Learning Treatment Policies for Empiric Antibiotic Prescription

by

Soorajnath Boominathan

S.B., Massachusetts Institute of Technology (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2020

Author . . . .
Department of Electrical Engineering and Computer Science
June 30, 2020

Certified by . . . .
David Sontag
Associate Professor
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Learning Treatment Policies for Empiric Antibiotic Prescription

by

Soorajnath Boominathan

Submitted to the Department of Electrical Engineering and Computer Science
on June 30, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Rising antibiotic resistance rates pose a serious public health threat and are largely driven by overuse and inappropriate use of antibiotics. Antibiotic stewardship efforts have been established around the world to improve prescription practices, but further optimization of antibiotic usage is still needed. Improvement is particularly necessary in the *empiric treatment setting*, the period of time immediately after a patient presents with an infection, during which clinicians must select a treatment without microbiological testing results.

In this thesis, we develop methods to learn treatment policies for empiric antibiotic prescription that are tailored to individual characteristics. We present three policy learning approaches and evaluate them in the setting of uncomplicated urinary tract infections (UTIs) using data from two Boston-area hospitals. All three approaches learn policies that significantly improve over clinicians and practice guidelines with respect to rates of inappropriate antibiotic therapy (IAT) and broad spectrum antibiotic usage, and are able to trade off between these two outcomes as desired.

We then address considerations important for deploying such learned policies as clinical decision support tools in real-world medical settings. We present techniques for learning treatment policies with the ability to defer to clinician decisions and strategies for improving the interpretability and transparency of the learned policies. We are able to successfully derive an effective, clinically intuitive treatment policy that uses fewer than 20 features. Even after accounting for several real-world treatment considerations, this policy is able to reduce rates of IAT by 20% and broad spectrum usage by nearly 50% relative to clinicians. We hope that the work presented in this thesis provides a meaningful step towards using machine learning to improve antibiotic stewardship practices in the future.

Thesis Supervisor: David Sontag
Title: Associate Professor

Acknowledgments

First and foremost, I would like to thank my thesis advisor, Prof. David Sontag, for making this MEng such an enriching experience. I am constantly amazed by his passion for research, the care he shows for his students, and the amount of time he dedicates to each and every single member of his research group. David pushed me to improve the boundaries of my abilities as a new researcher, and taught me how to ask questions that were both interesting and impactful. My only regret is that I was not able to work with him before my senior year.

I am also grateful to the Clinical Machine Learning group for being such a warm and welcoming group of people, especially to someone with very limited machine learning research experience when I joined the group. In particular, I would like to thank Mike Oberst, who has been an amazing mentor for the past two years. It's hard for me to express just how much I have learned and improved as a direct result of Mike's mentorship, from the low-level details of writing good research code to becoming a better writer and communicator to providing timely career advice. I am incredibly fortunate to have had you there to guide me through this project.

I would also like to thank Dr. Sanjat Kanjilal, who somehow found time to continue helping us drive this project forward while on the frontlines of the fight against COVID-19. We are all extraordinarily grateful for the important work you are doing during this unprecedented times.

I would also like to thank my friends who have made the last 5 years at MIT an unforgettable time. I am indebted to all of you for keeping me sane through all the ups and downs that accompany the undergrad experience at MIT. This has been the ride of a lifetime, and I am incredibly sad to see it come to an end.

Last, but certainly not least, I am forever grateful to my parents for showing me an infinite amount of support throughout my entire life. I don't know where I would be today without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the goal of this thesis: developing a clinical decision support system to help doctors select better antibiotic treatments. We first outline the antibiotic resistance problem and motivate the need for data-driven techniques that can learn more effective antibiotic prescription policies. We then highlight the challenges of deploying tools based on these techniques into real-world settings. We conclude the chapter with the contributions and organization of this thesis.

1.1 The Antibiotic Resistance Problem

The invention of antibiotics is one of the great achievements of modern medicine, but the rapid rise in pathogens exhibiting antibiotic resistance has become a major threat to global health in the 21^{st} century. Rising resistance rates have increased the difficulty of treating a wide variety of common infections, leading to higher medical costs, longer hospital stays, and most importantly, higher mortality rates. In the United States alone, antibiotic resistant infections cost the health care system more than $8 billion and lead to almost 23,000 deaths on an annual basis [11, 50].

Overuse and misuse of antibiotics are major drivers of the growth of antibiotic resistance. Studies have shown clear relationships between the volume of antibiotic exposures and incorrect antibiotic prescriptions with the development of resistant pathogens. Over 47 million unnecessary antibiotic prescriptions are made in the U.S.

annually, and a 2010 study found that the average American was prescribed roughly 22 standard units (e.g., a pill) of antibiotics in a year [11, 50]. In other countries, overuse is further encouraged by over-the-counter availability of common antibiotics.

Studies have also shown that the choice of antibiotic or treatment dosage is incorrect in 30-50% of cases [50]. These decisions have limited or no health benefit and expose patients to unnecessary negative side effects. Antibiotics are the largest cause of adverse drug reactions in patients, accounting for nearly a quarter of such events [30]. While the magnitude of these numbers are staggering, they also indicate that there are significant opportunities to improve the ways antibiotics are prescribed.

1.2 Learning Personalized Treatment Policies

As the antibiotic resistance problem has grown over the past several decades, the medical world has simultaneously been transformed by the use of vast amounts of data to improve the way patients are diagnosed and treated. The availability of patient data has been driven by the widespread adoption of electronic health records (EHRs), enabling researchers to access and analyze healthcare data at an unprecedented scale. In recent years, there has been particular interest in using machine learning techniques to learn personalized treatment policies that can improve patient health outcomes. This is particularly useful in highly heterogeneous conditions, where there is no single therapy that works uniformly across most of the population.

Prior work has developed reinforcement learning (RL) methods that uses clinical data to learn such personalized treatment regimes for the management of complex conditions such as sepsis [23]. Other work has developed models that can use individual genetic data to predict the most effective antiviral therapies to treat HIV [25]. Despite the extensive literature in this area, there has been limited work to develop methods to learn similar personalized treatment policies for prescribing antibiotics. We next motivate the need for such techniques in this domain and outline the specific problem that we will tackle in this work.

1.3 Improving Antibiotic Stewardship

The World Health Organization (WHO) has outlined a global action plan for combating antibiotic resistance with 5 goals, one of which is optimizing the usage of antibiotics, also known as 'antibiotic stewardship' [39]. The Centers for Disease Control (CDC) indicate that improved antibiotic stewardship programs can improve patient outcomes, reduce antibiotic resistance rates, and decrease health care costs [11].

In this work, we aim to improve antibiotic stewardship by developing decision algorithms that use patient data from the EHR to help doctors make prescription decisions that reduce inappropriate or unnecessary treatment. We are specifically interested in improving decisions made in the empiric treatment setting, the period immediately after a patient presents with an infection. During this time, doctors do not have access to microbiological test results indicating the effectiveness of various antibiotics for an infection; completing these tests typically take a couple days. They instead use patient health records and personal clinical experience to make this decision. This initial treatment decision often has a significant effect on patient outcomes, so it is crucial to make an effective choice here whenever possible [22].

Doctors face a difficult trade-off when making empiric treatment decisions. On one hand, the selected antibiotic should be effective against the infecting pathogen. If one only cares about this objective, doctors should treat patients with broad spectrum (2nd-line) agents, powerful antibiotics that are generally more effective and minimize risk of treatment failure. However, overuse of broad spectrum antibiotics has several negative consequences. They are associated with adverse patient-level side effects, such as secondary *C. dificile* infections and aortic dissections [3, 8]. Overuse also leads to rising resistance rates against these agents, reducing their future effectiveness in the population. As a result, reducing usage of broad spectrum agents is a major goal of many antibiotic stewardship programs, and clinicians should aim to treat patients with effective narrow spectrum (1st-line) treatments as much as possible.

Despite this goal, clinicians still frequently use broad spectrum agents in the empiric treatment setting; for some infections, these are the most frequently used class of treatments [19]. There is currently an unmet need for data-driven approaches that can identify effective narrow spectrum antibiotic treatments for patients. The work presented in this thesis aims to fill this gap by developing machine learning-based methods for learning treatment policies that are able to recommend effective antibiotics while minimizing the usage of broad spectrum antibiotics. If integrated into clinical workflows, such algorithms can have significant implications for improving antibiotic stewardship practices. However, as we discuss next, successfully deploying such tools into hospitals and other healthcare settings is a challenge in and of itself.

1.4 Deploying Clinical Decision Support Tools

Research into machine learning for healthcare-related applications has exploded in recent years, but these advances have not always translated into successful deployment of tools to support clinicians in real-world settings. Recent efforts to comprehensively survey the impact of deployed ML-based clinical decision support tools across various medical fields have produced conflicting results, with some indicating a positive impact on successful clinical diagnoses, and others suggesting that they have essentially no effect [47]. Other studies also show that decision support tools can suffer from subpar user interfaces and excessive alerts, leading to 'alert fatigue' in clinicians and further diminishing any potential impact these tools could have [34, 52]. It is clear that state-of-the-art performance on curated research datasets do not immediately translate into meaningful impact for patients and doctors.

In this thesis, we develop our methods for learning treatment policies with an emphasis on incorporating real-world considerations that would affect ease of deployment and integration into clinical settings. We aim to develop models for empiric antibiotic prescription that are interpretable, portable, and transparent, enabling doctors to make informed evaluations of the algorithm's recommendations and encouraging widespread adoption of the resulting treatment policies. We also consider the presence of the clinician in the decision-making process and develop methods for learning treatment policies with the ability to defer to clinician decisions, helping us overcome problems such as alert fatigue or lack of confidence in the support tool's decisions. Accounting for these factors will enable us to more easily deploy the methods developed in this work as a clinical decision support tool that can have a meaningful impact on antibiotic usage practices.

1.5 Contributions

Our primary contributions in this thesis are as follows:

1. **Effective policy learning methods for antibiotic prescription.** We formalize the task of prescribing antibiotics while balancing rates of effective therapy and broad spectrum usage as a policy learning problem in a setting with fully observed outcomes and multiple objectives. We present three methods for learning sets of policies in these settings that are optimal for a range of trade-offs between the multiple objectives, and highlight pros and cons of each approach. Using a dataset of patients with uncomplicated urinary tract infections (UTIs), we show that all our methods learn effective policies for antibiotic selection that exceed the performance of clinicians, practice guidelines, and previous baselines with respect to rates of ineffective therapy and broad spectrum usage while achieving a wide range of trade-offs between these objectives.

2. **Design for a clinical decision support system for empiric antibiotic prescription.** We address several considerations necessary for the construction of a deployed clinical decision support (CDS) tool for antibiotic prescription. We first address the problem of incorporating an option to defer to clinician decisions. We present and formalize the problem of learning to defer in settings where algorithm errors or interventions are costly, and extend our policy learning methods to obtain policies that defer on appropriate examples for each of these settings. We then outline strategies to make the proposed tool more interpretable, portable, and transparent to clinicians. We derive a simple, but highly effective, treatment policy that uses fewer than 20 features from the EHR

and is straightforward to deploy. We also identify two examples of situations where a CDS tool may make serious mistakes - treatment contraindications and an incomplete patient record in the EHR - and propose strategies for addressing these issues in a transparent manner in the system's design. Retrospective evaluation of our proposed CDS tool shows that it is able to reduce ineffective therapy rates by 20% and broad spectrum usage rates by nearly 50% relative to clinicians. It also reduces ineffective therapy rates by nearly 10% relative to a modified version of practice guidelines.

1.6 Thesis Organization

This thesis is organized as follows. In Chapter 2, we provide further background on the antibiotic treatment problem and survey related work that addresses this specific problem, as well as work related to our methods in the broader machine learning and biostatistics communities. In Chapter 3, we provide an overview of the data used in this work and define the patient cohort to be used for training and evaluating our methods on a real-world dataset. In Chapter 4, we present our policy learning approaches and evaluate their performance using synthetic data. In Chapter 5, we present the results of applying our policy learning methods to the real-world dataset.

In the last two chapters, we address considerations necessary for successful deployment of our treatment policies into clinical workflows. In Chapter 6, we examine approaches for extending our policy learning methods to accommodate deferral to clinician decisions. In Chapter 7, we outline the design of a deployable CDS tool for antibiotic prescription that relies on a simple, highly interpretable treatment policy and address potential limitations of this tool in real medical settings. In Chapter 8, we conclude with some additional discussion of our findings and steps for future work.

Chapter 2

Background and Related Work

In this chapter, we provide further background on the development of antibiotic resistance, the process of antibiotic prescription, and treatments for urinary tract infections (UTIs), the class of infections addressed in detail in this thesis. Finally, we survey related work that has previously tackled the problem of antibiotic prescription using machine learning techniques, along with work related to the methods we develop for learning treatment policies.

2.1 Background

2.1.1 Antibiotic Resistance and Testing

Antibiotic resistance occurs when bacteria evolve to develop resistance to antimicrobial agents to which they were previously susceptible. Resistance is driven by the high plasticity of bacterial genomes, which allows them to quickly mutate in response to changing environments, with the 'fittest' organisms surviving to pass on resistance to future generations. There are two major genetic mechanisms associated with the development of resistance: mutational resistance and horizontal gene transfer [32].

Mutational resistance occurs when a bacterial cell develop mutations in genes that affect antibiotic activity in some way. The antibiotic destroys the remaining susceptible members of the population, selecting for those with the resistant mutation.

Overuse of antibiotics accelerates this process and increases the proportion of resistant pathogens in the population. Mutations can lead to resistance in several ways, such as decreasing the drug's affinity for the bacteria, producing enzymes that modify the antibiotic molecule to render it ineffective, or the activation of pathways that remove the antibiotic from the bacterial cell [32]. Horizontal gene transfer (HGT) occurs when a bacteria acquires foreign DNA from bacteria of either the same or different species, potentially introducing genes conferring resistance to that organism [32].

Clinicians identify antibiotic resistance in an infecting pathogen using microbiological testing. After extracting a bacterial specimen from the patient's infection site, tests generally involves growing a culture of that specimen and examining its growth in the presence of different concentrations of an antibiotic. Specimens requiring a sufficiently high concentration of antibiotic to be killed are considered to be resistant to that agent. Two metrics are commonly used to quantify the activity of an agent against a pathogen: minimum inhibitory concentration (MIC) and disk diameter (DD). MIC measures the lowest concentration of antibiotic required to kill an antibiotic, while DD corresponds to the diameter of dead bacteria when an antibiotic is placed in the center of a plated culture [26]. Pre-determined breakpoints for these metrics are used to classify specimens into susceptible, intermediate, and resistant categories [6]. These testing procedures typically take a couple of days, and the results define a patient's antibiogram.

2.1.2 Antibiotic Prescriptions

When a patient first presents in the hospital with an infection, clinicians diagnose the patient by identifying the infection site and generating hypotheses about the infecting pathogen. Doctors then request microbiological testing to identify the pathogen and the patient's antibiogram, but this process can take anywhere between 24 to 72 hours [24]. A doctor typically has to make an antibiotic prescription prior to receiving this information; treatment selections made during this time are referred to as **empiric prescriptions**. These empiric treatment decisions are crucial, as failed therapies at this stage are associated with longer hospital stays and higher mortality rates

[22]. Treatment with an antibiotic to which the patient is resistant is also known as **inappropriate antibiotic therapy (IAT)**.

Empiric prescriptions are primarily guided by the patient's clinical presentation. There are numerous factors that drive this decision, including the infection site, any prior bacterial infections in this patient, and local resistance rates and antibiograms at the hospital. Doctor generally also account for patient characteristics, including age, special health conditions (e.g, pregnancy), and prior antibiotic exposures [24]. Doctors are forced to make several trade-offs when selecting an antibiotic therapy. In this work, we focus on the trade-off between effective therapy and antibiotic spectrum, which we highlighted in the introduction. In practice, however, there are other important variables, including treatment cost and the likelihood of patient adherence to a treatment regimen.

Once information about the infecting pathogen and the patient's antibiogram does become available, clinicians aim to *narrow the spectrum* of the selected antibiotic treatment as much as possible based on this data. As described in the introduction, unnecessary use of broad spectrum agents can lead to negative side effects and increase population-level resistance rates.

Selecting an empiric prescription is an extremely complex decision, and it is nearly impossible for doctors to incorporate the multitude of relevant factors when coming to a treatment decision. Numerous health organizations, such as the Infectious Diseases Society of America (IDSA), have published guidelines to assist clinicians in this process [15]. While these are certainly a useful tool, they are not a uniform solution to this difficult problem and are not a replacement for a doctor's clinical judgment.

2.1.3 Urinary Tract Infections (UTIs)

In this work, we focus primarily on empiric prescriptions for **urinary tract infections (UTIs)**. A UTI is a bacterial infection in any part of an individual's urinary system, and are among the most common infections in humans. In the U.S. alone, UTIs account for over 13 million annual ER and outpatient hospital visits, and over 4.7 million antibiotic prescriptions [19, 36]. UTIs are generally more common in women than men, and are especially dangerous when the infecting pathogen is resistant to antibiotic treatments.

We are particularly interested in **uncomplicated UTIs**, which refer to infections in the structurally normal lower urinary tract of otherwise healthy females. Nearly half of women have an uncomplicated UTI at some point in their life, and nearly a quarter will face recurrent infections [29]. *E. coli* is the mostly common pathogen in this infection, accounting for between 75-95% of infections [29]. We choose to focus on this condition because it is generally treated with a small, well-defined set of antibiotics, making comparisons to clinician and guideline decisions tractable and enabling evaluation of the clinical impact of our learned treatment policies. There are four commonly used empiric therapies to treat uncomplicated UTIs: nitrofurantoin (NIT), trimethoprim-sulfamethoxazole (SXT), ciprofloxacin (CIP), and levofloxacin (LVX). NIT and SXT are narrow spectrum (1st-line) agents, while CIP and LVX are broad spectrum (2nd-line) agents. We use these abbreviations throughout the remainder of this thesis.

Current prescription guidelines generally recommend using SXT as a 1st-line therapy unless its resistance rates exceed 10-20% in a given location [18]. Broad spectrum agents are intended to be a last-resort option in communities with high resistance rates to 1st-line therapies or for patients who cannot tolerate those therapies (e.g, due to allergies). Despite these guidelines targeting reduced usage of broad spectrum agents, these agents are the most common class used to treat uncomplicated UTIs, accounting for over 40% of prescriptions in some cases [19]. The excessive use of broad spectrum agents is likely driven by increasing resistance rates to 1st-line therapies, particularly SXT, pushing doctors to use more powerful agents to lower the risk of treatment failure. These troubling trends highlight a clear need for further optimization of antibiotic prescriptions for this class of infection, encouraging reduction of broad spectrum treatments while still maintaining or reducing current rates of IAT.

2.2 Related Work

In this section, we discuss prior work relevant to the policy learning methods we develop in this thesis, as well as recent work in the specific area of learning treatment policies for antibiotic prescription.

2.2.1 Learning Individualized Treatment Rules (ITRs)

The problem of learning individualized treatment rules (ITRs) - treatment policies tailored to individual characteristics - has been studied extensively in the biostatistics community. There are two broad classes of methods for this problem: *direct* approaches and *indirect* approaches. Indirect methods first learn models of the conditional distribution of treatment outcomes given patient characteristics for each treatment of interest. The selected treatment for a given patient is then the one that maximizes the estimated outcome. Approaches such as Q-learning, A-learning, and regret regression fall into this group [33, 43, 16].

A limitation of this type of approach is that the model class of the resulting ITR is dependent on the model class used for the conditional outcomes. If a linear ITR is desired, then it necessitates the use of a linear model class for the conditional outcome models. In cases where a simple outcome model does not accurately capture the true conditional outcome function, the learned ITR may be sub-optimal [31].

By contrast, *direct* approaches sidestep the problem of learning conditional outcome models. Instead, they construct an estimator of the expected outcome as a function of the *decision rule*, and optimize this estimator by searching directly over the space of allowed decision rules. This removes the dependence between the complexity of the learned ITR and the outcome models to avoid misspecification. For problems with binary treatments, this direct optimization problem is equivalent to weighted binary classification, and is referred to as outcome weighted learning [58]. Recent work has also focused on developing and analyzing convex surrogates for this problem to allow for efficient optimization of the objective and extend this to the multi-action setting [17, 57].

2.2.2 Cost-Sensitive Learning

Most work on learning ITRs assumes that one only has access to observational data that provides a single treatment outcome per individual. In the antibiotic prescription setting, however, the results of resistance testing for all antibiotics provide us with ground-truth information about the *counterfactual* outcomes of all possible treatments of interest. Thus, our setting is also closely related to cost-sensitive multi-class classification, which considers a loss function that is the multi-class equivalent of a weighted 0-1 loss for binary classification [10]. Prior work has proposed smooth convex surrogates for optimizing this objective, and shown that they are consistent for the Bayes-optimal classifier [48, 56, 59]. In this work, we develop a convex surrogate similar to one recently developed for learning ITRs and prove its consistency for the Bayes-optimal policy [17]. The cost-sensitive classification literature is primarily focused on learning *deterministic* policies, and that is what we study here as well. Deterministic policies are generally optimal for cases where exploration is not required, and stochastic policies can introduce further optimization difficulties [53].

2.2.3 Multi-Objective Policy Optimization

Our policy learning problem is centered around the trade-off between treatment effectiveness and broad spectrum antibiotic usage, so it is an inherently multi-objective setting. Multi-objective settings have been studied in the context of both single decisions and sequential decision-making [42, 51, 55]. In the sequential setting, these problems are generally formalized as a Markov Decision Process (MDP) with a vector-valued objective - where each dimension contains the 'reward' for one of the objectives - and a scalarization function that maps the vector objective to a scalar reward. This function is typically assumed to possess some simple structure (e.g., a linear combination of the vector objective), but is unknown at training time [35]. Methods for multi-objective policy learning maintain a set of policies that are optimal for different linear scalarization functions, and can be seen as indirect methods in their use of Q-functions to do so [1, 27, 35]. The set of objective values achieved by the set of optimal policies - each optimal for a different possible scalarization function - is known as the *Pareto frontier* [53].

2.2.4 Improving Empiric Antibiotic Prescription

Modeling patient resistance to antibiotics is an obvious first step towards learning an effective antibiotic treatment policy. The use of machine learning techniques to predict antibiotic resistance from patient EHR data has gained popularity in recent years. Prior work has used simple classification models, such as logistic regression or decision trees, to predict resistance to specific antibiotics in small patient cohorts [46, 49]. However, this body of work generally focuses on only the *prediction* problem and interpretation of features predictive of resistance, and does not use these predictive models to derive prescription policies that can be applied to improve antibiotic stewardship in practice.

There has been some recent progress towards the policy learning problem in this setting. Yelin et al. proposed two indirect approaches for learning improved antibiotic treatment policies for UTIs [54]. Both approaches involve first learning models of treatment effectiveness for each antibiotic. Their unconstrained approach then selects the antibiotic with the highest predicted effectiveness, while the constrained approach constructs a policy based on these predictions that is also forced to match the treatment distribution used by clinicians. However, they do not directly address how their policies affect usage of broad spectrum treatments, a crucial consideration for practical deployment and adoption of empiric prescription policies.

Another recent work developed a method for predicting treatment resistance in a way that accounts for the multi-objective nature of the empiric antibiotic prescription problem, selecting a threshold to binarize the predictions of a resistance model by optimizing a utility-based objective that incorporates factors such as treatment cost and the cost of a negative patient outcome [38]. However, their analysis was limited to predictive performance on a small cohort of bloodstream infections. They do not address how these predictions could be integrated into a decision tool for empiric prescription or examine impact on clinical outcomes.

In this thesis, we develop policy learning methods for antibiotic prescription with a central focus on the trade-off between IAT rates and usage of 2nd-line treatments, and learn *sets* of treatment policies expressing a wide range of trade-offs between these two objectives. We also develop *direct* methods for policy learning that do not require learning models of resistance for each antibiotic of interest beforehand. Neither of these points have been addressed in prior work in this area.

Chapter 3

Dataset and Cohort Selection

In this chapter, we provide an overview of the dataset used for training and evaluating the policy learning approaches presented in this work. We also define the patient cohort to be used for all our real-world experiments, along with a description of features used in our models.

3.1 Dataset Overview

The dataset used in this work consists of data from the electronic health record (EHR) for over 200,000 patients who submitted a specimen for microbiological testing at either Massachusetts General Hospital (MGH) or Brigham & Women's Hospital (BWH) between 2000 and 2016. The dataset contains both traditional structured data about patients (e.g, demographics, medications, etc.) and unstructured data (e.g, clinical notes). This study was approved by the Institutional Review Board of Massachusetts General Hospital with a waived requirement for informed consent.

3.1.1 Structured Data

We list the categories of patient-level information found in the EHR, highlighting information later used as model inputs.

- **Demographics**: The EHR contains a person's birth date, gender, race, and veteran status. The recorded race is a binary variable, so we are only able to identify whether or not patients are white.

- **Location**: The EHR records the specific location (e.g., clinic/facility name) at which each microbiological specimen was collected, as well as whether the patient was treated as an inpatient, outpatient, or in the ER/ICU.
- **Medications**: The EHR contains a record of any prescriptions given to a patient, including the name of the medication, the prescribed dosage, and the date of prescription.
- **Procedures / Diagnoses**: The EHR records all procedures conducted on or diagnoses given to a patient. This includes a human-readable name, date, and standardized billing code for the procedure/diagnosis.
- **Patient encounters**: Each patient ‘encounter’ refers to a single inpatient or outpatient visit to the hospital. The data contains the patient’s admission and discharge date for each visit, along with the location that they were admitted from and discharged to.
- **Labs**: The EHR contains names, dates, and raw numeric results of any lab tests ordered for a patient by physicians. Examples of recorded lab tests include white blood cell (WBC), lymphocyte, and neutrophil counts.
- **Microbiological testing data**: Our dataset contains microbiological testing results for all specimens sent to the labs at MGH and BWH. This includes the identity of the infecting pathogen and susceptibility testing to various antibiotics. The data contains the metric used for each test (MIC vs. DD) and the numeric value of the corresponding test result, as well as the date and location of specimen collection. We transform these numeric results into categorical phenotypes by applying the published 2017 CLSI breakpoints, which convert the results into one of three phenotypes: susceptible (S), intermediate (I), and resistant (R) [6]. We treat both intermediate and resistant phenotypes as resistant, which is generally how they are treated in clinical practice.

3.1.2 Text Data

The available unstructured data primarily consists of clinical notes. Across the full dataset from 2000-2016, we have over 22 million available notes. The notes come from 4 sources: (1) discharge summaries, (2) outpatient notes, (3) inpatient visit notes, and (4) legal medical record (LMR) notes. As the vast majority of patients in our dataset were treated in outpatient settings, outpatient notes are by far the most common source of text data.

The notes vary significantly in their formatting and structure. While there are several notes that share the same general writing structure and section patterns, it is extremely difficult to design an automated way of processing these notes into a more structured form (e.g., dividing into specific sections). Information relevant to antibiotic prescription is also extremely sparse in these notes. In Chapter 7, we discuss methods for extracting useful information from these notes in a systematic manner to supplement the data in the EHR.

3.2 Feature Construction

We use the available EHR data to construct features used as inputs for our treatment policies. While the data is recorded at a *patient* level, we wish to construct features at the *specimen* level, as we make separate treatment decisions for each specimen from a patient. We must also avoid constructing features containing information unavailable to clinicians at the time of an empiric treatment decision. For instance, while the microbiological data contains information about the infecting pathogen, clinicians would not have this information when selecting an empiric prescription.

Some pieces of data can be used directly as features without further processing, such as patient demographic information (e.g., age, race). However, the vast majority of features are computed as **windowed features**, which we describe in the next section. We then define two population-level features computed by aggregating patient-level data and describe the process for constructing them.

Feature	**Lower Bound**	**Backward Windows**
Prior antibiotic exposures	2	7, 14, 30, 90, 180, All
Prior resistance	7	14, 30, 90, 180, All
Prior infecting organisms	7	14, 30, 90, 180
Comorbidities	0	7, 14, 30, 90, 180
Procedures	0	7, 14, 30, 90, 180
Labs	0	7, 14, 30, 90, 180
Encounters	0	7, 14, 30, 90, 180

Table 3.1: List of all *windowed* features computed for each specimen. All numerical window lengths are in days. 'All' refers to summarizing a feature over *all* available patient history without a bound on the backward window.

3.2.1 Windowed Features

Most specimen features are computed by aggregating the corresponding patient's data within a specified backward time window from the specimen collection date; we refer to such features as *windowed features*. For example, we construct binary indicator features indicating whether a patient received a specific antibiotic in the 90 days prior to specimen collection. We compute this same feature for multiple backward window lengths from the specimen date. We also enforce a *lower bound* on the time window for some of these features, as we do not want to incorporate any data that would not be available to the doctor when making the empiric decision. For instance, we only consider antibiotic prescriptions made at least 2 days prior to the date of specimen collection when constructing features. Any prescriptions made within 2 days are considered to be an empiric prescription targeted against that specimen.

Table 3.1 contains a full list of these features and the time windows over which they are aggregated. All features except labs and encounters are binary indicators: 1 if the event corresponding to the feature occurred in the specified time window and 0 otherwise. Lab test features are averaged across all available test results for an individual in the specified backward window, and the number of encounters for a patient are counted up within the specified window.

3.2.2 Population-Level Features

Colonization Pressure

We define the **colonization pressure** of an antibiotic as the rate of resistance to that agent within a specified location and time period. In our work, we compute the colonization pressure for a given specimen as the proportion of specimens resistant to an antibiotic in the period ranging from 7 days before to 90 days before the date of specimen collection. We compute colonization pressure at three location hierarchies:

- **Floor-level:** resistance rate across specimens collected at the same floor/ward/clinic.
- **Hospital-level:** resistance rate for specimens collected at the same hospital and service category (e.g, MGH outpatient settings). There are two hospitals (MGH, BWH) and four service categories (outpatient, inpatient, ICU, and ER).
- **Overall:** resistance rate across all specimens in the dataset.

In total, we compute the colonization pressure for 25 antibiotics across all urinary tract specimens.

Total Antibiotic Usage

We also compute the hospital system-wide usage of several antibiotics in a specified time window prior to specimen collection. We calculate the total number of prescriptions of each antibiotic in the 90 days preceding specimen collection, then normalize this value by the total volume of inpatient and outpatient specimens during that time.

3.3 Uncomplicated UTI Cohort

We now define the uncomplicated UTI cohort used for the majority of experiments in this thesis. We first excluded all microbiological specimens collected prior to 2007, as there were shifts in antibiotic resistance testing methodologies in 2007 that rendered test results collected after this time incomparable to those collected before. We then limited our cohort to urinary tract specimens collected from non-pregnant women between the ages of 18-55 who had not undergone a surgical procedure or had catheters in the 90 days preceding specimen collection.[1] We also excluded any patients with a pyelonephritis diagnosis in the last 90 days. Patients who do not satisfy all of these criteria are generally considered to have 'complicated' UTIs. The remaining set of specimens contains all patients with uncomplicated UTIs.

Further filtering of the cohort was required to enable evaluation of our policy learning methods. We filtered to only specimens that underwent susceptibility testing to all 4 antibiotics considered in this work: NIT, SXT, CIP, and LVX. Without this information, we cannot determine whether a particular treatment would have been effective for a patient. We also limited our cohort to patients treated with exactly one of these four agents within the empiric treatment period, which we define as the period spanning 2 days prior to 1 day after the recorded date of specimen collection.

Finally, we separated our cohort into datasets used for training and evaluation of our policy learning methods. The training set consists of specimens collected from 2007-13; the test set contains specimens collected from 2014-16. To avoid any data leakage between training and test sets, we remove any specimens in the test set collected from patients who also had specimens in the training set. In total, our dataset consists of 15,806 specimens collected from 13,682 unique patients with UTIs between 2007 and 2016. Figure 3-1 provides a detailed overview of the filtering process used to derive the uncomplicated UTI cohort.

Table A.1 contains a summary of basic cohort statistics for the training and test sets. Resistance rates to 1st-line agents have generally remained steady, while resistance rates to 2nd-line agents have increased over time. This rise in resistance rates reaffirms the need to develop effective treatment policies that use fewer 2nd-line antibiotics. Clinicians have reduced the rate at which they prescribe 2nd-line agents over time, moving towards increased usage of NIT instead, but an even more dramatic shift is necessary to prevent the continued rise in broad spectrum resistance rates.

[1] We note that multiple specimens may be collected from a patient for the same infection. We treat specimens collected from the same individual within 14 days of one another as being from the same infection, and only keep the first specimen in such a sequence.

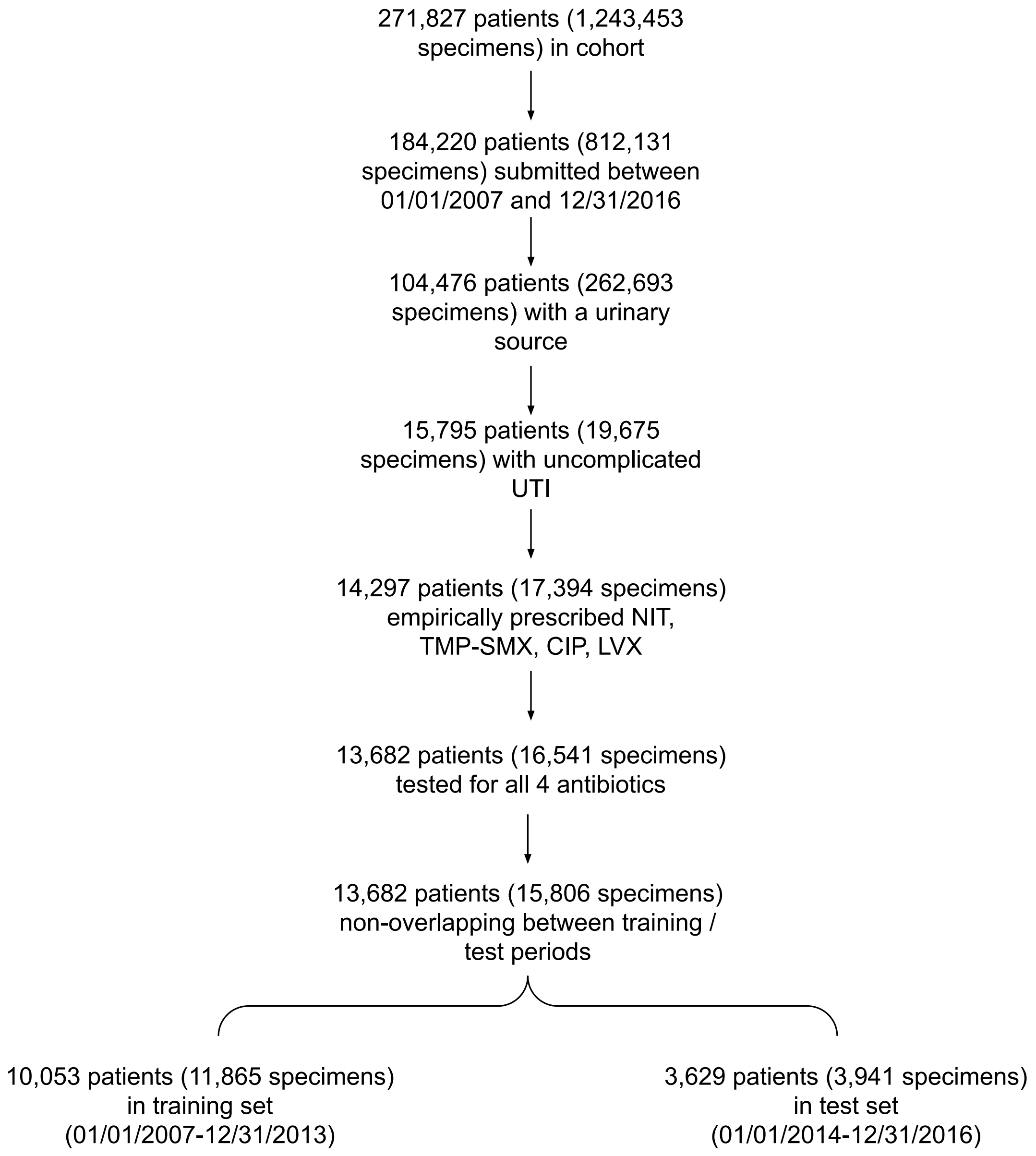

271,827 patients (1,243,453 specimens) in cohort
184,220 patients (812,131 specimens) submitted between 01/01/2007 and 12/31/2016
104,476 patients (262,693 specimens) with a urinary source
15,795 patients (19,675 specimens) with uncomplicated UTI
14,297 patients (17,394 specimens) empirically prescribed NIT, TMP-SMX, CIP, LVX
13,682 patients (16,541 specimens) tested for all 4 antibiotics
13,682 patients (15,806 specimens) non-overlapping between training / test periods
10,053 patients (11,865 specimens) in training set (01/01/2007-12/31/2013)
3,629 patients (3,941 specimens) in test set (01/01/2014-12/31/2016)

Figure 3-1: Filtering criteria for construction of the uncomplicated UTI cohort. Uncomplicated UTI is precisely defined as a UTI occurring in a non-pregnant women between 18 and 55 with no record of surgery, immunosuppression, indwelling catheters, or neurologic dysfunction in the 90 days prior to specimen collection.

Chapter 4

Indirect and Direct Methods for Policy Learning

In this chapter, we formalize our policy learning problem and identify unique characteristics of our setting that inform the development of our methods. We present three different methods for tackling this problem: two indirect approaches and one direct approach, and discuss trade-offs between these approaches. We conclude this chapter by highlighting the advantages of the direct method over indirect approaches in a synthetic environment.

4.1 Problem Overview

Our goal is to learn treatment policies that map specimen features to an antibiotic prescription. Most methods for learning such individualized treatment policies are designed for learning from *observational data*, in which one only observes the outcome of a single treatment for a patient e.g., the one that was actually given. In these cases, policy learning methods must develop estimators for the unobserved outcomes for a patient to derive good treatment policies. However, our setting is unique, as the results of antibiotic susceptibility testing in our data provides us with strong proxies for the counterfactual outcomes for a given specimen under all possible treatment options. Thus, we are learning in a setting with **fully observed outcomes**.

Our policy learning problem is also **multi-objective** in nature. As discussed previously, clinicians selecting an empiric treatment must make a trade-off between the *effectiveness* of a given antibiotic therapy (i.e, the likelihood of a treatment resulting in IAT) and the rate of *broad spectrum* antibiotic usage. The optimal trade-off between objectives may vary across locations or types of patients. Thus, instead of learning a single, fixed policy, we wish to learn a *set* of policies exhibiting different trade-offs between two objectives, and return this set to users of the model. Once the models are deployed, practitioners have the responsibility of selecting the policy and corresponding trade-off which makes the most sense in their particular context.

4.2 Problem Formalization

We first provide a general formalization of the multi-objective policy learning problem. We let $\mathcal{A} = [K]$ denote the action space, where K is the number of discrete actions, and denote features as $\mathbf{X} \in \mathbb{R}^m$.[1] Our goal is to learn a deterministic policy $\pi : \mathbb{R}^m \rightarrow \mathcal{A}$ which maps from features (i.e., patient characteristics) to a recommended action.

We are focused on optimizing between two objectives in our setting: treatment effectiveness and broad spectrum usage. The following formalization is specific to the two-objective setting, but the extension to incorporate additional objectives is straightforward. Our dataset is of the form $\{(\mathbf{X}_i, \mathbf{Y}_i, \mathbf{C}_i)\}_{i=1}^n$, where $\mathbf{X}_i \in \mathbb{R}^m$ are patient features, and $\mathbf{Y}_i, \mathbf{C}_i$ represent our competing objectives, a *benefit* and a *cost* respectively. In the antibiotic prescription setting, $\mathbf{Y}_i \in \{0, 1\}^K$, where $Y(a)$ is an indicator for whether antibiotic a is effective in treating an infection, and $\mathbf{C}_i \in \{0, 1\}^K$, where $C(a)$ is an indicator for whether the chosen antibiotic is broad spectrum.

We present three approaches in this chapter. Each approach will return a **set** of policies Π, where each element $\pi \in \Pi$ represents an optimal policy for some trade-off between Y and C. The first two methods are *indirect* approaches, requiring us to

[1] Notation: Bold-faced symbols (e.g., $\mathbf{X}$) denote vectors, and $X(i)$ denotes the i-th entry of a vector $\mathbf{X}$.

learn a separate model for the conditional mean of Y under each treatment,[2] denoted $f_a(x) \approx \mathbb{E}(\mathbf{Y}(a) \mid \mathbf{X} = x)$. The third approach is a *direct* approach - it does not not require learning predictive models for individual outcomes, but instead optimizes directly for a treatment policy. We provide a brief overview of these three approaches:

1. **Thresholding**: After learning treatment outcome models, convert each $f_a(x)$ into a binary prediction of treatment effectiveness $\hat{Y}(a)$ using carefully chosen thresholds. The policy chooses the treatment with lowest cost that is also predicted to be effective.

2. **Expected Reward Maximization**: We combine the two objectives $\mathbf{C}$ and $\mathbf{Y}$ into a single objective r_ω (a 'reward'), where ω parameterizes a linear trade-off between the two objectives. The policy selects the treatment that maximizes the expected value of r_ω under $f_a(x)$.

3. **Direct Policy Optimization**: We construct a surrogate loss function for the value of a treatment policy using the notion of 'reward' defined in approach (2). We optimize this objective to learn a model that directly maps from input features to a treatment decision.

4.3 Thresholding

The decision logic underlying the thresholding method is intuitive: we predict which treatments will be effective, and then choose the effective treatment with the lowest cost. Given our learned models $f_a(x)$ of predicted effectiveness, which output values between 0 and 1, we apply carefully-chosen thresholds to make a binary prediction of effectiveness $\hat{Y}(a)$ for each treatment. We combine these predictions with the fixed cost associated with broad spectrum antibiotics to choose the lowest-cost treatment that is still predicted to be effective.

[2] In our case, costs are determined by the choice of treatment itself, so we only need to model the conditional mean of treatment effectiveness, but it is straightforward to extend both methods to the case where all objectives must be modeled.

Formally, let the set of thresholds used to binarize each prediction be denoted by $\{t_a\}_{a=1}^K$; there is one threshold per action. We let $\hat{Y}_a(x)$ be an indicator[3] that represents whether treatment a is predicted to be effective for patient with features x, given by

$$\hat{Y}_a(x) = \mathbf{1}\left[f_a(x) \geq t_a\right]. \tag{4.1}$$

The treatment policy is then defined as the action that minimizes cost, among the treatments that are predicted to be effective:

$$\pi(x) = \arg\min_a \{C(a) \mid \hat{Y}_a(x) = 1\}. \tag{4.2}$$

If $\hat{Y}_a(x) = 0$ for all $a \in \mathcal{A}$, the treatment policy falls back to an option a that minimizes the cost $C(a)$. In the context of antibiotic prescription, this corresponds to defaulting to a first-line antibiotic.

We now discuss the process used to select the thresholds $\{t_a\}_{a=1}^K$. Recall that our goal is to learn a set of policies $\Pi = \{\pi_j\}_{j=1}^J$, where each $\pi_j \in \Pi$ expresses a different trade-off between the objectives of effectiveness and cost. We control this trade-off with a set of *cost constraints* $\{b_j\}_{j=1}^J$, where b_j represents the maximum cost that can be incurred by policy π_j.

Each threshold combination implicitly defines a policy with a fixed trade-off between effectiveness and other costs. To learn each policy π_j, we perform a brute force search over different choices of threshold combinations $\{t_a\}_{a=1}^K \in \mathcal{T}$, where $\mathcal{T}$ is a large (but finite) search space. For each threshold combination, we evaluate the corresponding policies on a validation set to obtain empirical estimates of its benefit $\mathbb{E}[Y(\pi(x))]$ and cost $\mathbb{E}[C(\pi(x))]$. π_j is then given by

$$\pi_j = \arg\max_\pi \{\mathbb{E}[Y(\pi(x))] : \mathbb{E}[C(\pi(x))] \leq b_j\}. \tag{4.3}$$

In words, we choose the policy for each b_j which achieves the highest mean value of Y in our validation set, subject to the constraint that the mean cost is less than b_j.

[3] Throughout, we use $\mathbf{1}[P]$ as an indicator function that is equal to 1 if the expression P is true, and 0 if P is false

While the logic behind this approach is highly interpretable, it has significant drawbacks. First, it is computationally intensive: it requires us to perform a brute force search over a large set of thresholds $\mathcal{T}$, and there are no guarantees that our finite search space will contain an optimal threshold combination. In addition, information is lost when thresholding predicted probabilities. Two treatments with equal cost may both be predicted to be effective after thresholding, but the underlying outcome models may be far more confident in the effectiveness of one treatment. We cannot recover this information after thresholding. In the next section, we present a method that circumvents these issues, while making the trade-off that the resulting decision logic (maximizing an expected reward) may be less interpretable to a lay audience.

4.4 Expected Reward Maximization

We motivate this approach by first considering the single-objective setting where we are only concerned with treatment effectiveness. In this case, the simplest approach for converting predictions of treatment effectiveness into a policy for a patient with features x would be to select the treatment a^* with the highest likelihood of effectiveness (i.e, $a^* = \arg\max_a f_a(x)$). We can extend this approach to the multi-objective setting by first *scalarizing* the values of the multiple objectives into a single value representing a notion of reward, then selecting the action that maximizes this quantity.

More formally, recall that our goal is to learn a deterministic treatment policy $\pi : \mathbb{R}^m \to \mathcal{A}$, which maps patient features to a deterministic decision. We combine our objectives with a linear preference parameter $\omega \in [0, 1]$ such that the reward is linear combination of our competing objectives. In our particular case, we parameterize this as follows, to account for the fact the our cost is a binary variable

$$\mathbf{r}_\omega = \omega \cdot \mathbf{Y} + (1 - \omega) \cdot (1 - \mathbf{C}), \tag{4.4}$$

where $\mathbf{r}_\omega(a)$ represents the reward under treatment a. We will omit the subscript where it is clear from context.

In the setting of linear preferences, commonly used in the multi-objective optimization literature [35, 45] we do not lose anything by restricting ourselves to deterministic policies, because there exists an optimal policy that is deterministic [42]. For a given preference ω, we can define the Bayes-optimal policy π^*_ω as the one that maximizes the expected reward for a given x. Formally, it is given by:

$$\pi^*_\omega(x) = \arg\max_{a \in A} \mathbb{E}[r_\omega(a) \mid x], \tag{4.5}$$

where the maximization is performed over the true (unknown) conditional expectations. In this work, we learn a *set* of policies Π that are each optimal according to some preference ω, allowing users to select a policy from this set which corresponds to their desired trade-off. This is referred to as the "decision support" setting - instead of obtaining an explicit preference ω from decision makers, we provide a set of alternatives that are each optimal for some ω [42].

Using this formalism and the definition of reward in Equation (4.4), we can use our models $f_a(x)$, which approximate $\mathbb{E}[Y(a)|X = x]$, to construct a prediction of this reward under each action a, and then define our treatment policy $\pi_\omega(x)$ as the one that chooses the action with the highest predicted reward

$$\pi_\omega(x) = \arg\max_{a} \omega \cdot f_a(x) + (1 - \omega) \cdot (1 - C(a)). \tag{4.6}$$

Constructing such a decision rule for each ω produces our desired set of policies Π. With this approach, we no longer need to perform a time-consuming enumeration over a large set of thresholds, and we can directly incorporate the information available in the predicted probabilities.

However, this approach still requires us to build predictive models of treatment effectiveness, and can introduce a trade-off between policy performance and interpretability. For instance, representing the outcome models f_a with linear functions allows us to interpret the learned policy and gain insight into features driving decisions by examining differences in coefficients.[4] In many settings, linear models may be too simple to accurately model outcomes, which can lead to poor performing models (and therefore policies). On the other hand, more complex models sacrifice the interpretability of the resulting policy. In the next section, we present a method which instead seeks to find a policy of the desired model class (e.g., linear) directly.

4.5 Direct Policy Optimization

In this approach, we seek to directly learn a policy which has an interpretable form, without learning any specific models of treatment effectiveness. We will use the same notion of reward defined in Section 4.4, and will optimize the (estimated) value of a treatment policy π, $\hat{V}_\omega(\pi)$. As before, we learn a range of policies corresponding to different values of ω. We first define the value of a policy $V_\omega(\pi)$ as

$$V_\omega(\pi) := \mathbb{E}_{x,r}\left[\sum_{a\in\mathcal{A}} r_\omega(a)\mathbf{1}\left[a = \pi(x)\right]\right], \tag{4.7}$$

where $\hat{V}_\omega(\pi)$ is the empirical estimate of this quantity. In this case, our goal is to find a function $\pi : \mathbb{R}^m \to \mathcal{A}$ which maximizes this objective. We note that any such policy can be written as $\pi(x) = \arg\max_{a\in\mathcal{A}} d(x, a)$ for some function $d : \mathbb{R}^m \times \mathcal{A} \to \mathbb{R}$. The optimal policy π^*_ω can then be written as

$$\pi^*_\omega = \arg\max_{\pi} \mathbb{E}_{x,r}\left[\sum_{a\in\mathcal{A}} r_\omega(a)\mathbf{1}\left[a = \arg\max_{a\in\mathcal{A}} d(x, a)\right]\right]. \tag{4.8}$$

We will omit ω in the remainder of this section, as we will choose a finite set of values for ω to generate a set of optimal policies π_ω for each.

We wish to optimize over the space of decision functions d using the empirical estimate of $V(\pi)$ to find an optimal policy, but the argmax operation makes this objective non-convex. Instead, we will use a differentiable *convex surrogate* objective [48, 59],

[4] If there are two actions, then this is a direct consequence of the formulation, and for more than two actions the policy can be interpreted as a set of linear classifiers by comparing the difference in coefficients for models of pairs of treatments.

in our case the multinomial deviance loss [17, 59]. This objective has several appealing properties: it is convex, differentiable, and yields a consistent estimator of the Bayes-optimal policy when solved to optimality. Concretely, we optimize over functions $f_a : \mathbb{R}^m \to \mathbb{R}$, where our resulting policy will be given by $\pi(x) = \arg\max_a f_a(x)$, and seek to minimize the following quantity in our empirical sample

$$\mathbb{E}_{x,r}\tilde{L}(f, x, r) \coloneqq -\mathbb{E}\left[\sum_{a \in \mathcal{A}} r(a) \log \frac{\exp f_a(x)}{\sum_{a'} \exp f_{a'}(x)}\right]. \tag{4.9}$$

In this work, we parameterize the functions f_a with a linear model, such that $f_a(x) = \theta_a^T x$. As noted, this objective yields a policy that is consistent for the Bayes-optimal policy when solved to optimality (proof provided in the Appendix):

Proposition 1. *For nonnegative rewards* $\mathbf{r}$*, and for an* f^* *that satisfies* $f^* = \inf_f \mathbb{E}_{x,r}\tilde{L}(f, x, r)$*, the corresponding policy* $\pi^*(x) = \arg\max_a f_a^*(x)$ *is equivalent to the the Bayes-optimal policy* $\pi^*(x) = \arg\max_a \mathbb{E}[r(a)|X]$.

Directly optimizing for a treatment policy in this way decouples the complexity of the outcome models in a given setting from the complexity of an effective treatment policy. This enables the learning of interpretable decision-making policies even in settings where modeling outcomes requires extremely sophisticated models. In the following section, we demonstrate a setting where this decoupling is crucial for achieving good performance.

4.6 Synthetic Evaluation: Direct vs. Indirect

In this section, we demonstrate a synthetic setting where direct policy learning can significantly outperform an indirect approach. In particular, this can occur when the true treatment outcome models are complex, but the optimal treatment rule is simple. For clarity, we illustrate the benefit of direct learning in a single-objective setting in these synthetic experiments, but an extension to multi-objective settings in the framework discussed previously is straightforward.

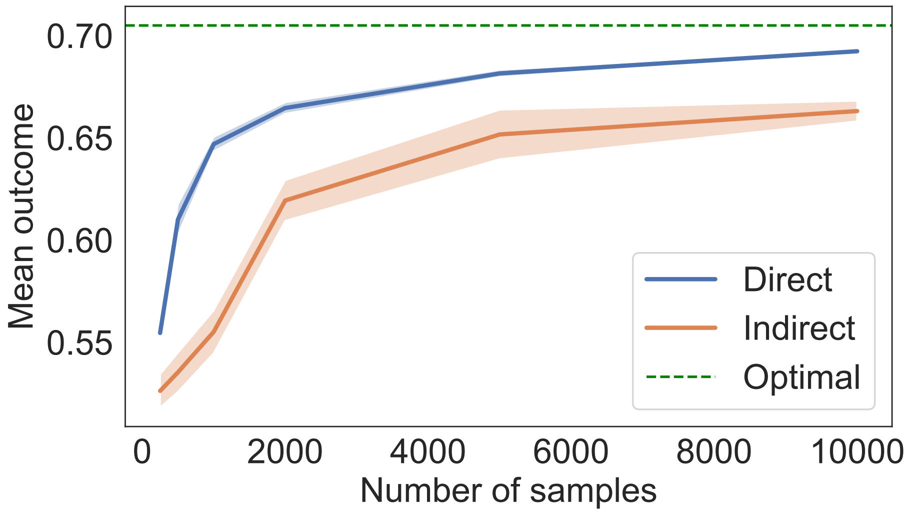


Figure 4-1: Comparison of indirect and direct policy learning approaches in synthetic setting. The direct method significantly outperforms the indirect approach, particularly in the low-training data regime, and approaches the performance of the Bayes-optimal policy.

Our environment consists of m-dimensional feature vectors $\mathbf{X} \in \mathbb{R}^m$ and an action space $\mathcal{A}$ with 3 actions. All feature values are drawn i.i.d. from a standard Gaussian distribution. $Y(a)$ is a binary random variable denoting the outcome of action a. The values of each $Y(a)$ for a given X are generated according to the following models:

$$Y(a) \mid X \sim \textbf{Bernoulli}\Big(\sigma\Big(X_a + \sum_{i=4}^{m} \alpha_i X_i^2 + \sum_{(i,j)\in S} \beta_i X_i X_j\Big)\Big) \tag{4.10}$$

for $a = 1, 2, 3$, where α_i, β_i are coefficients that are fixed across all 3 outcome models and S is a subset of all distinct pairs of features. These are all nonlinear functions of the features X, but the Bayes-optimal treatment rule for maximizing under these outcome models is given by an argmax over linear functions

$$\pi^*(X) = \underset{a\in\{1,2,3\}}{\arg\max}\ X_a. \tag{4.11}$$

We compare the performance of an indirect approach (expected reward maximization) and the direct policy optimization approaches for policy learning in this environment. In the indirect approach, we independently train logistic regression models

h_a to predict the outcomes $Y(a)$ for each a. The treatment rule is then defined as $\arg\max_a h_a(x)$. In the direct approach, we optimize the following loss function, where f is parameterized by a linear model:

$$\tilde{L}(f, x) = -\sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \mathbf{1}\left[Y(a) = 1\right] \log \frac{\exp f_a(x)}{\sum_{a'} \exp f_{a'}(x)} \tag{4.12}$$

The results are shown in Figure 4-1. We plot the mean outcome of both approaches on a held-out test set for various training set sizes. We only compute the mean performance on the subset of examples in the test set for which outcomes were not uniform across all 3 actions (i.e, not all 0 or 1), as performance on the remaining samples does not depend on the policy. We also plot the mean performance of the Bayes-optimal policy given in Equation (4.11).

We observe that direct policy learning significantly outperforms the indirect approach across a wide range of training set sizes and rapidly approaches the Bayes-optimal performance with far fewer samples. This synthetic experiment demonstrates that the direct learning approach, in contrast to the indirect approach, is able to take advantage of scenarios where the optimal treatment policy is simple, even when the true conditional outcome models are complex.

Chapter 5

Policy Learning for Uncomplicated UTIs

In this chapter, we evaluate the policy learning methods presented in Chapter 4 on the task of antibiotic prescription for uncomplicated UTIs. We first describe our setup for policy training and evaluation. We then evaluate our three approaches and compare their performance to several baselines, including clinicians and existing practice guidelines. We interpret our learned policies to identify important features for decision-making in this setting. Finally, we examine the utility of using data from complicated UTIs for learning better treatment policies for uncomplicated UTIs.

5.1 Experiment Setup

Recall that our patient cohort consists of uncomplicated UTI specimens collected between 2007-16 from two Boston-area hospitals (Chapter 3). We learned all treatment policies using a training set containing uncomplicated UTI specimens collected between 2007-13. Model hyperparameters for all methods were tuned by optimizing for the average validation performance across twenty 70%/30% train/validation splits of the training cohort. Further details of hyperparameter selection can be found in Appendix B.

We evaluated our learned policies on a test set containing specimens from 2014-16 with respect to the outcomes of IAT rate and the proportion of 2nd-line antibiotic usage. The IAT rate is defined as the proportion of specimens treated with an antibiotic to which they are resistant, while the 2nd-line usage rate is the proportion of specimens treated with either CIP or LVX.

5.1.1 Features

All features described in Chapter 3 were used as inputs to the models trained in this chapter. We filtered out features with zero variance in the training set (e.g., features that were not present in any specimens).

5.1.2 Indirect Methods

We parameterized the conditional treatment effectiveness models $f_a(x)$ for each antibiotic with logistic regression models. In practice, we trained our models to predict treatment resistance instead of predicting treatment effectiveness. We experimented with more complex, nonlinear model choices, such as random forests and XGBoost, but did not find significant gains in predictive performance on the validation set and thus chose not to use them in experiments. We trained separate models to predict resistance for each antibiotic, tuning the regularization strength and type ($L1$ vs. $L2$). Hyperparameters were chosen to optimize the average AUC across the validation sets.

Thresholding

As described in Chapter 4, the thresholding method requires the definition of a threshold search space $\mathcal{T}$. We implicitly defined $\mathcal{T}$ using a set of false negative rates (FNRs) for predicting resistance. Given a FNR value for a particular antibiotic, we used the training ROC curve for that agent's treatment resistance model to derive the probability threshold attaining that FNR. We note that a higher FNR corresponds to a higher resistance threshold (i.e., fewer specimens will be predicted resistant to that agent). The threshold space $\mathcal{T}$ is defined all combinations of thresholds corresponding to these FNR values.

Due to the high correlation between resistance to CIP and LVX, we constrained our search space to combinations where the resistance thresholds for the broad spectrum agents are equal to reduce the computation required during the tuning process. With this constraint, our search space consisted of 1,331 threshold combinations.

To produce a prescription decision given a specimen's features x, we generate predicted resistance probabilities for each antibiotic using the models $f_a(x)$, then select the narrowest spectrum antibiotic whose predicted resistance probability is under the resistance threshold corresponding to that agent. The treatment selection order used in the experiments, from narrowest to broadest, is: NIT $<$ SXT $<$ CIP $<$ LVX. If a specimen is predicted to be resistant to all 4 agents for a given set of thresholds, we default to prescribing NIT, as it was observed to have a significantly lower resistance rate than SXT in the training set (Table A.1).

Expected Reward Maximization

We provide a concrete instantiation of the reward function $\mathbf{r}_\omega$ to be used for decision making under this approach. We recall the definition of the composite reward function given in Section 4.4 used for the expected reward maximization and direct policy learning approaches. The treatment effectiveness vectors $\mathbf{Y}$ correspond to a patient's susceptibility to each antibiotic $Y_i(a) = \mathbf{1}\left[\text{patient } i \text{ is susceptible to antibiotic } a\right]$, and the cost vectors $\mathbf{C}$ for the treatments are a function of the class of the chosen antibiotic $C_i(a) = \mathbf{1}\left[a \text{ is a 2nd-line antibiotic}\right]$.

The composite treatment reward is defined as a linear combination of the effectiveness and costs for each antibiotic using the preference $\omega \in [0, 1]$, given by $\mathbf{r}_i = \omega \cdot \mathbf{Y}_i + (1 - \omega) \cdot (1 - \mathbf{C}_i)$. As ω is reduced, more weight is placed on avoiding 2nd-line antibiotic usage, even at the cost of additional cases of IAT. Varying ω allows us to learn a set of treatment policies that achieve different trade-offs between treatment effectiveness and broad spectrum usage.

5.1.3 Direct Policy Optimization

We recall from Chapter 4 that we parameterize the functions $f_a(x)$ used in the direct policy model with linear models for our experiments. We use the same reward function $\mathbf{r}_\omega$ defined in Section 4.4 to construct the surrogate loss. We optimize the surrogate loss using an Adam optimizer and add an L2 regularization term on the model weights [21]. We tune the number of training epochs using an early stopping criterion on the mean reward earned on the validation set. All models for the direct approach are implemented in PyTorch [40].

5.1.4 Baselines

IDSA Guidelines

The Infectious Diseases Society of America (IDSA) has published recommended practice guidelines for prescribing antibiotics to patients with UTIs [15]. We introduce a simplified adaptation of these guidelines to serve as a baseline for our learned policies. The decision logic in these guidelines is as follows:

If annual hospital-wide resistance rates to SXT exceeded 20% in the prior year, the guidelines recommend treatment with NIT *unless* the patient has a record of prior exposure or resistance to NIT in the past 90 days. If that is the case, the guidelines recommend treatment with CIP.

If the annual resistance rate to SXT did *not* exceed 20%, the guidelines recommend SXT as the second treatment option if the patient does not satisfy the conditions for treatment with NIT. If the patient has resistance or exposures to both 1st-line agents, the guidelines recommend CIP. In our dataset, annual resistance rates to SXT across all specimens exceed 20% in every year between 2007-16, so applying these guidelines to our patient cohort produces a policy that completely avoids SXT and always prescribes either NIT or CIP.

Clinician-adjusted Guidelines

Given the limited amount of prior exposure and resistance history available in the uncomplicated UTI cohort, strict adherence to the guidelines described in the previous section results in a policy with unrealistically low levels of 2nd-line treatment that would never be observed in actual clinical settings. We propose an adaptation of these guidelines that incorporates clinician decisions, allowing for a policy that uses more levels of 2nd-line usage and enabling comparison of our learned policies to a more reasonable baseline.

The adjusted guidelines only differ from the previously described guidelines on specimens where the guidelines recommend a 1st-line treatment, but clinicians empirically selected a 2nd-line treatment. Among these specimens, the guideline's decision is replaced with the clinician's decision with some fixed probability p. Adjusting p allows the adjusted guidelines to use more or less 2nd-line usage. Since clinicians use significantly more 2nd-line treatment than practice guidelines, increasing p leads to a baseline policy that uses higher levels of 2nd-line treatment. For all other specimens, the adjusted guidelines adhere to the decision of the unmodified practice guidelines.

Baselines from Yelin et al. (2019) [54]

We also compare our methods to the unconstrained and constrained treatment selection approaches introduced in [54]. These methods were presented specifically in the context of antibiotic prescription for UTIs, so they serve as the most direct comparison to our methods. Unlike our approaches, however, the methods in [54] only learn a single policy, not sets of policies that trade off across the multiple objectives.

In both approaches, we first learn conditional outcome models f_a to predict treatment resistance, just as in our indirect approaches. The policy defined by the unconstrained approach selects the treatment with the lowest predicted probability of resistance from these models:

$$\pi(x) = \arg\min_a f_a(x).$$

This approach does not constrain the rate of 2nd-line treatment in any way.

The constrained approach seeks to minimize the IAT rate while constraining the empirical treatment distribution of the learned policy to match that of clinicians. The resistance probabilities predicted by the outcome models are adjusted by adding treatment-specific costs to each prediction:

$$f'_a(x) = f_a(x) + c_a,$$

where c_a are treatment specific costs. The policy defined by the constrained approach then selects the treatment with the minimal adjusted 'score':

$$\pi_{\mathrm{C}}(x) = \arg\min_a f'_a(x).$$

The costs c_a are chosen to constrain the treatment distribution of π_{C} to match the empirical treatment distribution of clinicians. We solve for the costs by iteratively updating them according to the following equation until convergence:

$$c_a^{t+1} \leftarrow c_a^t + \alpha \cdot (\mathrm{Count}^{\pi^t}(a) - \mathrm{Count}^{\mathrm{doc}}(a)),$$

where α is a step size, $\mathrm{Count}^{\pi^t}(a)$ is the number of uses of treatment a in the policy defined by the costs at step t, and $\mathrm{Count}^{\mathrm{doc}}(a)$ is the number of clinician uses of treatment a.

5.2 Results

5.2.1 Predicting Resistance

We first examine the performance of the conditional outcome models $f_a(x)$ trained to predict resistance for each of the four antibiotics (Table 5.1). We are worst at predicting resistance to NIT, while we are relatively better at predicting resistance to the broad spectrum agents. When prediction is constrained to the subset of the

Antibiotic	All specimens	Prior exposure / resistance
NIT	0.563	0.605
SXT	0.593	0.673
CIP	0.637	0.764
LVX	0.637	0.766

Table 5.1: Test AUCs = for predicting resistance to antibiotics. There are 3941 total specimens in the test cohort, and 1033 specimens have a prior history of antibiotic exposure or resistance in the past 180 days.

test cohort with a prior recorded history of antibiotic exposure or resistance in the last 180 days (roughly 25% of the cohort), predictive performance is significantly improved for all four agents, with a particularly large improvement for the broad spectrum agents. Overall, however, reliably predicting resistance to these agents for patients with uncomplicated UTI is quite difficult due to the limited availability of patient history.

5.2.2 Policy Learning

We now examine the performance of our policy learning methods with respect to the clinical outcomes of interest: the IAT rate and 2nd-line usage rate. In Figure 5-1, we plot the performance of the policy sets learned by each of the three approaches on the test cohort. The three methods we have proposed all achieve similar performance across a wide range of trade-offs between the two objectives and significantly outperform clinicians, practice guidelines, and the methods presented in [54].

In the reward maximization and direct learning approaches, the reward weight ω successfully controls the trade-off learned by the policy, producing a reasonably 'smooth' policy performance frontier. As ω is reduced (i.e, treatment effectiveness is less important), policy performance moves down and to the right along the frontier shown in Figure 5-1. In the thresholding approach, the 2nd-line usage constraints b_j control the trade-offs between objectives fairly well, but the movement along the performance frontier is somewhat less consistent than the other two approaches.

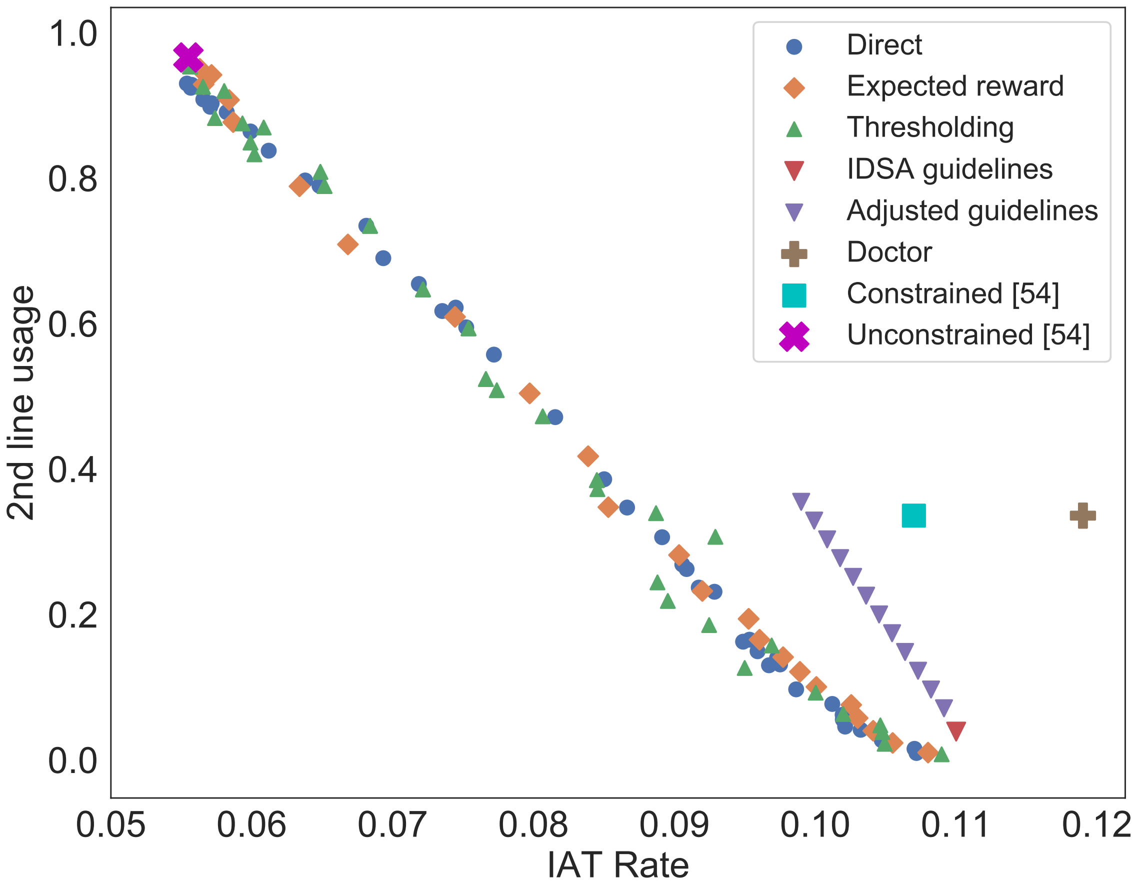


Figure 5-1: Comparison of test performance for policy sets learned by each of the three proposed learning approaches against clinicians and baselines. The IDSA and adjusted guidelines refer to the policies described in the first two sections of Section 5.1.4. The constrained and unconstrained baselines refer to the policies produced by the approaches presented in [54], discussed in the last section of Section 5.1.4.

Policies for the expected reward and direct learning approaches were calculated using values of ω in the interval $[0.85, 1.0]$, and outcomes for the adjusted baseline are shown for several values of the parameter p in the interval [0.0, 1.0]. Our approaches significantly outperform all baselines across a wide range of trade-offs between the IAT and 2nd-line usage rates.

	IAT	**2nd-line usage**
Doctor	11.9%	33.6%
Thresholding	8.9%	33.1%
Expected reward maximization	9.0%	28.2%
Direct learning	8.9%	30.4%
Constrained selection [54]	10.6%	33.6%

Table 5.2: Comparison of primary outcomes for learned policies using similar levels of 2nd-line treatment as clinicians.

	IAT	**2nd-line usage**
Doctor	11.9%	33.6%
Thresholding	10.9%	0.8%
Expected reward maximization	10.8%	1.0%
Direct learning	10.7%	0.9%
IDSA Guidelines	10.9%	3.9%

Table 5.3: Comparison of primary outcomes for learned policies using minimal levels of 2nd-line treatment.

We examine the performance of policies at a few points along the trade-off frontier. In Table 5.2, we provide the performance of policies learned by our approaches that use similar levels of 2nd-line treatments as clinicians. All three policies reduce the IAT rate by over 25% relative to clinicians while also achieving minor improvements in the usage of 2nd-line antibiotics. We also outperform the constrained selection approach from [54], reducing the IAT rate by over 15%. In Table 5.3, we examine the performance of learned policies that use extremely low levels of 2nd-line treatment. These policies are still able to achieve almost a 10% improvement in the IAT rate relative to clinicians while essentially eliminating all 2nd-line usage.

The unconstrained baseline proposed in [54] produces a policy achieving an extremely low IAT rate while using essentially only 2nd-line treatments. This is a product of the significantly lower resistance rates to 2nd-line treatments in both training

	IAT	**2nd-line usage**
Adjusted guidelines	10.8%	9.7%
Thresholding	9.5%	12.7%
Expected reward maximization	10.0%	10.0%
Direct learning	9.9%	9.7%

Table 5.4: Comparison of primary outcomes for learned policies relative to a clinician-adjusted modification of practice guidelines.

and test sets, which results in predicted resistance probabilities that are also almost always lower than predictions for 1st-line treatments. Such high usage of 2nd-line treatments would not be useful in clinical practice.

Our learned policies also outperform the simplified version of the IDSA guidelines, which achieve a 10.9% IAT rate at a 3.9% 2nd-line usage rate on the test cohort (Table 5.3). We note that strict adherence to these guidelines is already a significant improvement over current clinician performance with respect to both objectives, but all three of our proposed approaches are able to achieve similar IAT rates with lower 2nd-line usage. For instance, the direct optimization approach is able to learn a policy achieving a slightly lower IAT rate of 10.7% with only 0.9% 2nd-line usage, a reduction of over 75% relative to guidelines.

However, as noted in Section 5.1.4, these practice guidelines use very little 2nd-line treatment, and such low levels of broad spectrum usage would never occur in clinical practice. We compare our learned policies to the clinician-adjusted version of the practice guidelines that utilize more reasonable levels of 2nd-line treatment. We select the policy from each learned policy set that was learned with the parameter setting that minimizes IAT while achieving no more than 10% 2nd-line usage on the validation set. We apply the same criteria for selecting the parameter p in the clinician-adjusted guidelines. With this choice of parameters, we find that our learned policies still exhibit meaningful improvement over these clinician-adjusted guidelines (Table 5.4). For instance, the direct optimization approach reduces the IAT rate by 8% without a change in the rate of 2nd-line usage.

Overall, the results in this section indicate that the three policy learning methods introduced in Chapter 4 achieve similar performance relative to one another. However, all the approaches do exhibit large improvements in rates of treatment effectiveness and broad spectrum usage relative to clinicians, practice guidelines, and previous methods presented in the literature.

5.3 Comparison with Clinician Decisions

In this section, we perform a post-hoc analysis of our model's actions relative to clinician decisions to better understand where our model is able to achieve its improvements over clinician performance. To do so, we first group each specimen in the test cohort into four groups depending on (1) whether the clinician used a 2nd-line treatment and (2) whether the clinician's decision resulted in IAT. For the specimens in each of these four groups, we perform the same breakdown into four subgroups, based on the selected policy's treatment decisions for those specimens. Here, we only present this analysis for the policy learned by the thresholding approach at the trade-off listed in Table 5.4; analyzing policies learned via the other two approaches yields similar results. This breakdown is shown in Figure 5-2.

The learned policy successfully selects an appropriate 1st-line treatment in many cases where a clinician treated a patient with an appropriate 2nd-line treatment instead. The policy depicted in Figure 5-2 switches 80% of the specimens receiving appropriate 2nd-line treatment to an appropriate 1st-line treatment, while switching less than 7% of these cases to an inappropriate 1st-line treatment. Our learned policies are thus significantly better than clinicians at identifying candidates for which 2nd-line treatment is unnecessary.

The learned policy is also able to give an appropriate 1st-line treatment in nearly 50% of the cases where clinicians gave an inappropriate 1st-line treatment. Compared to clinicians, we are much more effectively able to select between the two 1st-line antibiotics, achieving lower IAT rates among the groups of patients for whom we assign NIT and SXT. While clinicians achieve IAT rates of 11.1% and 19.1% when they prescribe NIT or SXT, respectively, the policy shown in Figure 5-2 achieves far lower IAT rates of 9.8% and 6.7% when using these treatments.

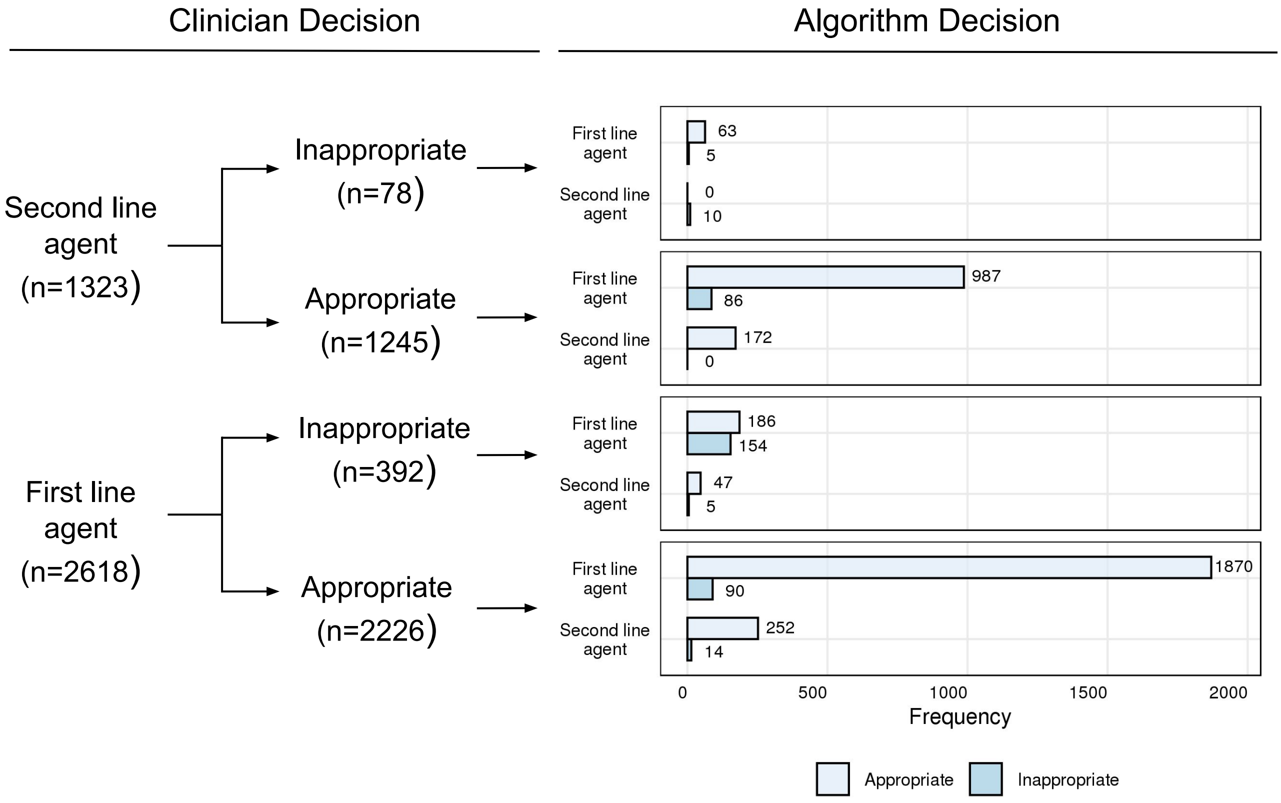


Figure 5-2: Breakdown of decisions made by thresholding policy at the trade-off in Table 5.4 relative to clinician decisions.

5.4 Feature Interpretation

In this section, we interpret the policies learned by our methods. We first analyze the most predictive features of resistance in the models trained for the two indirect approaches. We then describe approaches to extract the most important features used for decision-making by our direct model.

5.4.1 Treatment Resistance Models

We first examine the features learned by the individual logistic regression models trained to predict resistance for each antibiotic of interest. We extract the top 10 most positive and negative coefficients for predicting resistance to each agent; these are shown in Tables A.2 through A.9.

Many of the features predictive of resistance to NIT and SXT are unsurprising; for instance, we find that prior infection with a pathogen resistant to a particular agent is highly predictive of current resistance to that agent. Interestingly, recent exposure to SXT is highly predictive of current resistance to the agent, but prior exposure to NIT does *not* seem to have a large effect on current resistance. We observe that both prior resistance and exposures are highly predictive of resistance to the 2nd-line antibiotics.[1] As mentioned previously, resistance to these two agents is highly correlated, so the predictive features identified for these agents are almost identical.

5.4.2 Direct Policy Learning

The direct policy learning approach enables interpretation of the learned treatment policy to understand features important for decision-making. The linear model learns a $m \times |\mathcal{A}|$ matrix θ, where m is the dimensionality of the feature space and $\mathcal{A}$ is the action space. Each column contains the coefficients used to calculate the model output for a particular antibiotic. We extract the features important in our policy's decisions for recommending antibiotic a over antibiotic a' using pairwise differences in coefficient values in the corresponding columns of θ. The features i for which $\theta_a(i) - \theta_{a'}(i)$ is largest are most important in driving recommendation of a over a'.[2]

We first focus on the factors important in driving recommendation of NIT over SXT and vice versa. As noted in Section 5.3, our algorithm's ability to effectively select between these agents contributes significantly to its improvement over clinicians. We extract these features from a direct policy learned with reward parameter

[1] Both CIP and LVX belong to a class of antibiotic known as fluoroquinolones, which appear in the list of important features.

[2] We denote the value in the i^{th} row and j^{th} column of θ as $\theta_j(i)$.

$\omega = 0.85$, which learns a policy that uses essentially no 2nd-line treatment. The features identified in this analysis are listed in Tables A.10 and A.11. Many of the features identified here overlap with features that are highly predictive of resistance to one of these agents. For instance, prior resistance or exposure to SXT is an important feature in driving recommendation of NIT over SXT; similarly, prior resistance to NIT drives recommendation of SXT over NIT. Features that are *negatively* predictive of resistance to SXT also drive recommendation over NIT, such as being of a white race or having prior resistance to cefazolin.

We next examine the features predictive of 2nd-line usage over 1st-line usage, and vice versa. There is no obvious way to extract these features directly from θ. We instead fit a regularized logistic regression model that uses specimen features to predict the specimens for which the policy selected 2nd-line agents. This is a post-hoc analysis of our learned policy, not a prediction task, so we fit this model directly on specimens in the test cohort. The most positive (negative) coefficients in this model can be interpreted as the features most predictive of 2nd-line (1st-line) usage. We fit this model on the decisions of a direct policy learned with reward parameter $\omega = 0.91$, which achieves a 9.4% IAT rate with 14% 2nd-line usage on the test cohort.

The features obtained from this analysis are shown in Tables A.12 and A.13. While some of the identified features are unsurprising, such as prior resistance or exposure to 2nd-line treatments driving selection of a 1st-line treatment, we also observe several features that have not appeared in previous analyses. We find that prior comorbidities, including obesity, diabetes, and depression, are significant drivers of 2nd-line treatment in our learned policy. Examination of resistance rates in these populations during the training period (2007-13) confirms that these patients have higher resistance rates to 1st-line agents. For instance, 24% of patients with diabetes are resistant to NIT, while 27% are resistant to SXT. Among patients with depression, 20% are resistant to NIT, while 25% are resistant to SXT. These rates are significantly higher than the 11% and 19% resistance rates for NIT and SXT, respectively, across the full uncomplicated cohort.

We perform the same analysis on the clinicians' empiric treatment policy in the test cohort to compare the features learned by our model to features driving clinician usage of broad spectrum antibiotics. These results are shown in Tables A.14 and A.15. There is minimal overlap between the features used for treatment selection by clinicians and our learned policy. Clinicians seem to be creatures of habit: prior 2nd-line usage on a patient is highly predictive of the clinician choosing 2nd-line treatment again. Similarly, prior usage of NIT or SXT (a folate inhibitor) is predictive of 1st-line treatment. Location is also a significant driver of clinician behavior, as 2nd-line prescriptions are much more frequent in the ER than in outpatient settings. However, clinicians do use prior resistance to 2nd-line treatment as an indicator to avoid using 2nd-line antibiotics. This analysis suggests that clinicians are driven to 2nd-line usage by factors largely unrelated to resistance, and can significantly improve usage of these agents by instead focusing on more relevant aspects of a patient's medical history.

5.5 Learning from Complicated UTIs

So far, we have only used data from patients with uncomplicated UTIs to learn treatment policies, discarding data from patients with complicated UTIs that did not satisfy the selection criteria described in Section 3.3. In this section, we examine the utility of data from complicated UTIs for learning better policies to guide treatment selection in uncomplicated UTIs. We first provide an overview of the cohort of patients with complicated UTIs to highlight differences compared to the uncomplicated cohort. We then describe how this additional data can be incorporated into the policy learning process and evaluate its impact on performance.

5.5.1 Complicated Cohort Overview

Our training set of uncomplicated UTIs consists of 11,865 specimens collected between 2007-13 (Section 3.3). However, there are also 69,097 additional UTI specimens from this time period that were filtered out of the uncomplicated cohort due to at least one of the reasons specified in Section 3.3. Patients with complicated UTIs during the training period exhibit significantly higher resistance rates to all treatments, and

Antibiotic	**Complicated**	**Uncomplicated**
NIT	23.6%	11.2%
SXT	25.8%	19.6%
CIP	23.4%	5.3%
LVX	24.6%	5.1%

Table 5.5: Resistance rates for complicated and uncomplicated UTIs from 2007-13.

	Prior exposures		**Prior resistance**	
	Complicated	**Uncomplicated**	**Complicated**	**Uncomplicated**
NIT	20.2%	5.7%	18.5%	13.1%
SXT	20.2%	10.2%	31.0%	28.3%
CIP	18.7%	3.6%	39.5%	25.5%
LVX	22.1%	3.8%	32.4%	10.0%

Table 5.6: Prior history of antibiotic resistance and exposures in complicated and uncomplicated UTI cohorts.

this difference is particularly large for 2nd-line treatments (Table 5.5). Unlike the uncomplicated cohort, there is no meaningful difference between resistance rates for 1st and 2nd-line treatments in complicated UTIs.

While resistance rates are significantly higher among complicated UTIs, patients with these UTIs are also far more likely to have a prior history of antibiotic resistance or exposures compared to patients with uncomplicated UTIs. Table 5.6 shows the percentage of patients with any history of resistance or exposure to each of the 4 antibiotics; the gap in prior resistance rates is particularly large for all treatments. The significantly richer patient history may help us better learn underlying relationships between patient attributes and resistance that were not easily identified from the sparser data available for patients with uncomplicated UTIs.

5.5.2 Experiment Setup

Our setup generally follows the descriptions presented for our original set of experiments in Section 5.1. The only difference is in the choice of training set for both learning predictive models of resistance and the direct policy.

When learning models to predict resistance to an agent, we expand the training set to include all examples from complicated UTIs with a recorded resistance label for that antibiotic. All specimens in the uncomplicated cohort have resistance labels for all specimens (by construction), but 17% of specimens in the complicated cohort are missing labels for at least one treatment. Hyperparameters for these predictive models are tuned using validation sets consisting of *only* uncomplicated UTI patients using the same process outlined in Section 5.1.

When learning policies via the direct approach, our training set is expanded to include examples from complicated UTIs with recorded resistance labels for *all* four antibiotics, as all labels are necessary to construct the loss incurred by an example. Due to this additional restriction, the training set for the direct policy is smaller than the training sets for any of the predictive models used in indirect approaches. We also modify the surrogate loss to apply importance weights derived from kernel mean matching to training examples corresponding to complicated UTIs [14]. Details of the procedure used to compute these importance weights can be found in Appendix B.4.

5.5.3 Results

Expanding the training set yielded improvements in predicting resistance for all agents except NIT, with a particularly large increase in AUC for the 2nd-line treatments (Table 5.7). In Figure 5-3, we compare the full performance frontier of the policy sets learned from each approach using uncomplicated UTIs vs. all UTIs. Figure 5-4, shows a head-to-head comparison of the policies learned by each method when trained on all UTIs. The test performance of a selected few policies from each frontier are shown in Tables 5.8 and 5.9, alongside the performance of policies learned using only uncomplicated UTIs at similar trade-offs between outcomes.

	Full Cohort		**Prior Resistance / Exposure**	
Antibiotic	**Uncomp. UTIs**	**All UTIs**	**Uncomp. UTIs**	**All UTIs**
NIT	0.563	0.556	0.605	0.608
SXT	0.593	0.605	0.673	0.681
CIP	0.637	0.660	0.764	0.782
LVX	0.637	0.659	0.766	0.762

Table 5.7: Test AUCs for predicting resistance to antibiotics using models trained on data from only uncomplicated UTIs vs. all UTIs.

	All UTIs		**Uncomplicated UTIs**	
	IAT	**2nd-line usage**	**IAT**	**2nd-line usage**
Thresholding	9.0%	28.4%	8.9%	33.1%
Expected reward maximization	9.0%	21.4%	9.0%	28.2%
Direct learning	8.9%	21.6%	8.9%	30.4%

Table 5.8: Comparison of primary outcomes for learned policies trained using data from only uncomplicated UTIs vs. all UTIs.

	All UTIs		**Uncomplicated UTIs**	
	IAT	**2nd-line usage**	**IAT**	**2nd-line usage**
Thresholding	10.8%	0.9%	10.9%	0.8%
Expected reward maximization	10.9%	1.0%	10.8%	1.0%
Direct learning	10.5%	1.0%	10.7%	0.9%

Table 5.9: Comparison of primary outcomes for learned policies trained using data from only uncomplicated UTIs vs. all UTIs.

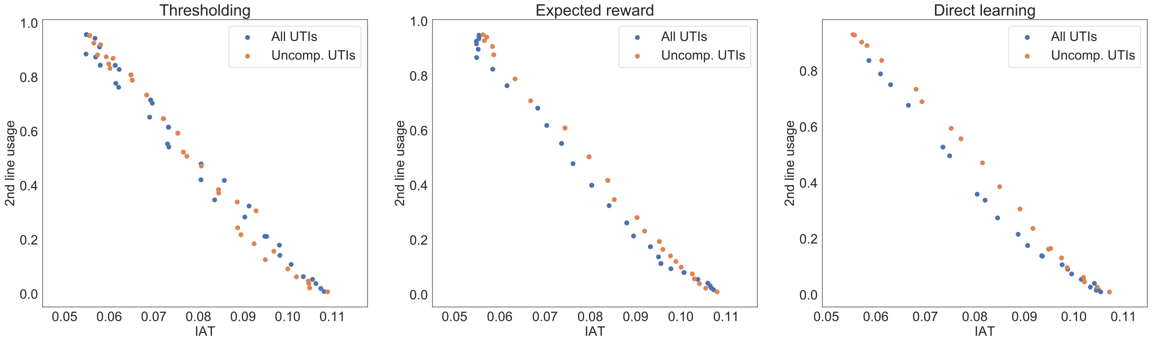


Figure 5-3: Comparison of policy performance frontiers learned using all UTIs and uncomplicated UTIs for thresholding (left), expected reward maximization (middle), and direct optimization (right). The expected reward and direct learning approaches learn superior policies when trained on the expanded dataset of UTIs.

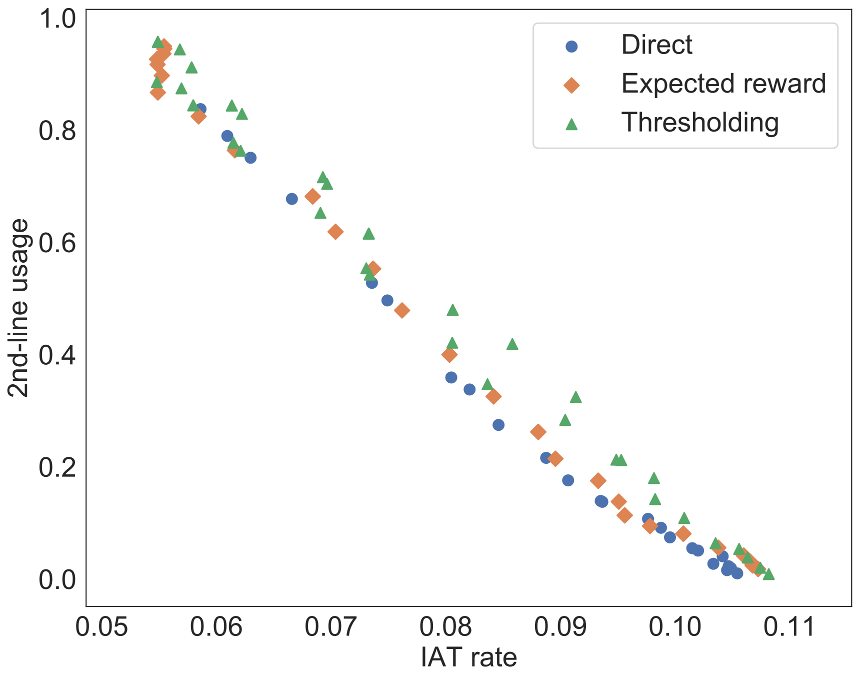


Figure 5-4: Comparison of policy performance frontiers learned using all UTIs for the three policy learning methods. The expected reward and direct learning approaches outperform the thresholding approach when trained on the expanded dataset of UTIs.

The improvements in predictive modeling translated into meaningful improvements in the performance of policies from the expected reward maximization approach across most trade-offs between IAT and 2nd-line usage, but did not yield an improvement in policies learned via the thresholding approach. Policies learned via the direct optimization approach also benefited from the additional training data, and actually exhibited the most improvement across most trade-offs between the two outcomes and slightly outperformed the expected reward maximization approach. In general, the improvements in performance occurred at trade-offs using higher levels of 2nd-line treatment (e.g., $> 10\%$), which aligns with the greater improvement in performance for predicting resistance to 2nd-line treatments noted previously. For an appropriate choice of the trade-off between outcomes, the direct learning approach yields a policy that reduces the IAT rate by 25% relative to clinicians while also reducing 2nd-line treatment levels by over 35% (Table 5.8).

The improved conditional outcome models and direct policies identify some important features that did not previously appear in analyses of policies learned from uncomplicated UTIs alone. For instance, we find that high colonization pressure to NIT and CIP are predictive of resistance to those respective agents. For the most part, however, the features most predictive of resistance to each agent largely stay the same. A comprehensive overview of the important features in these models, highlighting the newly identified features, are shown in Tables A.16 through A.23.

These results confirm the value of data from complicated UTI patients for learning policies to be applied on uncomplicated UTI patients. Despite having completely different distributions of features and labels (e.g., significantly higher resistance rates), the underlying models of resistance for each of these groups appear to be fairly similar and generalize effectively when evaluated on only patients with uncomplicated UTIs.

Chapter 6

Learning to Defer

In our work thus far, we have yet to account for a crucial component of the clinical decision-making process: doctors. Treatment policies developed for applications such as antibiotic prescription are not deployed in isolation. Instead, they are intended to be used as decision-support tools that are integrated into existing clinician workflows and decision-making processes. For instance, it is often not necessary for the algorithm to provide input on every decision, and clinicians are likely to grow tired of a tool that does so. As such, it is important that we develop algorithms that learn policies with the ability to defer to clinician decisions prior to deployment in real-world settings.

In this chapter, we consider the problem of learning to defer to clinicians in the empiric antibiotic prescription setting. We first present and formalize two settings with very different implications for when the algorithm should defer to clinicians. We then introduce three approaches for learning treatment policies with a deferral option in these settings, and end the chapter by presenting results from applying these approaches to the uncomplicated UTI cohort.

6.1 Goals of Deferral

There are several possible objectives when learning treatment policies with the ability to defer to an external decision-maker, and the desired objective is generally dependent on the setting of interest. In many contexts, we are interested in combining clinician and algorithm decisions in a way that improves overall performance of the decision-making system. A learned policy and clinicians may achieve poor performance on relatively disjoint portions of the population. By learning to defer on cases where the algorithm under-performs clinician decisions, the combined performance of the algorithm and clinicians may be better than either decision-maker alone.

In our antibiotic prescription setting, however, the results of Chapter 5 and some additional analyses suggest that doctors underperform learned treatment policies in both clinical outcomes of interest across all parts of the population. It is unlikely that combining clinician and algorithm decisions can significantly improve overall policy performance; thus, we are not interested in addressing this setting in this chapter. We instead examine two settings with alternate objectives for deciding when to defer:

1. **Algorithm errors are costly**. In many high-stakes decision-making settings, errors made by an algorithmic actor may be significantly more costly than errors made by a human. For instance, in medical settings, a poor decision provided by a decision support tool may have significant legal and ethical repercussions, while errors made by clinicians are common occurrences. In this setting, it is desirable to minimize the amount of decision-making errors made by non-human agents, and one might want the algorithm to defer on cases where it is likely to achieve poor performance, even if doctors do not perform any better on those cases. Minimizing algorithm errors on the portion of the population where it makes a decision may promote trust in the algorithm and increase adoption.

2. **Algorithm interventions are costly**. In some decision-making settings, each algorithm intervention may incur some cost, either monetary or psychological. In clinical settings, frequently alerting doctors with algorithm decisions may lead to a phenomenon known as ‘alert fatigue’, resulting in clinicians excluding these tools from their workflows and ignoring potentially valuable input from learned policies in the long run [34]. If an algorithm treats all cases with equal importance and always provides a recommendation, it becomes difficult for the clinician to identify when they should pay more attention to the tool’s input. In

> this setting, we wish to optimize the combined performance of the learned policy and clinicians, subject to an upper limit on the number of decisions taken by the algorithm. In other words, the algorithm should aim to provide input in the portion of the population where it performs significantly better than clinician decisions. Designing a system in this way may also increase the likelihood that clinicians incorporate algorithm input into their decision-making process.

Successfully achieving these objectives can play a significant role in the adoption of a newly deployed decision support tool, but they have not been explicitly examined in prior work on deferral in medical decision-making settings. In this chapter, we further formalize these settings and develop methods that can be adapted to learn policies that achieve either of these objectives.

6.2 Related Work

The problem of learning a classifier with a deferral or rejection option,[1] also known as *rejection learning*, has been studied extensively. Approaches to the general rejection learning problem generally assume the presence of a downstream *expert*, who can achieve essentially perfect performance on the task at hand, and imposes a cost for deciding to defer to this expert. The objective is to maximize overall performance subject to this deferral cost.

This problem formulation was first studied by Chow, who investigated the trade-off between a classifier's error and rejection rates and constructed a Bayes-optimal rule for minimizing error for a given rejection rate in the binary setting [5]. Early work developed and analyzed confidence-based approaches for learning to defer, constraining the rejector to be a simple thresholding function of the predictions of the learned classifier [2, 12]. Later work moved beyond the paradigm of threshold-based rejection decisions, proposing methods for jointly learning a classifier and a more general rejector function in the binary classification setting [7]. More recent work has extended both thresholding and joint learning approaches to multi-class settings.

[1] We use the terms *deferral* and *rejection* interchangeably in this section.

Theoretical work in this area has proven that thresholding-based approaches in the multi-class setting are consistent for the Bayes-optimal classifier/rejector pair, while jointly learning the classifier and rejector with a convex surrogate loss does not have this property [37].

However, our setting does not have a perfect downstream expert, as clinician decisions are severely flawed. There has been work towards learning models that adapt to an imperfect downstream expert's performance and seek to optimize the overall performance and fairness properties of the combined classifier and expert system [28]. Similar to [7], the approach presented in [28] jointly learns a classifier and rejector, but incorporates the expert's loss on each example into the objective function. This allows the rejector to account for the expert's performance when deciding where to defer. Other approaches to this problem involve estimating the uncertainty of both the learned classifier and an imperfect expert on a given example, leaving the final decision to the decision-maker with the lower estimated uncertainty [41].

These approaches assume that the ultimate goal is optimization of the combined performance of algorithms and human experts. As noted in Section 6.1, we are not as interested in this goal; instead, one of our desired objectives is optimizing the system's combined decision-making performance, subject to a limited number of allowed interventions from the algorithm. While this also requires the policy to defer in a way that adapts to the expert's performance, it ultimately has a different goal than the works just described, and in this section, we propose methods to achieve this alternate objective.

In Section 6.1, we noted that we are also interested in learning to defer in a way that minimizes algorithm errors when it chooses to make a decision. It is useful to examine this aim through the lens of *selective classification*, which provides a reframing of the rejection learning problem. While the general rejection learning framework assumes that deferral incurs a fixed cost, selective classification instead poses the problem as maximizing accuracy - or another metric of interest - subject to a constraint on the desired *coverage rate*, defined as the proportion of examples on which the classifier makes a decision.

Weiner et al. introduced the low error selective strategy (LESS) for selective classification in the binary setting, quantifying the difficulty of classifying an example by examining the difference between models trained with each of the two possible labelings of the example [9]. However, such an approach does not extend naturally to multi-class settings. Geifman et al. proposed a simple thresholding-based approach to convert a trained neural network into a selective multi-class classifier, only making decisions on examples where the model expressed sufficiently high confidence in one of the label classes [13].

In this work, we move beyond *selective classifiers* and develop methods for learning *selective treatment policies* that aim to maximize the average treatment outcome on the subset of patients where a decision is taken, which has not been addressed in previous work. In standard classification settings, each example is typically assumed to belong to exactly one target class, and this assumption plays a key role in previous work to construct loss functions for learning such classifiers with a deferral option. In policy learning settings, however, there is not necessarily a single 'correct' treatment. We instead have the notion of a treatment outcome (or reward), where it is possible for multiple treatments to be optimal (i.e., reward-maximizing) for the same patient.

As we have highlighted previously, the empiric antibiotic prescription setting is also multi-objective in nature; prior work on learning to defer has largely focused on learning in settings with a single objective. Here, we extend the methods introduced in Chapter 3 to learn policies that are both able to defer to human actors while learning policies that trade-off among the multiple clinical objectives of interest.

6.3 Methods

In this section, we first formalize each of the two settings introduced in Section 6.1, and define the objectives we wish to optimize in each setting. We then present three methods for learning treatment policies with the option to defer to clinicians, and adapt them to achieve the objectives for each setting of interest.

The first approach naturally incorporates deferral as an action in the direct policy learning framework introduced in Chapter 4. We assign a reward for deferral for each example, then simultaneously learn a good treatment policy and where to defer for a given objective. The second approach is analogous to the expected reward maximization approach: we learn conditional outcome models to predict the reward earned under each action, including deferral, and select the action maximizing the predicted reward. The final approach does not use a notion of reward; instead, we first learn a policy without deferral as an option, then learn a *rejector* function used to indicate where the original policy should defer instead of taking a decision.

6.3.1 Formalizing the Problem

We use the same notation introduced in Section 4.2. Our goal is to learn a policy $\pi : \mathbb{R}^m \to \mathcal{A}'$, mapping from specimen covariates to a recommended action. The action space $\mathcal{A}'$ consists of both available treatments and a 'defer' action. If the policy chooses to defer on a particular example, it falls back on the empiric prescription selected by the clinician for that case.

We introduce some useful terminology that will be used throughout this chapter. We define the **decision cohort** for a policy as the subset of examples where the policy makes a decision (i.e., does not defer), and the **coverage rate** of a policy as the proportion of examples for which the policy makes a decision. We now precisely define the objectives to be optimized for each of the settings discussed in Section 6.1.

Algorithm errors are costly.

We first consider the setting in which algorithm errors are costly. Here, our goal is to maximize the algorithm's performance in the decision cohort, subject to a constraint that the coverage rate exceeds some target value. We recall the definition of the composite reward function $r_\omega(a)$ introduced in Section 4.4. Before defining our objective, we first define the reward r_ω^{doc} for a clinician action a in this setting:

$$r_\omega^{\mathsf{doc}}(a) = r_\omega(a) + \alpha \cdot \mathbf{1}\left[a \text{ is suboptimal}\right], \tag{6.1}$$

where a suboptimal action is one that does not maximize r_ω. Intuitively, the α term introduces an asymmetric cost for suboptimal decisions taken by clinicians relative to the learned policies.

We define $h(x) \coloneqq \mathbf{1}\left[\pi(x) \neq \texttt{defer}\right]$, an indicator for whether the policy π makes a decision on an example with features x. Formally, we then wish to learn the policy π^* that is the solution to the following optimization problem:

$$\arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1 - h(x)) \cdot r_\omega^{\texttt{doc}}(a^{\texttt{doc}}(x))\Big] \text{ s.t. } \mathbb{E}[h(x)] \geq c, \tag{6.2}$$

where $a^{\texttt{doc}}(x)$ is the action selected by a doctor on an example with features x, and c is the desired lower bound on the coverage rate. We can re-write this optimization problem in its Lagrangian form to incorporate the lower bound on the coverage rate into a single objective:

$$\begin{aligned}
&\arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1 - h(x)) \cdot r_\omega^{\texttt{doc}}(a^{\texttt{doc}}(x)) + \lambda(h(x) - c)\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1 - h(x)) \cdot r_\omega^{\texttt{doc}}(a^{\texttt{doc}}(x)) \\
&\qquad\qquad\qquad + \lambda(h(x) \cdot (1 - c) + (1 - h(x)) \cdot (-c))\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot (r_\omega(\pi(x)) + \lambda \cdot (1 - c)) + (1 - h(x)) \cdot r_\omega^{\texttt{doc}}(a^{\texttt{doc}}(x) - \lambda \cdot c)])\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1 - h(x)) \cdot (r_\omega^{\texttt{doc}}(a^{\texttt{doc}}(x)) - \lambda)\Big]
\end{aligned}$$

The second line follows from the fact that $h(x)$ is an indicator variable, and the third line follows from rearranging terms. In the last step, we subtract the constant $\lambda \cdot (1 - c)$ from the objective function, which does not affect the identity of the optimal policy. We see that imposing a lower bound on the coverage rate modifies the optimization by shifting the rewards earned from choosing to defer downward by some fixed constant.

Algorithm interventions are costly.

We next consider the setting where algorithm interventions are costly. Here, we wish to optimize the performance of the learned policy π, subject to an upper bound on the coverage rate of π. Using the same notation introduced in the previous section, we wish to learn the policy π^* that is the solution to the following optimization problem:

$$\arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1-h(x)) \cdot r_\omega(a^{\mathsf{doc}}(x))\Big] \text{ s.t. } \mathbb{E}[h(x)] \leq c. \tag{6.3}$$

As in the previous section, we rewrite the objective in its Lagrangian form to incorporate the constraint on the coverage rate into the optimization problem and follow a series of similar steps:

$$\begin{aligned}
&\arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1-h(x)) \cdot r_\omega(a^{\mathsf{doc}}(x)) - \lambda(h(x) - c)\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1-h(x)) \cdot r_\omega(a^{\mathsf{doc}}(x)) \\
&\qquad\qquad\qquad - \lambda(h(x) \cdot (1-c) + (1-h(x)) \cdot (-c))\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot (r_\omega(\pi(x)) - \lambda \cdot (1-c)) + (1-h(x)) \cdot [r_\omega(a^{\mathsf{doc}}(x) + \lambda \cdot c)]\Big] \\
&= \arg\max_{\pi} \mathbb{E}_{x,r}\Big[h(x) \cdot r_\omega(\pi(x)) + (1-h(x)) \cdot (r_\omega(a^{\mathsf{doc}}(x)) + \lambda)\Big]
\end{aligned}$$

In contrast to the previous objective, we see that imposing an upper bound on the coverage rate shifts the rewards earned from choosing to defer upward by some fixed constant λ. In practice, we will choose the appropriate value for λ by examining the coverage rate achieved on a validation set.

6.3.2 Direct Policy Optimization with Deferral

We first present a direct approach to learning treatment policies with the option to defer, extending the approach presented in Section 4.5. Recall that our original direct learning method optimized the empirical estimate of the following convex surrogate to learn a value-maximizing policy:

$$\mathbb{E}_{x,r}\tilde{L}(f, x, r) \coloneqq -\mathbb{E}\left[\sum_{a \in \mathcal{A}} r(a) \log \frac{\exp f_a(x)}{\sum_{a'} \exp f_{a'}(x)}\right], \tag{6.4}$$

where each f_a is a linear function that produces a 'score' associated with selecting action a. To directly learn a policy that allows for deferral, we simply expand the action space to include an option corresponding to deferral and define the corresponding reward for deferring on each example. We directly extract the rewards for deferral in each setting from our development of the objectives in Section 6.3.1. In the setting where algorithm errors are costly, the reward for deferral is:

$$r(\texttt{defer}) = r(a^{\texttt{doc}}) + \alpha \cdot \mathbf{1}\left[a^{\texttt{doc}} \text{ is suboptimal}\right] - \lambda, \tag{6.5}$$

where α is a fixed positive parameter, and λ is a parameter that varies based on the desired coverage rate. In the setting where algorithm interventions are costly, the reward for deferral is:

$$r(\texttt{defer}) = r(a^{\texttt{doc}}) + \lambda, \tag{6.6}$$

where λ is again a positive parameter that depends on the upper bound on the coverage rate. Then, the new surrogate loss is given by:

$$\mathbb{E}_{x,r}\tilde{L_d}(f, x, r) := -\mathbb{E}\left[\sum_{a \in \mathcal{A}} r(a) \log \frac{\exp f_a(x)}{\sum_{a'} \exp f_{a'}(x)} + r(\texttt{defer}) \log \frac{\exp f_{\texttt{defer}}(x)}{\sum_{a'} \exp f_{a'}(x)}\right], \tag{6.7}$$

where the summations over a' are across all treatment options and the deferral action. As before, we can optimize the empirical estimate of this quantity in our training sample using gradient-based techniques. This surrogate remains convex and consistent for the Bayes-optimal solution to the objectives introduced in Section 6.3.1.

Conceptually, this approach is very simple, as it treats deferral no differently than any other treatment option and requires no modification to the original direct learning approach beyond the construct of a new reward. While it is limited to usage with the direct policy learning approach and cannot be used with the indirect methods, this is not a huge drawback in this setting given the comparable performance of direct and indirect approaches observed in Chapter 5.

6.3.3 Expected Reward Maximization

We next present an indirect approach to learning treatment policies with a deferral option, extending the expected reward maximization approach presented in Section 4.4. Recall that in our original approach, we first learned conditional treatment effectiveness models $f_a(x)$ predicting the probability that a patient will be susceptible to treatment a. For an example with features x, we then selected the reward-maximizing action, given by:

$$\arg\max_{a} r_\omega(x, a) = \omega \cdot f_a(x) + (1 - \omega) \cdot (1 - C(a)), \tag{6.8}$$

where $C(a)$ is an indicator for whether treatment a is a 2nd-line antibiotic. In order to incorporate deferral into this method, we also need to build an estimator of the reward for deferring on a given example.

We use the same reward definitions introduced in the previous section to learn functions $g : \mathbb{R}^m \to \mathbb{R}$ that predict the reward for choosing to defer. This function is learned from the empiric prescriptions (and corresponding rewards) selected by clinicians in the training set. Since the reward definition for deferral is different across the two settings we are considering, this estimator is also learned separately in each setting and for each value of the parameter λ.

The learned treatment policy is then given by:

$$\pi_\omega(x) = \begin{cases} \arg\max_a r_\omega(x, a) & \max_a r_\omega(x, a) > g(x) \\ \texttt{defer} & \text{otherwise} \end{cases}. \tag{6.9}$$

In words, we only choose to defer if the estimated reward for deferring predicted by g is greater than the reward-maximizing action among the available treatment options; otherwise, we stick with the original reward-maximizing treatment.

6.3.4 Deferring with a Rejector

Finally, we present a rejector-based approach to deferral to serve as a baseline for the previous two methods. This approach does not use a notion of a 'reward' for deferral, and hence does not seek to directly optimize the objectives presented in Section 6.3.1. Instead, we first learn a treatment policy without a deferral option using the methods presented in Chapter 4, then use a *rejector* function to convert some of the policy's original decisions to deferral. This post-processing stop requires a model of the cases where the algorithm should defer to achieve a particular objective.

Formally, we first learn a treatment policy $\pi : \mathbb{R}^m \to \mathcal{A}$, mapping from specimen features to a treatment option (note that this does not include a deferral option). We then learn a separate *rejector* $h : \mathbb{R}^m \to [0, 1]$, which maps from specimen features to a score quantifying the preference for deferral on this example. The method used to learn h depends on the particular setting and the corresponding deferral objective.

Finally, we create a new policy $\pi' : \mathbb{R}^m \to \mathcal{A} \cup \{\texttt{defer}\}$ that defers on specimens x where $h(x) > t$, where t is a probability threshold chosen from the training set based on the desired coverage rate, and takes action $\pi(x)$ otherwise. Higher values of t correspond to the treatment policy making decisions on larger proportions of the population (i.e, a higher coverage rate).

We next propose targets that can be used to learn the rejector function h in each of the two settings we have introduced. In settings where algorithm errors are costly, we train the rejector to predict a binary label indicating whether the clinician decision resulted in a suboptimal treatment (i.e., if treatment resulted in IAT or was an unnecessary usage of 2nd-line treatments). Intuitively, this rejector pushes the algorithm to defer on cases where clinicians make mistakes, since it incurs a much lesser cost than an algorithm error on the same instance.

In contexts where algorithm interventions are costly, we train the rejector to predict a binary label that is 1 for instances where either (1) the clinician decision was optimal or (2) the decision of the original policy π was suboptimal. Learning such a rejector leads the final policy to make decisions on cases where the algorithm takes

Setting	Target	Description
Errors are costly	Clinician suboptimality	Indicator for suboptimal clinician decision
Interventions are costly	Clinician/algorithm parity	Indicator for whether clinician decision was at least as good as algorithm decision

Table 6.1: Targets used for learning rejectors in settings where algorithm errors or interventions or costly.

an optimal decision, but the clinician does not, incentivizing the model to avoid unnecessary interventions where it cannot improve on clinician selections. These targets are summarized in Table 6.1.

Rejector-based approaches can be used with any of the policy learning methods presented in Chapter 4, as the training procedure for h does not depend on how π was learned. These approaches are also highly interpretable, as one can analyze the learned rejector to understand the features driving the model to defer. In practice, however, h may be extremely difficult to learn and require a more complex model class with low interpretability to be effective. Furthermore, the policy is learned separately from the rejector, and is trained to achieve good performance across the full cohort; it does not adapt to the rejector to perform better in the parts of the population where it actually ends up making a decision.

6.4 Experiments

6.4.1 Setup

Direct policy learning with deferral

We learn policies via the direct method for several trade-offs between IAT and 2nd-line usage rates. In both settings, training proceeds in two stages. We first train the model without allowing for deferral (i.e., excluding the corresponding reward from the loss function) until we have learned a good policy that only selects from among the four antibiotics. We then add the reward for deferral into the loss and continue training, using an early stopping criteria based on the validation reward.

By initializing the model with a good policy that does not defer using the first training phase, we mitigate overfitting to a particular small subset of the population where the algorithm ends up making decisions. We use the same train/validation sets for tuning hyperparameters (e.g., learning rate, regularization) as described in Section 5.1.3 for direct policy learning without deferral.

We now describe how to select actions given a policy trained via direct learning. In the previous chapter, we derived a policy π from the trained policy model by selecting the action with maximum score in the predicted policy distribution for a given example. In the setting where algorithm interventions are costly, we obtain the final policy in the same way. However, in the setting where algorithm errors are costly, we find that we need to slightly modify the way we derive a policy from the policy model to achieve good performance within the decision cohort.

For an example with features x, the policy model produces a distribution over the available actions, including deferral; let $d(x)$ be the predicted probability corresponding to the deferral action in this distribution. For a given coverage rate c, we choose a threshold t and defer on all examples where $d(x) > t$; otherwise, we select the action with the maximum value among the remaining options in the policy distribution (i.e., the highest value among the actual treatment options). The threshold t corresponds to the probability threshold that would yield a coverage rate of exactly c using this procedure on the training set. We found that this approach empirically yielded better results on the validation set for the setting where algorithm errors are costly, but did not yield much benefit for the setting where algorithm interventions are costly, and thus do not use it in the latter setting.

In the setting where algorithm errors are costly, we train models using only a single value of λ for each value of ω and construct policies for various coverage rates as described in the previous paragraph. In the setting where algorithm interventions are costly, we train models for several values of λ for each values of ω to learn policies for different coverage rates.

Expected reward maximization

We learn conditional outcome models for treatment effectiveness using logistic regression models, tuning hyperparameters as described in Section 5.1. We parameterize the deferral reward predictor g with a regularized linear regression model, and tune regularization strength using the same train/validation splits used for tuning the models of treatment resistance. As with the direct approach, we learn policies through the expected reward maximization approach for several values of λ and ω, re-fitting the predictor for the reward earned for deferral for each parameter setting.

Deferring with a rejector

We first learn policies without deferral for a variety of values of ω using the direct learning approach. For each of the two deferral settings, we then learn rejectors $h(x)$ to predict the corresponding targets described in Section 6.3.4. In each case, the rejector is parameterized by a logistic regression model, and hyperparameters are tuned using the same train/validation splits used for training the original policy.

In this approach, we are able to directly control the coverage rate by modifying the rejector threshold t. For a given target coverage rate c on the test set, we choose t to be the threshold that produces a coverage rate of exactly c on the training set.

Evaluation

We evaluate learned policies according to slightly different criteria across the two deferral settings we have outlined in this chapter. In the setting where algorithm errors are costly, we examine the algorithm performance in the decision cohort for several lower bounds on the policy coverage rate and for a variety of trade-offs between the objectives of IAT and 2nd-line antibiotic usage. In the setting where algorithm interventions are costly, we examine the overall performance of the model (*not* just on the decision cohort) for several upper bounds on the coverage rate and a variety of trade-offs between the two objectives.

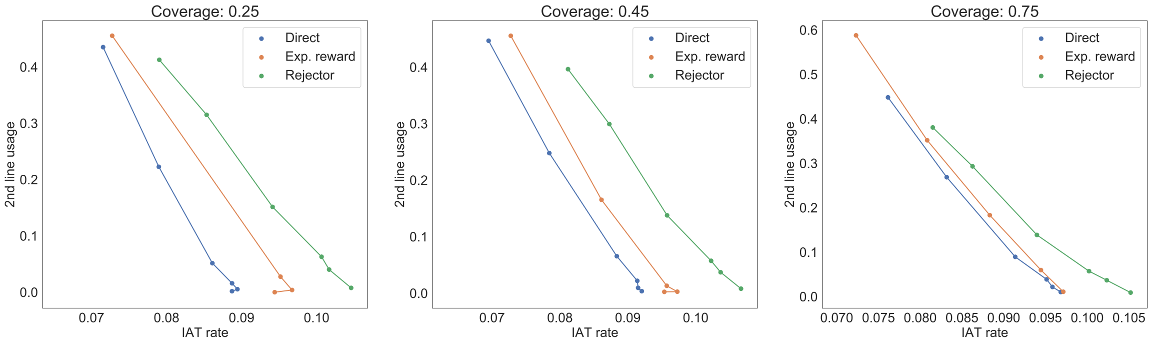


Figure 6-1: Performance frontiers of policies learned via direct learning, expected reward maximization, and the rejector-based approach on the test cohort at coverage rates of 25% (left), 45% (middle), and 75% (right). Each point represents the combined performance of the policy on the decision cohort for a different setting of the reward parameter ω.

Coverage	**25%**		**45%**		**75%**	
Method	**IAT**	**2nd-line**	**IAT**	**2nd-line**	**IAT**	**2nd-line**
Direct learning	8.9%	0.5%	9.1%	2.2%	8.3%	26.8%
Expected reward	9.4%	0.0%	9.2%	4.2%	8.1%	35.2%
Rejector-based	10.5%	0.8%	10.2%	5.7%	8.6%	29.3%

Table 6.2: Comparison of algorithm performance in the decision cohort across methods at various lower bounds on the coverage rate for selected trade-offs between IAT and 2nd-line usage.

6.4.2 Results

Minimizing algorithm errors

We first examine the performance of these approaches in the setting where we aim to minimize algorithm errors in the decision cohort. In Figure 6-1, we plot the performance frontier of policies obtained from each approach for several lower bounds on the desired coverage rate of the learned policies. Each point represents the performance of a policy within its corresponding decision cohort. In Table 6.2 we compare the performance of the three approaches in the decision cohort at coverage rates of 25%, 45%, and 75%

Across all coverage rates, the expected reward and direct learning approaches

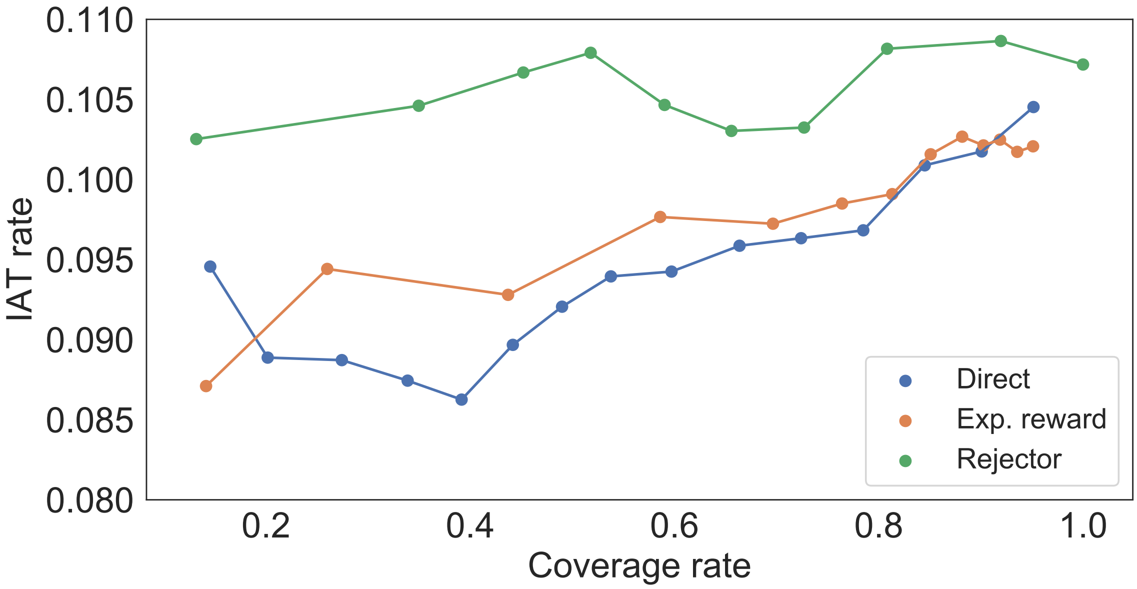


Figure 6-2: Comparison of policy IAT rates within decision cohort across a wide range of coverage rates in the setting where algorithms errors are costly. The policies depicted here use almost no 2nd-line treatments. The direct approach outperforms the expected reward and rejector-based approach across a range of coverage rates between 25% to 70%.

outperform the rejector-based approach. At higher coverage rates (e.g., 75%), the direct and expected reward approaches achieves similar performance in the decision cohort across a wide range of trade-offs between outcomes, as shown in the rightmost panel of Figure 6-1.

However, at lower coverage rates, policies learned via the direct learning approach achieve superior performance to those learned via the expected reward approach within the decision cohort. In the left and middle panels of Figure 6-1, we can also see that the gap in performance frontiers between the two approaches widens as the coverage rates is reduced from 45% to 25%, suggesting that the direct approach is able to better learn a policy specific to a small decision cohort, while the expected reward approach fails to do so.

In Figure 6-2, we provide an alternative view comparing the performance of these approaches. We plot IAT rates (within the decision cohort) of policies derived from each method across various coverage rates. The policies selected for this plot use essentially no 2nd-line treatments, so those values are not shown here. We can clearly see that the direct learning approaches outperforms the expected reward approach across a wide range of coverage rates from roughly 25% to 70%. As the required coverage rate is reduced from 90% to 40%, the decision cohort IAT rate of policies learned via the direct approach decrease from 10.2% to 8.6%, a relative reduction of over 15%. While this is a significant reduction in errors within the decision cohort, the learned policies are unable to identify a subpopulation where they can achieve close to perfect performance. Identifying such a subgroup appears to be extremely difficult to do in this particular setting.

Overall, however, the results of this section suggest that the direct approach is able to take advantage of jointly learning a good treatment policy and where to defer, particularly at intermediate coverage rates, and is able to outperform alternate approaches with respect to errors in the decision cohort.

Minimizing algorithm interventions

We next examine the performance of these approaches in the setting where algorithm interventions are costly. In Figure 6-3, we plot the performance frontier of policies learned via each approach for a few different upper bounds on the allowed coverage rate; each point on this plot represents the combined performance of clinician and algorithm decisions across the full test cohort. In Table 6.3, we compare the performance of a few of the policies depicted in Figure 6-3 for various trade-offs between IAT and 2nd-line usage.

The expected reward and direct learning approaches generally achieve similar performance across several coverage rates. There is some variation in performance - for instance, the expected reward approach achieves slightly better performance than direct learning at a coverage rate of 50%, while the reverse is true at a coverage rate of 75% - but neither approach appears to be significantly better than the other. At higher coverage rates (e.g., $> 50\%$), these two approaches produce policies that achieve minor improvements in performance over the rejector-based approach.

We can also examine the improvement in performance that the algorithm achieves over clinicians within the decision cohort for a given coverage rate. A larger improve-

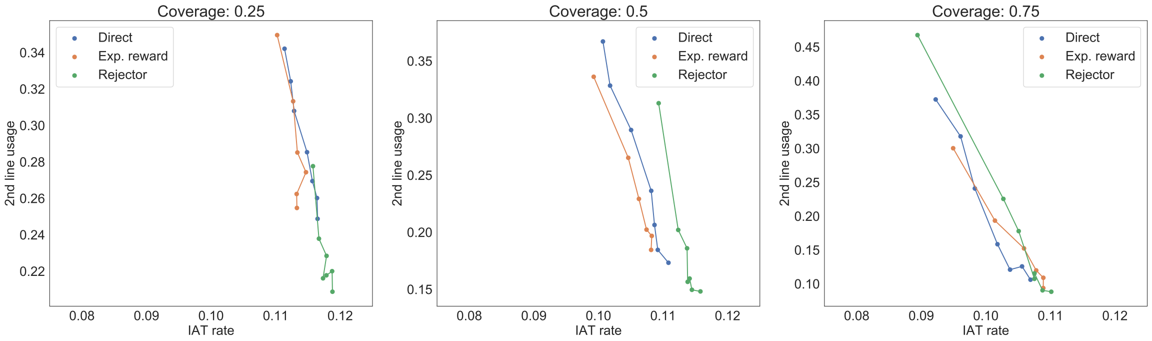


Figure 6-3: Performance frontiers of policies learned via direct learning, expected reward maximization, and the rejector-based approach on the test cohort at coverage rates of 25% (left), 50% (middle), and 75% (right). Each point represents the combined performance of clinicians and policy on the test cohort for a different setting of the reward parameter ω.

ment in the decision cohort leads to an improvement in combined performance of clinicians and algorithm across the entire cohort. In Table 6.4, we compare the improvement in the decision cohort for the policies shown in Table 6.3 at a coverage rate of 50% . The expected reward and direct learning approaches reduce the IAT relative to clinicians by about 10%, while reducing the IAT by over 80%. The rejector-based approach yields a relative improvement in IAT of around 12% while lowering 2nd-line usage by 70%. While the expected reward and direct learning approaches exhibit a larger improvement over clinicians in the decision cohort, these improvements translate into only minor improvements in overall performance, as seen in Table 6.3.

We note that the decision cohorts identified by all three of these policies consist of cases where clinicians achieve below average performance, as clinician decisions lead to a 11.9% IAT with 33.6% 2nd-line usage on the full test cohort. Our methods are successful at identifying populations where clinicians are less effective and targets decision-making toward these groups.

6.5 Summary

In this chapter, we examined how learning treatment policies with deferral may be useful in settings where either algorithm errors or interventions are costly. We pre-

Coverage	**25%**		**50%**		**75%**	
Method	**IAT**	**2nd-line**	**IAT**	**2nd-line**	**IAT**	**2nd-line**
Direct learning	11.5%	26.2%	10.9%	20.6%	10.4%	12.1%
Expected reward	11.5%	27.4%	10.7%	20.2%	10.8%	12.0%
Rejector-based	11.6%	27.8%	11.2%	20.2%	10.8%	11.5%

Table 6.3: Comparison of combined performance of algorithm and clinicians across test cohort across methods at various upper bounds on the policy's coverage rate for select trade-offs between IAT and 2nd-line usage.

	Algorithm		**Clinician**	
Method	**IAT**	**2nd-line usage**	**IAT**	**2nd-line usage**
Direct learning	10.9%	6.1%	13.2%	33.9%
Expected reward	11.0%	4.8%	13.3%	32.1%
Rejector-based	10.4%	11.4%	11.9%	40.2%

Table 6.4: Comparison of clinician and algorithm performance in the decision cohort across methods at a coverage rate of 50% for the learned policies with performance shown in Table 6.3.

sented three approaches for learning treatment policies with a deferral option in these settings. The first two approaches are extensions of the direct learning and expected reward maximization techniques introduced in Chapter 4, and use the notion of a 'reward' for choosing to defer to a clinician to learn an effective policy. The last approach involves learning a rejector function that identifies cases where a policy originally trained without a deferral option should forgo its initial decision.

In settings where algorithm errors are costly, the direct learning approach outperformed the expected reward and rejector-based approaches across a wide range of coverage rates, learning policies that achieved better performance within the subset of examples where the policy makes a decision. In settings where algorithm interventions are costly, the expected reward and direct learning approaches achieved slightly better performance than the rejector-based approaches at higher coverage rates, but the improvements observed here were all relatively minor.

While we evaluated our approaches on the task of antibiotic prescription, the methods presented here can be easily adapted to any medical decision-making scenario, helping to ease some of the difficulties associated with integrating the decisions of clinicians and newly deployed decision support algorithms. In the next chapter, we examine additional difficulties in deploying such tools into clinical settings.

Chapter 7

Towards a Deployable Decision Support Tool

Deploying machine learning-based clinical decision support (CDS) systems into real-world clinical settings is challenging. These systems have to integrate smoothly with existing clinical workflows and a hospital's digital infrastructure while preserving high standards of patient safety. In this chapter, we outline a CDS tool for empiric antibiotic prescription with the ultimate goal of easy and successful deployment across multiple hospitals. We first identify common critiques and failure modes of deployed CDS systems and propose desirable properties and features to mitigate these issues. We then outline strategies to achieve these properties in the antibiotic prescription setting. We derive a simplified - but highly effective - treatment policy that is more straightforward to deploy than policies discussed in previous chapters. We then examine the limitations of our proposed CDS tool in the presence of data missingness and treatment contraindications, and propose strategies for exposing these drawbacks to clinicians in a transparent way.

7.1 Deployment of Decision Support Tools

Advances in machine learning algorithms have dramatically improved performance in a variety of important medical decision-making tasks, but several studies have shown that integrating these algorithms into regular clinical workflows as decision support tools has been a challenge. Several common reasons have been identified for these deployment failures.

Clinicians consistently point to a lack of consideration for the human-computer interaction (HCI) aspects of CDS systems as the primary reason for lack of adoption [52]. They complain of the excessive computerized alerts introduced by these tools and the obtrusive integrations into clinical workflows. Support tools are often designed under the assumption that clinicians will identify situations in which they need assistance and make the effort to use an additional system to help them make a better decision. In practice, this is rarely the case.

Most machine learning-based CDS tools are also black boxes to clinicians, who may not have a clear understanding of the inputs being used by the model or the general underlying mathematical logic used to arrive at a decision [20]. Clinicians are unlikely to trust a system that does not provide a clear justification for its decisions, particularly in high-stakes situations. This *local* interpretability of the model is especially important in situations where the clinician's decision is significantly different from that of the CDS tool.

Clinicians also often desire *global* insights into the behaviors of the support tool, including a general description of its known strengths and weaknesses or examples of specific edge cases where the model may provide incorrect decisions [4]. Since patient safety is paramount in most settings, doctors need to be able to precisely identify situations in which they should down-weight the importance of an algorithm's suggestion and resort to their own clinical judgment.

Based on these common failure modes and clinician critiques, we can highlight a few desirable properties of CDS tools that are deployed into real-world settings. First, they must be embedded as much as possible within the existing clinician decision-making process without introducing an additional mental burden, but must be visible enough that they can still positively impact decision-making. Second, the models must either provide a highly interpretable justification for its treatment selection whenever it chooses to provide input to clinicians, or should be simple enough that clinicians can form a clear mental model of the attributes used by the model to make a decision. Finally, the tool should provide ways of flagging cases where there may be relevant information that it has *not* accounted for in its decision-making process, indicating where clinicians may need to examine additional context to make a decision.

In this chapter, we propose a CDS tool for antibiotic prescription with these desirable properties; namely, we would like the tool to impose a limited burden on clinicians while also being highly interpretable and transparent to clinicians regarding its own limitations. We have already made progress towards achieving some of these properties through the development of our methods in previous chapters:

- **Limiting mental burden on clinicians**. One approach to limit the burden of a decision support tool is to constrain the number of alerts it can produce. In Chapter 6, we developed methods to learn policies with the ability to defer to clinicians in settings with a constraint on the number of decisions that could be provided by the algorithm. Such a constraint could be specified *a priori* by individual clinicians to tailor policies and the volume of interventions to each physician's personal preferences, and could help mitigate the issue of excessive alerts in a systematic way.

- **Interpretability.** In Section 5.4.2, we showed how to extract features important for the selection of one treatment over another from a policy learned via the direct approach. For a given treatment decision, a decision support tool could identify and display the subset of these important features that are present in the current case and contributed to the model's decision to choose a particular treatment over other options. This would provide clinicians with clear insights into the reasons for particular choices made by the algorithm for each patient.

In the rest of this chapter, we describe additional strategies to design a CDS tool for antibiotic prescription that is interpretable, portable, and transparent. We first derive an effective treatment policy that uses fewer than 20 features; such a policy is more likely to be interpretable to clinicians and more portable than models based on hundreds of features. We then describe two important edge cases where model decisions may fail - unrecorded patient data and treatment contraindications - and propose changes to the design of a support tool to mitigate these potential failure modes and expose them to clinicians in a transparent manner.

7.2 A Simplified Treatment Policy

Deployed CDS tools should make decisions that are interpretable to clinicians while also being portable enough for easy integration across a variety of hospitals and clinical settings. A tool that depends on a simple underlying model requiring only a small number of features, while still preserving most of the effectiveness of a much larger model, achieves both these goals.

If a deployed CDS tool depends on a large model, clinicians may not be able to build an effective mental model of the system's decisions, even if an interface provides explicit indications of the pieces of information driving certain choices on a case-by-case basis. A smaller model may allow the clinician to have more trust in their understanding of the algorithm's decisions, and allow them to incorporate additional context to override recommendations if needed.

Large models are also more difficult to deploy across a wide range of settings. The necessary data inputs for such models may not be available at all hospitals, and conventions for naming and storing information may also vary, requiring system specialization at each deployed location. If the learned model only consists of a small set of features being combined in a relatively simple way, this would significantly reduce the effort needed for widespread policy deployment. In this section, we derive such a simple policy for empiric antibiotic prescription in uncomplicated UTIs and evaluate it against policies learned using much larger feature sets.

Our proposed policy consists of two steps that are relatively intuitive and highly interpretable. In the first step, we use a small linear model to identify whether the clinician should prioritize 1st- or 2nd-line treatment for a given patient and construct a corresponding treatment preference order. In the second step, we recommend the first treatment in the preference order to which the patient does not exhibit a strong

Antibiotic	**Resistance Indicators**
NIT	Any prior resistance to NIT
SXT	Any prior resistance to SXT Folate inhibitor exposure in past 30 days
CIP	Any prior resistance to CIP / LVX Fluoroquinolone exposure in past 30 days

Table 7.1: Features used as strong indicators for each of the treatments used in simple treatment policy.

indicator of resistance. In more detail, the steps are:

1. **Selection of treatment preferences**. We learn a function $f : \mathbb{R}^d \to [0, 1]$, mapping from specimen features to a value quantifying the preference for 2nd-line usage. Here, f is a highly regularized linear model trained to predict the 2nd-line usage of a policy previously learned on all features.

   When making a treatment decision, if $f(x) > t$ for a chosen threshold t and specimen x, then the policy prioritizes 2nd-line usage, and the preference order is given by: CIP $<$ NIT $<$ SXT. Otherwise, the policy prioritizes 1st line usage and the preference order is given by: NIT $<$ SXT $<$ CIP. [1] The threshold t controls the IAT vs. 2nd-line usage trade-off; a higher t results in a policy with lower 2nd-line usage. Our preference order prioritizes NIT over SXT in both cases due to the significantly lower resistance rates to NIT in our population.

2. **Treatment selection**. We select the first agent in the preference order from step 1 to which the patient does not have a 'strong indicator' of resistance. These indicators are derived from inspection of the most important features in the predictive models of resistance learned for each of these treatments.

[1] For simplicity, we do not include LVX in these preference lists, as treatment with CIP has the same outcome as LVX in the vast majority of uncomplicated UTIs.

In Section 5.5, we found that direct policies learned from training on all UTIs outperformed policies trained on only uncomplicated UTIs, so we use these policies as the starting point for extracting our simplified policy. We obtain the model used in step 1 of the policy by fitting a highly regularized logistic regression model to predict the 2nd-line usage of a direct policy trained on all features with $\omega = 0.92$. This linear model has **18** features with nonzero coefficients, listed in Table A.24 with the corresponding coefficient values.

Nearly half the 18 features in this model are directly associated with prior exposure or resistance to one of the 4 antibiotic treatment options in this setting. For instance, prior resistance to NIT or CIP/LVX drives the policy to prefer 2nd line or 1st line treatments, respectively. Interestingly, prior resistance to SXT actually drives the policy to prefer 1st-line rather than 2nd-line treatment as one might expect. Further inspection of our cohort shows that this group of patients also has extremely high resistance rates to 2nd-line treatments, comparable to their resistance rates for NIT. Since the likelihood of IAT is comparable regardless of whether NIT or a 2nd-line treatment is given, the policy learns to prioritize 1st-line treatment for these individuals. The remaining features in this model are a mixture of other prior antibiotic exposures, colonization pressure, demographics, and location features.

The resistance models trained on all UTIs also achieved better predictive performance than the models trained on only uncomplicated UTIs (Section 5.5). We thus extract the strong indicators of resistance for each antibiotic by inspecting the top 5 most predictive features of resistance in these models (Tables A.16, A.18, A.20). These features are listed in Table 7.1.

We evaluate the performance of this simplified policy on the test cohort of uncomplicated UTIs, and compare its performance to policies trained with all features using the direct learning and expected reward maximization approaches. The full performance frontiers are shown in Figure 7-1, and Table 7.2 contains comparisons for policies at a few points along this trade-off frontier. Despite using fewer than 20 features, the simple policy achieves comparable performance to both the direct learning and expected reward approaches trained on all UTIs for a wide range of trade-offs

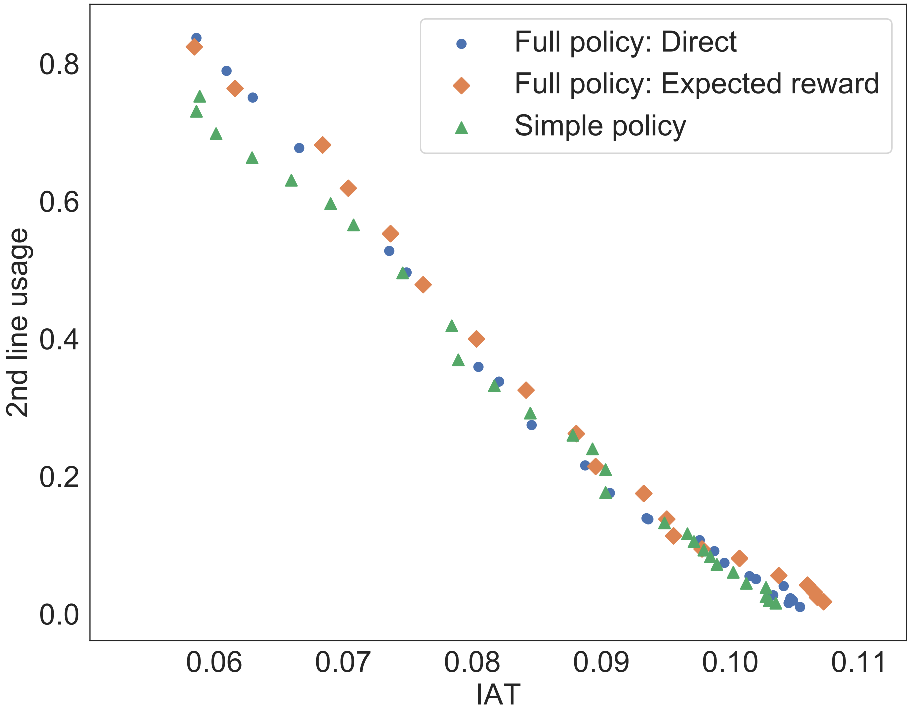


Figure 7-1: Comparison of simplified policy performance on the test cohort for several values of threshold t against policy sets learned via direct learning and expected reward maximization approaches on the dataset of all UTIs.

between IAT and 2nd-line usage, and even achieves slightly better performance with respect to IAT in the extremely low 2nd-line treatment regime.

We also evaluated our simplified policy at a trade-off that uses clinically reasonable levels of 2nd-line treatment against clinicians and the adjusted IDSA guidelines introduced in Chapter 5 (Table 7.3). We chose the threshold t in the first step of the simple policy to constrain 2nd-line usage in the validation set below 10%, and set the parameter p in the adjusted guidelines to constrain validation 2nd-line usage at the same level. The resulting policy reduces IAT by nearly 20% and 2nd-line usage by 70% relative to clinicians, and reduces IAT by over 10% relative to the adjusted guidelines with only a minor increase in 2nd-line usage.

We perform an in-depth analysis of our policy and guidelines at the trade-offs in Table 7.3 to better understand the gains made by our simplified policy. We produce breakdowns of the policy and guideline decisions as a function of the clinician decision as described in Section 5.3. These are shown in Figures 7-2 and 7-3, respectively.

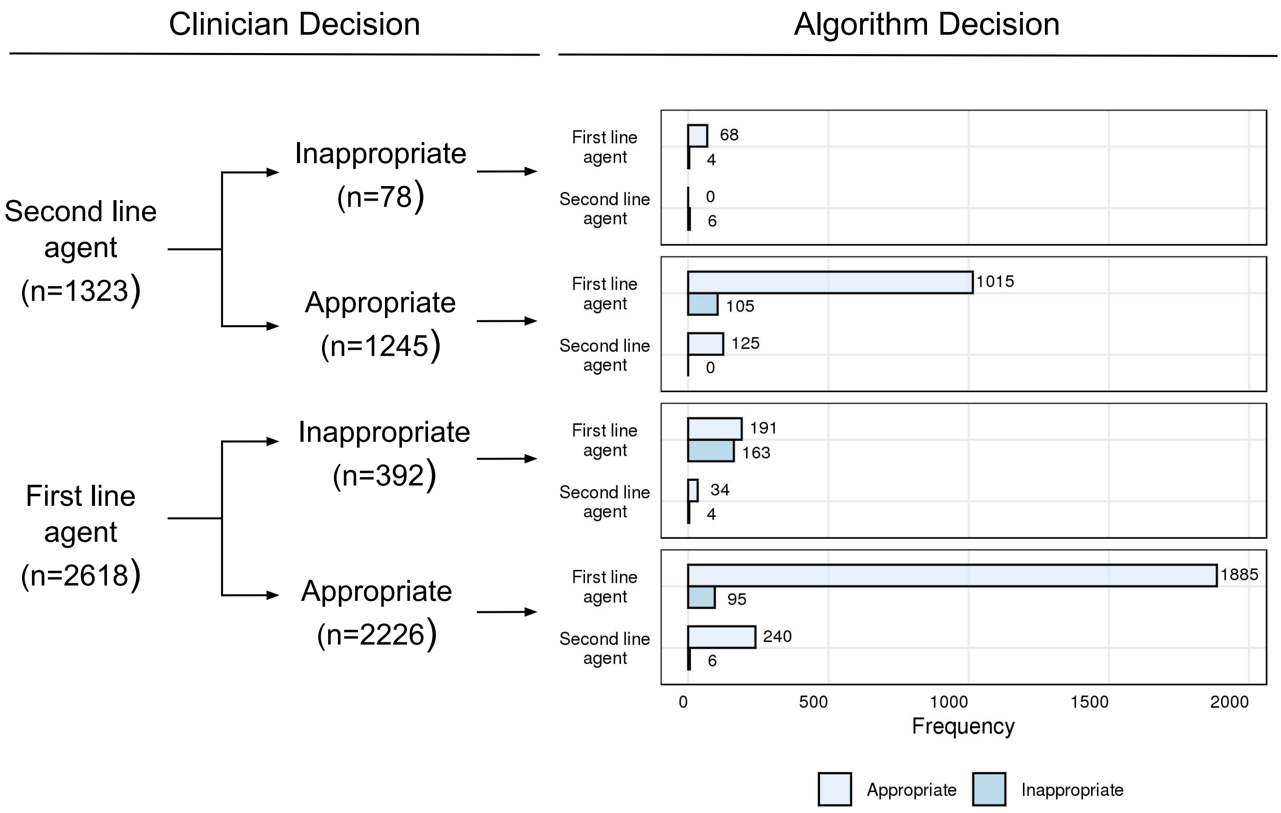


Figure 7-2: Breakdown of decisions made by simplified policy at the trade-off in Table 7.3 relative to clinician decisions.

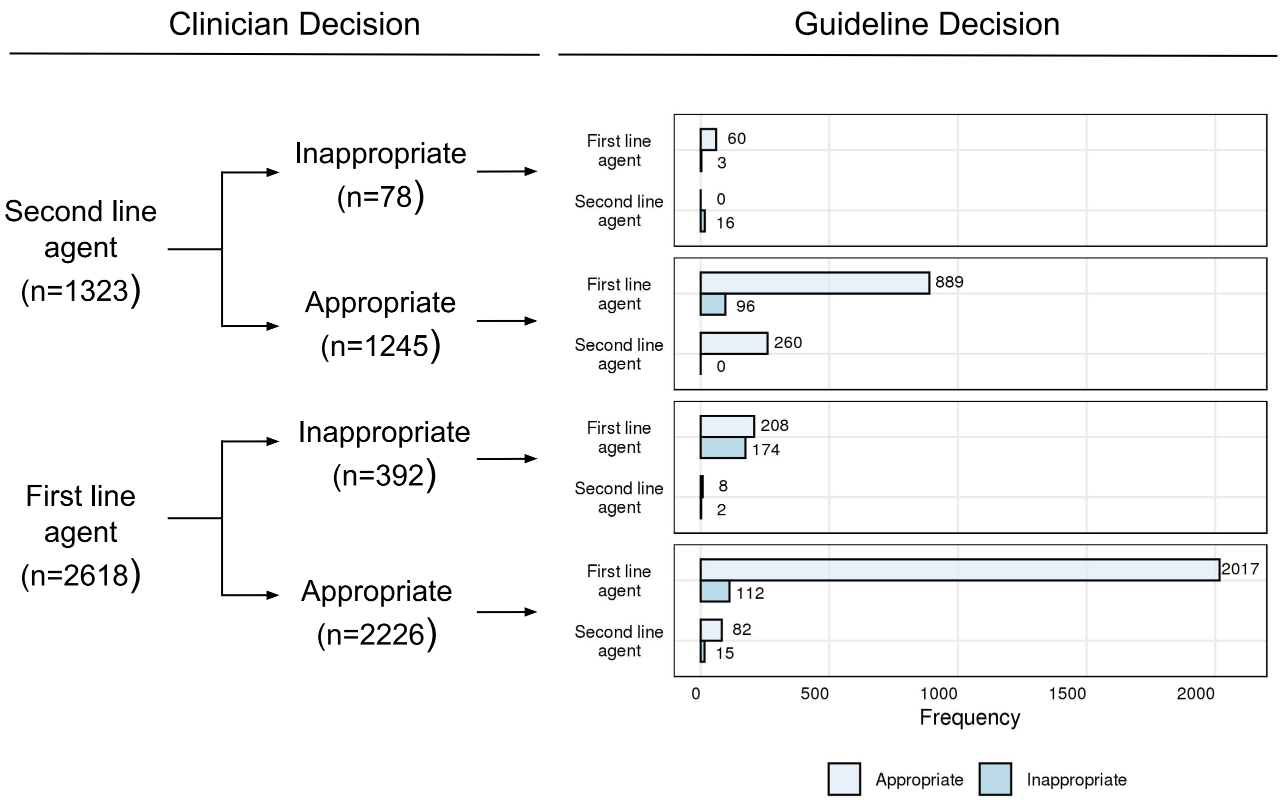


Figure 7-3: Breakdown of decisions made by adjusted IDSA guidelines at the trade-off in Table 7.3 relative to clinician decisions.

	IAT	2nd-line	IAT	2nd-line
Simplified policy	9.5%	13.2%	10.1%	4.5%
Expected reward	9.5%	13.8%	10.6%	4.2%
Direct optimization	9.4%	13.9%	10.4%	4.1%

Table 7.2: Comparison of primary outcomes for simplified policy relative to learning models with all available features at two points along the trade-off frontier.

	IAT	2nd-line
Simplified policy	9.7%	10.5%
Clinicians	11.9%	33.6%
Adjusted guidelines	10.8%	9.7%

Table 7.3: Comparison of primary outcomes for simplified policy relative to clinicians and modified version of IDSA guidelines.

We find that our policy improves upon guideline performance in a few ways. First, our policy is better at identifying appropriate 1st-line treatments for patients empirically treated with an appropriate 2nd-line agent - we do this successfully in 82% of cases where clinicians gave appropriate 2nd line treatment, while the guidelines only do this in 71% of cases. We are also better at identifying appropriate 1st-line treatments for cases where clinicians gave *inappropriate* 2nd-line treatment - we do this successfully in 87% of cases where clinicians gave inappropriate 2nd line treatment, while the guidelines only do this in 77% of cases.

Finally, our policy less frequently switches an effective empiric treatment given by clinicians to an ineffective treatment - our policy only does this in 5.9% of cases, while the guidelines do this in 6.4% of cases. However, we note that our policy is slightly worse at switching inappropriate 1st-line treatment decisions by clinicians to appropriate 1st-line treatments - we do this in 49% of cases, while the guidelines do it in 53% of cases.

These results suggest that only a small number of features are necessary to deploy an antibiotic prescription policy that achieves dramatic improvement over current clinician performance and practice guidelines. Given its simplicity, the policy proposed in this section does not require a sophisticated implementation, and could likely be easily embedded into any clinical workflow without significant trouble.

7.3 Handling Unrecorded Patient Data

So far, we have treated the EHR as a ground truth source of information. In reality, it is a virtual certainty that some data will be missing from this structured record. In some cases, the missing information may only be present in a patient's record of clinical notes, and may provide critical information for making an appropriate treatment decision. When a treatment policy is deployed, producing a treatment recommendation without accounting for these missing pieces of information could mislead a clinician - who may be more likely incorporate information from previous notes into their decision - and reduce trust in the system. We wish to better understand the significance of the data missingness problem in our setting and develop strategies for handling these cases in a CDS tool.

To get a sense for the importance of missingness in our data, we attempted to recover prior exposure and resistance information for the four antibiotic treatments considered in this work, as we have clearly identified these as the most relevant features for producing a treatment decision. If needed, the approach described in this section could easily be adapted to identify mentions related to other features in the notes. We take the following approach to extract unidentified prior exposures:

1. **Identify relevant terms**. We construct a linear bag-of-words (BoW) model that uses a patient's past history of clinical notes to predict whether the structured record indicates a prior exposure to a specific antibiotic. Analyzing terms with the most positive coefficients in this model suggest words associated with exposures to this antibiotic.

2. **Extract potential unrecorded exposures**. We extract the notes that do *not* have a record of prior exposure to the antibiotic of interest, but do contain

Antibiotic	Terms	Train	Test
NIT	macrobid, nitrofurantoin	0.8%	0.5%
SXT	bactrim	3.5%	4.2%
CIP	cipro, ciprofloxacin	5.5%	3.2%
LVX	levofloxacin, levaquin	1.3%	0.4%

Table 7.4: Terms used to identify candidates for unrecorded antibiotic exposures and proportion of cases identified as having unrecorded exposures in train and test cohorts.

the relevant terms identified in step 1 within their notes.

3. **Filtering**. The relevant terms can occasionally be used in contexts not referencing a prescription, such as an allergy to that medication. We perform a simple filtering step using string matching to remove these mentions.[2]

We examined around 15 of the notes flagged by this process and successfully verified that they actually referenced a previously unrecorded treatment. However, we do note that this process is imperfect and may result in some false negatives (i.e., fails to detect additional unrecorded exposures). This is just one approach to identifying these unrecorded pieces of data, and one could imagine building more sophisticated clinical entity extraction techniques to recover this information. Table 7.4 lists the terms used to identify potential unrecorded antibiotic exposures, along with the proportion of specimens in the training and test sets for which exposures were identified. Exposures to SXT and CIP were most frequently unrecorded in the EHR, but in general, there are relatively few instances of missing data for exposures to these agents.

We repeated this process to identify unrecorded mentions of prior resistance to these agents, but were unable to find any such mentions, suggesting that the record of microbiological tests in the EHR is essentially comprehensive.

We examined the impact of modifying features to include these previously unrecorded exposures on our simplified policy and the modified IDSA guidelines at the

[2] We note that allergies are important for making treatment decisions, but are distinct from treatment resistance. We address mentions of allergies explicitly in Section 7.4.

	Recorded data only		W/ unrecorded exposures	
	IAT	2nd-line	IAT	2nd-line
Simplified policy	9.7%	10.5%	9.7%	10.3%
Adjusted guidelines	10.8%	9.7%	10.7%	10.2%
Clinicians	11.9%	33.6%	11.9%	33.6%

Table 7.5: Comparison of primary outcomes for simplified policy relative to learning models with all available features at two points along the trade-off frontier.

trade-off described in Section 7.2 (Table 7.5). The impact of these updated features is minimal, but does appear to produce a slight improvement in the 2nd-line usage of the simplified policy. This relatively minor improvement is unsurprising given the small proportion of examples with missing exposure data in our test set.

These results suggest that data missingness is not a significant problem in developing an effective antibiotic prescription tool for the hospitals in our dataset, but it is likely that data missingness will play a much more important role in settings where the EHR record is not as complete. It is important to have mechanisms for recovering this data and potentially flagging them for clinician review. Notes which contain terms known to be relevant to the current treatment decision can provide clinicians with a starting point to review appropriate portions of a patient history, saving them the trouble of manually sorting through a lengthy history of notes to identify useful information. A deployed tool could also allow clinicians to update the structured record with previously missing data based on a review of the notes flagged by the system, helping further mitigate the data missingness problem for future models.

7.4 Treatment Contraindications

Policies deployed in clinical settings also need to provide mechanisms for identifying and accounting for a patient's treatment *contraindications*, pieces of information in a patient's medical record that prohibit certain treatments. In the antibiotic prescription setting, the most common and important contraindications are patient allergies to specific antibiotics (note that this is *not* the same as resistance to an antibiotic). Allergy information is not readily available in the structured portions of our data, but is generally recorded by clinicians in their notes when recording a patient's treatment plan. In this section, we first examine the volume of contraindications in our patient cohort and examine whether our policies continue to exhibit improvements over clinicians after accounting for them. We then provide suggestions for how a deployed support tool can integrate information about contraindications when providing recommendations to clinicians.

We extracted these relevant contraindications from available notes and examine the effect on the decisions of our learned policies. We particularly focused on: (1) *sulfa allergies*, which prohibit treatment with SXT (a sulfa-based antibiotic) and (2) *pyelonephritis*, a UTI involving kidney infection which are generally treated with 2nd-line agents. Note that when constructing the uncomplicated UTI cohort, we initially aimed to filter out patients with this condition (Section 3.3); however, it is likely that some cases with this diagnosis were not explicitly recorded in the EHR and slipped through our filters. These are by far the most common contraindications in our data.

We extracted sulfa allergies by identifying patients with at least one note in their history containing the term **sulfa** and an allergy-related term (e.g., allergies, allergic) in close proximity to one another in the text. Many notes have an "Allergies" section that records this information in an organized manner, but doctors also reference allergies in a variety of other ways, so we found this simple approach to be most effective. We also filter out references containing negation terms (e.g., no, never) immediately prior to the sulfa allergy reference, as clinicians sometimes specifically highlight the lack of an allergy.

We adopted a similar approach for extracting mentions of pyelonephritis, searching for notes that contain the term **pyelo**, a common abbreviation for the condition. Here, we only examined notes from the time period immediately preceding the recorded date of specimen collection for an infection, as past instance of pyelonephritis do not prohibit treatment with a 2nd-line agent. As with sulfa allergies, we filtered out notes that contain negated references to pyelonephritis. We examined around 15-20 of the selected notes and successfully verified the validity of the identified contraindication references.

In the training cohort, our filtering procedure finds that 817 (6.9%) examples have sulfa allergies, and 621 (5.2%) examples have mentions of pyelonephritis. In the test cohort, our filtering procedure finds that 259 (6.6%) examples have sulfa allergies, and 298 (7.6%) examples have mentions of pyelonephritis. These are non-trivial subgroups within our cohort, suggesting that treatment policies are likely to be significantly affected by these contraindications.

We compare the performance of our simplified policy from Section 7.2 and the modified IDSA guidelines after accounting for these contraindications. We start with the policies whose performance is shown in Table 7.3. We adjust the simplified policy by selecting the first treatment in the derived preference order for a given example to which the patient has *neither* a strong indicator of resistance nor a contraindication. The guidelines only select between NIT and CIP, and neither of the contraindications considered here prohibit treatment with CIP, so we simply adjust the original guidelines by switching to CIP in cases with relevant contraindications.

We show the results of these adjusted policies in Table 7.6. Overall, accounting for contraindications switches roughly 7% of the original decisions made by the simplified policy and modified guidelines. Both of these policies shift to use additional 2nd-line treatments with a slight reduction in IAT. This is what we would expect, as both contraindications considered here are ones that prohibit certain 1st-line treatments. While this is a substantial increase in 2nd-line treatment for only a small reduction in IAT, the performance of our simplified policy is still better than clinicians or practice guidelines. We reduce IAT and 2nd-line usage rates by 20% and 50% relative to clinicians, and reduce IAT rates by around 8% relative to guidelines at similar levels of 2nd-line usage. Our policy's improvement is robust to contraindications.

We now suggest ways in which deployed CDS tools can account for contraindications in their design. It is not feasible for an automated system to perfectly identify all contraindications from clinical notes, and failing to do so appropriately could result in extreme negative medical consequences for patients. It is inadvisable for the

	No Contraindications		**w/ Contraindications**	
	IAT	**2nd-line**	**IAT**	**2nd-line**
Simple policy	9.7%	10.5%	9.5%	17.7%
Adjusted guidelines	10.8%	9.7%	10.3%	16.4%
Clinicians	11.9%	33.6%	11.9%	33.6%

Table 7.6: Comparison of performance of simple policy and adjusted guidelines before and after accounting for contraindications referenced in clinical notes.

system to attempt to automatically account for contraindications and provide a single treatment recommendation to clinicians. Even a few failures to do so appropriately would likely result in significant loss of trust from the clinician's perspective.

We make two suggestions for how deployed prescription support tools can account for this issue. First, a deployed tool should provide a *ranking* of treatment options, advising the clinician of the next treatment option in the event that there is a known contraindication to the first option. In the case of our simplified policy, this ranking can be derived easily from the preference order produced in the 1st step. Treatments to which the patient has a strong indicator of resistance would be moved to the end of the ranking, and the remaining treatments would appear in the same relative positions as in the suggested preference ordering.

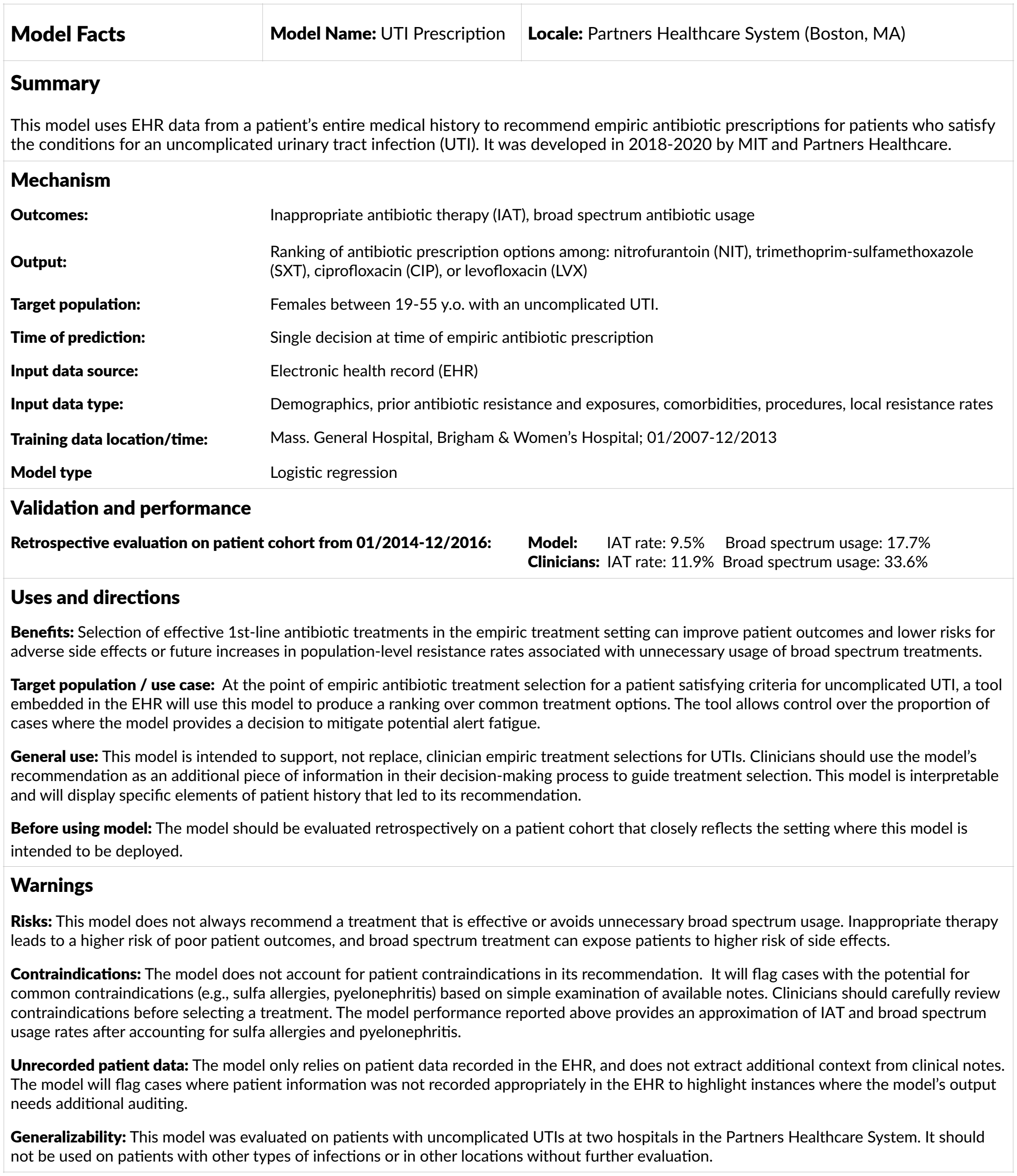
We can also extract such treatment rankings from policies learned via the expected reward maximization and direct policy learning approaches presented in Chapter 4. In the case of expected reward maximization, the ranking would contain treatments in order of the estimated reward for each option. The direct learning approach outputs 'scores' associated with the suitability of each treatment for a given example. Typically, we select the treatment maximizing this score; a treatment ranking would simply list the available options in descending order of these scores.

Second, the tool should flag notes that contain possible mentions of contraindications and display this information to clinicians for review. Examination of specific treatment decisions in our dataset revealed multiple cases where clinicians empirically treated a patient with an antibiotic to which they are allergic, despite records of this allergy in past notes. In the case of sulfa allergies, this was relatively rare - it occurred in less than 3% of the cases we identified - but not nonexistent. Flagging notes that are likely to contain information about contraindications can help mitigate such dangerous mistakes and provide another avenue for improving the quality of care through this decision support tool.

7.5 Summary

In clinical settings, doctors need to know when, where, and how to use the outputs of a machine learning-based decision support tool. A concise way to convey the pertinent information about a model to clinicians interacting with the tool is through a 'Model Facts' sheet, similar to the 'Drug Facts' information included with most medications [44]. We conclude this chapter by presenting a sample 'Model Facts' sheet for the simplified policy for empiric antibiotic prescription we have outlined in this chapter.

This sheet presents the high-level design of our proposed tool, its intended use cases, and the benefits and risks associated with the system. In the "Mechanism" section, we highlight the nature of the training data used to learn the underlying model, necessary data inputs, and the output that will be displayed to a clinician. In the "Uses and direction" section, we emphasize the role of this tool as a support system with the potential to reduce rates of ineffective therapy and broad spectrum usage if used in appropriate settings. We also describe the system's ability to identify features important for decision-making and the clinician's ability to control the volume of alerts produced by the tool. In the "Warnings" section, we explain the risks associated with using this tool in the presence of unrecorded patient data, treatment contraindications, and in new settings beyond the hospital system used for training and evaluation of our models. Designing a tool that meets the criteria described on this sheet and clearly communicating the properties and appropriate usage of the resulting system is likely to significantly improve the chances of successful deployment into regular clinical workflows.



Model Facts	**Model Name:** UTI Prescription	**Locale:** Partners Healthcare System (Boston, MA)

Summary

This model uses EHR data from a patient's entire medical history to recommend empiric antibiotic prescriptions for patients who satisfy the conditions for an uncomplicated urinary tract infection (UTI). It was developed in 2018-2020 by MIT and Partners Healthcare.

Mechanism

Outcomes:	Inappropriate antibiotic therapy (IAT), broad spectrum antibiotic usage
Output:	Ranking of antibiotic prescription options among: nitrofurantoin (NIT), trimethoprim-sulfamethoxazole (SXT), ciprofloxacin (CIP), or levofloxacin (LVX)
Target population:	Females between 19-55 y.o. with an uncomplicated UTI.
Time of prediction:	Single decision at time of empiric antibiotic prescription
Input data source:	Electronic health record (EHR)
Input data type:	Demographics, prior antibiotic resistance and exposures, comorbidities, procedures, local resistance rates
Training data location/time:	Mass. General Hospital, Brigham & Women's Hospital; 01/2007-12/2013
Model type	Logistic regression

Validation and performance

Retrospective evaluation on patient cohort from 01/2014-12/2016:	**Model:**	IAT rate: 9.5%	Broad spectrum usage: 17.7%
	Clinicians:	IAT rate: 11.9%	Broad spectrum usage: 33.6%

Uses and directions

Benefits: Selection of effective 1st-line antibiotic treatments in the empiric treatment setting can improve patient outcomes and lower risks for adverse side effects or future increases in population-level resistance rates associated with unnecessary usage of broad spectrum treatments.

Target population / use case: At the point of empiric antibiotic treatment selection for a patient satisfying criteria for uncomplicated UTI, a tool embedded in the EHR will use this model to produce a ranking over common treatment options. The tool allows control over the proportion of cases where the model provides a decision to mitigate potential alert fatigue.

General use: This model is intended to support, not replace, clinician empiric treatment selections for UTIs. Clinicians should use the model's recommendation as an additional piece of information in their decision-making process to guide treatment selection. This model is interpretable and will display specific elements of patient history that led to its recommendation.

Before using model: The model should be evaluated retrospectively on a patient cohort that closely reflects the setting where this model is intended to be deployed.

Warnings

Risks: This model does not always recommend a treatment that is effective or avoids unnecessary broad spectrum usage. Inappropriate therapy leads to a higher risk of poor patient outcomes, and broad spectrum treatment can expose patients to higher risk of side effects.

Contraindications: The model does not account for patient contraindications in its recommendation. It will flag cases with the potential for common contraindications (e.g., sulfa allergies, pyelonephritis) based on simple examination of available notes. Clinicians should carefully review contraindications before selecting a treatment. The model performance reported above provides an approximation of IAT and broad spectrum usage rates after accounting for sulfa allergies and pyelonephritis.

Unrecorded patient data: The model only relies on patient data recorded in the EHR, and does not extract additional context from clinical notes. The model will flag cases where patient information was not recorded appropriately in the EHR to highlight instances where the model's output needs additional auditing.

Generalizability: This model was evaluated on patients with uncomplicated UTIs at two hospitals in the Partners Healthcare System. It should not be used on patients with other types of infections or in other locations without further evaluation.

Figure 7-4: Sample 'model facts' sheet for a deployed version of the empiric prescription CDS tool outlined in this chapter.

Chapter 8

Conclusion

Rising antibiotic resistance rates pose a major public health challenge, and significant action is needed to mitigate the growth of this issue. Antibiotic stewardship - optimizing the usage of antibiotic agents - is a crucial step towards solving this problem, and the growing availability of EHR data provides the opportunity to learn personalized treatment policies that can reduce unnecessary or inappropriate antibiotic treatments. In this thesis, we have introduced methods for learning antibiotic treatment policies from patient EHR data, with the objective of providing effective treatments while also minimizing usage of broad spectrum antibiotics.

We evaluated these methods in the context of empiric prescriptions for a cohort of patients with uncomplicated UTIs, finding that our techniques can (1) produce dramatic improvement over clinicians and current practice guidelines with respect to both treatment success rates and usage rates of 2nd line treatment and (2) learn policies expressing a wide range of trade-offs between these outcomes. Our methods include both *direct* and *indirect* policy learning approaches; in the uncomplicated UTI setting, both classes of approaches achieve roughly the same performance with respect to clinical outcomes. While the different approaches have similar performance on this cohort, they have trade-offs with respect to other desired properties, including computational efficiency and interpretability of the resulting policies.

We have demonstrated the success of these policy learning approaches in the very specific clinical context of uncomplicated UTI; such a precise cohort definition was necessary for a fair comparison to clinician behaviors. More work is needed to develop similar cohort definitions for other classes of infectious diseases and to understand whether such treatment policies can be used in patients that do not satisfy these specific cohort inclusion criteria.

In this work, we also addressed several real-world considerations needed for successful deployment of a learned treatment policy, including the ability to defer to clinician decisions and improving the interpretability of learned policies and providing transparency surrounding the tool's limitations. Successful integration of these features into a deployed clinical decision support system can significantly ease the pain of introducing new tools into clinical workflows and encourage doctors to use them more frequently.

However, there are numerous other factors that must also be accounted for prior to deployment at a wider scale, including questions of algorithmic bias and fairness - are specific groups of patients receiving far worse treatment than before due to this learned policy? Our current analyses were also limited to data from a single hospital system; it is unclear whether these policies will transport effectively to locations with entirely different populations and resistance rate distributions. Further work is needed to collect similar large-scale microbiological datasets from other locations to better understand the robustness of learned policies to distributional shift.

The work in this thesis also rests on the assumption that antibiotic resistance tests are extremely reliable indicators for a treatment's success on a patient. In practice, this is not necessarily true - patients infected with a pathogen susceptible to their antibiotic treatment may not actually respond to that agent, and vice versa. Our dataset generally does not have long-term follow-up information on patients (especially from outpatient settings), so it is extremely difficult to identify whether patients recovered, and use these real-world outcomes (in place of laboratory test result) in our modeling process. However, other types of data sets, such as an insurance claims database, could potentially provide better longitudinal views of a patient's medical history, allowing us to derive outcomes that directly correspond to treatment success or failure. These outcomes would provide a more accurate view of the clinical impact of our new treatment policies.

More broadly, we believe that the dataset used in this work provides a unique test bed for studying the behavior of policy learning methods. Learning policies in this setting, where one has fully observed outcomes for all treatments of interest, is strictly easier than learning on observational data, where one also has to build estimators of counterfactual treatment outcomes from biased data. In this simplified setting, one can develop and analyze the behavior of policy learning methods with particular desired properties, *independent* of the complexity of estimating unobserved outcomes. In this work, we examined learning policies with multiple objectives and with the ability to defer to an external decision-maker; future work using this type of data could evaluate the robustness of policy learning techniques to nonstationarity or techniques for developing more interpretable policies.

We hope that the work presented in this thesis provides a significant step towards the deployment of clinical decision support tools for antibiotic prescription in a real-world setting, while also providing a useful dataset to motivate future work into more nuanced aspects of policy learning in medical settings.

Appendix A

Tables

A.1 Uncomplicated UTI Cohort Statistics

	Train (2007-13)	**Test (2014-16)**
n (specimens)	11,865	3,941
n (patients)	10,053	3,629
Demographics		
Age - Mean (SD)	34.1 (10.8)	33.6 (11.1)
% White	64.6%	63.0%
Resistance Rates		
NIT	11.2%	11.0%
SXT	19.6%	19.6%
CIP	5.3%	6.4%
LVX	5.1%	6.5%
Prescription Distribution		
NIT	15.9%	34.5%
SXT	41.5%	32.0%
CIP	39.2%	32.5%
LVX	3.3%	1.0%

Table A.1: Training and Test Cohort Statistics.

A.2 Predictive features in resistance models

Category	Feature	Time Window	Coefficient
Prior resistance	NIT	All	0.391
Prior resistance	NIT	180	0.237
Prior resistance	NIT	90	0.220
Comorbidity	Diabetes	180	0.133
Prior organism	Proteus	180	0.107
Prior resistance	NIT	30	0.103
Procedure	Surgery	180	0.097
Prior exposure	Macrolide-lincosamide	30	0.089

Table A.2: Features positively predictive of resistance to NIT

Category	Feature	Time Window	Coefficient
Prior organism	E. coli	180	-0.195
Prior organism	E. coli	90	-0.129
Prior resistance	SXT	All	-0.113
Prior exposure	Penicillin	180	-0.071
Col. pressure - overall	Penicllin	90	-0.065
Col. pressure - floor	Amoxicillin	90	-0.062
Demographics	White		-0.061
Col. pressure - floor	Doxycycline	90	-0.061

Table A.3: Features negatively predictive of resistance to NIT

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	SXT	180	1.104
Prior resistance	SXT	90	0.757
Prior resistance	SXT	All	0.411
Prior exposure	Folate inhibitor	7	0.400
Prior exposure	Folate inhibitor	30	0.328
Prior exposure	SXT	180	0.296
Prior exposure	Clarithromycin	All	0.254
Prior exposure	Pencillin	90	0.250

Table A.4: Features positively predictive of resistance to SXT

Category	**Feature**	**Time Window**	**Coefficient**
Demographics	White		-0.393
Prior resistance	Cefazolin	180	-0.373
Prior organism	E. coli	180	-0.202
Prior resistance	NIT	All	-0.152
Prior resistance	Cefazolin	All	-0.123
Prior resistance	Erythromycin	180	-0.101
Prior resistance	Cefazolin	90	-0.080
Prior resistance	Penicllin	All	-0.050

Table A.5: Features negatively predictive of resistance to SXT

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	CIP	180	1.238
Prior resistance	LVX	All	1.052
Prior exposure	Fluoroquinolone	14	0.663
Prior resistance	CIP	All	0.511
Prior exposure	Fluoroquinolone	180	0.295
Prior exposure	Fluoroquinolone	30	0.280
Prior resistance	CIP	90	0.226
Prior exposure	CIP	90	0.210

Table A.6: Features positively predictive of resistance to CIP

Category	**Feature**	**Time Window**	**Coefficient**
Demographics	White		-0.159
Col. pressure - overall	Ampicillin-sulbactam	90	-0.159
Location	Outpatients		-0.081
Col. pressure - floor	Amoxacillin	90	-0.037
Labs	Lymphocytes	30	-0.026
Labs	Lymphocytes	90	-0.019
Col. pressure - hospital	Doxycycline	90	-0.014
Prior resistance	Cefazolin	All	-0.013

Table A.7: Features negatively predictive of resistance to CIP

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	LVX	All	1.116
Prior resistance	CIP	180	0.779
Prior exposure	Fluoroquinolone	14	0.741
Prior resistance	CIP	All	0.479
Prior resistance	LVX	180	0.406
Prior exposure	Fluoroquinolone	30	0.331
Prior resistance	CIP	90	0.205
Prior exposure	Fluoroquinolone	All	0.173

Table A.8: Features positively predictive of resistance to LVX

Category	**Feature**	**Time Window**	**Coefficient**
Col. pressure - overall	Ampicillin-sulbactam	90	-0.193
Demographics	White		-0.149
Location	Outpatients		-0.084
Col. pressure - overall	Penicillin	90	-0.062
Col. pressure - floor	Amoxicillin	90	-0.062
Prior resistance	Erythromycin	180	-0.025
Labs	Lymphocytes	30	-0.026
Labs	Neutrophils	90	-0.016

Table A.9: Features negatively predictive of resistance to LVX

A.3 Important Features in Direct Policy Models

Category	Feature	Time Window	Coefficient Difference
Prior resistance	SXT	180	0.114
Prior resistance	SXT	All	0.096
Prior resistance	SXT	90	0.089
Prior exposure	SXT	7	0.042
Prior exposure	Folate inhibitor	7	0.042
Prior exposure	Clarithromycin	All	0.036
Prior resistance	GEN	All	0.034
Prior exposure	Cefoxitin	All	0.032

Table A.10: Most important features driving recommendation of NIT over SXT

Category	Feature	Time Window	Coefficient Difference
Prior resistance	NIT	All	0.099
Prior resistance	NIT	90	0.070
Prior resistance	NIT	180	0.065
Demographics	White		0.051
Prior resistance	Cefazolin	180	0.043
Prior resistance	Cefazolin	90	0.039
Prior resistance	NIT	30	0.036
Prior organism	Klebsiella	90	0.033

Table A.11: Most important features driving recommendation of SXT over NIT

A.4 Features predicting policy 2nd line usage

Category	Feature	Time Window	Coefficient Difference
Prior resistance	NIT	All	3.088
Prior exposure	Tetracycline	All	1.710
Comorbidity	Depression	180	1.462
Comorbidity	Obesity		1.344
Procedure	Surgery	180	1.238
Comorbidity	Diabetes	180	1.232
Comorbidity	Obesity	90	0.785
Prior exposure	NIT	All	0.621

Table A.12: Features driving selection of 2nd line treatment over 1st line treatment

Category	Feature	Time Window	Coefficient
Prior resistance	LVX	All	-2.869
Prior organism	E. coli	180	-1.166
Col. pressure - hospital	Penicillin	90	-1.033
Prior resistance	SXT	All	-0.915
Prior exposure	Beta-lactam combo	All	-0.819
Col. pressure - hospital	Oxacillin	90	-0.528
Prior exposure	Fluoroquinolone	All	-0.819
Prior exposure	CIP	All	-0.305

Table A.13: Features driving selection of 1st line treatment over 2nd line treatment

A.5 Features predicting clinician 2nd line usage

Category	Feature	Time Window	Coefficient
Prior exposure	CIP	All	0.404
Prior exposure	Fluoroquinolone	180	0.366
Demographics	White		0.297
Location	ER		0.252
Prior exposure	Fluoroquinolone	90	0.188
Prior resistance	Cefazolin	All	0.177
Labs	White Blood Count	7	0.095
Labs	Neutrophils	14	0.031

Table A.14: Features driving clinician selection of 2nd line treatment

Category	Feature	Time Window	Coefficient Difference
Location	Outpatient		-0.811
Prior resistance	CIP	180	-0.273
Prior resistance	CIP	All	-0.183
Prior exposure	Folate inhibitor	All	-0.130
Prior exposure	Azole	All	-0.070
Prior exposure	NIT	180	-0.067
Prior exposure	Amoxicillin	All	-0.042
Prior exposure	Penicillins	All	-0.039

Table A.15: Features driving clinician selection of 1st line treatment

A.6 Predictive features in resistance models: All UTIs

The features bolded in the following tables indicate features that were not previously identified as positively predictive of resistance to the corresponding agent from anslysis of predictive models trained on only uncomplicated UTIs.

Category	Feature	Time Window	Coefficient
Col. pressure - overall	**NIT**	**90**	0.746
Prior resistance	NIT	30	0.579
Col. pressure - overall	**Cefoxitin**	**90**	0.552
Prior resistance	NIT	All	0.531
Prior resistance	NIT	180	0.469
Prior resistance	NIT	90	0.334
Prior exposure	**Ansamycin**	**180**	0.287
Prior organism	Proteus	90	0.261
Prior organism	**Providencia**	**180**	0.257
Prior resistance	**IPM**	**All**	0.230

Table A.16: Features positively predictive of resistance to NIT

Category	Feature	Time Window	Coefficient
Prior exposure	Cephalosporin	14	-0.273
Prior organism	E. coli	90	-0.269
Col. pressure - overall	Doxycycline	90	-0.244
Prior organism	E. coli	180	-0.232
Prior organism	Enterococcus	30	-0.224
Col. pressure - overall	Oxacillin	90	-0.217
Prior resitance	Vancomycin	180	-0.200
Comorbidity	90	Hypothyroid	-0.180
Prior resistance	Streptomycin	180	-0.147
Prior exposure	Amoxicillin	14	-0.138

Table A.17: Features negatively predictive of resistance to NIT

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	SXT	All	0.864
Prior resistance	SXT	180	0.725
Prior exposure	SXT	30	0.552
Prior exposure	Folate inhibitor	14	0.528
Prior resistance	SXT	90	0.527
Prior resistance	**Tobramycin**	**30**	0.469
Prior exposure	Folate inhibitor	7	0.443
Prior resistance	SXT	30	0.358
Prior resistance	**Ceftriaxone**	**90**	0.317
Prior exposure	**Fluoroquinolone**	**14**	0.266

Table A.18: Features positively predictive of resistance to SXT

Category	**Feature**	**Time Window**	**Coefficient**
Demographics	White		-0.415
Prior organism	Staph coag. neg.	90	-0.385
Prior organism	Streptococcus	180	-0.370
Prior resistance	Cefazolin	All	-0.261
Prior organism	Staph coag. neg.	180	-0.257
Prior resistance	Penicillin	All	-0.253
Prior organism	Klebsiella	180	-0.250
Prior resistance	Cefazolin	90	-0.247
Prior resistance	Moxifloxacin	30	-0.239
Prior exposure	Antifungal	ALL	-0.236

Table A.19: Features negatively predictive of resistance to SXT

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	CIP	All	0.793
Prior resistance	LVX	All	0.594
Prior exposure	Fluoroquinolone	30	0.499
Prior resistance	CIP	180	0.496
Custom	**Nursing home**	**90**	0.434
Prior exposure	Fluoroquinolone	14	0.399
Infection site	**Skin/Soft Tissue**	**180**	0.384
Prior resistance	CIP	90	0.374
Col. pressure - floor	**CIP**	**90**	0.348
Location	**Inpatient**		0.334

Table A.20: Features positively predictive of resistance to CIP

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	Moxifloxacin	180	-0.390
Prior organism	Klebsiella	180	-0.381
Prior resistance	Moxifloxacin	90	-0.359
Prior resistance	SXT	30	-0.357
Prior resistance	Erythromycin	180	-0.342
Prior exposure	Cephalosporin	-0.319	
Comorbidity	Cong. Heart Failure	30	-0.279
Col. pressure - floor	Cefoxitin	90	-0.275
Col. pressure - hospital	Cefoxitin	90	-0.274
Prior organism	E. coli	180	-0.264

Table A.21: Features negatively predictive of resistance to CIP

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	LVX	All	0.449
Prior resistance	CIP	All	0.412
Prior resistance	LVX	180	0.304
Prior exposure	Fluoroquinolone	30	0.290
Prior resistance	CIP	180	0.246
Prior exposure	Fluoroquinolone	14	0.229
Prior resistance	LVX	90	0.219
Prior exposure	Fluoroquinolone	90	0.202
Prior exposure	Fluoroquinolone	180	0.176
Prior exposure	CIP	14	0.167

Table A.22: Features positively predictive of resistance to LVX

Category	**Feature**	**Time Window**	**Coefficient**
Location	Outpatient		-0.530
Col. pressure - overall	Erythromycin	90	-0.315
Col. pressure - hospital	Erythromycin	90	-0.255
Col. pressure - overall	Ampicillin	90	-0.225
Col. pressure - overall	Penicillin	90	-0.216
Col. pressure - overall	Oxacillin	90	-0.210
Col. pressure - hospital	Ampicillin	90	-0.198
Col. pressure - overall	Doxycycline	90	-0.197
Demographics	White		-0.176
Col. pressure - hospital	Penicillin	90	-0.174

Table A.23: Features negatively predictive of resistance to LVX

A.7 Features in Simple Policy

Category	**Feature**	**Time Window**	**Coefficient**
Prior resistance	NIT	All	3.233
Prior resistance	SXT	All	-0.996
Prior resistance	CIP	All	-3.439
Prior resistance	LVX	All	-3.013
Prior exposure	Fluoroquinolone	All	-0.627
Prior exposure	Fluoroquinolone	180	-1.112
Prior exposure	Fluoroquinolone	90	-2.391
Prior exposure	LVX	All	-0.834
Prior exposure	Macrolide-lincosamide	All	1.115
Prior exposure	Penicillins	All	1.020
Prior organism	E. coli	180	-1.284
Procedure	Surgery	180	1.080
Col. pressure - floor	Moxifloxacin	90	-0.991
Col. pressure - hospital	Moxifloxacin	90	-1.323
Col. pressure - floor	Penicillin	90	-0.436
Demographics	White		0.778
Location	Outpatient		0.561
Location	ER		-0.917

Table A.24: Features and corresponding coefficients for linear model f used in Step 1 of simple policy.

Appendix B

Experiment Details

B.1 Synthetic Experiments

The synthetic environment consists of a 10-dimensional feature space and an action space $\mathcal{A}$ with 3 actions. Each feature value is drawn i.i.d. from a standard normal distribution. The feature coefficient values (i.e, α_i, β_i) are selected manually to ensure that the mean outcomes for each action in the dataset are roughly 0.5. All these coefficients have magnitude larger than 1, to ensure that learning approximations to these values is necessary for learning a good predictive model.

In the indirect approach, we train logistic regression models to predict treatment outcomes for each action. We use the `saga` solver in `sklearn`'s logistic regression implementation to train models. Hyperparameters (L1 vs. L2 regularization and regularization strength) are tuned using 10-fold cross validation on the training set. Models are trained for a maximum of 100 iterations.

In the direct approach, the convex surrogate loss is optimized using SGD with a learning rate of 0.1 and L2 regularization with $\lambda = 0.001$. Models are trained for 50 epochs. This model was implemented using PyTorch.

Figure 4-1 shows the outcomes of indirect and direct policy learning using training sets of various sizes on a fixed test set of 10^6 samples drawn from the specified generative model. We only evaluate outcomes on samples where there was at least one ineffective and one effective treatment (i.e, not all 0 or 1 outcomes), as these are the only examples where the policy's decision can affect the outcome. We train both indirect and direct approaches on the same 25 randomly drawn training sets for each sample size, and plot the mean outcome and standard deviations for each setting across these samples in Figure 4-1.

B.2 Indirect Policy Learning

B.2.1 Resistance models

As described in Chapter 5, we use logistic regression models to predict resistance to each antibiotic of interest. These models are trained using the `liblinear` solver in `sklearn`'s logistic regression implementation for a maximum of 1000 iterations. Hyperparameters are tuned to optimize for the average AUC across the validation sets of twenty 70%/30% train/validation splits of the training set. There is no overlap between the individuals with specimens in the train and validation set in each split. We optimize logistic regression hyperparameters over the following grid:

```
{
  ‘C’: [0.001, 0.01, 0.1, 0.5, 1],
  ‘penalty’: [‘l1’, ‘l2’],
  ‘intercept_scaling’: [1, 1000]
}
```

The optimal hyperparameters chosen for each antibiotic are listed in Table B.1.

Antibiotic	**Penalty**	C	**Intercept scaling**
NIT	L2	0.01	1000
SXT	L1	0.1	1000
CIP	L1	0.1	1
LVX	L1	0.1	1

Table B.1: Optimal hyperparameters for resistance models

B.2.2 Thresholding

Our threshold search space is defined implicitly by a fixed set of false negative rates (FNRs) as follows: for each FNR value and antibiotic, the corresponding probability threshold is the one that achieves that FNR rate among the training set resistance predictions for that drug. Given the strong correlation between resistance to CIP and LVX, we constrain $\mathcal{T}$ to combinations where thresholds for CIP and LVX are the same. We use a set of 11 FNR values to define $\mathcal{T}$: [0.001, 0.015, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. Our threshold space $\mathcal{T}$ thus consists of $11^3 = 1,331$ possible combinations.

When a threshold combination results in predictions of resistance for all antibiotic treatments, the policy falls back on a default 1st-line antibiotic. We chose to always default to recommending NIT, since it has a significantly lower resistance rate than SXT in the training set. The optimal threshold combination t_j^* for a given budget constraint b_j is the combination yielding the lowest average IAT rate across the validation sets of twenty 70%/30% train/validation splits of the train cohort,[1] with the constraint that the average 2nd-line usage rate across these splits is no greater than b_j. In Figure 5-1, we show the results of optimal policies π_j for several budget constraint values b_j between 0 and 1. The reported outcomes in Section 5.2 are computed as the mean IAT and 2nd-line usage rates across 20 samples bootstrapped with replacement from the test cohort, where each samples is the same size as the full test set.

B.2.3 Expected reward maximization

The expected reward approach uses the same treatment outcome models used in the thresholding approach. The reported outcomes in Section 5.2 are computed as the mean IAT and 2nd-line usage rates across 20 samples bootstrapped with replacement from the test cohort, where each samples is the same size as the full test set. In Figure 5-1, we show the performance of policies derived using several values of ω in the range $[0.85, 1]$.

[1] These are the same validation splits used for tuning the hyperparameters of the logistic regression models.

B.3 Direct Policy Learning

The direct policy model is parameterized as a linear model and is trained using an Adam optimizer with a learning rate of 0.0001 and a L2 regularization penalty of 0.003. The learning rate and L2 regularization penalty on the model's weights are chosen to optimize the average validation performance across twenty 70%/30% train/validation splits of the training cohort. The number of training epochs is chosen using an early stopping criteria on the mean reward earned on validation set. We find that 50 epochs are sufficient for training. The policy frontier in Figure 1 contains the performance of models learned using values of ω in the range [0.85,1]. We run multiple trials using the same value of ω, and plot the mean IAT and 2nd-line usage outcomes from policies learned across 20 trials.

B.4 Learning from Complicated UTIs

Direct policy models trained on all UTIs were trained for 50 epochs using an Adam optimizer with a learning rate of 0.0001 and a L2 regularization penalty of 0.003. Hyperparameters were chosen using the same procedure described in Appendix B.3.

We now describe the importance weighting scheme used to modify the surrogate loss used in direct learning when training on all UTIs. Since we will only be evaluating our learned models on uncomplicated UTIs, we wish to up-weight the complicated examples in the training data with covariate values more 'similar' to the distribution in uncomplicated UTIs. Formally, we wish to learn importance weights β_i and optimize the following modification of the surrogate loss in Equation 4.9:

$$\hat{L}_{imp}(f) := -\sum_{i=1}^{n_{uncomp}} \sum_{a \in \mathcal{A}} r_i(a) \log \frac{\exp f_a(x_i)}{\sum_{a'} \exp f_{a'}(x_i)} - \sum_{i=1}^{n_{comp}} \beta_i \sum_{a \in \mathcal{A}} r_i(a) \log \frac{\exp f_a(x_i)}{\sum_{a'} \exp f_{a'}(x_i)} \tag{B.1}$$

where n_{comp}, n_{uncomp} are the number of complicated and uncomplicated examples in our training set, respectively. We use *kernel mean matching*, a method for adjusting for covariate shift between train and test sets by computing importance weights that seeks to match covariate distributions between the two datasets in a high-dimensional feature space [14]. In our case, the training examples correspond to complicated UTIs, while the test examples correspond to uncomplicated UTIs. We compute these weights β_i by solving the following convex optimization problem:

$$\min_{\beta} \frac{1}{2}\beta^T K\beta - \kappa^T\beta \text{ subject to } \beta_i \in [0, B] \text{ and } \left|\sum_{i=1}^{n_{comp}} \beta_i - n_{comp} \leq n_{comp}\epsilon\right|, \tag{B.2}$$

where K is the RBF kernel matrix for the data from complicated UTIs and κ_i is given by the expression:

$$\kappa_i = \frac{n_{uncomp}}{n_{comp}} \sum_{j=1}^{n_{uncomp}} k(x_i^{comp}, x_j^{uncomp}),$$

where $k(x_i^{comp}, x_j^{uncomp})$ is the value of the RBF kernel between the two specified examples. We choose $B = 3$ in Equation (B.2), and solve for the importance weights using `cvxopt`. We found that applying these importance weights to the loss function yielded small improvements in policy performance on the validation set comparing to the unweighted loss function and thus chose to retain this modification for learning the final policies for evaluation on the test set.

B.5 Learning to Defer

B.5.1 Direct learning with deferral

We first describe the experiment details for learning policies in the setting where algorithm errors are costly. In this setting, the reward for deferral is given by:

$$\begin{aligned} r_\omega(\texttt{defer}) = r_\omega(a^{doc}) + \omega \cdot \mathbf{1}\left[a^{doc} \text{ results in IAT}\right] + \\ 0.5 \cdot (1-\omega) \cdot \mathbf{1}\left[a^{doc} \text{ is 2nd-line}\right] - \lambda, \end{aligned}$$

where $r_\omega(a^{doc})$ is the original reward for the action taken by the clinician as defined by Equation 4.4. The second and third terms in this reward introduce the asymmetric penalty for suboptimal decisions made by clinicians, providing additional reward for taking decisions that result in IAT or 2nd-line treatment, respectively. In our work, $\omega > 0.8$ in all experiments, so we are introducing a larger asymmetry in the penalty incurred by IAT than by 2nd-line treatment.

We learn direct policy models for several values of ω in the range $[0.88, 0.94]$, setting $\lambda = 0.08$. In each case, we first train for 25 epochs without including the deferral option in the loss function, then train for 50 epochs with the reward for deferral. Models are trained using an Adam optimizer with learning rate of 0.0001 and L2 regularization of 0.003. For each learned model, we then derive the policy for a given coverage rate c using the procedure described in Section 6.4.1.

In settings where algorithm interventions are costly, the reward for deferral is:

$$r_\omega(\texttt{defer}) = r_\omega(a^{doc}) + \lambda.$$

We learn direct policy models for several values of ω in the range $[.88, .94]$ and for several values of ω in the range $[0, 0.10]$. The training procedure and hyperparameters are the same as described for the setting where errors are costly. In both settings, the reported values are average IAT and 2nd-line usage rates across 25 trials for each setting of the reward function parameters.

B.5.2 Expected reward maximization

We use the same definition of the reward functions for deferral as described in the previous section for both settings. The expected reward predictions for treatment actions other than deferral are derived using the same treatment outcome models learned as described in Section B.2.1.

In the setting where errors are costly, we derive policies for several values of ω in the range $[0.88, 0.94]$ and λ in the range $[0.04, 0.12]$. In the setting where interventions are costly, we derive policies for several values of ω in the range $[0.88, 0.94]$ and λ in the range $[0.0, 0.10]$. For each setting of these parameters in the reward function, we fit a L1 regularized linear regression model to predict the reward earned by choosing to defer, tuning the regularization strength to minimize the mean squared error in predicting the reward on a validation set.

In both settings, we report the mean IAT and 2nd-line usage rates across 20 samples bootstrapped with replacement from the test set.

B.5.3 Rejector-based deferral

We first learn policies via the direct learning approach for several values of ω in the range $[0.88, 0.94]$ using the original surrogate loss in Equation 4.9. In the setting where algorithm errors are costly, we then train a regularized logistic regression model as our rejector to predict where clinicians either use ineffective treatment or unnecessarily use 2nd-line treatment. In the setting where interventions are costly, we learn separate rejectors for each value of ω used to learn the initial policy models, as the target labels for depends on where the original policy makes an error. We tune hyperparameters for both rejectors over the same grid in Section B.2.1 on a validation set.

B.6 Constructing a Simplified Policy

The linear model f used for selection of the treatment preference order in Step 1 consists of the most important features for selection between 1st- and 2nd-line treatments. To learn f, we first train a model using direct policy learning on *all* features, setting $\omega = 0.92$ in the reward function. We train on the dataset of all UTIs with an importance-weighted loss function as described in Section 5.5. We then fit a logistic regression model with extremely high $L1$ regularization ($C = 0.001$) to predict where this direct policy uses 2nd-line treatment to obtain f. This regularization penalty was tuned on the validation set to yield a model that had a small number of nonzero coefficients while also achieving comparable performance to the full policies when used as part of the simplified policy.

Appendix C

Proofs

C.1 Theoretical Results for Direct Policy Learning

In this section we provide a self-contained proof of the consistency of our chosen loss function, which is known as the multinomial deviance loss [17] in the literature on multi-category cost-sensitive classification with convex surrogates. First, we note the following fact

Proposition 2. *The function $\mathbb{E}_{r|x}\tilde{L}(f, x, r)$ is convex in f for nonnegative rewards* $\mathbf{r}$

Proof. The expectation $\mathbb{E}_{r|x}$ preserves convexity, so we just need to confirm that $\tilde{L}(f, x, r)$ is convex in f, which we can do so by rewriting as

$$\tilde{L}(f, x, r) = \sum_{a \in \mathcal{A}} r(a) \left[\left(\log \sum_{a'} \exp f_{a'}(x) \right) - f_a(x) \right] \tag{C.1}$$

The inner term is a convex function of f because it is a non-negative sum of convex functions of f, namely log-sum-exp and $-f$. The outer sum is a non-negative sum, since the rewards are specified to be non-negative, which preserves convexity. □

Proposition 3. *For non-negative rewards* $\mathbf{r}$*, and for an f^* that satisfies $f^* = \inf_f \mathbb{E}_{x,r}\tilde{L}(f, x, r)$, the corresponding policy $\pi^*(x) = \arg\max_a f^*_a(x)$ is equivalent to the the Bayes-optimal policy $\pi^*(x) = \arg\max_a \mathbb{E}[r(a)|X]$*

Proof. First, we can write this as

$$\inf_f \mathbb{E}_{x,r}\tilde{L}(f,x,r) = \mathbb{E}_x \inf_{f(x)} \mathbb{E}_{r|x}\tilde{L}(f,x,r)$$

Because $\mathbb{E}_{r|x}\tilde{L}(f,x,r)$ is convex in f (see Proposition 2), we just need to find a critical point where $\frac{\partial}{\partial f_{a^*}}\tilde{L}(f(x),x,r) = 0, \ \forall a^* \in \mathcal{A}$. We can see that

$$\begin{aligned}
&\frac{\partial}{\partial f_{a^*}}\mathbb{E}_{r|x}\tilde{L}(f,x,r) \\
&= -\frac{\partial}{\partial f_{a^*}}\sum_{a\in\mathcal{A}}\mathbb{E}[r(a)|X]\log\frac{\exp f_a(x)}{\sum_{a'}\exp f_{a'}(x)} \\
&= -\frac{\partial}{\partial f_{a^*}}\mathbb{E}[r(a^*)|X]\log\frac{\exp f_{a^*}(x)}{\sum_{a'}\exp f_{a'}(x)} \\
&- \frac{\partial}{\partial f_{a^*}}\sum_{a\neq a^*}\mathbb{E}[r(a)|X]\log\frac{\exp f_a(x)}{\sum_{a'}\exp f_{a'}(x)} \\
&= -\mathbb{E}[r(a^*)|X]\left[1 - \frac{\exp f_{a^*}}{\sum\exp f_{a'}}\right] + \sum_{a\neq a^*}\mathbb{E}[r(a)|X]\frac{\exp f_{a^*}}{\sum_{a'}\exp f_{a'}} \\
&= -\mathbb{E}[r(a^*)|X] + \frac{\exp f_{a^*}(x)}{\sum_{a'}\exp f_{a'}(x)}\sum_a\mathbb{E}[r(a)|X] = 0 \\
\implies &\frac{\mathbb{E}[r(a^*)|X]}{\sum_a\mathbb{E}[r(a)|X]} = \frac{\exp f_{a^*}(x)}{\sum_{a'}\exp f_{a'}(x)}
\end{aligned}$$

From this, we can see that

$$\begin{aligned}
\arg\max_{a\in\mathcal{A}}\mathbb{E}[r(a)|X] &= \arg\max_{a\in\mathcal{A}}\frac{\mathbb{E}[r(a)|X]}{\sum_{a'}\mathbb{E}[r_{a'}|X]} \\
&= \arg\max_{a\in\mathcal{A}}\frac{\exp f_a}{\sum_{a'}\exp f_{a'}} \\
&= \arg\max_{a\in\mathcal{A}}\log\left(\frac{\exp f_a}{\sum_{a'}\exp f_{a'}}\right) \\
&= \pi^*(x)
\end{aligned}$$

Completing the proof that at optimality, the optimal

$$f^* = \arg\inf_{f(x)}\mathbb{E}_{r|x}\tilde{L}(f,x,r)$$

yields a calibrated decision rule $\pi^*(x)$ □

We make two minor remarks: First, as a practical matter, we drop the usual constraint (used to ensure uniqueness) that $\sum_a f_a(x) = 0$, as we impose ℓ_2 regularization on the weights of our $f_a(x) = \theta_a^T x$ functions in our experiments. Second, this formulation requires that the reward vector $\mathbf{r}$ is non-negative, but this can be relaxed in a straightforward way by replacing $r(a)$ with $\max_{a'} r(a') - r(a)$. We tried this latter formulation in our experiments and it did not have a significant impact on results.

Bibliography

[1] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. *Proceedings of the 25th International Conference on Machine Learning*, pages 41–47, 2008.

[2] Peter L Bartlett and Marten H Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(Aug):1823–1840, 2008.

[3] Y.-S. Chen S.-H. Lee Y.-S. Chen S.-C. Chen S.-C. Chang C.-C. Lee, M.-T. G. Lee. Risk of aortic dissection and aortic aneurysm in patients taking oral fluoroquinolone. *JAMA Internal Medicine*, 175, 2015.

[4] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. " hello ai": Uncovering the onboarding needs of medical practitioners for human-ai collaborative decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24, 2019.

[5] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

[6] CLSI. M100 performance standards for antimicrobial susceptibility testing. 2017.

[7] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016.

[8] K. E. Dingle, X. Didelot, and T.P. Quan. Effects of control interventions on clostridium difficile infection in england: an observational study. *Lancet Infectious Diseases*, 17:411 – 421, 2016.

[9] Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(May):1605–1641, 2010.

[10] Charles Elkan. The foundations of cost-sensitive learning. *IJCAI International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[11] Centers for Disease Control. Antibiotic use in the united states, 2017: Progress and opportunities. 2017.

[12] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Multiple reject thresholds for improving classification reliability. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 863–871. Springer, 2000.

[13] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pages 4878–4887, 2017.

[14] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching.

[15] K. Gupta, T.M. Hooton, K.G. Naber, B. Wullt, R. Colgan, L.G. Miller, G.J. Moran, L.E. Nicolle, R. Raz, A.J. Schaeffer, D.E. Soper, Infectious Diseases Society of America, and European Society for Microbiology / Infectious Diseases. International clinical practice guidelines for the treatment of acute uncomplicated cystitis and pyelonephritis in women: A 2010 update by the infectious diseases society of america and the european society for microbiology and infectious diseases. *Clinical Infectious Diseases*, 52, 2011.

[16] Robin Henderson, Phil Ansell, and Deyadeen Alshibani. Regret-regression for optimal dynamic treatment regimes. *Biometrics*, 66(4):1192–201, Dec 2010.

[17] Xinyang Huang, Yair Goldberg, and Jin Xu. Multicategory individualized treatment regime using outcome weighted learning. *Biometrics*, (August 2018):1216–1227, 2019.

[18] Timothy Jancel and Vicky Dudas. Management of uncomplicated urinary tract infections. *Western Journal of Medicine*, 176(1):51, 2002.

[19] Sarah Kabbani, Adam L Hersh, Daniel J Shapiro, Katherine E Fleming-Dutra, Andrew T Pavia, and Lauri A Hicks. Opportunities to improve fluoroquinolone prescribing in the united states for adult ambulatory care visits. *Clinical Infectious Diseases*, 67(1):134–136, 2018.

[20] Saif Khairat, David Marc, William Crosby, and Ali Al Sanousi. Reasons for physicians not adopting clinical decision support systems: critical analysis. *JMIR medical informatics*, 6(2):e24, 2018.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Marin H Kollef. The importance of appropriate initial antibiotic therapy for hospital-acquired infections. *The American journal of medicine*, 115(7):582–584, 2003.

[23] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[24] Surbhi Leekha, Christine L Terrell, and Randall S Edson. General principles of antimicrobial therapy. In *Mayo Clinic Proceedings*, volume 86, pages 156–167. Elsevier, 2011.

[25] Thomas Lengauer and Tobias Sing. Bioinformatics-assisted anti-hiv therapy. *Nature Reviews Microbiology*, 4(10):790–797, 2006.

[26] Hsi Liu, Thomas H. Taylor, Kevin Pettus, Stev Johnston, John R. Papp, and David Trees. Comparing the disk-diffusion and agar dilution tests for neisseria gonorrhoeae antimicrobial susceptibility testing. *Antimicrobial Resistance and Infection Control*, 5(1):1–6, 2016.

[27] Daniel J. Lizotte, Michael Bowling, and Susan A. Murphy. Linear fitted-Q iteration with multiple reward functions. *Journal of Machine Learning Research*, 13(1):3253–3295, 2012.

[28] David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, pages 6147–6157, 2018.

[29] Susan A Mehnert-Kay. Diagnosis and management of uncomplicated urinary tract infections. *American family physician*, 72(3):451–456, 2005.

[30] Richard V Milani, Jonathan K Wilt, Jonathan Entwisle, Jonathan Hand, Pedro Cazabon, and Jefferson G Bohan. Reducing inappropriate outpatient antibiotic prescribing: normative comparison using unblinded provider reports. *BMJ Open Quality*, 8(1), 2019.

[31] Qian Min and S.A. Murphy. Performance guarantees for individualized treatment rules. *The Annals of Statistics*, 39:1180–1210, 2011.

[32] Jose M. Munita and Cesar A. Arias. Mechanisms of antibiotic resistance. *Microbiology Spectrum*, 4(2), 2016.

[33] Inbal Nahum-Shani, Min Qian, Daniel Almirall, William E Pelham, Beth Gnagy, Gregory A Fabiano, James G Waxmonsky, Jihnhee Yu, and Susan A Murphy. Q-learning: a data analysis method for constructing adaptive interventions. *Psychological methods*, 17(4):478–94, Dec 2012.

[34] Karen C Nanji, Diane L Seger, Sarah P Slight, Mary G Amato, Patrick E Beeler, Qoua L Her, Olivia Dalleur, Tewodros Eguale, Adrian Wong, Elizabeth R Silvers, et al. Medication-related clinical decision support alert overrides in inpatients. *Journal of the American Medical Informatics Association*, 25(5):476–481, 2018.

[35] Sriraam Natarajan and Prasad Tadepalli. Dynamic preferences in multi-criteria reinforcement learning. *Proceedings of the 22nd International Conference on Machine Learning*, pages 601–608, 2005.

[36] Eric W Nawar, Richard W Niska, and Jianmin Xu. National hospital ambulatory medical care survey: 2005 emergency department summary. 2007.

[37] Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. On the calibration of multiclass classification with rejection. In *Advances in Neural Information Processing Systems*, pages 2582–2592, 2019.

[38] Mathupanee Oonsivilai, Yin Mo, Nantasit Luangasanatip, Yoel Lubell, Thyl Miliya, Pisey Tan, Lorn Loeuk, Paul Turner, and Ben Cooper. Using machine learning to guide targeted and locally-tailored empiric antibiotic prescribing in a children's hospital in cambodia. *Wellcome Open Research*, 3, 2018.

[39] World Health Organization. Global action plan on antibiotic resistance. 2015.

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[41] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

[42] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[43] Philip Schulte, Anastasios Tsiatis, Eric Laber, and Marie Davidian. Q- and a-learning methods for estimating optimal treatment regimes. *Statistical Science*, 29(4):640–661, 2014.

[44] Mark P Sendak, Michael Gao, Nathan Brajer, and Suresh Balu. Presenting machine learning model information to clinical end users with model facts labels. *NPJ Digital Medicine*, 3(1):1–4, 2020.

[45] TJ Stewart. A critical survey on the status of multiple criteria decision making theory and practice. *Omega*, 20(5-6):569–586, 1992.

[46] Timothy Sullivan, Osamu Ichikawa, Joel Dudley, Li Li, and Judith Aberg. The Rapid Prediction of Carbapenem Resistance in Patients With Klebsiella pneumoniae Bacteremia Using Electronic Medical Record Data. (September 2012):1–6, 2016.

[47] Liyuan Tao, Chen Zhang, Lin Zeng, Shengrong Zhu, Nan Li, Wei Li, Hua Zhang, Yiming Zhao, Siyan Zhan, and Hong Ji. Accuracy and effects of clinical decision

support systems integrated with bmj best practice–aided diagnosis: Interrupted time series study. *JMIR medical informatics*, 8(1):e16912, 2020.

[48] Ambuj Tewari and Peter L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

[49] M Cristina Vazquez-Guillamet, Rodrigo Vazquez, Scott T Micek, and Marin H Kollef. Predicting Resistance to Piperacillin-Tazobactam , Cefepime and Meropenem in Septic Patients With Bloodstream Infection Due to Gram-Negative Bacteria. 65(November):1607–1614, 2017.

[50] C Lee Ventola. The antibiotic resistance crisis. *Pharmacy and Therapeutics*, 40(4):277–283, 2015.

[51] C. Vira and Y. Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. North-Holland, 1983.

[52] Qian Yang, Aaron Steinfeld, and John Zimmerman. Unremarkable ai: Fitting intelligent decision support into critical, clinical decision-making processes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2019.

[53] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation. In H Wallach, H Larochelle, A Beygelzimer, F dAlché Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14610–14621. Curran Associates, Inc., 2019.

[54] Idan Yelin, Olga Snitser, Gal Novich, Rachel Katz, Ofir Tal, Miriam Parizade, Gabriel Chodick, Gideon Koren, Varda Shalev, and Roy Kishony. Personal clinical history predicts antibiotic resistance of urinary tract infections. *Nature Medicine*, 25(July), 2019.

[55] M. Zeleny and J. L. Cochrane. *Multiple Criteria Decision Making*. McGraw-Hill, New York, NY, USA, 1982.

[56] Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004.

[57] Ying-qi Zhao, Eric B Laber, and Bruce E Sands. Efficient augmentation and relaxation learning for individualized treatment rules using observational data. *Journal of Machine Learning Research*, 20:1–23, 2019.

[58] Yingqi Zhao, Donglin Zeng, A. John Rush, and Michael R. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107:1106–1118, 2012.

[59] Hui Zou, Ji Zhu, and Trevor Hastie. New multicategory boosting algorithms based on multicategory fisher-consistent losses. *Annals of Applied Statistics*, 2(4):1290–1306, 2008.