d}(x)\right]
\end{aligned}$$

For the upper bound function, it like the random forest prediction is a constant prediction across the entire hyper-rectangle, but is calculated from only the top m values rather than all the data points in the hyper-rectangle. Therefore, there are a finite number of unions of hyper-rectangles, $R_1, \dots, R_K$ such that j is the upper bound estimate for the training inputs in R_k. The rest of the proof follows directly from [35].

B.4 Tables

In the section below we present a series of computational comparisons between the exact MIOs presented by [32] and [33], their approximations and UMOTEM. Specifically Tables B.1, B.2 and B.3 show the full version of the Table 3.2, comparing the MIOs with UMOTEM and the predict and optimize method in terms of optimal objective and runtime. The subsequent tables provide more granular comparisons against the depth approximation method presented in [32] and the sampling approximation presented in [33]. If the results are listed as '-' then the approximation did not make sense (such as a 10% approximation when the maximum depth is 4). We see that if the approximation is selected correctly, the depth and sampling approximations often preform comparably to the exact MIO in terms of objective value. However if they are mis-selected, they risk producing sub-optimal recommendations. By comparison, UMOTEM is able to run consistently and produce reliable results even as the number of trees in the forest grows to 500 trees, the standard base parameter for number of trees in programs like R.

# trees	S^*_{MIO}	S^*_{UMOTEM}	Opt. Gap	Time (Mišić)	Time (Biggs et al.)	Time (UMOTEM)	Time (Pred. and Opt.)
10	1.16	1.16	0.0%	8.80	170.44	0.60	0.32
20	1.15	1.14	0.7%	10.05	430.88	1.01	0.58
30	1.12	1.12	0.3%	11.38	845.16	1.82	1.80
40	1.12	1.11	0.3%	17.40	NA	2.08	1.82
50	1.12	1.11	0.5%	17.77	NA	2.07	1.43
60	1.12	1.11	0.9%	17.97	NA	2.62	1.68
70	1.12	1.10	0.9%	18.95	NA	2.92	1.91
80	1.12	1.11	1.0%	21.34	NA	3.57	2.42
90	1.12	1.11	0.9%	21.71	NA	3.90	4.36
100	1.13	1.12	0.8%	28.35	NA	3.89	2.68
110	1.13	1.12	1.0%	28.11	NA	4.24	2.56
120	1.13	1.12	1.1%	30.40	NA	4.45	2.75
130	1.13	1.12	1.0%	32.02	NA	5.72	4.08
140	1.13	1.12	0.9%	37.07	NA	6.13	3.93
150	1.13	1.12	1.0%	47.66	NA	6.10	3.95
160	1.13	1.11	1.1%	33.91	NA	6.45	3.97
170	1.13	1.11	1.4%	46.48	NA	6.55	3.91
180	1.13	1.11	1.2%	46.40	NA	7.51	5.52
190	1.13	1.12	1.2%	51.89	NA	8.44	6.23
200	1.13	1.11	1.2%	50.98	NA	8.85	6.70
210	1.12	1.11	1.0%	58.90	NA	9.34	6.87
220	1.12	1.11	0.9%	63.48	NA	9.83	7.61
230	1.12	1.11	0.9%	67.62	NA	9.92	7.70
240	1.13	1.12	0.8%	61.13	NA	9.42	6.81
250	1.13	1.12	1.0%	63.72	NA	9.85	6.48
260	1.13	1.12	1.0%	76.51	NA	10.34	7.59
270	1.13	1.12	0.9%	71.29	NA	10.93	7.09
280	1.13	1.12	0.8%	79.50	NA	10.72	7.44
290	1.13	1.12	0.7%	72.89	NA	11.59	7.69
300	1.13	1.12	0.7%	100.98	NA	12.11	8.40
310	1.12	1.12	0.7%	84.23	NA	12.83	9.12
320	1.12	1.12	0.7%	78.76	NA	13.09	8.59
330	1.13	1.11	1.1%	100.26	NA	13.91	9.80
340	1.13	1.11	1.2%	82.58	NA	12.85	8.48
350	1.13	1.11	1.1%	88.34	NA	14.61	9.85
360	1.13	1.11	1.1%	86.12	NA	14.87	10.76
370	1.13	1.11	1.0%	144.62	NA	15.11	10.65
380	1.13	1.12	1.0%	135.74	NA	14.58	9.86
390	1.13	1.11	1.0%	163.63	NA	14.08	8.72
400	1.13	1.11	0.9%	136.00	NA	14.56	9.02
410	1.13	1.12	0.9%	175.69	NA	14.88	9.45
420	1.12	1.11	0.9%	126.56	NA	17.94	11.39
430	1.13	1.11	0.9%	168.93	NA	17.31	11.64
440	1.12	1.11	0.9%	363.95	NA	18.74	11.78
450	1.12	1.11	0.9%	162.43	NA	19.21	12.35
460	1.12	1.11	0.9%	234.65	NA	19.69	12.72
470	1.12	1.11	1.0%	157.78	NA	20.19	12.95
480	1.13	1.11	1.0%	336.14	NA	19.45	11.89
490	1.12	1.11	1.0%	162.12	NA	19.72	12.50
500	1.12	1.11	1.0%	273.53	NA	20.07	12.95

Table B.1: Solubility Dataset: Objective function value and times

# trees	S^*_{MIO}	S^*_{UMOTEM}	Opt. Gap	Time (Mišić)	Time (Biggs et al.)	Time (UMOTEM)	Time (Pred. and Opt.)
10	78.62	78.62	0.0%	8.23	12.68	0.06	0.09
20	78.49	77.01	1.9%	9.98	27.85	0.13	0.05
30	78.39	76.75	2.1%	9.13	27.64	0.21	0.09
40	78.32	76.91	1.8%	10.83	57.30	0.23	0.09
50	78.36	77.12	1.6%	12.45	52.46	0.22	0.12
60	78.23	76.92	1.7%	11.12	175.75	0.28	0.11
70	78.30	76.93	1.8%	12.55	188.95	0.31	0.14
80	78.29	76.64	2.1%	13.98	204.79	0.38	0.15
90	78.27	76.75	1.9%	14.74	322.96	0.41	0.22
100	78.25	76.83	1.8%	14.31	454.15	0.42	0.22
110	78.28	76.80	1.9%	15.20	335.95	0.48	0.20
120	78.23	76.73	1.9%	16.11	661.49	0.46	0.22
130	78.21	76.69	2.0%	17.49	837.14	0.57	0.30
140	78.14	76.54	2.0%	19.59	908.61	1.66	0.29
150	78.11	76.61	1.9%	20.96	NA	0.67	0.30
160	78.18	76.65	2.0%	24.38	NA	0.71	0.34
170	78.18	76.68	1.9%	26.33	NA	0.76	0.28
180	78.19	76.71	1.9%	28.18	NA	0.75	0.39
190	78.21	76.79	1.8%	26.14	NA	1.04	0.44
200	78.21	76.78	1.8%	28.35	NA	0.92	0.39
210	78.21	76.83	1.8%	25.71	NA	0.88	0.39
220	78.23	76.89	1.7%	32.50	NA	1.10	0.48
230	78.24	76.86	1.8%	34.31	NA	1.03	0.47
240	78.25	76.90	1.7%	28.93	NA	0.86	0.40
250	78.22	76.77	1.8%	30.65	NA	0.90	0.42
260	78.23	76.82	1.8%	36.22	NA	0.94	0.39
270	78.21	76.76	1.8%	31.94	NA	0.96	0.46
280	78.21	76.76	1.9%	41.29	NA	1.20	0.50
290	78.22	76.72	1.9%	34.33	NA	1.23	0.56
300	78.20	76.72	1.9%	44.40	NA	1.30	0.49
310	78.21	76.70	1.9%	35.24	NA	1.38	0.65
320	78.22	76.74	1.9%	36.40	NA	1.42	0.55
330	78.24	76.76	1.9%	36.27	NA	1.42	0.65
340	78.24	76.77	1.9%	40.35	NA	1.28	0.54
350	78.24	76.73	1.9%	51.22	NA	1.46	0.63
360	78.26	76.78	1.9%	51.94	NA	1.60	0.70
370	78.26	76.81	1.9%	46.78	NA	1.51	0.72
380	78.26	76.80	1.9%	60.73	NA	2.83	0.67
390	78.25	76.79	1.9%	51.59	NA	1.45	0.63
400	78.26	76.72	2.0%	51.48	NA	1.47	0.67
410	78.27	76.75	1.9%	58.34	NA	1.55	0.69
420	78.26	76.76	1.9%	60.99	NA	1.61	0.75
430	78.24	76.76	1.9%	55.50	NA	1.67	0.80
440	78.24	76.78	1.9%	58.07	NA	1.70	0.76
450	78.25	76.80	1.9%	59.42	NA	1.75	0.85
460	78.26	76.83	1.8%	65.71	NA	1.85	0.81
470	78.27	76.83	1.8%	56.90	NA	1.86	0.90
480	78.26	76.83	1.8%	62.20	NA	1.78	0.85
490	78.26	76.83	1.8%	70.14	NA	1.82	0.87
500	78.26	76.83	1.8%	83.58	NA	1.89	0.83

Table B.2: Concrete Dataset: Objective function value and times

# trees	S^*_{MIO}	S^*_{UMOTEM}	Opt. Gap	Time (Mišić)	Time (Biggs et al.)	Time (UMOTEM)	Time (Pred. and Opt.)
10	7.43	6.69	10.1%	19.03	430.14	0.09	0.05
20	7.45	6.60	11.5%	27.90	NA	0.17	0.14
30	7.42	6.74	9.2%	51.05	NA	0.33	0.17
40	7.42	6.77	8.8%	52.25	NA	0.39	0.21
50	7.41	6.76	8.8%	59.61	NA	0.38	0.18
60	7.38	6.75	8.5%	74.04	NA	0.51	0.22
70	7.37	6.65	9.8%	99.74	NA	0.55	0.27
80	7.38	6.63	10.1%	101.49	NA	0.76	0.34
90	7.36	6.72	8.7%	160.62	NA	0.81	0.30
100	7.37	6.69	9.2%	149.70	NA	0.85	0.31
110	7.37	6.67	9.4%	148.24	NA	0.85	0.36
120	7.36	6.68	9.3%	150.39	NA	0.93	0.36
130	7.36	6.71	8.9%	190.65	NA	1.18	2.57
140	7.36	6.69	9.1%	283.68	NA	1.12	0.47
150	7.36	6.69	9.1%	239.16	NA	1.13	0.49
160	7.37	6.70	9.0%	275.28	NA	1.32	0.55
170	7.37	6.69	9.3%	256.81	NA	1.30	0.53
180	7.37	6.70	9.2%	286.65	NA	2.00	0.67
190	7.37	6.71	8.9%	309.63	NA	1.44	0.71
200	7.37	6.71	9.0%	271.79	NA	1.62	0.77
210	7.38	6.72	9.0%	313.02	NA	1.64	0.80
220	7.38	6.70	9.2%	311.98	NA	1.74	0.87
230	7.38	6.72	8.9%	483.85	NA	1.97	0.74
240	7.37	6.77	8.1%	410.55	NA	1.70	0.66
250	7.38	6.76	8.4%	407.12	NA	1.78	0.74
260	7.37	6.76	8.3%	516.57	NA	1.83	0.78
270	7.37	6.77	8.2%	424.92	NA	1.91	0.78
280	7.37	6.76	8.3%	529.45	NA	2.14	0.84
290	7.37	6.77	8.2%	489.21	NA	2.28	0.96
300	7.37	6.78	8.1%	503.57	NA	2.44	0.93
310	7.37	6.77	8.1%	624.52	NA	2.68	1.01
320	7.37	6.78	8.0%	687.43	NA	2.75	1.21
330	7.37	6.76	8.3%	470.36	NA	2.82	1.20
340	7.37	6.76	8.3%	552.13	NA	2.63	1.06
350	7.36	6.75	8.3%	725.11	NA	3.00	1.17
360	7.36	6.75	8.3%	616.68	NA	3.02	1.32
370	7.36	6.75	8.2%	623.33	NA	3.28	1.22
380	7.36	6.76	8.2%	922.70	NA	3.11	1.21
390	7.36	6.75	8.2%	1044.48	NA	2.93	1.15
400	7.36	6.75	8.2%	989.18	NA	2.98	1.16
410	7.36	6.76	8.1%	1024.08	NA	3.10	1.24
420	7.36	6.76	8.1%	1026.23	NA	3.23	1.36
430	7.36	6.76	8.1%	1161.09	NA	3.31	1.54
440	7.36	6.77	8.0%	1256.51	NA	3.50	1.41
450	7.36	6.77	8.0%	1188.74	NA	3.43	1.44
460	7.36	6.77	8.0%	1196.92	NA	3.56	2.59
470	7.36	6.83	7.2%	828.52	NA	3.64	1.44
480	7.36	6.84	7.1%	1220.30	NA	3.54	1.49
490	7.36	6.80	7.6%	1255.77	NA	3.60	1.54
500	7.36	6.81	7.5%	1216.24	NA	4.83	1.54

Table B.3: Wine Dataset: Objective function value and times

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	-3.41	-3.41	-3.51	-1.66	0.60	1.02	1.09	1.16	1.16	1.16	1.16
20	-4.14	-4.14	-3.44	-0.23	0.59	1.03	1.14	1.15	1.15	1.15	1.14
30	-4.30	-4.30	-3.40	-0.41	0.57	1.10	1.12	1.12	1.12	1.12	1.12
40	-4.26	-4.26	-3.24	-0.40	0.76	1.12	1.12	1.12	1.12	1.12	1.11
50	-4.42	-4.42	-3.44	-0.69	0.83	1.02	1.12	1.12	1.12	1.12	1.11
60	-4.32	-4.32	-3.42	-0.61	1.06	1.11	1.12	1.12	1.12	1.12	1.11
70	-4.21	-4.21	-3.31	-0.22	0.97	1.11	1.12	1.12	1.12	1.12	1.10
80	-4.23	-4.23	-3.35	-0.26	1.08	1.11	1.12	1.12	1.12	1.12	1.11
90	-4.31	-4.31	-4.04	-0.33	1.08	1.12	1.12	1.12	1.12	1.12	1.11
100	-4.41	-4.41	-4.13	-0.36	1.10	1.13	1.13	1.13	1.13	1.13	1.12
110	-4.43	-4.43	-3.76	-0.28	1.10	1.13	1.13	1.13	1.13	1.13	1.12
120	-4.46	-4.46	-3.44	-0.07	1.10	1.13	1.13	1.13	1.13	1.13	1.12
130	-4.52	-4.52	-4.08	0.13	1.08	1.13	1.13	1.13	1.13	1.13	1.12
140	-4.48	-4.48	-4.12	-0.23	1.07	1.12	1.13	1.13	1.13	1.13	1.12
150	-4.54	-4.54	-4.09	-0.23	1.10	1.11	1.13	1.13	1.13	1.13	1.12
160	-4.63	-4.63	-4.09	-0.14	1.09	1.11	1.13	1.13	1.13	1.13	1.11
170	-4.54	-4.54	-4.20	0.00	1.08	1.11	1.13	1.13	1.13	1.13	1.11
180	-4.51	-4.51	-4.20	0.04	1.09	1.12	1.13	1.13	1.13	1.13	1.11
190	-4.53	-4.53	-4.04	0.00	1.08	1.12	1.13	1.13	1.13	1.13	1.12
200	-4.50	-4.50	-4.05	-0.26	1.09	1.12	1.13	1.13	1.13	1.13	1.11
210	-4.41	-4.41	-4.07	0.10	1.09	1.12	1.12	1.12	1.12	1.12	1.11
220	-4.36	-4.36	-4.08	-0.38	1.11	1.12	1.12	1.12	1.12	1.12	1.11
230	-4.36	-4.36	-4.05	-0.35	1.11	1.12	1.12	1.12	1.12	1.12	1.11
240	-4.37	-4.37	-4.05	0.67	1.10	1.12	1.13	1.13	1.13	1.13	1.12
250	-4.41	-4.41	-4.06	0.55	1.10	1.12	1.13	1.13	1.13	1.13	1.12
260	-4.43	-4.43	-3.96	0.39	1.10	1.12	1.13	1.13	1.13	1.13	1.12
270	-4.40	-4.40	-3.95	0.41	1.09	1.11	1.13	1.13	1.13	1.13	1.12
280	-4.45	-4.45	-3.96	0.69	1.09	1.12	1.13	1.13	1.13	1.13	1.12
290	-4.48	-4.48	-3.94	0.45	1.07	1.12	1.13	1.13	1.13	1.13	1.12
300	-4.51	-4.51	-3.96	0.69	1.08	1.12	1.13	1.13	1.13	1.13	1.12
310	-4.49	-4.49	-3.94	0.65	1.09	1.12	1.12	1.12	1.12	1.12	1.12
320	-4.52	-4.52	-3.95	0.41	1.08	1.12	1.12	1.12	1.12	1.12	1.12
330	-4.55	-4.55	-3.63	0.41	1.09	1.13	1.13	1.13	1.13	1.13	1.11
340	-4.59	-4.59	-3.61	0.88	1.10	1.13	1.13	1.13	1.13	1.13	1.11
350	-4.58	-4.58	-3.61	0.83	1.10	1.13	1.13	1.13	1.13	1.13	1.11
360	-4.57	-4.57	-3.60	0.82	1.10	1.13	1.13	1.13	1.13	1.13	1.11
370	-4.57	-4.57	-0.66	0.75	1.12	1.13	1.13	1.13	1.13	1.13	1.11
380	-4.56	-4.56	-0.66	0.90	1.12	1.13	1.13	1.13	1.13	1.13	1.12
390	-4.54	-4.54	-0.67	0.83	1.12	1.13	1.13	1.13	1.13	1.13	1.11
400	-4.54	-4.54	-0.75	0.82	1.10	1.13	1.13	1.13	1.13	1.13	1.11
410	-4.57	-4.57	-0.73	0.94	1.11	1.13	1.13	1.13	1.13	1.13	1.12
420	-4.57	-4.57	-0.72	0.58	1.11	1.12	1.12	1.12	1.12	1.12	1.11
430	-4.58	-4.58	-0.64	0.83	1.11	1.12	1.13	1.13	1.13	1.13	1.11
440	-4.60	-4.60	-0.66	0.93	1.12	1.12	1.12	1.12	1.12	1.12	1.11
450	-4.60	-4.60	-0.66	0.57	1.12	1.12	1.12	1.12	1.12	1.12	1.11
460	-4.62	-4.62	-0.69	0.69	1.12	1.12	1.12	1.12	1.12	1.12	1.11
470	-4.61	-4.61	-0.66	0.61	1.12	1.12	1.12	1.12	1.12	1.12	1.11
480	-4.63	-4.63	-0.66	0.77	1.12	1.13	1.13	1.13	1.13	1.13	1.11
490	-4.61	-4.61	-0.64	0.96	1.12	1.12	1.12	1.12	1.12	1.12	1.11
500	-4.63	-4.63	-0.64	0.71	1.11	1.12	1.12	1.12	1.12	1.12	1.11

Table B.4: Solubility Dataset: Optimal objective found by [32]'s depth approximations

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	-3.05	-1.05	-0.33	0.41	0.75	0.75	0.75	0.99	1.12	1.16	1.16
20	-0.56	-0.07	-0.03	0.01	0.08	0.54	0.74	0.91	1.12	1.15	1.14
30	-0.86	-0.01	0.36	0.32	0.40	0.73	0.87	0.92	1.10	1.12	1.12
40	-0.44	0.20	0.49	0.56	0.63	0.96	NA	1.06	1.12	1.12	1.11
50	-0.48	-0.01	0.40	0.58	0.96	0.96	0.97	1.01	1.06	1.12	1.11
60	-0.60	-0.10	0.22	0.57	0.58	0.77	0.80	0.90	NA	1.12	1.11
70	-0.68	-0.10	0.13	NA	0.68	0.68	0.83	NA	NA	1.12	1.10
80	-0.52	0.00	NA	0.69	1.00	1.00	1.00	1.05	1.09	1.12	1.11
90	-0.73	NA	0.38	0.60	NA	NA	NA	NA	NA	1.12	1.11
100	NA	0.20	0.47	0.62	NA	NA	NA	NA	NA	1.13	1.12
110	-0.89	-0.06	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
120	0.09	0.60	0.72	0.86	NA	NA	NA	NA	NA	1.13	1.12
130	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
140	-0.13	0.36	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
150	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
160	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
170	0.41	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
180	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
190	-0.34	0.47	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
200	NA	0.60	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
210	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
220	0.19	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
230	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
240	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
250	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
260	-0.07	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
270	0.26	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
280	0.23	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
290	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
300	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
310	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.12
320	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.12
330	0.39	0.78	0.91	NA	NA	NA	NA	NA	NA	1.13	1.11
340	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
350	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
360	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
370	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
380	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
390	NA	0.86	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
400	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
410	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.12
420	0.37	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
430	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
440	0.70	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
450	0.56	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
460	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
470	0.70	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
480	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.13	1.11
490	0.64	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11
500	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.12	1.11

Table B.5: Solubility Dataset: Optimal objective found by [33]'s sample approximations

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	-	51.35	51.35	33.89	65.95	55.53	70.38	75.48	78.62	78.62	78.62
20	-	55.94	55.94	62.07	65.94	63.57	77.56	77.56	78.49	78.49	77.01
30	-	55.30	55.30	62.94	67.19	73.15	77.76	77.76	78.39	78.39	76.75
40	-	57.41	57.41	63.60	69.63	76.59	78.32	78.32	78.32	78.32	76.91
50	-	56.37	56.37	63.42	63.69	76.08	78.36	78.36	78.36	78.36	77.12
60	-	55.38	55.38	63.60	73.86	76.71	78.23	78.23	78.23	78.23	76.92
70	-	54.19	54.19	66.16	72.03	76.48	78.30	78.30	78.30	78.30	76.93
80	-	54.11	54.11	62.02	73.71	78.14	78.29	78.29	78.29	78.29	76.64
90	-	54.56	54.56	61.98	73.94	75.86	78.27	78.27	78.27	78.27	76.75
100	-	53.97	53.97	64.23	77.81	78.25	78.25	78.25	78.25	78.25	76.83
110	-	54.42	54.42	64.28	77.00	78.28	78.28	78.28	78.28	78.28	76.80
120	-	54.39	54.39	63.86	76.84	78.11	78.23	78.23	78.23	78.23	76.73
130	-	54.70	54.70	63.58	75.52	78.11	78.21	78.21	78.21	78.21	76.69
140	-	54.08	54.08	63.14	75.81	78.14	78.14	78.14	78.14	78.14	76.54
150	53.84	53.84	48.36	61.54	78.11	78.11	78.11	78.11	78.11	78.11	76.61
160	53.78	53.78	48.43	63.18	78.03	78.18	78.18	78.18	78.18	78.18	76.65
170	53.71	53.71	48.57	65.55	78.10	78.18	78.18	78.18	78.18	78.18	76.68
180	53.97	53.97	48.53	65.74	78.01	78.19	78.19	78.19	78.19	78.19	76.71
190	54.06	54.06	48.52	65.98	78.10	78.21	78.21	78.21	78.21	78.21	76.79
200	53.94	53.94	48.52	62.79	78.09	78.21	78.21	78.21	78.21	78.21	76.78
210	53.88	53.88	48.70	62.92	78.09	78.21	78.21	78.21	78.21	78.21	76.83
220	53.77	53.77	48.72	62.51	78.12	78.23	78.23	78.23	78.23	78.23	76.89
230	53.85	53.85	48.91	65.13	78.18	78.24	78.24	78.24	78.24	78.24	76.86
240	53.93	53.93	48.67	65.03	78.15	78.25	78.25	78.25	78.25	78.25	76.90
250	53.96	53.96	48.53	62.93	78.16	78.22	78.22	78.22	78.22	78.22	76.77
260	54.21	54.21	48.48	62.66	78.13	78.23	78.23	78.23	78.23	78.23	76.82
270	54.28	54.28	48.50	62.38	78.17	78.21	78.21	78.21	78.21	78.21	76.76
280	54.38	54.38	48.41	63.05	78.18	78.21	78.21	78.21	78.21	78.21	76.76
290	54.54	54.54	42.41	61.56	78.22	78.22	78.22	78.22	78.22	78.22	76.72
300	54.39	54.39	42.27	62.48	78.20	78.20	78.20	78.20	78.20	78.20	76.72
310	54.46	54.46	42.37	61.20	78.18	78.21	78.21	78.21	78.21	78.21	76.70
320	54.47	54.47	42.43	61.54	78.19	78.22	78.22	78.22	78.22	78.22	76.74
330	54.40	54.40	42.45	62.57	78.21	78.24	78.24	78.24	78.24	78.24	76.76
340	54.81	54.81	61.48	62.33	78.21	78.24	78.24	78.24	78.24	78.24	76.77
350	54.60	54.60	61.38	62.26	78.24	78.24	78.24	78.24	78.24	78.24	76.73
360	54.75	54.75	61.38	76.42	78.15	78.26	78.26	78.26	78.26	78.26	76.78
370	54.48	54.48	49.24	75.18	78.20	78.26	78.26	78.26	78.26	78.26	76.81
380	54.32	54.32	53.78	75.20	78.20	78.26	78.26	78.26	78.26	78.26	76.80
390	54.16	54.16	51.97	63.88	78.13	78.25	78.25	78.25	78.25	78.25	76.79
400	54.16	54.16	50.50	63.82	78.15	78.26	78.26	78.26	78.26	78.26	76.72
410	54.19	54.19	50.49	63.87	78.08	78.27	78.27	78.27	78.27	78.27	76.75
420	54.17	54.17	51.80	75.05	78.08	78.26	78.26	78.26	78.26	78.26	76.76
430	54.10	54.10	51.04	62.70	78.13	78.24	78.24	78.24	78.24	78.24	76.76
440	54.24	54.24	51.93	64.80	78.14	78.24	78.24	78.24	78.24	78.24	76.78
450	54.28	54.28	51.03	63.89	78.15	78.25	78.25	78.25	78.25	78.25	76.80
460	54.43	54.43	51.11	74.52	78.16	78.26	78.26	78.26	78.26	78.26	76.83
470	54.39	54.39	55.25	74.53	78.17	78.27	78.27	78.27	78.27	78.27	76.83
480	54.45	54.45	51.82	75.41	78.17	78.26	78.26	78.26	78.26	78.26	76.83
490	54.33	54.33	54.70	75.38	78.17	78.26	78.26	78.26	78.26	78.26	76.83
500	54.34	54.34	54.68	76.22	78.16	78.26	78.26	78.26	78.26	78.26	76.83

Table B.6: Concrete Dataset: Optimal objective found by [32]'s depth approximations

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	73.73	73.73	73.70	75.48	75.48	75.48	78.62	78.62	78.62	78.62	78.62
20	71.49	71.53	73.62	75.71	75.71	75.71	78.49	78.49	78.49	78.49	77.01
30	62.40	68.55	75.03	75.07	74.81	77.74	77.74	77.74	78.39	78.39	76.75
40	72.16	72.16	75.00	77.39	77.88	77.53	77.83	77.83	78.32	78.32	76.91
50	70.17	74.50	74.75	75.62	77.44	78.36	78.36	78.36	78.36	78.36	77.12
60	72.56	75.32	76.27	75.95	75.95	75.95	77.51	78.23	78.23	78.23	76.92
70	65.21	73.00	76.97	78.02	78.02	78.02	78.30	78.30	78.30	78.30	76.93
80	74.32	75.76	77.06	76.59	76.90	77.82	77.82	78.29	78.29	78.29	76.64
90	73.48	76.37	78.00	78.00	77.47	78.00	78.00	78.00	78.00	78.27	76.75
100	71.86	75.14	77.27	77.27	78.01	77.64	77.88	78.25	78.25	78.25	76.83
110	73.88	77.78	78.28	78.28	78.28	78.28	78.28	78.28	78.28	78.28	76.80
120	74.81	76.00	77.13	77.33	78.11	78.23	78.23	78.23	78.23	78.23	76.73
130	77.92	77.92	77.92	77.92	77.92	77.92	78.11	78.21	78.21	78.21	76.69
140	67.59	75.88	77.87	77.87	78.04	78.04	78.04	78.14	78.14	78.14	76.54
150	76.01	77.27	77.95	77.95	77.95	78.11	78.11	78.11	78.11	78.11	76.61
160	75.63	77.94	77.94	77.94	77.94	77.94	78.03	78.03	78.18	78.18	76.65
170	77.13	77.96	78.04	78.04	78.04	78.18	78.18	NA	NA	78.18	76.68
180	74.28	78.11	78.19	78.19	78.19	78.19	78.19	78.19	NA	78.19	76.71
190	77.81	78.09	78.09	78.09	78.09	78.09	78.09	78.09	NA	78.21	76.79
200	71.93	78.08	78.21	78.21	78.21	78.21	78.21	78.21	78.21	78.21	76.78
210	77.89	77.67	78.09	NA	NA	78.09	NA	NA	NA	78.21	76.83
220	77.33	78.06	78.06	78.17	78.17	78.17	78.23	78.23	78.23	78.23	76.89
230	77.44	78.07	78.07	78.07	78.07	78.18	78.24	78.24	78.24	78.24	76.86
240	76.74	78.09	78.09	78.09	78.19	78.19	78.25	78.25	78.25	78.25	76.90
250	76.80	77.68	77.96	78.16	78.16	78.16	NA	NA	NA	78.22	76.77
260	77.24	77.49	78.08	77.70	NA	NA	NA	NA	NA	78.23	76.82
270	77.57	78.04	NA	78.04	78.04	78.04	NA	NA	NA	78.21	76.76
280	75.99	76.72	77.97	77.97	NA	NA	78.21	NA	NA	78.21	76.76
290	77.04	78.14	78.14	78.14	NA	NA	78.18	NA	NA	78.22	76.72
300	76.82	78.12	78.12	78.12	78.12	NA	78.12	78.20	78.20	78.20	76.72
310	77.43	77.50	78.06	78.06	78.21	78.21	78.21	NA	NA	78.21	76.70
320	77.48	78.05	78.19	78.19	78.19	NA	NA	NA	NA	78.22	76.74
330	75.62	78.07	78.21	NA	NA	NA	NA	NA	NA	78.24	76.76
340	76.42	77.00	NA	NA	NA	78.21	NA	NA	NA	78.24	76.77
350	77.23	77.29	78.15	78.15	NA	78.24	NA	NA	NA	78.24	76.73
360	77.92	NA	77.92	78.10	NA	NA	NA	NA	NA	78.26	76.78
370	76.91	78.20	78.20	78.20	NA	NA	NA	NA	NA	78.26	76.81
380	77.11	78.05	78.05	78.05	NA	NA	NA	NA	NA	78.26	76.80
390	77.63	77.63	78.03	78.19	NA	NA	NA	NA	78.25	78.25	76.79
400	77.40	77.84	78.00	78.00	NA	NA	NA	NA	NA	78.26	76.72
410	77.43	78.01	78.01	78.17	78.27	NA	NA	NA	NA	78.27	76.75
420	76.66	78.05	78.10	NA	NA	NA	NA	NA	NA	78.26	76.76
430	76.90	78.04	78.04	78.09	78.24	78.24	NA	NA	NA	78.24	76.76
440	77.07	78.19	78.24	78.24	NA	NA	NA	NA	NA	78.24	76.78
450	77.50	78.20	78.20	NA	NA	NA	NA	NA	NA	78.25	76.80
460	77.74	77.79	78.03	78.09	NA	NA	NA	NA	NA	78.26	76.83
470	77.98	77.98	78.27	78.27	NA	NA	NA	NA	NA	78.27	76.83
480	77.85	NA	78.22	78.22	NA	NA	NA	NA	NA	78.26	76.83
490	78.22	NA	78.22	NA	NA	NA	NA	78.26	NA	78.26	76.83
500	77.37	77.98	78.13	78.21	78.21	78.21	NA	NA	NA	78.26	76.83

Table B.7: Concrete Dataset: Optimal objective found by [33]'s sample approximations

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	6.02	6.02	6.03	5.99	5.85	7.21	7.38	7.43	7.43	7.43	6.69
20	5.87	5.87	5.99	6.21	6.74	7.30	7.39	7.45	7.45	7.45	6.60
30	5.64	5.64	5.25	6.39	6.80	7.35	7.38	7.42	7.42	7.42	6.74
40	5.65	5.65	5.43	5.94	7.02	7.31	7.42	7.42	7.42	7.42	6.77
50	5.65	5.65	5.47	6.19	6.96	7.34	7.41	7.41	7.41	7.41	6.76
60	5.57	5.57	5.56	6.55	7.05	7.34	7.35	7.38	7.38	7.38	6.75
70	5.56	5.56	5.60	6.54	7.18	7.29	7.36	7.36	7.37	7.37	6.65
80	5.59	5.59	5.64	6.50	6.97	7.28	7.37	7.37	7.38	7.38	6.63
90	5.54	5.54	5.63	6.40	7.23	7.29	7.34	7.36	7.36	7.36	6.72
100	5.51	5.51	5.54	5.74	7.23	7.35	7.35	7.37	7.37	7.37	6.69
110	5.51	5.51	5.54	6.46	7.02	7.32	7.36	7.37	7.37	7.37	6.67
120	5.52	5.52	5.92	6.48	6.86	7.33	7.36	7.36	7.36	7.36	6.68
130	5.50	5.50	5.95	6.85	7.21	7.36	7.36	7.36	7.36	7.36	6.71
140	5.51	5.51	6.57	6.64	7.30	7.36	7.36	7.36	7.36	7.36	6.69
150	5.50	5.50	6.13	6.95	7.19	7.36	7.36	7.36	7.36	7.36	6.69
160	5.52	5.52	6.53	7.00	7.33	7.37	7.37	7.37	7.37	7.37	6.70
170	5.49	5.49	6.29	6.79	7.29	7.37	7.37	7.37	7.37	7.37	6.69
180	5.50	5.50	6.30	6.98	7.29	7.37	7.37	7.37	7.37	7.37	6.70
190	5.49	5.49	6.32	6.88	7.26	7.37	7.37	7.37	7.37	7.37	6.71
200	5.52	5.52	5.84	6.70	7.30	7.37	7.37	7.37	7.37	7.37	6.71
210	5.52	5.52	6.22	6.93	7.34	7.37	7.37	7.38	7.38	7.38	6.72
220	5.52	5.52	6.25	7.08	7.32	7.38	7.38	7.38	7.38	7.38	6.70
230	5.53	5.53	6.41	7.07	7.35	7.38	7.37	7.38	7.38	7.38	6.72
240	5.52	5.52	6.53	6.78	7.34	7.37	7.37	7.37	7.37	7.37	6.77
250	5.53	5.53	6.40	7.17	7.32	7.38	7.37	7.38	7.38	7.38	6.76
260	5.54	5.54	6.28	7.16	7.35	7.37	7.37	7.37	7.37	7.37	6.76
270	5.55	5.55	6.17	7.18	7.35	7.37	7.37	7.37	7.37	7.37	6.77
280	5.56	5.56	6.21	7.10	7.35	7.37	7.37	7.37	7.37	7.37	6.76
290	5.56	5.56	6.39	7.20	7.36	7.37	7.37	7.37	7.37	7.37	6.77
300	5.54	5.54	6.39	7.17	7.36	7.35	7.37	7.37	7.37	7.37	6.78
310	5.55	5.55	6.49	7.17	7.36	7.37	7.37	7.37	7.37	7.37	6.77
320	5.55	5.55	5.80	7.21	7.28	7.37	7.37	7.37	7.37	7.37	6.78
330	5.55	5.55	6.17	7.23	7.30	7.37	7.37	7.37	7.37	7.37	6.76
340	5.55	5.55	6.02	7.21	7.32	7.37	7.37	7.37	7.37	7.37	6.76
350	5.55	5.55	6.23	7.21	7.36	7.36	7.36	7.36	7.36	7.36	6.75
360	5.55	5.55	6.22	7.20	7.35	7.35	7.36	7.36	7.36	7.36	6.75
370	5.54	5.54	5.95	7.20	7.34	7.36	7.36	7.36	7.36	7.36	6.75
380	5.55	5.55	6.10	7.20	7.35	7.36	7.36	7.36	7.36	7.36	6.76
390	5.54	5.54	6.34	7.11	7.32	7.36	7.36	7.36	7.36	7.36	6.75
400	5.54	5.54	6.39	7.10	7.36	7.36	7.36	7.36	7.36	7.36	6.75
410	5.54	5.54	6.38	7.05	7.35	7.36	7.36	7.36	7.36	7.36	6.76
420	5.53	5.53	5.98	6.82	7.34	7.36	7.36	7.36	7.36	7.36	6.76
430	5.54	5.54	6.19	7.25	7.34	7.34	7.36	7.36	7.36	7.36	6.76
440	5.53	5.53	6.16	7.24	7.34	7.34	7.36	7.36	7.36	7.36	6.77
450	5.53	5.53	5.55	7.19	7.34	7.36	7.36	7.36	7.36	7.36	6.77
460	5.53	5.53	6.16	7.19	7.31	7.35	7.36	7.36	7.36	7.36	6.77
470	5.53	5.53	6.02	7.25	7.35	7.36	7.36	7.36	7.36	7.36	6.83
480	5.53	5.53	6.06	7.24	7.35	7.36	7.36	7.36	7.36	7.36	6.84
490	5.53	5.53	6.25	7.26	7.34	7.36	7.36	7.36	7.36	7.36	6.80
500	5.53	5.53	6.05	7.26	7.33	7.35	7.36	7.36	7.36	7.36	6.81

Table B.8: Wine Dataset: Optimal objective found by [32]'s depth approximations

# trees	Approximation Level									S^*_{MIO}	S^*_{UMOTEM}
	10%	20%	30%	40%	50%	60%	70%	80%	90%		
10	6.16	6.50	7.17	7.39	7.26	7.40	7.40	7.40	7.40	7.43	6.69
20	6.56	7.01	7.04	7.29	7.29	7.29	7.45	7.45	7.45	7.45	6.60
30	6.81	7.09	7.09	7.37	7.37	7.37	7.38	7.41	NA	7.42	6.74
40	6.99	6.96	7.03	7.21	7.29	7.37	7.37	7.39	7.40	7.42	6.77
50	6.93	7.20	NA	7.30	7.36	7.34	NA	NA	NA	7.41	6.76
60	7.06	7.12	7.21	7.28	7.29	7.35	7.35	NA	NA	7.38	6.75
70	7.12	7.23	7.19	NA	7.25	7.30	7.33	NA	NA	7.37	6.65
80	7.08	7.23	7.27	7.28	7.31	7.32	7.36	7.36	NA	7.38	6.63
90	7.00	7.14	7.27	7.28	NA	NA	NA	NA	NA	7.36	6.72
100	NA	7.17	7.19	NA	7.27	7.34	NA	NA	NA	7.37	6.69
110	6.96	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.67
120	6.91	7.25	7.30	7.32	7.33	7.34	NA	NA	NA	7.36	6.68
130	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.71
140	7.10	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.69
150	7.05	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.69
160	NA	7.29	NA	NA	NA	NA	NA	NA	NA	7.37	6.70
170	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.69
180	7.19	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.70
190	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.71
200	7.14	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.71
210	7.09	NA	NA	NA	NA	NA	NA	NA	NA	7.38	6.72
220	7.19	NA	NA	NA	NA	NA	NA	NA	NA	7.38	6.70
230	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.38	6.72
240	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.77
250	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.38	6.76
260	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.76
270	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.77
280	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.76
290	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.77
300	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.78
310	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.77
320	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.78
330	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.76
340	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.37	6.76
350	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.75
360	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.75
370	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.75
380	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.76
390	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.75
400	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.75
410	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.76
420	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.76
430	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.76
440	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.77
450	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.77
460	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.77
470	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.83
480	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.84
490	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.80
500	NA	NA	NA	NA	NA	NA	NA	NA	NA	7.36	6.81

Table B.9: Wine Dataset: Optimal objective found by [33]'s sample approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	8.26	7.18	8.99	7.83	8.49	8.72	8.79	8.92	9.01
20	7.35	7.46	8.26	8.33	11.22	9.51	9.83	10.11	10.21
30	7.63	7.74	8.62	9.09	12.21	11.01	12.54	12.41	12.23
40	7.80	9.00	9.07	9.57	11.81	13.31	17.56	17.61	18.92
50	10.08	9.99	11.37	12.00	16.85	18.90	13.54	19.09	22.22
60	8.87	10.49	11.46	12.99	16.61	19.48	15.81	19.57	17.64
70	8.77	10.51	10.28	13.86	17.74	24.36	19.38	22.86	18.49
80	9.30	11.03	10.37	12.40	20.82	20.04	20.93	21.37	18.45
90	11.19	9.13	11.86	14.80	20.39	26.00	22.78	23.67	22.27
100	9.07	11.14	12.77	13.18	23.31	27.29	27.06	23.20	25.07
110	9.62	11.41	10.89	16.09	21.41	23.09	23.73	26.45	22.38
120	11.59	9.70	11.48	23.45	24.44	25.60	25.47	24.83	24.57
130	9.64	10.04	11.71	22.85	27.23	26.31	39.99	27.91	29.54
140	10.05	10.35	11.96	18.78	23.81	32.47	32.37	31.23	31.71
150	10.50	10.51	12.66	17.04	27.55	34.45	30.38	43.17	43.03
160	10.40	10.94	12.99	17.94	27.42	30.08	34.00	34.28	32.21
170	10.53	12.19	13.95	19.33	31.76	33.88	39.85	38.89	42.15
180	11.64	12.61	14.28	18.06	33.64	33.90	38.84	38.60	42.53
190	11.65	11.68	13.78	21.10	32.16	34.24	48.23	44.50	44.78
200	11.54	11.87	14.11	24.34	36.15	43.87	40.69	46.30	50.71
210	11.72	12.00	13.94	21.91	32.98	43.60	42.73	43.65	52.35
220	12.19	11.98	14.67	21.57	39.83	46.90	41.12	49.31	51.39
230	12.03	12.73	15.86	20.57	40.37	45.09	54.88	53.86	53.99
240	12.37	12.62	14.64	36.72	43.23	41.18	43.38	42.80	49.34
250	12.06	12.29	15.21	41.51	40.13	47.55	45.59	49.17	49.33
260	12.42	12.79	15.07	39.70	50.62	45.74	54.08	54.91	61.25
270	12.38	12.75	15.79	40.72	44.43	43.91	49.51	50.08	50.59
280	12.31	12.70	16.45	46.51	42.42	50.08	51.88	54.99	57.26
290	12.83	12.95	17.08	57.54	47.98	45.61	49.64	57.18	53.30
300	12.92	13.20	16.92	44.95	44.42	45.20	64.61	67.33	71.06
310	13.00	13.21	17.31	44.03	52.99	57.07	53.90	55.43	58.15
320	13.18	14.25	18.44	44.37	52.07	65.58	53.40	55.64	58.59
330	14.61	13.78	18.61	53.47	50.48	61.46	58.42	72.71	73.08
340	14.01	15.24	18.74	32.83	45.56	64.23	69.42	63.27	61.66
350	14.59	14.58	19.46	55.35	49.72	64.42	75.93	65.21	69.12
360	14.66	15.02	19.23	47.65	43.43	50.55	63.95	75.56	61.29
370	15.58	14.44	22.33	47.63	51.15	65.62	77.89	76.68	82.06
380	14.77	15.17	23.25	38.07	55.52	65.64	62.12	66.98	77.63
390	15.15	16.61	22.97	51.44	62.93	71.36	65.43	75.22	82.16
400	14.82	15.79	27.24	66.51	60.44	65.56	64.69	68.81	81.27
410	15.10	15.55	27.23	58.02	60.04	65.58	63.78	68.57	76.29
420	16.89	15.67	27.43	90.59	59.58	68.67	71.95	68.77	77.13
430	15.73	16.01	24.62	56.51	66.82	74.09	84.21	81.18	86.87
440	15.79	15.96	27.22	70.87	68.87	66.62	77.91	81.28	91.12
450	16.20	15.70	27.22	85.77	66.23	82.14	80.92	85.82	91.77
460	17.25	16.24	25.89	68.39	71.14	75.82	84.99	90.42	99.12
470	16.25	16.97	25.87	67.02	68.38	73.53	85.07	91.06	92.99
480	16.80	16.56	25.46	50.17	77.88	88.44	79.00	85.18	89.73
490	17.26	16.96	25.25	61.13	77.32	77.87	80.73	93.46	95.70
500	17.75	17.65	26.53	87.48	72.85	81.31	87.43	103.66	112.63

Table B.10: Solubility Dataset: Runtimes of [32]'s depth approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	9.67	11.15	24.80	28.60	48.48	54.42	93.93	104.58	135.03
20	11.83	36.71	78.29	129.21	156.59	226.79	250.10	246.63	255.36
30	13.37	73.11	99.52	210.86	236.17	246.60	532.12	518.48	825.85
40	50.50	78.12	217.79	278.45	367.86	375.55	NA	814.44	1027.44
50	16.17	171.43	252.70	307.00	324.22	680.97	1209.57	1055.21	1334.71
60	16.65	102.78	175.06	212.08	235.49	1162.65	1281.80	1833.20	NA
70	67.40	111.99	485.61	NA	1234.45	1016.78	3750.17	NA	NA
80	30.58	333.40	NA	903.64	1018.15	1525.14	1860.83	2398.59	2922.37
90	217.02	NA	725.11	829.09	NA	NA	NA	NA	NA
100	NA	1223.45	2367.50	2886.98	NA	NA	NA	NA	NA
110	81.69	453.46	NA	NA	NA	NA	NA	NA	NA
120	306.31	625.72	1159.73	1673.34	NA	NA	NA	NA	NA
130	NA	NA	NA	NA	NA	NA	NA	NA	NA
140	61.23	629.63	NA	NA	NA	NA	NA	NA	NA
150	NA	NA	NA	NA	NA	NA	NA	NA	NA
160	NA	NA	NA	NA	NA	NA	NA	NA	NA
170	37.93	NA	NA	NA	NA	NA	NA	NA	NA
180	NA	NA	NA	NA	NA	NA	NA	NA	NA
190	133.44	544.48	NA	NA	NA	NA	NA	NA	NA
200	NA	808.31	NA	NA	NA	NA	NA	NA	NA
210	NA	NA	NA	NA	NA	NA	NA	NA	NA
220	104.92	NA	NA	NA	NA	NA	NA	NA	NA
230	NA	NA	NA	NA	NA	NA	NA	NA	NA
240	NA	NA	NA	NA	NA	NA	NA	NA	NA
250	NA	NA	NA	NA	NA	NA	NA	NA	NA
260	312.45	NA	NA	NA	NA	NA	NA	NA	NA
270	341.87	NA	NA	NA	NA	NA	NA	NA	NA
280	956.86	NA	NA	NA	NA	NA	NA	NA	NA
290	NA	NA	NA	NA	NA	NA	NA	NA	NA
300	NA	NA	NA	NA	NA	NA	NA	NA	NA
310	NA	NA	NA	NA	NA	NA	NA	NA	NA
320	NA	NA	NA	NA	NA	NA	NA	NA	NA
330	702.80	1773.14	2138.75	NA	NA	NA	NA	NA	NA
340	NA	NA	NA	NA	NA	NA	NA	NA	NA
350	NA	NA	NA	NA	NA	NA	NA	NA	NA
360	NA	NA	NA	NA	NA	NA	NA	NA	NA
370	NA	NA	NA	NA	NA	NA	NA	NA	NA
380	NA	NA	NA	NA	NA	NA	NA	NA	NA
390	NA	5974.07	NA	NA	NA	NA	NA	NA	NA
400	NA	NA	NA	NA	NA	NA	NA	NA	NA
410	NA	NA	NA	NA	NA	NA	NA	NA	NA
420	2594.10	NA	NA	NA	NA	NA	NA	NA	NA
430	NA	NA	NA	NA	NA	NA	NA	NA	NA
440	1600.20	NA	NA	NA	NA	NA	NA	NA	NA
450	3422.48	NA	NA	NA	NA	NA	NA	NA	NA
460	NA	NA	NA	NA	NA	NA	NA	NA	NA
470	740.43	NA	NA	NA	NA	NA	NA	NA	NA
480	NA	NA	NA	NA	NA	NA	NA	NA	NA
490	713.00	NA	NA	NA	NA	NA	NA	NA	NA
500	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table B.11: Solubility Dataset: Runtimes of [33]'s sampling approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	-	7.29	7.02	7.60	7.42	7.94	7.99	8.14	8.56
20	-	7.32	6.97	7.70	7.98	7.90	10.11	10.29	11.44
30	-	7.32	7.02	7.72	8.16	8.26	9.21	9.49	9.24
40	-	7.09	7.36	7.89	7.97	9.00	10.22	10.21	10.57
50	-	7.07	7.54	7.98	8.41	10.19	10.61	10.79	11.44
60	-	7.43	7.57	7.99	9.84	10.72	11.37	11.62	11.87
70	-	7.44	7.32	8.55	9.87	10.83	13.73	12.80	12.90
80	-	7.24	7.31	8.21	10.01	11.42	13.09	12.80	13.16
90	-	7.40	7.53	8.37	11.07	11.20	13.04	13.68	14.04
100	-	7.53	7.58	8.58	10.89	11.88	14.01	14.63	14.90
110	-	7.66	7.42	8.98	11.39	11.87	14.23	15.58	15.97
120	-	7.43	7.55	8.55	12.41	13.50	15.38	18.10	16.65
130	-	7.59	7.62	8.73	12.97	13.82	15.83	17.08	17.50
140	-	7.46	7.68	8.93	13.86	14.92	18.03	18.47	18.12
150	7.37	7.52	8.67	13.44	16.13	18.80	21.16	20.40	21.46
160	7.65	7.60	9.22	15.05	15.98	19.56	21.73	21.54	21.69
170	7.38	7.91	9.17	12.64	17.15	21.69	20.70	21.27	20.79
180	7.55	7.56	9.32	14.77	17.17	20.13	21.48	22.49	22.05
190	7.35	7.70	9.39	18.86	17.84	21.57	24.09	23.03	22.93
200	7.36	7.93	9.57	15.54	19.06	25.62	23.88	26.26	27.02
210	7.62	7.84	9.58	15.42	20.81	22.77	24.68	24.62	25.32
220	7.52	7.74	9.74	14.15	18.64	27.41	27.01	26.14	26.10
230	7.59	7.79	10.81	18.02	20.58	31.06	29.40	26.86	29.61
240	7.57	8.92	9.94	18.34	24.72	27.88	31.72	30.64	28.84
250	8.61	7.95	10.85	17.03	21.87	26.71	30.77	29.39	29.47
260	7.73	8.43	11.33	15.96	23.62	29.25	32.57	33.38	35.78
270	8.26	8.29	10.32	21.95	22.43	27.84	30.61	30.97	31.38
280	7.73	8.14	11.02	19.61	25.24	28.61	31.84	34.90	35.23
290	7.69	8.21	10.63	17.90	25.53	29.73	35.49	33.15	33.51
300	7.82	8.13	11.58	18.16	23.60	32.32	36.07	39.13	41.65
310	7.81	9.05	10.88	24.94	27.83	31.61	36.98	35.13	36.74
320	7.75	8.70	10.98	27.05	27.85	37.72	35.33	35.48	35.95
330	7.85	8.19	11.05	21.21	25.09	33.39	39.03	36.70	37.74
340	8.84	8.20	11.72	23.75	28.72	41.75	41.48	39.38	43.21
350	7.86	9.49	11.79	23.08	31.85	36.94	38.25	46.89	44.54
360	9.14	8.48	10.48	21.53	27.60	36.12	47.16	45.42	45.43
370	8.12	8.30	14.20	22.98	27.78	44.29	47.12	46.24	49.69
380	7.93	9.53	12.64	20.39	30.99	45.42	40.96	50.33	51.22
390	9.27	9.70	11.78	28.02	30.14	43.51	52.49	48.88	54.00
400	9.42	9.16	13.19	24.77	32.83	44.92	52.77	49.55	54.28
410	9.49	8.40	13.37	25.59	31.39	40.48	48.90	50.85	54.21
420	9.41	8.44	11.83	31.61	31.09	46.38	50.96	52.13	52.50
430	8.22	8.93	12.03	35.12	31.76	42.83	51.70	53.60	53.44
440	8.13	8.65	12.12	31.37	37.40	47.77	52.43	59.80	55.68
450	10.01	8.56	12.32	31.52	32.93	44.03	54.05	55.72	56.14
460	8.22	8.74	12.40	27.51	33.85	49.85	55.88	56.85	57.48
470	8.38	8.79	12.43	24.20	34.74	52.58	56.74	57.91	58.94
480	8.37	9.37	12.64	30.24	35.30	52.41	58.39	60.95	61.97
490	9.14	8.79	12.70	29.94	37.68	53.06	61.57	64.08	60.63
500	9.52	10.12	12.85	28.80	39.65	49.64	60.05	60.63	64.04

Table B.12: Concrete Dataset: Runtimes of [32]'s depth approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	8.73	9.26	13.33	12.47	13.78	14.50	11.57	15.32	15.82
20	11.27	9.85	11.38	17.33	17.09	19.11	13.65	17.38	17.85
30	9.95	14.59	13.85	17.49	24.42	22.55	26.71	24.72	27.83
40	13.44	13.32	16.83	24.99	21.71	23.02	24.57	27.63	44.67
50	10.87	18.31	19.55	23.04	24.34	26.31	45.24	39.21	48.41
60	11.36	16.12	19.59	40.74	135.18	110.60	118.87	128.19	105.39
70	22.30	17.66	28.00	36.71	51.90	52.97	72.25	140.85	186.74
80	19.52	28.65	42.23	112.94	47.93	92.89	155.91	139.26	97.63
90	16.93	20.06	25.30	64.33	79.10	118.13	195.59	101.65	266.98
100	18.14	27.15	35.36	47.52	116.25	177.16	130.24	178.46	275.86
110	19.35	27.22	55.97	43.08	56.13	58.67	100.82	98.79	300.99
120	23.50	49.76	70.12	69.71	173.02	86.05	76.51	257.33	330.32
130	18.14	38.10	78.64	72.60	60.68	170.01	166.30	316.08	449.87
140	16.62	28.57	118.89	94.45	203.92	196.19	484.79	382.72	529.71
150	20.75	80.91	67.59	127.02	819.29	1347.97	1306.40	1679.08	2328.01
160	29.27	48.22	89.49	166.07	289.01	287.91	304.07	568.99	559.94
170	19.95	295.89	726.91	852.73	1148.62	1519.99	1883.67	NA	NA
180	24.48	50.66	99.30	165.60	268.71	1902.40	1881.57	2490.90	NA
190	24.53	106.95	835.14	1044.39	1483.80	1791.81	2248.39	2876.57	NA
200	27.92	53.86	121.37	196.75	467.20	551.70	397.89	796.17	3189.27
210	198.82	441.27	1025.73	NA	NA	2080.16	NA	NA	NA
220	22.25	78.78	88.60	179.81	494.87	671.88	651.35	1138.89	1388.76
230	30.35	51.83	335.46	436.49	715.24	897.72	1220.37	1220.83	3761.16
240	33.17	212.54	283.33	537.31	743.35	1023.56	1018.66	1544.26	2078.70
250	30.46	114.09	203.79	364.76	309.49	640.69	NA	NA	NA
260	63.91	119.80	174.80	1188.12	NA	NA	NA	NA	NA
270	63.17	197.64	NA	1811.23	2008.74	2218.59	NA	NA	NA
280	41.91	169.41	254.46	469.84	NA	NA	3296.56	NA	NA
290	41.43	103.01	291.51	1718.85	NA	NA	3488.51	NA	NA
300	39.94	112.15	186.10	599.71	956.47	NA	1841.03	2631.08	4814.39
310	48.64	124.82	325.17	598.14	1014.85	1224.01	1154.91	NA	NA
320	73.24	263.72	315.20	786.04	1007.63	NA	NA	NA	NA
330	51.46	1021.28	1527.57	NA	NA	NA	NA	NA	NA
340	66.17	343.14	NA	NA	NA	3563.53	NA	NA	NA
350	153.84	349.15	428.30	928.57	NA	1520.00	NA	NA	NA
360	464.43	NA	1648.78	2246.87	NA	NA	NA	NA	NA
370	143.49	405.09	744.16	2328.35	NA	NA	NA	NA	NA
380	59.62	296.92	529.75	717.44	NA	NA	NA	NA	NA
390	510.95	1083.23	1372.90	2960.74	NA	NA	NA	NA	7055.02
400	95.51	231.87	589.37	2818.48	NA	NA	NA	NA	NA
410	48.94	351.90	366.05	986.29	1498.57	NA	NA	NA	NA
420	110.06	358.71	1016.11	NA	NA	NA	NA	NA	NA
430	136.36	414.48	1041.15	1071.61	1835.11	2222.55	NA	NA	NA
440	184.71	143.50	557.92	1006.94	NA	NA	NA	NA	NA
450	144.40	689.40	1165.70	NA	NA	NA	NA	NA	NA
460	75.51	349.55	647.03	1347.26	NA	NA	NA	NA	NA
470	130.04	488.45	591.98	1191.30	NA	NA	NA	NA	NA
480	162.73	NA	1086.74	1572.86	NA	NA	NA	NA	NA
490	706.72	NA	2048.95	NA	NA	NA	NA	6791.25	NA
500	250.88	1230.94	2090.88	3037.14	3956.66	5693.66	NA	NA	NA

Table B.13: Concrete Dataset: Runtimes of [33]'s sampling approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	7.46	7.81	9.24	8.33	8.93	8.63	9.93	21.13	28.31
20	7.10	8.89	7.75	8.28	9.42	18.34	20.86	17.05	29.64
30	8.17	8.41	8.67	8.21	15.96	34.84	43.48	55.26	48.99
40	7.55	8.93	9.88	11.04	19.54	36.74	59.44	52.18	53.13
50	8.77	8.42	8.83	10.03	18.39	45.31	66.07	49.15	61.97
60	9.21	8.34	9.09	10.65	33.40	43.35	80.20	90.60	79.89
70	8.33	9.05	9.47	10.12	29.86	85.01	128.28	81.05	93.06
80	8.20	9.01	9.23	12.50	42.33	114.52	122.50	111.48	108.06
90	8.81	8.98	9.50	11.67	32.15	71.29	158.20	135.51	162.28
100	8.84	8.26	9.88	11.20	47.65	84.32	177.97	104.09	136.22
110	8.93	8.59	10.39	10.61	47.37	87.08	125.47	187.03	153.24
120	8.63	7.89	8.50	11.60	36.60	129.00	127.43	152.27	199.19
130	8.71	8.24	10.08	18.88	99.51	155.50	176.60	205.97	178.34
140	8.96	8.57	10.02	20.88	66.42	235.07	210.07	195.79	223.91
150	8.29	9.27	9.98	14.57	102.13	159.40	139.32	228.00	174.45
160	8.63	9.62	10.94	22.67	110.94	146.15	181.94	247.60	264.02
170	8.59	9.69	10.04	22.45	86.62	148.18	173.62	233.42	231.00
180	8.62	8.77	10.59	17.77	116.93	157.68	178.74	260.05	252.50
190	8.86	8.83	11.56	18.82	82.96	187.35	293.42	286.54	278.43
200	8.69	9.16	12.99	22.96	127.20	177.11	234.36	247.73	239.96
210	8.13	9.07	13.56	18.18	83.79	253.28	255.83	327.44	259.50
220	8.86	8.44	13.41	21.46	91.14	190.93	309.75	274.79	285.00
230	8.74	8.33	11.01	19.21	102.50	164.80	201.90	263.76	272.24
240	8.10	8.29	11.45	26.34	104.09	263.59	332.13	309.62	337.84
250	8.91	8.45	11.06	22.56	133.54	322.62	349.14	378.64	384.65
260	9.61	9.41	11.31	22.67	112.10	294.94	446.12	386.10	445.51
270	8.44	9.79	13.08	24.97	120.01	310.61	385.65	424.22	366.16
280	8.51	8.70	12.79	24.92	151.70	288.70	385.33	405.60	444.43
290	9.08	8.71	11.65	28.17	146.41	331.33	390.85	415.99	466.28
300	8.78	9.19	11.99	29.20	131.36	327.91	541.82	439.60	428.74
310	8.91	8.88	13.44	34.21	128.09	427.07	595.81	537.88	520.80
320	8.97	9.74	17.30	27.93	234.20	506.85	474.60	406.83	616.62
330	9.31	8.93	18.06	35.98	174.44	384.53	475.07	462.11	420.01
340	9.45	9.01	17.35	32.51	231.79	496.46	668.32	390.23	448.84
350	9.40	9.00	16.48	33.33	225.25	574.38	485.06	717.97	498.00
360	8.79	10.01	16.09	37.81	183.64	388.63	484.59	449.99	491.94
370	8.97	9.88	15.67	38.33	216.92	430.68	604.53	531.45	562.86
380	9.61	9.90	13.49	33.46	229.88	560.49	551.72	835.65	795.88
390	9.53	9.86	15.52	42.01	257.21	461.89	870.14	749.19	926.18
400	9.60	9.95	16.03	39.82	376.47	470.90	830.38	864.26	825.96
410	9.65	9.99	17.62	46.04	238.70	673.08	925.45	944.72	950.24
420	9.03	9.14	22.60	46.42	284.13	732.23	950.95	960.92	911.77
430	9.97	10.25	15.67	49.87	231.48	864.95	940.84	1065.40	1062.24
440	9.78	10.71	17.94	46.83	294.87	994.43	1097.23	1065.68	1113.59
450	9.89	10.14	18.62	38.25	249.92	706.52	1011.14	978.42	1101.15
460	9.80	10.09	15.44	32.71	395.76	795.28	948.41	1094.30	1178.36
470	9.99	10.36	18.94	42.14	404.64	677.84	1124.71	1184.06	1036.77
480	10.09	10.42	20.73	34.83	257.99	849.74	945.48	1135.34	1085.56
490	9.92	10.49	17.20	51.61	322.10	817.17	1069.67	1421.99	1278.90
500	9.33	10.51	17.10	46.66	256.21	943.14	1473.89	1201.16	1080.92

Table B.14: Wine Dataset: Runtimes of [32]'s depth approximations

# trees	Approximation Level								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
10	10.12	27.86	33.89	23.03	22.36	33.31	126.28	153.80	176.57
20	18.51	27.61	71.96	728.03	827.19	902.79	1466.86	1363.49	1778.97
30	32.07	35.86	37.97	77.39	268.62	683.39	2432.21	3112.18	NA
40	24.34	76.29	123.11	189.24	2608.89	2982.73	3991.36	4103.35	5585.68
50	268.34	1404.66	NA	3049.95	3891.69	4208.42	NA	NA	NA
60	29.54	201.49	284.70	197.19	349.61	6033.12	6449.31	NA	NA
70	70.97	219.95	419.23	NA	4217.45	6136.51	6972.27	NA	NA
80	56.16	217.28	213.73	574.26	2651.09	3367.99	3851.38	4020.50	NA
90	126.98	793.15	1438.34	2191.92	NA	NA	NA	NA	NA
100	NA	1589.83	2940.74	NA	4381.19	6805.26	NA	NA	NA
110	103.86	NA	NA	NA	NA	NA	NA	NA	NA
120	173.27	1495.36	2167.12	3154.28	4119.08	5007.18	NA	NA	NA
130	NA	NA	NA	NA	NA	NA	NA	NA	NA
140	654.44	NA	NA	NA	NA	NA	NA	NA	NA
150	121.31	NA	NA	NA	NA	NA	NA	NA	NA
160	NA	4937.85	NA	NA	NA	NA	NA	NA	NA
170	NA	NA	NA	NA	NA	NA	NA	NA	NA
180	844.27	NA	NA	NA	NA	NA	NA	NA	NA
190	NA	NA	NA	NA	NA	NA	NA	NA	NA
200	497.43	NA	NA	NA	NA	NA	NA	NA	NA
210	440.14	NA	NA	NA	NA	NA	NA	NA	NA
220	1077.39	NA	NA	NA	NA	NA	NA	NA	NA
230	NA	NA	NA	NA	NA	NA	NA	NA	NA
240	NA	NA	NA	NA	NA	NA	NA	NA	NA
250	NA	NA	NA	NA	NA	NA	NA	NA	NA
260	NA	NA	NA	NA	NA	NA	NA	NA	NA
270	NA	NA	NA	NA	NA	NA	NA	NA	NA
280	NA	NA	NA	NA	NA	NA	NA	NA	NA
290	NA	NA	NA	NA	NA	NA	NA	NA	NA
300	NA	NA	NA	NA	NA	NA	NA	NA	NA
310	NA	NA	NA	NA	NA	NA	NA	NA	NA
320	NA	NA	NA	NA	NA	NA	NA	NA	NA
330	NA	NA	NA	NA	NA	NA	NA	NA	NA
340	NA	NA	NA	NA	NA	NA	NA	NA	NA
350	NA	NA	NA	NA	NA	NA	NA	NA	NA
360	NA	NA	NA	NA	NA	NA	NA	NA	NA
370	NA	NA	NA	NA	NA	NA	NA	NA	NA
380	NA	NA	NA	NA	NA	NA	NA	NA	NA
390	NA	NA	NA	NA	NA	NA	NA	NA	NA
400	NA	NA	NA	NA	NA	NA	NA	NA	NA
410	NA	NA	NA	NA	NA	NA	NA	NA	NA
420	NA	NA	NA	NA	NA	NA	NA	NA	NA
430	NA	NA	NA	NA	NA	NA	NA	NA	NA
440	NA	NA	NA	NA	NA	NA	NA	NA	NA
450	NA	NA	NA	NA	NA	NA	NA	NA	NA
460	NA	NA	NA	NA	NA	NA	NA	NA	NA
470	NA	NA	NA	NA	NA	NA	NA	NA	NA
480	NA	NA	NA	NA	NA	NA	NA	NA	NA
490	NA	NA	NA	NA	NA	NA	NA	NA	NA
500	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table B.15: Wine Dataset: Runtimes of [33]'s sampling approximations

Appendix C

Proofs and Figures of Chapter 4

C.1 Proofs of analytical results

Proof of Proposition 4.4.3 When the susceptible population is large enough (greater than 100), the number of daily new infections i_t is considered independent Poisson random variable with parameter $\frac{\beta_n S_t I_t}{N}$ (li2018parameter, hong2020estimation). With populations at the county or state level, we can approximate the Poisson distribution with $\mathcal{N}\left(\frac{\beta_n S_t I_t}{N}, \frac{\beta_n S_t I_t}{N}\right)$. If $i_t \sim \mathcal{N}\left(\frac{\beta_n S_t I_t}{N}, \frac{\beta_n S_t I_t}{N}\right)$ then $\frac{i_t - \hat{i_t}}{\sqrt{S_t I_t}} \sim \mathcal{N}\left(\frac{\sqrt{S_t I_t}}{N}\left(\beta_n - \hat{\beta}_n^t\right), \frac{\beta_n}{N}\right)$ Because we have $|\hat{\beta}_n^t - \beta_n| = \frac{\delta}{\sqrt{S_t I_t}}$, $\frac{|i_t - \hat{i_t}|}{\sqrt{S_t I_t}} \sim \mathcal{N}\left(\frac{\delta}{N}, \frac{\beta_n}{N}\right)$ making Z_t identically distributed within the same wave. Because the sequence of Z_t is independent and identically distributed, it is exchangeable

Proof of Proposition 4.5: Consider martingale M_t and note that

$$
\begin{aligned}
E[M_t|M_{t-1}] &= M_{t-1} E\left[\frac{s}{1-\exp(-s)} \exp\left(-s p_t\right)\right] \\
&\geq M_{t-1} \frac{s}{1-\exp(-s)} \exp\left(-s E[p_t]\right) \\
&= M_{t-1} \frac{s}{1-\exp(-s)} \exp\left(-s E\left[\frac{\sum_{u=T_{n-1}}^{t} 1_{Z_u \geq Z_t}}{t - T_{n-1}}\right]\right) \\
&= M_{t-1} \frac{s}{1-\exp(-s)} \exp\left(-\frac{s}{t-T_{n-1}} E\left[\sum_{u_1=T_{n-1}}^{T_n} 1_{Z_{u_1} \geq Z_t} + \sum_{u_2=(T_n+1)}^{t} 1_{Z_{u_2} \geq Z_t}\right]\right) \\
&= M_{t-1} \frac{s}{1-\exp(-s)} \exp\left(-\frac{s}{t-T_{n-1}} E\left[B_1 + B_2\right]\right)
\end{aligned}
$$

The inequality above follows simply from Jensen's inequality, while the subsequent equality is directly the definition of $p_t = \frac{\sum_{u=T_{n-1}}^{t} 1_{Z_u \geq Z_t}}{t - T_{n-1}}$. Next, we split the sum of indicator variables within the expectation into two parts. The first sum is the days during wave n where the predicted error was higher than the error for day t, which is part of wave $n+1$. The second sum is the days during wave $n+1$, where the error on that day is higher than on day t. We can consider these two sums as independent binomial distributions. The first binomial, B_1, is distributed as $B(T_n - T_{n-1}, P(Z_{u_1} \geq Z_t))$, where $T_n - T_{n-1}$ is the number of days in wave

n. The second binomial, B_2, is distributed as $B(t - T_n, P(Z_{u_2} \geq Z_t))$.

$$E[M_t|M_{t-1}] \geq M_{t-1}\frac{s}{1-\exp(-s)}\exp\left(-s\left(\frac{(T_n - T_{n-1})q_n + (t - T_n)q_{n+1}}{t - T_{n-1}}\right)\right) \tag{C.1}$$

$$= M_{t-1}\frac{s}{1-\exp(-s)}\exp\left(-sq_n\frac{T_n - T_{n-1}}{t - T_{n-1}}\right)\exp\left(-sq_{n+1}\frac{t - T_{n-1}}{t - T_{n-1}}\right)\exp\left(-sq_{n+1}\frac{T_{n-1} - T_n}{t - T_{n-1}}\right) \tag{C.2}$$

$$\geq M_{t-1}\frac{s\exp\left(-s(q_n + q_{n+1})\right)}{1-\exp(-s)}\exp\left(\frac{sq_{n+1}(T_n - T_{n-1})}{t - T_{n-1}}\right) \tag{C.3}$$

$$\geq M_{t-1}\frac{s\exp\left(-s(q_n + q_{n+1})\right)}{1-\exp(-s)}\left(1 + \frac{sq_{n+1}(T_n - T_{n-1})}{t - T_{n-1}}\right) \tag{C.4}$$

$$\geq M_{t-1}\frac{s\exp\left(-s(q_n + q_{n+1})\right)}{1-\exp(-s)}\left(1 + sq_{n+1}\frac{(T_n - T_{n-1}) - (t - T_n)}{T_n - T_{n-1}}\right) \tag{C.5}$$

By definition we have that $q_n = P(Z_{u_1} \geq Z_t)$ and $q_{n+1} = P(Z_{u_2} \geq Z_t)$, which we substitute in when we take the expectation of both binomials. (C.4) follows since $\exp(x) \geq 1 + x$. Finally, (C.5) follows by taking the first derivative of the convex function and finding the tangent line at T_n.

Proof of Lemma 4.5 When daily cases are increasing in a wave, then we have that $S_{u_2}I_{u_2} \leq S_tI_t$. We have from the proof of Proposition 4.4.3 that

$$Z_t \sim \mathcal{N}\left(\frac{\sqrt{S_tI_t}}{N}|\beta_{n+1} - \hat{\beta}_n|, \frac{\beta_{n+1}}{N}\right)$$

Because we are using an estimate of infection rate from wave n to predict for wave $n + 1$. Similarly, we have that

$$Z_{u_2} \sim \mathcal{N}\left(\frac{\sqrt{S_{u_2}I_{u_2}}}{N}|\beta_{n+1} - \hat{\beta}_n|, \frac{\beta_{n+1}}{N}\right)$$

From this we can see that Z_t and Z_{u_2} are similarly distributed but Z_t has an expectation with a higher magnitude. If we defined another variable, Z_{t_2} which has the same distribution as Z_{u_2}, then we would have that $P(Z_{u_2} \geq Z_t) \leq P(Z_{u_2} \geq Z_{t_2}) = 0.5$

$S_t * (1 - \frac{N}{R_0 * S_t}) \geq I_t$

Proof of Lemma 4.5: First let's assume that $\beta_n \leq \beta_{n+1}$

$$
\begin{aligned}
&P(Z_{u_1} \geq Z_t) \\
&= P\left(\frac{|\hat{i}_{u_1} - i_{u_1}|}{\sqrt{S_{u_1} I_{u_1}}} \geq \frac{|\hat{i_t} - i_t|}{\sqrt{S_t I_t}}\right) \\
&= \int_0^\infty P\left(\frac{|\hat{i}_{u_1} - i_{u_1}|}{\sqrt{S_{u_1} I_{u_1}}} \geq x\right) P\left(\frac{|\hat{i_t} - i_t|}{\sqrt{S_t I_t}} = x\right) dx \\
&= \int_0^\infty P(|\hat{i}_{u_1} - i_{u_1}| \geq \sqrt{S_{u_1} I_{u_1}} x) P(|\hat{i_t} - i_t| = \sqrt{S_t I_t} x) dx \\
&\leq \int_0^\infty P(|\hat{i}_{u_1} - i_{u_1}| \geq \sqrt{S_{u_1} I_{u_1}} x) * 2P(i_t = \hat{i_t} + \sqrt{S_t I_t} x) dx \\
&= \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^\infty P\left(|\hat{i}_{u_1} - i_{u_1}| \geq (x - \hat{i_t}) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}}\right) P(i_t = x) dx \\
&= \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^\infty P\left(|(\hat{i}_{u_1} - E[i_{u_1}]) + (E[i_{u_1}] - i_{u_1})| \geq (x - \hat{i_t}) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}}\right) P(i_t = x) dx \\
&\leq \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^\infty P\left(|\hat{i}_{u_1} - E[i_{u_1}]| + |E[i_{u_1}] - i_{u_1}| \geq (x - \hat{i_t}) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}}\right) P(i_t = x) dx \\
&\leq \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^\infty P\left(|E[i_{u_1}] - i_{u_1}| \geq (x - \hat{i_t}) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}} - \delta \frac{\sqrt{S_{u_1} I_{u_1}}}{N}\right) P(i_t = x) dx \\
&= \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^\infty P\left(|E[i_{u_1}] - i_{u_1}| \geq \left(x - \hat{i_t} - \frac{\delta \sqrt{S_t I_t}}{N}\right) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}}\right) P(i_t = x) dx
\end{aligned}
$$

We start with the definition of Z, and use the fact that the number of cases on a given day can be approximated with a Normal distribution, $\mathcal{N}(\frac{\beta SI}{N}, \frac{\beta SI}{N})$. We assume that the new infection rate is higher than the previous wave. This means that we are more likely to underestimate the number of new cases than overestimate when using the infection rate of the old wave, ie

$$P(i_t = \hat{i_t} + \sqrt{S_t I_t} x) + P(i_t = \hat{i_t} - \sqrt{S_t I_t} x) \leq 2P(i_t = \hat{i_t} + \sqrt{S_t I_t} x).$$

We can then use Chernoff bound ([110]) to get an upperbound on

$$P\left(|E[i_{u_1}] - i_{u_1}| \geq \left(x - \hat{i_t} - \frac{\delta \sqrt{S_t I_t}}{N}\right) \sqrt{\frac{S_{u_1} I_{u_1}}{S_t I_t}}\right)$$

because i_{u_1} is normally distributed. For the sake of notation, we define

$$(\mu, \sigma^2) := (\frac{\beta_{n+1} S_t I_t}{N}, \frac{\beta_{n+1} S_t I_t}{N})$$

, so that $i_t \sim \mathcal{N}(\mu, \sigma^2)$. We also note that by definition, $i_{u_1} \sim \mathcal{N}(\frac{\beta_n S_{u_1} I_{u_1}}{N}, \frac{\beta_n S_{u_1} I_{u_1}}{N})$ and $\hat{i_t} = \frac{\beta_n S_t I_t}{N}$, which implies that $\frac{S_{u_1} I_{u_1}}{\text{var}[i_{u_1}] S_t I_t} = \frac{N}{\beta_n S_t I_t} = \frac{1}{\hat{i_t}}$

$$
\begin{aligned}
&P(Z_{u_1} \geq Z_t) \\
&\leq \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^{\infty} 2\exp\left(-\frac{\left(x - \hat{i_t} - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{2\text{var}[i_{u_1}]} \frac{S_{u_1} I_{u_1}}{S_t I_t}\right) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) dx \\
&= \frac{2}{\sqrt{S_t I_t}} \int_{\hat{i_t}}^{\infty} 2\exp\left(-\frac{\left(x - \hat{i_t} - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{2\hat{i_t}}\right) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) dx \\
&= \frac{4}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i_t}}^{\infty} \exp\left(-\frac{1}{2}\left(\frac{\left(x - \hat{i_t} - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{\hat{i_t}} + \frac{(x-\mu)^2}{\sigma^2}\right)\right) dx \\
&\leq \frac{4}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i_t}}^{\infty} \exp\left(-\frac{1}{2}\left(\frac{\left(x - \hat{i_t} - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{\sigma^2} + \frac{(x-\mu)^2}{\sigma^2}\right)\right) dx
\end{aligned}
$$

The last inequality came from the fact we assumed that the infection rate of the new wave $\beta_n \leq \beta_{n+1}$, this means that $\hat{i_t} = \frac{\beta_n S_t I_t}{N} \leq \frac{\beta_{n+1} S_t I_t}{N} = E[i_t] = \sigma^2$. Because $\exp(\frac{-1}{x})$ has a

positive derivative as long as $x > 0$, we can say, $\exp(\frac{-1}{\hat{i}_t}) \leq \exp(\frac{-1}{\sigma^2})$

$$= \frac{4}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i}_t}^{\infty} \exp\left(-\frac{1}{2\sigma^2}\left(\left(x - \hat{i}_t - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2 + (x-\mu)^2\right)\right) dx$$

$$= \frac{4}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i}_t}^{\infty} \exp\left(-\frac{1}{2\sigma^2}\left(2\left(x - \frac{\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} + \mu}{2}\right)^2 + \frac{1}{2}\left(\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right)\right) dx$$

$$= \frac{4\exp\left(-\frac{1}{4\sigma^2}\left(\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right)}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i}_t}^{\infty} \exp\left(-\frac{1}{\sigma^2}\left(x - \frac{\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} + \mu}{2}\right)^2\right) dx$$

$$\leq \frac{4\exp\left(-\frac{1}{4\sigma^2}\left(\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right)}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i}_t}^{\infty} \frac{1}{1 + \frac{1}{\sigma^2}\left(x - \frac{\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} + \mu}{2}\right)^2} dx$$

$$= \frac{4\exp\left(-\frac{1}{4\sigma^2}\left(\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right)}{\sqrt{S_t I_t}\sqrt{2\pi}} \left[\frac{\pi}{2} + \tan^{-1}\left(\frac{\frac{\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} + \mu}{2} - \hat{i}_t}{\sigma}\right)\right]$$

$$\leq \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4\sigma^2}\left(\hat{i}_t + \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right)$$

$$= \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{\left(\beta_n \frac{S_t I_t}{N} - \beta_{n+1}\frac{S_t I_t}{N} + \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{4\beta_{n+1}\frac{S_t I_t}{N}}\right)$$

$$= \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4}\frac{(\beta_{n+1} - \beta_n - \frac{\delta}{\sqrt{S_t I_t}})^2}{\beta_{n+1}}\frac{S_t I_t}{N}\right)$$

Second, let's assume that $\beta_n \geq \beta_{n+1}$. This inequality follows very closely the previous

case when $\beta_n \leq \beta_{n+1}$ so we focus on highlighting where the proof differs.

$$
\begin{aligned}
&P(Z_{u_1} \geq Z_t) \\
&= P\left(\frac{|\hat{i}_{u_1} - i_{u_1}|}{\sqrt{S_{u_1}I_{u_1}}} \geq \frac{|\hat{i_t} - i_t|}{\sqrt{S_tI_t}}\right) \\
&\leq \int_0^\infty P(|\hat{i}_{u_1} - i_{u_1}| \geq \sqrt{S_{u_1}I_{u_1}}x) * 2P(i_t = \hat{i_t} - \sqrt{S_tI_t}x)dx \\
&= \frac{2}{\sqrt{S_tI_t}} \int_{-\infty}^{\hat{i_t}} P\left(|\hat{i}_{u_1} - i_{u_1}| \geq (\hat{i_t} - x)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}}\right) P(i_t = x)dx \\
&= \frac{2}{\sqrt{S_tI_t}} \int_{-\infty}^{\hat{i_t}} P\left(|(\hat{i}_{u_1} - E[i_{u_1}]) + (E[i_{u_1}] - i_{u_1})| \geq (\hat{i_t} - x)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}}\right) P(i_t = x)dx \\
&\leq \frac{2}{\sqrt{S_tI_t}} \int_{-\infty}^{\hat{i_t}} P\left(|\hat{i}_{u_1} - E[i_{u_1}]| + |E[i_{u_1}] - i_{u_1}| \geq (\hat{i_t} - x)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}}\right) P(i_t = x)dx \\
&\leq \frac{2}{\sqrt{S_tI_t}} \int_{-\infty}^{\hat{i_t}} P\left(|E[i_{u_1}] - i_{u_1}| \geq (\hat{i_t} - x)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}} - \delta\frac{\sqrt{S_{u_1}I_{u_1}}}{N}\right) P(i_t = x)dx \\
&= \frac{2}{\sqrt{S_tI_t}} \int_{-\infty}^{\hat{i_t}} P\left(|E[i_{u_1}] - i_{u_1}| \geq \left(\hat{i_t} - x - \frac{\delta\sqrt{S_tI_t}}{N}\right)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}}\right) P(i_t = x)dx
\end{aligned}
$$

Here, we assume that the new infection rate is lower than the previous wave. This means that we are more likely to overestimate the number of new cases than underestimate when using the infection rate of the old wave, ie $P(i_t = \hat{i_t} + \sqrt{S_tI_t}x) + P(i_t = \hat{i_t} - \sqrt{S_tI_t}x) \leq 2P(i_t = \hat{i_t} - \sqrt{S_tI_t}x)$.

Like before, we can then use Chernoff bound (matni2019tutorial) to get an upperbound on $P\left(|E[i_{u_1}] - i_{u_1}| \geq \left(\hat{i_t} - x - \frac{\delta\sqrt{S_tI_t}}{N}\right)\sqrt{\frac{S_{u_1}I_{u_1}}{S_tI_t}}\right)$ because i_{u_1} is normally distributed. For the sake of notation, we define $(\mu, \sigma^2) := (\frac{\beta_{n+1}S_tI_t}{N}, \frac{\beta_{n+1}S_tI_t}{N})$, so that $i_t \sim \mathcal{N}(\mu, \sigma^2)$.

$$
\begin{aligned}
&P(Z_{u_1} \geq Z_t) \\
&\leq \frac{2}{\sqrt{S_tI_t}} \int_{\hat{i_t}}^\infty 2\exp\left(-\frac{\left(\hat{i_t} - x - \frac{\delta\sqrt{S_tI_t}}{N}\right)^2}{2\text{var}[i_{u_1}]}\frac{S_{u_1}I_{u_1}}{S_tI_t}\right)\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)dx \\
&= \frac{4}{\sigma\sqrt{S_tI_t}\sqrt{2\pi}} \int_{\hat{i_t}}^\infty \exp\left(-\frac{1}{2}\left(\frac{\left(\hat{i_t} - x - \frac{\delta\sqrt{S_tI_t}}{N}\right)^2}{\hat{i_t}} + \frac{(x-\mu)^2}{\sigma^2}\right)\right)dx \\
&\leq \frac{4}{\sigma\sqrt{S_tI_t}\sqrt{2\pi}} \int_{\hat{i_t}}^\infty \exp\left(-\frac{1}{2}\left(\frac{\left(\hat{i_t} - x - \frac{\delta\sqrt{S_tI_t}}{N}\right)^2}{\hat{i_t}} + \frac{(x-\mu)^2}{\hat{i_t}}\right)\right)dx
\end{aligned}
$$

The last inequality came from the fact we assumed that the infection rate of the new wave $\beta_n \geq \beta_{n+1}$, this means that $\hat{i_t} = \frac{\beta_n S_tI_t}{N} \geq \frac{\beta_{n+1}S_tI_t}{N} = E[i_t] = \sigma^2$. Because $\exp(\frac{-1}{x})$ has a

positive derivative as long as $x > 0$, we can say, $\exp(\frac{-1}{\hat{i_t}}) \geq \exp(\frac{-1}{\sigma^2})$

$$
\begin{aligned}
&= \frac{4}{\sigma\sqrt{S_t I_t}\sqrt{2\pi}} \int_{\hat{i_t}}^{\infty} \exp\left(-\frac{1}{2\hat{i_t}}\left(\left(\hat{i_t} - x - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2 + (x-\mu)^2\right)\right) dx \\
&\leq \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4\hat{i_t}}\left(\hat{i_t} - \frac{\delta\sqrt{S_t I_t}}{N} - \mu\right)^2\right) \\
&= \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{\left(\beta_n \frac{S_t I_t}{N} - \beta_{n+1}\frac{S_t I_t}{N} - \frac{\delta\sqrt{S_t I_t}}{N}\right)^2}{4\beta_n \frac{S_t I_t}{N}}\right) \\
&= \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4}\frac{(\beta_n - \beta_{n+1} - \frac{\delta}{\sqrt{S_t I_t}})^2}{\beta_n}\frac{S_t I_t}{N}\right)
\end{aligned}
$$

We see that the final bound on $P(Z_{u_1} \geq Z_t)$ if $\beta_n \geq \beta_{n+1}$ has a very similar final structure as that if the infection rate is increasing. Therefore, we can summarize the bound as:

$$
P(Z_{u_1} \geq Z_t) \leq \sqrt{\frac{2\pi}{S_t I_t}} \exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{S_t I_t}})^2}{\max(\beta_n, \beta_{n+1})}\frac{S_t I_t}{N}\right)
$$

Proof of Theorem 4.5: Recall from Proposition 4.5,

$$
E[M_t|M_{t-1}] \geq M_{t-1}\frac{s\exp\left(-s(q_n + q_{n+1})\right)}{1-\exp(-s)}\left(1 + sq_{n+1}\frac{(T_n - T_{n-1}) - (t - T_n)}{T_n - T_{n-1}}\right)
$$

Substituting upper bounds on q_n (Lemma 4.5) and q_{n+1} (Lemma 4.5), we get that

$$
\begin{aligned}
&E[M_t|M_{t-1}] \geq \\
&M_{t-1}\frac{s\exp\left(-s\left(\sqrt{\frac{2\pi}{S_t I_t}}\exp\left(-\frac{1}{4}\frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{S_t I_t}})^2}{\max(\beta_n,\beta_{n+1})}\frac{S_t I_t}{N}\right) + 0.5\right)\right)}{1-\exp(-s)}\left(1 + s0.5\frac{(T_n - T_{n-1}) - (t - T_n)}{T_n - T_{n-1}}\right).
\end{aligned}
$$

This proves the final result.

H_t is a submartingale

Proof of Lemma C.1:

- $E[|H_t|] \leq \infty$: M_t is a submartigale and H_t is a finite shift on M_t
- H_t is based on only randomness generated by time t: The only variability in H_t comes from M_t
- $E[H_t|H_{t-1}] \geq H_{t-1}$:

$$
\begin{aligned}
&E[H_t|H_{t-1}] \\
&= E\left[M_t - \sum_{u=T_n+1}^{t}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right)|M_{t-1}\right] \\
&= E\left[M_t|M_{t-1}\right] - \sum_{u=T_n+1}^{t}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&\geq M_{t-1}\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(t-T_n)}{T_n-T_{n-1}}\right) \\
&- \sum_{u=T_n+1}^{t}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&= M_{t-1} + M_{t-1}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(t-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&- \sum_{u=T_n+1}^{t}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&\geq M_{t-1} + \left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(t-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&- \sum_{u=T_n+1}^{t}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&= M_{t-1} - \sum_{u=T_n+1}^{t-1}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right) \\
&= H_{t-1}
\end{aligned}
$$

Proof of Lemma 4.5. For all t, there exists a strictly greater than 0 probability, ϵ that the next T_λ data points are the strangest that the model has seen (ie each one is stranger than everything that's come before it). Then the p-scores for these points will be:

$$
\begin{aligned}
\{p_t, p_{t+1}, ..., p_{t+T_\lambda}\} &= \left\{\frac{1}{t}, \frac{1}{t+1}, \ldots, \frac{1}{t+T_\lambda}\right\} \\
&\geq \left\{\frac{1}{1}, \frac{1}{2}, \ldots, \frac{1}{T_\lambda}\right\}
\end{aligned}
$$

Then, $M_{T_\lambda} \geq M_{t-1}\left(\frac{s}{1-\exp(-s)}\right)^{T_\lambda}\prod_{i=1}^{T_\lambda}\left(\exp(-\frac{s}{i})\right)$.

The function $\left(\frac{s}{1-\exp(-s)}\right)^{T_\lambda}\prod_{i=1}^{T_\lambda}\left(\exp(-\frac{s}{i})\right)$ grows with T_λ, so we pick an integer high enough that $\left(\frac{s}{1-\exp(-s)}\right)^{T_\lambda}\prod_{i=1}^{T_\lambda}\left(\exp(-\frac{s}{i})\right) > \lambda$. Given that $M_{t-1} \geq 1$, then if the martingale grows at such a rate, it will hit λ before $t+T_\lambda$, meaning that $P(T \leq t+T_\lambda) \geq \epsilon$.

Proof of Theorem 4.5: Recall that the n^{th} wave starts at time T_n. Then, by optional stopping theorem, we have that

$$
\begin{aligned}
E[H_{T_n}] &\leq E[H_T] \\
E[M_{T_n}] &\leq E\left[M_T - \sum_{u=T_n+1}^{T}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right)\right] \\
M_0 &\leq \lambda - E\left[\sum_{u=T_n+1}^{T}\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\frac{(T_n-T_{n-1})-(u-T_n)}{T_n-T_{n-1}}\right)-1\right)\right] \\
M_0-\lambda &\leq -\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\right)-1\right)E[T-T_n] \\
&\quad + \frac{s^2\exp\left(-s(q_n+q_{n+1})\right)q_{n+1}}{1-\exp(-s)}E\left[\sum_{u=T_n+1}^{T}\frac{(u-T_n)}{T_n-T_{n-1}}\right] \\
&= -\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\right)-1\right)E[T-T_n] + \frac{s^2\exp\left(-s(q_n+q_{n+1})\right)q_{n+1}}{(1-\exp(-s))(T_n-T_{n-1})}E\left[\sum_{u=1}^{T-T_n}u\right] \\
&= -\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}\right)-1\right)E[T-T_n] \\
&\quad + \frac{s^2\exp\left(-s(q_n+q_{n+1})\right)q_{n+1}}{(1-\exp(-s))(T_n-T_{n-1})}E\left[\frac{(T-T_n)(T-T_n+1)}{2}\right] \\
&\leq -\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}-\frac{sq_{n+1}}{2(T_n-T_{n-1})}\right)-1\right)E[T-T_n] \\
&\quad + \frac{s^2\exp\left(-s(q_n+q_{n+1})\right)q_{n+1}}{2(1-\exp(-s))(T_n-T_{n-1})}E\left[(m(T-T_n))^2\right] \\
&= -\left(\frac{s\exp\left(-s(q_n+q_{n+1})\right)}{1-\exp(-s)}\left(1+sq_{n+1}-\frac{sq_{n+1}}{2(T_n-T_{n-1})}\right)-1\right)E[T-T_n] \\
&\quad + \frac{s^2\exp\left(-s(q_n+q_{n+1})\right)q_{n+1}}{2(1-\exp(-s))}(T_n-T_{n-1})m^2 \\
E[T-T_n] &\leq \frac{\lambda - M_0 + \frac{s^2\exp(-s(q_n+q_{n+1}))q_{n+1}}{2(1-\exp(-s))}(T_n-T_{n-1})m^2}{\left(\frac{s\exp(-s(q_n+q_{n+1}))}{1-\exp(-s)}\left(1+sq_{n+1}-\frac{sq_{n+1}}{2(T_n-T_{n-1})}\right)-1\right)}
\end{aligned}
$$

We have by definition of a λ-detectable wave that $E[T-T_n] \leq m(T_n-T_{n-1})$, for some $m \leq 1$. We can substitute $m=1$ in to the equation above to get a tighter bound on the expected time till detection than $E[T-T_n] \leq T_n - T_{n-1}$. We can use the tighter bound to find a new value of m and iterate. This would result in finding smaller and smaller values of m until we have that:

$$
m(T_n-T_{n-1}) = \frac{\lambda - M_0 + \frac{s^2\exp(-s(q_n+q_{n+1}))q_{n+1}}{2(1-\exp(-s))}(T_n-T_{n-1})m^2}{\left(\frac{s\exp(-s(q_n+q_{n+1}))}{1-\exp(-s)}\left(1+sq_{n+1}-\frac{sq_{n+1}}{2(T_n-T_{n-1})}\right)-1\right)}
$$

This is a simple quadratic in terms of m, which we can solve to find that:

$$
m = \frac{m_1 \pm \sqrt{m_1^2 - 4m_2(\lambda - M_0)}}{2m_2}
$$

where:

$$m_1 = \left(\frac{s \exp\left(-s(q_n + q_{n+1})\right)}{1 - \exp(-s)} \left(1 + sq_{n+1} - \frac{sq_{n+1}}{2(T_n - T_{n-1})} \right) - 1 \right) (T_n - T_{n-1})$$
$$m_2 = \frac{s^2 \exp\left(-s(q_n + q_{n+1})\right) q_{n+1}}{2(1 - \exp(-s))} (T_n - T_{n-1})$$

As long as $\frac{m_1 + \sqrt{m_1^2 - 4m_2(\lambda - M_0)}}{2m_2} > 1$ then the wave iterative substitution process for m holds (ie when you take $m = 1$, the bound provided by optional stopping theorem is smaller). Translating this we have that $\lambda - M_0 < m_1 - m_2$, which we can put into terms of wave length as:

$$T_n - T_{n-1} > \frac{\lambda - M_0 + \frac{s^2 \exp(-s(q_n + q_{n+1}))q_{n+1}}{2(1 - \exp(-s))}}{\frac{s \exp(-s(q_n + q_{n+1}))}{1 - \exp(-s)} \left(1 + \frac{sq_{n+1}}{2}\right) - 1}$$

In this case, we can take $m = \frac{m_1 - \sqrt{m_1^2 - 4m_2(\lambda - M_0)}}{2m_2}$. By substituting all the parameters in, we can write the bound for detection as:

$$E[T - T_n] \leq \frac{m_1 - \sqrt{m_1^2 - 4m_2(T_n - T_{n-1})(\lambda - M_0)}}{2m_2}$$

where:

$$m_1 = \left(\frac{s \exp\left(-s \left(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4} \frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c \right) + 0.5 \right) \right)}{1 - \exp(-s)} \left(1 + \frac{s}{2} - \frac{s}{4(T_n - T_{n-1})} \right) - 1 \right) (T_n - T_{n-1})$$
$$m_2 = \frac{s^2 \exp\left(-s(\sqrt{\frac{2\pi}{Nc}} \exp\left(-\frac{1}{4} \frac{(|\beta_n - \beta_{n+1}| - \frac{\delta}{\sqrt{Nc}})^2}{\max(\beta_n, \beta_{n+1})} c \right) + 0.5) \right)}{4(1 - \exp(-s))}$$

c is a constant such that $\frac{S_t I_t}{N} \geq c$ for all t in waves n and $n+1$, thus proving Theorem 4.5.

C.2 Graphs

C.2.1 September 01, 2020 Figures

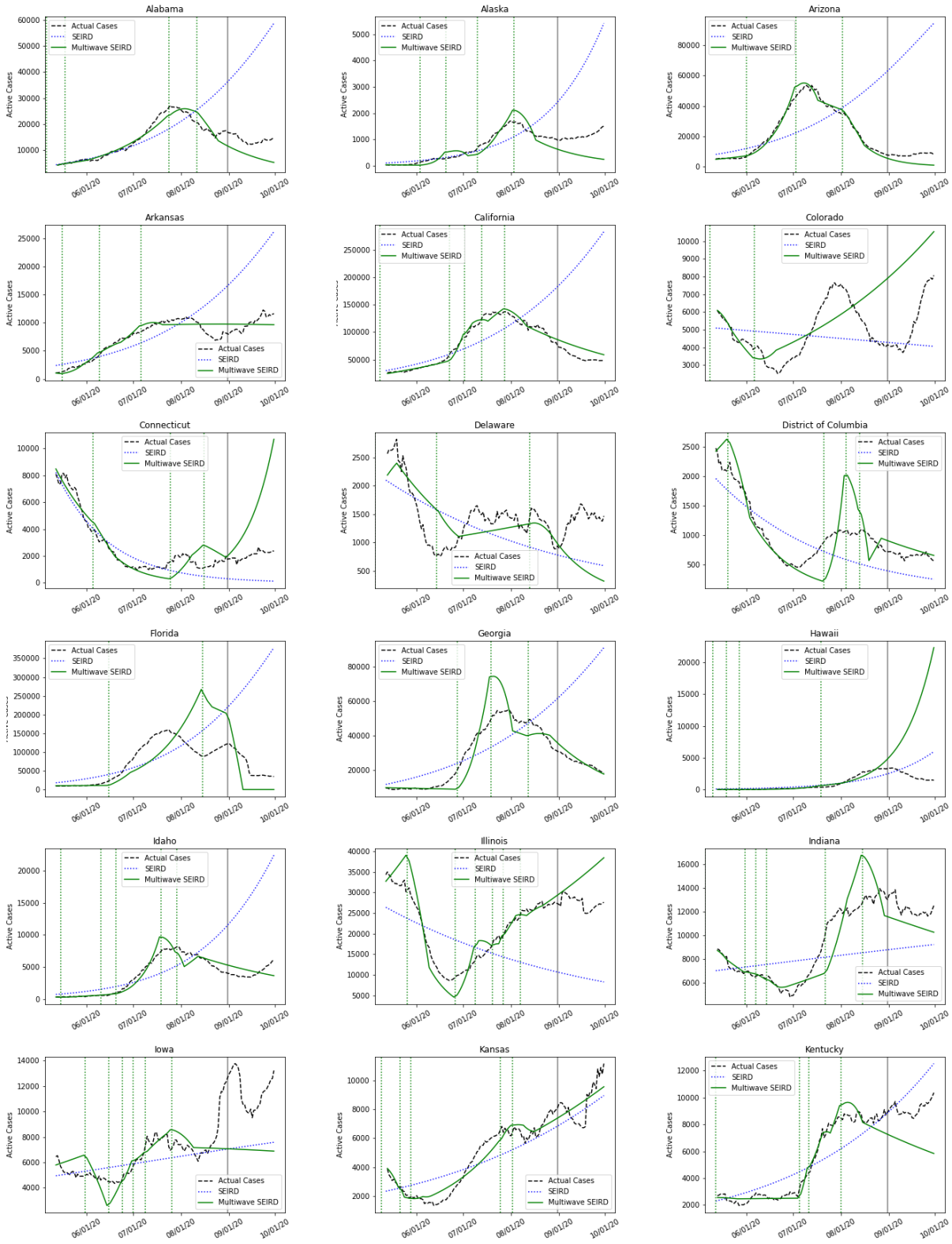


Figure C.1: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the A-K states when trained up to September 1, 2020

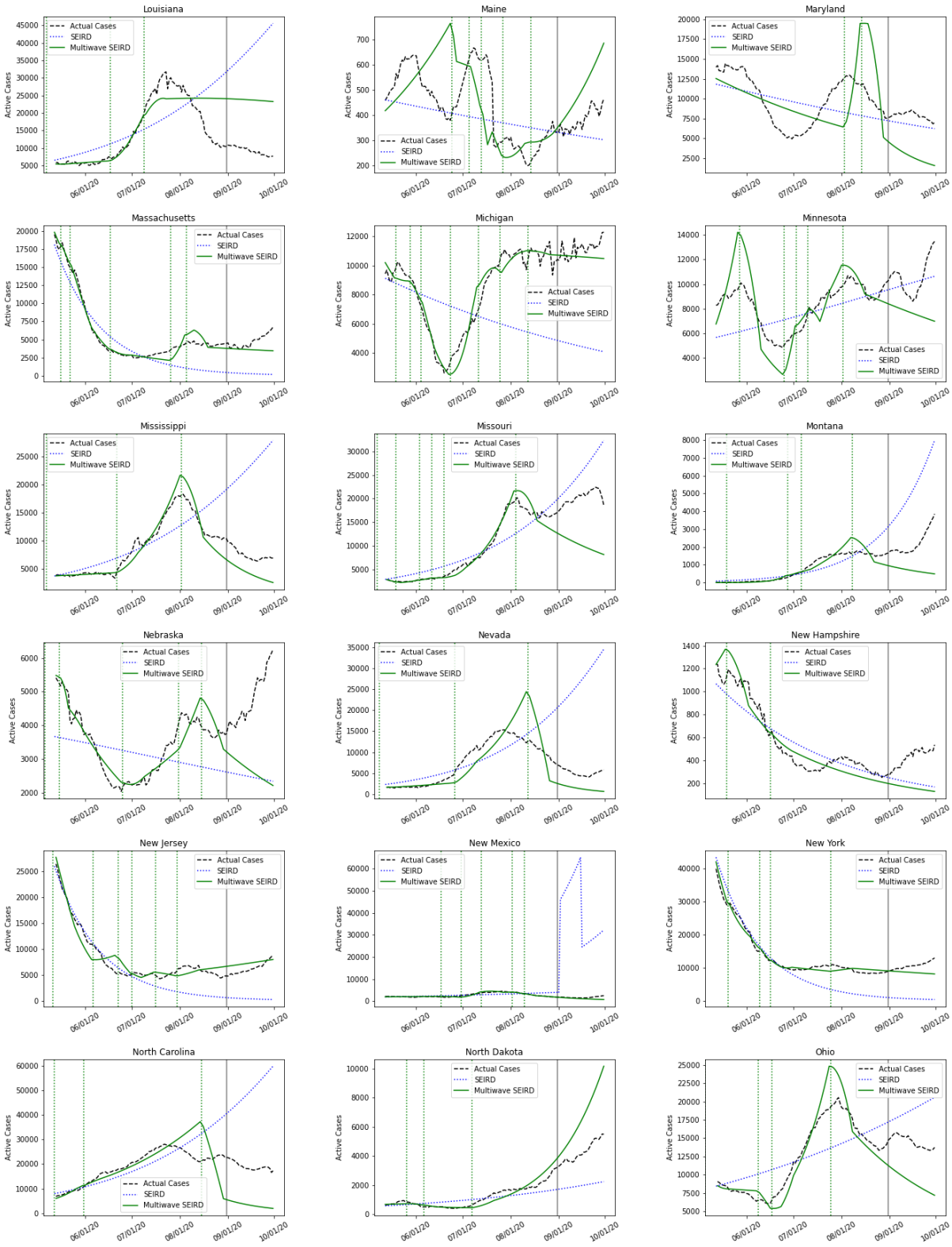


Figure C.2: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the L-O states when trained up to September 1, 2020

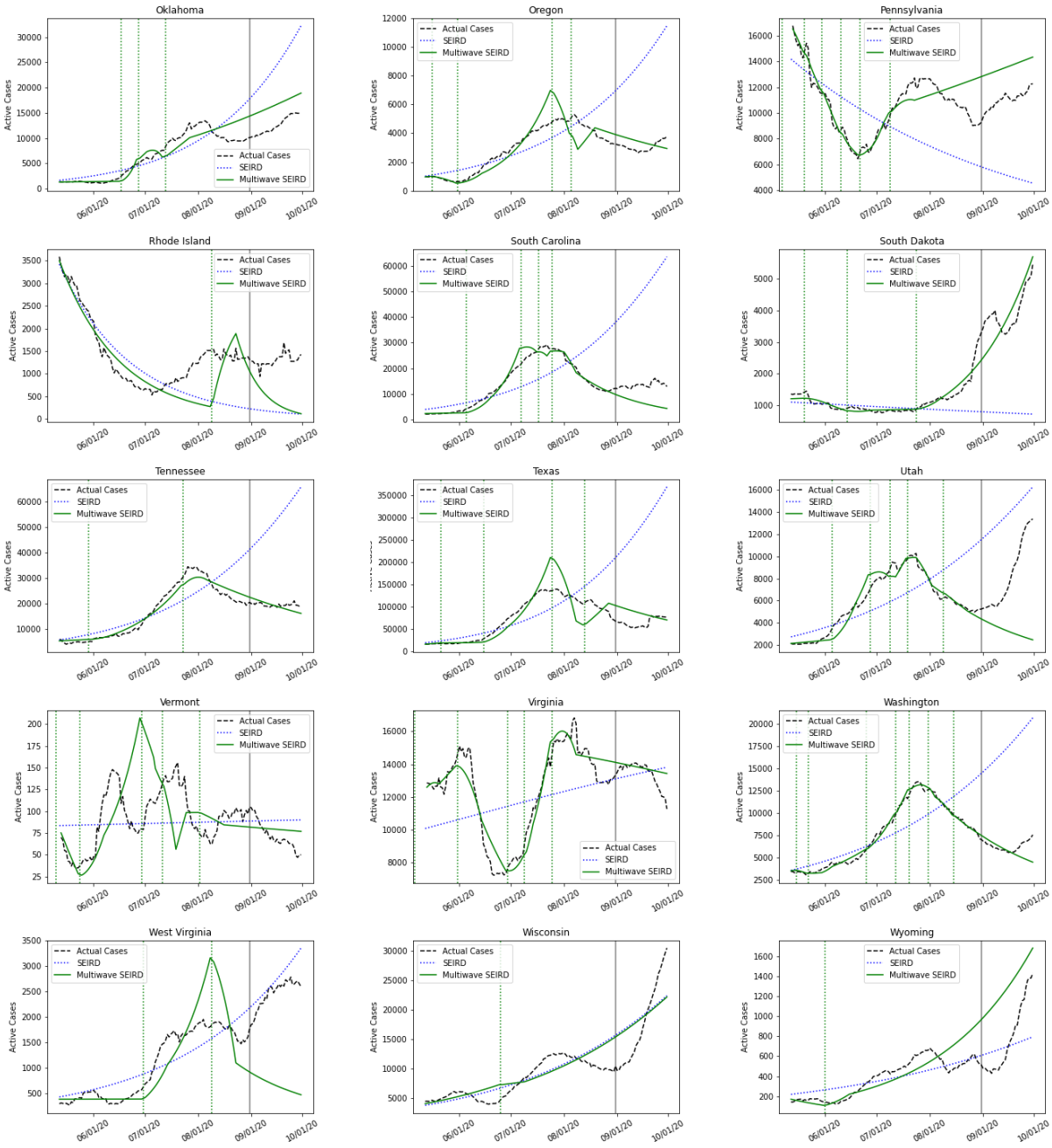


Figure C.3: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the O-Z states when trained up to September 1, 2020

C.2.2 January 15, 2021 Figures

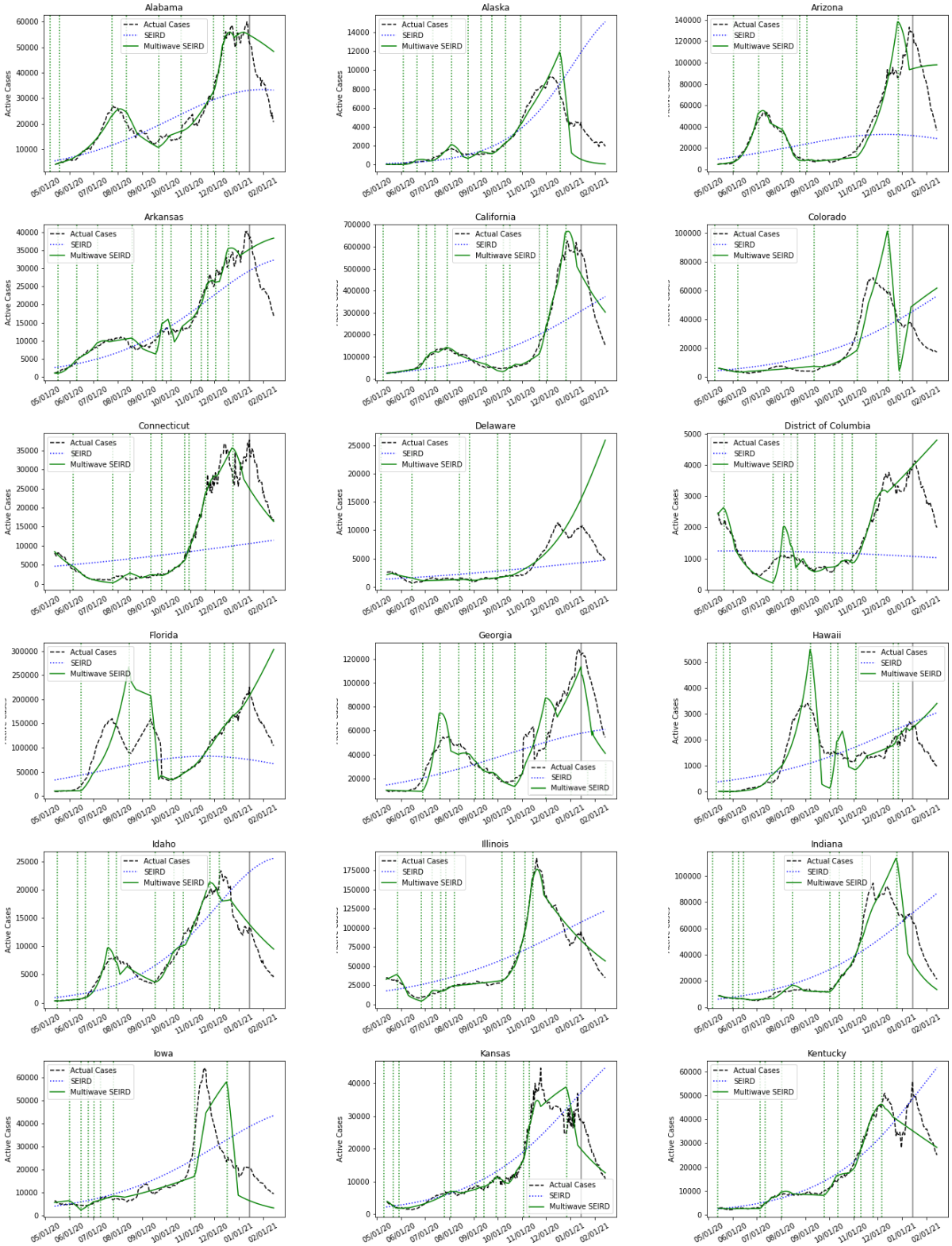


Figure C.4: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the A-K states when trained up to January 15, 2021

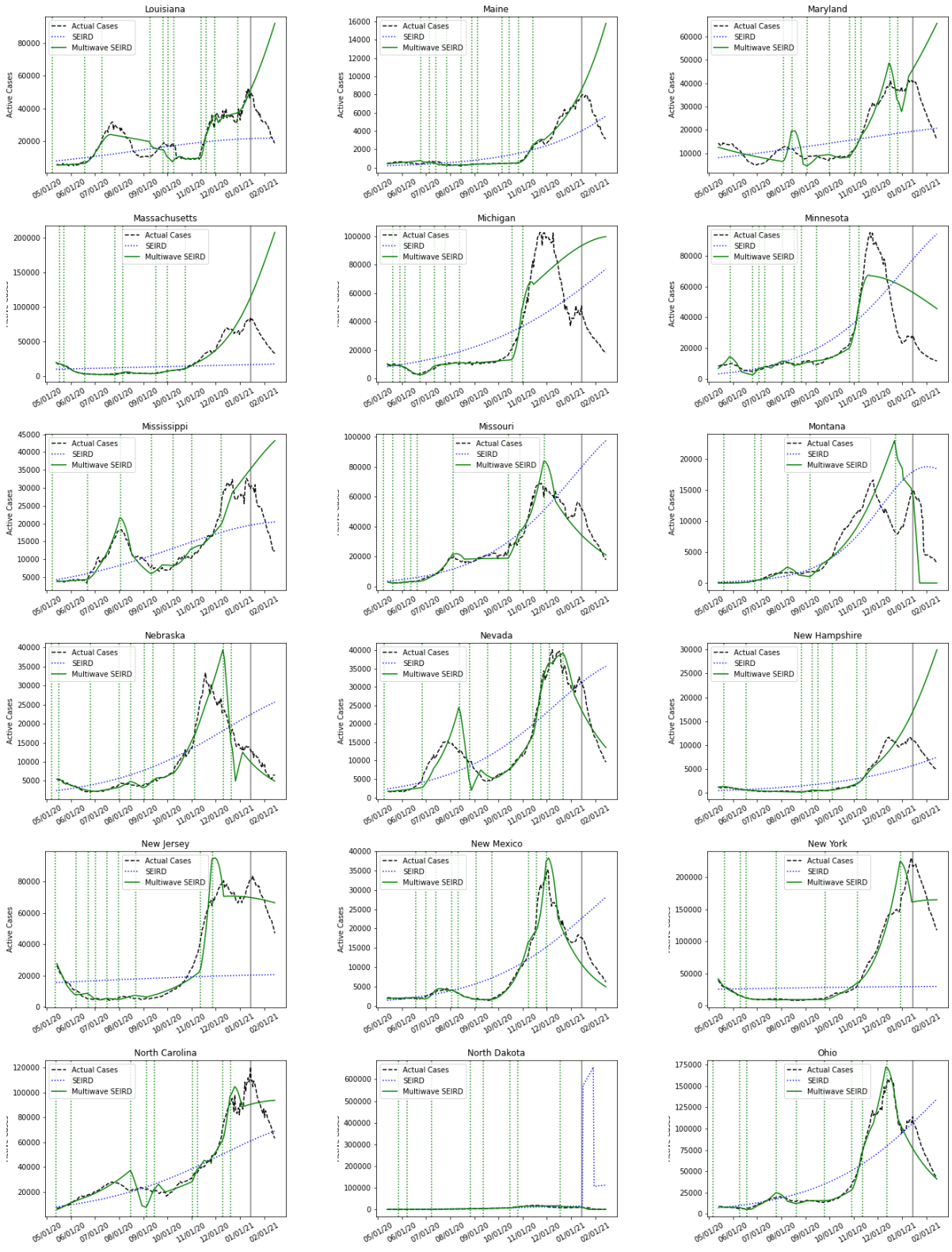


Figure C.5: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the L-O states when trained up to January 15, 2021

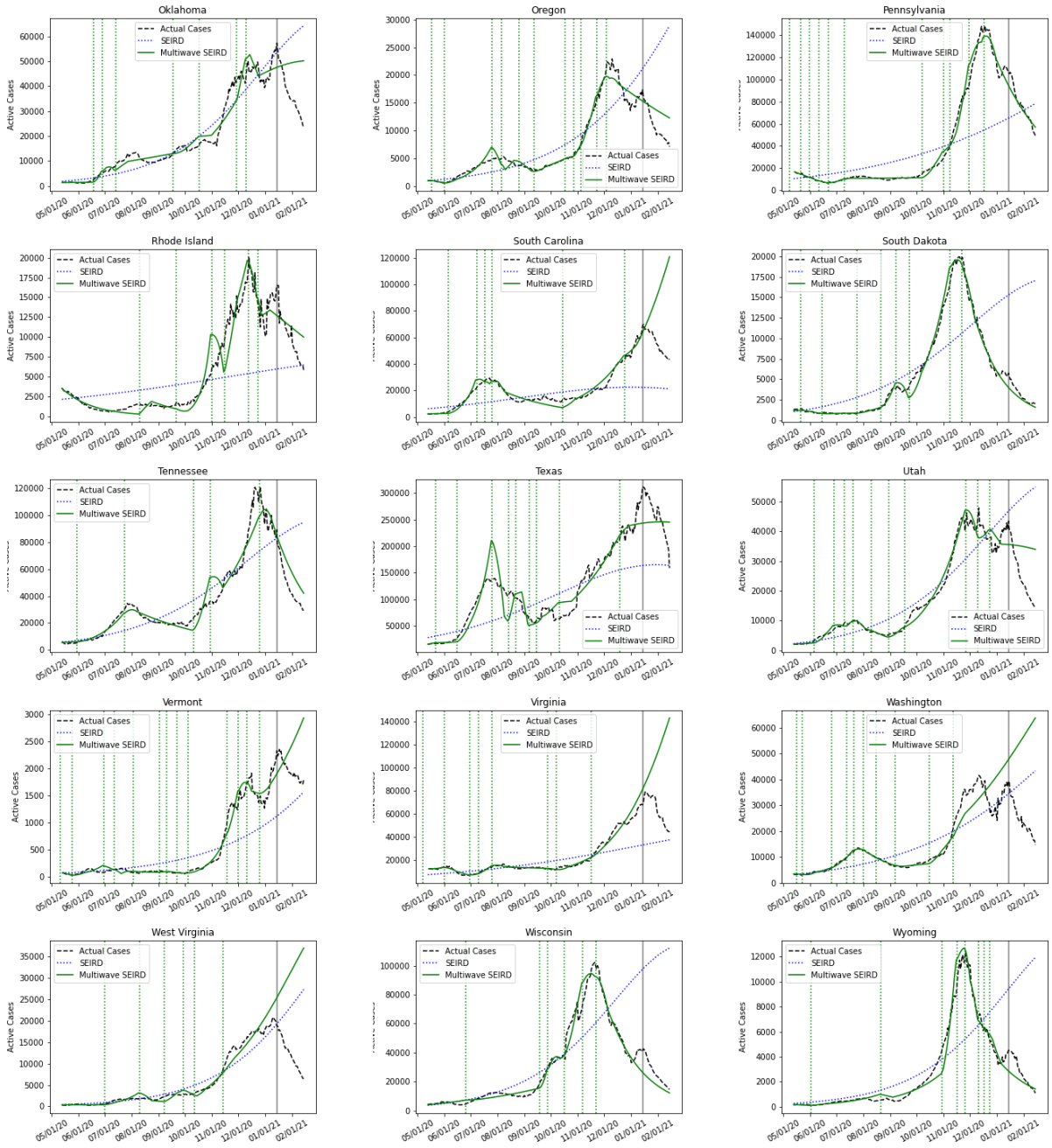


Figure C.6: SEIRD and multipeak SEIRD in-sample and out-of-sample predictions in the O-Z states when trained up to January 15, 2021

C.3 Tables

Table C.1: In-Sample and Out-of-Sample R-squared for Lasso Predicting Infection Rates

State	**In-Sample R-squared**	**Out-of-Sample R-squared**
Alabama	0.75	0.66
Alaska	0.65	0.52
Arizona	0.85	0.63
Arkansas	0.62	0.49
California	0.80	0.78
Colorado	0.64	0.52
Connecticut	0.73	0.73
Delaware	0.64	0.61
District of Columbia	0.43	0.39
Florida	0.62	0.75
Idaho	0.67	0.54
Illinois	0.60	0.50
Indiana	0.64	0.56
Iowa	0.71	0.67
Kansas	0.69	0.66
Kentucky	0.44	0.33
Louisiana	0.66	0.40
Maine	0.69	0.37
Maryland	0.40	0.43
Massachusetts	0.55	0.42
Michigan	0.77	0.72
Minnesota	0.79	0.68
Mississippi	0.78	0.53
Missouri	0.85	0.75
Montana	0.87	0.81
Nebraska	0.52	0.38
Nevada	0.47	0.55
New Hampshire	0.71	0.62
New Jersey	0.59	0.44
New Mexico	0.80	0.73
New York	0.94	0.89
North Carolina	0.54	0.06
North Dakota	0.95	0.93
Ohio	0.54	0.62
Oklahoma	0.43	0.34
Oregon	0.78	0.69
Pennsylvania	0.82	0.74
Rhode Island	0.80	0.70
South Carolina	0.82	0.75
South Dakota	0.75	0.75
Tennessee	0.66	0.55
Texas	0.32	0.38
Utah	0.68	0.56
Vermont	0.61	0.32
Virginia	0.78	0.62
Washington	0.76	0.69
West Virginia	0.83	0.69
Wisconsin	0.73	0.60
Wyoming	0.70	0.63
Average	0.68	0.59

Table C.2: Number of Detected Waves and Out-of-Sample MAPEs for the multiwave SEIRD models trained up to January 15, 2021 with different lambdas

state	**lambda**	**Number of Detected Waves**	**Out-of-Sample MAPE**
Alabama	205	10.0	0.05
	225	11.0	0.05
	250	6.0	0.79
	250	6.0	1.00
	300	6.0	0.52
Alaska	205	11.0	0.10
	225	10.0	0.04
	250	10.0	0.04
	250	11.0	0.01
	300	8.0	0.08
Arizona	205	8.0	0.03
	225	9.0	0.07
	250	6.0	2.59
	250	6.0	2.59
	300	8.0	0.12
Arkansas	205	13.0	0.04
	225	11.0	0.05
	250	7.0	0.11
	250	9.0	0.04
	300	7.0	0.05
California	205	12.0	0.02
	225	8.0	0.02
	250	8.0	0.02
	250	5.0	0.02
	300	5.0	0.02
Colorado	205	7.0	0.12
	225	7.0	0.13
	250	7.0	0.13
	250	5.0	0.14
	300	6.0	0.09
Connecticut	205	10.0	0.05
	225	8.0	0.02
	250	8.0	0.02
	250	8.0	0.02
	300	8.0	0.02
Delaware	205	7.0	0.26
	225	7.0	0.26
	250	6.0	0.01
	250	6.0	0.01

	300	6.0	0.01
District of Columbia	205	11.0	0.05
	225	10.0	0.03
	250	6.0	0.06
	250	6.0	0.06
	300	6.0	0.06
Florida	205	9.0	0.08
	225	9.0	0.04
	250	8.0	0.04
	250	8.0	0.04
	300	8.0	0.05
Georgia	205	9.0	0.23
	225	7.0	0.03
	250	4.0	0.05
	250	4.0	0.05
	300	4.0	0.05
Hawaii	205	11.0	0.06
	225	11.0	0.06
	250	10.0	0.09
	250	8.0	0.09
	300	8.0	0.09
Idaho	205	11.0	0.03
	225	11.0	0.03
	250	10.0	0.03
	250	8.0	0.03
	300	7.0	0.04
Illinois	205	11.0	0.01
	225	10.0	0.05
	250	5.0	0.17
	250	7.0	0.01
	300	8.0	0.01
Indiana	205	11.0	0.05
	225	10.0	0.10
	250	6.0	0.05
	250	6.0	0.05
	300	5.0	0.18
Iowa	205	9.0	0.06
	225	11.0	0.07
	250	11.0	0.00
	250	9.0	0.06
	300	5.0	0.07
Kansas	205	13.0	0.02
	225	10.0	0.02
	250	12.0	0.07

	250	12.0	0.07
	300	10.0	0.07
	205	11.0	0.05
	225	7.0	0.09
Kentucky	250	8.0	0.03
	250	4.0	0.08
	300	4.0	0.08
	205	12.0	0.12
	225	8.0	0.05
Louisiana	250	6.0	0.02
	250	6.0	0.02
	300	8.0	0.16
	205	12.0	0.20
	225	11.0	0.20
Maine	250	8.0	0.04
	250	8.0	0.04
	300	8.0	0.04
	205	10.0	0.10
	225	8.0	0.09
Maryland	250	7.0	0.01
	250	4.0	0.10
	300	4.0	0.10
	205	9.0	0.29
	225	8.0	0.28
Massachusetts	250	10.0	0.02
	250	7.0	0.28
	300	7.0	0.27
	205	10.0	0.22
	225	12.0	0.31
Michigan	250	9.0	0.31
	250	11.0	0.01
	300	9.0	0.04
	205	11.0	0.17
	225	10.0	0.15
Minnesota	250	8.0	0.16
	250	6.0	0.15
	300	6.0	0.15
	205	8.0	0.08
	225	9.0	0.08
Mississippi	250	9.0	0.08
	250	7.0	0.14
	300	6.0	0.15
	205	10.0	0.04
	225	7.0	0.05
Missouri			

	250	10.0	0.07
	250	6.0	0.05
	300	6.0	0.06
	205	7.0	0.03
	225	8.0	0.03
Montana	250	9.0	0.08
	250	9.0	0.08
	300	9.0	0.08
	205	12.0	0.02
	225	12.0	0.03
Nebraska	250	8.0	0.09
	250	9.0	0.03
	300	7.0	0.03
	205	11.0	0.01
	225	11.0	0.01
Nevada	250	7.0	0.29
	250	8.0	0.01
	300	6.0	0.12
	205	9.0	0.35
	225	7.0	0.35
New Hampshire	250	6.0	0.34
	250	6.0	0.34
	300	7.0	0.34
	205	10.0	0.01
	225	7.0	0.01
New Jersey	250	7.0	0.18
	250	6.0	0.01
	300	6.0	0.01
	205	11.0	0.06
	225	6.0	0.03
New Mexico	250	6.0	0.03
	250	4.0	0.13
	300	5.0	0.01
	205	8.0	0.02
	225	15.0	0.13
New York	250	6.0	0.02
	250	7.0	0.24
	300	12.0	0.05
	205	10.0	0.01
	225	6.0	0.03
North Carolina	250	8.0	0.02
	250	6.0	0.14
	300	8.0	0.05
	205	9.0	0.02

North Dakota

	225	10.0	0.02
	250	10.0	0.02
	250	8.0	0.03
	300	8.0	0.13
	205	10.0	0.04
	225	11.0	0.06
Ohio	250	7.0	0.21
	250	7.0	0.21
	300	8.0	0.21
	205	8.0	0.04
	225	8.0	0.03
Oklahoma	250	7.0	0.10
	250	7.0	0.10
	300	7.0	0.10
	205	12.0	0.02
	225	10.0	0.01
Oregon	250	10.0	0.01
	250	10.0	0.08
	300	9.0	0.09
	205	12.0	0.01
	225	12.0	0.03
Pennsylvania	250	8.0	0.05
	250	10.0	0.08
	300	10.0	0.08
	205	7.0	0.01
	225	6.0	0.06
Rhode Island	250	6.0	0.06
	250	4.0	0.07
	300	4.0	0.07
	205	7.0	0.09
	225	11.0	0.06
South Carolina	250	8.0	0.07
	250	8.0	0.07
	300	6.0	0.22
	205	9.0	0.02
	225	9.0	0.01
South Dakota	250	9.0	0.01
	250	9.0	0.01
	300	7.0	20.60
	205	6.0	0.02
	225	5.0	0.15
Tennessee	250	5.0	0.15
	250	6.0	0.17
	300	4.0	0.18

Texas	205	10.0	0.02
	225	9.0	0.02
	250	9.0	0.02
	250	9.0	0.05
	300	6.0	0.02
Utah	205	11.0	0.03
	225	13.0	0.02
	250	7.0	0.04
	250	6.0	0.03
	300	6.0	0.03
Vermont	205	14.0	0.04
	225	10.0	0.06
	250	8.0	0.12
	250	8.0	0.12
	300	7.0	0.05
Virginia	205	9.0	0.12
	225	11.0	0.12
	250	8.0	0.12
	250	7.0	0.12
	300	6.0	0.04
Washington	205	11.0	0.15
	225	8.0	0.14
	250	9.0	0.10
	250	6.0	0.05
	300	4.0	0.18
West Virginia	205	7.0	0.23
	225	5.0	0.23
	250	5.0	0.23
	250	5.0	0.23
	300	6.0	0.24
Wisconsin	205	7.0	0.04
	225	7.0	0.04
	250	7.0	0.02
	250	6.0	0.02
	300	5.0	0.09
Wyoming	205	9.0	0.03
	225	5.0	0.07
	250	5.0	0.08
	250	6.0	0.09
	300	7.0	0.09

References

[1] U. Nations. "The 17 goals — sustainable development." (2024), URL: https://sdgs.un.org/goals..

[2] H. Jouglaf. "The increasing appeal of green tech investing." (2023), URL: https://www.financialpoise.com/the-increasing-appeal-of-green-tech-investing/#:~:text=In%20a%20continuance%20of%20a,fuel%20supply%20%E2%80%93%20yet%20another%20first.

[3] "Summary: The inflation reduction act of 2022." (2023), URL: https://www.democrats.senate.gov/imo/media/doc/inflation_reduction_act_one_page_summary.pdf.

[4] R. Bland, A. Granskog, and T. Nauclér. "Accelerating toward net zero: The green business building opportunity." (2022), URL: https://www.mckinsey.com/capabilities/sustainability/our-insights/accelerating-toward-net-zero-the-green-business-building-opportunity.

[5] "Demand for electric cars is booming, with sales expected to leap 35% this year after a record-breaking 2022." (2023), URL: https://www.iea.org/news/demand-for-electric-cars-is-booming-with-sales-expected-to-leap-35-this-year-after-a-record-breaking-2022.

[6] F. M. Center, *Ford to lead america's shift to electric vehicles with new mega campus in tennessee and twin battery plants in kentucky*, Sep. 2021. URL: https://media.ford.com/content/fordmedia/fna/us/en/news/2021/09/27/ford-to-lead-americas-shift-to-electric-vehicles.html.

[7] P. A. Eisenstein, *After slow start, stellantis bets big on evs*, Oct. 2022. URL: https://www.forbes.com/wheels/news/stellantis-ev-plans/.

[8] N. E. Boudette and C. Davenport, *G.m. will sell only zero-emission vehicles by 2035*, Jan. 2021. URL: https://www.nytimes.com/2021/01/28/business/gm-zero-emission-vehicles.html.

[9] B. Murray, *The paradox of declining renewable costs and rising electricity prices*, Jun. 2019. URL: https://www.forbes.com/sites/brianmurray1/2019/06/17/the-paradox-of-declining-renewable-costs-and-rising-electricity-prices/?sh=8085e661d5b4.

[10] E. Cooke, *Signal: Charging patent grants increasing, as california passes v2g mandate*, Aug. 2023. URL: https://www.just-auto.com/news/bidirectional-charging-patent-grants-are-increasing-as-california-passes-v2g-mandate/?cf-view.

[11] L. Shi, T. Lv, and Y. Wang, "Vehicle-to-grid service development logic and management formulation," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 4, pp. 935–947, 2019.

[12] G. Broneske and D. Wozabal, "How do contract parameters influence the economics of vehicle-to-grid?" *Manufacturing & Service Operations Management*, vol. 19, no. 1, pp. 150–164, 2017.

[13] M. Caramanis and J. M. Foster, "Management of electric vehicle charging to mitigate renewable generation intermittency and distribution network congestion," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Ieee, 2009, pp. 4717–4722.

[14] S. B. Peterson, J. Whitacre, and J. Apt, "The economics of using plug-in hybrid electric vehicle battery packs for grid storage," *Journal of Power Sources*, vol. 195, no. 8, pp. 2377–2384, 2010.

[15] Y. Zhang, M. Lu, and S. Shen, "On the values of vehicle-to-grid electricity selling in electric vehicle sharing," *Manufacturing & Service Operations Management*, vol. 23, no. 2, pp. 488–507, 2021.

[16] M. T. Kahlen, W. Ketter, and J. van Dalen, "Electric vehicle virtual power plant dilemma: Grid balancing versus customer mobility," *Production and Operations Management*, vol. 27, no. 11, pp. 2054–2070, 2018.

[17] M. U. Ali, A. Zafar, S. H. Nengroo, S. Hussain, M. Junaid Alvi, and H.-J. Kim, "Towards a smarter battery management system for electric vehicle applications: A critical review of lithium-ion battery state of charge estimation," *Energies*, vol. 12, no. 3, p. 446, 2019.

[18] L. Chen, L. He, and Y. Zhou, "An exponential cone programming approach for managing electric vehicle charging," *Operations Research*, 2023.

[19] S. M. Varma, F. Castro, and S. T. Maguluri, "Electric vehicle fleet and charging infrastructure planning," *arXiv preprint arXiv:2306.10178*, 2023.

[20] O. Q. Wu, Ş. Yücel, and Y. Zhou, "Smart charging of electric vehicles: An innovative business model for utility firms," *Manufacturing & Service Operations Management*, vol. 24, no. 5, pp. 2481–2499, 2022.

[21] L. He, G. Ma, W. Qi, and X. Wang, "Charging an electric vehicle-sharing fleet," *Manufacturing & Service Operations Management*, vol. 23, no. 2, pp. 471–487, 2021.

[22] C. M. Flath, J. P. Ilg, S. Gottwalt, H. Schmeck, and C. Weinhardt, "Improving electric vehicle charging coordination through area pricing," *Transportation Science*, vol. 48, no. 4, pp. 619–634, 2014.

[23] J. Hu, S. You, M. Lind, and J. Østergaard, "Coordinated charging of electric vehicles for congestion prevention in the distribution grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 703–711, 2013.

[24] C. Kaps, S. Marinesi, and S. Netessine, "When should the off-grid sun shine at night? optimum renewable generation and energy storage investments," *Management Science*, 2023.

[25] X. Peng, O. Q. Wu, and G. Souza, "Renewable, flexible, and storage capacities: Friends or foes?" *Available at SSRN 3983678*, 2021.

[26] N. Löhndorf and D. Wozabal, "The value of coordination in multimarket bidding of grid energy storage," *Operations Research*, vol. 71, no. 1, pp. 1–22, 2023.

[27] D. F. Salas and W. B. Powell, "Benchmarking a scalable approximate dynamic programming algorithm for stochastic control of grid-level energy storage," *INFORMS Journal on Computing*, vol. 30, no. 1, pp. 106–123, 2018.

[28] M. Shaked and J. Shanthikumar, *Stochastic Orders* (Springer Series in Statistics). Springer New York, 2007, ISBN: 978-0-387-34675-5. URL: https://books.google.com/books?id=rPiToBK2rwwC.

[29] D. Bertsimas, A. O'Hair, S. Relyea, and J. Silberholz, "An analytics approach to designing combination chemotherapy regimens for cancer," *Management Science*, vol. 62, no. 5, pp. 1511–1531, 2016.

[30] L. Baardman, M. C. Cohen, K. Panchamgam, G. Perakis, and D. Segev, "Scheduling promotion vehicles to boost profits," *Management Science*, vol. 65, no. 1, pp. 50–70, 2019.

[31] K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi, "Analytics for an online retailer: Demand forecasting and price optimization," *Manufacturing & Service Operations Management*, vol. 18, no. 1, pp. 69–88, 2016.

[32] V. V. Mišić, "Optimization of tree ensembles," *Operations Research*, vol. 68, no. 5, pp. 1605–1624, 2020.

[33] M. Biggs, R. Hariss, and G. Perakis, "Constrained optimization of objective functions determined from random forests," *Production and Operations Management*, 2022.

[34] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," 1984.

[35] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[36] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, *et al.*, "Xgboost: Extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[37] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Machine learning*, vol. 42, no. 3, pp. 287–320, 2001.

[38] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[39] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.

[40] V. V. Mišić, "Data, models and decisions for large-scale stochastic optimization problems," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[41] D. Bertsimas and V. V. Mišić, "Exact first-choice product line optimization," *Operations Research*, vol. 67, no. 3, pp. 651–670, 2019.

[42] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma, "Strong mixed-integer programming formulations for trained neural networks," *Mathematical Programming*, pp. 1–37, 2020.

[43] R. Bunel, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar, "A unified view of piecewise linear neural network verification," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 4795–4804.

[44] C.-H. Cheng, G. Nührenberg, and H. Ruess, "Maximum resilience of artificial neural networks," in *International Symposium on Automated Technology for Verification and Analysis*, Springer, 2017, pp. 251–268.

[45] M. Fischetti and J. Jo, "Deep neural networks and mixed integer linear optimization," *Constraints*, vol. 23, no. 3, pp. 296–309, 2018.

[46] A. Kumar, T. Serra, and S. Ramalingam, "Equivalent and approximate transformations of deep neural networks," *arXiv preprint arXiv:1905.11428*, 2019.

[47] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," *arXiv preprint arXiv:1711.07356*, 2017.

[48] O. Besbes, R. Phillips, and A. Zeevi, "Testing the validity of a demand model: An operations perspective," *Manufacturing & Service Operations Management*, vol. 12, no. 1, pp. 162–183, 2010.

[49] Y.-C. Chen and V. V. Mišić, "Assortment optimization under the decision forest model," *arXiv preprint arXiv:2103.14067*, 2021.

[50] J. Kim, J.-P. P. Richard, and M. Tawarmalani, "A reciprocity between tree ensemble optimization and multilinear optimization," 2022.

[51] M. Mistry, D. Letsios, G. Krennrich, R. M. Lee, and R. Misener, "Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded," *INFORMS Journal on Computing*, vol. 33, no. 3, pp. 1103–1119, 2021.

[52] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[53] N. Kallus and X. Mao, "Stochastic optimization forests," *Management Science*, 2022.

[54] I. V. Tetko, V. Y. Tanchuk, T. N. Kasheva, and A. E. Villa, "Estimation of aqueous solubility of chemical compounds using e-state indices," *Journal of chemical information and computer sciences*, vol. 41, no. 6, pp. 1488–1493, 2001.

[55] J. Huuskonen, "Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 3, pp. 773–777, 2000.

[56] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," *Cement and Concrete research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[57] S. Aeberhard, D. Coomans, and O. de Vel, "The classification performance of rda," *Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep*, pp. 92–01, 1992.

[58] L. Baardman, S. B. Boroujeni, T. Cohen-Hillel, K. Panchamgam, and G. Perakis, "Detecting customer trends for optimal promotion targeting," *Manufacturing & Service Operations Management*, 2020.

[59] W. H. Organization. "Naming the coronavirus disease (covid-19) and the virus that causes it." (2020), URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it..

[60] D. S. Hui, E. I. Azhar, T. A. Madani, F. Ntoumi, R. Kock, O. Dar, G. Ippolito, T. D. Mchugh, Z. A. Memish, C. Drosten, *et al.*, "The continuing 2019-ncov epidemic threat of novel coronaviruses to global health—the latest 2019 novel coronavirus outbreak in wuhan, china," *International Journal of Infectious Diseases*, vol. 91, pp. 264–266, 2020.

[61] W. H. Organization. "Who director-general's opening remarks at the media briefing on covid-19 - 11 march 2020." (2020), URL: https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020..

[62] W. H. Organization. "Transmission of sars-cov-2: Implications for infection prevention precautions." (2020), URL: https://www.who.int/news-room/commentaries/detail/transmission-of-sars-cov-2-implications-for-infection-prevention-precautions.

[63] B. Haynes, N. E. Messonnier, and M. S. Cetron, "First travel-related case of 2019 novel coronavirus detected in united states: Press release, tuesday, january 21, 2020," 2020.

[64] U. B. of Labor Statistics. "Employment situation news release." (May 2020), URL: https://www.bls.gov/news.release/archives/empsit_05082020.htm.

[65] U. B. of Economic Analysis. "Gross domestic product, 2nd quarter 2020 (second estimate); corporate profits, 2nd quarter 2020 (preliminary estimate)." (Aug. 2020), URL: https://www.bea.gov/news/2020/gross-domestic-product-2nd-quarter-2020-second-estimate-corporate-profits-2nd-quarter.

[66] M. Wang. "Stanford research provides a snapshot of a new working-from-home economy." (2020), URL: https://news.stanford.edu/2020/06/29/snapshot-new-working-home-economy/#:~:text=We%5C%20see%5C%20an%5C%20incredible%5C%2042,working%5C%20on%5C%20their%5C%20business%5C%20premises..

[67] J. H. University. "Coronavirus covid-19 global cases by the center for systems science and engineering (csse) at johns hopkins university (jhu)." (2020), URL: https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6

[68] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[69] F. Brauer, "Compartmental models in epidemiology," in *Mathematical epidemiology*, Springer, 2008, pp. 19–79.

[70] B. Cantó, C. Coll, and E. Sánchez, "Estimation of parameters in a structured sir model," *Advances in Difference Equations*, vol. 2017, no. 1, p. 33, 2017.

[71] D. Marcondes, "Robust parameter estimation in dynamical systems via statistical learning with an application to epidemiological models," *arXiv preprint arXiv:2007.14229*, 2020.

[72] C. Li, Y. Pei, M. Zhu, and Y. Deng, "Parameter estimation on a stochastic sir model with media coverage," *Discrete Dynamics in Nature and Society*, vol. 2018, 2018.

[73] J. Baek, V. F. Farias, A. Georgescu, R. Levi, T. Peng, D. Sinha, J. Wilde, and A. Zheng, "The limits to learning an sir process: Granular forecasting for covid-19," *arXiv preprint arXiv:2006.06373*, 2020.

[74] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and information systems*, vol. 51, no. 2, pp. 339–367, 2017.

[75] V. V. Veeravalli and T. Banerjee, "Quickest change detection," in *Academic press library in signal processing*, vol. 3, Elsevier, 2014, pp. 209–255.

[76] G. Texier, M. Farouh, L. Pellegrin, M. L. Jackson, J.-B. Meynard, X. Deparis, and H. Chaudet, "Outbreak definition by change point analysis: A tool for public health decision?" *BMC medical informatics and decision making*, vol. 16, no. 1, pp. 1–12, 2016.

[77] S. Jiang, Q. Zhou, X. Zhan, and Q. Li, "Bayessmiles: Bayesian segmentation modeling for longitudinal epidemiological studies," *medRxiv*, pp. 2020–10, 2021.

[78] J. Juodakis and S. Marsland, "Epidemic changepoint detection in the presence of nuisance changes," *arXiv preprint arXiv:2008.08240*, 2020.

[79] C. Anastassopoulou, L. Russo, A. Tsakris, and C. Siettos, "Data-based analysis, modelling and forecasting of the covid-19 outbreak," *PloS one*, vol. 15, no. 3, e0230405, 2020.

[80] G. C. Calafiore, C. Novara, and C. Possieri, "A time-varying sird model for the covid-19 contagion in italy," *Annual reviews in control*, 2020.

[81] G. Gaeta, "A simple sir model with a large set of asymptomatic infectives," *arXiv preprint arXiv:2003.08720*, 2020.

[82] G. Giordano, F. Blanchini, R. Bruno, P. Colaneri, A. Di Filippo, A. Di Matteo, and M. Colaneri, "Modelling the covid-19 epidemic and implementation of population-wide interventions in italy," *Nature medicine*, vol. 26, no. 6, pp. 855–860, 2020.

[83] L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, *et al.*, "Propagation analysis and prediction of the covid-19," *Infectious Disease Modelling*, vol. 5, pp. 282–292, 2020.

[84] A. J. Kucharski, T. W. Russell, C. Diamond, Y. Liu, J. Edmunds, S. Funk, R. M. Eggo, F. Sun, M. Jit, J. D. Munday, *et al.*, "Early dynamics of transmission and control of covid-19: A mathematical modelling study," *The lancet infectious diseases*, vol. 20, no. 5, pp. 553–558, 2020.

[85] J. T. Wu, K. Leung, and G. M. Leung, "Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in wuhan, china: A modelling study," *The Lancet*, vol. 395, no. 10225, pp. 689–697, 2020.

[86] Q.-V. Pham, D. C. Nguyen, W.-J. Hwang, P. N. Pathirana, *et al.*, "Artificial intelligence (ai) and big data for coronavirus (covid-19) pandemic: A survey on the state-of-the-arts," 2020.

[87] E. Kaxiras and G. Neofotistos, "Multiple epidemic wave model of the covid-19 pandemic: Modeling study," *Journal of Medical Internet Research*, vol. 22, no. 7, e20912, 2020.

[88] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton, *et al.*, "Estimating the effects of non-pharmaceutical interventions on covid-19 in europe," *Nature*, vol. 584, no. 7820, pp. 257–261, 2020.

[89] B. Tang, W. Zhou, X. Wang, H. Wu, Y. Xiao, and S. Tang, "Controlling multiple covid-19 epidemic waves: An insight from a multi-scale model linking the behavior change dynamics to the disease transmission dynamics," *medRxiv*, 2021.

[90] T. Tat Dat, P. Frédéric, N. T. Hang, M. Jules, N. Duc Thang, C. Piffault, R. Willy, F. Susely, H. V. Lê, W. Tuschmann, *et al.*, "Epidemic dynamics via wavelet theory and machine learning with applications to covid-19," *Biology*, vol. 9, no. 12, p. 477, 2020.

[91] H. Fang, L. Wang, and Y. Yang, "Human mobility restrictions and the spread of the novel coronavirus (2019-ncov) in china," *Journal of Public Economics*, vol. 191, p. 104 272, 2020.

[92] A. I. Friedson, D. McNichols, J. J. Sabia, and D. Dave, "Did california's shelter-in-place order work? early coronavirus-related public health effects," National Bureau of Economic Research, Tech. Rep., 2020.

[93] L. Chiou and C. Tucker, "Social distancing, internet access and inequality," National Bureau of Economic Research, Tech. Rep., 2020.

[94] H. Allcott, L. Boxell, J. C. Conway, B. A. Ferguson, M. Gentzkow, and B. Goldman, "What explains temporal and geographic variation in the early us coronavirus pandemic?" National Bureau of Economic Research, Tech. Rep., 2020.

[95] X. Chen, M. Li, D. Simchi-Levi, and T. Zhao, "Allocation of covid-19 vaccines under limited supply," *Available at SSRN 3678986*, 2020.

[96] J. R. Birge, O. Candogan, and Y. Feng, "Controlling epidemic spread: Reducing economic losses with targeted closures," *University of Chicago, Becker Friedman Institute for Economics Working Paper*, no. 2020-57, 2020.

[97] D. Bertsimas, L. Boussioux, R. Cory-Wright, A. Delarue, V. Digalakis, A. Jacquillat, D. L. Kitane, G. Lukin, M. Li, L. Mingardi, *et al.*, "From predictions to prescriptions: A data-driven response to covid-19," *Health care management science*, pp. 1–20, 2021.

[98] S. Agca, J. R. Birge, Z. Wang, and J. Wu, "The impact of covid-19 on supply chain credit risk," *Available at SSRN 3639735*, 2020.

[99] X. Cao, D. Zhang, and L. Huang, "The impact of covid-19 on labor market and gender inequality: Evidence from a gig economy platform," *Available at SSRN 3666725*, 2020.

[100] M. L. Li, H. T. Bouardi, O. S. Lami, T. A. Trikalinos, N. K. Trichakis, and D. Bertsimas, "Forecasting covid-19 and analyzing the effect of government interventions," *medRxiv*, 2020.

[101] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[102] Z. Harchaoui, E. Moulines, and F. R. Bach, "Kernel change-point analysis," in *Advances in neural information processing systems*, 2009, pp. 609–616.

[103] K. Haynes, P. Fearnhead, and I. A. Eckley, "A computationally efficient nonparametric approach for changepoint detection," *Statistics and Computing*, vol. 27, no. 5, pp. 1293–1305, 2017.

[104] Y. Adulyasak, O. Benomar, A. Chaouachi, M. Cohen, and W. Khern-am-nuai, "Data analytics to detect panic buying and improve products distribution amid pandemic," *Available at SSRN 3742121*, 2020.

[105] Z. Xu, A. Elomri, L. Kerbache, and A. El Omri, "Impacts of covid-19 on global supply chains: Facts and perspectives," *IEEE Engineering Management Review*, vol. 48, no. 3, pp. 153–166, 2020.

[106] D. A. Mollenkopf, L. K. Ozanne, and H. J. Stolze, "A transformative supply chain response to covid-19," *Journal of Service Management*, 2020.

[107] E. F. Long, E. Nohdurft, and S. Spinler, "Spatial resource allocation for emerging epidemics: A comparison of greedy, myopic, and dynamic policies," *Manufacturing & Service Operations Management*, vol. 20, no. 2, pp. 181–198, 2018.

[108] A. Teytelman and R. C. Larson, "Multiregional dynamic vaccine allocation during an influenza epidemic," *Service Science*, vol. 5, no. 3, pp. 197–215, 2013.

[109] UNCTD, *Global e-commerce jumps to $26.7 trillion, covid-19 boosts online sales, \NoCaseChange{https://unctad.org/news/global-e-commerce-jumps-267-trillion-covid-19-boosts-online-sales}, accessed: 07.15.2021*, 2021.

[110] N. Matni and S. Tu, "A tutorial on concentration bounds for system identification," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 3741–3749.